國立臺灣大學電機資訊學院電子工程學研究所

博士論文

Graduate Institute of Electronics Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Doctoral Dissertation

用於多相機追蹤系統之高效與精確的行人重識別技術研究

Learning Efficient and Effective Person Re-identification in Multi-Camera Tracking System and Beyond

劉致廷

Chih-Ting Liu

指導教授：簡韶逸 博士

Advisor: Shao-Yi Chien, Ph.D.

中華民國 111 年 07 月

July 2022

doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

國立臺灣大學博士學位論文
口試委員會審定書

用於多相機追蹤系統之高效與精確的行人重識別技術研究

Learning Efficient and Effective Person Re-identification in Multi-Camera Tracking System and Beyond

本論文係劉致廷君（F06943014）在國立臺灣大學電子工程學研究所完成之博士學位論文，於民國 111 年 7 月 27 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

簡韶逸（指導教授）	
陳銘憲	王鈺強
莊永裕	
曹昱	賴尚宏
楊家驤	陳駿丞

所長：

doi:10.6342/NTU202202005

致謝

一段漫長的旅程終於結束了。碩逕博的五年就像雲霄飛車一般，經歷了高峰與低谷。或許五年說長也不長，但我知道這些都是如此的得來不易，從喜歡研究到失去熱情，最後又再次因為遇見許多貴人而找到自己的價值，找到解決問題時的那種成就感。一路上真的要感謝非常多人，回首從我大學做專題的那年，就能在每個人生的階段都遇到影響我很深的人，真的是如此的幸運。

首先想感謝我的指導教授簡韶逸老師。從大二修老師開的交電，到後來辦活動需要跟當時是副主任的簡老師交流，就深深感受到老師的個人魅力。老師並不是一個會跟你娓娓道來一些人生大道理的人，卻能在簡單的談話中了解到老師的處世哲學，或許就是因為如此親民與日常的講話方式，讓身為學生的我感受到被信任也讓我完全可以信任這位老師。認識老師到現在八年了，老師依舊如此親切，除了指導教授的身分外，在我心中老師就像我的大哥，當我迷惘的時候總是能從老師身上得到一些精神上的幫助。還記得找老師談逕讀博士的時候，老師說到他不太會影響博士要做什麼，他希望博士生要深度的認識自己知道自己想做什麼。這五年間，老師真的給我了非常大的自由度也盡其所能地給我幫助，不論是經費設備上的，還是嘗試各方面的實習，或是在研究上對我的進度與方向規劃，簡老師都不會設限我應該怎麼做。我想我能在最後成為了我心目中那個理想的博士也是因為找對了指導教授吧！

再來就是一路上指導過我的教授們。我最想感謝的是真的像大哥一般的李宏毅老師，如果沒有您生動的教學內容，我一定不會踏上機器學習這條路，也不會下定決心一定要往軟體的方向走。從一開始聽聞有老師電路學上課打電路的槍戰遊戲給大家看，到網路上自學您開的機器學習，接著面對面跟著老師一起做了兩年的專題研究，到後來即使不是老師的研究生也還是會在路上巧遇時閒聊幾句。想想能在對的時間認識這位十一萬訂閱的 Youtuber 真的是太榮幸了! 再來也是在我初踏上這條路的恩師李琳山教授。大四找研究所的時候其實很猶豫不知道要找簡老師還是李老師，還記得那天進到論文堆到天花板的李老師辦公室深談了許久，老師告訴我了一句最重要的話：「不要害怕選擇，有選擇是一件幸福的事！」這句話從此影響我深遠，讓我勇敢的做每一次的選擇，讓我深深相信，沒有錯的選擇，過去的每一次選擇都是造就了現在的你。接著我想感謝在我博士期間指導過我也一起合作論文的王鈺強教授、陳祝嵩教授還有陳駿丞博士。我的研究能力、寫論文的能力其實都並不如世界各地頂尖的研究者，也不是典型在各大頂尖會議發表論文的博士生，但是不論是從博士初期花了不少心力指導我論文能力的王教授到後來花了

doi:10.6342/NTU202202005

許多時間陪我摸索新方向的陳教授與陳博士，他們都不厭其煩地給了我很多很多的觀念與技術，讓我有機會知道我還有什麼不足，應該要增進什麼技能。最後，我想感謝在我三次實習中的主管黃毓文博士、張毓麟博士、賴尚宏教授還有 Mentor 王建詒，每一次的實習都讓我知道未來想要什麼不要什麼，尤其是在最後那段微軟實習的日子，剛好是我在人生低潮時的解藥，這段時間拋開了實驗室的雜事，讓我在如此自由的環境中重新的認識自己，對於自己的價值有更多的了解，也更清楚我該追求什麼。

再來我想先感謝我的父母與家人，生活中交集雖然不多，但這一路以來，總是全力支持我人生路上的各種決定。沒有你們過去的栽培，我也無法進到影響我最深的建中與台大。我也從來不用擔心是否需要額外打工幫忙家裡，因為你們總把家裡打理得好好的。 接著我想謝謝人生路上與我有交集的朋友們，最重要的還是那群研究所還留在台大的大學同學們，安薇、志軒、魁哥、奕達、柏翔、宗宏等等(太多了啦)，生活中能與這些人一起玩樂一起創造回憶大大消除了讀博士的各種壓力與憂鬱。還有謝謝實驗室的同學們，剛進來就跟研究狂致緯學長一起做事，也因為一起參加了兩次 CVPR workshop 讓我對於研究的世界有更多的憧憬，偉志跟裕盛學長則是兩位博士模板，讓我在要簽博的時候有對象可以學習。還有兩位我剛讀博士之後一起合作的碩士生，曼妤跟禹澄，我想你們兩個應該是我最認真帶的人了吧，一起投了兩篇論文，一起出國，一起重訓，讓我博士初期過得很充實。還有後來的在賢、昱愷、凱翔、子傑、紹軒，在博理四樓最歡樂的實驗室中，跟你們一起聊天打屁，一起去邦食堂吃兩個小時，一起去宜蘭玩，真的是讓害怕孤獨的我有了很多依靠。當然還有沒提到的汶璁，我想你應該是我博士後期在 lab 最重要的人了(笑)。最後的那段日子，小時候不能玩的世紀帝國恰巧來到我的世界，時常跟你討論 Viper 又有多神，時常跟你和奕達決戰真的讓我在苦悶的博士生涯有了很多樂趣。

最後的最後，我想感謝我自己，只有你自己最知道你經歷了什麼，還記得曾經煩躁地望著每天就會多出好幾篇 SOTA 的論文，曾經在身邊的朋友一個一個畢業之後感到孤獨、曾經下定決心走進韶逸辦公室後又被心靈大師說服重拾信心。感謝我自己從來沒有放棄去發掘自己的價值，也謝謝我在最後努力的朝向各種夢幻的工作前進、努力，希望我能保持這樣的信念繼續的走下去！

2022.07.27 劉致廷

doi:10.6342/NTU202202005

中文摘要

不論是如智慧家庭般的小系統，到如校園的中規模系統，或是到一個城市這樣的大規模系統下，攝影機無所不在。 有了這些相機，我們可以建構一個智慧的環境。 「多目標多相機追蹤」就是其中最關鍵的技術，它可以追蹤行人經過不同相機之後的軌跡，有了這些軌跡，我們就可以進而分析在這個環境下每個人的行走模式。 因為「多目標多相機追蹤」是一個複雜的問題，在我們的博士論文中，較著重在其中的子研究領域，稱作「行人重識別」。行人重識別目的在給定兩個已經產生好的跨相機行人定界框後，根據圖片外觀來判斷是否為同一個人。一個好的行人重識別技術可以直接的影響到整體「多目標多相機追蹤」的表現。在我們的博士論文中，我們專注於實際世界會有的情況，像是運算量跟表現上的取捨，或是在沒有人工標籤的情況下學習行人重識別模型。首先第一部分，因為在實際場域下較常處理視訊序列而非單一張影像，因此我們著重於視訊行人重識別。我們設計了一個創新的自注意力機制的架構，它可以學習空間與時間中該專注的部分。接著我們提出了一個基於空間與時間上優化的輕量架構版本，使其可以在相似的表現下降低了硬體的耗能與運算量。另外，我們探討了目前現有資料集的問題。我們提出了一個簡單卻有效的前處理方式來減少在資料集中的雜訊與錯誤，可以幫助正在做此方向研究的研究員不再因為資料集的錯誤而無法提出有效的解決方法。第二部分，我們專注於處理半監督式學習的行人重識別，也就是資料集中只有少部分的資料有標籤。我們提出了一個創新的分群機制，它可以根據有標籤的資料分布來幫助在無標籤的資料上正確地分群，進而利用分群後偽標籤來學習模型。第三部分，我們希望學習非監督式的行人重識別，也就是在目標環境並且所有資料都沒有標籤的情況下來學習模型。我們依舊是採用分群方式來給定資料偽標籤，但是提出了兩個創新的修正機制來修正本來因為分群錯誤而產生的錯誤偽標籤。

另一方面，在實際情況下常常因為硬體限制而無法順利的運行複雜的神經網路模型，因此，「濾波器剪枝」就是一個可以移除不重要的濾波器的解決方式。在我們的博士論文中，我們提出兩個剪枝的方式，第一種是層向剪枝。我們會根據每一層對損失函數的影響來定義每一層的敏感度，接著會從最不敏感的層來做剪枝。第二種是全局剪枝，也就是全局地估測每個濾波器的重要性。特別的是，我們提出要把重要性估測結合每個濾波器對目標硬體資源的影響，這樣在最終目標資源下，我們可以更準確的估測每個濾波器的重要性。 最後，結合我們所提出的剪枝與行人重識別技術，我們建構了一個及時多目標多相機追蹤系統，這個系統利用一台電腦模擬真實環境下分散式運算的狀況，來執行行人偵測、行人追蹤與行人重識別。在我們提出的運算優化方法下，此系統可達到即時的運行，也就是每秒可以處理超過三十個幀。通過大量的實驗，我們所有提出的方法同時具有準確性與計算效率，可以很有效的部屬到真實場域中。

doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

Learning Efficient and Effective Person Re-identification in Multi-Camera Tracking System and Beyond

Chih-Ting Liu
Advisor: Prof. Shao-Yi Chien

Graduate Institute of Electronics Engineering
National Taiwan University
Taipei, Taiwan

July 2022

doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

Learning Efficient and Effective Person Re-identification in Multi-Camera Tracking System and Beyond

By

Chih-Ting Liu

Dissertation

Submitted in partial fulfillment of the requirement
for the degree of Doctor of Philosophy
in Electronics Engineering
at National Taiwan University
Taipei, Taiwan, R.O.C.

July 2022

Approved by：

Approved by：

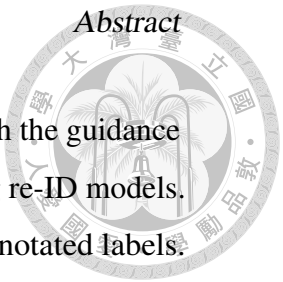
Approved by Director：

doi:10.6342/NTU202202005

Abstract

Surveillance cameras are seen everywhere in the world, which can be embedded into a small system, such as a smart home, a smart campus, or to a large system like a smart city. With the cameras, we can enable the intelligence. Multi-Target Multi-Camera Tracking (MTMCT) plays a critical role in the core techniques. It aims at tracking multiple people captured under different camera views. With MTMCT, we can extract the walking trajectories of some specific people and further analyze the patterns of them. Since MTMCT is a complicate problems, we specifically focus on the sub-problem that is suitable for research, which called *Person Re-identification* (re-ID). Re-ID aims at matching two cropped pedestrians under different cameras with only appearance cues. The performance of the re-ID will explicitly influence that of the MTMCT system. In this dissertation, we address multiple aspects of re-ID, and especially focus more on the real-world scenarios, such as the trade-off between computation and performance, or how to learn under data without labels. The first part is for video-based re-ID. In the system, it is more common to match two pedestrians with their image sequence along time. We demonstrate a novel model architecture for learning self-attention across space and time and propose a spatially and temporally efficient version that can maintain the performance but with a more light-weight structure. Then, we also explore the problems in the existing benchmark for data and evaluation metrics. We further propose an easy pre-processing technique to reduce the noise in the dataset and help the community focus on extracting invariant visual appearance. The second part is for learning re-ID with only few labeled data, which called semi-supervised

doi:10.6342/NTU202202005



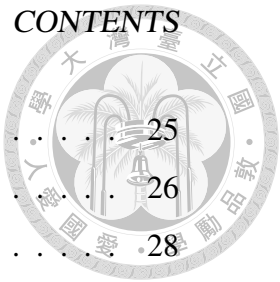
re-ID. We adopt novel clustering methods on the unlabeled data with the guidance of the labeled ones to progressively learn pseudo-labels for training re-ID models. The third part is for learning the re-ID model even without any annotated labels. This work simplifies the problem into cross-domain re-ID that we have data with labels in the source domain and aim to learn the model on data totally unlabeled in target domain. We propose two rectification mechanisms that can help clean the original noise generated from the pseudo-labels of typical clustering algorithm.

On the other hand, for a practical system in our life, if we cannot perform a model in real-time owing to the hardware constraints, "Network Filter Pruning" is a solution to remove unimportant filters in a complicated neural network. In this dissertation, we propose two kinds of pruning techniques. The first one is called layer-wise pruning. We measure the sensitivity of each layer, which means the impact on loss of a unit weight in that layer, and start pruning on the less sensitive layer. The other technique focuses on global pruning, which measures the importance of each filter at the same time and remove the less important ones. Specifically, in this work, we combine the importance estimation with the hardware constraints, which makes it more accurate based on hardware impact of each weight. With the pruning technique, we combine them with the proposed re-ID algorithms. We build a real-time MTMCT system on one machine to simulate distributed multiple cameras in an environment that perform pedestrian detection, tracking and re-identification at the same time. With all the proposed techniques, we can largely reduce all the complicated computation in neural network and make the whole system operate in real-time, which achieves larger than 30 FPS. The proposed algorithms are all quantitatively and qualitatively evaluated in various benchmarks on re-ID and image classification. Experimental results all show that our techniques are efficient and effective in these applications.

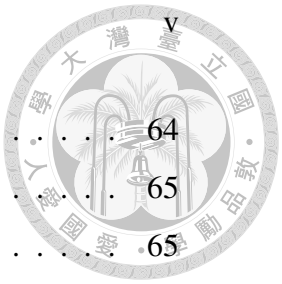
doi:10.6342/NTU202202005

Contents

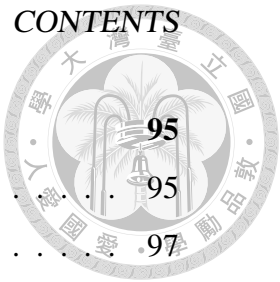
doi:10.6342/NTU202202005



doi:10.6342/NTU202202005



doi:10.6342/NTU202202005



doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

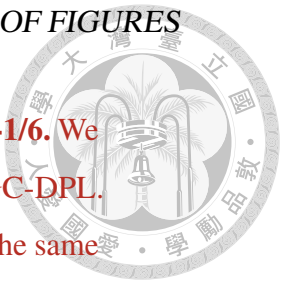
doi:10.6342/NTU202202005

List of Figures

doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

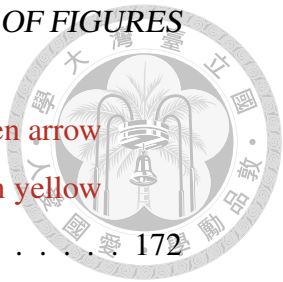


doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

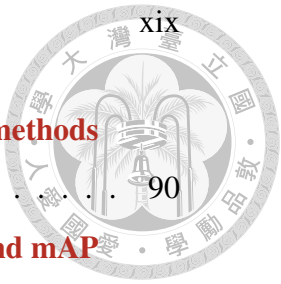


doi:10.6342/NTU202202005

List of Tables

doi:10.6342/NTU202202005

doi:10.6342/NTU202202005



doi:10.6342/NTU202202005

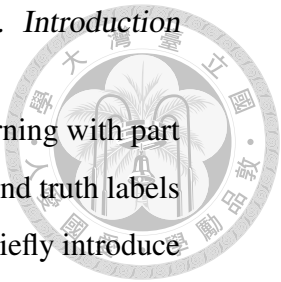
doi:10.6342/NTU202202005

Chapter 1

Introduction

Multi-Target Multi-Camera Tracking (MTMCT) is the core technique in the intelligent surveillance system. This task aims at tracking every identity that walks across different cameras under the environment. With MTMCT, we can analyze the walking patterns of pedestrians to improve the route planning of hypermarket or amusement park. Also, we can cut the videos from multiple cameras to generate the video summarization of the targeted person with MTMCT. Since building an MTMCT system is a complicated problem, researchers split it into a pipeline structure that contains three computer vision tasks, Pedestrian Detection, Single-Camera Tracking (SCT), and Multi-Camera Tracking (MCT). Under each camera, Pedestrian Detection needs to first regress the bounding boxes of each person in each frame. Then, with SCT, we can link the bounding boxes of the same person to construct a continuous sequence, which called a trajactory. Last, MCT aims to associate the trajactories of the same identities across different cameras. The association of MCT can be based on appearance or temporal cues. However, it is still a challenging problem because of the diverse conditions between all cameras. To simplify the problems, researchers formulate the task called **Person Re-identification** (re-ID) [9], which aims to match two images captured under different cameras based on only the "appearance cues". In this dissertation, we focus on the person re-identification under multiple real-world scenarios, such as

doi:10.6342/NTU202202005



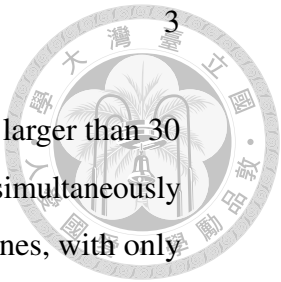
supervised learning based on video sequences, semi-supervised learning with part of the identities labeled, or even unsupervised learning with no ground truth labels of the target environment, which is the most difficult task. We will briefly introduce re-ID and our contributions in Sec. 1.1.

After learning powerful deep-learning models on high-performance GPU servers, how to deploy them on edge devices such as the mobile phones, surveillance cameras or embedded system on autonomous cars is a critical issue. Typically, the low-power computation units can not support the inference of complicated models in real-time. Thus, network pruning is one of the methods to reduce the computation of neural network [10]. With network pruning, we can eliminate the unimportant weights or filters inside the network and thus directly reduce the number of parameters, the number of operations and the inference latency of it. In this dissertation, we tackle the filter pruning on convolutional neural networks (CNN) for two aspects, layer-wise filter pruning and globally determined filer pruning, which will be introduced in Sec. 1.2

On the other hand, recently, the issue that learning the models with sensitive and private data, such as captured pedestrians or human face images, is being brought to public attention. To avoid privacy leakage, in a real-world scenario, we should not transmit the data captured by local devices to a global central server for learning a model. Instead, we should learn the models on local clients but also get a comparable performance. Federated learning (FL) is an emerging technique that can achieve the goal mentioned above [11]. In this dissertation, we formulate a novel FL framework to simultaneously improve the generic representation of global model and the local personalized performance for better user experience. We will elaborate the detailed concept in Sec. 1.3.

After solving multiple sub-tasks in MTMCT systems, we combine the proposed techniques of video person re-ID and global filter pruning to construct a real-time online demo system. This prototype system consists of three parts that are parallelly executed, video streaming, person detection combined with SCT, and MCT with

doi:10.6342/NTU202202005



light-weight person re-ID. Each part can perform in real-time with larger than 30 frame-per-second (FPS), and particularly, our MTMCT system can simultaneously execute on three camera streams, which means three unique pipelines, with only one RTX 2080 GPU released in 2018. It shows that in the future, with our system design, it is possible to deploy multiple cameras under an environment and each pipeline is executed separately with a low-power embedded GPU. We will briefly illustrate our system in Sec. 1.4. The following sub-sections are detailed introductions of my research topics:

1.1 Person Re-identification

Person re-identification (re-ID) tackles the problem of matching images of the same person in a camera network, which has drawn much attention in recent years because of its wide applications in the intelligent surveillance system. As shown in Fig 1.1(a), re-ID is formulated as a retrieval problem. Given a query image, we aims to rank the candidate images in gallery set captured by "different" cameras. There are two main challenges in re-ID. The first one is the generalization of open-set recognition problem. In typical machine learning tasks, we have training and testing set. Although images in testing set are unseen, those are still in the same classes as the training set. However, in re-ID, the identities in the two sets are totally different and non-overlapped. In contrast, we have to learn the general feature representation from the training set and apply it on testing set of unseen identities. The second challenge is the large intra-class variation. The same person under different cameras are with intensive variation of appearance, which can be caused by lighting, pose, viewpoint angles and occlusion, etc. Since the success of deep-learning, many existing works obtained huge improvements with the aid of CNN to learn the features with large-scale dataset. The most popular benchmarks are image-based supervised re-ID, such as Market-1501 [12] and DukeMTMC-reID [1] dataset. With different kinds of CNN designs and loss functions, lots of

doi:10.6342/NTU202202005

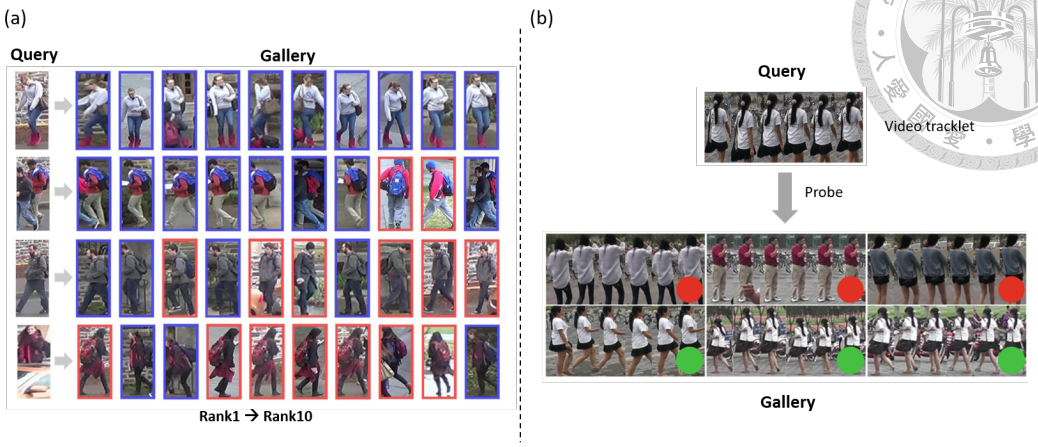
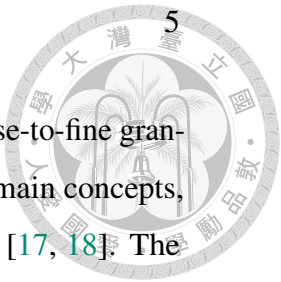


Figure 1.1: **(a) Person re-identification.** This is an image matching problem between query and gallery. The correct matches are in blue rectangles, where the wrong matches are in red. **(b) Video-based person re-ID**. The matching of query and galleries are all based on video tracklets. The pictures are modified from [1, 2].

previous works [13, 14, 15, 16] achieve promising results. TransReID [16] even reached 95% accuracy on Market-1501 dataset. However, there are still some limitations and gaps when we want to deploy those high-performance re-ID models into a real-world scenario. There are three practical aspects we want to tackle, video-based re-ID, semi-supervised re-ID and cross-domain unsupervised re-ID:

Video-based Person Re-ID Practically in a MTMC system, when we want to perform the cross-camera matching (re-ID), the input data is not only one image for an identity. Instead, we can obtain a trajectory that contains multiple continuous images. Therefore, we focus on the task called video-based person re-ID that given the query tracklet (part of the whole trajactory), we want to match and rank the gallery tracklets captured under different cameras, as shown in Fig 1.1(b). In this dissertation, we proposed a state-of-the-art self-attention non-local model that can simultaneously attend on important features across spatial and temporal dimensions given the input image sequence. Furthermore, in extension, to efficiently deploy the computation-demand self-attention module, we proposed four kinds of reduction al-

doi:10.6342/NTU202202005



gorithms, spatial stripe, temporal hierarchy, axial attention, and coarse-to-fine granularity. The detailed descriptions are illustrated in Chapter 2. The main concepts, methods and experiments are published in our preliminary works [17, 18]. The source codes of [17] are at https://github.com/jackie840129/STE-NVAN, and the codes of [18] are at https://github.com/jackie840129/CF-AAN.

Semi-supervised Person Re-ID Regardless of video-based or image-based settings, supervised learning needs lots of annotated training data, and especially, the cross-camera pairs are difficult for annotation. In the real-world deployment, one might not be able to collect such a large amount of labeled data in a scene of interest for training purpose. What might be seen is that only a small part of identities (and their data) during a time period are collected and labeled. Semi-supervised learning tries to solve the problem that among the whole training data, most of them are unlabeled and some of them are labeled. Specifically, in re-ID, the labeled data are some of the identities across different cameras and the remaining unlabeled data are from a separate set of identities. In other words, the identities of labeled and unlabeled training set are non-overlapped. In this dissertation, we proposed a semantics-guided clustering method with deep progressive learning, which can progressively assign pseudo-labels to the unlabelled ones. With the truly labeled and pseudoly labeled training data, we can obtain promising results. The detailed descriptions are illustrated in Chapter 3. The main concepts, methods and experiments are published in our preliminary works [19].

Cross-domain Unsupervised Person Re-ID As mentioned above, annotations for large-scale re-ID data is time-consuming and impractical. Instead of semi-supervised manner that part of the data are labeled, how to perform re-ID in an unsupervised way would be a challenging issue to solve. We address the popular problem called cross-domain unsupervised re-ID, which aims at learning re-ID on target domain which is totally unlabeled with the aid of some labeled data on a source domain. The discrepancy between two domains may lead to unpromising

doi:10.6342/NTU202202005

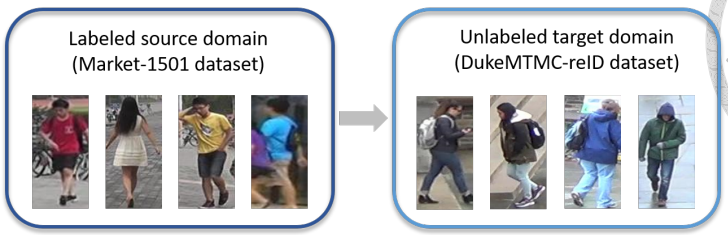


Figure 1.2: Domain difference in cross-domain unsupervised re-ID.

results. As shown in Fig. 1.2, two datasets may have some inherent bias based on the weather, dressing and the ethnicity. In this dissertation, we proposed a dual learning scheme of data clustering with the aid of two rectification methods, which utilize the feature distribution and the camera information to resolve the original inferior results of clustering. The details are illustrated in Chapter 4. The main concepts, methods and experiments are published in our preliminary works [20].

1.2 Filter Pruning for CNN Models

Network pruning is a common solution for optimizing the well-trained models. In the past, [10] remove unimportant "weights" in CNN model, which can reduce the number of parameters. However, in convolutional network, weight pruning can not reduce the number of floating points operations (FLOPs) owing to the unchanged number of filters. Thus, filter pruning, also known as channel pruning is more popular because we can structurally remove the whole filter at a time. Fig. 1.3 demonstrates the basic idea of filter pruning. For the i-th convolutional layer, originally it has four filters, and the output feature map i will has also four channels. If the second filter is unimportant, we can remove it and consequently reduce one channel of the output i, which becomes three. In addition, because the output i is also the input of $(i+1)$-th layer, we can further eliminate the second channel of all the filters in the $(i+1)$-th convolutional layer. Then, the next questions are how to define the importance of filters and where to start pruning the filters. Some works focus on pruning layer-by-layer [21, 22, 23]. If we target on removing 50%

doi:10.6342/NTU202202005

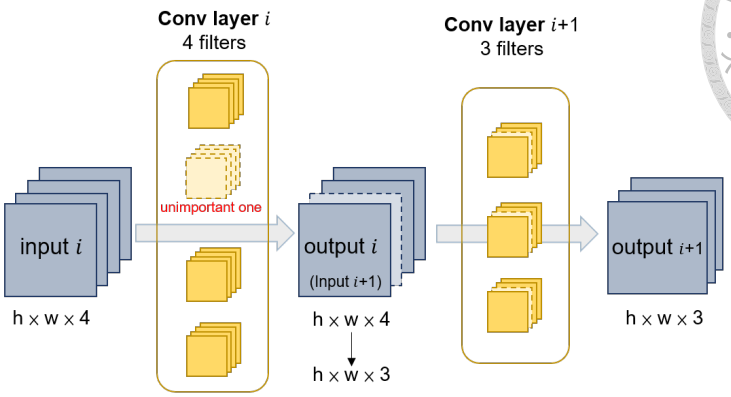


Figure 1.3: **Illustration of Filter Pruning.** If we remove the 2^{nd} filter in layer i, we will reduce the 2^{nd} channel of both output i and the filters in layer $i+1$.

parameters, we can prune 50% filters from the first layer to the last layer separately. On the other hand, recently, most works focus on pruning globally [24, 25, 26], which means we determine the redundant filters based on the whole network and remove the corresponding number of filters. In this dissertation, we focus on both aspects and proposed novel algorithms to resolve the existing problems:

Layer-wise Filter Pruning In the past, L1-norm [8] is the common way to determine the importance of a filter. However, the range of weight values between layers are different. Most works will have to estimate importance of filters and prune them layer-by-layer, which will incur the other question, which layer should we prune first? In this dissertation, we focus on two problems. The first one is the estimation of importance, where we proposed a modified filter sparsity evaluation metric from [27]. Our method will not be influenced by the different range of weight values and required no other hyper-parameters for tuning a suitable sparsity metric. The second one is the order of layers for pruning. We proposed a metric called layer performance sensitivity, which is calculated by the computation and performance trade-off of that layer. Thus, we can prune the layer with lowest sensitivity first to reduce the performance degradation of the models. The details are illustrated in Chapter 5. The main concepts, methods and experiments are

doi:10.6342/NTU202202005

published in our preliminary works [28, 29].

Global Filter Pruning Because layer-wise filter pruning is still time-consuming when finding first suitable layer for pruning, recently, most works adopt the global filter pruning to estimate the importance at a time. This importance estimation is based on the loss impact of each filter [26]. In detail, given a batch of data, a filter is called "unimportant" if the difference of values calculated by model's loss function is small when removing that one. Therefore, the importance estimation will be no longer based on the value of the weights. However, typically, the purpose we want to apply pruning is that we hope the model to fit some hardware constraints, such as the number of FLOPs and parameters left. The intuition is that if we can remove fewer filters to meet the hardware constraints, the performance will retain higher. In this dissertation, we proposed a constraint-aware importance estimation mechanism that can estimate the importance of a filter both based on the impact of loss and constraints, which also achieve the state-of-the-art performance. The details are illustrated in Chapter 6. The main concepts, methods and experiments are published in our preliminary works [30]. The source codes of [30] are at https://github.com/mediaic/CAIE-Filter-Pruning.

1.3 Federated Learning

With the progress of deep-learning, recently the community starts to focus on the privacy issues when we acquire lots of sensitive training data from the internet. Commonly, we will collect the data from local devices and transmit them to some central server for learning the model. However, the private information may leak to public among the data transmission process. Thus, the task called Federated Learning (FL) occurs, which is formulated by one central server and multiple local clients. It aims at learning an aggregated models from data distributed across local devices [11]. The restriction is that the data can only be trained on local clients, and any privacy-sensitive information cannot be sent to central

doi:10.6342/NTU202202005

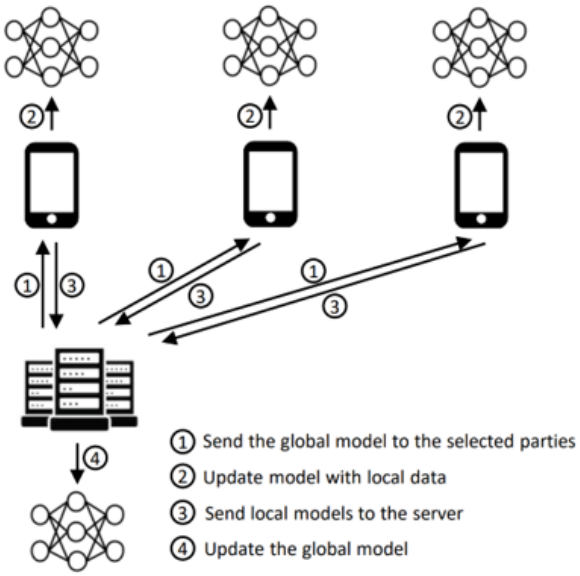
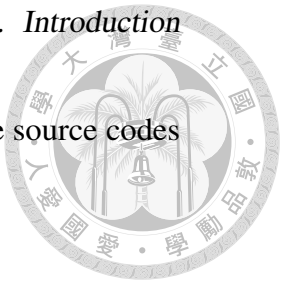


Figure 1.4: **Basic Concept of Federated Learning.** This figure is sourced from [3], where parties means local clients in this dissertation.

server. Typically, they will only send the weight or training gradients of local model to the server. As shown in Fig. 1.4, training is on local side, and the aggregation for the updated model is on the server side. In this dissertation, **we focus on some novel aspects that nearly no papers have addressed**. The first one is the open-set FL problem, which has mentioned in Sec. 1.1. Some well-known computer vision tasks such as face recognition and re-ID are all open-set problems. Because the public face benchmark datasets are sufficient enough for us to formulate it as a FL problem, in this dissertation, we conduct our FL experiments on face recognition benchmark. The other aspect is the generic or personalized representations of the trained model. What is the goal for the learned model in FL setup? What we anticipate is an aggregated generalized model that can obtain better performance on novel dataset or just a personalized model that can make better user experience on the local device. In our dissertation, we tackle the two purposes at the same time and proposed an end-to-end architecture that can learn the two feature representations simultaneously with meticulously designed loss functions. The details are illustrated in Chapter 7. The main concepts, methods

doi:10.6342/NTU202202005



and experiments are published in our preliminary works [31]. The source codes of [31] are at https://github.com/jackie840129/FedFR.

1.4 Real-time Efficient Online MTMCT System

MTMCT system is commonly composed of three sub-tasks, pedestrian detection and single-camera tracking (SCT) in each camera and multi-camera tracking (MCT) across cameras, as illustrated in Fig. 1.5. Pedestrian detection aims at regressing the bounding boxes of people in each frame. In the past, DPM [32] is an effective solution. With the rising of deep-learning, powerful object detection algorithm, such as Mask-RCNN [33] and Yolov4 [34] can be used for detecting humans. For SCT, we need to link the bounding boxes of the same identity into a continuous trajactory, or called track. Recently, FairMOT [35] and GNN-MOT [36] utilize appearance features to help match the current detection boxes and the previously tracked tracks. After SCT, the tracks will be sent to the unified MCT module, which are mainly based on the re-ID features. With robust learned re-ID models, we can compare the feature distance of each track and associate the tracks with smallest distance. To improve the cross-camera matching, we can even combine some timestamp information to help filter those pairs that are impossibly matched along time [37].

An MTMCT system can be constructed offline or online. For only improving the tracking performance, most works adopt the offline setting [38, 39, 40], which means it can globally obtain the complete trajactories from the first frame to the last frame of each video and offline extract, combine and match the features to associate the cross-camera pairs. However, in a real-world scenario, the setting above are unrealistic. On one hand, at any timestamp, we can only utilize frames appeared before to match the candidates. Furthermore, in an online system, we need to instantly match the trajactory to the previously appeared identities in other cameras. On the other hand, the challenge to build an efficient MTMCT system

doi:10.6342/NTU202202005

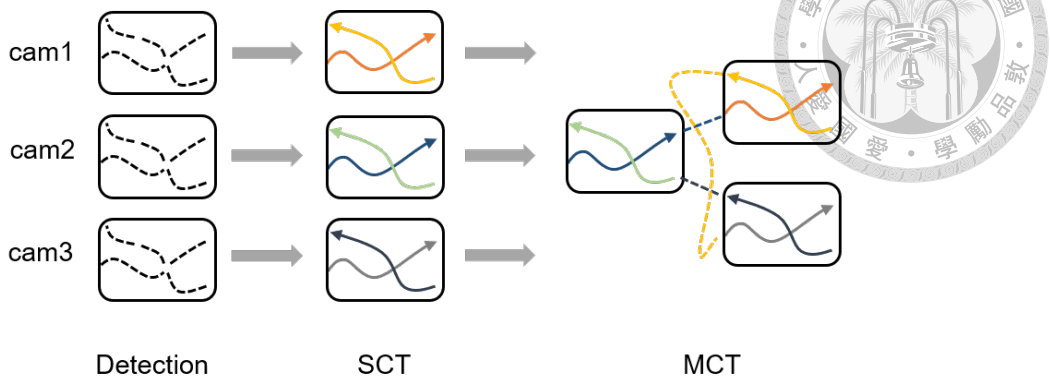


Figure 1.5: **Framework of Multi-Target Multi-Camera Tracking System.**

is to make it operate in real-time, which means running in about 30 FPS. As mentioned above, each sub-task is executed with CNN model, which takes lots of GPU computations. Even we split the system into three parallel pipelines, we still need to achieve the real-time speed of all of them. If we cannot achieve it, lots of input frames would be dropped, resulting in inferior tracking performance and poor user watching experience. In this dissertation. We build an prototype of real-time online MTMCT system. It can at most synchronously operate all the sub-tasks with three camera streams as input. To simulate the real-world scenario that all the sub-tasks operate on its embedded devices with low computation power, we put all tasks together on only one PC-level GPU. It is worth noting that in order to reduce the computation-demanded re-ID module in our system, we integrate our filter pruning technique [30] and video-based efficient re-ID modules [17]. The details of each sub-module is illustrated in Chapter 8. The source codes of our system are at https://github.com/jackie840129/MTMCT.

1.5 Contributions and Publications

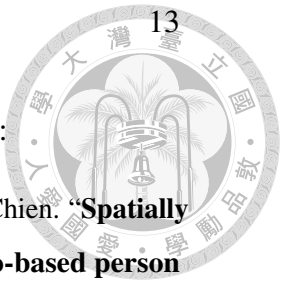
In this dissertation, we consistently focus on the practical situations in each computer vision sub-task. We proposed multiple methods that can not only achieve **effective** performance but also operate with **efficient** computation cost. Moreover,

doi:10.6342/NTU202202005

we address the new potential training framework called federated learning that can optimize the local models without privacy leakage. Last, we combine our proposed methods and construct an "real-time online" MTMCT system. We summarize our dissertation into the contributions as the following:

- In video-based re-ID, we proposed a backbone with non-local attention that can achieve state-of-the-art performance. In the meanwhile, we proposed four kinds of reduction mechanisms that can largely reduce the complex computation in the non-local attention module.
- In semi- and un-supervised re-ID, we adopt the clustering methods to generate pseudo-labels for those unlabeled ones. We proposed rectification mechanisms based on the inherent characteristics, such as the already labeled data or the ID of captured camera to help rectify the noises and errors in the original pseudo-labels.
- In network filter pruning, we proposed a layer-wise pruning algorithm that can iteratively chose the least sensitive layer to prune, which can minimize the performance drop. In addition, in the global filter pruning, we proposed a constraint-aware importance estimation metric that can accurately measure the importance of filters based on the loss impact and the target hardware constraint impact.
- In the field of federated learning (FL), we tackle the open-set FL problem. To best utilize the existing benchmark and dataset, we focus on face recognition scenario and proposed an end-to-end framework that can jointly optimize the generic feature representation and the local personalized representation.
- Last but not the least, we build the prototype of practical online real-time MTMCT system. It combines efficient pedestrian detection, SCT and MCT models. In MCT, we combine our proposed video-based re-ID architecture and our constraint-aware filter pruning to largely reduce the computation cost.

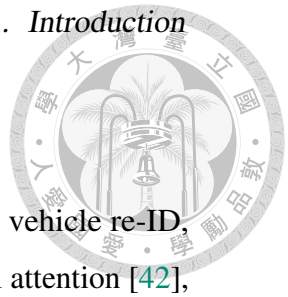
doi:10.6342/NTU202202005



The core of the dissertation relies on the following publications:

- Chih-Ting Liu, Chih-Wei Wu, Yu-Chiang Frank Wang and Shao-Yi Chien. "**Spatially and temporally efficient non-local attention network for video-based person re-identification.**" In *Proceedings of British Machine Vision Conference (BMVC)*, 2019. [17]
- Chih-Ting Liu, Jun-Cheng Chen, Chu-Song Chen, and Shao-Yi Chien. "**Video-based Person Re-identification without Bells and Whistles**". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRw)*, 2021. [18]
- Chih-Ting Liu, Yu-Jhe Li, Shao-Yi Chien, and Yu-Chiang Frank Wang. "**Semantics-guided clustering with deep progressive learning for semi-supervised person re-identification.**" In *arXiv preprint arXiv:2010.01148*, 2020. [19]
- Chih-Ting Liu, Man-Yu Lee, Tsai-Shien Chen, and Shao-Yi Chien. "**Hard samples rectification for unsupervised cross-domain person re-identification.**" In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2021. [20]
- Chih-Ting Liu, Tung-Wei Lin, Yi-Heng Wu, Yu-Sheng Lin, Heng Lee. Yu Tsao, and Shao-Yi Chien. "**Computation-performance optimization of convolutional neural networks with redundant filter removal.**" In *IEEE Transactions on Circuits and Systems I: Regular Papers (TCAS-1)*, 2019. [28]
- Yu-Cheng Wu, Chih-Ting Liu, Bo-Ying Chen, and Shao-Yi Chien. "**Constraint-aware importance estimation for global filter pruning under multiple resource constraints.**" In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRw)*, 2020. [30]
- Chih-Ting Liu, Chien-Yi Wang, Shao-Yi Chien, and Shang-Hong Lai. "**FedFR: Joint optimization federated framework for generic and personalized face recognition.**" In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022. [31]

doi:10.6342/NTU202202005



1.6 Other Publications

During my doctoral research, I also focus on a similar task called vehicle re-ID, which contains joint domain learning [41], orientation-aware spatial attention [42], channel-wise attention [43], adaptive region pooling in backbone [44], learning with inherent space-time priors [45] and learning with synthetic data [39]. Interested readers may refer to these publications for more details.

doi:10.6342/NTU202202005

Chapter 2

Video-based Person Re-identification

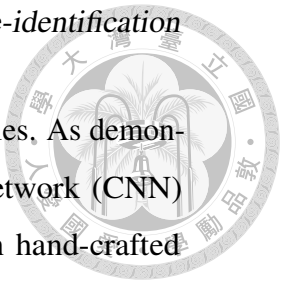
The content from Sec. 2.1 to Sec. 2.3 are based on our previous work [17], which introduce the non-local backbone and its spatial and temporal reduction mechanisms. From Sec. 2.4 to Sec. 2.7, which is based on our another work [18], are our findings of existing problems in benchmark and proposed following solutions. The content contains a pre-process technique and two reduction methods in non-local-based backbone model.

2.1 Introduction

Person re-identification (re-ID) tackles the problem of retrieving pedestrian images/videos across non-overlapping cameras. Previous approaches mostly focus on image-based re-ID, where each pedestrian possesses multiple cross-camera images for retrieval [12, 46, 47, 9, 48, 49]. Recently, video-based re-ID has drawn significant attention in literature since retrieving pedestrian video sequences is more realistic and critical in real-world surveillance applications [50, 51, 2, 52]. With the emergence of large-scale video-based re-ID datasets [2, 52], researchers design deep neural networks to learn robust representation for videos [2, 53, 54, 55, 56, 57].

To perform video-based re-ID, typical methods require learning a mapping function to project the video sequences to a low-dimensional feature space, where

doi:10.6342/NTU202202005



re-ID can then be performed by comparing distances between samples. As demonstrated by numerous works, training the convolutional neural network (CNN) as a mapping function has dominated over classic methods with hand-crafted features [58, 59, 60]. Usually, they obtain re-ID features for a sequence by aggregating image features with average or maximum pooling [53, 2]. However, their approaches fail to handle occlusion or spatial misalignment in video sequences since it treats all images in a sequence with equal importance [56]. In order to distill relevant information for re-ID, some works integrate Recurrent Neural Network to learn the spatial-temporal dependency in an end-to-end training manner [53, 61, 62]. Recently, several works propose attention mechanism to weight the importance of different frames or different spatial locations to aggregate a better representation [55, 56, 57]. While these methods successfully capture both the spatial and temporal characteristics of video sequences, they only explore the aggregation of high-level features for representation, which might not be sufficiently robust for fine-grained classification tasks such as re-ID [63, 64].

In this chapter, we first aim to improve the representation for video sequences by exploiting spatial and temporal characteristics in both low-level and high-level features. Inspired by Wang *et al*. [65], we propose a Non-local Video Attention Network (NVAN) by introducing the non-local attention layer into an image classification CNN model. The non-local attention layer enriches the local image feature with global sequence information by generating attention masks according to features of different frames and different spatial locations. By inserting non-local attention layers at different feature levels, NVAN explores the spatial and temporal diversity of a sequence and alters its feature representation subsequently rather than combining individual image features with a set of weights as in previous works. It is worth noting that we are the first work applying self-attention mechanism into re-ID field. Our NVAN model surpasses all state-of-the-art video-based re-ID methods by a large margin on the challenging MARS [2] dataset, proving that exploiting global information for multi-level features is crucial for learning

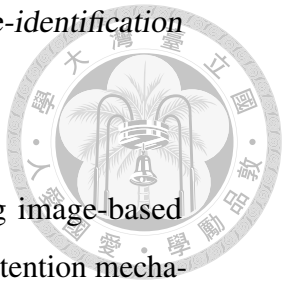
doi:10.6342/NTU202202005

representation for video sequences.

While applying non-local attention layer to multi-level features significantly improves the re-ID performance, it comes at a great cost in terms of computation complexity. In fact, it increases the total floating point operations (FLOPs) by 99.3%, making it difficult to scale up to practical applications. To alleviate such challenge, we take advantage of the space-time redundancy in pedestrian videos and propose a Spatially and Temporally Efficient Non-local Video Attention Network (STE-NVAN). We first reduce the granularity of attention masks in non-local attention layers by exploiting the spatial redundancy exhibited in pedestrian images. On the other hand, we explore the temporal redundancy between video frames to aggregate image-wise information into a representative video feature with a hierarchical structure. By reducing the computation complexity both spatially and temporally, our STE-NVAN cut down 72.7% of FLOPs compared to original NVAN with only 1.1% drop in rank-1 accuracy on MARS dataset. Our proposed STE-NVAN demonstrates a much superior trade-off between performance and complexity compared to existing video-based re-ID methods. The contributions of our work can be summarized as follows:

- We introduce the non-local attention operation into the backbone CNN at multiple feature levels to incorporate both spatial and temporal characteristics of pedestrian videos into the representation.
- We significantly reduce the computation count for our Non-local Video Attention Network by exploring the spatial and temporal redundancy presented in pedestrian videos.
- Extensive experiments validate that our proposed model not only outperforms state-of-the-art methods in re-ID accuracy but also requires less computation count than existing attention methods for video-based re-ID.

doi:10.6342/NTU202202005



2.1.1 Related Work

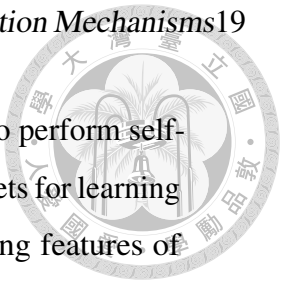
In this subsection, we briefly review the related works regarding image-based person re-ID, video-based person re-ID and the usage of previous attention mechanisms for the re-ID problem.

Image-based Person Re-ID This field has been extensively studied over the years. With the success of CNNs [66, 67, 68, 46, 63], deep features learned from the networks has replaced hand-crafted features [9, 12, 69, 59] for representing pedestrian images. As suggested by Zheng *et al*. [70], these networks can be categorized into discriminative learning and metric learning. Discriminative learning learns deep features for identity classification with the help of the cross-entropy loss [66, 67, 68]. As for metric learning, Hermans *et al*. [46] use the triplet loss to teach the network to push together features of the same person and pull away features of different people. In this work, we utilize both loss functions to train our network for video-based person re-ID.

Video-based Person Re-ID The video-based version is an extension of image-based person re-ID. Zheng *et al*. [2] introduce a large-scale dataset to enable the learning of deep features for video-based re-ID. They first train a CNN to extract image features then aggregate them into a sequence features with average/maximum pooling. Other works [53, 54, 61] adopt Recurrent Neural Networks to summarize image-wise features into a single feature by exploiting temporal relation within a sequence.

Attention in Video-based Re-ID Recently, attention mechanisms are introduced for capturing spatial and temporal characteristics of pedestrian sequences within the deep features. Xu *et al*. [71] introduce the joint attentive spatial and temporal pooling network to extract sequence features by jointly considering the query and gallery pairs with an affinity matrix. Li *et al*. [55] learn attention weights to combine features of different spatial locations and different temporal frames

doi:10.6342/NTU202202005



into a sequence feature. Chen *et al*. [56] utilize techniques in [72] to perform self-attention on each video snippet and co-attention between video snippets for learning sequence features. Fu *et al*. [57] learn sequence features by mining features of discriminative regions and select important frames with a parameter-free attention scheme. While these works achieve promising results by introducing spatial and temporal attention on top of high-level features obtained from image-based CNNs, they overlook the importance of utilizing video characteristics at intermediate feature levels. In contrast, our proposed NVAN is able to refine intermediate features with spatial and temporal information of videos and our efficient STE-NVAN model substantially reduces the computation cost for incorporating video characteristics at lower feature levels.

2.2 Proposed Non-Local Video Attention Network and Two Reduction Mechanisms

Given an image sequence of any pedestrians, we aim to learn a CNN to extract its feature representation that enables video-based person re-ID in the embedding space. The key to learning a representative feature for a sequence is to incorporate video characteristics into the feature itself. To this end, we introduce the non-local attention layer into the CNN to explore the spatial and temporal dependency of a video sequence. We propose a Non-local Video Attention Network (NVAN) in Sec. 2.2.1 to apply such operations at different feature levels. However, we observe incredibly large computation complexity with the introduction of attention mechanisms. Hence, we further propose the Spatially and Temporally Efficient Non-local Video Attention Network (STE-NVAN) in Sec. 2.2.2 to alleviate the computation cost by exploiting spatial and temporal redundancy which exists in pedestrian videos.

doi:10.6342/NTU202202005

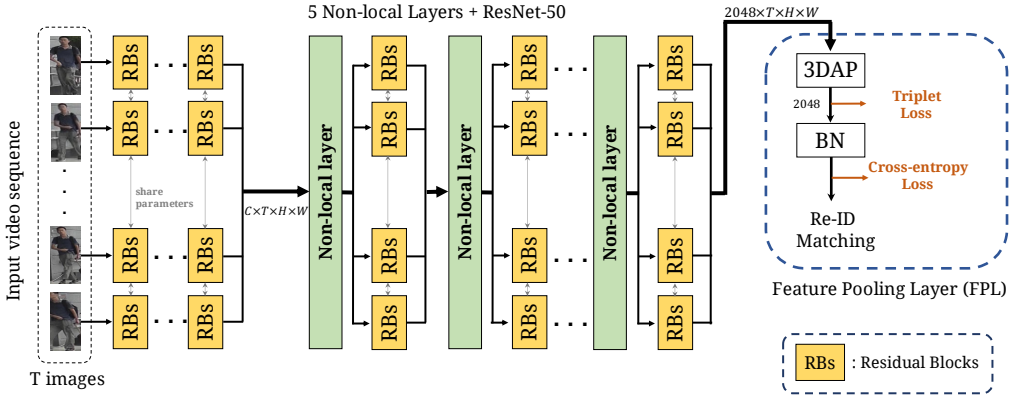


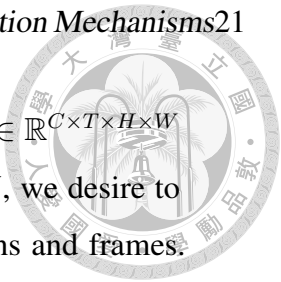
Figure 2.1: **Overview of our NVAN.** In NVAN, given T sampled images as input, the backbone ResNet-50 embedded with 5 Non-local layers generates T features, which incorporates the spatial and temporal information of videos at multi-levels with the help of Non-local Attention Layers. The features are then pooled into one vector in FPL for loss optimization and re-ID matching.

2.2.1 Non-local Video Attention Network

To extract features for an image sequence, we take input as a subset of video frames selected by restricted random sampling (RRS) strategy and forward through a backbone CNN network incorporating non-local attention layers and a feature pooling layer (FPL) to obtain the representation vector for video-based re-ID, as shown in Figure 2.1.

Restricted Random Sampling (RRS) There are several ways to handle the long-range temporal structure. To balance speed and accuracy, we adopt the restricted random sampling strategy [55, 73]. Given an input video $\mathbf{V}$, we divide it into T chunks $\{C_t\}_{t=[1,T]}$ of equal duration. For training, we randomly sample an image I_t in each chunk. As for testing, we use the first image of each chunk. The video is then represented by the ordered set of sampled frames $\{I_t\}_{t=[1,T]}$.

Non-local Attention Layer To embed video characteristics into the features, we introduce the non-local layer proposed by Wang *et al*. [65] into the backbone



CNN, as illustrated in Figure 2.2. Given an input feature tensor $X \in \mathbb{R}^{C \times T \times H \times W}$ obtained from a sequence of T feature maps of size $C \times H \times W$, we desire to exchange information between features across all spatial locations and frames. Let $x_i \in \mathbb{R}^C$ sampled from X, the corresponding output $y_i \in \mathbb{R}^C$ of non-local operation can be formulated as follow:

$$y_i = \frac{1}{\sum_{\forall j} e^{\theta(x_i)^T \phi(x_j)}} \sum_{\forall j} e^{\theta(x_i)^T \phi(x_j)} g(x_j). \tag{2.1}$$

Here, $i, j = [1, THW]$ indexes all locations across a feature map and all frames. We first project x to a lower dimensional embedding space $\mathbb{R}^{C'}$ by using linear transformation functions θ, ϕ, g ($1 \times 1 \times 1$ convolution). Then, the response of each location x_i is computed by the weighted average of all positions x_j by using Embedded Gaussian instantiation. The Equation 2.1 in non-local layer is a self-attention mechanism which is also mentioned in [65]. The overall non-local layer is finally formulated as $Z = W_z Y + X$, where the output of non-local operation is added to the original feature tensor X with a transformation W_z ($1 \times 1 \times 1$ convolution) that maps Y to the original feature space $\mathbb{R}^C$. The intuition behind the non-local operation is that when extracting features at a specific location in a specific time, the network should consider the spatial and temporal dependency within a sequence by attending on the non-local context. In our person re-ID scheme, we embed five non-local layers into our backbone CNN which is a ResNet-50 network [74] to comprehend the semantic relation presented in videos, as shown in Figure 2.1.

Feature Pooling Layer (FPL) After passing the image sequence through the backbone CNN and non-local attention layers, we employ the feature pooling layer to obtain the final feature for re-ID. We apply 3D average pooling (3DAP) along the spatial and temporal dimension to aggregate the output features of each image into a representative vector, followed by a batch normalization (BN) layer [75]. We train the network by jointly optimizing the cross-entropy loss and the soft-margin batch-hard triplet loss [46]. Interestingly, we empirically find that optimizing

doi:10.6342/NTU202202005

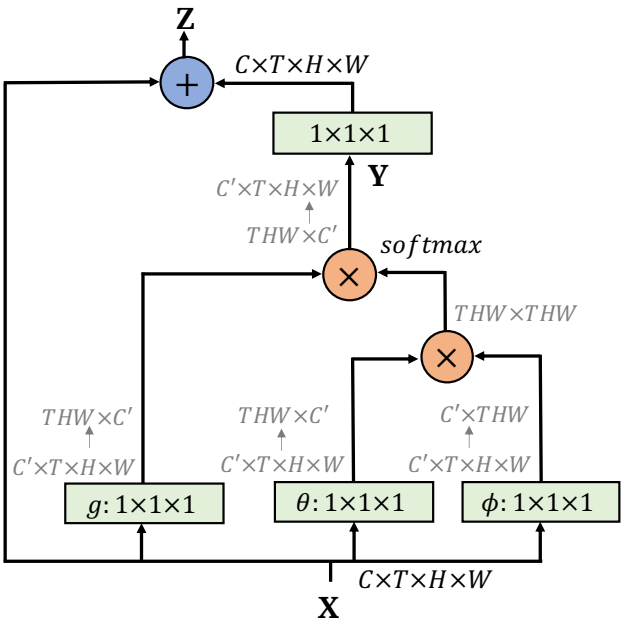


Figure 2.2: **Details of non-local attention Layer.** The non-local attention layer is a self-attention mechanism.

cross-entropy loss on the final feature while optimizing triplet loss on the feature before BN results in the best re-ID performance. A rational explanation is that the embedding space without normalization is more suitable for distance metric learning such as the triplet loss, while the normalized feature space forces the model to classify samples on a more constraint angular space with cross-entropy loss [46, 76, 77]. It is worth noting that this findings is proved by an image re-ID work [78] afterwards.

2.2.2 Spatially and Temporally Efficient Non-local Video Attention Network

While our proposed NVAN is able to capture sophisticated properties of video sequence with the help of non-local operations, we observe a significant increase in the computation complexity, where FLOPs ramps up to two times compared to the original ResNet-50 model, as shown in Table 2.1 that we use $T = 8$ images as

doi:10.6342/NTU202202005

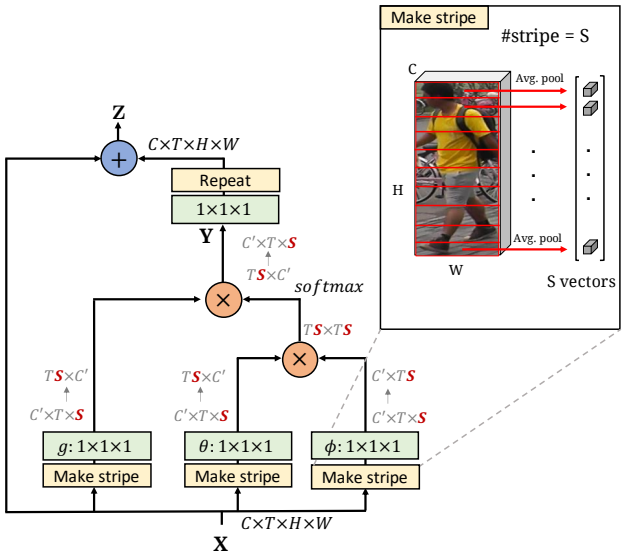


Figure 2.3: **Spatial Reduction Non-local Layer.** We use "Make Stripe" module to average pool the features of each stripe. Before the residual addition, we repeat the tensor of shape $C \times T \times S$ to $C \times T \times H \times W$.

input sequence. For scaling NVAN to practical usage scenarios, we introduce two complexity reduction techniques to cut down the computation count.

Spatial Reduction with Pedestrian Part Characteristics Originally, the introduced non-local operations perform dense affinity calculation between features of all THW positions to obtain a fine attention mask. This results in heavy computation of complexity $O(C'T^2H^2W^2 + CC'THW)$ for each non-local attention layer. Applying the non-local attention layer to lower feature levels incurs larger complexity since low level features are typically of higher H, W. To alleviate such effect, we group the features along the horizontal direction to form a more compact representation of the feature tensor. The intuition is that pixels of the same horizontal stripe tend to share similar characteristics which can be utilized to generate coarse representation of the image. It is worth noting that while similar ideas have been explored in re-ID literature [15, 62, 59], they use this concept to generate finer features for re-ID. In contrast, we exploit this redundancy to obtain coarser

representation. We partition the original feature tensor $X \in \mathbb{R}^{C\times T\times H\times W}$ into S horizontal groups by adding the "Make stripe" module at the input of non-local operations. The resulting tensor $X' \in \mathbb{R}^{C\times T\times S}$ requires only $O(C'T^2S^2+CC'TS)$ to complete the operation, which is irrelevant to the spatial size of feature maps. This dramatically reduces the computation complexity and enables us to deploy non-local operation to lower feature levels with constant computation cost. We name it Spatial Reduction Non-local Layer and illustrate the idea in Figure 2.3.

Temporal Reduction with Hierarchical Structure During our experiments, we observe that features refined by non-local operations are often temporally similar since non-local operation aims to embed global temporal information into the features. Inspired by this observation, we exploit the temporal redundancy between features of different frames and propose a hierarchical structure to reduce the heavy computation of extracting sequence feature. We illustrate this idea in Figure 2.4. After passing a sequence of images through a series of convolutions (Residual blocks) and non-local attention layers, we apply max pooling across features of adjacent frames and reduce the temporal feature dimension by a factor of 2. We perform the same reduction operation after another stacks of Residual blocks until the temporal dimension is reduced to 2, which is then sent to FPL for final feature summarization. This temporal reduction technique cuts down the computation required for extracting sequence feature with Residual blocks and non-local attention layers. By applying both the Spatial Reduction Non-local Layers and the Hierarchical Temporal Reduction structure, we come up with the final Spatially and Temporally Efficient Non-local Video Attention Network (STE-NVAN) for video-based person re-ID.

2.3 Experiments of NVAN and STE-NVAN

We evaluate our approach on two large-scale video-based person re-ID datasets, MARS [2] and DukeMTMC-VideoReID [52]. We conduct ablation studies to

doi:10.6342/NTU202202005

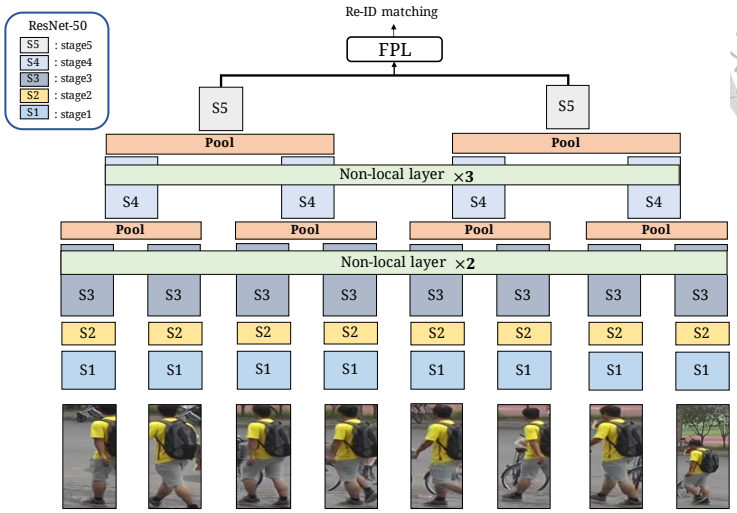


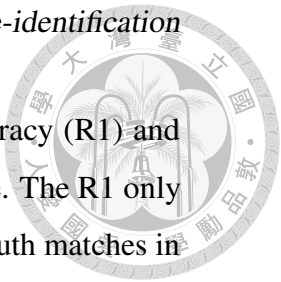
Figure 2.4: **Temporal Reduction with Hierarchical Structure.** We apply max-pooling across adjacent features after the stages with non-local layers to construct our hierarchical structure.

validate the effectiveness of non-local operations and the two proposed reduction methods. We compare our NVAN and STE-NVAN models to existing state-of-the-arts to demonstrate that our proposed models display superior performance while requiring less computation counts.

2.3.1 Experimental Setup

Datasets and Evaluation Protocal MARS [2] is one of the large video-based person re-ID datasets, consisting of 17,503 tracks and 1,261 identities. Each track has 59 frames on average. Deformable Part Model [32] is employed to detect pedestrians and GMCP [79] is used to track pedestrians. To make the dataset even more challenging, they include 3,248 distractor tracks in the dataset. DukeMTMC-VideoReID [52] is another large-scale benchmark recently introduced for video-based person re-ID. It comprises 4,832 tracks and 1,404 identities and 408 distractor identities. Each track contains 168 frames on average. Detection and tracking ground truth are manually labeled. In the following literature, DukeMTMC-VideoReID will be abbreviated as "DukeV" for convenience. In our experiments,

doi:10.6342/NTU202202005



we adopt the standard train/test split and report both rank-1 accuracy (R1) and Mean Average Precision (mAP) to evaluate the re-ID performance. The R1 only measures the accuracy of first rank, but owing to multiple ground truth matches in gallery set, mAP can comprehensively measure the performance of each algorithm, which is illustrated in [12].

Implementation Detail For the RRS strategy described in Sec. 2.2.1, we segment each video into $T = 8$ chunks and sampled 8 images as the input sequence. Each frame is resized to 256×128 and synchronously augmented with random horizontal flip for each track. We adopt the ImageNet pre-trained ResNet-50 [74] as our backbone network, and modified $conv5\_1$ to stride 1 instead of stride 2 to better adapt the re-ID task. For our NVAN, we insert 2 non-local attention layers after $conv3\_3, con3\_4$ and another 3 after $con4\_4, con4\_5, con4\_6$ respectively. As for STE-NVAN, we set $S = 16$ in Spatial Reduction Non-local layer and perform max-pooling right after the second and the fifth non-local attention layer to reduce temporal dimension. We train our network for 200 epoch with both cross-entropy loss and triplet loss [46] and choose Adam optimizer with an initial learning rate of 10^{-4} and decay it by 10 every 50 epochs. Following the suggestion in [46], we sample 8 identities, each with 4 tracks, to form a batch of size $8 \times 4 \times 8 = 256$ images.

2.3.2 Ablation Studies

Effectiveness of Non-local Attention Layer and Two Reduction Methods We first compare our NVAN model with two baseline models to demonstrate the power of non-local operations. The two baseline models (ResNet-50) use the same backbone network as NVAN but without non-local attention layers. The only difference between the two baselines is that one replace the 3DAP in FPL with 3D maximum pooling operation. The first three rows in Table 2.1 illustrate the results. It reveals that non-local operations improve the R1 and mAP significantly by $2.7\%, 3.7\%$ on

doi:10.6342/NTU202202005

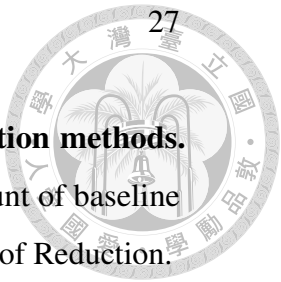


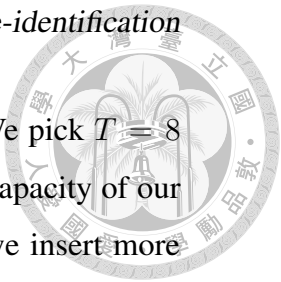
Table 2.1: **Comparisons of different baselines with two reduction methods.** This table shows the performance results and the computation count of baseline models, NVAN and STE-NVAN. The "Reduc." is the abbreviation of Reduction.

Method	Feature Aggregation	MARS		DukeV		# FLOPs
		R1	mAP	R1	mAP	
ResNet-50	FPL	87.3	79.1	95.0	92.7	30.4 G
ResNet-50	max-FPL	86.3	76.6	95.4	92.4	30.4 G
NVAN	FPL	**90.0**	**82.8**	**96.3**	**94.9**	60.0 G
NVAN+Spatial Reduc.	FPL	89.7	82.5	96.3	94.7	30.4 G
NVAN+Temporal Reduc.	FPL	89.2	81.2	95.6	93.7	40.4 G
STE-NVAN	FPL	88.9	81.2	95.2	93.5	**16.5 G**

MARS and $1.3\%, 1.6\%$ on DukeV. The improvement confirms the effectiveness of incorporating spatial and temporal characteristics in the sequence feature of different semantic levels. However, we observe an dramatic 99.3% increase in FLOPs accompanying the introduction of non-local operations. Therefore, we propose two reduction techniques by exploiting spatial and temporal redundancy in pedestrian videos. Table 2.1 shows that our spatial reduction strategy cuts down the FLOPs to approximately the same level as baseline networks while only incurring 0.3% R1/mAP drop on MARS and 0.2% mAP drop on DukeV. As for temporal reduction, we save 32.6% of FLOPs from NVAN and sustain only 1.1% R1 loss on both datasets and 1.7% and 1.2% mAP loss. Finally, by applying both spatial and temporal reduction techniques on NVAN, which is our STE-NVAN, we achieve $\mathbf{72.7}\%$ FLOPs reduction compare to NVAN and requires $\mathbf{45.7}\%$ less FLOPs compare to the baseline that doesn't employ any attention mechanism. It shows that our proposed STE-NVAN not only improves the re-ID performance but also demonstrates a more efficient method of extracting sequence features.

Analysis of NVAN To better understand the property of non-local operations, we conduct analysis on NVAN regarding RRS strategy and number of inserted non-local attention layers. In Table 2.2, we discover that by increasing the number of frames T sampled from a sequence in RRS, re-ID performance increases steadily

doi:10.6342/NTU202202005



as more frames provide richer information about a pedestrian. We pick $T = 8$ for our NVAN and STE-NVAN in consideration of the memory capacity of our machine. On the other hand, we observe performance gain as we insert more non-local attention layers. In Table 2.3, we insert a non-local layer at $conv4\_6$ for "1 layer" and insert 3 non-local layers at $conv3\_4, conv4\_5, conv4\_6$ for "3 layers". We insert 5 non-local layers for NVAN and STE-NVAN since it performs the best.

Analysis of STE-NVAN Next we investigate the parameters for designing STE-NVAN. Starting from NVAN, we apply the spatial reduction techniques to group features into horizontal stripes in non-local attention layer. Table 2.4 shows that while increasing number of stripes S does not introduce excessive additional FLOPs, it improves the re-ID performance subtly. As for analyzing temporal reduction, we increase the pooling operations throughout the network. For comparison, "in 3DAP" in Table 2.5 is the NVAN model that pools all features after the last convolutional layer. By employing additional pooling after the non-local layers located in $stage4$ ("+ stage 4"), we reduce 10.7% of FLOPs from NVAN. And by introducing another additional pooling after non-local layers at $stage3$ ("+ stage 3"), we remove 32.7% of FLOPs from NVAN while only dropping 0.8% and 0.7% of R1 on MARS and DukeV.

2.3.3 Comparison with State-of-the-art Approaches

Table 2.6 reports the comparison of our NVAN and STE-NVAN to state-of-the-art video-based person re-ID approaches. For STA [57], we display their results sampling 8 images per sequence to be fair with our method. On MARS, our NVAN achieves $\mathbf{90.0}\%$ in R1 and $\mathbf{82.8}\%$ in mAP, surpassing all methods by a large margin. Our efficient STE-NVAN also performs better than all methods in R1 and breaks even with STA in mAP despite using less FLOP than NVAN. On the other hand, our NVAN and STE-NVAN still displays competitive results on DukeV, where re-ID on DukeV is easier than MARS since detection are manually

doi:10.6342/NTU202202005

Table 2.2: **Comparison of NVAN network with different # frames of RRS strategy.**

# frames	MARS		DukeV	
	R1	mAP	R1	mAP
$T = 4$	89.0	81.0	95.3	92.7
$T = 6$	89.4	81.6	95.6	93.4
$T = 8$	90.0	82.8	96.3	94.9

Table 2.3: **Comparison of NVAN network with different # non-local layers embedded.**

# non-local layers	MARS		DukeV	
	R1	mAP	R1	mAP
1 layer	89.0	81.8	95.8	93.7
3 layers	89.0	82.4	96.3	94.9
5 layers	90.0	82.8	96.3	94.9

Table 2.4: **Comparison of different # stripes in spatial reduction non-local layer.**

# stripes	MARS	DukeV	#FLOPs
	R1	R1	
$S = 4$	89.6	96.3	30.4G
$S = 8$	89.5	96.1	30.4G
$S = 16$	89.7	96.3	30.4G

Table 2.5: **Comparison of different pooling position combinations in hierarchical structure.**

Pooling positions	MARS	DukeV	#FLOPs
	R1	R1	
in 3DAP	90.0	96.3	60.0G
+stage 4	89.8	96.1	53.6G
+stage 3	89.2	95.6	40.4G

annotated. The superior re-ID performance on two benchmark datasets proves the value of applying non-local operations for extracting a better representation of videos.

To take the computation complexity into consideration, we compare our method with existing methods that also uses attention mechanisms on the performance-

doi:10.6342/NTU202202005

Table 2.6: **Comparison with state-of-the-arts approaches on MARS and DukeV**

Methods	Source	MARS		DukeV	
		R1	mAP	R1	mAP
CNN+Kiss. [2]	ECCV16	65.0	45.6	-	-
SeeForest [54]	CVPR17	70.6	50.7	-	-
LatentParts [80]	CVPR17	70.6	50.7	-	-
TriNet [46]	arXiv17	79.8	67.7	-	-
ETAP-Net(supervised) [81]	CVPR18	80.8	67.4	83.6	78.3
STAN [55]	CVPR18	82.3	65.8	-	-
CSACSE+OF [56]	CVPR18	86.3	76.1	-	-
STA (N=8) [57]	AAAI19	86.2	81.2	96.0	**95.0**
NVAN (ours)	-	**90.0**	**82.8**	**96.3**	94.9
STE-NVAN (ours)	-	88.9	81.2	95.2	93.5

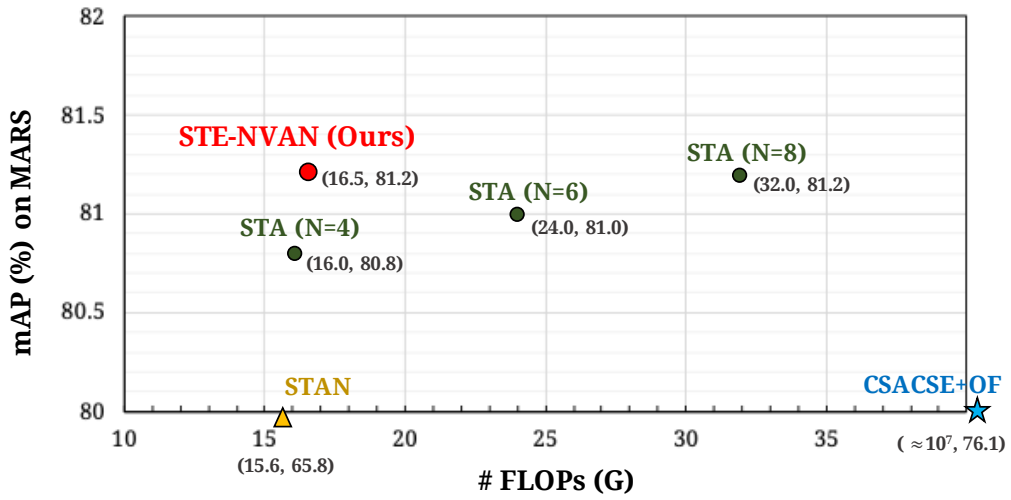


Figure 2.5: **Computation-performance plot of our proposed STE-NVAN and existing methods with attention mechanisms.**

computation plot in Figure 2.5. We visualize mAP on MARS dataset for the performance and # of FLOPs for computation counts. For STA, we report three variants of their with different numbers of sampled frames per sequence to better demonstrate their trade-off. Results show that our proposed STE-NVAN exhibits a much better mAP-FLOPs trade-off compared to current state-of-the-arts. STAN [55] and CSACSE+OF [56] even lands outside of the plot since their mAP and FLOPs are beyond the scale of our plot. The results not only indicates the advantage of our

doi:10.6342/NTU202202005

proposed spatial and temporal reduction techniques but also reveal the importance of considering computation complexity when design feature extractors for video sequences.

doi:10.6342/NTU202202005

2.4 Problems in Existing Benchmarks and Solutions

Recently, many works continuously improve the performance of video-based re-ID on benchmarks [82, 83, 84], the most commonly used methods for tackling video sequences are the 3D convolution layer [85] and non-local operation [65], which can effectively aggregate the features along the spatial and temporal dimensions. However, in contrast to image-based setting that the training and testing images of pedestrians are chosen with the least noise from their belonged tracklets, the video-based re-ID faces more unexpected challenges owing to the imperfect bounding box detection.

MARS [2], the largest video-based re-ID dataset so far, adopted traditional DPM [32] as the pedestrian detector and applied GMMCP tracker [79] with color histogram as image features, which is not robust enough for linking people under a complicated environment with occlusion. As Fig. 2.6 illustrated, the bounding boxes generated by the weak detector cannot well fit the desired identity (the girl with white dress). Recently, Gu *et al*. [82] proposed the appearance preserving module (APM) inserted before the 3D convolution to align the features along the temporal axis based on each anchor (the center) frame of the 3D sliding windows. Although the method achieves the state-of-the-art performance, it still cannot resolve the problems when the center frame contains unexpected noise, such as the fourth frame in Fig 2.6, where the APM will align the third and fifth frames (if the filter size along the temporal axis is 3) according to the appearance of the man with blue T-shirt.

Figure 2.6: **Misaligned video tracklets in MARS dataset.**

doi:10.6342/NTU202202005

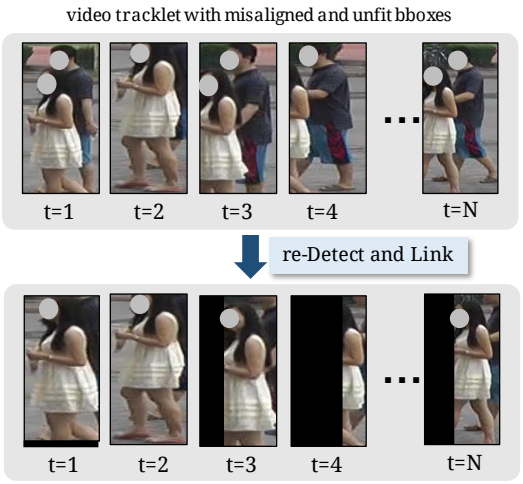


Figure 2.7: **Tracklet processed by our DL module.** The tracklet after DL is less interfered by the man in blue T-shirt.

Since efficient deep-learning algorithms are well-developed for object detection and tracking in the past few years [86, 33, 34, 87], to help the community for the further development of invariant representation without the hassle of the spatial and temporal alignment, we revised the original dataset with our proposed simple but effective **re-Detect and Link** (**DL**) module. Because we cannot obtain the original video stream containing the whole image frame, our DL module serves as a pre-processing technique on the re-ID data. Given the original noisy cropped sequence, we first apply a pretrained efficient object detector [34] to generate much tighter bounding boxes. If there are multiple pedestrian candidates, we will link the pedestrians based on their image features using ID-discriminative embedding (IDE) [70]. Last, according to the aspect ratio and the position of the bounding box, we resize and pad it to the desired image size, as shown in Fig 2.7 and the details are in Sec. 2.5.1.

2.4.1 Re-evaluation of Current Methods

Surprisingly, with only the input data processed by our DL module first, even the C2D baseline method [82], which only averages the features of each image

doi:10.6342/NTU202202005

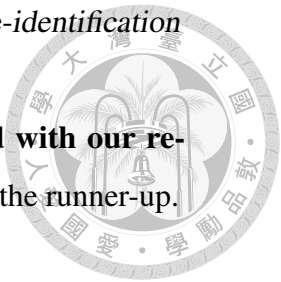


Table 2.7: **Performance of recent state-of-the-arts reproduced with our re-Detect and Link (DL) on MARS** [2]. The score with underline is the runner-up.

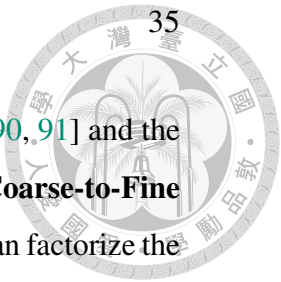
Method	Original Results		**w/ our DL**	
	mAP	rank-1	mAP	rank-1
FT-WFT [89]	82.9	88.6	83.8	90.0
P3D-C [88, 82]	83.1	88.5	85.0	91.0
C2D [82]	83.4	88.9	84.9	91.0
Non-Local [82, 17]	85.0	89.6	**86.2**	**91.4**
TCLNet [83]	<u>85.1</u>	<u>89.8</u>	<u>85.8</u>	90.8
AP3D [82]	**85.1**	**90.1**	85.4	<u>91.0</u>

generated by 2D ResNet-50 [74], or the normal 3D convolution model P3D-C [88] can achieve promising results. As shown in Table 2.7, we conduct more experiments on the original and the processed data using some recent state-of-the-arts reproduced by ourselves. From the table, it can be seen that originally the AP3D with the APM module proposed by Gu *et al*. [82] (the last row) can boost about 2% in mAP compared to its P3D-C counterpart (the second row). However, with the aligned input images generated by our DL module, it only increase 0.4% in mAP. This shows that the state-of-the-art AP3D cannot extract more discriminative features for re-ID given the already aligned data. Furthermore, we can see that the self-attention based Non-local Network [82, 17] combined with our DL module can achieve the new state-of-the-arts, which means the self-attention on the less noisy data can generate more representative re-ID features. Thus, in the next step, we focus on the Non-local Network but developing an efficient baseline model which can perform comparable results.

2.4.2 Efficient Non-local Network

Non-local Network achieves state-of-the-art performance on video-based re-ID, but its high computation cost remains an issue for practical usage. Each feature point along spatial and temporal dimensions needs to compute its self-attention map for all other points. To reduce the computation while retaining the performance of

doi:10.6342/NTU202202005



Non-local Network on re-ID, following the idea of axial-attention [90, 91] and the multi-granularity (coarse-to-fine) structure in [92], we propose the **Coarse-to-Fine Axial-Attention Network** (CF-AAN). With the axial attention, we can factorize the 3D attention operation into three 1D attention ones sequentially along the height-, width- and temporal-axis. To further boost the efficiency, in contrast to [92] that adding the coarse-to-fine module after the whole model backbone, we directly integrate it into our axial-attention. We split the input tensor into multiple scales along the channel dimension, and transform the spatial dimension from coarse to fine scales. To the best of our knowledge, we are the first to adopt axial-attention in video-based re-ID. Our DL+CF-AAN approach not only achieves the state-of-the-art performance on two large-scale datasets [2, 52], but also significantly save the computation as compared with vanilla Non-local Network, which can be regarded as an efficient baseline self-attention method.

2.4.3 Labeling Noise in Existing Dataset

In addition to the application of our DL module that can significantly improve the performance, we also find that there are multiple **labeling errors or noises** in the MARS testing data. As shown in Fig. 2.8(a), the two tracklets are labeled as different identities (ID 142 and 184) but are actually the same person. Or in Fig. 2.8(b), the tracklet with ID 404 in camera 2 also appears in the distractor class (ID 0), which will make the model easily match the two tracklets but counted as an error matching in the evaluation. There are also some ambiguous cases that cannot be distinguished even by human. As in Fig 2.8(c), the ID 318 is the man in blue behind but the bounding boxes also contains the woman in white (ID 322). Thus, we revise the labels in the testing set and the original evaluation protocol. The details will be described in Sec. 2.6.4. We hope that the release of our DL processed test data update on MARS can help the community to validate their methods on a clean testing set and push the further development of improved representation.

doi:10.6342/NTU202202005

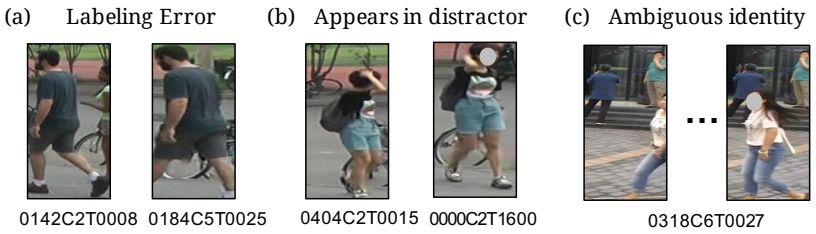


Figure 2.8: **Illustration of the labeling errors and ambiguous cases in MARS [2] testing set.** More samples and details can be found in Sec. 2.6.4

2.4.4 Contributions

Our contributions for tackling the noisy dataset can be highlighted as follows:

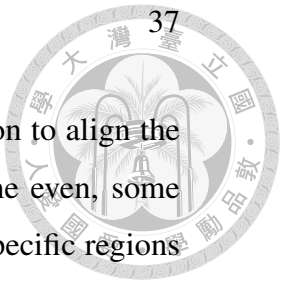
- We propose a re-Detect and Link module that can align the noisy tracklet on the image level, which makes a simple method achieving comparable performance.
- Besides the aligned data, we additionally provide revised identity labels and evaluation protocol in MARS testing set, which helps validate the new methods on a corrected benchmark.
- A baseline Coarse-to-Fine Axial Attention Network (CF-ANN) is proposed, which performs axial-attention from coarse to fine levels, which not only reduces the computation cost but achieves the promising performance.

2.4.5 Related Work

We first briefly review the recent development of video-based re-ID after our proposed work [17], and some works related to self-attention and related to dataset revision.

Video-based Person Re-identification Inspired by the success of 3D CNN on action recognition [93, 94], the work [95] first adopted the 3D convolution to automatically learn the relation from low- to high-level features along spatial and temporal dimensions. In order to resolve the alignment problems, Gu *et al*. [82]

doi:10.6342/NTU202202005

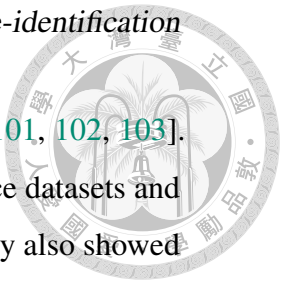


then proposed an APM module inserted before the 3D convolution to align the features among each 3D filter. In contrast to treating each frame even, some works utilized the attention mechanism that can focus on some specific regions representing the identity better [55, 57, 92]. Zhang *et al*. [92] explored the attention mechanism with a global reference, which can effectively learn the attention more on the region with close relation to the global guidance. Besides performing attention on the last layer of CNN features, our previous work [17] perform attention on mid-level features inside the backbone. Compared to those methods, our model is based on the self-attention operation and added with computation efficient structures into the model design.

Self-Attention Since the self-attention based Transformer [72] obtained a great success in nature language processing, recently many works started to tackle the problems in computer vision with self-attention [65, 96, 97, 98, 90, 99, 100, 91]. The plain type of the self-attention is the non-local network [65] without the position encoding and multi-head attention and was proposed to solve the problem of video classification. Because the non-local self-attention is computation demanding, axial-attention [90, 99] were proposed to factorize the operation into multiple 1D self-attentions, which can extremely reduce the cost. Dosovitskiy *et al*. [97] and Carion *et al*. [100] even integrated the whole transformer respectively into the image classification and object detection tasks, and they all obtained comparable performance to the methods with original CNN backbone. Our work focus on adding the efficient axial-attention module with our proposed coarse-to-fine structure into typical CNN to learn spatially and temporally attentive feature representation.

Dataset and Evaluation Protocol Revision In the field of person re-identification, there is no work exploring and revising the original imperfect data or discussing the evaluation protocols, labeling errors, and the ambiguous cases in the testing set. We found that in the field of face detection, there are some works investi-

doi:10.6342/NTU202202005



gating the noise in the labels or the bias in evaluation protocols [101, 102, 103]. Mathisas *et al*. [101] provided improved annotations of existing face datasets and evaluation criteria that resolved the original problems. Besides, they also showed that when properly used, a simple vanilla baseline can reach top performance on face detection. Lin *et al*. [102] and Zhang *et al*. [103] both tried to remove the data with labeling errors before training by utilizing the inherent data distributions. Compared to our work, we adopt pretrained deep learning-based object detector to refine the original test data that are unfit to the target identity. With the aligned data, even a simple baseline method can achieve outstanding performance. Moreover, we manually check the errors with the existing re-ID evaluation protocol and provide some revision of not only the labels but the evaluation protocol.

2.5 Proposed DL+CF-AAN Framework

Fig. 2.9 demonstrates the pipeline of our re-Detect and Link (DL) techniques and the proposed Coarse-to-Fine Axial Attention Network (CF-AAN). Given an original imperfect video tracklet $\mathcal{V}$ with N images, $\mathcal{V} = \{I_1, I_2, .., I_N\}$, we first adopt our DL module to obtain the processed tracklet $\mathcal{V}'$, which is more robust and aligned. The detail of our DL will be described in Sec. 2.5.1. Then, as the typical pipeline of video-based re-ID, we sample T frames from $\mathcal{V}'$ as the input of our CF-AAN. Our network consists of a backbone CNN and multiple Coarse-to-Fine Axial Attention (CF-AA) modules, which are separately inserted between the CNN blocks. The operations in our CF-AAN are described in Sec. 2.5.2. Last, in Sec. 2.5.3, the video features generated by our CF-AAN will be aggregated with the masks created along with DL module and optimized with the common losses for re-ID.

doi:10.6342/NTU202202005

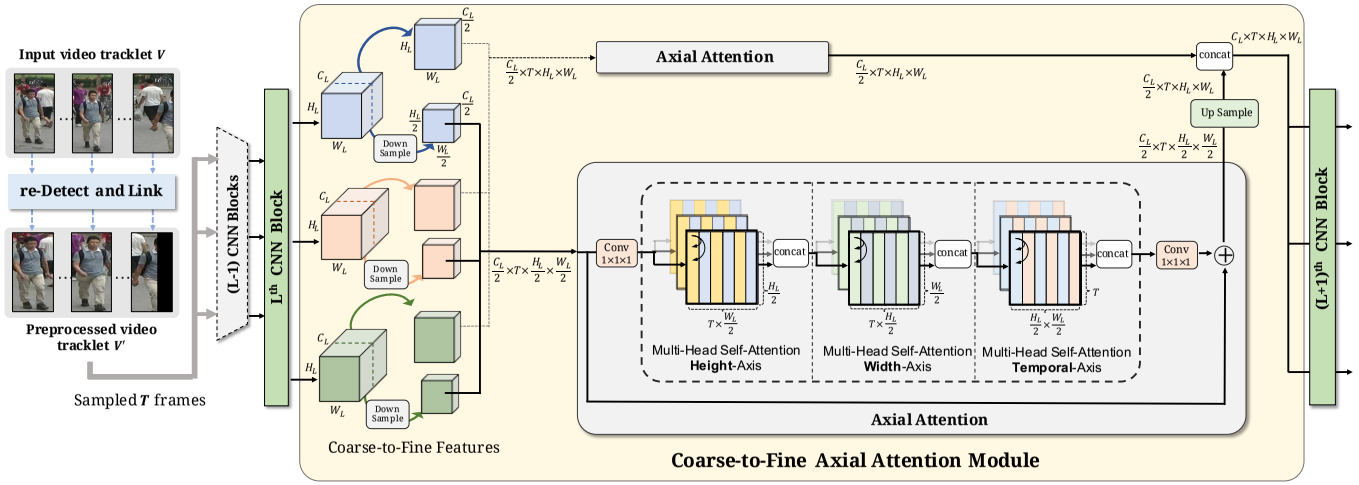


Figure 2.9: **Pipeline of our DL and CF-AAN architecture.** The original tracklet $\mathcal{V}$ is first fed into the DL module and become the processed tracklet $\mathcal{V}'$, which will then be sampled and fed to CF-AAN. We demonstrate one CF-AA module between the L^{th} and $(L+1)^{th}$ CNN block. There are two scales of features and the axial-attention will perform on each of them. The outputs will be up-sampled and concatenated to become the input of the next CNN block.

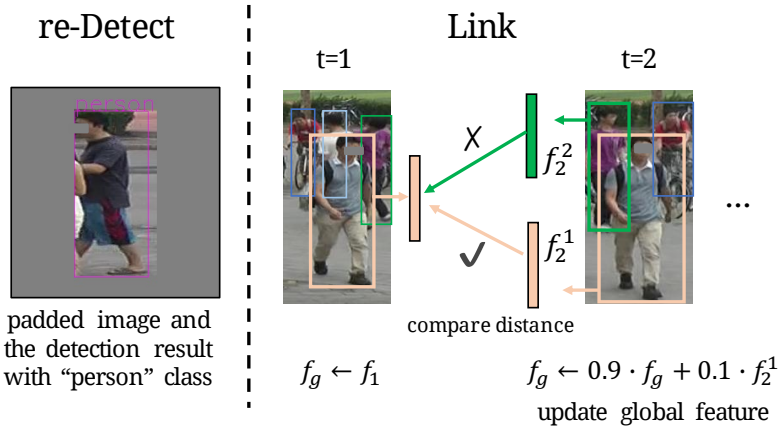


Figure 2.10: **Illustration of the re-Detect and Link module.**

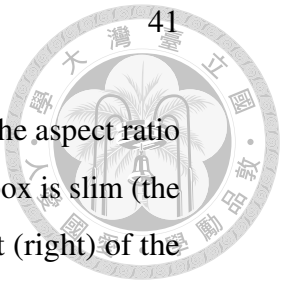
2.5.1 Data Alignment with Re-detect and Link Module

With the noisy video tracklet $\mathcal{V}$ with N images, we sequentially perform our re-Detect and Link (DL) method on each video frame and create a new processed tracklet $\mathcal{V}'$ with N frames, too. As illustrated in Fig. 2.10, first, all images are padded and fed to the object detector [34] to generate candidate bounding boxes with the "person" class. For the first frame, if there are multiple candidates, we will assume that the bounding box with larger area is the desired one. Then, similar to the feature-based real-time object tracking [87], we extract the feature f of the cropped image I_1' by the IDE feature extractor trained on the original dataset [2], and save it as the global feature $f_g = f_1$. Next, for each consecutive frame i, if there are multiple candidates, we will compare each extracted feature f_i^j to the global feature f_g and choose the one with the smallest Euclidean distance, where j is the index of the candidate bounding box in i^{th} frame. After choosing the candidate for the i^{th} frame, the global feature will then be updated by

$$f_g = \alpha f_g + (1 - \alpha) f_i \,, \tag{2.2}$$

where α is set to 0.9 in our case.

Note that in re-ID datasets, we cannot obtain the original full image frame captured by cameras and perform our DL method. Thus, after we apply object detection on the noisy cropped image, we may obtain a new cropped identity with



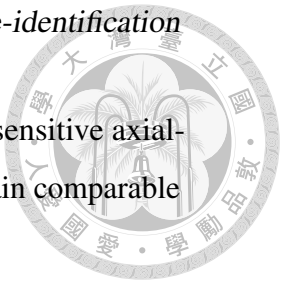
only part of his/her appearance, as shown in Fig 2.7. According to the aspect ratio and the position of the bounding box in the image, if the bounding box is slim (the height is much larger than the width) and its position is on the left (right) of the image, we will shift it to the right (left), resize it based on its original aspect ratio and pad it to the desired image size. Furthermore, we also create a mask M_i of the output image I_i' representing whether each pixel is the padded one or not. This mask will then be applied in the feature aggregation of our CF-AAN, which will be described in Sec. 2.5.3.

Discussion Comparing to other methods proposing an automatically learned feature alignment mechanism inside their backbone model [104, 82], our DL module adopts an additional object detector to help reduce the original noise in the data. It seems that our method requires additional computation cost but and utilizes extra information. However, we want to point out that the goal behind our DL module is to simulate a nowadays real-life scenario with efficient and robust deep learning-based object detection and tracking **before** re-ID. Thus, when it really comes to the re-ID phase, actually there has been no need for this additional cost of DL module on the input tracklet. Furthermore, as shown in the Table 2.7, with the aligned data, the simplest baseline can obtain a promising re-ID result and the original state-of-the-art methods that specifically deal with the problems of misalignment will not retain its competitiveness. We think that with the release of the data processed by our simple alignment method, it can help the community explore more on the attention-based methods or the methods for learning invariant feature representation.

2.5.2 Coarse-to-Fine Axial Attention Network

As shown in Table 2.7, the existing self-attention based Non-local Network can achieve the best result on the aligned data. However, the efficiency is the main drawback. We propose a simple method called Coarse-to-Fine Axial-Attention

doi:10.6342/NTU202202005



Network that contains a coarse-to-fine mechanism and a position-sensitive axial-attention which dramatically reduce the computation burden but retain comparable performance.

Self-Attention: We first introduce the typical 3D self-attention [65] operation as follows. Given an input feature map $x \in \mathbb{R}^{C_{in} \times T \times H \times W}$ with channels C_{in}, temporal length T, height H, and width W, the output y at position $o = (i, j, t)$, $y_o \in \mathbb{R}^{C_{out}}$, is computed by aggregating all the projected input as :

$$y_o = \sum_{p \in \mathcal{N}} \text{softmax}_p(q_o^T k_p) v_p \tag{2.3}$$

where $\mathcal{N}$ is the set of the whole HWT locations, and queries q_o, keys k_o, and values v_o are three different linear projections of the input x_o, $\forall o \in \mathcal{N}$ from dimension C_{in} to intermediate $C_{q,k}$ for query and key projection or C_{out} for value projection. As opposed to convolution which only captures local relations, this mechanism allows us to capture related but non-local context in the whole feature map. Commonly, it will be inserted into multiple locations between the backbone CNN layers, and each complexity is $\mathcal{O}(H^2W^2T^2)$.

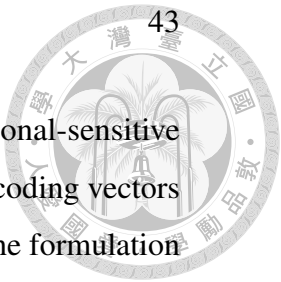
Axial-Attention: To reduce the computation of non-local self-attention, in 2D image classification tasks, the axial-attention has been proposed [90], they factorized the 2D self-attention operation into two 1D axial-attentions. When applied to our video-based re-ID, the 3D self-attention will be consecutively factorized into height-axis, width-axis and the temporal-axis. With this transformation, the complexity can be reduced to $\mathcal{O}(H^2WT + HW^2T + HWT^2)$. The formulation of the axial-attention, with the height-axis as an example, is as follows.

$$y_o = \sum_{p \in \mathcal{N}_{H \times 1 \times 1}} \text{softmax}_p(q_o^T k_p) v_p \tag{2.4}$$

where the location p only lies along the H axis.

Furthermore, based on the concept proposed in the Transformer [72], many works start to encode the positional encoding into the self-attention structure [100,

doi:10.6342/NTU202202005



96, 98]. Thus, the final method we adopt is based on the positional-sensitive axial-attention proposed in [99], where the learnable positional encoding vectors depends on the query vectors, key vectors and the value vectors. The formulation is as follows with the height-axis as an example.

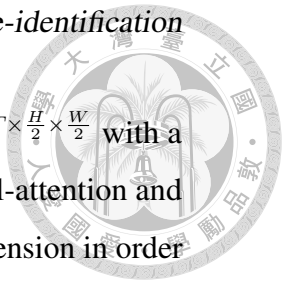
$$y_o = \sum_{p \in \mathcal{N}_{H\times 1\times 1}} \text{softmax}_p(q_o^T k_p + q_o^T r^q_{p-o} + k_p^T r^k_{p-o})(v_p + r^v_{p-o}) \tag{2.5}$$

where the r^q_{p-o}, r^k_{p-o}, and r^v_{p-o} are the learned relative positional embedding. Besides, in practice, as shown in Fig 2.9, we will extend the single-head attention into multi-head attention to generate a mixture of affinities. To retain the complexity, if there are M parallel single-head attentions, in the m^{th} head, each dimension of the q^m,k^m, and v^m will be shrunk to $\frac{C_{q,v}}{M}$ and $\frac{C_{out}}{M}$. The dimension of the learnable positional vectors r^q_{p-o}, r^k_{p-o} and r^v_{p-o} are also shrunk but the vectors are shared across each head. Thus, the final output z_o will be the concatenation of each head, $z_o = concat_m(y_o^m)$, with the same dimension C_{out}. Last, after conducting the axial-attention (AA) along the three dimensions, we will project the output feature from dimension C_{out} back to C_{in} and added with the input tensor x to become a new refined tensor x', which is formulated as follows.

$$x' = x + Conv(\text{AA}^T(\text{AA}^W(\text{AA}^H(x)))) \tag{2.6}$$

Coarse-to-Fine Axial Attention: In addition to multi-head attention that learns different structure of affinities, we propose a Coarse-to-Fine Axial-Attention module (CF-AA) that not only makes the self-attention learn on different scales of the spatial dimension but further reduce the computation. Different from [92], which can only perform multi-scale structure on the last layer of CNN backbone with the smallest resolution, we can apply our structure along with the axial-attention from the mid-level stage to high-level stage inside the backbone. As shown in Fig. 2.9, we split the input tensor x with S scales along the channel dimension and for the s^{th} scale, we downsample the spatial resolution to $H_s \times W_s$, where $H_s = \frac{H}{2^{s-1}}$ and $W_s = \frac{W}{2^{s-1}}$. Thus, if $S = 2$ as an example, the original input tensor x will

doi:10.6342/NTU202202005



be split into $x_1 \in \mathbb{R}^{\frac{C_{in}}{2} \times T \times H \times W}$ with a fine scale and $x_2 \in \mathbb{R}^{\frac{C_{in}}{2} \times T \times \frac{H}{2} \times \frac{W}{2}}$ with a coarse scale. The split tensors are then separately fed into the axial-attention and the outputs are upsampled and concatenated along the channel dimension in order to retain the original tensor size.

2.5.3 Feature Aggregation and Optimization

Our CF-AAN contains a 2D CNN backbone and several CF-AA modules inserted between the CNN blocks. After the last CNN layer, there will be T tensors with size $\mathbb{R}^{C' \times H' \times W'}$. As mentioned in Sec. 2.5.1, because there are some input pixels which are the padded ones without any information, we first downsample the mask M to M' according to the spatial dimension H' and W', and utilize the mask to average-pool on the desired spatial region to generate T vectors with C' dimension. Then, we aggregate the features with the typical average operation followed by a Batch-Normalization (BN) layer [75] to create the final feature representation $f_{\mathcal{V}}$ of the video tracklet. To optimize the network, we follow the two loss combinations in BoT [78], which consists of a batch-hard triplet loss [46] on the features before BN and a cross-entropy loss [70] on the identity classifier (a fully-connected layer) after the feature $f_{\mathcal{V}}$.

2.6 Experimental Results

In this Section, we conduct extensive evaluation and ablation studies of the proposed approach in addition to the analysis and correction of data noise and labeling errors for the evaluation dataset. Same as our previous work [17], we evaluate the proposed method on two large-scale datasets, MARS [2] and DukeMTMC-VideoReID [52], abbreviated as DukeV. We use the rank-1 (R1) in the Cumulative Matching Characteristics (CMC) and the mean Average Precision (mAP) [12] as evaluation metrics.

doi:10.6342/NTU202202005

2.6.1 Implementation Details.

re-Detect and Link. Our object detector is the Yolov4 [34] pretrained on the COCO dataset [105]. The IDE [2] model for linking the candidates is a ResNet-50 [74]. We perform our DL module both on MARS and DukeV dataset. However, because only the MARS dataset is adopted with traditional detector and tracker, where the data in DukeV is manually labeled, the processed data of DukeV is almost the same as before.

CF-AAN. For our CF-AAN, we adopt ImageNet pre-trained ResNet-50 [74] as our backbone. Similar to the structure of Non-local Network [65], we insert 5 CF-AA modules, 2 after $conv3\_3$, $conv3\_4$ and another 3 after $conv4\_4$, $conv4\_5$, and $conv4\_6$ respectively. In our coarse-to-fine structure, we split the feature into four levels ($S = 4$) and in each axial-attention, we set the number of head $M = 2$. Thus, the total number of heads in a coarse-to-fine axial-attention module is equals to 8, which is similar to the original axial-attention network [99]. In the training stage, we sample $T = 6$ images as an input tracklet. Each frame in a tracklet is resized to 256×128 and synchronously augmented with random horizontal flip. As for the optimizer, Adam with weight decay 5×10^{-5} is adopted. We train the model for 220 epochs. The learning rate is initialized to 10^{-4} and multiplied by 0.1 after every 50 epochs. In the testing stage, for each tracklet, we split it into several 6-frame clips, and then the feature representations for each clip are averaged to become the final representation.

2.6.2 Ablation Study

In Table 2.8, we conduct ablation study on our proposed re-Detect and Link (DL) module and our Course-to-Fine Axial-Attention Network (CF-AAN). Besides the re-ID performance, we also calculate the computation cost of inference in terms of GFLOPs. We first analyze the effectiveness of the DL module on our baseline method (the first two rows). Our "Baseline" method, with 24.52 GFLOPs

doi:10.6342/NTU202202005

Table 2.8: **The Ablation Study of our DL and CF-AAN.** We compare the effectiveness of our DL and all the components in CF-AAN with the computation cost (GFLOPs) and performance on MARS. Except the baseline itself, all other computation costs are the increase comparing to the baseline method. C_B: the computation cost of the baseline method.

Method	w/ our DL	Self-attention Module				**# GFLOPs**	**MARS**	
		Self-attention	# of heads	Posi. Encoding	# of scales		mAP	R-1
Baseline	✘	✘	✘	✘	✘	24.520 (C_B)	83.4	87.7
	✔	✘	✘	✘	✘		85.1	89.7
Non-local	✔	3D self-attention	1	✘	1	C_B+17.213	86.2	**91.4**
Axial-based	✔	Axial-attention	1	✘	1	C_B+0.361	86.0	91.1
	✔	Axial-attention	8	✘	1	C_B+0.361	86.2	91.2
	✔	Axial-attention	8	Sinusoidal	1	C_B+0.377	86.0	91.1
	✔	Axial-attention	8	Relative	1	C_B+0.424	86.4	91.2
	✔	Axial-attention	8	Relative	2	C_B+0.245	86.4	91.3
	✔	**Axial-attention**	**8**	**Relative**	**4**	C_B**+0.126**	**86.5**	91.3

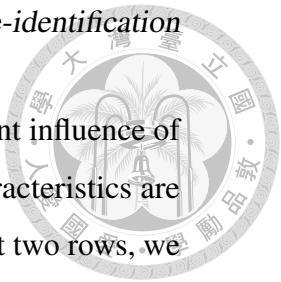
doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

Figure 2.11: **Examples of video tracklets processed by our DL.**

operations, contains the same ResNet-50 backbone, types of losses and training details but without all the axial-attention modules, which is just the average of features in each frame. We can clearly see that with the aligned data processed by DL, there is an obvious improvement of the performance (1.7% in mAP). Thus, the alignment of the input video tracklet is crucial and important for the subsequent feature extraction. We also demonstrate some extra examples in Fig. 2.11. We can see that the problems of misalignment in the left tracklet and the multiple candidates in the right tracklet are resolved after processed with the DL module.

Next, we compare the self-attention based methods. The first one is Non-local Network (the 3^{th} row), which is with single head 3D self-attention but without the positional encoding. Although it can improve about 1.1% in mAP compared to the baseline, the computation also increases ($+\mathbf{17.213}$ GFLOPs), which is extremely large and almost equal to the baseline. After replacing the operation with axial-attention, the computation can reduce to only $+0.361$ GFLOPs, while the performance slightly decrease owing to its factorized self-attentions. With the multi-head structure (the 5^{th} row), it can retain the computation cost but increase the performance. We then apply two types of positional encoding to explore their effectiveness. The first one is the sinusoidal encoding (the 6^{th} row) which is the same as the experiments in [96] and the learnable relative positional embedding



(the 7^{th} row) proposed in [99]. We can see that there is no significant influence of all kinds of positional encoding but the relative and learnable characteristics are the best for re-ID, which can achieve 86.4% in mAP. Last, in the last two rows, we demonstrate the benefits brought by our coarse-to-fine structure. We can see that, because the spatial dimensions decrease in the coarser scale, the total operations also decrease. When the number of scales is 4, the operation can increases only 0.126 GFLOPs compared with the baseline, which is only about $\mathbf{1}\%$ of those in Non-local Network. Furthermore, owing to the coarse-to-fine structure that makes the self-attention learn on different scales, the performance even increases to $\mathbf{86.5}\%$ in mAP on MARS dataset. The CF-AAN with four scales is our final model performing the video-based re-ID.

2.6.3 Comparison with State-of-the-art Approaches

We compare recent state-of-the-art approaches with our methods on MARS and DukeV datasets in Table 2.6. We can see that in the past, the methods that globally perform attention mechanism on the last CNN features are the mainstream for dealing with video tracklet [55, 106, 56, 57]. However, the noise and unaligned appearance between frames make it hard to learn a robust attention score. In another way, TCLNet [83] conduct the attention frame by frame, which is less interfered by the alignment problems. AP3D [82] is the recent work that adopts 3D convolution with a feature alignment module inserted between 3D CNN blocks. We can see that once reducing this unaligned problem, a 3D CNN can achieve the best results (in R-1). The MG-RAFA [92] is also the attention-based method, but they adopt the multi-granularity (multi-scales) structure on the output of the CNN features, where the features will then be fed to their global attention methods. This structure obtains the best results in mAP. Our method consists of a simple but effective pre-processing DL module followed by an extremely efficient CF-AAN. Different from [92], our coarse-to-fine structure is inserted with the axial-attention module between the backbone CNN blocks. We can see that our methods achieve

doi:10.6342/NTU202202005

Table 2.9: **Comparison with state-of-the-arts (%).** The score with underline is the runner-up.

Method	MARS		DukeV	
	mAP	R-1	mAP	R-1
DRSA (CVPR18)[55]	65.8	82.3	-	-
EUG (CVPR18)[107]	67.4	80.8	78.3	83.6
DuATM (CVPR18)[106]	67.7	81.2	-	-
TKP (ICCV19)[108]	73.3	84.0	91.7	94.0
M3D (AAAI19)[95]	74.1	84.4	-	-
Snippet (CVPR18)[56]	76.1	86.3	-	-
STA (AAAI19)[57]	80.8	86.3	94.9	96.2
VRSTC (CVPR19)[84]	82.3	88.5	93.5	95.0
NVAN (BMVC19)[17]	82.8	90.0	94.9	96.3
FT-WFT (AAAI20)[89]	82.9	88.6	-	-
TCLNet (ECCV20)[83]	85.1	89.8	96.2	**96.9**
AP3D (ECCV20)[82]	85.1	90.1	95.6	96.3
MG-RAFA (CVPR20)[92]	85.9	88.8	-	-
DL+CF-AAN (**Ours**)	**86.5**	**91.3**	**96.2**	96.7

promising performance, which outperform AP3D [82] $\mathbf{1.4}\%$ in mAP and $\mathbf{1.2}\%$ in R-1 on the MARS dataset. Although the data in DukeV are manually labeled, our model still can retain comparable performance. Thus, in summary, with almost no extra computation cost compared to the baseline, where conducting the DL module is also effortless in real-life scenario, we are the state-of-the-art in terms of the popular mAP metric for the video-based person re-ID task.

2.6.4 Label Cleaning and New Evaluation Protocols

As described in Sec. 2.4.3, we found some labeling errors or ambiguous cases in the MARS dataset. Thus, we manually check the testing data of the unmatched ones in evaluation and propose a new protocol which additionally address three kinds of new situations: labeling errors, duplication in distractor, and ambiguous identity.

doi:10.6342/NTU202202005

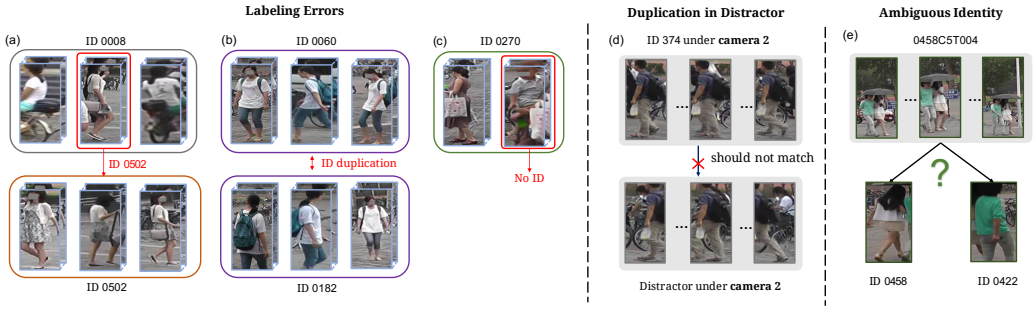


Figure 2.12: **Three kinds of label noises in the MARS testing data.**

Purely labeling errors by annotators: There are also three kinds of labeling errors shown in Figs. 2.12(a)-(c). The first one is that a tracklet may be annotated as another existing identity (2.12(a)). Or, there are completely two groups of tracklets labeled as a different person but in fact with the same identity (2.12(b)). Sometimes the tracklet does not belong to any other identities in the testing set. As Fig. 2.12(c) shows, the identity 270 is the woman but the tracklet marked with red box is the baby she holds. For those three cases, we fix the annotation with the correct or new identity.

Duplication in Distractor Class: In the original evaluation protocol of MARS [2], if a query tracklet matches a gallery tracklet with the same identity but under the same camera, this match will be ignored because re-ID aims at matching pairs across cameras. However, the "distractor class (ID 0)" in MARS consists of not only the false positive bounding boxes created by pedestrian detector but also some duplicated bounding boxes of the tracklets in testing set. As shown in Fig. 2.12(d), the tracklet with ID 374 under camera 2 will easily match the same tracklet in distractor and strangely counted as an incorrect match. Thus, we revise the evaluation protocol that if a tracklet matches the other one under the same camera with its same identity or the distractor class, they will both be ignored.

Ambiguous Identity: There are some ambiguous cases in the dataset. As the tracklet in Fig. 2.12(e), the unfit bounding box contains two persons (ID 485 and

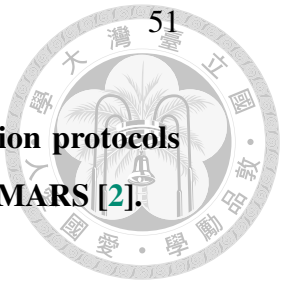


Table 2.10: **Performance evaluated with/without new evaluation protocols (N.E.) and the computation cost of recent methods with DL on MARS [2].**

Method (**w/ our DL**)	w/o N.E. (mAP)	**w/ N.E.** (mAP)	# GFLOPs
C2D [82]	84.9	87.5	**24.520**
P3D-C [88, 82]	85.0	87.5	26.030
AP3D [82]	85.4	88.2	26.369
TCLNet [83]	85.8	88.4	30.150
Non-Local [82, 17]	86.2	88.6	41.733
CF-AAN (**ours**)	**86.5**	**88.9**	24.646

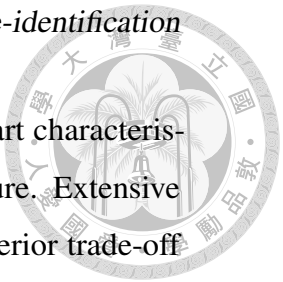
ID 422) from the beginning to the end of the tracklet. With our DL, there is only one person left but the true identity cannot be even distinguished by human. For those cases, we will add an additional ambiguous identity of the tracklet and in the evaluation process, the matches of those identities will all be counted as the correct ones.

Similar to Table 2.7, we reproduce some existing methods not only with data processed by our DL but evaluated under our new protocols, which are shown in Table 2.10. Furthermore, with their released codes, we also demonstrate the computation cost in inference time with fairly 6-frames clip as input data in terms of GFLOPs. We can see that all methods can improve largely by 2.5% in mAP, but our CF-AAN still achieves the best result (88.9% in mAP). When regarding the computation cost, those of our CF-AAN are comparable to the ones of the simplest C2D baseline method and promisingly, also lower than all existing state-of-the-arts.

2.7 Summary

We introduce a Non-local Video Attention Network (NVAN) which incorporates multiple non-local attention layers to extract spatial and temporal video characteristics from low to high feature levels, which enrich the representation of videos in person re-identification. To alleviate the computation cost, we proposed a Spatially and Temporally Efficient Non-local Video Attention Network (STE-NVAN), which

doi:10.6342/NTU202202005



spatially reduce the non-local operation by utilizing pedestrian part characteristics and temporally reduce the operation with hierarchical structure. Extensive experiments are conducted to prove that our STE-NVAN is a superior trade-off between performance and computation. Furthermore, we tackle the problem of existing dataset, where the unaligned input sequence hinder the performance of state-of-the-art methods. We present a simple re-Detect and Link module to further process the datasets, which can significantly refine the data generated with obsolete methods. Then, we proposed Coarse-to-Fine Axial-Attention Network, which significantly improves the original non-local module in terms of computational cost with three 1D position-sensitive axial-attentions and the proposed coarse-to-fine structure while achieving the state-of-the-art performance. With our refined data, we find that several baseline models can achieve comparable results with current state-of-the-arts. In addition, we also disclose the errors not only for the identity labels but also the evaluation protocol for the test data of MARS. With these findings, we hope the release of corrected data can encourage the community for the further development of invariant representation on view, pose, illumination, and other variations without the hassle of the spatial and temporal alignment and dataset noise.

doi:10.6342/NTU202202005

Chapter 3

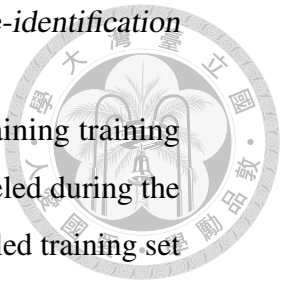
Image-based Semi-supervised Person Re-identification

3.1 Introduction

With the emergence of large-scale datasets [12, 1], in supervised image-based re-ID, methods employing deep convolutional neural networks (CNN) have demonstrated great successes [109, 68, 46, 70, 67, 63, 110]. Yet, in practical scenarios, one might not be able to collect such a large amount of labeled data in a scene of interest for training purposes. Instead, one typically encounters semi-supervised setting in real-world re-ID tasks. More precisely, one can collect a number of fully labeled pedestrian data across camera views during specific time period, while the remaining training data under such views observed at other time periods remain unlabeled. Thus, one cannot easily apply and train existing supervised re-ID methods on semi-supervised data.

To address the aforementioned problem of semi-supervised person re-ID, we can consider two possible settings. Recent works like [111, 112] assumes that each identity has at least one image in the training set. However, in practice, identity labels of labeled and unlabeled ones **do not** overlap (e.g., re-ID of different time periods). Thus, we follow the setting in [113, 114] that only a small part of

doi:10.6342/NTU202202005



the identities (and their data) are seen and available. For the remaining training data, they are from a separate set of identities and are totally unlabeled during the training process. In other words, the identities of labeled and unlabeled training set are non-overlapped.

It is worth noting that, the above semi-supervised person re-ID setting is rarely addressed but practical and also challenging, since the number of identities is unknown in the unlabeled set. With only a small part of labeled identities available in this semi-supervised setting, we need to exploit the unlabeled images to assign pseudo-labels for training purposes. Existing works like [113, 114] simply apply K-means clustering on the unlabeled data, and then assign pseudo-labels to these data according to clustering results. However, they need to assume that the number of cluster K (i.e., identity) is known before training. They directly use the ground truth number of identities to obtain the best results, which might not be sufficiently practical either. Furthermore, assigning pseudo-labels to all unlabeled data as [113] needs to be carefully handled, otherwise undesirable labeling errors would degrade the performance of the re-ID model.

To address semi-supervised person re-ID with labeled and unlabeled training data sharing disjoint identity labels, we propose a *Semantics-Guided Clustering with Deep Progressive Learning (SGC-DPL)* framework. By jointly exploiting labeled and unlabeled training data, our SGC-DPL aims to augment original label information for learning re-ID models. With the guidance of labeled training data, we first advance the affinity propagation (AP) [115] and propose the Semantics-Guided AP (SG-AP), which is a clustering technique without knowing the number of cluster K. Then, we identify and assign pseudo-labels for the unlabeled training data based on the clustering results in a progressive fashion. That is to say, we will gradually enlarge the number of unlabeled data be assigned pseudo-labels for alleviating the errors in the original clustering results. In addition, different from [114, 116], our progressive learning approach does not require any pre-defined selection threshold or the total number of the assigned unlabeled data, which is

doi:10.6342/NTU202202005

also determined by the guidance of the labeled data.

To the best of our knowledge, in the task of person re-ID, we are among the first to leverage the knowledge in labeled set to perform clustering without knowing the number of cluster in advance. Furthermore, we do not require heuristic hyperparameters selection in our AP-based learning model due to our jointly/iteratively exploiting labeled and unlabeled training data.

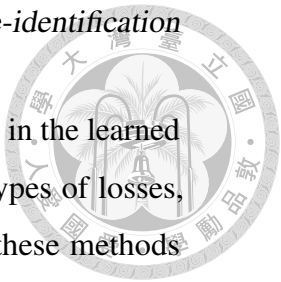
We now highlight the contributions of this work:

- We address the task of semi-supervised person re-ID with labeled/unlabeled training data sharing disjoint identity labels.
- With the guidance of labeled data, our proposed Semantics-Guided Clustering with Deep Progressive Learning (SGC-DPL) framework can jointly exploit the labeled and unlabeled training data in a progressive fashion, while no prior knowledge of the number of identities and the amount of assigned unlabeled data are needed.
- Our model performs favorably against state-of-the-art semi-supervised re-ID approaches, and produces impressive results when comparing to fully-supervised methods.

3.2 Related Work

Supervised person re-ID As mentioned before, with the recent success of deep learning, recent re-ID methods [12, 70, 68, 109, 46, 15, 78] rely on learning CNN models using a large number of labeled training data. Once the learning of complete, re-ID can be simply performed by matching features of query and gallery images. Generally, two types of loss functions are considered for training re-ID CNN networks: identity classification and verification losses. The former is viewed as the cross-entropy loss [68, 70], which encourages the network to correctly recognize the identities of input images. On the other hand, popular verification loss like triplet loss [46, 117] are utilized to encode input images, so

doi:10.6342/NTU202202005



that positive and negative image pairs can be distinguished properly in the learned embedding space. Recent works like [78] jointly use these two types of losses, and very promising results are reported. As noted above, while these methods achieve promising re-ID performance, they require a large amount of labeled data for training purposes, which is often not practical in real-world re-ID applications.

Semi-supervised person re-ID Since collecting and annotating a large amount of training data are often not applicable in real-world applications, how to design and train re-ID models in a semi-supervised setting would be of increasing interest. Some works [118, 119] approach this setting by utilizing labeled training data for synthesizing unlabeled ones via Generative Adversarial Network (GAN) [120]. Once the synthesized images are generated, the multi-pseudo regularized labels can be assigned like [118] or the labels are determined according to the relation of labeled and unlabeled data in the feature space [119]. However, the generated data are not visually robust and the real unlabeled data are also neglected for training the network. A number of works focus on one-example or few-example settings [111, 121, 122, 112], i.e., assuming that only one or few images of each identity are available in the training set, while the remaining ones are of the same identities but unlabeled during training. In [111], a region metric learning method is proposed, which identifies neighbors of the same identity labels and forms a discriminative metric. Wu *et al.* [112] propose a learning method for the unlabeled data which contains the exclusive loss and a progressive pseudo-labels estimation technique. While the above setting requires semi-supervised learning models, one might not be able to collect labeled data for each identity in advance and cannot expect that the identities in unlabeled data would remain the same.

Semi-supervised affinity propagation and Progressive learning Without knowing the number of clusters in advance, affinity propagation (AP) is a suitable clustering solution. Some works [123, 124] propose semi-supervised AP that utilizes the labeled data as additional constraints when clustering on the unlabeled data.

doi:10.6342/NTU202202005

However, both constraints assume a shared label space between those labeled and unlabeled data, which is not suitable for real-world settings. Our proposed semantics-guided AP can learn an adaptive AP mechanism on the labeled set and adapt it to the disjoint unlabeled set.

Progressive learning, which is in the field of self-paced learning (SPL) [125], aims to obtain knowledge from easy to hard samples in a pre-defined scheme, and the self-paced paradigm is theoretically analyzed in [126, 127]. In the semi-supervised re-ID field, many works [116, 112, 114] adopt the progressive learning scheme but they all need to determine a heuristic parameter for the selection threshold or the cardinality of the unlabeled set for training between each iteration. In this paper, we exploit the labeled data and propose a progressive learning method that can automatically generate the suitable threshold for data selection.

Unsupervised person re-ID We note that, a number of unsupervised person re-ID works are presented, which will also be compared in the following sections. BUC [128] and AE [129] try to directly learn discriminative CNN representations on the target unlabeled dataset. For example, Lin *et al*. [128] utilize bottom up clustering to leverage the pseudo-labels, while Ding *et al*. [129] adaptively select image pairs for training re-ID. On the other hand, most works choose to learn the representation of the unlabeled data with the aid of a source dataset in the other domain. MAR [130] propose the soft-multilabel technique for the data in target domain which is based on the relation of the unlabeled data to all the labeled source identities, while SSG [131] utilize a self-similarity grouping to mine the potential similarities for both global and local features. Since annotating at least a small amount of data in the target domain is practical for real-world re-ID applications, we will focus on the semi-supervised setting as described above and will not address the pure or cross-domain unsupervised settings.

doi:10.6342/NTU202202005

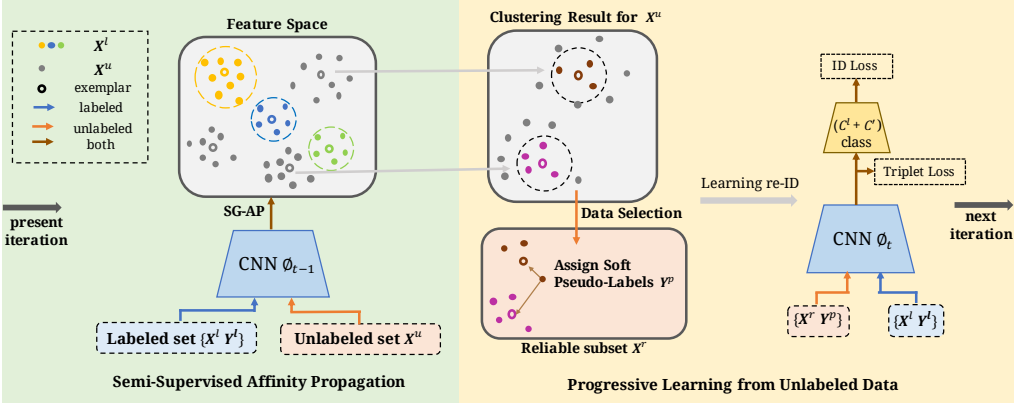


Figure 3.1: **Overview of our proposed SGC-DPL for semi-supervised re-ID.** At each iteration t, we perform semantics-guided affinity propagation (SG-AP) to jointly cluster labeled and unlabeled data and progressively select a subset from unlabeled data for soft pseudo-label assignment. This augments labeled dataset without knowing the exact number of ID labels in advance.

3.3 Semantics-Guided Clustering with Deep Progressive Learning

For the sake of completeness, we first define the problem formulation of semi-supervised re-ID and the notations used in this paper. Assume that we have access to a set of N^l labeled images $X^l = \{x_i^l\}_{i=1}^{N^l}$ and their associated label set $Y^l = \{y_i^l\}_{i=1}^{N^l}$, where $y_i^l \in [1, 2, ..., C^l]$ and C^l denotes the number of identities in the labeled data. In addition, another set of N^u images $X^u = \{x_j^u\}_{j=1}^{N^u}$ without any label information are also available during training. Note that the number of identities C^u in the unlabeled set X^u is *unknown* (which is different from [113, 114]), while their identities are *non-overlapped* with Y^l.

Instead of training the CNN model using $\{X^l,Y^l\}$ only, we additionally leverage the image from X^u to augment the labeled training data. As depicted in Fig. 3.1, we propose *Semantics-Guided Clustering with Deep Progressive Learning (SGC-DPL)* for solving this semi-supervised person re-ID task. This is realized by our semantics-guided affinity propagation (SG-AP) and progressive data selec-

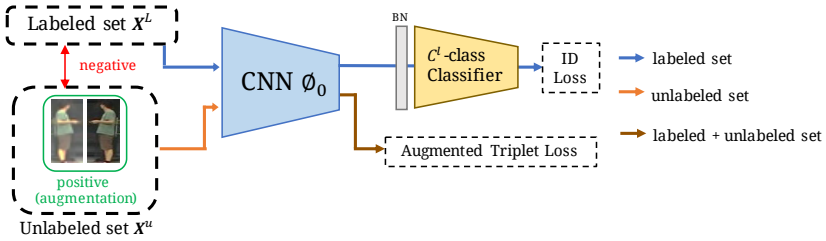


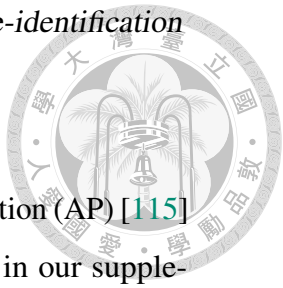
Figure 3.2: **Model initialization for semi-supervised re-ID.** To initialize the re-ID model, the ID/triplet losses are observed from $\{X^l, Y^l\}$, while the augmented triplet loss is additionally observed by exploiting positive pairs from X^u and negative pairs across X^l and X^u.

tion strategies. This would iteratively assign soft pseudo-labels Y^p to a selected subset $X^r \subset X^u$, and augment labeled data for training standard re-ID models.

3.3.1 Model Initialization in Semi-Supervised Re-ID

We now present our model initialization process, which is depicted in Fig. 3.2. Following the model architecture and the training strategy described in [78], we first use a CNN as a feature extractor ϕ, and thus the features of labeled images $\phi(x^l)$ are used to train the batch-hard triplet loss [46]. A BatchNorm [75] and a fully-connected layer are used to construct a C^l-class classifier for optimizing the identity classification loss (ID loss) [68].

In our semi-supervised re-ID task, ID labels are non-overlapped between training data X^l and X^u. Inspired by [132], we further propose an *augmented triplet loss* that utilize the unlabeled set to generate additional positive and negative pairs. To be more specific, given any image in X^u, we first perform data augmentation for an unlabeled image as a novel image with the same label (and thus form a positive pair). On the other hand, we also randomly pick any two images from X^l and X^u (one from each) to form a negative pairs. Therefore, the original triplet loss will be observed by such augmented positive and negative pair data.



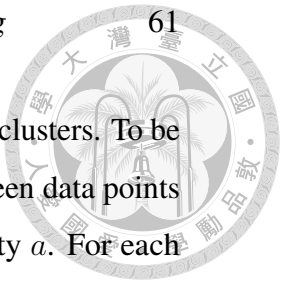
3.3.2 Semi-Supervised Affinity Propagation

Without knowing the number of clusters in advance, Affinity Propagation (AP) [115] is a robust unsupervised clustering algorithm, which is analyzed in our supplementary materials with DBSCAN [133]. To jointly exploit labeled and unlabeled training data for learning re-ID models, we present a novel algorithm of semantics-guided affinity propagation (SG-AP), which is a semi-supervised clustering method. Based on AP, we additionally perform clustering on labeled data to generate semantics (i.e., ID label) guidance for clustering on unlabeled set. That is, we aim at preserving the consistency between the clustering and identity outputs, and augment labeled data from the unlabeled data set for semi-supervised training purpose. Next in Sec. 3.3.3, we will briefly review AP algorithm followed by our proposed semantics-guided affinity propagation described in Sec. 3.3.4.

3.3.3 Brief Review of Affinity Propagation

Given a set of unlabeled data points $X = \{x_1,\ x_2, ..., x_N\}$, AP takes one similarity matrix s between data points as input, where each similarity element $s(i, j)$ shows how likely x_j would serve as an exemplar for x_i. The similarity score can be calculated via $s(i, j) = -\|\phi(x_i) - \phi(x_j)\|_2^2$, where $i \neq j$. This formula indicates the negative euclidean distance between feature points. Note that this distance metric is concurrently optimized by triplet loss in re-ID task, which is also beneficial to the clustering result. *Without* pre-defining the number of objective clusters, AP only needs to define a score $s(i, i)$ for each data point i so that data points with larger $s(i, i)$ are more likely to be chosen as cluster exemplars. These values are called "***preferences***". Such preferences will greatly affect the final clustering result after the learning procedure of AP. However, it is hard to decide the proper preference value for each data point while the value is usually given based on heuristic experiments. In the original AP [115] algorithm, the preference values are "equally" assigned to all the data as $s(i, i) = p\ \ \forall i$, where p is either set to be as the median of the pairwise similarities, which results in a moderate

doi:10.6342/NTU202202005



number of clusters, or their minimum resulting in a small number of clusters. To be more precise of AP learning procedure, two values are passed between data points during internal clustering iteration: responsibility r and availability a. For each step t, responsibility $r_t(i,j)$ is calculated by the similarity matrix s and a_{t-1}, and availability $a_t(i,j)$ is calculated by r_{t-1}. Finally, for a data point x_i, the exemplar of x_i is selected by:

$$c_i \leftarrow \arg\max_{x_j}\{r(i,j) + a(i,j)\}, \tag{3.1}$$

where c_i denotes the exemplar for x_i when convergence.

3.3.4 Semantics-Guided Affinity Propagation

While AP is an effective unsupervised clustering algorithm not requiring the prior knowledge of the number of clusters, it cannot be directly applied to semi-supervised re-ID tasks. This is because that performing clustering on the unlabeled dataset does not necessarily output data clusters corresponding to desirable ID labels. Moreover, assuming all data points possess the same preference with value p hinders the clustering results. To overcome the above challenges, we present *semantics-guided affinity propagation (SG-AP)*, which jointly exploit labeled and unlabeled training data. With the semantics (i.e. ID label) guidance of labeled data, our goal is to cluster and assign psuedo-labels for unlabeled ones to augment the labeled data for training purposes.

To solve the aforementioned problem that preference values of all data are equally assigned, our SG-AP first introduces an adaptive preference function that generates a suitable preference of each data point based on the observed feature distribution, which is produced by calculating the similarities between each data point to the others. The core idea is that, if the distance between a point x_i to other points is larger than the one between x_j to others, the point x_i should has a lower possibility to be a cluster exemplar than x_j does, which results in a lower preference value. To achieve this goal, we first define the Similarity Ranking

doi:10.6342/NTU202202005

coefficient (SR) of each x_i as:

$$SR(x_i) = N \times \frac{\sum_{j=1,j\neq i}^{N} s(i,j)}{\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} s(i,j)}, \tag{3.2}$$

where N is the number of clustered data and $s(i,j)$ indicates the element in the similarity matrix s of the data points. The summation of the similarities among data point x_i to other N points ($\sum_{j=1,j\neq i}^{N} s(i,j)$) will be normalized and multiplied by N to represent the relative ranking value of x_i be chosen as a cluster exemplar among the N data points. Then, we can define the adaptive preference of x_i as:

$$Adaptive\ Preference(x_i) = s(i,i) = SR(x_i) \times p, \tag{3.3}$$

where the $SR(x_i)$ serves as an adaptive ranking weight for the original preference p, resulting in different preference values for each data point based on its similarities to the other data points. Note that both the elements in similarity matrix s and p are negative values; therefore, a data point with high relative ranking to be a cluster exemplar will result in a smaller SR and a larger preference $s(i,i)$.

Although we have adaptive preference, the constant p is still determined by heuristic experiments (median or minimum of similarities), and that might lead to undesirable cluster results on the unlabeled data. To exploit the semantics information (ID labels) in the labeled set, we proposed our SG-AP in the semi-supervised manner. This is realized by enforcing the clustering of labeled data to fit the desirable ID labels. That is, given labeled and unlabeled data $\{X^l, X^u\}$, we first calculate (3.3) with p initially set as the median of similarity matrix observed from the labeled set, where N equals to $N^l + N^u$. Since the number of identities C^l is known, we can search for p^* that makes the number of exemplars of labeled set after clustering best matches C^l. If the number of exemplars is larger than C^l (i.e., over-clustering), smaller p will be considered (and vice versa). This searching process can be sped up with Binary Search on data pair similarities observed from X^l. With $p = p^*$ and $N = N^u$ in (3.3), we perform clustering on X^u and obtain C' exemplars, and such results are guided by the semantics information observed in X^l as described above.

doi:10.6342/NTU202202005

3.3.5 Progressive Learning from Unlabeled Data

Our SG-AP performs clustering on unlabeled data based on the semantics guidance of labeled dataset. To jointly exploit labeled and unlabeled data for training effective re-ID models, the second stage in our SGC-DPL is to progressively assign soft pseudo-labels for selected unlabeled data with high confidence, so that learning of semi-supervised re-ID models can be further achieved.

3.3.6 Progressive Data Selection Strategy

To better leverage the clustering results after our SG-AP process, we now present a data selection strategy by choosing a reliable subset X^r from the unlabeled set X^u in a progressive fashion, as shown at the right part of Fig. 3.1. For each cluster, if the instances x_i^u of that cluster whose feature-level distance to the exemplar $x_{c_i}^u$ is smaller than a threshold τ, we will select such instances with the corresponding labels into the reliable subset X^r. The threshold τ will be progressively enlarged to bring in more unlabeled data to effectively train the re-ID model. A formal definition for X^r can be formulated as follows:

$$X^r = \{x_i^u \mid \|\phi(x_i^u) - \phi(x_{c_i}^u)\|_2^2 < \tau\} \tag{3.4}$$

It is worth noting that, different from most existing progressive learning strategies which typically utilize pre-defined thresholds for data selection [116, 112, 114], our threshold τ can be observed from the labeled set directly. To be more specific, $\tau = \tau_l + d_t$, where τ_l is determined based on labeled set, which dominates the threshold value and d_t is for enlarging the threshold gradually based on the SGC-DPL iteration. Since τ_l is seen as the expected maximum distance between an exemplar and its positive members, we utilize the distance distribution of data pairs in the labeled set to leverage informative hints within such data selection process. Fig. 3.3 depicts the distributions of feature distances within positive and negative pairs in the labeled set X^l on semi-supervised Market-1501 dataset [12]. From Fig. 3.3, it is obvious that we can pick a threshold which is data-dependent

doi:10.6342/NTU202202005

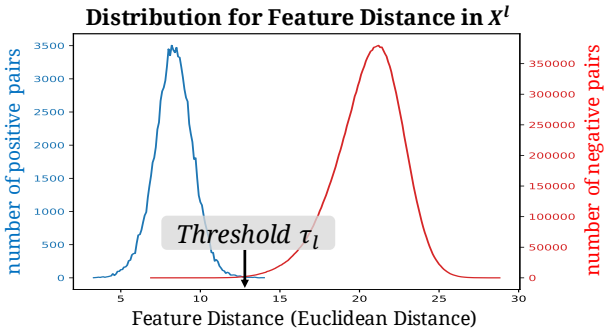


Figure 3.3: **Determining threshold τ_l for progressive data selection.** We illustrate the distributions of distance between pairwise data of X^l on Market-1501 with semi-supervised setting. The blue and red curves are those for positive and negative pairs, respectively. The intersection of the two curves indicates the threshold τ_l which minimizes the data assignment errors for that dataset.

and separates positive and negative pairs with minimum errors. With this observation, the threshold τ_l can be assigned as the distance value on the intersection line, using the labeled training data of interest. Then, between each progressive learning iteration in SGC-DPL, τ_l will be gradually increased till all the instances are selected into the reliable set accordingly.

3.3.7 Soft Pseudo-label Assignment

To train our re-ID model in this semi-supervised setting, the above process allows us to select reliable data X^r based on SG-AP results. In order to assign pseudo-labels Y^p for such data without the prior knowledge of cluster/ID numbers, we choose to assign soft pseudo-labels to alleviate possible clustering or label assignment errors. That is, given a data point x_i^r in X^r and C' cluster exemplars, the soft pseudo-label vector y_i^p is defined as follows:

$$y_i^p = softmax([-d(i,1), -d(i,2), ..., -d(i,C')]) \tag{3.5}$$

where $d(i,j)$ is the feature distance for data x_i^r to the j^{th} exemplar in the unlabeled set. In other words, for x_i^r, the logit of the j^{th} element in y_i^p depends on the distance

doi:10.6342/NTU202202005

between x_i^r and the j^{th} exemplar. The smaller the distance is, the larger the logit is.

After obtaining the reliable data and its soft pseudo-labels $\{X^r, Y^p\}$, such augmented data will be added to the original labeled set X^l for jointly learning for re-ID model. With refined model, the resulting feature extractor will be utilized for SG-AP and progressive data selection in the next iteration.

3.3.8 Learning Objective of Our Model

To train our entire SGC-DPL framework for achieving semi-supervised re-ID, we alternate between the above SG-AP and progressive data selection process for assigning soft pseudo-labels to unlabeled data, which augment the original training set $\{X^l, Y^l\}$ to an updated one $\{X^l, Y^l, X^r, Y^p\}$. We then re-fine our model with the new training data in that iteration by jointly optimizing batch-hard triplet loss and ID loss as [78]. Since new C' identities are added to the original training set, the classifier in our re-ID model will be expanded to train the ID loss with Y^p and Y^l, where y_i^p is a soft label vector used in the cross-entropy loss. In addition, our model is trained using the triplet loss. Since data pairs in $\{X^r\}$ are unlabeled, we determine the identity for selecting positive and negative pairs of x_i^r by our SG-AP clustering results.

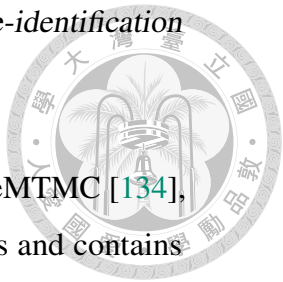
3.4 Experiments

3.4.1 Datasets

We evaluate our method on two benchmarks, Market-1501 [12] and DukeMTMC-reID [1], which are two large-scale datasets with multiple cameras.

Market-1501. The Market-1501 [12] is composed of 32,668 labeled images of 1,501 identities collected from 6 camera views. The dataset is split into two fixed parts: 12,936 images from 751 identities for training and 19,732 images from 750 identities for testing. During testing phase, 3368 query images from 750 identities

doi:10.6342/NTU202202005



are used to retrieve people in the gallery set.

DukeMTMC-reID. The DukeMTMC-reID [1] is a subset of DukeMTMC [134], which is created for re-ID purpose. It is collected from 8 cameras and contains 36,411 labeled images belonging to 1,404 identities. 702 identities with 16,522 images are used for training, and 2,228 images from other 702 identities are used for query images retrieving the rest 17,661 gallery images.

3.4.2 Experimental Settings and Protocols

We employ the standard metrics of the cumulative matching curve (CMC) and the mean Average Precision (mAP). We report the rank-1 accuracy in CMC and the mAP for the testing set in both datasets. We follow the semi-supervised settings in [113, 114], which splits the training set into two parts: one is labeled and the remaining is unlabeled, according to the proportion ratio of person identities. The ratios are set as 1/3, 1/6, and 1/12. For example, for the 1/6 case, only about 125 among 751 identities in the training set of Market-1501 [12] are labeled across cameras and the remaining images in the training set are unlabeled. For fair comparison to some state-of-the-arts, we also adopt the setting that only 50 identities (50 ID) are labeled.

3.4.3 Implementation Details

We employ ResNet-50 [74] as the backbone in our feature extractor ϕ. The 2048-d feature vectors produced by last layer of our feature extractor are used for re-ID and trained with batch-hard triplet loss as well as the PK training strategy suggested by Hermans *et al*. [46]. We sample $P = 16$ different identities and $K = 4$ images for each person at a time to form a batch data of size 64. To improve the supervised training performance, we also follow some of the tricks proposed in [78], which contains the BNNeck, warmup and the REA. We use Stochastic Gradient Descent (SGD) to optimize our model ϕ_t for total 200 epochs with the augmented training set $\{X^l, Y^l, X^r, Y^p\}$ and with the initial learning rate of 0.01 decaying by 10

doi:10.6342/NTU202202005

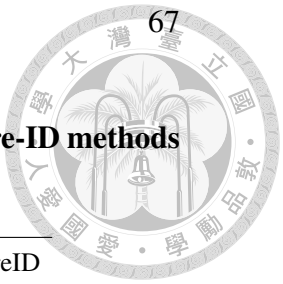


Table 3.1: **Comparisons with unsupervised and semi-supervised re-ID methods on Market-1501 and DukeMTMC-reID(%).**

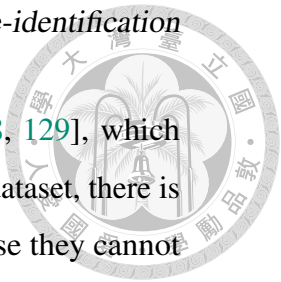
Method	Supervision	Market-1501		DukeMTMC-reID	
		Rank-1	mAP	Rank-1	mAP
BUC [128]	purely unsupervised	66.2	38.3	47.4	27.5
AE [129]		77.5	54.0	63.2	39.0
MAR [130]	cross-domain unsupervised	67.7	40.0	67.1	48.0
SSG [131]		80.0	58.3	73.0	53.4
MMT [135]		87.7	71.2	78.0	65.1
POE [112]	one-example	55.8	26.2	48.8	28.5
ID-disjoint semi-supervised					
UMDL [136]	50 ID labeled	35.6	13.4	19.5	8.3
PUL [114]		50.9	24.8	36.5	21.5
MVC [113]		49.9	24.9	35.7	22.5
MVSPC [114]		62.1	40.9	51.5	31.5
Ours		**83.8**	**65.3**	**74.4**	**56.1**
MVC [113]	1/3 ID labeled	75.2	52.6	57.6	37.8
MVSPC [114]		80.1	62.8	70.8	50.3
Ours		**91.1**	**76.4**	**82.2**	**66.5**
BoT [78]	fully-supervised	94.5	85.9	86.4	76.4

every 50 epochs. The total training iterations of our SGC-DPL framework is set as $t = 8$. In the internal semantics-guided affinity propagation (SG-AP), the searching process of p^* will be terminated if the clustered results on X^l match the number of identities (C^l) or converge to a fixed number of exemplars for 5 iterations. In our progressive data selection, the d_t is initially set as 0 and gradually added with a step size 1 to bring in more unlabeled data.

3.4.4 Comparison with Existing Methods

We first compare our methods with existing two unsupervised settings, one-example setting and the fully-supervised approaches, and report the results on the two

doi:10.6342/NTU202202005



datasets in Table 3.1. For the purely unsupervised methods [128, 129], which directly exploit the target unlabeled data without utilizing a source dataset, there is still a performance gap to the fully-supervised method [78] because they cannot learn the cross-camera image variation in the dataset. AE [129] achieve a great improvement because they additionally utilize a style transfer network provided by [137], which implies that generating various positive pairs in the domain of interest would help. For the cross-domain unsupervised methods [130, 131, 135], they can initialize and update the model with the aid of all labeled data in source dataset; therefore, the performance can be more satisfactory when comparing to the methods above. We note that, while the one-example setting POE [112] adopt a common semi-supervised setting and they also utilize the progressive learning to leverage reliable unlabeled data, a significant performance gap between theirs and fully-supervised BoT [78] is also observed. This indicates that this one-example semi-supervised learning approach cannot produce promising performance in such settings.

On the other hand, as the setting considered in [136, 116, 113, 114], our semi-supervised person re-ID utilizes **disjoint** identities in labeled and unlabeled set. For fair and complete comparisons, we only report results with 50 labeled identities and the ratio of labeled identities as 1/3 in Table 3.1. Other different ratios of labeled data (1/6 and 1/12) are reported in Table 3.2. For the setting that only 50 identities are labeled, it is clear that our SGC-DPL performed against the state-of-the-art MVSPC [114] by a large margin such as $\mathbf{24.4}\%$ and $\mathbf{24.6}\%$ in terms of mAP on Market-1501 and DukeMTMC-reID, respectively. The promising results can also be observed in the setting that 1/3 of the identities are labeled. Our superiority over these state-of-the-art approaches demonstrates the effectiveness of the proposed SG-AP and the guided progressive learning against the K-means clustering without guidance and the pre-defined self-paced learning in [114]. We then further analyze the effectiveness of each component in the next section.

doi:10.6342/NTU202202005

Table 3.2: **Ablation studies of the proposed method in terms of R-1 and mAP (%).** Note that Init., Clus., P.L. and Pseu.-labels indicate the uses of techniques discussed in Sec. 3.3.1, Sec. 3.3.2, Sec. 3.3.6 and Sec. 3.3.7. All methods in this table share the same backbone model.

Experimental setting	Components				M-1/12		M-1/6		M-1/3	
	Init.	Clus.	P.L.	Pseu.-labels	R-1	mAP	R-1	mAP	R-1	mAP
$\{X^l, Y^l\}$	✘	✘	✘	✘	56.8	30.2	68.0	43.3	82.6	61.2
$\{X^l, Y^l, X^u\}$	✔	✘	✘	✘	62.8	38.4	74.0	50.6	83.4	63.7
$\{X^l, Y^l, X^u\}$	✔	*AP*	✘	*Hard*	78.4	55.4	83.5	63.8	88.5	71.9
$\{X^l, Y^l, X^u\}$	✔	*AP*	✘	*Soft*	79.0	57.0	85.5	65.9	88.6	72.5
$\{X^l, Y^l, X^u\}$	✔	*AP*	✔	*Soft*	81.5	61.0	85.9	69.7	89.4	73.7
$\{X^l, Y^l, X^u\}$**(Ours)**	✔	***SG-AP***	✔	***Soft***	**87.9**	**71.6**	**89.8**	**74.9**	**91.1**	**76.4**
All training data	Fully-supervised training				R-1 / mAP : 91.3 / 79.1					

3.4.5 Ablation Studies

To assess the effectiveness of each introduced component in our SGC-DPL, we conduct ablation studies and report the results on Market-1501 in Table 3.2. The experiment is composed of three kinds of semi-supervised settings, which depends on the ratio of labeled identities (1/3, 1/6 or 1/12) on Market-1501 dataset (M) and thus denoted as M-1/3, M-1/6, and M-1/12. We also demonstrate the same results on DukeMTMC-reID dataset in Table 3.3.

Ablation studies on Market-1501 We first assess our initialization strategy in Sec. 3.3.1 using $\{X^l, Y^l, X^u\}$. As listed in the first two rows in Table 3.2, the re-ID model with our initialization strategy outperformed the naive model trained on $\{X^l, Y^l\}$ only, especially on the M-1/12. With initialization confirmed, we next consider assigning hard/soft pseudo-labels to all unlabeled data simply based on standard AP, without semantics guidance from the labeled set. The results are shown in the third and fourth rows in Table 3.2, indicating that our soft pseudo-labels can alleviate the errors in AP. From the fifth row of this table, we see that

doi:10.6342/NTU202202005

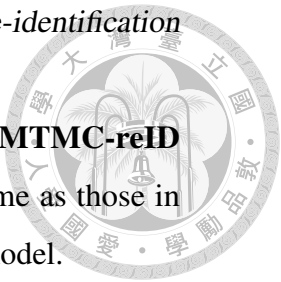


Table 3.3: **Ablation studies of the proposed method on DukeMTMC-reID in terms of R-1 and mAP (%).** Note that the settings are the same as those in Market-1501. All methods in this table share the same backbone model.

Experimental setting	Components				D-1/12		D-1/6		D-1/3	
	Init.	Clus.	P.L.	Pseu.-labels	R-1	mAP	R-1	mAP	R-1	mAP
$\{X^l, Y^l\}$	✘	✘	✘	✘	46.2	27.0	61.2	40.6	71.5	53.1
$\{X^l, Y^l, X^u\}$	✔	✘	✘	✘	50.0	29.1	65.0	43.5	73.7	54.9
$\{X^l, Y^l, X^u\}$	✔	*AP*	✘	*Hard*	63.3	44.0	73.2	55.4	79.0	62.0
$\{X^l, Y^l, X^u\}$	✔	*AP*	✘	*Soft*	65.9	47.1	73.8	55.8	79.4	62.4
$\{X^l, Y^l, X^u\}$	✔	*AP*	✔	*Soft*	68.9	50.9	74.7	57.3	78.7	63.3
$\{X^l, Y^l, X^u\}$**(Ours)**	✔	***SG-AP***	✔	***Soft***	**74.1**	**56.4**	**77.6**	**61.0**	**82.2**	**66.5**
All training data	Fully-supervised training				R-1 / mAP : 85.5 / 71.3					

applying our progressive learning strategy for selecting reliable data to augment the labeled training set would help, while replacing the standard AP by our SG-AP would achieve the best results (i.e., our proposed SGC-DPL). Take M-1/12 for example, where only 60 identities are labeled, when comparing to the baseline approach of using soft pseudo-labels by standard AP only (i.e., the fourth row in Table 3.2), the performance was increased by a large margin from $\mathbf{57.0}$ to $\mathbf{71.6}$ in mAP, which confirms our ability in jointly exploiting labeled and unlabeled data for improved re-ID learning. Finally, we see that with one third of labels observed (i.e., M-1/3), our model was able to produce comparable performances as the fully-supervised model with the same backbone and same training methods produced by ourselves. (i.e., the last row in Table 3.2).

3.4.6 Ablation Studies on DukeMTMC-reID

We also provide the ablation studies on the other large-scale dataset, DukeMTMC-reID [1] (D), with three semi-supervised settings depend on the ratio of labeled identities (1/3, 1/6 and 1/12) and thus denoted as D-1/3, D-1/6, and D-1/12 in Table 3.3. By the way, we did not apply our method on MSMT17 dataset [49]

doi:10.6342/NTU202202005

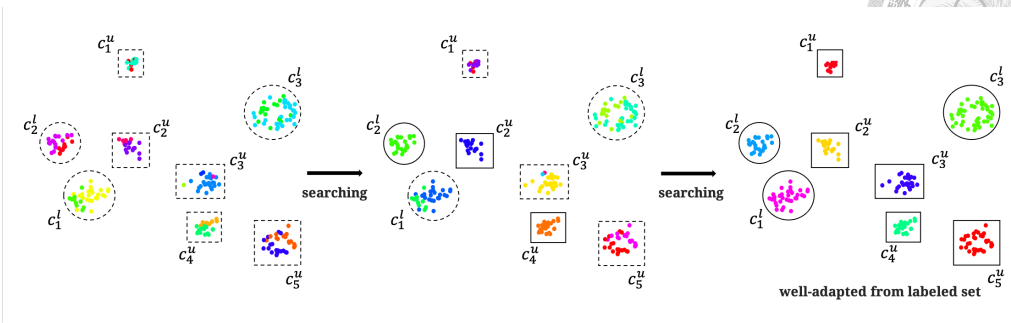


Figure 3.4: **2D t-SNE visualization of internal SG-AP clustering results on sampled X^l and X^u from the M-1/6 dataset.** Data with the same color represent instances of the same *cluster*, while labeled/unlabeled data with the same ground truth identity are bounded by circles/rectangles. Note that instances bounded by dotted circles/rectangles indicate mismatch between clustering and ID labels, while those by solid circles/rectangles denote the match between them.

because it is no longer available now. Same as the comparison on Market-1501 [12] in the main paper, it also shows the effectiveness of our proposed SGC-DPL framework. When comparing to the baseline method using soft pseudo-labels by standard AP only (i.e., the fourth row in Table. 3.3), the performance on D-1/12 was increased by a large margin, too. Furthermore, the performance of our SGC-DPL on D-1/3 can also approach that with the fully-supervised method.

3.4.7 Visualization of SG-AP

To demonstrate the effectiveness of our SG-AP, as shown in Fig. 3.4, we visualize the clustering results across internal searching process in our SG-AP. Data points with the same color represent the same cluster after our SG-AP, while the labeled and unlabeled data are bounded by circles and rectangles, respectively. And, each circle/rectangle indicates a ground truth ID label (e.g., we have c_1^l to c_3^l and c_1^u to c_5^u to denote the ID labels for labeled and unlabeled data, respectively in Fig. 3.4). From the left hand side of this figure, we see that our SG-AP initially divided instances in both labeled and unlabeled data of the same ground truth ID into

doi:10.6342/NTU202202005

Table 3.4: **Preliminary experiments with Affinity Propagation and DBSCAN.** This table shows the clustering results with different clustering algorithms and the re-ID performance after training for one iteration.

Methods	Param. setting	M-1/3				M-1/6			
		#cluster	#ID	R1	mAP	#cluster	#ID	R1	mAP
AP	default	565	501	87.9	70.8	589	626	82.7	62.8
DBSCAN	default	1		–	–	1		–	–
DBSCAN	SSG [131]	381		88.1	70.5	266		81.0	60.7

multiple clusters, which is not desirable. With our SG-AP progresses for searching suitable p described in Sec. 3.3.4, the number of clusters on labeled set would match C^l as shown in the right part of Fig. 3.4, which also guide the unlabeled ones for improved clustering results (e.g., c_1^u to c_5^u in the right most part in Fig. 3.4).

3.4.8 Analysis of Different Clustering Algorithms

Initially, we conduct experiments to evaluate the effectiveness between two widely used clustering solutions that are both no need for deciding the number of clusters in advance, Affinity Propagation (AP) [115] and DBSCAN [133]. We did not conduct the experiments of K-means clustering with different K to validate the robustness as in [114] because we think that even the possible range of the number of identities is also unknown. For AP, we all adopt the default hyperparameters proposed in [115]. For the hyperparameters in DBSCAN, we adopt two settings, one is with the default values and the other one is proposed in SSG [131]. Experiments are conducted on the unlabeled set of M-1/3 and M-1/6, whose feature extractors are only initialized on each $\{X^l, Y^l\}$, respectively. Table 3.4 shows the results. We demonstrate the number of cluster in the first iteration and its ground truth number of identities. In addition, we also show the re-ID performance after the first training iteration ($t = 1$) with hard pseudo-labels and without progressive learning on $\{X^l, Y^l, X^u, Y^p\}$. It can be seen that with the default setting in DBSCAN, we obtain an undesirable clustering results. With the meticulous design in DBSCAN

doi:10.6342/NTU202202005

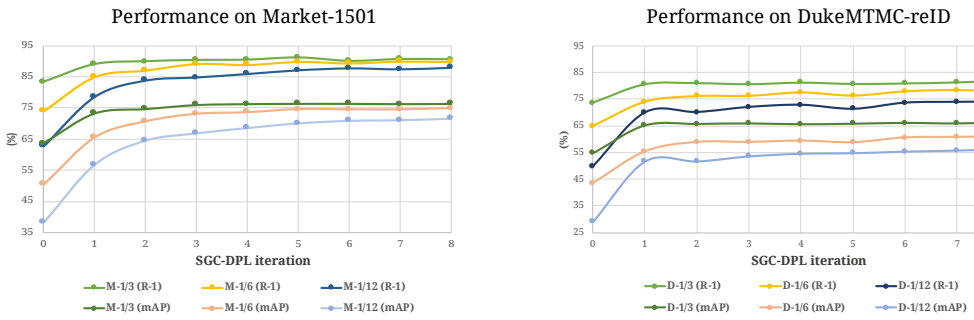


Figure 3.5: **Performance on two datasets along the SGC-DPL iteraions.** We see that the performances generally converged after the 5^{th} iteration. Thus, we had $t = 8$ in our work which would be a reasonable choice.

that follows SSG, the performance can just compete against the AP with default values. Thus, in our SGC-DPL, we choose to adopt AP for clustering the unlabeled data.

3.4.9 Analysis of total #iterations in SGC-DPL

We analyze the hyper-parameter t, which is the total number of iterations in our SGC-DPL framework. Fig. 3.5 shows the performance along the SGC-DPL iterations in terms of rank-1 and mAP on Market-1501 and DukeMTMC-re-ID datasets, respectively. Each dataset consists of three semi-supervised settings considered. The 0^{th} iteration represents the model performance after our initialization method. From these figures, we observe that the performance converged after the 5^{th} iteration in both datasets. Thus, we set $t = 8$ which would be a reasonable choice for the proposed SGC-DPL framework.

3.4.10 Visualization of our Progressive Learning Strategy

For each iteration in our progressive learning strategy, we will create a reliable subset for each cluster with the threshold τ which is automatically generated based on the labeled set. The τ will be increased progressively in each iteration to enlarge

doi:10.6342/NTU202202005

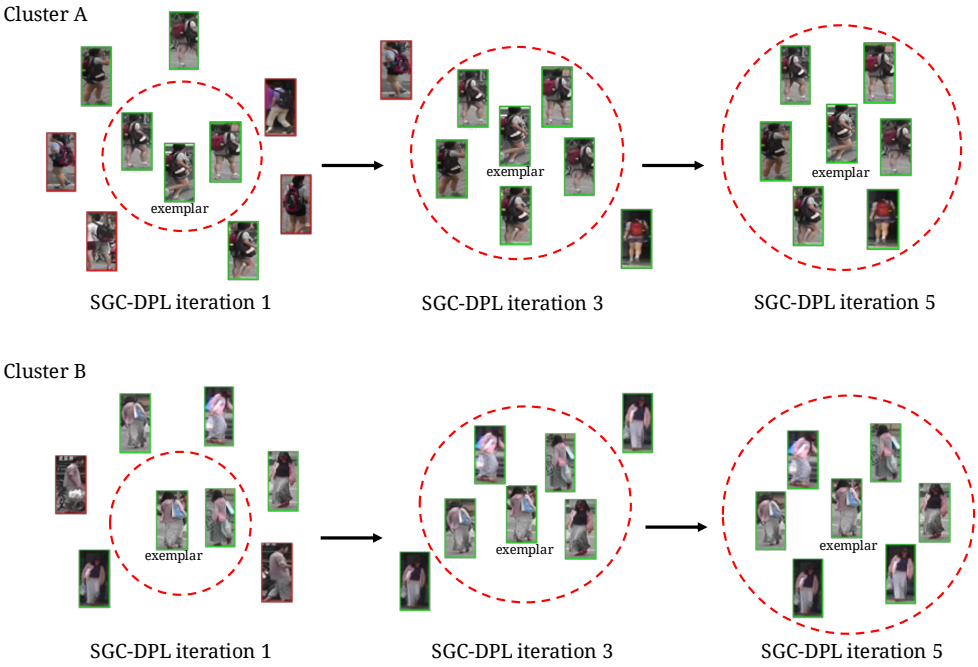


Figure 3.6: **Visualization of our progressive learning strategy on M-1/6.** We illustrate example results of selected two clusters by SGC-DPL. The images in green bounding boxes represent those with the same ID (as that of the cluster exemplar), while images in red bounding boxed are not. The red dotted circle denotes the reliable data subset selected. We see that the ID labels were noisy in the beginning of clustering. Reliable data selected over iterations would update both pseudo-label prediction and clustering, which effectively augment labeled data from unlabeled data for improved learning.

the subset till all the samples are in the subset. Fig. 3.6 shows two visualized cluster examples in our SG-DPL iterations on M-1/6 dataset. Each row represents the cluster members of the same exemplar along the iterations. The images with green border are with the same ground truth identities to the exemplar, and those with red are not. The red circle represents the reliable subset. We can observe that for the first iteration, the cluster results contain some errors which includes the incorrect identities. However, with our threshold for the reliable subset, we only assign pseudo-labels to the correct samples. As the network be optimized on the

Table 3.5: Comparisons with the state-of-the-arts on VeRi-776 [6] (%).

Method	Supervision	R1	mAP
Ours	1/12 labeled	76.2	38.0
	1/6 labeled	78.7	43.6
	1/3 labeled	84.3	56.7
	fully-sup	90.1	64.7
RAM [138]	fully-sup	88.6	61.5
GRF-GGL [139]		89.4	61.7

Table 3.6: Comparisons with the state-of-the-arts on CUB-200 [7].

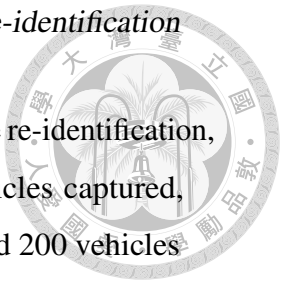
Method	Supervision	R1	NMI [140]
Ours	1/12 labeled	47.2	56.5
	1/6 labeled	48.1	58.7
	1/3 labeled	48.8	59.3
	fully-sup	49.7	59.3
Proxy [141]	fully-sup	49.2	59.5
Smart+ [142]		49.8	59.9

correct data, the cluster results will be more accurate. Furthermore, as the threshold be enlarged, more correct data will be assigned pseudo-labels for learning re-ID model.

3.4.11 Extension on other tasks

Although our SGC-DPL mainly tackled the semi-supervised setting practically in the task of person re-ID, it can be generally applied and extended to other tasks with the same setting. Therefore, we extended our SGC-DPL to the tasks of vehicle re-ID on the VeRi-776 [6] and image retrieval on the CUB-200 [7] datasets to verify the generalization ability. Different from other semi-supervised settings, the identities are also disjoint between labeled and unlabeled set in the training data and the ratio of the labeled data is also set as 1/3, 1/6 or 1/12. Compared to person re-ID, vehicle re-ID is a more challenge task owing to the large variation between the same vehicles captured from different views (i.e. rear and front) and the similar appearance between vehicles with the same car model, color and views. The image retrieval on CUB-200 is quite challenging, too. There are only 100 classes in the original training set and we would only have 17 labeled classes if the 1/6 setting is applied. Table 3.5 and 3.6 show the promising results that with the guidance of labeled set, our SGC-DPL can compete against the fully-supervised (fully-sup) state-of-the-arts approaches.

doi:10.6342/NTU202202005



Implementation details of the extension experiments For vehicle re-identification, the widely used VeRi-776 dataset [6] contains 776 different vehicles captured, which is split into 576 vehicles with 37,778 images for training and 200 vehicles with 11,579 images for testing. The training details all follow those in our main paper for person re-ID, which contains the same CNN backbone and the same three training tricks proposed in [78].

For image retrieval, we adopt CUB-200 dataset [7]. This dataset is a fine-grained bird dataset containing 11,788 images of 200 bird species. Following existing methods [141, 142], we use the first 100 categories with 5,864 images for training, and the remaining 100 categories with 5,924 images for testing. The ratio for the labeled data in our semi-supervised setting is also applied on the 100 training classes. For learning on labeled or pseudo-labeled data, we follow the triplet training network proposed in [142]. The reason for choosing [142] but not other state-of-the-arts is that adopting this purely triplet training can easily demonstrate the performance improvement with or without our SGC-DPL. In Table 3 & 4 of our main paper, the performances of the "fully-sup" setting produced by ourselves are the upper-bound of our SGC-DPL method on two datasets, which means we directly train the supervised network with all training data.

3.5 Summary

In this work, we presented a novel Semantics-Guided Clustering with Deep Progressive Learning (SGC-DPL) framework for semi-supervised person re-ID. Our core novelty lies in the proposed clustering algorithm, semantics-guided affinity propagation (SG-AP). Without the prior knowledge of the cluster numbers, we are able to cluster unlabeled data with the semantics-preserving guarantees, under the guidance of labeled data. Together with the progressive learning strategy, our model is able to select unlabeled data and assign soft pseudo-ID labels, which allows one to augment the labeled training dataset and thus results in improved

doi:10.6342/NTU202202005

re-ID performances. Qualitative and quantitative results confirm the design of our SGC-DPL framework, which performed favorably against recent semi-supervised methods while achieving comparable performances as fully-supervised ones did.

doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

Chapter 4

Image-based Unsupervised Person Re-identification

4.1 Introduction

Person re-ID tackles the problem of matching images of the same person across non-overlapping cameras, which has drawn much attention in recent years because of its wide applications in the intelligent surveillance system. As mentioned before, many existing works obtained great success by adopting supervised learning approaches with the aid of deep CNN [15, 13, 14]. However, these methods depend on large-scale labeled dataset that entails significantly high annotation costs, which are impractical to be applied in real-world scenarios. Thus, how to perform person re-ID in an "**unsupervised**" manner would be a critical yet challenging issue to solve.

Cross-domain unsupervised re-ID, which aims at learning re-ID on the target unlabeled domain with the aid of labeled data on a source domain, is one of the unsupervised problems that has been continuously addressed. To exploit the discriminative characteristics inherently accessible in the target domain, some recent works focused on clustering-based methods [143, 116, 131, 144] which acquire pseudo identity labels by clustering the unlabeled data. Thus, the estimated corre-

doi:10.6342/NTU202202005

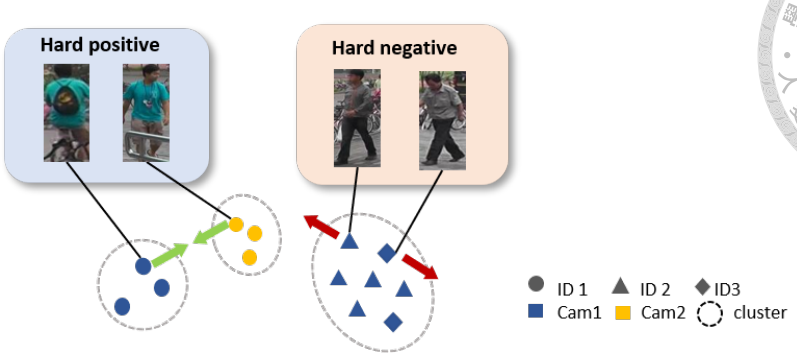


Figure 4.1: **Problems in clustering-based re-ID methods.** Motivated by the problems of hard training samples, our work aims to rectify them by pulling close the hard positive pairs and pushing away the hard negative ones.

spondence can help CNN for the unsupervised training. However, the underlying drawback of clustering-based methods is that the capability of re-ID model highly relies on the "quality" of the clustering results. In other words, the inconsistency between the generated pseudo-labels and the ground truth labels would undesirably degrades the re-ID performance, which generally arise from the misclustered hard training pairs. For instance, the same identity pairs captured under different cameras with intensive variations of the appearance could be possibly misclustered to different groups (we call it the **hard positive**). Or two people with similar appearance but only with subtle difference are likely to be clustered into the same group and be assigned with the same pseudo-label (we call it the **hard negative**). These two situations are harmful for re-ID model learning because they all degrade the discriminative ability for identifying people. With the above observations, we propose a ***Hard Samples Rectification (HSR)*** learning scheme which contains two components in the dual aspects: 1) an inter-camera mining technique (ICM) which utilizes the feature distribution and the camera ID information to resolve the shortcomings in the original clustering results caused by the hard positive pairs. 2) a part-based homogeneity technique (PBH) to split the possible hard negative pairs within a cluster into different groups by their features of local parts. Fig. 4.1 illustrates the common problems of the clustering-based methods and the objective

doi:10.6342/NTU202202005

of our proposed approaches.

Normally, data points with similar appearance will be projected into the near-by region in the feature space. In view of the intent of clustering, those adjacent feature points will be assigned with the same pseudo-labels. However, simply selecting positive pairs within each cluster is ineffective for model learning because it neglect the hard positive pairs that are clustered to different groups with appearance variations under different camera views. To better alleviate this problem, we consider the intrinsic properties beyond clustering results and propose the inter-camera mining technique (ICM), which utilizes the context information of the camera ID that can be easily obtained in the re-ID dataset. Specifically, for each anchor image in the training procedure, we will mine and pull close those possible hard positive pairs which are mutually similar but with **different camera views**. By taking the advantage of these hard samples, ICM can concomitantly rectify the cluster quality and steadily improve the re-ID performance.

As the number of training epochs grows, the capability of CNN model will encounter a bottleneck that some misclustered hard negative pairs would never be assigned to different clusters owing to their same pseudo-labels in CNN training. To refine the cluster containing hard negative pairss, we propose the part-based homogeneity technique (PBH) which forcibly regroups the imperfect cluster with part-based features. With the PBH, we can split the hard negative samples among a cluster and assign them with different pseudo-labels. The critical idea behind is that the part-based feature gives a finer insight of a person; thus, with our PBH, the hard negative samples among a cluster will have the chance be rectified and assigned with different pseudo-labels.

The main contributions of this work can be summarized as follows:

- We proposed an inter-camera mining technique (ICM) to mine potentially hard positive samples and alleviate the clustering bias of human appearance.
- The proposed part-based homogeneity technique (PBH) effectively regroups the imperfect clusters containing hard negative samples.

doi:10.6342/NTU202202005

- We conduct extensive experiments on two large-scale benchmarks and our HSR achieves promising performances in cross-domain unsupervised person re-ID.

4.2 Related Work

We introduced some related work of unsupervised re-ID, where the supervised re-ID has been introduced in Chapter 2&3.

To reduce the labeling effort, unsupervised cross-domain re-ID methods focus on leveraging prior knowledge of labeled source domain to improve the model performance on unlabeled target domain of interest. In UMDL [136], the proposed asymmetric multi-task dictionary learning aims to learn a shared dictionary across domains. Due to the rise of deep learning, recent works address this unsupervised re-ID problem based on deep learning frameworks. Some approaches focus on utilizing the image-to-image translation. SPGAN [145] aims to translate images from the source domain to the target domain while preserving the self-similarity of the original identity. HHL [132] introduce a Hetero-Homogeneous learning to enforce camera invariance by translating images between cameras and enforce the domain connectedness by constructing negative pairs between domains simultaneously. However, these methods depend upon the quality of generated images and overlook the discriminative information in target domain. While these methods aim to reduce the discrepancy between source and target domains, some other works leverage auxiliary information other than visual similarity. Yu *et al*. [130] propose to learn a soft multilabel vector for each unlabelled target image based on the labelled people from an auxiliary dataset as the reference information. Still, reference images from auxiliary dataset might not be precise enough to represent the characteristic of images in target dataset.

To mine the potential supervision in unlabeled target domain, another train of thought for solving the unsupervised problems are proposed by utilizing the

doi:10.6342/NTU202202005

clustering algorithm to estimate pseudo identity labels. In PUL [116], a progressive unsupervised learning is proposed to learn re-ID information based on iterations between clustering and CNN fine-tuning with reliable selected data. Yu *et al*. [143] develop a cross-view asymmetric metric learning based on clustering labels. Recently, SSG [131] propose a self-similarity grouping to mine potential similarities for both global and local features and PAST [146] introduce a new ranking-based triplet loss to avoid selecting unreliable samples from clustering results. However, these learning methods is seriously impeded by the incorrectly estimated pseudo-labels from clustering algorithms. Thus, our framework tackles the above problem beyond the original clustering methods from two aspects: pulling close and pushing away the potentially hard positive and hard negative pairs respectively, which can further rectify the unreliable pseudo-labels.

4.3 Hard Samples Rectifications

4.3.1 Overview of our HSR Learning Scheme

We first define the notation to be used in this paper. Given an unlabeled target dataset $\left\{I_{c,i}^{t}\right\}_{i=1}^{N_t}$ containing total N_t training images, where c denotes the camera ID of image $I_{c,i}^{t}$, and a source labeled dataset which serves as a preliminary knowledge base for learning re-ID, the goal of our model is to learn the discriminative ability to perform person re-ID on the target dataset. With those two kinds of data, we first learn a feature extractor ϕ on the labeled source dataset as a pretrained feature embedding function $\phi(\cdot, \theta_s)$, where θ_s is the parameters learned on the source domain. Then, the learning scheme is shown in Fig. 4.2. Same as [131], we utilize an unsupervised clustering algorithm called DBSCAN [133], which does not required the knowledge of exact number of identities, to generate the pseudo-labels for the target unlabeled images based on the extracted feature vectors $\phi(I_{c,i}^{t}, \theta_s)$. With each "estimated" pseudo-label y_i^t available, we can learn the re-ID model with the typical supervised manner, which consists of the cross-entropy

doi:10.6342/NTU202202005

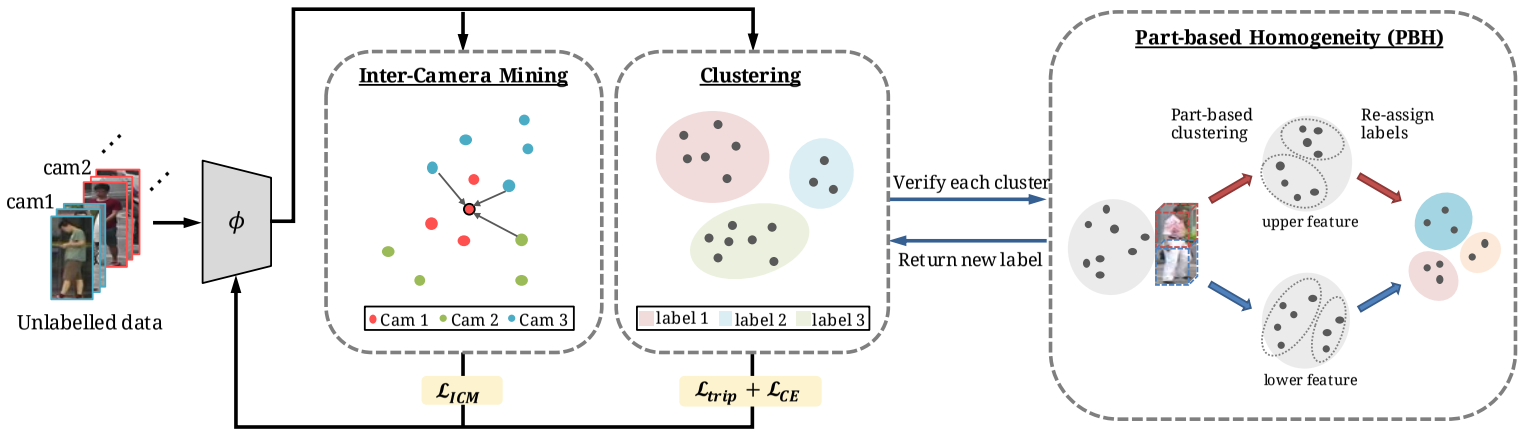
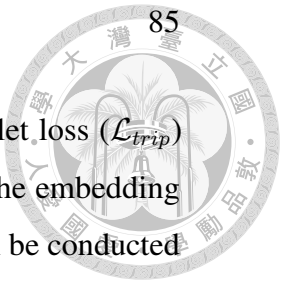


Figure 4.2: **Overview of the proposed HSR learning scheme.** Initially, the feature extractor ϕ is pretrained on the source dataset. For each iteration after clustering, we first rectify the hard negative pairs in the imperfect clusters with our part-based homogeneity technique (PBH) by splitting and regrouping the samples. The new refined pseudo-label is then employed as the supervised information to fine-tune the model along with the cross-entropy loss and triplet loss. In the other aspect, we apply inter-camera mining technique (ICM) as a complement of clustering results by pulling close the possible hard positive pairs which are mutually top-K closest to the anchor image and at the same time captured in different camera views.



loss ($\mathcal{L}_{CE}$) that helps correctly classify the identities and the triplet loss ($\mathcal{L}_{trip}$) for controlling the distance of the positive and negative pairs in the embedding feature space. The clustering and network optimization stages will be conducted iteratively, and the performance of re-ID model and the quality of clustering results will improve steadily. However, it will reach a bottleneck owing to the situations caused by the hard samples as mentioned above.

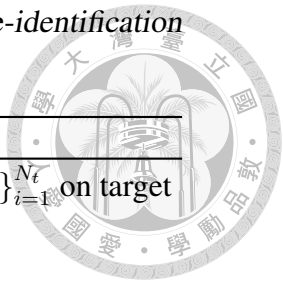
we propose Hard Samples Rectification (HSR) learning scheme, which dually rectifies the hard positive and negative samples with two components: inter-camera mining (ICM) and part-based homogeneity (PBH) techniques, as shown in Fig. 4.2. During training, ICM will mine possible hard positive pairs with different camera views and apply triplet loss to pull close those pairs in the feature space. On the other hand, PBH technique will refine the potential imperfect clusters by splitting the hard negative pairs within the same group.

4.3.2 Inter-Camera Mining

As mentioned in Section 4.1, hard positive pairs may be assigned to different pseudo-labels due to the variance of appearance under different cameras. After several iterations of clustering and network training, it will leads to a vicious cycle that the positive pairs used to optimize the model are only those with similar appearance, which goes against the goal of person re-ID to match people "across" cameras. Thus, we propose an inter-camera mining technique as a role of assisting the original clustering method to mine the hard positive samples.

In practice, shown in Algorithm 1, we first compute the similarity matrix $\mathbf{S} \in \mathbb{R}^{N_t \times N_t}$ for all target images, where the element in the i-th row and j-th column is the negative Euclidean distance of $\phi(I_i^t)$ and $\phi(I_j^t)$. Then, after sorting each row in descending order, we form the possible hard positive ranking list of each image by selecting its top-K closest images according to the matrix $\mathbf{S}$, denoted as $\mathit{Rank}(I_i^t)$ with a total length of K. It is worth noting that in order to emphasize on "inter-camera" positive pairs, **we remove those images captured**

doi:10.6342/NTU202202005



Algorithm 1: Inter-Camera Mining

Input: Image feature vectors $\{\phi(I_i^t)\}_{i=1}^{N_t}$ and its camera ID $\{c_i\}_{i=1}^{N_t}$ on target domain

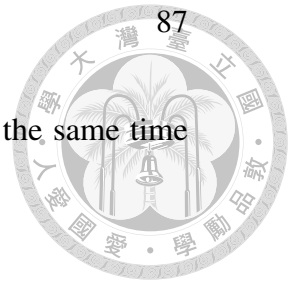
Output: Possible hard positive pairs

1: Calculate similarity matrix $\mathbf{S} \in \mathbb{R}^{N_t \times N_t}$.
2: **for** i=1 ; $i \leq N_t$; i=i+1 **do**
3: Sort $\mathbf{S}[i]$ in descending order.
4: *Rank*(I_i^t) = top-K images $\left\{I_j^t\right\}_{j=1}^{K}$ in $\mathbf{S}[i]$ with $c_j \neq c_i$
5: **end for**
6: **Mutual-K** : Choose image pairs (I_i^t , I_j^t) conformed to $I_j^t \in$ *Rank*(I_i^t) and $I_i^t \in$ *Rank*(I_j^t).
7: return all chosen pairs.

by the same camera as the image I_i^t. To ensure the robustness and correctness of our inter-camera mining, inspired by Dekel *et al*. [147], we additionally conduct a K mutually best-buddies pairs technique. That is to say, for every image I_j^t in *Rank*(I_i^t), I_i^t should as well be in *Rank*(I_j^t). Thus, only the image pair (I_i^t , I_j^t) that meets the above requirement would be taken into account as a reliable hard positive pair in the following CNN training.

With the mined hard positive pairs, we additionally apply the triplet loss $\mathcal{L}_{ICM}$, where the selection of positive samples is based on our ICM mining results. Notes that it differs from the original $\mathcal{L}_{trip}$ which samples the positive pairs based on the pseudo-labels generated by original clustering algorithm. As for the choice of negative samples of each anchor I_i^t in $\mathcal{L}_{ICM}$, we choose from the images with different pseudo-labels from I_i^t and at the same time not in its rank list *Rank*(I_i^t). Different from [146], we embed the accessible camera information and the mutual similarity, which benefits the correctness and the robustness of additional triplet pairs mining. With our $\mathcal{L}_{ICM}$ iteratively shortening the distance of these mined hard positive samples, it can progressively ensure the ability of our model to match

doi:10.6342/NTU202202005



person regardless of the variation between camera views and at the same time improve the quality of the clustering results.

4.3.3 Part-based Homogeneity

Different people with only subtle difference are possibly assigned with the same pseudo-labels, which would degrade the model ability to discriminatively identify people in detail. In the aim of separating imperfect clusters which possibly contain hard negative pairs, we develop a novel method called part-based homogeneity (PBH) as a rectification technique by utilizing the local features which provide finer information other than the global one. First, we need to define the imperfectness of a cluster and select the candidates for applying our PBH. To this end, we utilize Silhouette score [148], which is an evaluation metric for measuring how well a sample is clustered to its group without the requirement of the ground truth labels. By computing the mean Silhouette score of data in each cluster i, denoted as $mSil(i)$, we can further select the imperfect cluster with its $mSil(i)$ smaller than an empirically predefined threshold λ. Our proposed PBH technique is then applied on every selected cluster to refine the original clustering results, as illustrated in Fig. 4.3.

To start with, we split and pool the output feature maps of every sample in the selected cluster j into two parts: upper and lower features, which are formulated as $\{f_{u,i}\}_{i=1}^{N_j}$ and $\{f_{l,i}\}_{i=1}^{N_j}$, where N_j is the number of samples in cluster j. Then, we respectively employ the K-means clustering with K = 2 on $\{f_{u,i}\}_{i=1}^{N_j}$ and $\{f_{l,i}\}_{i=1}^{N_j}$ to observe the data distribution of the finer local features. Consequently, each sample is assigned with two temporary labels based on the groups of its upper and lower features, denoted as y_u and y_l. With the part-based label pair (y_u, y_l), we can re-assign new pseudo-labels to the samples in cluster j according to a look-up table, as shown in Fig. 4.3. The idea behind is that only the data with both similar local parts, which means the same (y_u, y_l), can be assigned with the same pseudo-label. Notably, because the number of contained ground truth identities in the imperfect

doi:10.6342/NTU202202005

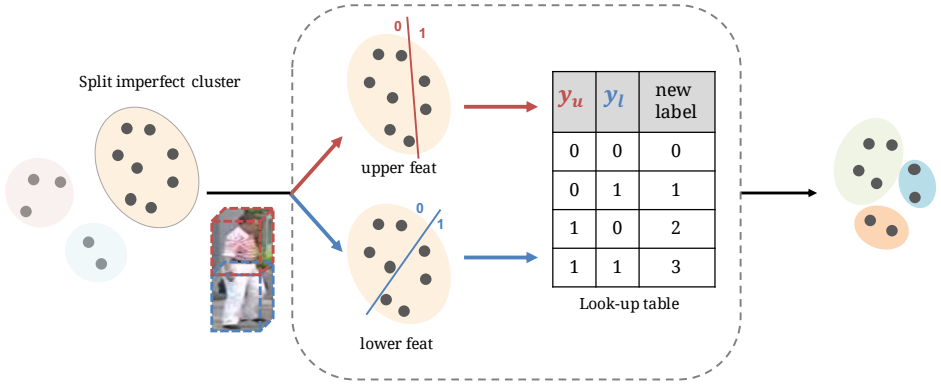


Figure 4.3: **Illustration of part-based homogeneity technique.** We extract local features of upper part and lower part for each sample in the imperfect cluster and apply K-means clustering on the local features respectively to obtain two kinds of part-based labels. With the two temporary local labels, the cluster is then split into at most four different groups according to the look-up table.

cluster is unknown, we suppose that if the cluster is defined as an imperfect one, it would contain at least two ground truth labels. Furthermore, the progress of iterative learning can ensure that even the selected imperfect cluster contains more than two ground truth labels, the split clusters would still have the chance to be defined as imperfect ones in the next iteration. In summary, by considering the local features, our PBH maintains the homogeneity within the new cluster and avoids assigning the same pseudo-label to globally similar hard negative pairs.

4.3.4 Optimization Procedure

For each iteration, after clustering the unlabelled data by DBSCAN, we would first verify the imperfect clusters and adopt the proposed PBH technique to refine the original estimated pseudo-labels. Then, we jointly utilize the triplet loss ($\mathcal{L}_{trip}$) and the cross-entropy loss ($\mathcal{L}_{CE}$) to optimize the CNN network with those updated pseudo-labels. Besides the positive and negative pairs sampled from the pseudo-labels, we also jointly adopted the triplet loss $\mathcal{L}_{ICM}$ according to our ICM sampling

technique. The overall loss function can be written as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \mathcal{L}_{trip} + \mathcal{L}_{ICM} \tag{4.1}$$

Specifically, for all the triplet losses, we adopt the improved version called weighted triplet loss proposed in [48] which can be formulated as follow:

$$\mathcal{L} = \sum_{a,p,n} F(w_p d(\phi(I_a), \phi(I_p)) - w_n d(\phi(I_a), \phi(I_n))), \tag{4.2}$$

where I_a indicates an anchor image, with its associated positive and negative pairs I_p and I_n , respectively. The w_p and w_n are the adaptive weights calculated according to the normalized feature distances between the anchor and its training pairs, and we use a soft-plus function $F(x) = \log(1 + e^x)$ as a penalty function instead of a typical hinge function with margin.

4.4 Experiments

4.4.1 Datasets and Evaluation Protocol

Same as before, we evaluate our approach on two large-scale person re-ID benchmarks: Market-1501 [12] and DukeMTMC-ReID [1], abbreviated as "Market" and "Duke" in the following sections. Rank-1 (R1) accuracy (%) and the Mean Average Precision (mAP, %) are used to evaluate the re-ID performance. In our experiments, the label information of the training data in the target domain is not available during the whole learning process. Notes that in the following tables, "Duke $\rightarrow$ Market" means we pretrained on the Duke dataset (source) and tested on the Market dataset (target), and vice versa.

4.4.2 Implementation Details

We adopt ResNet-50 [74] as our feature extractor ϕ and use the last 2048-d feature vector to represent the data in both training and clustering. Notes that we split the last feature map before average pooling into the upper and lower local feature

doi:10.6342/NTU202202005

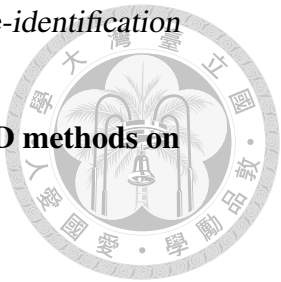


Table 4.1: **Comparisons with state-of-the-art unsupervised re-ID methods on Market and Duke.**

Methods	Duke → Market		Market → Duke	
	R1	mAP	R1	mAP
PUL [116]	45.5	20.5	30.0	16.4
CAMEL [143]	54.5	26.3	-	-
SPGAN [145]	58.1	26.9	46.9	26.4
HHL [132]	62.2	31.4	46.9	27.2
MAR [130]	67.7	40.0	67.1	48.0
PAST [146]	78.4	54.6	72.4	54.3
SSG [131]	80.0	58.3	73.0	53.4
*p*MR-SADA [149]	83.0	59.8	74.5	55.8
GDS-H [144]	81.1	61.2	73.1	55.1
HSR (Ours)	**85.3**	**65.2**	**76.1**	**58.1**

parts in our proposed PBH technique. The image size is 256×128 and augmented with random erasing and horizontal flip. Each mini-batch is with size 32, which consists of 8 randomly sampled pseudo-identities, and for $\mathcal{L}_{trip}$, each contains 4 sampled images in their cluster, but for $\mathcal{L}_{ICM}$, the 4 samples come from the possible hard positive ranking list. Empirically, we set K = 10 in the ICM, the number of local features = 2 (upper and lower) in PBH, and set the threshold $\lambda = mean(mSil) - 3std(mSil)$ in our PBH, where $mean(mSil)$ and $std(mSil)$ denote the average and standard deviation of $mSil$ of all clusters. We choose the SGD optimizer with the learning rate = 0.005 to optimize the model for 10 epochs in each iteration, where the total #iterations is 30.

4.4.3 Comparison with State-of-the-arts

We compare our proposed HSR with existing state-of-the-art unsupervised cross-domain re-ID methods on Market and Duke datasets in Table 4.1. Based on the common settings, we use Duke as the source dataset when test on Market and vice versa. We can see that our HSR outperforms all the compared methods significantly on both datasets.

Among the compared methods, PUL [116], CAMEL [143] and two latest

doi:10.6342/NTU202202005

Table 4.2: **Ablation studies of proposed methods in terms of R1 and mAP** (%).

Experimental setting	loss functions & components				Duke → Martket		Market → Duke	
	$\mathcal{L}_{CE}$	$\mathcal{L}_{trip}$	$\mathcal{L}_{ICM}$	PBH	R1	mAP	R1	mAP
Direct Transfer					50.1	20.9	36.2	18.3
Baseline	✓	✓			72.9	46.3	60.2	42.2
Baseline w/ PBH	✓	✓		✓	74.5	47.1	63.5	44.6
Baseline w/ $\mathcal{L}_{ICM}$	✓	✓	✓		83.8	63.3	73.5	54.4
HSR (Ours)	✓	✓	✓	✓	**85.3**	**65.2**	**76.1**	**58.0**

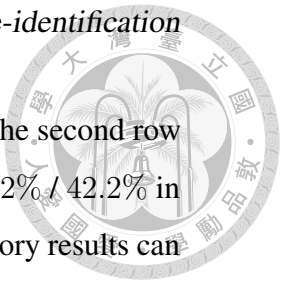
approaches, PAST [146] and SSG [131], also aim to exploit discriminative information in target domain based on pseudo-label estimation. Different from them, our HSR focusing on mining hard positive and hard negative samples to assist the unreliable clustering results thus gives a promising gain in the performance. Specifically, HSR achieve rank-1 accuracy = 85.3% and mAP = 65.2%, which outperforms the best of these approaches by margins of $\mathbf{5.3}\%$ and $\mathbf{6.9}\%$ in Market-1501. Similar improvement can be seen in DukeMTMC-ReID by achieving rank-1 = 76.1% and mAP = 58.0%, with a margins of $\mathbf{3.1}\%$ and $\mathbf{3.7}\%$. In summary, our method effectively enhance the model capability by alleviating the effect of hard cases in clustering-based methods.

4.4.4 Ablation Study

To further analyze our proposed method, we perform ablation study to evaluate the effectiveness of each component in HSR on both datasets. Results are shown in Table 4.2. First, we directly apply the model pretrained on source dataset to the target dataset, denoted as "**Direct Transfer**". The inferior performance due to the discrepancy between domains reveals the necessity of applying unsupervised method for learning person re-ID on target domain.

Performance of baseline method To directly learn the representation for unlabelled target domain, we utilize the clustering algorithm (DBSCAN) as a baseline method to generate pseudo-labels and train the CNN model in supervised manner

doi:10.6342/NTU202202005



with cross-entropy loss ($\mathcal{L}_{CE}$) and triplet loss ($\mathcal{L}_{trip}$). As shown in the second row of the Table 4.2, the baseline model achieves 72.9% / 46.3% and 60.2% / 42.2% in terms of R-1/mAP on Market and Duke respectively. The unsatisfactory results can be explained by the inaccurate pseudo-labels, thus degrades the model capability to identify people.

The effectiveness of ICM For ICM and PBH, We first validate the effectiveness of proposed ICM with $\mathcal{L}_{ICM}$ in the fourth row of Table 4.2. First, a significant improvement can be observed by adding ICM to the baseline model (Baseline w/ $\mathcal{L}_{ICM}$), which gains $\mathbf{17.0}\%$ and $\mathbf{12.2}\%$ in mAP when tested on Market and Duke. This demonstrates that our inter-camera mining adequately assist the model learning for identifying people regardless of the cross-camera scene variation.

To validate the robustness of our proposed technique, we also calculate the precision of the selected positive pairs from ICM. Specifically, for every samples I_i^t in the target domain, we compute the average of true positive rate of their corresponding *Rank*(I_i^t). It can progressively rise up to 82% in our iterative learning process, and notably, we can also reach a maximum rate of 17% in our *Rank*(I_i^t) of "hard positive samples", which possess the same ground truth identities yet are originally assigned into different clusters. This show that our ICM not only alleviate the impact of extensive variation across camera views but concurrently remedy the original clustering result and favorably generate possible hard positive pairs for re-ID learning.

The effectiveness of PBH The hard negative pairs being clustered to a same group is a critical factor that will hinder the model ability for distinguishing different people in detail. With our PBH, the imperfect cluster will be split into multiple group and re-assigned the new pseudo-labels. The performance results in the last row of Table 2.8 shows that adding with PBH, our final HSR learning scheme can improve $\mathbf{1.9}\%$ and $\mathbf{3.6}\%$ in terms of mAP on the two datasets compared to the "Baseline w/ $\mathcal{L}_{ICM}$".

doi:10.6342/NTU202202005

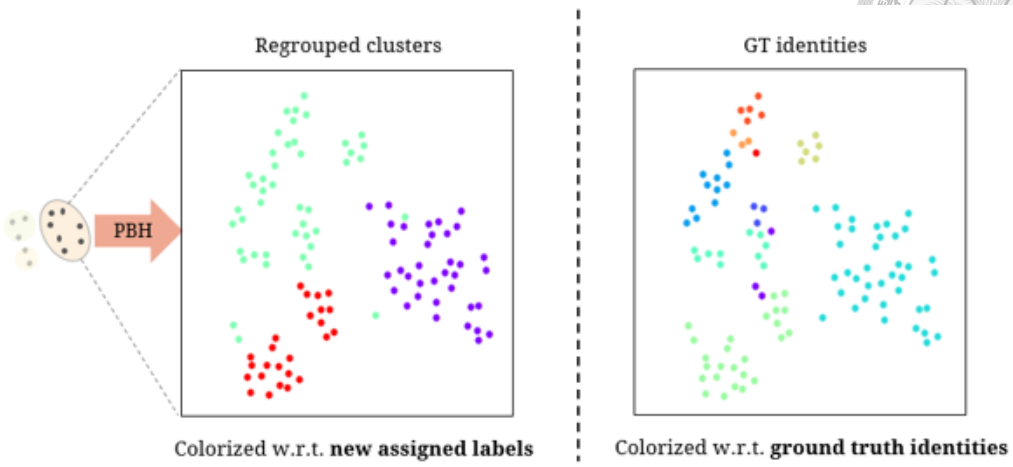


Figure 4.4: **Visualization of features within a sampled imperfect cluster via t-SNE.** Left: Multiple included ground truth identities within a single cluster, each of which is shown in a color. Right: Regrouped clusters from PBH. Samples with the same color indicates same new pseudo-label.

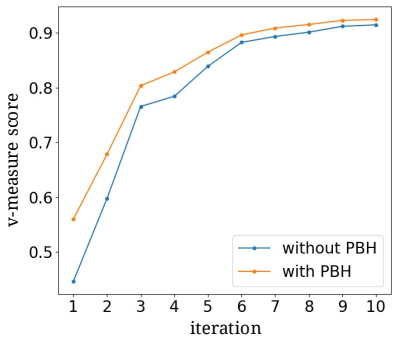
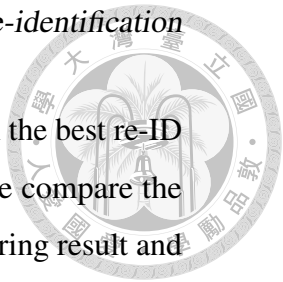


Figure 4.5: **Visualization of V-measure score w/ and w/o PBH.** V-measure score between the original clustering result and the one applied with PBH along the training iteration.

To further validate the effectiveness of PBH, we visualize the features, their ground truth labels and the new assigned labels in an imperfect cluster via t-SNE [150]. As illustrated in Fig. 4.4, the sampled imperfect cluster which is originally assigned with only one pseudo-label actually contains multiple ground truth identities indicated by different colors. By applying PBH to split and regroup the samples, where data with the same rectified pseudo-label are in the same

doi:10.6342/NTU202202005



color, we can refine the original clustering result and steadily obtain the best re-ID performance. In addition, to measure the overall effectiveness, we compare the V-measure score [151] in each iteration between the original clustering result and the one applied with our PBH, where V-measure is the overall evaluation measure of cluster quality that satisfies several desirable properties of cluster solutions. As illustrated in Fig. 4.5, with the aid of our PBH, the V-measure score constantly exceed the one of original clustering result, which indicates that our PBH fairly improve the cluster quality.

4.5 Summary

In this work, we introduce a hard samples rectification (HSR) learning scheme to address the issue of hard samples that degrades the performance in clustering-based methods. Specifically, we propose an inter-camera mining technique to match people under various camera views, and a part-based homogeneity technique to split hard negative pair within same cluster in a part-based manner. With our HSR, the model can learn a discriminative representation for unlabelled target images and receive a significant improvement of re-ID performance.

doi:10.6342/NTU202202005

Chapter 5

Layer-wise Filter Pruning for Neural Network

5.1 Introduction

Convolutional Neural Network (CNN) has been widely used since it attained significant improvement the first time on ImageNet Classification Challenge [152]. In recent years, various advanced architectures of cNN are proposed [153, 74], which achieve state-of-the-art performance on many computer vision tasks, such as image segmentation [154], object detection [33], and image super resolution [155]. The general trend of designing a well-performed network is making it deeper and more complex. However, it also increases the number of parameters and convolution operations at the same time, which means it will consume substantial storage and computational resources. Commonly, we can conduct the training stage of deep CNNs on high-performance GPU clusters, but for the Internet of Things (IoT) applications, we consider more about the inference stage on local devices with lower computation ability, such as mobile phones or surveillance cameras. Local computation on embedded system is more preferred than cloud-based solution owing to the real-time processing, better privacy, and no transmission bandwidth constraint. Under these considerations, it is much more difficult to employ those

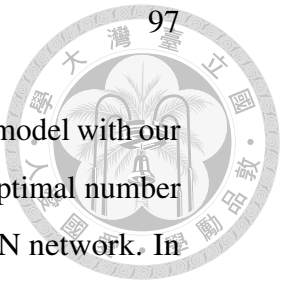
doi:10.6342/NTU202202005

high computation-demanding models on edge devices.

For these purposes, various works focus on optimizing the deep neural networks by removing the redundancy. In the past few years, Han *et al*. achieved impressive compression rates on VGGNet [153] by pruning parameters with small magnitudes [10]. With these compression methods, we can efficiently reduce the parameters in fully-connected layers or in the filters of convolutional layers. However, the pruning result on convolutional layers leads to sparse weight matrix with the same model architecture. Therefore, without alternative libraries or specific hardware accelerators conducting sparse operations, the compressed network with weight pruning cannot actually help reduce the computation time on general processors, such as GPUs and DSPs. Rather than weight pruning, filter removal (or filter pruning) is another aspect of pruning, which is beneficial for general computing platforms. CNNs with large capacity usually have redundancy among different filters; thus, Li *et al*. [8] and Chen *et al*. [27] first propose methods to optimize the model architecture by removing the entire convolution filter at a time according to different definition of filter importance. They both probe the filter importance layer-by-layer and remove a portion of unimportant filters layer-wisely.

This layer-wise filter pruning incurred a problem that which layer we should prune the model first. Based on the definition of filter sparsity [27], our previous work [29] first analyzes the sensitivity of a group of layers, such as the first, middle and last part. By pruning the less sensitive part, we can obtain lower performance drop with the same number of parameters left. Nonetheless, there are still no clear and systematic methodology for probing the sensitivity of a CNN network. In this paper, inspired by rate-distortion optimization (RDO) technique widely employed in video and image coding, we define the sensitivity of each convolutional layer and propose a new **computation-performance optimization (CPO)** algorithm to successively choose the proper layers to reduce the computation by filter removal when given some performance constraints. The layer with the lowest sensitivity will be pruned first, and by monitoring the performance sensitivity globally, we can

doi:10.6342/NTU202202005



derive the number of filters to remove for each layer. By pruning the model with our filter removal method and the CPO algorithm, we can find out the optimal number of channels in each layer of a deep well-trained but redundant CNN network. In order to prove the effectiveness of the proposed method, a deep CNN model for image super resolution (SR) and three models for image classification are employed. Compared with the previous work [29], our method achieves larger reduction of computation and parameters under the same performance constraint. Furthermore, we also compare the experimental results with the state-of-the-art filter pruning method [8] to show that the proposed method can reduce more parameters.

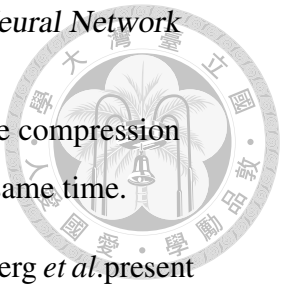
Specifically, our contributions are:

- Exploring the redundancy of convolutional layers with their sensitivity and filter sparsity,
- Proposing the Computation-Performance Optimization (CPO) method for systematically reducing computation operations under given performance drop constraint, and
- Applying CPO on SR and image classification tasks, which leads to significant improvement of computation reduction.

5.2 Backgrounds of Pruning

Several methods have been proposed in order to compress networks. Pruning is shown to be effective in reducing the network complexity and over-fitting. By eliminating weight connections, it sometimes even leads to performance gain. Early works such as Optimal Brain Damage (OBD) [156] and Optimal Brain Surgeon (OBS) [157] compute the saliency of each individual parameter through second order derivatives and remove those with lower saliencies. However, with the growing scale of modern network architectures, it becomes unrealistic to compute the saliency of every parameter. Therefore, Han *et al.*remove weights whose magnitudes are smaller than a certain threshold [10]. In addition to pruning, they

doi:10.6342/NTU202202005



incorporate weight sharing and Huffman coding to further boost the compression rate while still being able to retain the original performance at the same time.

From the perspective of filters within convolutional layers, Jaderberg *et al*.present an approximation of full rank filter banks as a combination of rank-1 filter basis and reduces the inference time [158]. Group-wise Brain Damage [159] revisits the concept of OBD and leverages the fact that convolutions are in practice matrix multiplications. They group together entries of the convolution filters and reduce them to zeros in a coordinated way. Anwar *et al*.introduce three levels of structured sparsity, which are channel-wise, filter-wise and intra-filters strided sparsity when it comes to pruning weights and filters [160]. They also point out that other compressing techniques (e.g. quantization) are orthogonal to pruning, and will enable greater computation and storage savings.

Then, many works start to directly remove filters [27, 8, 29], they point out that removing redundant filters to alter the network architecture can dramatically save the computation. Among those network compression and optimization techniques described above, filter removal is one of the methods with high potential since it can be used not only for model size compression but also for computation reduction for general computing platforms. In the well-performed work conducted by Li *et al*. [8], they use the *L1-norm* of every filter to rank the removing order. In addition, the number of filters to be removed in each layer is decided by observations and empiricism. Different from their *L1-norm* calculation and based on the sparsity definition in our previous work [29], in this paper, we provide a well defined metric, performance sensitivity (PS), to measure the layer's sensitivity for filter pruning. With the guide of PS and the constraint of a given expected performance drop, we can layer-wisely remove sparse filters and fine-tune the model to find the suitable number of filters to remove for each layer. Experiments on image classification and image super resolution are conducted to prove the effectiveness of the proposed method and our method can be comparable or even better than the state-of-the-art [8].

doi:10.6342/NTU202202005

5.3 Proposed Layer-wise Filter Removal

The operations of the i-th CNN layer involve convolving a 3-D tensor (input, $\mathbf{x_i} \in \mathbb{R}^{C_i \times Y_i \times X_i}$) with N_i different 3-D tensors (filters, $F_{i,n} \in \mathbb{R}^{C_i \times H_i \times W_i}$) to extract different features and then generating a 3-D feature map tensor (output, $\mathbf{y_i} \in \mathbb{R}^{N_i \times Y_i' \times X_i'}$), where C_i, Y_i, X_i are channel, height and width of the i-th input tensor, H_i, W_i are height and width of one filter, and N_i is the number of convolution filters in the i-th layers, which is equal to C_{i+1}, the number of channels of the next layer. Y_i', X_i' are slightly different from Y_i, X_i owing to the boundary of convolution, and the output tensor is also the input tensor of the next layer. The filter pruning procedure is based on removing one complete filter of the i-th layer at a time, which reduces $C_i H_i W_i X_i' Y_i'$ operations. And furthermore, it will also eliminate one feature map channel at the next layer, so it will concurrently reduce $N_{i+1} H_{i+1} W_{i+1} X_{i+1} Y_{i+1}$ operations.

Given a well-pretrained CNN model, in the network optimization process with layer-wise filter removal, the number of filters and which filters to be removed among each layer are two important parameters we need to determine. Within one layer, we define the filter sparsity and rank the possible candidates to be removed; between layers, we propose a **Computation-Performance Optimization** (CPO) algorithm to take sparsity and performance sensitivity, which will be defined later, into consideration to iteratively determine the sequence of layer-wise reducing factors (the ratio of removed filters). Specifically, owing to different tasks or applications, the users may have an expected acceptable performance drop after model pruning. Therefore, we can adaptively prune the CNN network according to any expected drop. Fig. 5.1(a) illustrates our CPO system. Given a trained CNN network and an expected performance drop by the user, the CPO system will iteratively determine the reducing factor of every convolutional layer for the "Filter Removal Process". After pruning and retraining, we will do "Performance Evaluation" and start the next CPO iteration to generate the next reducing factor.

doi:10.6342/NTU202202005

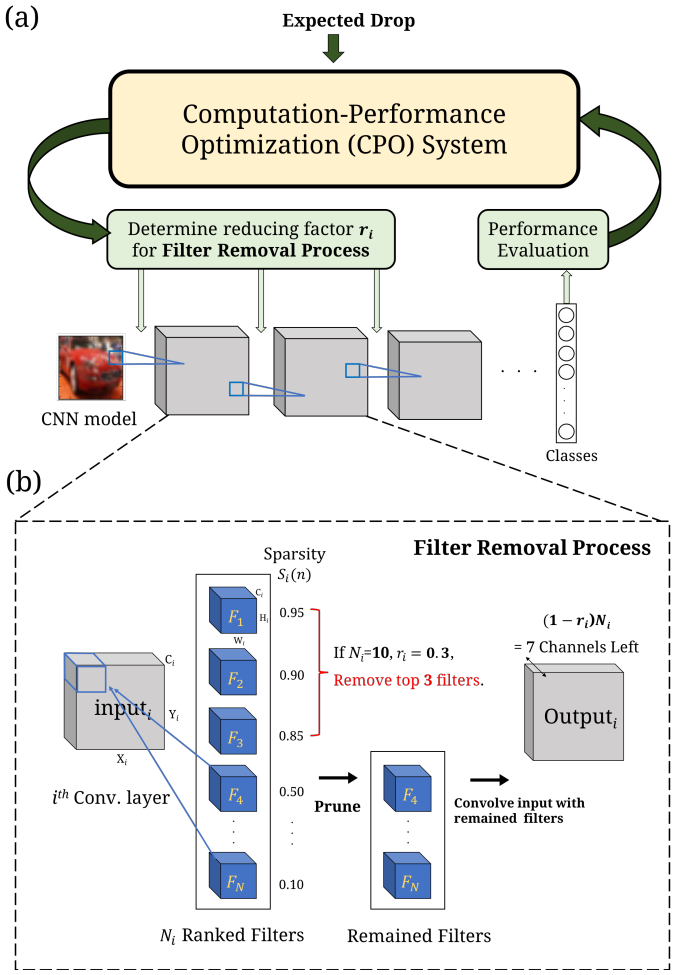


Figure 5.1: **(a) Flow of Conducting CPO System. (b) Intra-layer Filter Removal Process.** Figure (a) illustrates the whole CPO pruning algorithm. Given an expected drop by the user, the system will iteratively prune the well-trained CNN model by determining the layer-wise reducing factors, and evaluate the model performance to start the next iteration. Figure (b) demonstrates the intra-layer filter removal process with a given reducing factor. We first rank the filters in the i-th layer by sparsity and remove the first $N_i r_i$ filters. When $N_i = 10$,$r_i = 0.3$, after pruning, 7 filters will exist and the output feature map will remain 7 channels, too.

doi:10.6342/NTU202202005

5.3.1 Definition of Sparsity

The criterion of redundancy is defined layer-by-layer according to their weight distribution. For a specified layer i, we use M_i to represent the mean value of all absolute filter weights:

$$M_i = \frac{\sum_{n,c,h,w} |F_{i,n,c,h,w}|}{N_i \times C_i \times W_i \times H_i}, \tag{5.1}$$

where n, c, h, w are the indices of the filter tensor F. Then the Sparsity $S_i(n)$ of the n-th filter at layer i can be written as:

$$\begin{aligned} S_i(n) &= \frac{\sum_{c,h,w} \sigma(F_{i,n,c,h,w})}{C_i \times W_i \times H_i}, \\ \sigma(x) &= \begin{cases} 1, \text{if } |x| < M_i \\ 0, \text{otherwise} \end{cases} \end{aligned} \tag{5.2}$$

In other words, for a specific layer i, if a filter has several coefficients which are less than the mean value, $S_i(n)$ is close to 1, which means this filter is more redundant than others. We then rank the filters in the i-th layer in descending order according to their sparsity values. When we conduct CPO at the i-th layer afterwards, the filters ranked higher will be removed first.

5.3.2 Definition of Reducing Factor

Considering that the number of convolution filters vary from layer to layer, it is not convenient for us to compute the exact number of redundant filters when conducting CPO algorithm. We thus define the reducing factor r_i, $0 \leq r_i \leq 1$ for the i-th layer. The value of r_i is the ratio of the numbers of removed filters to all filters at the layer i. Fig. 5.1 (b) demonstrates an example of intra-layer filter removal process. For the i-th layer, there are $N_i = 10$ filters, and originally the output feature map will contain 10 channels. We first calculate the filter sparsity $S_i(n)$, and construct the ranked sparsity list. If we set $r_i = 0.3$, $N_i r_i = 3$ filters on the top of the ranked list will be removed and 3 channels of the output feature map will be removed as well.

doi:10.6342/NTU202202005

5.3.3 Concept of Computation-Performance Optimization

The next step is to determine the reducing factors while considering the global effects of filter removal across layers. Inspired by the rate-distortion optimization (RDO) technique in video and image coding [161, 162], we propose the concept of computation-performance optimization (CPO) for CNN optimization. In video and image coding systems, RDO is the method to determine the optimal bit allocation to achieve the minimized distortion δ^* under a given bit-rate constraint R_c:

$$\delta > \delta^* \; \forall \; R < R_c \tag{5.3}$$

To solve this problem, in [162], a post-compression rate-distortion optimization (PCRDO) is proposed for image coding. An image is decomposed into several small coding unit CU_i, and $\Delta\delta_i/\Delta R_i$ is calculated, where $\Delta\delta_i$ is the global distortion reduction when coding unit CU_i in included, and ΔR_i is the required bitrate for this coding unit. Given any λ, the set of coding units

$$\{CU_i \;\; |\Delta\delta_i/\Delta R_i > \lambda\} \tag{5.4}$$

is an RDO solution under the total bit-rate

$$R_c = \sum\{R_i \;\; |\Delta\delta_i/\Delta R_i > \lambda\}. \tag{5.5}$$

Inspired by PCRDO, we model the filter removal process as a computation-performance optimization (CPO) problem, that is, we would like to achieve the minimized performance drop D^* under a given computation budget ζ_c. A group of filters is then employed as the small unit, and the associated $\Delta D/\Delta\zeta$ is derived for selecting the filter to be removed. Similarly, the units with larger $\Delta D/\Delta\zeta$ are kept, that is, the units with smaller $\Delta D/\Delta\zeta$ are removed to achieve computation-performance optimization. Note that $\Delta D/\Delta\zeta$ can be viewed as a kind of performance sensitivity.

doi:10.6342/NTU202202005

5.3.4 Definition of Performance Sensitivity

For one specific convolutional layer, we can deduce the potential redundancy of filters by calculating the filter sparsity with (5.2), and consequently we can first remove the filters with high sparsity. However, for the entire CNN model, we have no clear criterion for determining which layer we can conduct filter pruning first. Based on the concept of CPO, we define the Performance Sensitivity (PS) of the i-th convolutional layer with the change of reducing factor (r_i) and performance drop (D):

$$\mathrm{PS}_i(\Delta r_i, \Delta D) = \frac{\Delta D}{\Delta \zeta} = \frac{\Delta D}{\Delta r_i \times W_i \times H_i \times C_i}, \tag{5.6}$$

where D is the performance drop, which is a positive value, and ΔD is the change of drop between two pruning steps. Δr_i is the change of reducing factor, and the whole denominator part approximates the computation change $\Delta \zeta$ for removing a portion of filters in the i-th layer.

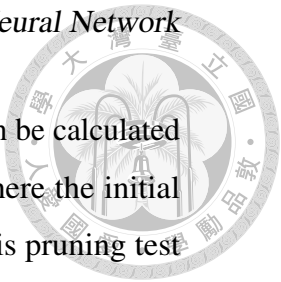
The Performance Sensitivity (PS) represents the aptness of being pruned for a layer. When conducting our Computation-Performance Optimization (CPO) with reducing factor, it is more likely to assign higher reducing factors to the layers with lower PS values.

5.3.5 Computation-Performance Optimization

The problem left now is to determine the exact number of filters allowed to be removed for each single layer without apparent performance drop. Based on the concept of CPO, the Performance Sensitivity (PS) is employed to balance the trade-off between "Computation Reduction" and "Performance Drop". The whole algorithm flow is demonstrated in Algorithm 2.

First, we need to find out the PS value of each layer. By emulating the method in finding layer-wise sensitivity [163], we iteratively set $r = 0.5$ to halve the number of filters in one layer and meanwhile keep the rest untouched. With those remained parameters, we retrain the model for few epochs to fine-tune the model

doi:10.6342/NTU202202005



until convergence. After obtaining the performance drop D_i, PS_i can be calculated with $\Delta r_i = 0.5 - 0 = 0.5$ and $\Delta D_i = D_i - 0 = D_i$ in (5.6), where the initial reducing factor and the initial drop are both zero. Finally, with this pruning test for each layer respectively, we can construct the sensitivity list for the subsequent steps. The reason why we choose half over other fractions to probe the sensitivity is two-fold. For one, if a too small portion of filters is pruned away, the performance will be quickly restored through the retraining process, resulting in unstable PS values. For the other, it inherently fits the property of Binary Search (BS) [164] to find out the appropriate reducing factor for the consecutive procedures in the proposed CPO algorithm.

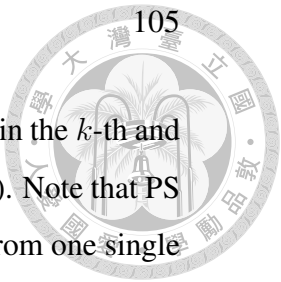
Second, with the PS list obtained by setting $r_i = 0.5$ for each layer i respectively, we sort it in ascending order and start removing filters from the least sensitive layer, which is called "the current layer" in following descriptions. Following the Binary Search order, we increase the reducing factor from $r_i = 0.5$ to $0.75, 0.875...$ for the current layer, unless one of the following conditions is met:

1. When the performance drop becomes intolerable.
2. When the updated PS value of the current layer becomes larger than that of the runner-up.
3. When the layer run out of filters to remove.

The performance drop is intolerable when it exceeds the expected drop D_{exp}, which is decided by the user of our CPO system. When it happens, we will take a step back by following the binary search order. That is to say, instead of removing $N_i r_{i,k}$ filters which causes unexpected drop for the k-th step, we remove $N_i \frac{(r_{i,k}+r_{i,k-1})}{2}$ filters. If it is still intolerable, we will search the suitable reducing factor for the current layer until the drop becomes smaller than D_{exp}. Next, we move on to removing the filters at the next least sensitive layer.

The second condition appears during the sensitivity updating within the current layer. The goal of updating the PS value is to evaluate the performance change owing to the computation reduction, or we can say the incrementally increased

doi:10.6342/NTU202202005

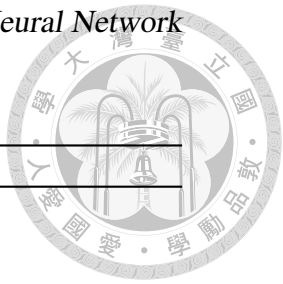


reducing factor. That is, for example, if $r_{i,k} = 0.75$ and $r_{i,k-1} = 0.5$ in the k-th and $(k-1)$-th steps, respectively, Δr_i is set as $0.75 - 0.5 = 0.25$ in (5.6). Note that PS will become higher and higher when we continue removing filters from one single layer, which is a kind of diminishing marginal utility. Therefore, when we detect that the PS value of the current layer has grown larger than the runner-up layer i' in the sensitivity list, it strongly suggests that this reducing factor influences the whole performance too much, and we will then switch to the next layer i'.

In addition, once we decide the number of filters to remove in the i-th layer, the PS value of the $(i+1)$-th layer in the original sensitivity list is also updated because of channel reduction. As mentioned in Sec. 5.3.2, if we remove n_i filters in the current layer i, every filter in the $(i+1)$-th layer will also be reduced by n_i channels, which causes the reduction of computation at that layer. Therefore, we need to modify the denominator term C_{i+1} of (5.6), and it will simultaneously increase the PS_{i+1} value.

The procedure described above will iterate through all of the layers that are available for filter pruning. For the experiments in SR, we will not prune the last convolutional layer because the number of filters at that layer is only one. Meanwhile for the experiments in image classification, the available layers for filter pruning in our CPO algorithm differ from model to model. It depends on the original architecture (VGGnet, ResNet and so on) or whether the hidden layers exist between the last convolutional layer and the output layer. If there is only one linear layer which maps the feature map of the last convolutional layer to the class prediction, we will not remove the filters in the last convolutional layer because it will correspondingly remove some of the parameters in the next linear layer and meanwhile greatly influence the prediction performance. We will discuss the details in Sec. 5.4.

doi:10.6342/NTU202202005



Algorithm 2: Flow of CPO

Input : A trained CNN model, An expected performance drop D_{exp}

Start probing the Performance Sensitivity as follows ;
repeat through every convolutional layer
    Prune the i-th layer with $r_i = 0.5$ and obtain the corresponding drop (D_i) after retraining few epochs;
    Calculate the Performance Sensitivity (PS_i) according to (5.6) and add it to PS list;
    Recover the i-th layer by setting $r_i = 0$;
until *all available layers are iterated through*;
Start determining the actual r_i for every layer as follows;
repeat for picking the minimum i from sorted PS list
    Start pruning the i-th layer as follows ;
    Loop *k for $r_{i,k}$ starting from 0.5 in BS order :*
        Prune the layer with $r_{i,k}$ for the k-th iteration and retrain the model for few epochs;
        Update the PS value and check the termination conditions;
        if *it meets the conditions 1) or 2).* **then**
            repeat
                Step back to $r_i = \frac{(r_{i,k}+r_{i,k-1})}{2}$;
            until *the drop is acceptable*;
            break the loop;
        else if *it meets conditions 3).* **then**
            break the loop;
        else
            Go to next iteration with larger reducing factor.
    EndLoop
    Update the PS list owing to channel reduction;
    Remove the i-layer in PS list and sort it again;
until *sorted PS list is empty*;

doi:10.6342/NTU202202005

5.3.6 FLOPs and Parameters Calculation

After conducting CPO, we can significantly remove a large number of parameters, which will result in smaller storage size and less floating point operations (FLOPs). To quantify the operations remained in all convolutional layers, we follow the equations:

$$FLOPs = \sum_i N_i \times (W_i \times H_i \times C_i) \times (X_i \times Y_i), \tag{5.7}$$

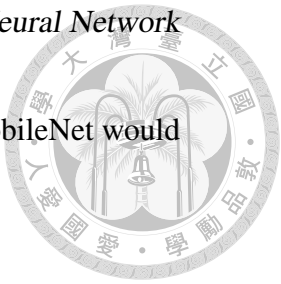
where W_i, H_i, and C_i are the width, height, and number of channels of N_i filters in the i-th layer respectively, while X_i and Y_i are the width and height of the convolved input. To calculate the parameters in the meantime, we just remove the $(X_i \times Y_i)$ terms of the shifting window operations of convolution in (5.7), and thus we can obtain the parameter size inside all convolutional layers.

5.4 Experiments

5.4.1 Experimental Setup

Model and Datasets We conduct experiments on two tasks to demonstrate our CPO algorithm, which are image super-resolution and image classification. For image super-resolution, we employ the residual CNN model in Very Deep Super Resolution [155] (VDSR). This model is constructed only with convolutional layers; therefore, the model size and the computation time will not be influenced by the fully-connected layers. As for image classification, we construct a modified VGG-19 model [153], a self-designed ResNet-32 model and a self-designed MobileNet-22 model, which are modified from ResNet-34 [74] and Mobilenet-v1 [165], respectively. These three networks are dedicated models for training and testing on Cifar-10 dataset [166]. Cifar-10 is a small dataset including 10 categories, and all of which are composed of 3-channel RGB images with the resolution of 32×32. The following experiments will be conducted on the four

doi:10.6342/NTU202202005



models. Notes that owing to the page length, the experiments of MobileNet would be only demonstrated in our paper [28].

Setup for Comparison with Baseline Methods For the tasks mentioned above, we first construct a baseline method as what was done in our previous work [29], called **Uniform Removal (UR)**. UR will remove filters with a fixed reducing factor r_i across all available layers. After that, we retrain for few epochs to recover the performance. The number of retraining epochs may be slightly larger than that in [29] because we find that the performance drop will be stable if we make the retraining stage converge. To prove the effectiveness of our method, we perform CPO on the original model and set the expected drop D_{exp} same as the validation drop obtained from UR. We then compare the remaining parameters and FLOPs between CPO and UR. We also conduct some experiments with D_{exp} set as other values to observe the trade-off between performance and computation. The expected drop and the drop monitored in CPO algorithm are all tested with the validation set, which is seen but not trained during the procedure. After our CPO algorithm, we will test the model with a totally unseen testing set to evaluate our model performance. It is worth noticing that once one filter is removed, the number of channels in every filter of the next layer will consequently decrease by one. This is the reason why the actual parameters removed will be more than the percentage of filters we attempt to remove in UR. In addition, we conduct CPO experiments from lower D_{exp} to higher one, and we will use the model pruned by lower D_{exp} as the base model to continue the next experiment of higher D_{exp}.

Hardware Simulation Our proposed method is trying to reduce the entire convolutional filters that have less contribution to the network. Therefore, it genuinely reduces not only the parameters but also the FLOPs when performing on any hardware platform. We simulate the operations on the Systolic CNN AcceLErator Simulator (SCALE-sim) proposed by ARM [167] to prove the reduction of computation. This tool can help generate the computation cycles and the DRAM

doi:10.6342/NTU202202005

read/write bandwidth. In the following sections, we will only show the total cycle count of network inference stage for simplicity.

Shallow and Deep Model Comparison Aside from comparing the results between UR and CPO, we also design a shallow model with comparable parameters to the deep model pruned by our CPO system. The results will show the trade-off between training time and the performance of the model.

5.4.2 Experiments on Fully Convolutional VDSR network

We obtain the well-trained VDSR model from the official website [168]. Because we have no knowledge of which validation set the model was originally validated on, we choose Set5 [169] as our validation set and Set14 [170] as the final testing set, both with $\times 2$ scale. These two datasets are commonly used in image super-resolution tasks. The model structure is a 20-layer residual CNN as illustrated in Fig. 5.2. The input is an interpolated low-resolution (ILR) image with one channel (Y channel), and the output is the derived high-resolution (HR) one. During training and validation, for convenience, we will use input images with sizes of 41×41 that are randomly cropped from the dataset, which is same as the settings in VDSR. When performing testing phase, the input and output sizes can be arbitrary depending on the testing image. Among the convolutional layers, each of the first 19 layers has 64 filters, but only one filter exists in the last layer to generate the residual part, which is added by the low resolution image to finally generate the high-resolution one. Since the last layer of VDSR has only one filter, we perform filter-pruning on the rest of the layers. There are two main reasons for conducting our experiments on VDSR. First, the fully-convolutional model can help us clearly evaluate the performance of our filter removal method. Second, since the task of SR is difficult in computer vision, we are interested in finding the redundancy of an SR model.

Table 5.1 shows the experimental results of pruning VDSR network. We use

doi:10.6342/NTU202202005

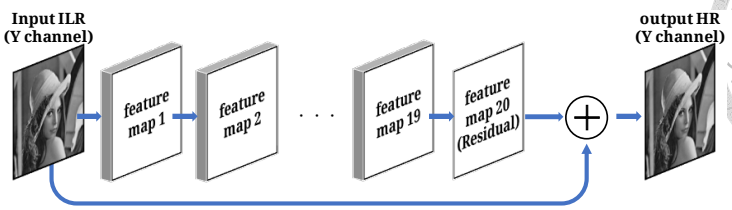


Figure 5.2: **Network structure of VDSR.** The 20-layer residual network is composed of 20 times convolutions and nonlinear operations. There is no pooling layer, so the output of the residual part has the same size as the input. ILR means Interpolated Low-Resolution and HR represents High-Resolution.

PSNR (dB) to evaluate the performance. The first four columns are the model settings and performance results. The last column is the summed computation latency calculated in cycle count after performing the network inference on one input image, which has the size of 41×41 in this case, with the SCALE-sim neural network simulator. In all tables, "Val" means validation and "Params" represents parameters. First, the upper part is the performance of the original trained model, which achieves $40.26dB$ on the validation set and $33.08dB$ on the Set14 testing set. The baseline UR results are in the middle. Owing to the unchanged size of the feature maps among the VDSR network, the percentage of parameters and FLOPs remained, which are calculated by (5.7), are the same. Note that the PSNR drop of the results are slightly different from those in [29] because we conduct 5 retraining epochs instead of 3 as mentioned above. We show three results with different reducing factors. When we increase the reducing factor, the remaining parameters decrease but the performance will also drop accordingly.

At the bottom part of Table 5.1, it shows the results with our proposed CPO. The first column is the user expected drop (D_{exp}), and the second column is the final validation drop after the last pruning and retraining iteration. Note that the three settings whose expected drops are marked with ($*$) correspond to the three baseline UR methods. It can be seen that CPO can achieve results superior to UR for every reducing factor. With comparable PSNR drop on unseen testing set, CPO

doi:10.6342/NTU202202005

Table 5.1: **Experimental results of VDSR.** This table shows the settings and performance results of the original model, the models pruned after UR and our proposed CPO. Set14 is our unseen testing set and the last column is the cycle count after our SCALE-sim CNN hardware simulator.

Original Model				
Reducing Factor	Val PSNR (dB)	Params / FLOPs	Set14 PSNR (dB)	Latency (Cycle)
0	40.26	6.7×10^5/ 1.1×10^9	33.08	7.6×10^6
Uniform Removal (UR) [29]				
Reducing Factor	Val Drop (dB)	Params / FLOPs Remained (%)	Set14 Drop (dB)	Latency (Cycle)
0.0625	0.10	87.91 / 87.91	0.11	5.9×10^6
0.1250	0.13	76.60 / 76.60	0.19	5.5×10^6
0.2500	0.26	56.31 / 56.41	0.29	3.8×10^6
CPO (Ours)				
D_{exp} (dB)	Final Val Drop (dB)	Params / FLOPs Remained (%)	Set14 Drop (dB)	Latency (Cycle)
0.10*	0.08	**72.31 / 72.31**	0.12	$\mathbf{5.2 \times 10^6}$
0.13*	0.14	**64.68 / 64.68**	**0.15**	$\mathbf{4.7 \times 10^6}$
0.18	0.18	54.25 / 54.25	0.2	4.0×10^6
0.26*	0.26	**51.52 / 51.25**	**0.28**	$\mathbf{3.8 \times 10^6}$
0.32	0.32	45.85 / 45.85	0.34	3.4×10^6

is able to achieve more computation reduction. It can save about 50% of parameter storage and computation with minor performance drop (Val:0.26dB/Test:0.28dB). In addition, it also shows that about 30% of the computations are redundant given only a negligible drop (Val:0.05dB/Test:0.12dB). For a model purely containing convolutional layers, our method will surely alleviate the computational burden on the hardware. Furthermore, we also conduct two other expected drop settings to observe the trend between performance drop and computation reduction. We find that we only get marginally additional computation reduction when we have already removed the majority of redundancy in the model. The reason we speculate is that the difficult SR task is a regression task rather than a classification task, so

doi:10.6342/NTU202202005

Table 5.2: **Comparison of Shallow Model and Pruned Deep Model (VDSR).** This table shows the trade-off between training a shallow model and pruning a given well-trained deep model. We choose the CPO results with $D_{exp} = 0.32$ to do the comparison.

	Val PSNR (dB)	Params / FLOPs Remained (%)	Set14 PSNR (dB)	Training time (epochs)
Original	40.26	100 / 100	33.08	100
Shallow	39.82	44.54 / 44.54	32.62	**80**
CPO(0.32)	**39.94**	45.85 / 45.85	**32.74**	305

there might be not so much redundancy in the VDSR model. However, we can still eliminate 15% more of the parameters than UR method did when there is a negligible drop.

Designing a shallow model can also reduce the computation cost. However, a non-deep network may sometimes fail to perform well. Table 5.2 shows our experimental results. The shallow VDSR model is composed of 10 convolutional layers, and the remained parameters are comparable to the model pruned after CPO with the settings of $D_{exp} = 0.32$. We can see that our method can perform well on the validation and testing sets. We use the number of training epochs to evaluate the training time. Although our method can perform better, the trade-off is that it may take more time to retrain the model with conducting CPO than simply train the shallow model from scratch. In addition, the retraining time is also higher than that in UR (5 epochs) which is mentioned in Sec. 5.4.1. we think that increased retraining time is not the main consideration. The purpose of our algorithm is to reduce the burden when we deploy the model on some hardware devices at last. Therefore, we can use a powerful GPU first to eliminate the redundancy as much as possible by reasonable offline training, and then deploy the model on those hardware devices.

doi:10.6342/NTU202202005

5.4.3 VGG-19 on Cifar-10 Image Classification

The following are the experiments on Cifar-10 image classification task with three dedicated neural network models. The purpose of all our experiments is to observe the impact of removing filters in convolutional layers. Hence, there are no hidden fully-connected (FC) layers in our self-designed models. We only use one linear layer to map the flattened feature map after the last pooling layer to the classes prediction.

The first self-designed model is a modified VGG-19 network, which contains 16 convolutional layers and one linear layer. There are five convolution parts, and the settings in each part such as the number and sizes of convolution filters are all the same as the original network. The final output layer which contains 512×10 parameters will map the feature vector to the 10 classes output. Moreover, we implemented Batch-Normalization (BN) [75] after every convolution layer, which are not shown in the table. BN layers store additional statistical information of the preceding feature maps and also contain the parameters of linear shifting operation; therefore, the removal of filters will result in corresponding reduction of the variables in BN layers, too. Furthermore, we preserve all the filters in the last convolutional layer in order not to accordingly influence the last output layer, which contains less parameters but plays a role in the feature-to-class transformation. In summary, filter-pruning will only be performed on the first 15 layers in our modified VGG-19 network to gain undistracted insight into how the removal of filters in convolutional layers affects the performance.

The experimental results of VGG-19 on Cifar-10 are shown in Table 5.3. We randomly choose ten percent of the training data as our validation set. Then, we train the unpruned VGG-19 network by ourselves. It achieves 93.16% validation accuracy and 92.98% testing accuracy. Based on the model, we conduct both UR and our CPO algorithm. Same as the notation in Table. 5.1, we use $(*)$ to represent the value of D_{exp} which is set as the "Val Drop" evaluated after performing UR.

It can be seen in Table 5.3 that the remaining parameters of the new architecture

doi:10.6342/NTU202202005

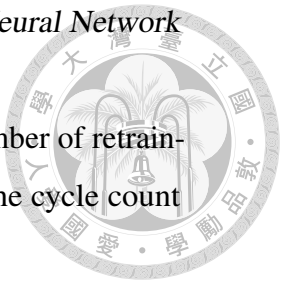


Table 5.3: **Experimental results of VGG-19 on Cifar-10.** The number of retraining epochs after every pruning iteration is 8. The last column is the cycle count after our SCALE-sim CNN hardware simulator.

Original Model				
Reducing Factor	Val Acc (%)	Params / FLOPs	Testing Acc (%)	Latency (Cycle)
0	93.16	2.0×10^7/ 3.9×10^8	92.98	3.7×10^6
Uniform Removal (UR) [29]				
Reducing Factor	Val Drop (%)	Params / FLOPs Remained (%)	Testing Drop (%)	Latency (Cycle)
0.250	1.60	58.47 / 56.78	1.25	2.0×10^6
0.375	2.34	41.84 / 39.72	1.89	1.4×10^6
0.500	3.72	27.96 / 25.70	3.52	9.5×10^5
0.625	5.32	16.84 / 14.72	4.01	5.1×10^5
CPO (Ours)				
D_{exp} (%)	Final Val Drop (%)	Params / FLOPs Remained (%)	Testing Drop (%)	Latency (Cycle)
0.25	0.14	28.57 / 54.09	0.50 (0.40)	1.6×10^6
0.58	0.44	15.68 / 42.26	0.83 (0.76)	1.2×10^6
1.60*	1.58	**10.23 / 28.11**	1.83 (1.34)	$\mathbf{7.9 \times 10^5}$
2.34*	2.16	**5.75 / 20.18**	2.49 (**1.73**)	$\mathbf{5.5 \times 10^5}$
3.72*	3.58	**2.49 / 16.88**	4.44 (**2.72**)	$\mathbf{4.5 \times 10^5}$
5.32*	5.18	**2.07 / 13.74**	6.29 (**3.91**)	$\mathbf{3.5 \times 10^5}$

derived from our CPO algorithm are radically less than that of UR. Notice that the reduction in FLOPs of the models derived by CPO are not as much as the reduction in parameters. The explanation is that owing to the low PS values of the last few layers in VGG-19, as Fig 5.3 shows, we prefer to remove the filters at those layers. However, the X_i, Y_i of those feature maps are small because we've gone through many pooling layers. Therefore, the overall reduction in FLOPs will not be as much as that in parameters. Next, for the testing drop in the CPO part, the original performance after retraining for 8 epochs in the final iteration is a little less than the results of UR. Our conjecture is that the models suffer much more parameter reduction after CPO than after UR; therefore, we need more retraining epochs for

doi:10.6342/NTU202202005

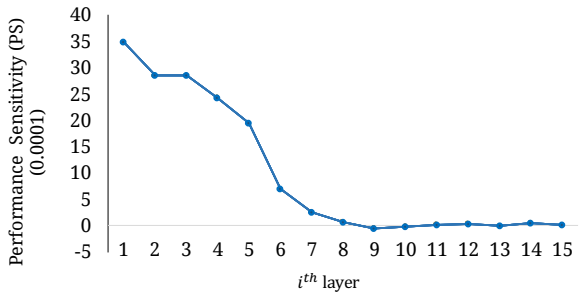
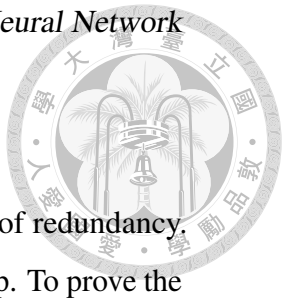


Figure 5.3: **VGG-19 Performance Sensitivity List.**

the last iteration in CPO to recover the performance. Hence, the number in the parentheses on the right side is the drop after retraining for 40 epochs. It shows that the model can recover to the comparable performance as expected. We also conduct two CPO experiments to observe the redundancy when given negligible drops. We find that almost 70% parameters can be removed in the redundant VGG-19 network.

Tabel 5.4 also shows the performance of training a shallow model. The model is constructed with 5 convolutional layers, where each layer contains 64, 128, 256, 512 and 512 filters. Cifar-10 is less complicated in comparison with other large image classification dataset, for which VGG-19 is originally designed. Therefore, the shallow model with only 5 convolutional layers can even achieve 89.56% testing accuracy. This time we choose the CPO results with comparable performance (90.26%) to compare the remaining computation. We can claim that our CPO is able to detect a great amount of redundancy in VGG-19 and remove almost 97% of parameters. However, same as the VDSR experiment stated above, we need to spend more training time to achieve this performance. In details, we use one Geforce 1080Ti GPU to train the Cifar-10 dataset, and it only takes less than fifteen seconds to train for an epoch. Therefore, in practice, our CPO method takes reasonable training time for some small dataset.

doi:10.6342/NTU202202005



5.4.4 ResNet-32 on Cifar-10 Image Classification

It has been a well-known fact that VGG-19 contains great amount of redundancy. Therefore, it can be easily pruned without apparent performance drop. To prove the effectiveness of our CPO, we modify the original architecture of ResNet-34 [74] and design a dedicated ResNet-32 to train on Cifar-10. ResNet is recently one of the most powerful networks, and meanwhile it contains less parameters than the VGG network. Our ResNet-32 possesses 31 convolutional layers and 1 FC layer to map to 10 classes. Among the convolutional layers, except for the first layer, every 2 layers constitute a residual block. There are three stages presented in the model, and each contains 10 convolutional layers. The sizes of output feature maps at the end of each stage are 32×32, 16×16, and 8×8. We utilize 1×1 convolutions to deal with the conflict when the input and output of a residual shortcut have different channel dimensions, which is also the design adopted by the original ResNet model. In details, our ResNet-32 only contains approximately 9% parameters compared to the above VGG-19 network.

It is worth noticing that in our experiments, not every convolutional layer is free to be pruned. On one hand, for a residual block with input and output of identical dimensions, the shortcut is an identity mapping. Hence we do not prune the second layer of the residual block with an eye to maintaining the number of channels of the output feature map, which will be added with the identity mapping. On the other hand, same as the method proposed in [8], when a residual block has input and output of different dimensions, we remove filters in the 1×1 shortcut convolution

Table 5.4: **Comparison of Shallow Model and Pruned Deep Model (VGG-19).** This table also shows the trade-off between shallow and pruned deep model.

	Params / FLOPs Remained (%)	Testing Acc (%)	Training time (epochs)
Original	100 / 100	92.98	300
Shallow	19.51 / 17.17	89.56	**200**
CPO(3.72)	**2.49 / 16.88**	90.26	620

doi:10.6342/NTU202202005

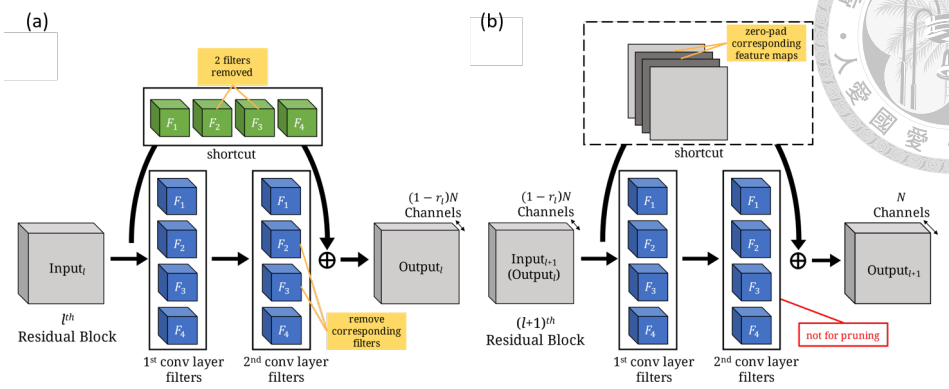


Figure 5.4: **(a)Shortcut Convolution Pruning.(b)Shortcut Identity Padding.** For the residual block with convolution shortcut (a), we first remove the filters in the shortcut and then remove the corresponding filters. For the identity mapping after the shortcut convolution (b), we will do zero-padding to make the dimension of feature maps consistent.

with a given reducing factor, and then remove the filters in the second layer of the residual block according to the indices removed in the shortcut convolution. Fig. 5.4 (a) demonstrates this pruning process, our system will first determine the filter be removed in the shortcut convolution, like the second and third filters, and subsequently remove the corresponding filters in the second convolution layer to avoid dimension conflict. Additionally, when filters are removed in a residual block with convolution shortcut, the identity mapping of the next residual block will cause a conflict. Fig. 5.4 (b) shows the situation. The channel of the input feature map is reduced owing to the filter removal of the preceding layer. Because we will not prune the second convolution layer as mentioned above, there will be a conflict on the identity mapping. Therefore, we will zero-pad the feature map to match the dimension when performing identity addition. Considering the situations about dimension consistency above, the first layer and the FC layer are intact as well. Therefore, there is a limitation on the maximum portion of parameters that can be pruned. It is worth noting that the methods for pruning on skip-connection layers in ResNet has been addressed and also been improved recently [26]. Thus, in Chapter 6, we will use the advanced methods in our filter pruning.

The experimental results can be found in Table. 5.5. All the settings are the same

doi:10.6342/NTU202202005

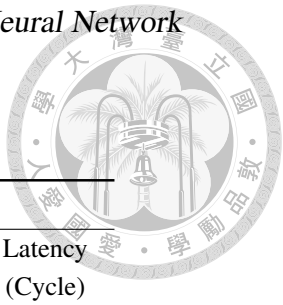


Table 5.5: **Experimental results of ResNet-32.**

Original Model				
Reducing Factor	Val Acc (%)	Params / FLOPs	Testing Acc (%)	Latency (Cycle)
0	94.66	1.9×10^6/ 2.8×10^8	94.58	1.8×10^6
Uniform Removal (UR) [29]				
Reducing Factor	Val Drop (%)	Params / FLOPs Remained (%)	Testing Drop (%)	Latency (Cycle)
0.125	0.64	85.34 / 76.60	0.86	1.5×10^6
0.250	1.44	71.30 / 56.31	1.25	9.6×10^5
0.375	1.88	57.89 / 39.14	1.89	7.7×10^5
0.500	2.84	45.09 / 25.08	3.52	5.3×10^5
CPO (Ours)				
D_{exp} (%)	Final Val Drop (%)	Params / FLOPs Remained (%)	Testing Drop (%)	Latency (Cycle)
0.25	0.32	74.92 / 81.88	0.61 (0.30)	1.5×10^6
0.64*	0.48	**60.27 / 60.88**	0.89 (**0.56**)	$\mathbf{1.1 \times 10^6}$
1.44*	1.28	**38.11 / 42.63**	2.16 (1.47)	$\mathbf{8.2 \times 10^5}$
1.88*	1.68	**30.62 / 36.81**	2.61 (**1.84**)	$\mathbf{7.1 \times 10^5}$
2.84*	2.78	**23.98** / 26.31	3.54 (**3.06**)	5.4×10^5

as VGG-19, except that the testing drop in the parentheses are the drop retraining for 100 epochs after the final CPO iteration. We speculate that ResNet-32 model involves more complicated operations (shortcut convolution and zero-padding); therefore, it needs more epochs to recover the performance. Still, we can see that all the computation reduction done by CPO performs better than that by UR, except one point where the remaining parameters are less than UR but FLOPs is a little bit more. This is also caused by removing most of the parameters at the last convolution stage. In summary, given a well-performed ResNet model (Test Acc:94.58%) with less parameters, although we can not remove as much parameters as we did for VGG-19 network, we still can eliminate 30% of parameters with negligible drop (0.30%) and remove almost 80% of weights with an accuracy that is still higher than 90%.

The comparisons of the performance with shallow networks in Sec. 5.4.2, 5.4.3

doi:10.6342/NTU202202005

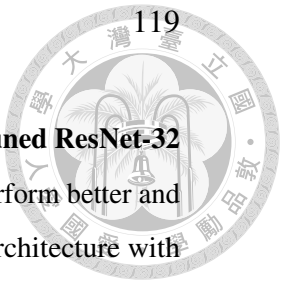


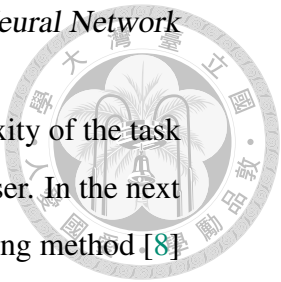
Table 5.6: **Comparison of fine-tuning and random initialization of pruned ResNet-32 on Cifar-10.** This table shows that fine-tuning the model after CPO can perform better and more effective than training the model from scratch of the same model architecture with random weight initialization.

	Extra epochs	Val Acc (%)	Test Acc (%)
CPO (1.44)	100	93.76	93.11
Random Initialization	100	92.14	91.60
	260	93.70	93.20

show that a deep but "thin" network, which means the number of filters in each layer is less than those in the original model, perform better than a shallow network with comparable parameters. We now conduct another experiment on ResNet-32 to discuss the importance of first conducting CPO on the original deep but "fat" model and then fine-tune on it. Without CPO, one can exhaustively try different light-weight architectures, randomly initialize the weight and train the model from scratch to meet the expected performance metrics. However, it is not efficient for optimization and the performance may not be as expected. Table. 5.6 shows the results. The first row is the performance of our CPO with $D_{exp} = 1.44$. As the experiments conducted above, the pruned architecture found by CPO will be fine-tuned for extra 100 epochs to recover the performance. 93.76% and 93.11% are the validation and testing accuracy after fine-tuning. The next two rows are the experiments performed on the same architecture of that in **CPO (1.44)** but with random initialization on the weights. It shows that if we train from scratch with the same number of epochs, the performance is still far from the result after CPO. Not until the 260-th epochs does it reach the comparable performance to ours. Therefore, initializing the model with the original weights helps us speed up the training process and at the same time retain excellent performance.

For image classification, we use Cifar-10 dataset to quickly demonstrate the efficacy of our proposed CPO method. And after conducting experiments on the self-designed networks, we can claim that no matter what kind of architectures,

doi:10.6342/NTU202202005



our CPO can alter the structure effectively according to the complexity of the task and the expected acceptable performance drop D_{exp} given by the user. In the next section, we will compare our CPO to the state-of-the-art filter pruning method [8] at that time with two standard models proposed in their experiments, which are a VGG-16 model that is also trained on Cifar-10 dataset and a ResNet-34 model trained on Imagenet [152] dataset.

5.4.5 Comparison with Current Filter Pruning Method

Recently, the outstanding work [8] utilized similar concept of filter-pruning and achieved impressive compression rate. They determine which filters to be pruned within a single layer L_i by calculating the *L1-norm* of each filter $F_{i,j}$ in that layer, i.e. the *L1-norm* of filter j in i-th layer is $s_j = \sum |F_{i,j}|$. And they remove those with smaller s_j first. In addition, they decide the number of filters to be pruned for each layer based on observations and empiricism. They independently prune each layer by different numbers of filters and respectively inspect their performance on the validation set. According to the layer's resilience to filter pruning, they empirically assign the suitable number of filters to be pruned for each stage of convolutional layers, where the convolutional layers in the same stage have the same size of feature maps. Therefore, layers in the same stage will have the same number of filters left afterwards. We choose two networks which are commonly used and also in their experiments, VGG-16 and ResNet-34, to demonstrate our CPO method and show the performance and computation comparisons.

Table 5.7 shows the performance of the original and pruned models. The upper part is the results in [8], and the lower part is ours. In order to have the same representation in their experiment, we use "**Error**" to represent miss classification rate (MCR). Note that their unpruned VGG-16 model is unreleased, so we train a model from scratch with exactly identical architecture, which results in a slight difference of the performance error. In addition, there is no expected performance drop (D_{exp}) in their method; therefore, we perform our CPO method with D_{exp} set

doi:10.6342/NTU202202005

Table 5.7: **VGG-16 Comparison between CPO ($D_{exp} = 0.1$) and [8].** Both experiments use VGG-16 trained on Cifar-10 as the targeted model and retrain 40 epochs afterwards. CPO achieves more reduction in parameters and FLOPs and meanwhile maintains the performance.

Model	Error	FLOPs Remained	Params Remained
VGG-16 [8]	6.75%	3.13×10^8 (100%)	1.5×10^7 (100%)
Pruned [8]	6.60%	2.06×10^8 (65.8%)	5.4×10^6 (36.0%)
VGG-16 (ours)	6.48%	3.13×10^8 (100%)	1.5×10^7 (100%)
Pruned (CPO)	6.66%	$\mathbf{1.97 \times 10^8}$ **(53.0%)**	$\mathbf{3.3 \times 10^6}$ **(22.1%)**

as 0.1%. The result shows that CPO achieves more reduction in parameters and FLOP. Moreover, the error performance of the CPO-pruned model (6.66%) is also comparable to theirs (6.60%).

Next, the ResNet-34 network trained on famous Imagenet dataset is the other pruned target. Imagenet contains one million training images with one thousand label classes. Therefore, it is considered to be hard to discover the redundancy within the model that is trained on this large dataset. We perform our proposed CPO with the pruning method elaborated in Sec 5.4.4. Because of the great amount of training time required for one training epoch, we only conduct three epochs for every retraining stage and for every epoch we only randomly sample $1/3$ images for training. Table 5.8 shows the experimental results of the two pruning methods. Notice that the performance of the two unpruned models may not be identical because of the different released sources, and our pretrained ResNet-34 model is obtained from Pytorch [171]. Obviously, the computation reduction of the two methods are not as much as that in VGG-16. By performing our CPO with $D_{exp} = 1.3\%$, we can discover 14.4% of redundancy for parameters and obtain less than 1% drop of the performance, which is better than the other. Nonetheless, when it comes to FLOPs, our method only removes 15% redundancy. The reason is same that it tends to eliminate the filters at the last few layers because removing the filters in those layers harm the performance less. Compared with [8], they

doi:10.6342/NTU202202005

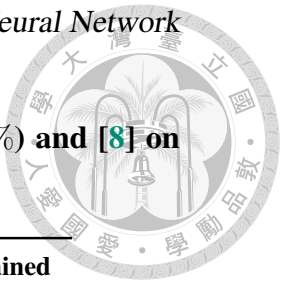


Table 5.8: **ResNet-34 Comparison between CPO ($D_{exp} = 1.3\%$) and [8] on Imagenet.**

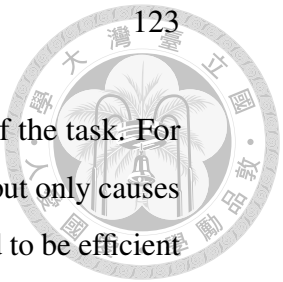
Model	Error	FLOPs Remained	Params Remained
ResNet-34 [8]	26.77%	3.64×10^9 (100%)	2.16×10^7 (100%)
Pruned [8]	27.83%	$\mathbf{2.76 \times 10^9}$ **(75.8%)**	1.93×10^7 (89.2%)
ResNet-34 (ours)	26.68%	3.64×10^9 (100%)	2.16×10^7 (100%)
Pruned (CPO)	**27.66%**	3.11×10^9 (85.4%)	$\mathbf{1.85 \times 10^7}$ **(85.6%)**

empirically decide the number of filters to be removed in every convolutional stage according to the layer sensitivity. Therefore, by removing the filters in the first few layers, the FLOPs can be pruned more (75.8% remained) but at the same time, the classification performance can not be controlled. Conducting the CPO to ResNet-34 model on Imagenet dataset takes almost a week with two Geforce 1080Ti GPUs. Therefore, we are not able to conduct many experiments to progressively increase the expected drop to improve the performance. We believe that if we progressively increase the D_{exp} to a higher number and conduct the retraining stage for more than three epochs, like the 8 epochs in previous experiments but at the same time increasing the training time for the corresponding proportion, or conduct the experiments with paralleled and high bandwidth GPU devices in order to reduce the training time, we can obtain a model with less computation by our proposed CPO. In addition, for this kind of model trained on large dataset, besides conducting filter pruning to specifically alter the model architecture, we can still combine other methods like quantization to help deploy the model on edge devices.

5.5 Summary

In this work, we present a Computation-Performance Optimization (CPO) method by removing filters in convolutional layers of a neural network. It utilizes Sparsity, Reducing Factor, and Performance Sensitivity to determine which and how many filters to prune in one layer. With an expected drop given by the user, CPO can

doi:10.6342/NTU202202005



effectively alter the model structure according to the complexity of the task. For super-resolution, it reduces more than 50% of parameters in VDSR but only causes about 0.28dB in performance drop. Furthermore, CPO is also proved to be efficient when applied to image classification. We conduct VGG-19 and ResNet-32 (on Cifar-10) to demonstrate that it can eliminate great amount of parameters and FLOPs without significant drop in accuracy. Compared with previous works, CPO provides a solution to determining the layer-wise hyperparameters of filter pruning, and achieves superior results.

doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

Chapter 6

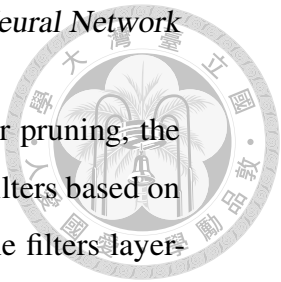
Global Filter Pruning for Neural Network

In recent years, global filter pruning has been more popular than layer-wise pruning. In this chapter, we will introduce and demonstrate our contributions in global filter pruning.

6.1 Introduction

As mentioned before, the computaion demanding CNNs are hard to be deployed on embedded devices. Thus, how to optimize and accelerate the heavy network but maintain the performance at the same time would be a critical issue to solve. Network pruning is a common solution for optimizing the model. Given a large and deep neural network, the goal of pruning is trying to obtain an optimal sub-network with acceptable performance drop, or even sometimes resulting in a small gain, by exploring and removing the redundant parts of the model. Based on the categories of a unit being pruned at a time, filter pruning [8], also known as channel pruning [22], is one promising technique which effectively reduces the computation cost by regarding a structural portion, such as a convolutional filter or a channel in output feature maps, of the model as a unit when performing network

doi:10.6342/NTU202202005

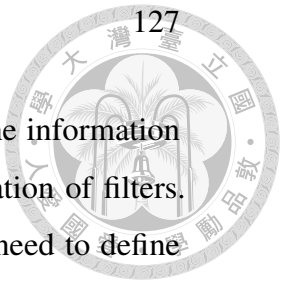


pruning. Furthermore, among all the relevant approaches of filter pruning, the global method [24, 25, 26, 172] which determines the redundant filters based on the whole network is usually more popular than those pruning the filters layer-by-layer [21, 8, 23] because globally removing the redundancy is more flexible and time-efficient. In detail, the procedure of the global method can be described as follow: the pruning algorithm first repeatedly removed unimportant filters in a given trained CNN until the pruned network satisfied given pruning objectives. Then, the fine-tuning training process will be conducted to retain performance. As for the pruning objective, commonly, it would be the number (or ratio) of filters left, or other computation resource constraints such as floating-point operations (FLOPs), number of total parameters, inference latency, and so on.

The core component of global filter pruning is the process to determine the "importance" of each filter and thus iteratively remove the least important one in the whole network. The correctness of the importance estimation will explicitly affect the final performance results. In the past, the L1-norm [8] or sparsity [29] of a filter is used, while recently, most works [24, 26, 172] achieve outstanding results by estimating the importance of a filter based on its impact on the loss when being removed. However, we found that the "pruning objective" at hand is not taken into consideration during importance estimation in those works, which reduces the correlation between the estimated importance of filters and the optimal solution for pruning the network under the specific objective. Molchanov *et al.* [24] has proposed a method to consider the given pruning objective during importance estimation, but the integration method cannot be proved to achieve an optimal result. In addition, when the objective contains multiple resource constraints, which is practical in the real-world scenario, previous methods can only keep pruning the model until it separately matches all constraints. Because they cannot jointly consider all objectives, the pruning results will not meet all constraints accurately at the same time and consequently lead to worse performance.

To solve the problems mentioned above, we propose a novel method called

doi:10.6342/NTU202202005

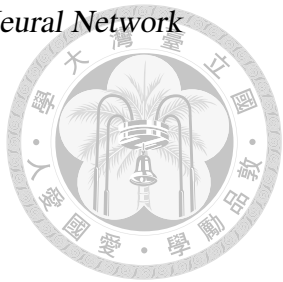


Constraint-Aware Importance Estimation (CAIE) to integrate the information of given resource constraints into the original importance estimation of filters. Given any single resource constraint as in other works, we first need to define the "resource impact" of a filter, which is the normalized amount of resource reduction in the whole network when the specific filter is removed. Then, by mathematical derivation, we can combine the original impact based on the loss function and the impact based on the specific constraint to estimate the new constraint-based importance of a filter. Additionally, when we encounter multiple resource constraints, our integration method can be easily *generalized* to the formulation with multi-constraints. With our CAIE, in each pruning iteration, we can remove the filters that will make the pruned sub-network most close to the objective but with the least impact on the loss function, which is, therefore, the optimal solution compared to others. Moreover, when applying our generalized estimation method under multi-constraints, we can achieve the best performance over state-of-the-arts and simultaneously meet all the given constraints accurately.

We now highlight the contributions of this work:

- We propose a novel method called Constraint-Aware Importance Estimation (CAIE) to estimate the importance of filters in combination with the given resource constraint, which can obtain the best results compared to those only based on the loss function.
- The proposed method can be easily generalized into the pruning problem under multiple constraints, which is practical to real-world scenarios.
- Under the same amount of resource consumption of the pruned model, we can achieve the state-of-the-art performance results with our proposed CAIE method.

doi:10.6342/NTU202202005



6.2 Related Work

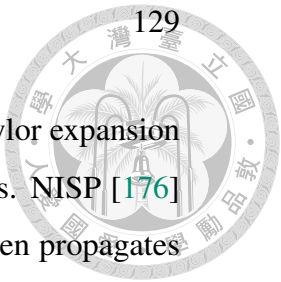
6.2.1 Filter Pruning

Network pruning is a common method to obtain a compact network from a large one by removing the redundant parts. Comparing to the traditional weight pruning [10], which only removes the redundant parameter individually, in CNNs, filter pruning can effectively reduce the computation consumption by treating a convolutional filter in the model as the unit for pruning. To determine the redundant filters, some works focus on evaluating the importance of filters in a single layer and remove unimportant ones "layer-by-layer". In contrast, others are interested in the global method that evaluate and prune filters based on the whole network.

Layer-wise filter pruning Among the methods pruning layer-by-layer, some works [8, 173, 29, 174] believe that there is a strong correlation between the importance of a filter and its corresponded parameter-dependent values, such as its L1-norm [8], L2-norm [173], sparsity [29] or the distance to the geometric median of filters in a single layer [174]. On the other hand, some works introduce using training data to yield the criterion for filter removal [21, 22, 23]. Thinet [21] and CP [22] select filters that can minimize the feature reconstruction error layer-by-layer by solving LASSO problem [175]. Moreover, DCP [23] adopts additional discrimination-aware losses to guide the selection of redundant filters and enhance the discrimination ability of features. However, all of the methods mentioned above can only compare filters in the same layer; in other words, the cross-layer comparison is not available. Furthermore, layer-wise filter pruning is time-consuming and requires a pre-defined pruning ratio for each layer, which may reduce the flexibility of the left network and cause a worse performance result. Therefore, we focus on solving the filter pruning problem with the global method.

Global filter pruning To make the estimated values for filter importance globally comparable, Molchanov *et al.* [24] utilizes a layer-wise normalization technique

doi:10.6342/NTU202202005

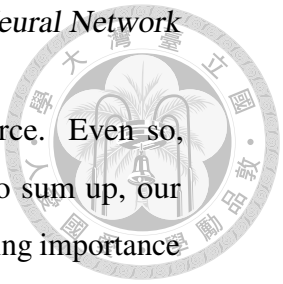


to rescale the original importance score, which is generated by Taylor expansion of the impact on the loss function caused by the removal of filters. NISP [176] measures the importance of features in the final response layer then propagates the importance score for each filter in the whole network from the final response layer to the first layer. Some works [25, 177] try to take advantage of batch-normalization (BN) layers [75]. They enforce the sparsity of the scaling factor γ in the BN layer by adding a regularization term in training and prune filters depending on a global threshold over the value of γ. Recently, to assess the importance of a filter more accurately, Molchanov *et al.* [26] modifies the Taylor expansion method in [24] so that the additional normalization is not required. Combining the Taylor expansion method and the sparsity enforcement technique, GBN [172] introduces "Gate Decorator" and applies it on BN layers for importance estimation and sparsity training. Although they can obtain a promising performance, except the work [24], none of these methods consider the given constraint during importance estimation of filters.

6.2.2 Constraint-based Network Optimization

Some works try to integrate the information of given constraints when optimizing the network. Molchanov *et al.* [24] add a regularization term about the given constraint to the original score of the filter importance. However, this method introduces an extra parameter λ to control the amount of regularization, which is selected empirically and sensitive to the magnitude of the regularization term and the original importance score. LCP [178] adopts an evolutionary algorithm to offset the importance of a filter with the given constraints, which is originally evaluated by the impact on the loss. Though LCP also considers the given constraints when searching the offset value, their belief that the optimal importance estimation of filters is based on the impact on loss is incorrect since it should also be related to the given constraint. Morphnet [179], a work about network architecture search, introduces the resource-weighted regularizer in the loss function to search the

doi:10.6342/NTU202202005



proper width of each layer that is optimal to the targeted resource. Even so, Morphnet can only concern about a single resource at a time. To sum up, our method can easily merge the information of the given constraints during importance estimation in global filter pruning.

6.3 Constraint-Aware Importance Estimation

To clearly unify and compare to the works with filter pruning, we will first introduce the preliminaries and define the optimization problem of original filter pruning in the global method. Then, in order to solve the problem under a single constraint, we re-formulate the global filter pruning problem and solve it with our proposed Constraint-Aware Importance Estimation (CAIE), which is then generalized to deal with problems under multiple constraints. At last, we will summarize our iterative pruning and fine-tuning scheme with CAIE in.

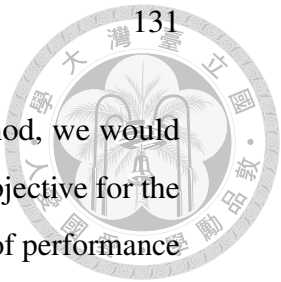
6.3.1 Preliminaries of Filter Pruning

Given a network with parameters θ and a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ with N training data and label pairs (x_i, y_i), the goal of network training is to minimize the given loss function $\mathcal{L}(\mathcal{D}; \theta)$. In filter pruning, we first define the set of removable filters in the network with parameters θ as $\mathcal{F}(\theta)$. Then, the pruning algorithm will aim to yield a smaller model with left parameters $\theta_{F'}$ that can also minimize the loss function $\mathcal{L}(\mathcal{D}; \theta_{F'})$ by removing a subset of filters $F \subset \mathcal{F}(\theta)$ in the network under a specific constraint, which can be formulated as follow:

$$\underset{F}{\operatorname{argmin}} \mathcal{L}(\mathcal{D}; \theta_{F'}) \quad s.t.\ \mathcal{C}(\theta_{F'}) \leq C \tag{6.1}$$

where $F \cup F' = \mathcal{F}(\theta)$, C is the given pruning constraint and $\mathcal{C}(\theta)$ is the amount of concerned resource consumption for the network with parameters θ. In a typical filter pruning problem, the common pruning constraint C is the maximum expected number of filters left, and the corresponded measurement $\mathcal{C}(\cdot)$ would be the total number of filters $|\mathcal{F}(\cdot)|$.

doi:10.6342/NTU202202005



Specifically, when under the setting of pruning in global method, we would first obtain a well-trained network with parameters θ^*. Thus, the objective for the optimization problem can be change into minimizing the difference of performance caused by removing the filters in the network whose parameters are initialized with $\theta = \theta^*$. This difference is commonly evaluated by calculating the *loss impact* [24, 26, 172], defined as $\ell(F)$, when removing the filter set F for θ. The $\ell(F)$ can be formulated as:

$$\ell(F) = \mathcal{M}(\mathcal{L}(\mathcal{D}; \theta),\ \mathcal{L}(\mathcal{D}; \theta_{F'})), \tag{6.2}$$

where $\mathcal{M}(\cdot)$ is a distance metric function such as squared difference [26] or absolute difference [24, 172]. Therefore, with $\ell(F)$, the optimization problem is re-formulated as:

$$\underset{F}{\operatorname{argmin}}\ \ell(F) \quad s.t.\ \mathcal{C}(\theta_{F'}) \leq C. \tag{6.3}$$

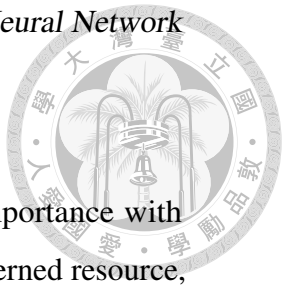
This problem would then be solved with greedy strategy: iteratively estimating the importance of each filter f, $\mathcal{I}(f)$, in the network left and pruning those least important ones that can minimize the loss impact until the given constraint is satisfied. Commonly, the importance of a filter $\mathcal{I}(f)$ is assigned as its loss impact in this greedy process:

$$\mathcal{I}(f) = \ell(f)\ . \tag{6.4}$$

During implementation, instead of evaluating $\mathcal{I}(f)$ for all filters in $\mathcal{F}(\theta)$ with $|\mathcal{F}(\theta)|$ pruned models in total, which is time-consuming, $\mathcal{I}(f)$ is usually estimated by first-order Taylor approximation [26], where all required gradients can be obtained by back-propagation at once.

The effectiveness of the criterion "selecting the least important filter" is based on the assumption that the impact of a single filter in the removed filter set can be considered individually, which is only valid when a small number of filters are removed. This is why in the solution to problem (6.3), they iteratively remove part of the most unimportant filters and then re-settle the problem with the network left as a "new initialization".

doi:10.6342/NTU202202005



6.3.2 Single-constraint Importance Estimation

Among previous works, we found that the estimation of filter importance with (6.4) is a lack of information about the given constraint or the concerned resource, which decreases the credibility of acquiring the best pruning result. Hence, we propose a **Constraint-Aware Importance Estimation** (CAIE) method, which aims to simultaneously combine the information of constraint and performance during importance estimation for a single filter. To better derive our solution, we will first re-formulate the original pruning problem (6.3) into the summation of the individual contribution to performance change and resource reduction for a single filter without losing authenticity.

First of all, those common choices of metric function $\mathcal{M}(\cdot)$ ensure the linearity of loss impact, as a result, problem (6.3) can be rewritten as:

$$\underset{F}{\operatorname{argmin}} \sum_{f \in F} \ell(f) \quad s.t.\ \mathcal{C}(\theta_{F'}) \leq C. \tag{6.5}$$

In practical usage, the given constraint C could be **the maximum value of a certain type of computation resource**, such as FLOPs. To better solve the pruning problem under such scenarios, we introduce the *resource impact*, $r(F)$, of a filter set F, which is the proportion of the reduction in resource consumption while pruning F:

$$r(F) = \frac{\mathcal{C}(\theta) - \mathcal{C}(\theta_{F'})}{\mathcal{C}(\theta)} \ . \tag{6.6}$$

Therefore, we can rewrite (6.5) with our defined resource impact:

$$\underset{F}{\operatorname{argmin}} \sum_{f \in F} \ell(f) \quad s.t.\ r(F) \geq R \ , \tag{6.7}$$

where the pruning objective $R = \frac{\mathcal{C}(\theta) - C}{\mathcal{C}(\theta)}$ is **the minimum proportion of total reduction given a constraint** C.

Last, since we will resolve problem (6.7) iterativly during the process of pruning, we can apply an useful assumption when a small number of filters are removed at a time: "the resource impact of a filter set F is equal to the sum of

doi:10.6342/NTU202202005

resource impact of individual filter f in the set F". Accordingly, we formulate the optimization problem of **single-constraint pruning** as:

$$\underset{F}{\operatorname{argmin}} \sum_{f \in F} \ell(f) \quad s.t. \ \sum_{f \in F} r(f) \geq R \, . \tag{6.8}$$

In particular, as in previous works, applying the constraint C with the number (or ratio) of filters left is just a special case of (6.8) with $r(f) = \frac{1}{|\mathcal{F}(\theta)|}$:

$$\underset{F}{\operatorname{argmin}} \sum_{f \in F} \ell(f) \ \ s.t. \ \sum_{f \in F} r(f) = \frac{|F|}{|\mathcal{F}(\theta)|} \geq F_0, \tag{6.9}$$

where $F_0 = \frac{|\mathcal{F}(\theta)| - C}{|\mathcal{F}(\theta)|}$ is the minimum ratio of filters to be removed to the total number of filters.

Now, given this **new optimization problem** (6.8), we want to find the optimal solution through ranking filters by estimating suitable importance score function g, $\mathcal{I}(f) = g(\ell(f), r(f))$, which contains information about the given constraint. Intuitively, $\mathcal{I}(f)$ should possess the following characteristics:

1. If two filters f_1, f_2 have the same value of resource impact, $r(f_1) = r(f_2)$, the importance should be dominated by the corresponding loss impact.
2. If two filters have the same loss impact value, $\ell(f_1) = \ell(f_2)$, the one with larger resource impact should have higher priority of being pruned since removing the filter with a higher reduction in resource consumption is more beneficial to the progress of pruning.

To possess the property mentioned above, we propose the **Constraint-Aware Importance Estimation** (CAIE) method under a single constraint, which is a feasible form of importance $\mathcal{I}(f)$ to solve problem (6.8):

$$\mathcal{I}_{sing}(f) = \frac{\ell(f)}{r(f)} \, . \tag{6.10}$$

We first qualitatively illustrate the effectiveness of our CAIE in Fig. 6.1, where the path colored in red is with our method, and the two axes represent the total number of loss impact and resource impact. Because our importance estimation $\mathcal{I}_{sing}(f)$

doi:10.6342/NTU202202005

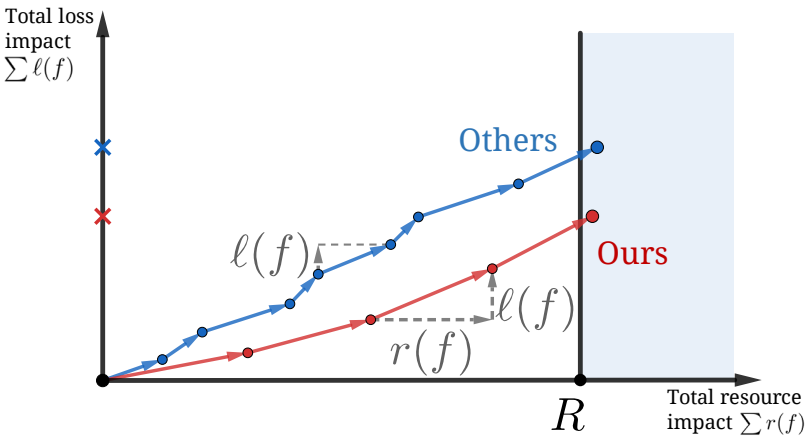


Figure 6.1: **Comparison of our CAIE to others under the single-constraint pruning problem.** The paths colored in red and blue denote the pruning process generated by our method and others, respectively. Each colored vector between two points illustrates the loss impact $\ell(f)$ (vertical component) and the resource impact $r(f)$ (horizontal component) when removing the filter f. Our method, which considers the two components jointly, can generate a better result with a lower total amount of loss impact under the same pruning objective R.

represents the performance drop per unit of resource reduction, comparing to the original $\mathcal{I}(f)$ that only based on $\ell(f)$, greedily selecting those filters with lower $\mathcal{I}_{sing}(f)$ would lead to the smallest performance drop when reaching the needed total amount of resource reduction R. To confirm the correctness of our CAIE under single constraint (6.10), we give a more rigorous proof as follow:

Proof. $F_{\mathcal{I}}$ and F^* indicate our solution and the optimal solution respectively. In general, resource impact of a single filter $r(f)$ is far less than the pruning objective R, which implies that we can neglect the difference between total resource reduction $\sum_f r(f)$ and pruning objective R in both solutions, hence, $\sum_{f\in F_{\mathcal{I}}} r(f) = \sum_{f\in F^*} r(f)$.

Let $S_0 := F_{\mathcal{I}} \cap F^*$ and suppose that $F_{\mathcal{I}} \neq F^*$, we have $S_1 := F_{\mathcal{I}} \setminus (S_0) \neq \emptyset$

doi:10.6342/NTU202202005

and $S_2 := F^* \setminus (S_0) \neq \emptyset$. Then,

$$\begin{aligned}\sum_{f\in S_1} r(f) &= \sum_{f\in F_{\mathcal{I}}} r(f) - \sum_{f\in S_0} r(f) \\ &= \sum_{f\in F^*} r(f) - \sum_{f\in S_0} r(f) = \sum_{f\in S_2} r(f)\,.\end{aligned} \tag{6.11}$$

Based on the facts that $F_{\mathcal{I}} \cap S_2 = \emptyset$, $S_1 \subset F_{\mathcal{I}}$ and the criterion "**selecting least importance filter**" in the pruning algorithm, we have

$$\max_{f\in S_1} \mathcal{I}(f) \leq \max_{f\in F_{\mathcal{I}}} \mathcal{I}(f) \leq \min_{f\in S_2} \mathcal{I}(f)\,. \tag{6.12}$$

Therefore,

$$\begin{aligned}\sum_{f\in S_1} \ell(f) &= \sum_{f\in S_1} \frac{\ell(f)}{r(f)} r(f) = \sum_{f\in S_1} \mathcal{I}(f) r(f) \\ &\leq \max_{f\in S_1} \mathcal{I}(f) \cdot \sum_{f\in S_1} r(f) \\ &\leq \min_{f\in S_2} \mathcal{I}(f) \cdot \sum_{f\in S_2} r(f) \\ &\leq \sum_{f\in S_2} \mathcal{I}(f) r(f) \\ &= \sum_{f\in S_2} \frac{\ell(f)}{r(f)} r(f) = \sum_{f\in S_2} \ell(f)\,,\end{aligned} \tag{6.13}$$

and

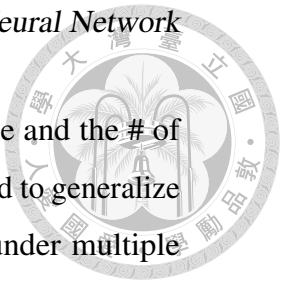
$$\begin{aligned}\sum_{f\in F_{\mathcal{I}}} \ell(f) &= \sum_{f\in S_0} \ell(f) + \sum_{f\in S_1} \ell(f) \\ &\leq \sum_{f\in S_0} \ell(f) + \sum_{f\in S_2} \ell(f) = \sum_{f\in F^*} \ell(f)\,.\end{aligned} \tag{6.14}$$

We can see that with (6.14), the total loss impact of our solution $F_{\mathcal{I}}$ will always be equal to or less than that of the optimal solution F^*. In other words, our solution $F_{\mathcal{I}}$ is thus an optimal solution to the problem of single-constraint pruning (6.8). □

6.3.3 Multiple-constraint Importance Estimation

In practical scenarios, given any desired platform, we may need to jointly consider some pruning constraints of different resources at the same time, such as regarding

doi:10.6342/NTU202202005



the # of parameters left of the model owing to the memory storage and the # of FLOPs based on the platform's computing power. Therefore, we need to generalize the aforementioned single-constraint pruning problem into that under multiple constraints and generalize the solution with our CAIE.

In formulation, when given k constraints $\{C_i\}_{i=1}^{k}$, we can first generalize (6.8) to the problem of **multiple-constraint pruning**:

$$\underset{F}{\operatorname{argmin}} \sum_{f\in F} \ell(f) \quad s.t. \ \sum_{f\in F} r_i(f) \geq R_i, \ \forall i \leq k \ , \tag{6.15}$$

with $r_i(f) = \frac{\mathcal{C}_i(\theta)-\mathcal{C}_i(\theta_{f'})}{\mathcal{C}_i(\theta)}$ and the pruning objective $R_i = \frac{\mathcal{C}_i(\theta)-C_i}{\mathcal{C}_i(\theta)}$ for **each concerned resource** i. Specifically, we can neglect the resource i when $\mathcal{C}_i(\theta) - C_i < 0$, which means its consumption is already lower than the given constraint; thus, the pruning objective of resource i should be modified as $R_i = \max(\frac{\mathcal{C}_i(\theta)-C_i}{\mathcal{C}_i(\theta)}, 0)$.

To better derive the solution, we need to jointly consider all different resource impacts when removing one filter f. We define the joint resource impact and the overall pruning objective as the **vector form**, $\vec{r}(f) = \langle r_1(f), r_2(f), ..., r_k(f)\rangle$ and $\vec{R} = \langle R_1, R_2, ..., R_k\rangle$, in the resource space $\mathbb{R}^k$. With the linearity of vectors, the total resource impact $\sum_{f\in F} r_i(f)$ for all resource i when pruning the filter set F can thus be easily obtained by summation of the resource impact vectors $\vec{r}(f)$:

$$\left\langle \sum_{f\in F} r_1(f), \sum_{f\in F} r_2(f), ..., \sum_{f\in F} r_k(f) \right\rangle = \sum_{f\in F} \vec{r}(f) \ . \tag{6.16}$$

Furthermore, in the space $\mathbb{R}^k$, we found that the direction of $\vec{R}$ is the optimal direction for pruning because the objective point $\boldsymbol{R} = (R_1, R_2, ..., R_k)$ is the closet point on the boundary of the constraints in problem (6.15) to the origin of the resource space. Hence, when finding the optimal solution, we only need to focus on the components of $r(\vec{f})$ with a positive contribution to the direction of $\vec{R}$. Consequently, we define the ***effective resource impact***, $r_e(f)$, as the scalar projection of $\vec{r}(f)$ onto $\vec{R}$:

$$r_e(f) = \vec{r}(f) \cdot \frac{\vec{R}}{|\vec{R}|} = \frac{\sum_i r_i(f) R_i}{\sqrt{\sum_i R_i^2}} \ , \tag{6.17}$$

doi:10.6342/NTU202202005

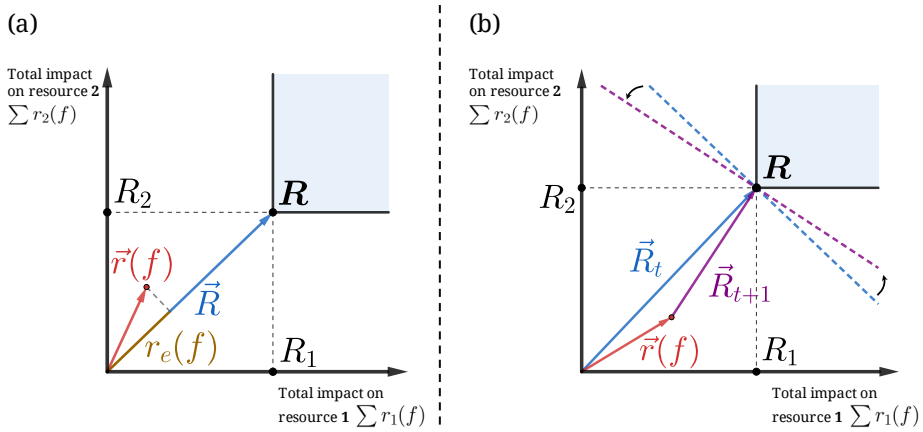


Figure 6.2: **(a) Illustration of the effective resource impact. (b) The modified problem in multiple-constraint pruning.** We take the problem under two resource constraints (R_1, R_2) as an example and demonstrate the resource impact on the resource plane. (a): The effective resource impact $r_e(f)$ is the scalar projection of $\vec{r}(f)$ to the objective vector $\vec{R}$. (b): Dotted lines are the boundaries of the constraints in the modified problem (6.19) with vectors $\vec{R}_t$ and $\vec{R}_{t+1}$ in pruning iteration t and $t+1$. As we remove some filters in the network, we will adjust the objective vector from $\vec{R}_t$ to $\vec{R}_{t+1}$ in order to make the pruning direction still point to the point $\boldsymbol{R}$.

which is also illustrated in Fig. 6.2 (a). With $r_e(f)$, our ***generalized*** **Constraint-Aware Importance Estimation** (CAIE) under multiple constraints $\mathcal{I}_{mul}(f)$ can be defined as the formula similar to that in single-constraint pruning (6.10):

$$\mathcal{I}_{mul}(f) = \frac{\ell(f)}{r_e(f)} \,. \tag{6.18}$$

It's worth noting that in fact, the importance $\mathcal{I}_{mul}(f)$ is the optimal solution to the following modified problem:

$$\operatorname*{argmin}_{F} \sum_{f \in F} \ell(f) \quad s.t. \sum_{f \in F} r_e(f) \geq |\vec{R}| \,, \tag{6.19}$$

which is transformed from the problem under single constraint (6.8) with the substitution in some of the notations. Although the boundary of constraints in original problem (6.15) and that in the modified one (6.19) are distinctive, shown

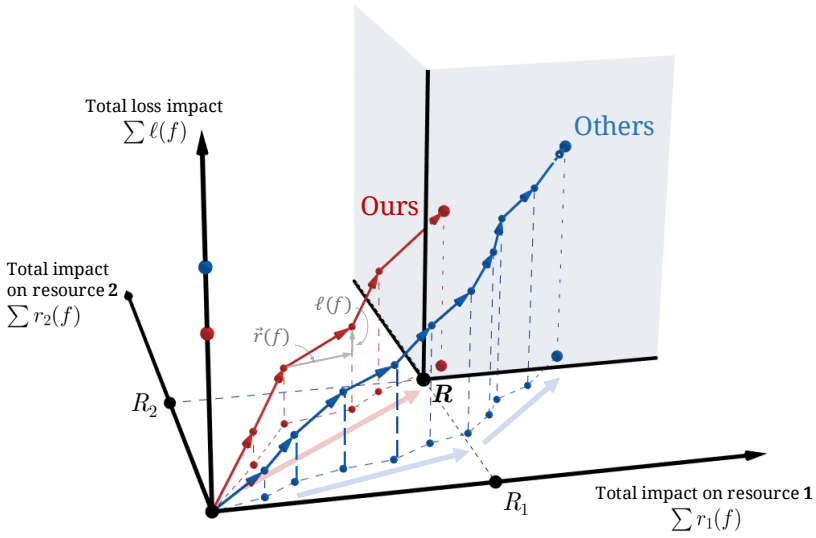
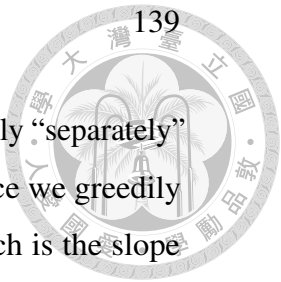


Figure 6.3: **Comparison of our CAIE to others under the multiple-constraint pruning problem.** We demonstrate the pruning problem under two constraints. The paths colored in red and blue denote the pruning process generated by our method and others respectively. Each colored vector between two points is composed by the loss impact $\ell(f)$ and the resource impact vector $\vec{r}(f)$ when removing the filter f. Our method with CAIE is able to reach the pruning objectives simultaneously and also jointly considers all the impacts to generate a better result with lower total loss impact under the objective $\boldsymbol{R}$.

in Fig. 6.2 (b), the point $\boldsymbol{R}$ is also the closet point on the boundary in (6.19) to the origin, which is the same as problem (6.15). Moreover, from iteration t to $t+1$ among the pruning process, we will also adjust the objective vector $\vec{R}$ whenever we remove a small number of filters. Thus, also shown in Fig. 6.2 (b), we can always consider the optimal direction in the process and consequently be able to reach the final pruning objective point $\boldsymbol{R}$ accurately.

Last, same as the signle-constraint pruning, in Fig. 6.3, we demonstrate the effectiveness of our CAIE method under multiple constraints (colored in red) when comparing to others only based on the loss impact (colored in blue), where the x-y plane is the resource space, and the z-axis denotes the total loss impact. With our CAIE, the path which represents the iterative pruning result always moves



toward the pruning objective point $\boldsymbol{R}$ while previous methods can only "separately" match each constraint (i.e., first meet R_1 then R_2). Moreover, since we greedily select the filters with the smallest importance score $\mathcal{I}_{mul}(f)$, which is the slope of performance change on the effective reduction of all the resources, we can acquire the result with the smaller overall impact on loss than others when it meets $\boldsymbol{R}$. To summarize, our CAIE method can generate the optimal result under any combination of resource constraints.

Algorithm 3: Global Filter Pruning with CAIE

Input: Pre-trained network parameters θ^*, dataset $\mathcal{D}$, k pruning constraints $\{C_i\}_{i=1}^{k}$

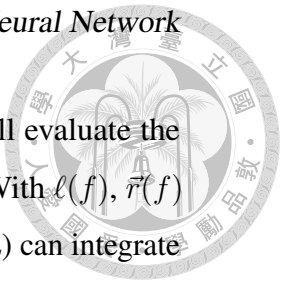
Output: pruned network parameters θ_p^*

1: Set θ^* as the initialization of the concerned network θ
2: **while** θ not satisfy given constraints **do**
3: Estimate the loss impact $\ell(f)$ for each filter f in θ with n mini-batches of data in $\mathcal{D}$
4: Evaluate the pruning objective vector $\vec{R}$ from θ and the resource impact vector $\vec{r}(f)$ for each filter f in θ
5: Calculate the importance score $\mathcal{I}(f)$ with formula (6.17) and (6.18)
6: Remove a filter set F containing m least important filters based on $\mathcal{I}(f)$ and acquire a left network $\theta_{F'}$
7: Set $\theta_{F'}$ as the concerned network θ for next iteration
8: **end while**
9: Fine-tuning θ with $\mathcal{D}$ and yield a pruned model θ_p^*

6.3.4 The Overall Pruning Scheme

Our algorithm of global filter pruning follows the procedure of "iteratively pruning then fine-tuning", illustrated in Algorithm 3. During a single iteration in the pruning stage to remove a small number of filters, we first estimate the loss impact $\ell(f)$ of

doi:10.6342/NTU202202005



each filter f following the method proposed in [26]. Next, we will evaluate the resource impact vector $\vec{r}(f)$ and the pruning objective vector $\vec{R}$. With $\ell(f)$, $\vec{r}(f)$ and $\vec{R}$ at hands, our Constraint-Aware Importance Estimation (CAIE) can integrate those components with formula (6.17) and (6.18) to generate the importance $\mathcal{I}(f)$ of each filter as the criterion for pruning. The iterative pruning process will continue until the pruned network satisfies the given constraints, then followed by the fine-tuning process to further boost the performance at last.

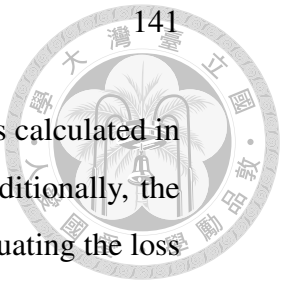
6.4 Experiments

6.4.1 Implementation Details

Dataset Our CAIE is evaluated on the CIFAR-10 [166] and ImageNet ILSVRC-12 [152]. The CIFAR-10 dataset contains 50k training images and 10k test images in 10 classes, while the ImageNet dataset contains 1.28M training images and 50k test images in 1000 classes. We follow the standard process of data augmentation in both datasets. For training data of CIFAR-10, the process contains padding images to 40×40, randomly cropping a 32×32, then normalizing with the mean and standard deviation of the dataset; for testing data of CIFAR-10, we only apply data normalization to the images. For the training set in ImageNet, the process contains re-sizing images to 256×256, randomly cropping a 224×224 patch, randomly flipping horizontally, and normalizing them with mean and variance of ImageNet. For the testing set in ImageNet, we re-size images to 256×256, crop them into a 224×224 patch then apply data normalization.

Loss impacts We choose the method proposed by Molchanov *et al.* [26] to evaluate the loss impact $\ell(f)$ of a filter f. The formula of loss impact is $\ell(f) = (\gamma_c \frac{\partial \mathcal{L}}{\partial \gamma_c} + \beta_c \frac{\partial \mathcal{L}}{\partial \beta_c})^2$, where γ_c and β_c are the scaling and shifting parameters of "the following BatchNorm [75] (BN) layer" in the channel c corresponding to the filter f. The gradients $\frac{\partial \mathcal{L}}{\partial \gamma_c}$ and $\frac{\partial \mathcal{L}}{\partial \beta_c}$ in the formula above would be computed whenever

doi:10.6342/NTU202202005



a mini-batch of training data is given. We average the loss impacts calculated in n mini-batches using a running average with coefficient 0.9. Additionally, the gradients would also be utilized in updating the network when evaluating the loss impacts of filters. If the network such as ResNet [74] or MobileNetV2 [180] contains residual blocks or depthwise convolutional layers that some specific layers should be pruned in the same way, as described in [26], we will sum over the corresponding loss impacts in the same channel of these grouped layers as the overall loss impact.

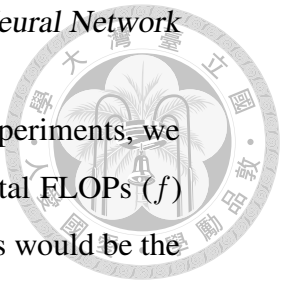
Resource impacts The i^{th} resource impact $r_i(f)$ of the filter f among the same layer share the same value. Thus, we can evaluate the resource impact for layer l by randomly removing a filter in layer l plus the corresponding channel of each filter in the succeeding $(l+1)^{th}$ layer, then measuring the proportion of reduction in the concerned resources. Also, when we encounter residual blocks or depthwise convolutions containing grouped layers, we will randomly remove an output channel in these layers following the rule in the works [181, 172] to evaluate the overall resource impact.

Pruning and fine-tuning In both stages, the batch size is set to be 256 and 64 in CIFAR-10 and ImageNet, respectively. In a single pruning iteration, the number of batches n used for estimating importance and the number of pruned filters m are set to be 30 and 25 in both datasets. For optimizing the neural network, we use SGD with an initial learning rate 10^{-3} and runs for 30 epochs in ImageNet and 240 epochs in CIFAR-10 of the fine-tuning stage. The learning rate will decay by 5 every 10 epochs in ImageNet and every 80 epochs in CIFAR-10.

6.4.2 Evaluation

We conduct experiments in Table 6.1 and Table 6.2 to verify our method when given pre-trained models, such as the ResNet series [74], MobileNetV2 [180] and

doi:10.6342/NTU202202005



the VGG network [153], and some resource constraints. In our experiments, we adopt two commonly used concerned resources, which are the total FLOPs (f) and the network parameters (p), respectively. The given constraints would be the maximum proportion of the specific resource that could remain. For example, ($f_{.33},\ p_{.31}$) indicates that there should be at most 33% FOLPs and 31% parameters left of the pruned model. The performance of a pruned model would then be evaluated by its top-1 accuracy (P. Top-1) and the accuracy drop after pruning (Top-1↓).

Effectiveness of our CAIE We conduct ablation studies in Table 6.1 to compare the pruning results of the model **with** and **without** applying our CAIE on ImageNet and CIFAR-10. Each block in the table containing five rows shows a group of experiments on the pre-trained model and its original top-1 accuracy. The first row in each block is the "baseline" result without CAIE, which only takes the loss impact as the importance while pruning until separately meets the constraints. Since the constraints only determine the ending point of the baseline pruning, we will examine our CAIE pruning with specific constraints under which the baseline pruning obtains the same pruning result as the first row. The second and the third rows are used to confirm the correctness of our CAIE under a single constraint. As we can see, the results in these two rows have the same amount of the constrained resource compared to the baseline result yet reach better performance. The fourth row demonstrates the flexibility of our CAIE that can accurately adapt to the given multiple constraints. In some cases ($f_{.44},\ p_{.20}$ in VGG16-BN), the result generated by CAIE may not meet the constraints simultaneously, but the performance is still better than that in the baseline result. Last, the fifth row is to affirm the effect of CAIE when the given multiple constraints are set to the same as the resource consumption of baseline results, such as ($f_{.33},\ p_{.26}$) is the same as (32.83, 25.94) in the first block. As we can see, our results can still outperform the baseline for all the models. Altogether, our CAIE can always yield improvement in performance

doi:10.6342/NTU202202005

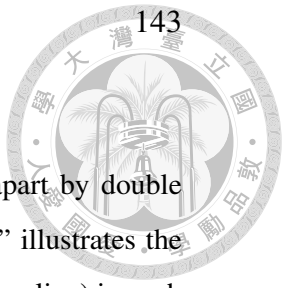


Table 6.1: **Ablation Studies of our CAIE.** Each block spaced apart by double line represents a group of experiments. The column "w/ − w/o CAIE" illustrates the performance gain after applying our CAIE comparing to the first row (baseline) in each block.

Model	**Constraints**	**w/ CAIE**	FLOPs left (%)	Param. left (%)	P. Top-1 (%)	Top-1↓ (%)	w/ − w/o CAIE (%)
ImageNet [152]							
ResNet-50 (orig. top-1 : 76.13%)	$f_{.33},\ p_{.31}$	✘	**32.83**	**25.94**	71.57	4.56	-
	$f_{.33}$	✔	32.95	49.40	73.90	2.23	2.33
	$p_{.26}$	✔	46.64	25.80	71.96	4.17	0.39
	$f_{.33},\ p_{.31}$	✔	32.90	30.76	72.39	3.74	0.82
	$f_{.33},\ p_{.26}$	✔	**32.47**	**25.89**	71.92	4.22	**0.34**
ResNet-50 (orig. top-1 : 76.13%)	$f_{.65},\ p_{.70}$	✘	**64.83**	**64.27**	75.59	0.54	-
	$f_{.65}$	✔	64.58	85.72	76.02	0.11	0.43
	$p_{.65}$	✔	79.80	64.70	75.80	0.33	0.21
	$f_{.65},\ p_{.70}$	✔	64.95	69.88	75.83	0.30	0.24
	$f_{.65},\ p_{.65}$	✔	**64.81**	**64.61**	75.69	0.44	**0.10**
ResNet-34 (orig. top-1 : 73.31%)	$f_{.78},\ p_{.79}$	✘	**77.55**	**71.43**	72.67	0.64	-
	$f_{.78}$	✔	77.47	90.43	73.15	0.16	0.48
	$p_{.72}$	✔	85.89	71.29	72.72	0.59	0.05
	$f_{.78},\ p_{.79}$	✔	77.43	78.94	72.91	0.40	0.24
	$f_{.78},\ p_{.72}$	✔	**77.72**	**71.32**	72.73	0.58	**0.06**
MobileNetV2 (orig. top-1 : 71.88%)	$f_{.50},\ p_{.32}$	✘	**49.80**	**31.49**	62.64	9.24	-
	$f_{.50}$	✔	49.77	67.58	67.23	4.65	4.59
	$p_{.32}$	✔	67.38	31.88	63.84	8.04	1.20
	$f_{.50},\ p_{.50}$	✔	49.72	48.24	6.28	5.60	3.64
	$f_{.50},\ p_{.32}$	✔	**49.63**	**31.97**	63.13	8.75	**0.49**
CIFAR-10 [166]							
VGG16-BN (orig. top-1 : 93.34%)	$f_{.44},\ p_{.20}$	✘	**43.32**	**9.93**	92.94	0.40	-
	$f_{.44}$	✔	44.00	12.55	93.06	0.28	0.12
	$p_{.10}$	✔	42.90	9.69	93.02	0.32	0.08
	$f_{.44},\ p_{.20}$	✔	43.07	12.19	93.11	0.23	0.17
	$f_{.44},\ p_{.10}$	✔	**42.43**	**9.89**	92.98	0.36	**0.04**
ResNet-34 (orig. top-1 : 94.13%)	$f_{.40},\ p_{.15}$	✘	**29.90**	**14.48**	93.34	0.79	-
	$f_{.30}$	✔	29.82	19.95	93.48	0.65	0.14
	$p_{.15}$	✔	35.69	14.79	93.46	0.67	0.12
	$f_{.40},\ p_{.15}$	✔	35.10	14.88	93.50	0.63	0.16
	$f_{.30},\ p_{.15}$	✔	**29.64**	**14.79**	93.40	0.73	**0.06**

doi:10.6342/NTU202202005

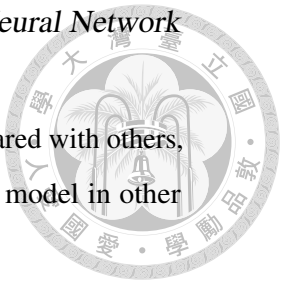


Table 6.2: **Comparison to state-of-the-arts on ImageNet.** To compared with others, we set the resource constraints based on the resource left of the pruned model in other works.

Model	Orig. Top-1 (%)	Method	FLOPs left (%)	Param. left (%)	P. Top-1 (%)	Top-1↓ (%)
ResNet-50	76.18	Taylor-FO-BN-56% [26]	32.76	30.86	71.69	4.49
	76.13	**Ours** ($f_{.33},\ p_{.31}$)	32.90	30.76	**72.39**	**3.74**
ResNet-50	72.88	Thinet-30 [21]	34.66	28.49	68.42	4.46
	76.13	**Ours** ($f_{.33},\ p_{.26}$)	32.47	25.89	**71.92**	**4.22**
ResNet-50	76.15	FPGM-only 30% [174]	58.80	-	75.59	0.56
	76.13	**Ours** ($f_{.55}$)	54.77	77.35	**75.62**	**0.53**
ResNet-50	76.18	Taylor-FO-BN-81% [26]	65.03	69.92	75.48	0.70
	76.13	**Ours** ($f_{.65},\ p_{.70}$)	64.95	69.88	**75.83**	**0.30**
ResNet-50	-	NISP-50-B [176]	55.99	56.18	-	0.89
	76.13	**Ours** ($f_{.56}, p_{.56}$)	55.89	55.84	**75.25**	**0.88**
ResNet-34	73.31	Taylor-FO-BN-82% [26]	77.74	78.90	72.83	0.48
	73.23	Li *et al.* [8]	75.80	89.20	72.17	1.04
	73.31	**Ours** ($f_{.78},\ p_{.79}$)	77.43	78.94	**72.91**	**0.40**

when given any constraints.

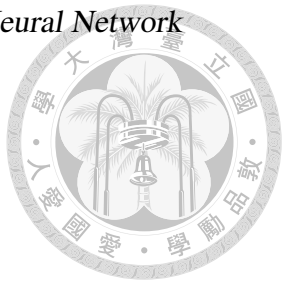
Comparison to state-of-the-arts In Table 6.2, we compare our CAIE with others on the ImageNet. Given a pruning result in other works, we will conduct experiments with our CAIE under the resource constraints corresponding to the resources left of others. Compared to the state-of-the-art Taylor-FO-BN [26], which contains the same calculation of loss impact and the pruning procedure as ours, results with CAIE can achieve better performance. Furthermore, compared to those with different importance estimation and the pruning process [21, 174, 176, 8], our method can still obtain the best results. It is worth noting that for a fair comparison, we did not show the results of GBN [172] because they apply other losses to reinforce the sparsity when training.

doi:10.6342/NTU202202005

6.5 Summary

In this work, we propose a novel method called Constraint-Aware Importance Estimation (CAIE) to estimate the importance of filters in the network under the given multiple resource constraints, which integrates information of the impact on the considered resources with the impact on loss function when removing a filter. We demonstrate the effectiveness of our method, and we can achieve state-of-the-art performance comparing to others under the same amount of resource consumption of the pruned model.

doi:10.6342/NTU202202005



doi:10.6342/NTU202202005

Chapter 7

Joint Generic and Personalized Federated Learning

In this chapter, we focus on the new trend for training a distributed models, called federated learning (FL). Owing to the more widespread applications of locally face recognition than person re-ID and more well-established large-scale benchmarks for face recognition, **we first explore the possibility of the combination for federated learning and face recognition**. Fig. 7.1 shows our expected FL scenario, where the face images on local clients are private and we aim at improving both the generic and personalized face representation of the pre-trained face model. It is worth noting that the mainstream training framework of re-ID is derived from face recognition because face recognition has been addressed for a long time. Specially, re-ID and face recognition are both the open-set problem, where the classes of training and testing set are different and non-overlapped. Thus, metric learning are the main solution in both tasks. The following sections will illustrate our proposed joint learning FL framework on face recognition benchmark but in our future work, illustrated in Sec. 7.7, we will apply our proposed joint learning scheme on existing person re-ID benchmarks to also formulate them as a FL problem.

doi:10.6342/NTU202202005

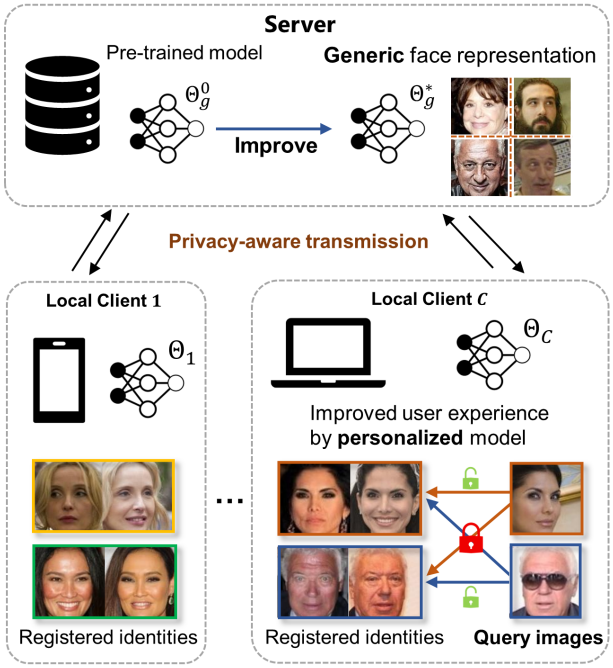
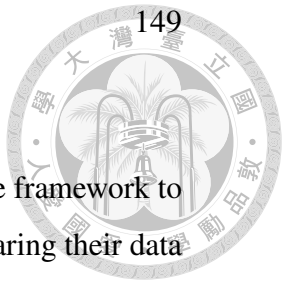


Figure 7.1: **The Federated Learning (FL) setup for face recognition**. Given a pre-trained face recognition model, we aim to simultaneously improve the generic face representation at the server, and produce an optimal personalized model for each client without transmitting private identities' images or features out of the local devices.

7.1 Introduction

Face recognition has been an active and vital topic among computer vision community for a long time. The state-of-the-art training frameworks formulate face recognition as a metric learning problem, and employ the large-scale identity classification as the proxy task to learn face features, which could discriminate between different identities robustly. Recently, the quick evolution of softmax-based loss functions for identity classification greatly promote the performance of face recognition. However, the training of face recognition model heavily relies on centralizing a huge amount of personal face images, which are usually not accessible due to the uprising privacy concern in many countries. Therefore, it is necessary to navigate the development of face recognition under the premise of

doi:10.6342/NTU202202005



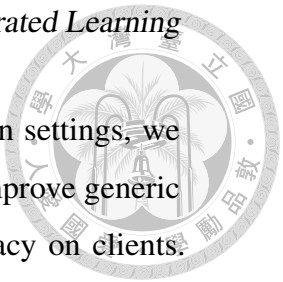
privacy preservation.

Federated learning (FL) provides a distributed and privacy-aware framework to train models where multiple clients collaboratively learn without sharing their data with the central server or other clients. A classical FL method called FedAvg [11] aggregates and averages the gradients from local clients on the server, and transmit the updated model back to the clients for the next round of local optimization. In the past few years, there has been significant progress in FL [182] on image classification task, which boosts the performance of aggregated global model under diverse FL scenarios. However, these approaches cannot be directly applied onto face recognition due to several critical reasons:

1. Face recognition is an open-set classification task, where training and testing identity classes are different.
2. The identity classes between local clients are different, which results in different model architectures in clients.
3. In a more practical setup for face recognition [183], the FL training starts from a publicly available face recognition model, rather than from scratch as in traditional FL.

In order to address these aforementioned issues, a recent work FedFace [183] proposed an FL framework for face recognition model training in a privacy-aware manner. It tackles the challenging setup where each of the participating clients has face images of only one identity. It employs a mean feature initialization method for the local identity proxy and a spreadout regularizer [184] at the server side to ensure that the identity proxies from the local clients are well separated. However, FedFace is limited as it only addressed a single scenario. In the real-world face recognition applications, local edge devices could be registered by multiple identities. Moreover, there exists a serious privacy concern in FedFace as it requires the local device to transmit the identity proxy to the server, which could violates the FL protocol [185]. A concurrent work [186] tries to mitigate this privacy concern through the Differential Privacy approach.

doi:10.6342/NTU202202005

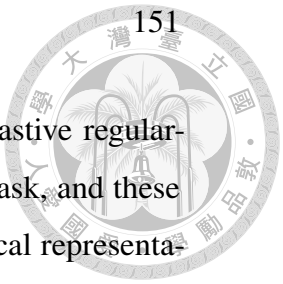


To enable federated learning in more realistic face recognition settings, we propose a novel framework called **FedFR**, which could jointly improve generic and personalized face representations without breaking the privacy on clients. First, we leverage the globally shared dataset to regularize the training on local clients, as the local client only has much less identities than the pre-trained dataset. With the additional transmission of the shared class embedding matrix, it can effectively prevent the local model from over-fitting and also improve the generic representation at the server. Secondly, in order to reduce the computation overhead and improve the training efficiency, a novel hard negative sampling strategy is proposed to select the most critical data samples from the globally shared dataset. In addition, a contrastive loss applied on the local face representation during training could further restrict the local model drifting. Last but not least, we are interested in simultaneously optimizing the user experience on local clients, which is not explored in previous works. Although personalized FL [187] has been studied for a while, those methods are sub-optimal on the face recognition task. We propose a Decoupled Feature Customization (**DFC**) module, which consists of a feature transformation layer and one-vs-all binary classifiers. The module locally learns a customized feature space which is optimized for recognizing the registered identities at each client.

We validate FedFR on IJB-C [188] dataset for the generic recognition model performance under different FL scenarios. We also build the personalized face recognition evaluation protocol with MS-Celeb-1M [189] dataset to validate the effectiveness of the proposed DFC module. Each technique in FedFR could substantially improve both generic and personalized face representations. Our main contributions are summarized as follows:

- We propose a novel joint optimization federated learning framework FedFR, which can effectively improve both generic and personalized face recognition models under different scenarios while strictly following the privacy constraints.

doi:10.6342/NTU202202005



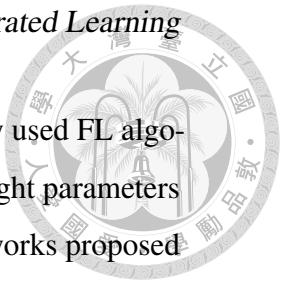
- Several training techniques (hard negative sampling, contrastive regularization) are proposed and tailored for the face recognition task, and these techniques can better bridge the gap between global and local representations.
- We propose the Decoupled Feature Customization (DFC) module, which is the key component to enable concurrent optimization of the personalized face recognition model. The proposed binary classification objectives are also effective for optimizing the performance on each client.
- Experimental results show that our proposed solution can consistently outperform previous approaches in several challenging generic and personalized FL benchmarks.

7.2 Related Work

Face Recognition Recently, great progress has been achieved in face recognition with large-scale training data [190, 189, 191], sophisticated network structures [192, 74] and advanced designs for softmax-based loss functions [77, 193, 194]. However, these state-of-the-art methods are not directly applicable to the federated learning setting since they assume centralized data is available on a server. Without the access to private face images from local clients, the feature learning is prohibited as the model cannot compare features between different identities. In addition, how to leverage additional identities to improve the feature incrementally based on a pre-trained face recognition model was never discussed in previous works, as they always assumed to train the model from scratch. In our federated setup, we aim to improve a publicly available pre-trained face recognition model at the server from multiple clients in a collaborative manner, while keeping the private face images and identity features at the local clients.

Federated Learning Federated Learning (FL) [3, 182, 195] is a learning setup in machine learning which aims to learn a model over multiple disjoint clients while

doi:10.6342/NTU202202005



maintaining local data privacy. The most well-known and commonly used FL algorithm is FedAvg [11], which learns a global model by averaging weight parameters across local models trained on private client datasets. Many recent works proposed to improve FedAvg from different perspectives: model convergence [196, 197], robustness [198], communication [199], and non-IID clients [200, 201]. Most of the previous computer vision related FL works only studied image classification tasks with small-scale datasets (e.g. MNIST, CIFAR-10). To the best of our knowledge, FedFace [183] is the only one which addressed the face recognition model training in the federated setup. To enhance the pre-trained FR model, it applies the spreadout regularizer [184] at the server side to ensure the identity proxies from clients are well separated. Our work differs in that we do not transmit identity prototypes as it could leak the private identity info from clients. Moreover, our work is scalable to different scenarios where each client contains more than one identity.

Personalized Federated Learning Personalized FL [187] aims to learn a customized model to meet each client's objective. Instead of training a single "general" model which is optimized for generic metric, this FL setup seeks to acknowledge the data heterogeneity among clients by constructing a "personalized" model which fits each client's need. Many recent techniques [202, 203, 204] proposed to leverage multi-task learning (MTL) [205] methods to incorporate clients' task objectives into the FL framework. Another stream of approaches [206, 207] employed meta-learning to learn a decent initial model that can be adapted to each client after some steps of local fine-tuning. Besides, [208] showed that conducting post-processing (e.g. fine-tuning) onto a generic FL model could achieve comparable results with other personalized methods. However, the latter two streams of approaches would require an additional stage for local adaptation. Our framework employs the MTL based approach which can optimize general and customized face recognition models simultaneously.

doi:10.6342/NTU202202005

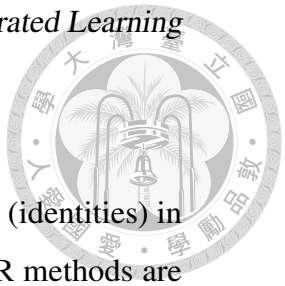
7.3 Proposed FedFR

In this work, we build a novel FL framework for the face recognition (FR) task. In the following, we will first establish the proposed FL setup for joint generic and personalized face recognition. Next, we introduce some preliminaries of our framework, which are some basic techniques popularly employed in FR and FL respectively. Then, we will describe technical details of the proposed FedFR solution.

7.3.1 Problem Setup

Face recognition systems are widely applied on local user devices. Typically, the deployed model is trained on a public dataset in advanced on a server. To continuously improve the generic face representation, the intuitive way is to collect the images stored in local devices (clients) and update the model trained with augmented data. However, as mentioned previously, due to privacy issues, it is prohibited to upload any identity-related information, such as the face images and its features. Federated learning (FL) provides a framework to train models where multiple clients collaboratively learn without sharing their data with the server or with other clients. As shown in Fig 7.1, different from typical FL setting that learns the model from scratch, in face recognition, we target on **how to enhance the generic representation of pre-trained model by leveraging the data on clients under the privacy constraint**. Besides, we also focus on the optimized user experience. Although an improved generic model can implicitly achieve it, a client-specific personalized model optimized by local objectives could achieve optimal performance on the device. Thus, we jointly consider the situation that **whether we can obtain a personalized face model which is dedicated to recognize the registered identities on each client**. To the best of our knowledge, we are the first to explore the personalized FL setup in face recognition.

doi:10.6342/NTU202202005



7.3.2 Preliminaries

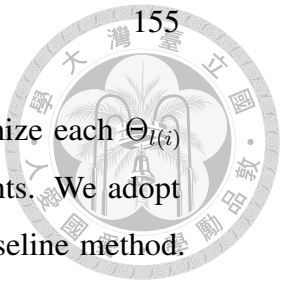
Face Recognition FR is an open-set problem, where the classes (identities) in training and testing are different. In the training phase, current FR methods are typically based on an identity classification objective, where the model embeds an input image into a high-dimensional representation and generates the class logits by computing the similarity between the input feature and all class embeddings (proxies). Then a softmax cross-entropy loss will be adopted to supervise the model. In our setting, the pre-trained generic face model is trained with the commonly used Cosface loss [77], which adopts an additive margin softmax. Formally, given the face embedding model Θ and an input image x with y-th class, we can obtain its deep feature $f = \Theta(x) \in \mathcal{R}^d$. There is also a class embedding matrix $\Phi \in \mathcal{R}^{d \times K}$, where K is the total number of classes and the j-th column Φ_j means the learned proxy of j-th class. Following Cosface loss, the original j-th logit ($\Phi_j \cdot f + b$) will be simplified by ignoring the bias b and normalizing the $\|f\|$ and $\|\Phi_j\|$ to 1, which is just the cosine similarity $\cos\theta_j$. Last, the additive margin softmax cross-entropy loss for x will be computed as follows:

$$\mathcal{L}_{cos} = -\log \frac{e^{s(\cos\theta_y - m)}}{e^{s(\cos\theta_y - m)} + \sum_{j \neq y}^{K} e^{s\cos\theta_j}}, \tag{7.1}$$

where s and m are the scaling constant and the additive margin, respectively. During the testing stage, the learned face embedding model Θ will embed the query face image into a d-dim face feature, and the system would compare the cosine distance between the query feature and pre-registered features for identity authentication.

Federated Learning In our FL setup, we consider C local client nodes and one central server with the face recognition model Θ_g^0 pre-trained on a publicly available large dataset D_g, which has N_g images from K_g identities. Each local client i is initialized with $\Theta_{l(i)}^0 = \Theta_g^0$ and registered with $N_{l(i)}$ images from $K_{l(i)}$ identities, which is much smaller than the public one. Our objective is to simultaneously

doi:10.6342/NTU202202005



improve the model Θ_g for generic face representation and optimize each $\Theta_{l(i)}$ for personalized client customization under the privacy constraints. We adopt the most commonly used FL algorithm, **FedAvg** [11], as our baseline method. Due to the mutual exclusive classes between local clients, we follow previous FL works [209, 210] that only send the backbone model Θ to the server, and keep the class embedding matrix on clients. The steps for collaborative training by server and clients are as follows:

1. In the t-th communication round, the server sends the global model Θ_g^t to all client nodes.
2. The i-th client updates the model $\Theta_{l(i)}^t$ at round t based on $N_{l(i)}$ local data and local learned class embedding $W_{l(i)}$ with Cosface loss $\mathcal{L}_{cos}$, which is a $K_{l(i)}$-class classification problem.
3. The local clients only send the backbone model $\Theta_{l(i)}^t$ to the server. The server will update the global model by taking a weighted average of them as follows:

$$\Theta_g^{t+1} = \frac{1}{N} \sum_{i \in [C]} N_{l(i)} \cdot \Theta_{l(i)}^t, \tag{7.2}$$

   where N is the total number of training images across all client nodes.
4. Last, the updated global model will then be transmitted to each client and steps $2-4$ are repeated until convergence.

FedAvg can perform well on clients with IID-distributed data. However, for our face recognition setup, the identity distributions on each client are different. Just optimizing on local data with limited number of identities to obtain $\Theta_{l(i)}^t$ could harm the original performance of the pre-trained model (as shown in the experimental results). Furthermore, although $\Theta_{l(i)}^t$ can improve the personalized representation for these $K_{l(i)}$ identities, it will be continuously updated by the global model along the communication rounds, which cannot achieve optimal performance for the local users.

doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

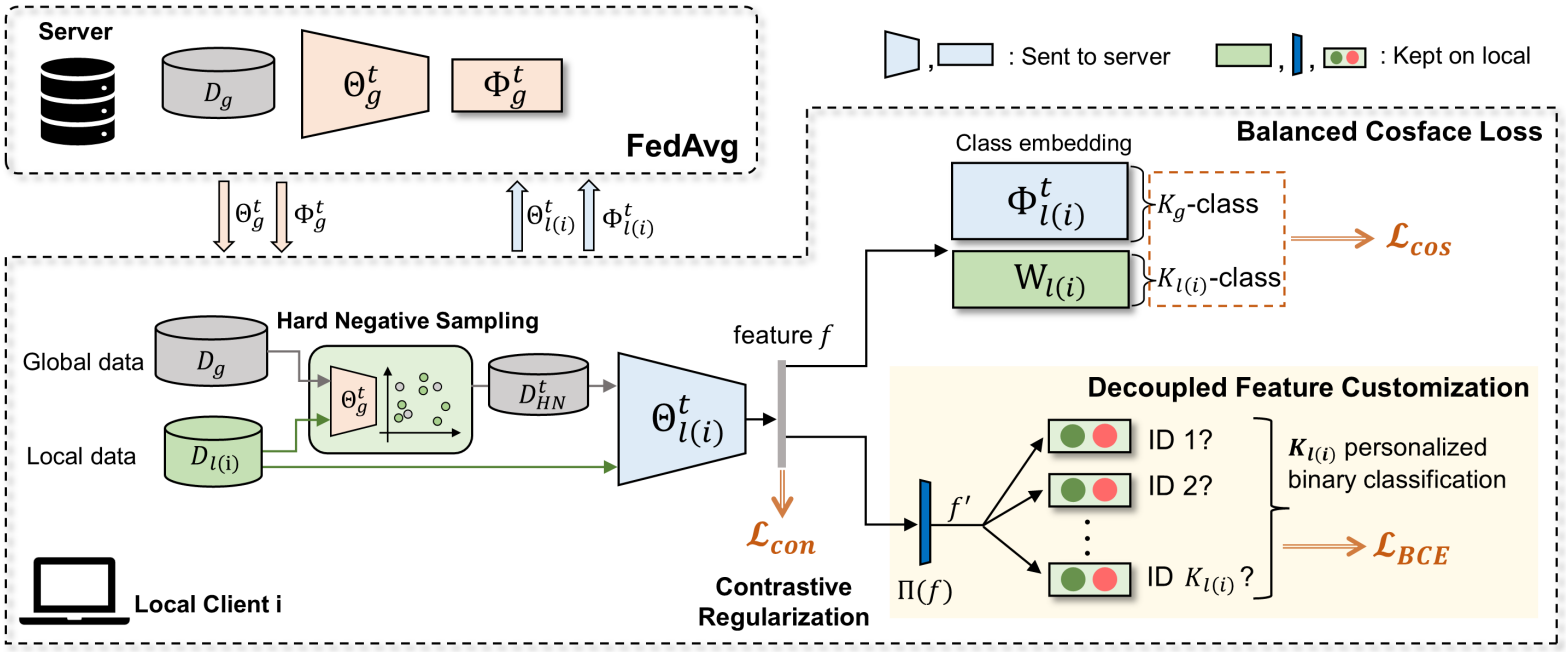


Figure 7.2: **FedFR.** We demonstrate the overall architecture of our method. For model on each client i, it will be optimized with balanced Cosface loss, a contrastive regularization and the binary cross entroy in our Decoupled Feature Customization branch. After training, the backbone model $\Theta_{l(i)}$ and global class embedding $\Phi_{l(i)}$ will be uploaded for FedAvg.

7.3.3 FedFR: Joint Optimization Federated Framework

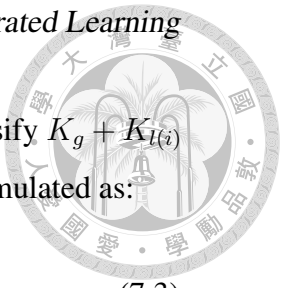
To tackle the issues in FedAvg, we propose a joint optimization framework **FedFR**, which can effectively improve the generic face representation at the server with the use of globally shared data, and also optimize the personalized recognition performance simultaneously at local clients. We first provide an overview of FedFR, and the system architecture is also illustrated in Figure 7.2. Built upon the baseline FL pipeline, we introduce several novel techniques:

1. We employ the globally shared dataset D_g to better regularize the local model training, which could prevent the model from over-fitting on local identities.
2. The Hard Negative Sampling strategy is introduced to select the most critical data from D_g to significantly reduce the computation on local clients.
3. The Contrastive Regularization is employed to control the drift of model parameters and better bridge the gap between global and local representations.
4. To simultaneously optimize face representation for local clients, we propose the **Decoupled Feature Customization** module to transform the global representation for better fitting the local distributions. The corresponding margin-based binary classification loss $\mathcal{L}_{BCE}$ establishes a better local objective to supervise the learning of the decoupled branch

We elaborate each technique in details as follows.

Leveraging Globally Shared Data Some previous FL works on image classification [211, 212] has shown that leveraging globally shared dataset can better address the issue of heterogeneous clients. In the face recognition FL setup, the global dataset D_g which was used for pre-training the server model can be naturally shared to all the local clients. We could further regularize the training of local clients by providing the class embedding matrix Φ_g of the shared K_g identities. As shown in Figure 7.2, given the shared dataset D_g on client i, the local client could build a more "balanced objective" by concatenating $\Phi^t_{l(i)} = \Phi^t_g$ with the local private

doi:10.6342/NTU202202005



embedding matrix $W_{l(i)}$ as a new learnable proxies and learn to classify $K_g + K_{l(i)}$ identities with $\mathcal{L}_{cos}$. Thus, our balanced Cosface loss would be formulated as:

$$\mathcal{L}_{cos} = -\log \frac{e^{s(\cos\theta_y - m)}}{e^{s(\cos\theta_y - m)} + \sum_{j \neq y}^{K_g + K_{l(i)}} e^{s\cos\theta_j}}, \tag{7.3}$$

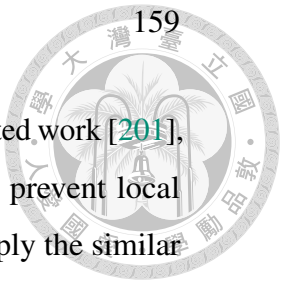
where the denominator is added with additional K_g negative terms. For the end of each round t, beside sending the backbone $\Theta^t_{l(i)}$ back to server, the learned class embeddings $\Phi^t_{l(i)}$ related to K_g global identities can also be sent back and updated by:

$$\Phi^{t+1}_g = \frac{1}{N} \sum_{i \in [C]} N_{l(i)} \cdot \Phi^t_{l(i)}. \tag{7.4}$$

Hard Negative Sampling Strategy Jointly training with D_g can prevent model from over-fitting on local data. However, the large number of public data will also increase the computation burden on local clients, which will enlarge the training time and degrade the communication efficiency between server and clients. To obtain a better trade-off, we propose a hard negative (HN) sampling strategy to only choose a subset D_{HN} from D_g, which is critical for learning with $D_{l(i)}$. The proposed technique is described as follows.

At the start of each communication round t on local client i, we first forward the global and local data to Θ^t_g to generate their features. Then we can calculate the pair-wise cosine similarity between them. To make the training more efficient but at the same time maintain the performance, we only sample the "hard" global data for model learning, which is with similarity larger than threshold t_{HN} to any of the local data. Intuitively, with larger t_{HN}, the less global data will be used for training. We decide the threshold by leveraging the inherent feature space of the pre-trained model. As mentioned above, the pre-trained model is trained with Cosface loss, where the similarity of each sample to its proxy should be larger than those to others by a margin m. Thus, if any negative pair with similarity larger than $t_{HN} = m$, they should be served as a hard negative pair.

doi:10.6342/NTU202202005



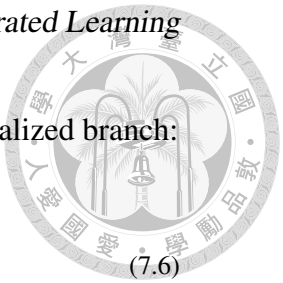
Contrastive Regularization on Local Clients Inspired by the related work [201], which proposed a model-contrastive loss on the local training to prevent local model from deviating too much from the global model, we also apply the similar regularization on our face recognition task. Namely, we aim to decrease the distance between the face representation learned by the local model at time t ($f = \Theta^t_{l(i)}(x)$) and the one learned by the global model ($f_{glob} = \Theta^t_g(x)$), and increase the distance between the face representation learned by the local model at time t ($f = \Theta^t_{l(i)}(x)$) and time $t-1$ ($f_{prev} = \Theta^{t-1}_{l(i)}$). Thus, the local contrastive loss term $\mathcal{L}_{con}$ is defined as

$$\mathcal{L}_{con} = -\log \frac{\exp(\text{sim}(f, f_{glob})/\tau)}{\exp(\text{sim}(f, f_{glob})/\tau) + \exp(\text{sim}(f, f_{prev})/\tau)}, \tag{7.5}$$

where "sim$(\cdot,\cdot)$" measures the cosine similarity between face features, and τ denotes a temperature hyperparameter.

Decoupled Feature Customization With the contrastive regularization, the local model can avoid over-parameterizing for the local objective and continuously improve the generic face representation. However, it will go against the goal which we aim to simultaneously obtain a personalized model to improve the local user experience. Thus, as shown in Figure 7.2, we propose a novel Decoupled Feature Customization (DFC) module to resolve this seemly contradicting scenario. In order not to influence the feature f for generic representation, inspired by [213], we adopt a transformation $\Pi(f)$ with a fully-connected layer to map it to a client-specific feature space, which can recognize the $K_{l(i)}$ identities well. To achieve this goal, there should be a local objective for optimization. Inspired by [214], we propose to adopt the binary classification on each local identity for the personalized purpose. Given the transformed feature $f' = \Pi(f)$, we will feed it into $K_{l(i)}$ binary classification branches (which the total trainable weight vectors are denoted as $\Omega_{l(i)}$). The k-th module contains learnable parameters which only target on classifying the positive samples from the k-th class and the negative samples from "any other" classes. Formally, we follow the loss in the related work that used

doi:10.6342/NTU202202005



margin-based binary cross-entropy ($\mathcal{L}_{BCE}$) to supervise our personalized branch:

$$\mathcal{L}_{BCE} = \frac{\lambda}{s'} \cdot \log\Big(1 + \exp\big(-s' \cdot (g(\cos\theta_k) - m')\big)\Big) + \frac{1-\lambda}{s'} \cdot \sum_{j \neq k} \log\Big(1 + \exp\big(s' \cdot (g(\cos\theta_j) + m')\big)\Big), \tag{7.6}$$

where $\cos\theta_j$ is the cosine similarity of transformed input feature f' and the j-th weight vector $\Omega_{l(i),j}$ in the binary classification, b is the learned bias, and the function $g(z) = 2((z+1)^{t'}/2) - 1$ is used to increase the empirical dynamic range of cosine similarity. The notations λ, s' and m' all follow those in the related work, which are the balanced factor, scaling constant and cosine margin.

It is worth mentioning that although there are only $K_{l(i)}$ binary classification branches, not only the local data but the global data can be used to optimize our DFC module because each branch only needs to recognize "whether it is the k-th identity or not". This objective just well-fits our personalized goal that given an unseen query image, a well-performed local face recognition system should quickly determine whether it is the registered identity or not.

Learning Pipeline Our overall learning framework is based on FedAvg, where there will be T communication rounds and in each round, the local clients will update the model for E epochs. In the local client training, the model will be optimized in an end-to-end manner with the total objective $\mathcal{L}_{total}$, which is formulated as:

$$\mathcal{L}_{total} = \alpha_1 \mathcal{L}_{cos} + \alpha_2 \mathcal{L}_{con} + \alpha_3 \mathcal{L}_{BCE}, \tag{7.7}$$

where all the modules $\Theta^t_{l(i)}, \Phi^t_{l(i)}, W^t_{l(i)}, \Pi^t_{l(i)}, \Omega^t_{l(i)}$ and bias b would be updated. However, only the $\Theta^t_{l(i)}$ and $\Phi^t_{l(i)}$ will be sent back for globally averaged with (7.2) and (7.4). In the testing phase, Θ_g is used for generic evaluation and $[\Theta_{l(i)}, \Pi_{l(i)}]$ is used for personalized evaluation. More details of the whole algorithm pipeline are in the supplementary materials, which is in Sec. 7.6.

doi:10.6342/NTU202202005

7.4 Experiments

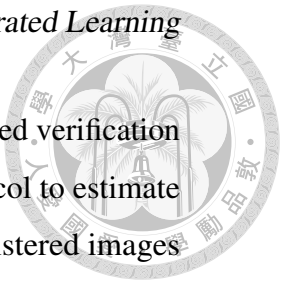
7.4.1 Experimental Setup

Dataset We use the MS-Celeb-1M [189] as the training dataset. To avoid the long-tail distribution, we manually select 10k identities from the dataset where each identity contains 100 face images. Within the selected subset, 6000 identities (K_g) are used for pre-training the global model, and the other 4000 identities are equally distributed into local clients. For each identity in each local client, we use 60 images for local training, and 40 images for personalized model evaluation, respectively. Besides MS-Celeb-1M, IJB-C [188] dataset which contains 3531 identities with diverse appearance is used for evaluating the generic model performance. The selected list for FL training will be released for fair comparison in the future.

Evaluation Metrics For the generic model evaluation, we strictly follow the IJB-C evaluation protocol, which is commonly used in the face recognition community. We report the true acceptance rates (TAR) at different false acceptance rates (FAR) for 1:1 verification protocol, and true positive identification rates (TPIR) at different false positive identification rates (FPIR) for 1:N identification protocol.

Regarding the personalized model evaluation, we carefully build up the metrics and protocols as we are the first to investigate the personalized face recognition setup. The evaluation is supposed to only focus on the face recognition user experience of the registered identities on each local client. Therefore, we establish two evaluation protocols to better measure the client-specific performance: 1) Firstly, similar to the 1:1 verification protocol in IJB-C, we establish a list of positive pairs and negative pairs for evaluation. In each client, we formulate genuine matches from local identities and build up imposter matches by pairing one local identity with a random identity from other clients. For the 40 local clients scenario where each client is registered with 100 identities, there are 7.8k positive pairs and about 630 million negative pairs in one client. We average the

doi:10.6342/NTU202202005



true acceptance rates (TAR) across all clients as the final personalized verification performance. 2) Secondly, we build up an 1:N identification protocol to estimate the login experience on a local client (device). Intuitively, the registered images from one local identity are combined to form its gallery feature. And the testing images from all clients are taken as the probe features. For the 40 local clients scenario, there are 100 gallery features and 160k probe features in one client. Similarly, we average the true positive identification rates (TPIR) across all clients as the final personalized identification performance.

Implementation Details For the backbone face model, we adopt the same 64-layer CNN architecture from [76, 77], which outputs a 512-dimensional feature vector. The image preprocessing techniques are the same as [193], where the image is cropped to size 112×112 and the pixel value is normalized to $[-1, 1]$. To simplify our network training, all hyper-parameters in $\mathcal{L}_{cos}$, $\mathcal{L}_{con}$ and $\mathcal{L}_{BCE}$ are empirically set as the same ones in the related work, where m=m'=0.4, s=s'=30, τ=0.5, λ=0.7 and t'=3. For $\mathcal{L}_{total}$, the α_1, α_2 and α_3 are empirically set as 1, 5 and 10. We adopt SGD optimizer with weight decay 5×10^{-4} and learning rate 0.001. For the FL setup, we conduct T=30 communication rounds and in each round the local clients conduct E=4 epochs.

7.4.2 Ablation Studies

Effectiveness of each modules To validate the effectiveness of each proposed module, we report the ablation studies in Table. 7.1. The experiments are conducted with one central server and 40 clients, where each client contains 100 identities. The performance is evaluated both on the generic and personalized benchmark. If it is under the FL setup, the global model Θ_g will be used to test on the generic evaluation and each local model $\Theta_{l(i)}$ will be tested on personalized data, where the shown scores are the average over all clients. Notes that for the 1:N identification in personalized evaluation, we average the feature of training images based on their

doi:10.6342/NTU202202005

Table 7.1: **Ablation Studies.** We conduct FL experiments with **40** clients; each client contains **100** identities. (results are in %)

Setup	Modules			Generic Evaluation (IJB-C)				Personalized Evaluation			
	HN. sampled Global data	Contrastive	DFC. Branch	1:1 TAR @ FAR		1:N TPIR @ FPIR		1:1 TAR @ FAR		1:N TPIR @ FPIR	
				1e-5	1e-4	1e-2	1e-1	1e-6	1e-5	1e-5	1e-4
Centrally trained on 6k IDs (pre-training)				76.42	84.58	72.06	80.30	56.28	72.50	71.73	82.33
Federated Learning on 4k IDs	✘	✘	✘	73.79	83.71	67.59	78.53	67.33	85.70	82.77	92.27
	✔	✘	✘	76.79	84.64	72.76	80.76	81.75	91.91	91.97	96.09
	✔	✔	✘	77.41	85.17	73.60	81.25	77.77	89.57	89.58	94.60
	✔	✔	✔	**77.60**	**85.21**	**73.60**	**81.27**	**88.32**	**95.46**	**95.17**	**97.94**
Centrally trained on 10k IDs				77.56	85.99	73.30	82.14	93.72	97.39	98.58	99.40

doi:10.6342/NTU202202005

doi:10.6342/NTU202202005

identities as the gallery features in that client.

The first row is the performance of pre-trained model trained on public data with 6k classes, which is the target model that needs to be improved. Start from 2^{nd} to 5^{th} row, the FL setup is employed where 4k augmented IDs are added but with privacy constraints. And for the last row, it is the ideal situation that we can centrally optimize the model with data of 10k IDs. We can see that in the second row, our baseline method which directly optimizes the model with local data and perform FedAvg on the server cannot perform well. The performance is even worse than the pre-trained one owing to the over-fitting on local data. Leveraging the public data is a solution, but it may suffer from long training time and large computation overhead. With our proposed Hard Negative sampling strategy where only a subset of global data serving as negative pairs to the local data, in the 3^{rd} row, not only the generic representation but also the personalized evaluation can be boosted. Contrastive regularization is designed for regularizing the local model from training towards the undesired local minimum. We can see that in the 4^{th} row, the performance improves greatly on generic evaluation. However, under the same feature space parameterized by Θ, a more generalized representation will harm the performance for recognizing specific identities on clients. Thus, in the 5^{th} row, which is our final proposed FedFR architecture with the DFC branch, we decouple the feature from the original feature space to a new one with a transformation $\Pi_{l(i)}$, and optimize this space with binary cross-entropy loss tailored for the personalization. We can see that with Θ_g for generic representation and $[\Theta_{l(i)}, \Pi_{l(i)}]$ for personalized evaluation, both of them can achieve superior results.

Analysis of the t_{HN} in Hard Negative Sampling In our experiments, we choose t_{HN} equals to the margin m=0.4 in Cosface used in pre-training the model. To validate the effectiveness, as shown in Figure 7.3, we demonstrate the global performance on IJB-C and its training efficiency under different hard negative

doi:10.6342/NTU202202005

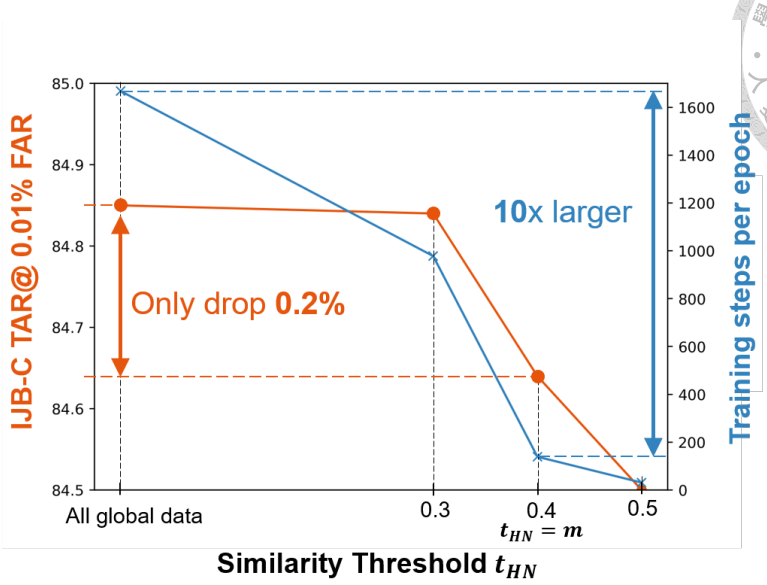


Figure 7.3: The generic model performance and the model training efficiency under different Hard Negative thresholds.

thresholds with 100 IDs per client. The training efficiency is measured in terms of the training steps per epoch. We can see that with t_{HN}=0.4, the number of sampled global data can be largely reduced by **10** times but with only 0.2% drop of the global performance, which is the best trade-off configuration in our experiments.

7.4.3 Comparison with FedFace

To compare the results with FedFace [183], as shown in Figure 7.4, we construct the FL setting with diverse identities per client under total 100 clients, which is from 40 to 1. We demonstrate the results of the pre-trained model, ideal central training (upper bound), FedFace and our proposed FedFR. Because FedFace cannot be adopted on multiple IDs in a client and their FL dataset is not released, we re-implement their method on our setting that uses Cosface loss as the local objective if the number of ID is larger than 1, and also apply spreadout regularizer at the server side to separate the class proxies from clients. From the comparison on the generic model performance, FedFace could easily over-fit on local dataset and performs inferior to the pre-trained model in these scenarios. In contrast, our

doi:10.6342/NTU202202005

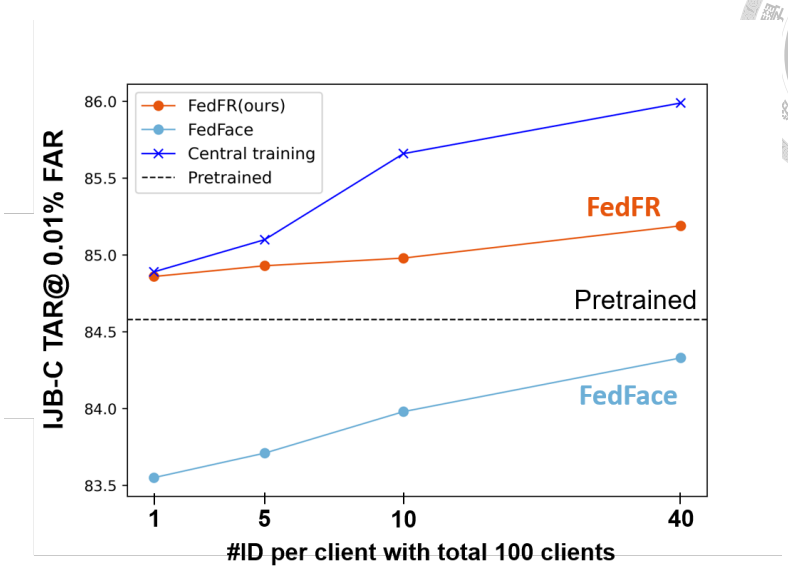


Figure 7.4: **Generic model performance compared to FedFace.** We fix the number of clients to 100 and conduct 4 scenarios of different #IDs in one client.

proposed FedFR can still improve the generic face representation under the most challenging scenario where there is only one identity in the client.

7.4.4 Comparison with Personalized FL Methods

To validate the effectiveness of our Decoupled Feature Customization (DFC) module, we compare with the latest personalized FL method [208], which is a two-stage local adaptation approach. For fair comparison, we re-implement the "Fine-tune" and "KD" local adaptation methods, which were shown to be effective in image classification tasks, in our face recognition setup. In the first stage, the server and clients collaboratively learn to obtain a great generic model, where we use the proposed hard negative sampling strategy and the contrastive regularization in the experiments. Then, in the second stage, each client separately optimizes its local model for personalization. For the "Fine-tune" method, we directly optimize each model with Cosface loss with the local and sampled global data. For the "KD" method, it is with a Knowledge Distillation technique that besides the original Cosface loss, we also supervise the output logits of local model (student) by the

doi:10.6342/NTU202202005

Table 7.2: **Comparison of other personalized techniques.** It is conducted on 40 clients with 100 IDs per each.

Method	**Modules**	**Personalized Evaluation**			
		1:1 TAR @ FAR		1:N TPIR @ FPIR	
		1e-6	1e-5	1e-5	1e-4
Yu et al.	Fine-tune	73.81	86.21	88.37	93.90
2020	KD	75.82	87.65	89.50	94.67
Ours	Cosface	82.93	91.88	90.67	95.59
(w/ branch)	**BCE**	**88.32**	**95.46**	**95.17**	**97.94**

logits generated from original global model (teacher) with KL-Divergence loss. As illustrated in Table. 7.2, our proposed one-stage personalization method can outperform the two local adaptation strategies. In addition, we also conduct a variant of our method, which is also a decoupled branch but adopts a Cosface loss with multi-class classification for supervision. It is clearly verified that the proposed binary classification objective better fits the need for the personalized face recognition on clients.

7.5 Summary

In this work, we address the face recognition model training under the practical federated learning setting, where each client is initialized with the pre-trained model. We propose a novel joint optimization framework FedFR, which can improve the generic face representation of the global model and at the same time enhance the personalized user experience. While the proposed hard negative sampling and contrastive regularization can efficiently bridge the gap between global and local training, the Decoupled Feature Customization (DFC) module is another novel component to enable concurrent optimization of the personalized face recognition model. The effectiveness of the proposed solution is verified on several challenging generic and personalized face recognition benchmarks. We hope that the work and the release of the personalized FR benchmark can facilitate the future research on the federated learning for face recognition.

doi:10.6342/NTU202202005

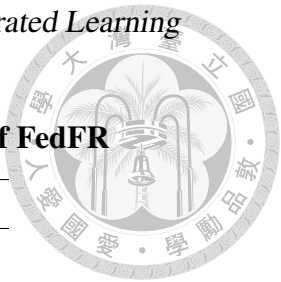


Table 7.3: **Choices of models on Personalized Evaluation of FedFR**

Evaluated Model	Personalized Evaluation			
	1:1 TAR @ FAR		1:N TPIR @ FPIR	
	1e-6	1e-5	1e-5	1e-4
Θ_g	70.95	84.40	80.30	88.98
$\Theta_{l(i)}$	81.46	92.17	91.15	95.64
$[\Theta_{l(i)}, \Pi_{l(i)}]$(**Ours**)	**88.32**	**95.46**	**95.17**	**97.94**

7.6 Supplementary Materials of FedFR

7.6.1 Learning Pipeline of FedFR

Algorithm 4 demonstrates our whole FedFR pipeline in detail. Given a central server and C local clients, we will conduct T communication rounds and in each round, the local client conduct E training epoch. After local training, all the $\Theta^t_{l(i)}$ and $\Phi^t_{l(i)}$ will be sent to server and perform FedAvg to generate Θ^{t+1}_g and Φ^{t+1}_g. To evaluate the global generic evaluation, the model Θ_g will be used and for each local personalized evaluation, we will concatenate the $[\Theta_{l(i)}, \Pi_{l(i)}]$ as the personalized backbone to generate customized feature representation.

7.6.2 Models for Personalized Evaluation

For personalized face recognition, we want to validate that using $[\Theta_{l(i)}, \Pi_{l(i)}]$ can obtain the best results compared to using Θ_g or $\Theta_{l(i)}$ for personalized evaluation. Table 7.3 demonstrate the results on different models trained under our FedFR framework. The first row is the personalized result of global model Θ_g. We can see that it is not suitable for user customization. The second row is the local backbone $\Theta_{l(i)}$ without the concatenation of $\Pi_{l(i)}$. We can see that the local feature can improve the performance, but with our Decoupled Feature Customization, the personalized feature ($[\Theta_{l(i)}, \Pi_{l(i)}]$) can well-fit the local distribution and outperform all the others.

doi:10.6342/NTU202202005

Algorithm 4: The FedFR framework

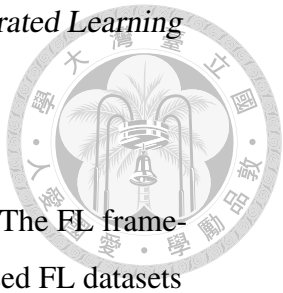
Input: #communication rounds T, #clients C, #local epochs E, hard negative threshold t_{HN}, hyper-parameter α_1, α_2, α_3, learning rate μ

Output: global model Θ_g, personalized models $\Pi_{l(i)} and \Theta_{l(i)}$ (i = 1, 2, ... C)

Server executes:
initialize Θ_g^0 and Φ_g^0 with the pre-trained model
for $t = 0, 1, ..., T-1$ **do**
  for $i = 1, 2, ..., C$ **do**
    Send the global models Θ_g^t and Φ_g^t to client i
    $\Pi_{l(i)}, \Theta_{l(i)}^t, \Phi_{l(i)}^t \leftarrow$ **ClientTraining** (i, Θ_g^t, Φ_g^t)
  end for
  $\Theta_g^{t+1} \leftarrow \frac{1}{N} \sum_{i \in [C]} N_{l(i)} \cdot \Theta_{l(i)}^t$
  $\Phi_g^{t+1} \leftarrow \frac{1}{N} \sum_{i \in [C]} N_{l(i)} \cdot \Phi_{l(i)}^t$
end for
return Θ_g^T, $\Phi_{l(i)}^T$, $\Pi_{l(i)}^T$ (i = 1, 2, ... C)

ClientTraining (i, Θ^t, Φ^t):
$\Theta_{l(i)}^t, \Phi_{l(i)}^t \leftarrow \Theta^t, \Phi^t$
$W_{l(i)}^t, \Pi_{l(i)}^t, \Omega_{l(i)}^t \leftarrow W_{l(i)}^{t-1}, \Pi_{l(i)}^{t-1}, \Omega_{l(i)}^{t-1}$
$D_{HN}^t \leftarrow$ select hard negative sets with threshold t_{HN} from D_g
for epoch e = 1, 2, ..., E **do**
  for each batch **B** = {x, y} of $D_{l(i)} \bigcup D_{HN}^t$ **do**
    $\mathcal{L}_{total} \leftarrow \alpha_1 \mathcal{L}_{cos}(\Theta_{l(i)}^t, \Phi_{l(i)}^t, W_{l(i)}^t, \mathbf{B}) + \alpha_2 \mathcal{L}_{con}(\Theta_{l(i)}^t, \Theta_{l(i)}^{t-1}, \Theta^t, \mathbf{B}) + \alpha_3 \mathcal{L}_{BCE}(\Theta_{l(i)}^t, \Pi_{l(i)}^t, \Omega_{l(i)}^t, \mathbf{B})$
    $\Theta_{l(i)}^t \leftarrow \Theta_{l(i)}^t - \mu \bigtriangledown_{\Theta_{l(i)}^t} \mathcal{L}$
    $\Phi_{l(i)}^t \leftarrow \Phi_{l(i)}^t - \mu \bigtriangledown_{\Phi_{l(i)}^t} \mathcal{L}$
    $\Pi_{l(i)}^t \leftarrow \Pi_{l(i)}^t - \mu \bigtriangledown_{\Pi_{l(i)}^t} \mathcal{L}$
    $W_{l(i)}^t \leftarrow W_{l(i)}^t - \mu \bigtriangledown_{W_{l(i)}^t} \mathcal{L}$
    $\Omega_{l(i)}^t \leftarrow \Omega_{l(i)}^t - \mu \bigtriangledown_{\Omega_{l(i)}^t} \mathcal{L}$
  end for
end for
return $\Pi_{l(i)}^t, \Theta_{l(i)}^t, \Phi_{l(i)}^t$

doi:10.6342/NTU202202005



7.7 Future Work

Pedestrian data captured by surveillance cameras is also sensitive. The FL framework for optimizing re-ID model is necessary. We follow the proposed FL datasets in [215] that four datasets are served as clients, shown in Table 7.4. The testing set of the clients are served as the personalized evaluation in our setting. The other four datasets are unseen and served as the generic evaluation. We preliminary adopt the baseline method FedAvg [11] to analyze the performance under FL scenario.

Table 7.4: Statistics of person re-ID FL datasets.

Types	Datasets	Train ID	Train Img	Test		
				Test ID	Query Img	Gallery Img
Clients (Personalized)	Market1501 [12]	751	12936	750	3368	19732
	DukeMTMC [1]	702	16522	702	2228	17661
	CUHK03 [47]	767	7365	700	1400	5328
	MSMT17 [49]	1041	30248	3060	11659	82161
Unseen (Generic)	VIPeR [9]	-	-	316	316	316
	iLIDS [51]	-	-	60	60	60
	GRID [216]	-	-	125	125	125
	PRID [216]	-	-	100	100	649

Table 7.5 show the preliminary results, it can be seen that adopting FedAvg can achieve comparable results of personalized evaluation to the training only on the local data itself. In the future, we will explore the combination of our proposed (1) hard negative sampling, (2) contrastive regularization, and (3) decoupled feature transformation techniques on federated re-ID benchmarks.

Table 7.5: **Results on clients and unseen datasets.** Thanks Shu-Yu Lin for helping conduct experiments.

Method	**Market**		**Duke**		**CUHK03**		**MSMT17**	
	mAP	R1	mAP	R1	mAP	R1	mAP	R1
local supervised	68.2	87.4	57.0	77.2	39.2	43.2	28.8	60.0
FedAvg [11]	70.8	88.0	57.7	74.6	29.6	31.8	32.8	51.7
Method	**VIPeR**		**iLIDS**		**GRID**		**PRID**	
	mAP	R1	mAP	R1	mAP	R1	mAP	R1
FedAvg [11]	48.4	38.6	39.7	28.0	20.8	39.7	70.0	77.2

doi:10.6342/NTU202202005

Chapter 8

Prototype of Real-time Online MTMC Tracking System

8.1 Introduction

Multi-Target Multi-Camera Tracking (MTMCT) algorithms have been studied in many existing works [38, 39, 40]. However, they only focus on "offline processing", which means given all recorded surveillance videos, they separately perform pedestrian detection on each frame, single-camera tracking on each video, and multi-camera tracking on each complete trajactory (also called track) across all cameras. Although this pipeline achieves the best performance, it is not practical in real-world surveillance system. In realistic situation, we are only able to deploy an **"online"** MTMCT system, where each module can only process the video frames had been recorded, and can not acquire any frame in the future.

To build an online system, the easiest way is transmitting all recorded frames from cameras to the cloud server and processing the detection, tracking, and trajactories association on it; then sending the results of matching pedestrians across cameras to the end user, which is shown in Fig. 8.1. Nonetheless, as mentioned in Chapter 1, this system will encounter many challenges owing to increasing resolution of videos, computation burden of each computer vision algorithm and

doi:10.6342/NTU202202005

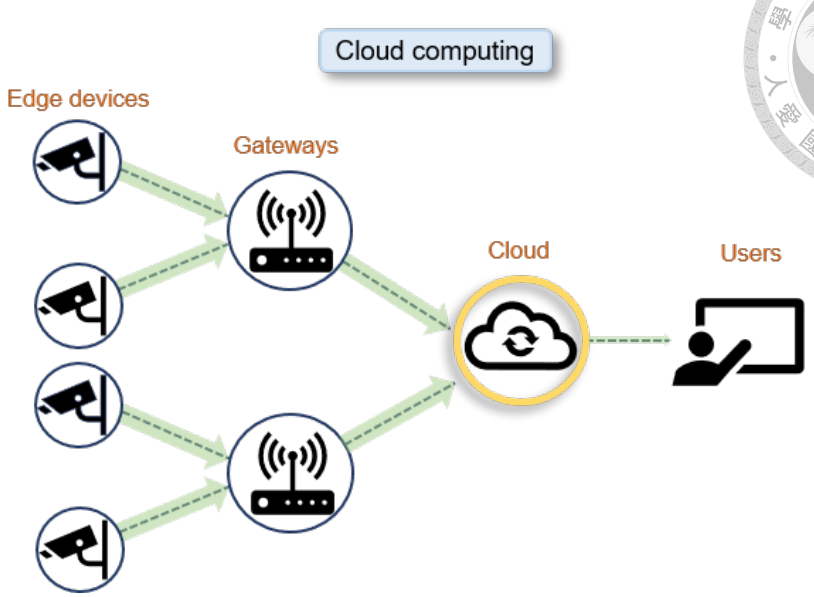


Figure 8.1: **Typical cloud computing scheme.** The thickness of green arrow illustrates the amount of transmission data. The circle in yellow represents the unit responsible for computing.

the privacy sensitive issues. First, the resolution will directly influence the available transmission bandwidth. Green arrows in Fig. 8.1 illustrate the amount of transmission data. If the bandwidth is not enough, it may result in delays in traveling time between cameras and gateways, or gateways and cloud server. This is not desirable in an online system if we expect it operating in the real-time speed (about 30 FPS). The second problem is that the computation of complicated algorithms are all on the cloud, which is illustrated with yellow circle in Fig. 8.1. With the development of deep-learning, each sub-task in MTMC system needs CNN for generating high-performance results. To operate the whole system in the real-time speed with data from multiple cameras simultaneously, we will need a powerful GPU on the cloud server, which causes high power consumption and costs. Last but not the least, the privacy is also an issue. Gateways are typically deployed under the same local network of cameras, whereas the cloud server is not. The transmission process between gateways and cloud may be hacked and thus the private data will has the chance to be leaked into the public internet.

doi:10.6342/NTU202202005

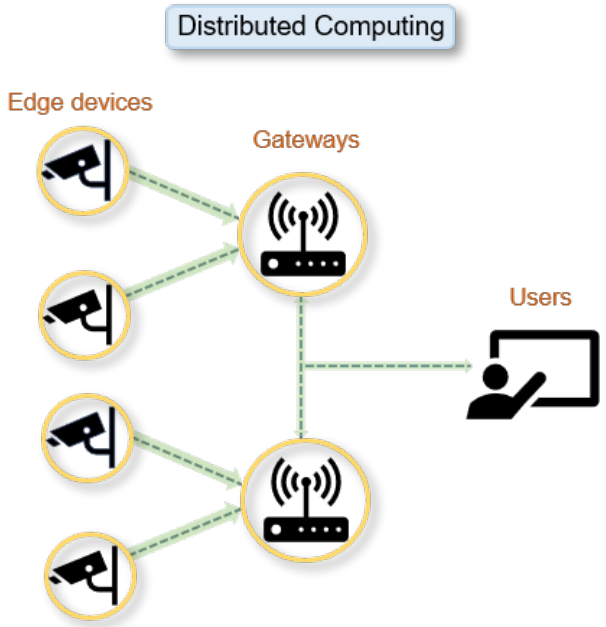


Figure 8.2: **Proposed distributed computing scheme.** The thickness of green arrow illustrates the amount of transmission data. The circle in yellow represents the unit responsible for computing.

To solve the aforementioned issues, we propose a distributed online MTMC system, as shown in Fig. 8.2. First, we enable the computation ability on each local device, including the cameras and gateways. Because the low-power constraints and low computation ability of the embedded SoC (System on Chip) engine, each device can only execute part of the computation of the whole system, such as only the object detection or the re-ID feature extraction; furthermore, in order to increase the latency, each task can only utilize a light-weight architecture of the CNN models. With the on-device computation, we can shrink the amount of data for transmission, for example, after the object detection on local cameras, we can only transmit the cropped bounding boxes for single-camera tracking. Second, we remove the role of central server, where each gateway will have the ability to communicate with others. Ideally, after the gateways compute re-ID features for a track, it will need to associate gallery features from other cameras. Thus, with the real-time communication, each gateway can contain a shared information

doi:10.6342/NTU202202005

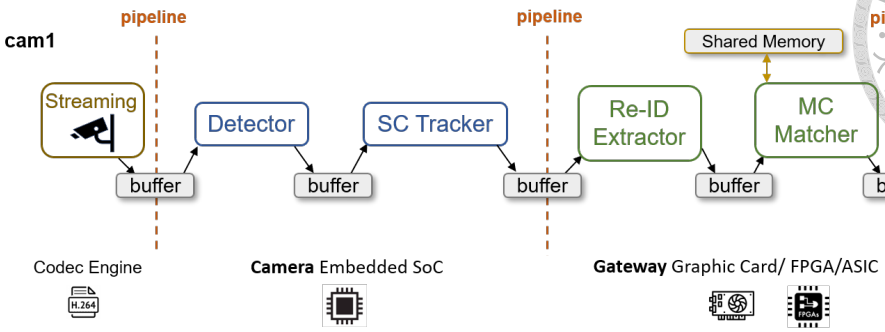


Figure 8.3: **Ideal pipeline framework for practical MTMCT system.**

memory of all the trajectories having exited their belonged camera scene. Without the central server, there will be no privacy issues of the system and it can protect the leakage of sensitive data.

In this dissertation, we practically build the prototype of our proposed online MTMC tracking system. The overall system architecture is introduced in Sec. 8.2. In order to operate it in a real-time speed, for object detection and single-camera tracking, we adopt state-of-the-art algorithms and the off-the-shelf network optimization mechanism, which will be described in Sec. 8.3. For the design of multi-camera tracking, we adopt our proposed video-based person re-ID and constraint-aware pruning algorithm to perfectly make our model perform efficiently. The details will be introduced in Sec. 8.4.

8.2 System Architecture

In order to make our prototype system flexible for replacing each module with existing stat-of-the-art CNN architectures, we build our system under the Python framework [217]. As shown in Fig. 8.3, to increase the processing speed, under each camera stream, we split our system into four parallelly executed pipelines (each is a thread in Python), where the bottom line illustrates the ideal hardware engine for running this distributed MTMCT system. The first thread is for video streaming, which will be executed on the Codec engine on the surveillance camera.

doi:10.6342/NTU202202005

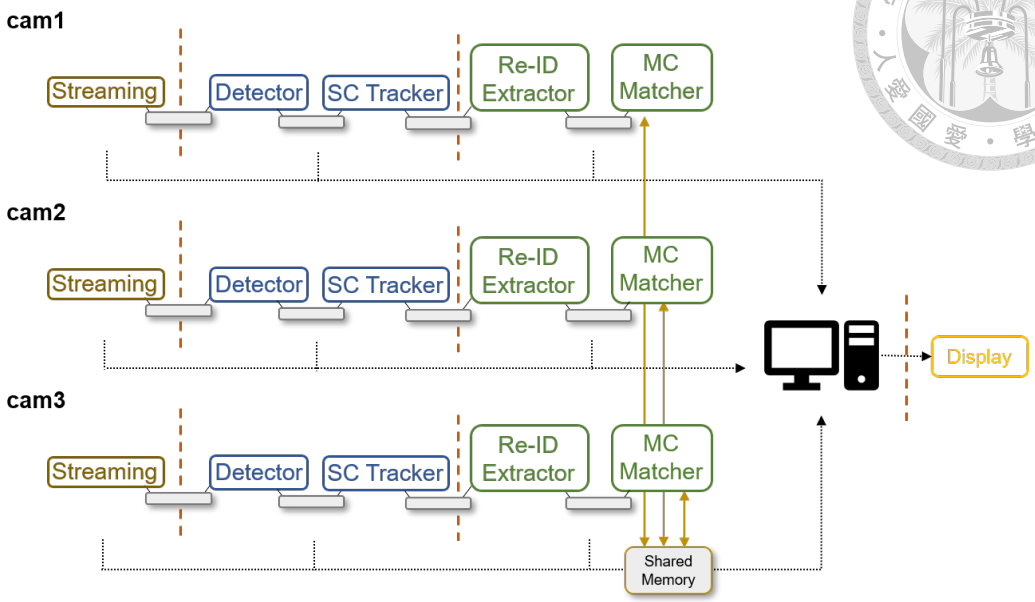
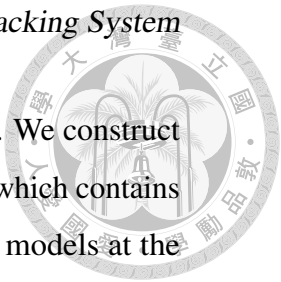


Figure 8.4: **Simulated pipeline framework for our MTMC system.**

Furthermore, if the camera is embedded with SoC system, we can execute some light-weight models on it. We put the detector and single-camera (SC) tracker on the second thread for generating tracks of people. Then, the instant results will be written to the output buffer and sent to the gateways of the local network. We assume gateway has more computation power than the camera, which may contains low-end graphic card or specific ASIC and FPGA. In the third thread, the re-ID extractor will first generate features of each track. Owing to the online characteristic that the length of each track will increase along time, we will continuously accumulate the features and compute the running average to represent the track. To perform multi-camera (MC) matching, the gateways will query the features of other pedestrians having left their camera views, which are saved in a shared memory that is synchronized across all gateways. The last thread is for displaying the results on original frame image. We build the display system with Flask package [218], which enables users monitoring each frame from all cameras through the front-end web page.

To quickly demonstrate our online system, we construct a simplified version,

doi:10.6342/NTU202202005



where we remove the transmission between each embedded device. We construct three independent camera streams on one single mid-end computer, which contains only one GPU for simulating all computing units executing CNN models at the same time. Fig. 8.4 demonstrates our final system architecture. We further compare the specs of our demo PC system with the low-end embedded system on the market in Table. 8.1. It can be seen that it is reasonable for simulating the whole system, which ideally contains multiple embedded systems, with only a single mid-end GPU. In detail, if we demonstrate the system with three camera streams, there will be 6 threads simultaneously execute the CNN models with the aid of GPU. Thus, using one mid-end GPU can not only reduce the burden for building and maintaining the communication protocols between devices but also help us correctly evaluate the latency and effectiveness of our algorithms.

In the following sections, we will introduce the implementation details of each module and also demonstrate the optimization methods for making our system execute in a real-time speed (∼30FPS). In brief, we adopt TensorRT [219] in our detection module and our CAIE pruning technique on our spatially efficient video-based re-ID module.

Table 8.1: **Hardware Specs of my PC and embedded devices on the market.**

My PC	
Processor	Intel Core i7-8700K (6-cores)
RAM	16GB
GPU	Name : Nvidia RTX 2070 CUDA Core : 2304 Mem. bandwidth : 448 GB/s
Nvidia Jetson Xavier NX	
Processor	NVIDIA Carmel ARM v8.2 (6-cores)
RAM	8GB
GPU	CUDA Core : 384 Mem. bandwidth : 59.7 GB/s

doi:10.6342/NTU202202005

8.3 Pedestrian Detection and Tracking

For the two sub-tasks, we directly survey state-of-the-art algorithms to meet our needs. In object detection, we can choose the effective Yolov4 [34] series, and in SCT, we can adopt the fast and accurate DeepSort [220]. However, concatenating those two tasks are regarded as "two-stage" tracking-by-detection method. It is inefficient because we need to execute two types of CNNs consecutively, one for detection and the other for tracking. In recent years, one-stage method has been more and more popular since it only needs one CNN for both detection and tracking. JDE [87] utilize the final features of Yolov3 [221] as the representation of pedestrians in DeepSort, and FairMOT [35] adopt CenterNet [222] as the backbone to generate the bounding boxes and the features at the same time. Although these one-stage methods can largely reduce the computation burden, as the analysis in [35], there exists some inherent contradictions. For pedestrian detection, the final features of the backbone are used to classify whether it is a pedestrian or not and regress the coordinates. Thus, the network will try to make the appearance features of all pedestrians more similar to each other and more different from other non-pedestrian classes. In contrast, for tracking, we need discriminative features of persons which can help correctly associate the same identity in the environment. With this contradiction, we cannot both enhance the performance of tracking and detection at the same time. Recently, an outstanding work [5] proposed a simple but very effective one-stage tracking method called ByteTrack. It first suggest that we should utilize a powerful and efficient detection module. With the rapid progress of object detection, they choose to adopt the recently proposed YoloX [4], which has achieved the best trade-off on all benchmarks. Then, with accurate bounding boxes at hands, they proposed their BYTE tracker, which is the advanced version of outdated IoUTracker [223]. This tracker didn't need any appearance feature but surprisingly achieve promising results. In our MTMCT system, we choose to adopt ByteTrack and we will briefly introduce the details in the following.

ByteTrack contains two parts, YoloX for detection and BYTE tracker for SCT,

doi:10.6342/NTU202202005

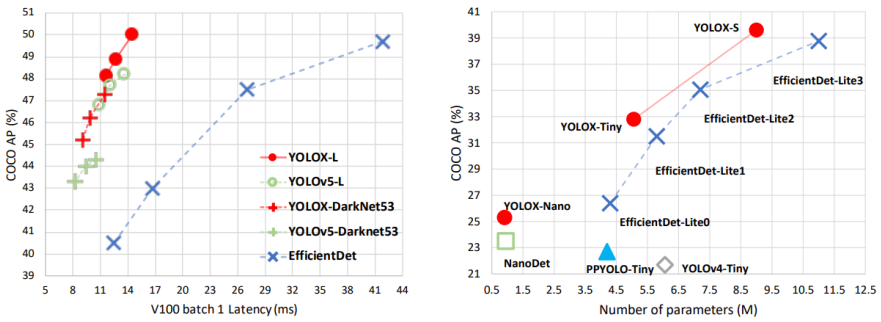


Figure 8.5: Speed-accuracy trade-off of accurate models (left) and Size-accuracy curve of lite models on mobile devices (right) for YOLOX and other state-of-the-art object detectors. **This figure and captions are all copied from [4].**

which is a two-stage traditional IoUTracker. YoloX is an improved version of Yolov5 [224], where the backbone model is the same as them. The difference is that YoloX utilize the anchor-free detection head which is similar to FCOS [225]. With anchor-free design, there is no need for calculating the suitable anchor sizes at first. YoloX also proposed many training techniques, and readers can refer to their paper for more information. In brief, YoloX achieve the best trade-off between the performance and the inference latency, which is shown in Fig. 8.5.

After detecting the pedestrians, ByteTrack [5] proposed that we should keep both the boxes with high confidence and low confidence. Previous works only retain the high-confidence boxes for data association owing to avoiding the false positive boxes. However, when some true positive boxes are occluded, their confidence scores decrease and they will be filtered out automatically. This results in fragmentation and ID switch of the trajactories, which will degrade the overall performance. Their BYTE tracker subvert the concept and keep the boxes into two parts for their proposed two-stage association. In the first stage, they associate the boxes with high confidence with the tracked tracks and the unmatched tracks will be left for the second stage. Then, in the second stage, the boxes with low confidence will be matched to those left tracks. It is worth noting that in each matching phase, only the IoU calculated by bounding boxes will be used, which means there is no

doi:10.6342/NTU202202005

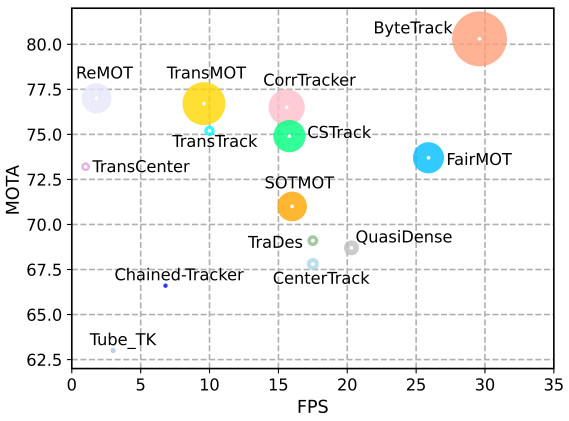


Figure 8.6: ByteTrack achieves the best performance and best FPS. **This figure is copied from [5].**

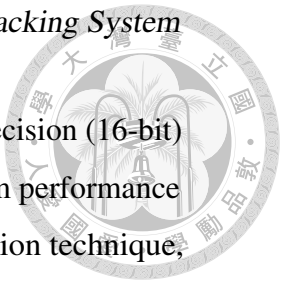
CNN inference in the BYTE tracker. The unmatched low-confidence boxes will be removed and only the unmatched high-confidence boxes will have the chance to be initialized as the new-born tracks. Readers can refer to their papers for more details, and Fig. 8.6 illustrates their trade-off comparing to other tracking algorithms on the popular MOT benchmarks [226].

For the implementation details, considering the computation costs, we adopt YoloX-S model as the detection backbone, and utilize the pre-trained weights provided by [5] which is trained on MOT17 dataset [227]. The input image size of our model is reisized to 1088×608, which is commonly used in the tracking tasks. In the following, we will use "ByteTrack" to represent the YoloX-S detection and BYTE tracker.

8.3.1 Problems Related to Inference Latency

Performed on our PC, if there is only one camera stream, it can operate in real-time speed, which is at most $\sim$ 37 FPS. However, if we parallely execute three camera streams, it can only achieve average 25 FPS, as shown in the upper part of Table 8.2. To improve the latency, in recent Nvidia GPUs (including low-levels

doi:10.6342/NTU202202005



embedding devices and PC-level graphic cards), it supports half-precision (16-bit) floating points operations, which will barely influence the algorithm performance in inference stage. Thus, we directly apply this inherent optimization technique, which is shown in the middle part of Table 8.2. For three parallel camera streams, it can increase about 1 FPS obviously. Nonetheless, it does not meet our need of the real-time speed. We then apply the optimization toolkit called TensorRT [219] proposed by Nvidia. It can optimize the model architecture and the scheduling of load-store operations from memory which are dedicated for the running platform. As a result, as shown in the bottom part of Table. 8.2, our system can operate in real-time speed, which can process at the 36.1 FPS. This represents that out object detection and tracking tasks are not the bottleneck of the whole MTMC system. If we can optimize the afterward re-ID feature extraction and MC matching with real-time speed, the whole system can meet our needs.

8.3.2 Problems Related to Tracking Performance

For our demo system, we utilize DukeMTMC [134] videos as the input streams. However, the ground truth annotations have been taken down owing to the privacy issues. We can only visualize the bounding boxes to evaluate our system. Thus, after visualizing some difficult cases, we found that our ByteTrack cannot perform well when two pedestrians walk through each other in the same horizontal line. When they are overlapped, the association of only using IoU of bounding boxes will fail, which is shown in Fig. 8.7 that after the overlap of two persons, the

Table 8.2: **Optimization of ByteTrack in FPS.**

Methods	System	FPS
ByteTrack	1 camera	37.0
	3 cameras	25.1
ByteTrack + **half-precision**	1 camera	37.0
	3 cameras	26.1
ByteTrack + half-precision + **TensorRT**	1 camera	36.5
	3 cameras	**36.1**

doi:10.6342/NTU202202005

Figure 8.7: **Visualization of two identities before/after the overlap with original ByteTrack.**

identity will switch to each other.

To make a trade-off between the processing speed and the tracking performance, we embedded a light-weight ResNet-18 [74] (R-18) as the appearance feature extractor into original ByteTrack. Before associating with the high-confidence bounding boxes, we will first extract the appearance features and associate them with the feature distance. Although the ResNet-18 is not a powerful feature extractor, we only need it to help match the bounding box and the track with high visual similarity. Thus, the threshold of the feature distance is small enough to ensure that we will not match two identities with slightly similar visual appearance. The overall detection and tracking flow in our system is shown in Fig. 8.8. To maintain the latency, we also apply TensorRT on our R-18 network. The degradation of running latency can be seen in Table 8.3, where it only decreases 1 FPS in our system. The visualization can be seen in Fig. 8.9, where the bounding boxes and the corresponding identities are correct along the timestamps.

Table 8.3: **Degradation of our R-18+ByteTrack in FPS.**

Methods	**System**	**FPS**
Optimized ByteTrack	3 cameras	36.1
Optimized ByteTrack **+ R-18**	3 camera	35.1

doi:10.6342/NTU202202005

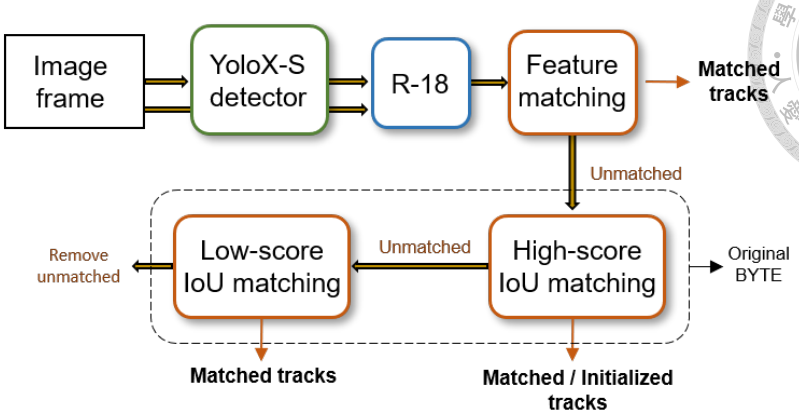


Figure 8.8: **The flow of our detection and tracking system under one camera.**

Figure 8.9: **Visualization of two identities before/after the overlap with our ByteTrack+R-18.**

8.4 Multi-Camera Tracking and Latency Improvement

For multi-camera tracking, we first need to generate the re-ID features of each track, which is anticipated to be invariant of a person across cameras. Because each person under a single camera contains a track with continuous cropped bounding boxes along the time, we can adopt our proposed non-local video-based re-ID model introduced in Chapter 2. However, we cannot directly obtain the complete track owing to the online setting, where the bounding boxes of each identity are sent to MCT module instantly and continuously. Thus, we decide to generate the

doi:10.6342/NTU202202005

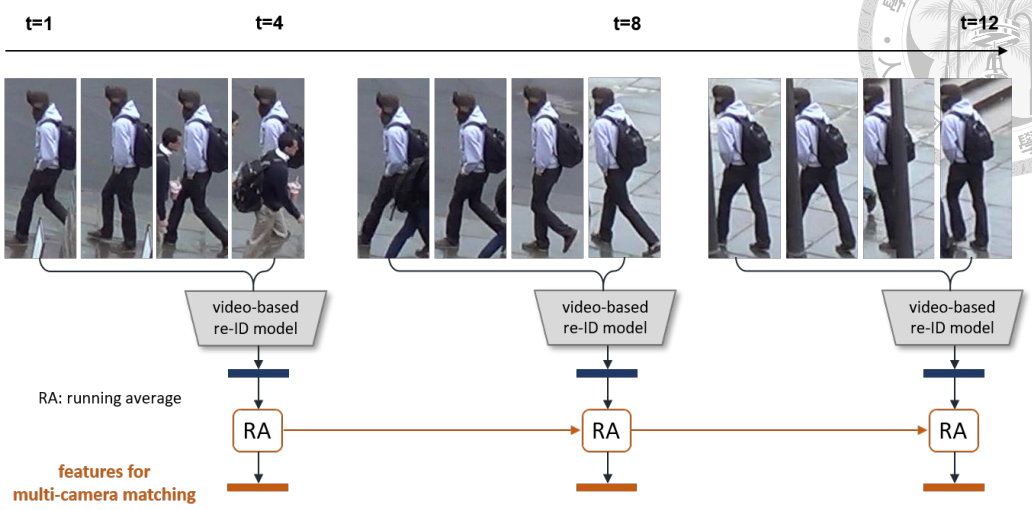


Figure 8.10: **The flow of generating re-ID features of each track.**

re-ID feature once we collect T frames of that track, and adopt the running average to represent the overall features from the start of the track. In our system, we choose $T = 4$, and Fig. 8.10 demonstrates our flow for first utilizing re-ID model and then applying running average to generate features for multi-camera matching.

To choose a suitable re-ID model, we compare the performance and latency of each model we've proposed. In our system, the cropped person image is resized to 256×128 and thus, the testing input of our model is with size $4 \times 3 \times 256 \times 128$, where 4 is the number of frames and 3 is the number of image channels. Table 8.4 illustrates the results. We compare the latency of four different models on CPU and GPU, where the second row is the spatially efficient version of NVAN that the non-local attention is split to multiple stripes. The latency is calculated by the average of model inference in 100 times. It is worth noting that there is no obvious difference of latency on GPU between each model. We think that because GPU can accelerate the matrix multiplication, if only one model and one input data are executed on the GPU, there is no burden for it and instead, the model with other non-matrix-multiplication operations will increase the inference time. In contrast, if we put the model on the hardware with only 6-cores CPU, we can see the difference of each model. Considering the performance and the latency, in our

doi:10.6342/NTU202202005

Table 8.4: **Comparison of model latency (sec) and performance on DukeV**

Methods	Latency on CPU	Latency on GPU	mAP
NVAN [17]	0.143	0.012	96.1
Spatial-NVAN [17]	0.101	0.013	96.1
STE-NVAN [17]	0.088	0.013	95.5
CF-AAN [18]	0.118	0.015	96.2

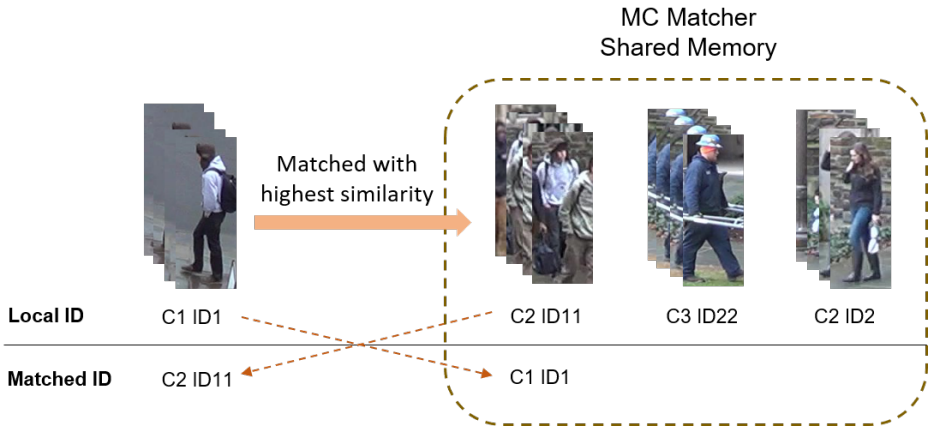


Figure 8.11: **The flow of query the cross-camera ID from shared memory.** After matching with highest similarity, we will record the camera and ID of each other, where C1 means camera 1.

system, we adopt the "Spatial-NVAN" as our finalized model in the re-ID feature extractor.

After obtaining the features, we can perform multi-camera matching, which is the last step in Fig. 8.4. The MC matcher contains a shared memory which contains the features of each person left their cameras. For the current track in the view, we will query the memory every 40 frames to match whether there is a same person leaving other cameras. If the matched feature similarity (the largest one) is larger than a threshold, we will save the matching information and display on the videos. Fig. 8.11 demonstrates our matching flow, where the ID 1 in camera 1 will be matched to the ID 11 under camera 2.

doi:10.6342/NTU202202005

Table 8.5: **Comparison of system FPS with different re-ID model**

System				FPS
#cameras	Detector	SCT	Re-ID	
3	YoloX	BT+R-18	-	35.1
3	YoloX	BT+R-18	NVAN	29.1
3	YoloX	BT+R-18	Spatial-NVAN	30.0
3	YoloX	BT+R-18	R-34	34.5
3	**YoloX**	**BT+R-18**	**Spatial-NVAN + CAIE** ($f_{.75}$)	**33.6**

8.4.1 Improvement of Latency with our CAIE Pruning

Table. 8.5 demonstrates the system FPS with different re-ID models, where 'BT' is the BYTE tracker. As shown in the first three rows, using the computationally expensive NVAN will degrade the average FPS from 35.1 to 29.1. After changing to proposed Spatial-NVAN, it can increase to 30.0. However, we still hope to increase the latency. One of the choices is adopting TensorRT as in Sec. 8.3.1. In this dissertation, instead, we choose to combine our proposed CAIE pruning illustrated in Chapter 6, into the proposed Spatial-NVAN model. In CAIE, we first need to target on some hardware constraints; thus, we explore another light-weight model, which is the Resnet-34 in our experiments, to served as the target # FLOPs of the CAIE. As shown in the fourth row of Table 8.5, using R-34 (ResNet-34) as the re-ID feature extractor can maintain the system with 34.5 FPS. Therefore, because the FLOPs of R-34 is about $0.75\times$ of the Spatial-NVAN, we set the constraint $f_{.75}$ in our CAIE. The last row of Table 8.5 shows our final result. We didn't illustrate the re-ID evaluation on DukeV but the pruned model still can achieve the same performance of the unpruned one. For the latency, our pruned model can increase the FPS from 31.0 to average $\mathbf{33.6}$ of all three cameras, which is larger than the real-time speed.

It is worth noting that although the FLOPs of pruned model is equal to that of R-34, it cannot truly reflect on the latency. The number of layers, number of channels and the acceleration of GPU will all influence the final inference speed. The optimal way is to measure the model latency as the hardware constraint in our

doi:10.6342/NTU202202005

CAIE, and owing to the complicate reform of CAIE if using latency as constraints, we only adopt the basic version, where the advanced one is introduced in the thesis [228].

8.5 Visualization of our System

Fig. 8.12 demonstrates two successfully matched identities across cameras. Let's first see Fig. 8.12(a), where the matched identity is walking from camera 3 to camera 2. In the frame No. 211 under camera 3, the man is tracked with ID 2. After exiting camera 3 and entering camera 2, the man with locally tracked ID 22 is correctly matched to previously appeared ID 2 under camera 3. Fig. 8.12(b) illustrates the other example that the ID 25 under camera 2 is also correctly matched to the ID 19 under camera 1.

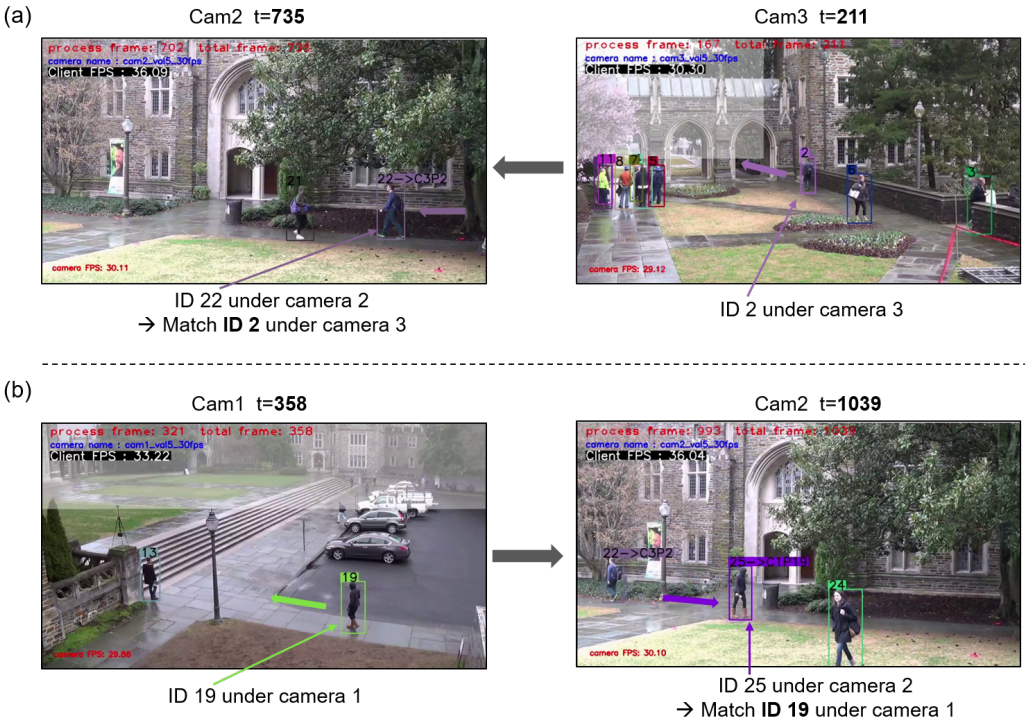


Figure 8.12: **Visualization of two successfully matched identities in our system.**

doi:10.6342/NTU202202005

8.6 Summary

In this chapter, we introduce a distributed MTMCT system, which can alleviate the burden of transmission bandwidth and the privacy leakage problem. Under this scenario, we construct a prototype online system executed on one PC with mid-level graphic card to simulate the real-world scenario and evaluate the performance of each sub-module. With our optimization, each sub-module can operate in real-time speed, which is larger than 30 FPS. Specifically, for the re-ID model, we combine our proposed spatial-NVAN [17] and proposed CAIE pruning [30] to reduce the complex computation of filter convolutions.

doi:10.6342/NTU202202005

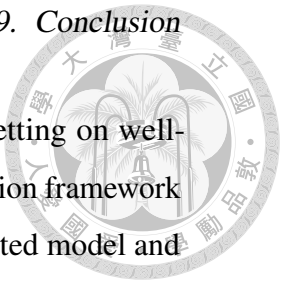
doi:10.6342/NTU202202005

Chapter 9

Conclusion

In this dissertation, we target on building a real-world multi-target multi-camera tracking system. The core technique is the person re-identification (re-ID) for multi-camera tracking. Thus, we address many practical scenarios in the learning of re-ID model to make it efficient and effective. In Chapter 2, we address the video-based supervised re-ID, where the state-of-the-art non-local video attention network is proposed. However, to make it more efficient, we propose spatial and temporal reduction versions to alleviate the computation burden of self-attention operations. Also, we found that the noise in the dataset influences the fair comparison of different algorithms. We propose a simple pre-processing technique to refine the input of each method and also achieve the best trade-off between performance and computation. In Chapter 3 and Chapter 4, we tackle the practical semi- and un-supervised scenarios respectively. With those unlabeled data, we apply cluster mechanism to generate pseudo-labels of them. In our works, we propose rectification methods to help reduce the noise and errors in the pseudo-labels and progressively utilize them to learn the re-ID models. In Chapter 7, we focus on the new learning framework called federated learning (FL). To prevent the sensitive data such as face images or pedestrian images from being hacked through the data transmission, we can only learn the models on multiple local devices and transmit non-private information to the central server for aggregation, such as the

doi:10.6342/NTU202202005



locally learned neural networks. In our work, we apply the FL setting on well-developed face recognition dataset and first propose a joint optimization framework for simultaneously improving the generic representation of aggregated model and the personalized representation of local models.

To further reduce the model latency on the target hardware, we also focus on model pruning. Chapter 5 focus on layer-wise filter pruning, where we can probe the sensitivity of each layer and choose the most insensitive layer to prune first. In Chapter 6, we follow the most popular global filter pruning method but propose that we should also consider the hardware constraints when estimating the importance of each filter. We propose the constraint-aware importance estimation technique to achieve the best accuracy under the same constraints of others.

Last, we build a prototype of distributed multi-target multi-camera tracking system. Our system is operated with the online setting that we can only obtain the present and previous frames of the cameras, which is practical in the real-world environment. With off-the-shelf algorithms in fast object detection, single-camera tracking, and the combination of proposed efficient video-based re-ID and constraint-aware pruning, our self-build system can promisingly achieve real-time speed (> 30 FPS) when simultaneously operating on three parallel camera streams.

In the future, first, we expect to explore the deployment of FL on person re-ID datasets, where the privacy issues are concerned most in the recent research of artificial intelligence. Then, we also want to improve the effectiveness of multi-camera tracking in our proposed system. The cross-camera matching are only based on appearance feature now, and we can further improve the matching criterion by adding the spatial constraints or temporal constrains that can be easily obtained in the multi-camera system. We hope that our prototype system can facilitate the community for quickly building an efficient and effective multi-camera system with state-of-the-art researches in computer vision.

doi:10.6342/NTU202202005

Reference

[1] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. ix, 3, 4, 53, 65, 66, 70, 89, 170

[2] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "Mars: A video benchmark for large-scale person re-identification," in *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 868–884. ix, x, xvii, xviii, 4, 15, 16, 18, 24, 25, 30, 32, 34, 35, 36, 40, 44, 45, 50, 51

[3] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: vision, hype and reality for data privacy and protection," *arXiv preprint arXiv:1907.09693*, 2019. ix, 9, 151

[4] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021. xvi, 177, 178

[5] Y. Zhang, P. Sun, Y. Jiang, D. Yu, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," *arXiv preprint arXiv:2110.06864*, 2021. xvi, 177, 178, 179

[6] X. Liu, W. Liu, T. Mei, and H. Ma, "A deep learning-based approach to progressive vehicle re-identification for urban surveillance," in *Proceedings*

doi:10.6342/NTU202202005

of European Conference on Computer Vision (ECCV). Springer, 2016, pp. 869–884. xviii, 75, 76

[7] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011. xviii, 75, 76

[8] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016. xix, xx, 7, 96, 97, 98, 116, 120, 121, 122, 125, 126, 128, 144

[9] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2008, pp. 262–275. 1, 15, 18, 170

[10] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015. 2, 6, 96, 97, 128

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017. 2, 8, 149, 152, 155, 170

[12] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1116–1124. 3, 15, 18, 26, 44, 53, 55, 63, 65, 66, 71, 89, 170

[13] G. Wang, Y. Yuan, X. Chen, J. Li, and X. Zhou, "Learning discriminative features with multiple granularities for person re-identification," in *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2018, pp. 274–282. 4, 79

doi:10.6342/NTU202202005

[14] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2138–2147. 4, 79

[15] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, "Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2018, pp. 480–496. 4, 23, 55, 79

[16] S. He, H. Luo, P. Wang, F. Wang, H. Li, and W. Jiang, "Transreid: Transformer-based object re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 013–15 022. 4

[17] C. Liu, C. Wu, Y. F. Wang, and S. Chien, "Spatially and temporally efficient non-local attention network for video-based person re-identification," in *Proceedings of British Machine Vision Conference (BMVC)*, 2019. 5, 11, 13, 15, 34, 36, 37, 44, 49, 51, 184, 187

[18] C.-T. Liu, J.-C. Chen, C.-S. Chen, and S.-Y. Chien, "Video-based person re-identification without bells and whistles," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2021, pp. 1491–1500. 5, 13, 15, 184

[19] C.-T. Liu, Y.-J. Li, S.-Y. Chien, and Y.-C. F. Wang, "Semantics-guided clustering with deep progressive learning for semi-supervised person re-identification," *arXiv preprint arXiv:2010.01148*, 2020. 5, 13

[20] C.-T. Liu, M.-Y. Lee, T.-S. Chen, and S.-Y. Chien, "Hard samples rectification for unsupervised cross-domain person re-identification," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 429–433. 6, 13

doi:10.6342/NTU202202005

[21] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5058–5066. 6, 126, 128, 144

[22] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1389–1397. 6, 125, 128

[23] Z. Zhuang, M. Tan, B. Zhuang, J. Liu, Y. Guo, Q. Wu, J. Huang, and J. Zhu, "Discrimination-aware channel pruning for deep neural networks," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2018, pp. 875–886. 6, 126, 128

[24] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *arXiv preprint arXiv:1611.06440*, 2016. 7, 126, 128, 129, 131

[25] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2736–2744. 7, 126, 129

[26] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, "Importance estimation for neural network pruning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 264–11 272. 7, 8, 117, 126, 129, 131, 140, 141, 144

[27] C.-F. Chen, G. G. Lee, V. Sritapan, and C.-Y. Lin, "Deep convolutional neural network on ios mobile devices," in *Proc. 2016 IEEE International Workshop on Signal Processing Systems (SiPS)*. IEEE, 2016, pp. 130–135. 7, 96, 98

doi:10.6342/NTU202202005

[28] C.-T. Liu, T.-W. Lin, Y.-H. Wu, Y.-S. Lin, H. Lee, Y. Tsao, and S.-Y. Chien, "Computation-performance optimization of convolutional neural networks with redundant filter removal," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 5, pp. 1908–1921, 2018. 8, 13, 108

[29] C.-T. Liu, Y.-H. Wu, Y.-S. Lin, and S.-Y. Chien, "Computation-performance optimization of convolutional neural networks with redundant kernel removal," in *IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2018, pp. 1–5. 8, 96, 97, 98, 108, 110, 111, 114, 118, 126, 128

[30] Y.-C. Wu, C.-T. Liu, B.-Y. Chen, and S.-Y. Chien, "Constraint-aware importance estimation for global filter pruning under multiple resource constraints," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2020, pp. 686–687. 8, 11, 13, 187

[31] C.-T. Liu, C.-Y. Wang, S.-Y. Chien, and S.-H. Lai, "Fedfr: Joint optimization federated framework for generic and personalized face recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 10, 13

[32] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Ieee, 2008, pp. 1–8. 10, 25, 32

[33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969. 10, 33, 95

[34] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020. 10, 33, 40, 45, 177

[35] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *In-*

doi:10.6342/NTU202202005

ternational Journal of Computer Vision, vol. 129, no. 11, pp. 3069–3087, 2021. 10, 177

[36] J. Li, X. Gao, and T. Jiang, "Graph networks for multiple object tracking," in *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020. 10

[37] C.-W. Wu, C.-T. Liu, W.-C. Tu, Y. Tsao, Y.-C. F. Wang, and S.-Y. Chien, "Space-time guided association learning for unsupervised person re-identification," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 2261–2265. 10

[38] Y. Hou, Z. Wang, S. Wang, and L. Zheng, "Adaptive affinity for associations in multi-target multi-camera tracking," *IEEE Transactions on Image Processing (TIP)*, vol. 31, pp. 612–622, 2021. 10, 171

[39] K.-S. Yang, Y.-K. Chen, T.-S. Chen, C.-T. Liu, and S.-Y. Chien, "Tracklet-refined multi-camera tracking based on balanced cross-domain re-identification for vehicles," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, June 2021, pp. 3983–3992. 10, 14, 171

[40] P. Ren, K. Lu, Y. Yang, Y. Yang, G. Sun, W. Wang, G. Wang, J. Cao, Z. Zhao, and W. Liu, "Multi-camera vehicle tracking system based on spatial-temporal filtering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021, pp. 4213–4219. 10, 171

[41] C.-T. Liu, M.-Y. Lee, C.-W. Wu, B.-Y. Chen, T.-S. Chen, Y.-T. Hsu, and S.-Y. Chien, "Supervised joint domain learning for vehicle re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2019, pp. 45–52. 14

doi:10.6342/NTU202202005

[42] T.-S. Chen, C.-T. Liu, C.-W. Wu, and S.-Y. Chien, "Orientation-aware vehicle re-identification with semantics-guided part attention network," in *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, Cham, 2020, pp. 330–346. 14

[43] T.-S. Chen, M.-Y. Lee, C.-T. Liu, and S.-Y. Chien, "Viewpoint-aware channel-wise attentive network for vehicle re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2020, pp. 574–575. 14

[44] T.-S. Chen, C.-T. Liu, and S.-Y. Chien, "Adaptive region pooling for fine-grained representation learning," 2021. 14

[45] C.-W. Wu, C.-T. Liu, C.-E. Chiang, W.-C. Tu, and S.-Y. Chien, "Vehicle re-identification with the space-time prior," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2018, pp. 121–128. 14

[46] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017. 15, 18, 21, 22, 26, 30, 44, 53, 55, 59, 66

[47] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *CVPR*, 2014. 15, 170

[48] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6036–6046. 15, 89

[49] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer gan to bridge domain gap for person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 79–88. 15, 70, 170

doi:10.6342/NTU202202005

[50] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, "Person re-identification by descriptive and discriminative classification," in *Scandinavian conference on Image analysis*. Springer, 2011, pp. 91–102. 15

[51] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identification by discriminative selection in video ranking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 12, pp. 2501–2514, 2016. 15, 170

[52] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Exploit the unknown gradually: One-shot video-based person re-identification by step-wise learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 15, 24, 25, 35, 44

[53] N. McLaughlin, J. Martinez del Rincon, and P. Miller, "Recurrent convolutional network for video-based person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1325–1334. 15, 16, 18

[54] Z. Zhou, Y. Huang, W. Wang, L. Wang, and T. Tan, "See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4747–4756. 15, 18, 30

[55] S. Li, S. Bak, P. Carr, and X. Wang, "Diversity regularized spatiotemporal attention for video-based person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 369–378. 15, 16, 18, 20, 30, 37, 48, 49

[56] D. Chen, H. Li, T. Xiao, S. Yi, and X. Wang, "Video person re-identification with competitive snippet-similarity aggregation and co-attentive snippet embedding," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1169–1178. 15, 16, 19, 30, 48, 49

doi:10.6342/NTU202202005

[57] Y. Fu, X. Wang, Y. Wei, and T. Huang, "Sta: Spatial-temporal attention for large-scale video-based person re-identification," 2019. 15, 16, 19, 28, 30, 37, 48, 49

[58] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, "Person re-identification by symmetry-driven accumulation of local features," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 2360–2367. 16

[59] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2197–2206. 16, 18, 23

[60] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 2288–2295. 16

[61] Y. Yan, B. Ni, Z. Song, C. Ma, Y. Yan, and X. Yang, "Person re-identification via recurrent feature aggregation," in *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 701–716. 16, 18

[62] L. Chen, H. Yang, J. Zhu, Q. Zhou, S. Wu, and Z. Gao, "Deep spatial-temporal fusion network for video-based person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017, pp. 63–70. 16, 23

[63] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2285–2294. 16, 18, 53

doi:10.6342/NTU202202005

[64] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, "Hierarchical bilinear pooling for fine-grained visual recognition," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2018, pp. 574–589. 16

[65] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7794–7803. 16, 20, 21, 32, 37, 42, 45

[66] T. Xiao, H. Li, W. Ouyang, and X. Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1249–1258. 18

[67] Y. Sun, L. Zheng, W. Deng, and S. Wang, "Svdnet for pedestrian retrieval," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3800–3808. 18, 53

[68] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian, "Person re-identification in the wild," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1367–1376. 18, 53, 55, 59

[69] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino, "Custom pictorial structures for re-identification." in *British Machine Vision Conference*, vol. 1, no. 2. Citeseer, 2011, p. 6. 18

[70] L. Zheng, Y. Yang, and A. G. Hauptmann, *arXiv preprint arXiv:1610.02984*, 2016. 18, 33, 44, 53, 55

[71] S. Xu, Y. Cheng, K. Gu, Y. Yang, S. Chang, and P. Zhou, "Jointly attentive spatial-temporal pooling networks for video-based person re-identification," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4733–4742. 18

doi:10.6342/NTU202202005

[72] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2017, pp. 5998–6008. 19, 37, 42

[73] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 20–36. 20

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. 21, 26, 34, 45, 66, 89, 95, 107, 116, 141, 151, 181

[75] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015. 21, 44, 59, 113, 129, 140

[76] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 212–220. 22, 162

[77] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5265–5274. 22, 151, 154, 162

[78] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang, "Bag of tricks and a strong baseline for deep person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2019, pp. 0–0. 22, 44, 55, 56, 59, 65, 66, 67, 68, 76

doi:10.6342/NTU202202005

[79] A. R. Zamir, A. Dehghan, and M. Shah, "Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs," in *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2012, pp. 343–356. 25, 32

[80] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 384–393. 30

[81] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5177–5186. 30

[82] X. Gu, H. Chang, B. Ma, H. Zhang, and X. Chen, "Appearance-preserving 3d convolution for video-based person re-identification," *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 32, 33, 34, 36, 41, 48, 49, 51

[83] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen, "Temporal complementary learning for video person re-identification," *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 32, 34, 48, 49, 51

[84] R. Hou, B. Ma, H. Chang, X. Gu, S. Shan, and X. Chen, "Vrstc: Occlusion-free video person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7183–7192. 32, 49

[85] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497. 32

doi:10.6342/NTU202202005

[86] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2015, pp. 91–99. 33

[87] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," *arXiv preprint arXiv:1909.12605*, 2019. 33, 40, 177

[88] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5533–5541. 34, 51

[89] P. Pathak, A. E. Eshratifar, and M. Gormish, "Video person re-id: Fantastic techniques and where to find them," 2019. 34, 49

[90] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, "Axial attention in multidimensional transformers," *arXiv preprint arXiv:1912.12180*, 2019. 35, 37, 42

[91] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 603–612. 35, 37

[92] Z. Zhang, C. Lan, W. Zeng, and Z. Chen, "Multi-granularity reference-aided attentive feature aggregation for video-based person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 407–10 416. 35, 37, 43, 48, 49

[93] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6299–6308. 36

doi:10.6342/NTU202202005

[94] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, 2012. 36

[95] J. Li, S. Zhang, and T. Huang, "Multi-scale 3d convolution network for video based person re-identification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8618–8625. 36, 49

[96] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 3286–3295. 37, 43, 47

[97] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. 37

[98] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," *arXiv preprint arXiv:1906.05909*, 2019. 37, 43

[99] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-deeplab: Stand-alone axial-attention for panoptic segmentation," *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 37, 43, 45, 48

[100] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 37, 43

[101] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 720–735. 38

doi:10.6342/NTU202202005

[102] W.-A. Lin, J.-C. Chen, and R. Chellappa, "A proximity-aware hierarchical clustering of faces," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 294–301. 38

[103] Y. Zhang, W. Deng, M. Wang, J. Hu, X. Li, D. Zhao, and D. Wen, "Global-local gcn: Large-scale label noise cleansing for face recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7731–7740. 38

[104] Z. Zheng, L. Zheng, and Y. Yang, "Pedestrian alignment network for large-scale person re-identification," *TCSVT*, vol. 29, no. 10, pp. 3037–3045, 2018. 41

[105] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 740–755. 45

[106] J. Si, H. Zhang, C.-G. Li, J. Kuen, X. Kong, A. C. Kot, and G. Wang, "Dual attention matching network for context-aware feature sequence based person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5363–5372. 48, 49

[107] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5177–5186. 49

[108] X. Gu, B. Ma, H. Chang, S. Shan, and X. Chen, "Temporal knowledge propagation for image-to-video person re-identification," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9647–9656. 49

doi:10.6342/NTU202202005

[109] M. M. Kalayeh, E. Basaran, M. Gökmen, M. E. Kamasak, and M. Shah, "Human semantic parsing for person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1062–1071. 53, 55

[110] X. Zhang, H. Luo, X. Fan, W. Xiang, Y. Sun, Q. Xiao, W. Jiang, C. Zhang, and J. Sun, "Alignedreid: Surpassing human-level performance in person re-identification," *arXiv*, 2017. 53

[111] J. Li, A. J. Ma, and P. C. Yuen, "Semi-supervised region metric learning for person re-identification," *IJCV*, vol. 126, no. 8, pp. 855–874, 2018. 53, 56

[112] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Bian, and Y. Yang, "Progressive learning for person re-identification with one example," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2872–2881, 2019. 53, 56, 57, 63, 67, 68

[113] X. Xin, J. Wang, R. Xie, S. Zhou, W. Huang, and N. Zheng, "Semi-supervised person re-identification using multi-view clustering," *Pattern Recognition*, vol. 88, pp. 285–297, 2019. 53, 54, 58, 66, 67, 68

[114] X. Xin, X. Wu, Y. Wang, and J. Wang, "Deep self-paced learning for semi-supervised person re-identification using multi-view self-paced clustering," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2631–2635. 53, 54, 57, 58, 63, 66, 67, 68, 72

[115] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007. 54, 60, 72

[116] H. Fan, L. Zheng, C. Yan, and Y. Yang, "Unsupervised person re-identification: Clustering and fine-tuning," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 4, p. 83, 2018. 54, 57, 63, 68, 79, 83, 90

doi:10.6342/NTU202202005

[117] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 55

[118] Y. Huang, J. Xu, Q. Wu, Z. Zheng, Z. Zhang, and J. Zhang, "Multi-pseudo regularized label for generated data in person re-identification," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1391–1403, 2018. 56

[119] G. Ding, S. Zhang, S. Khan, Z. Tang, J. Zhang, and F. Porikli, "Feature affinity based pseudo labeling for semi-supervised person re-identification," *IEEE Transactions on Multimedia*, 2019. 56

[120] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680. 56

[121] Z. Liu, D. Wang, and H. Lu, "Stepwise metric promotion for unsupervised video person re-identification," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2429–2438. 56

[122] S. Bak and P. Carr, "One-shot metric learning for person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2990–2999. 56

[123] I. Givoni and B. Frey, "Semi-supervised affinity propagation with instance-level constraints," in *Artificial Intelligence and Statistics*, 2009, pp. 161–168. 56

[124] X. Zhu and Z. Ghahraman, "Learning from labeled and unlabeled data with label propagation," 2002. 56

doi:10.6342/NTU202202005

[125] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2010, pp. 1189–1197. 57

[126] L. Jiang, D. Meng, S.-I. Yu, Z. Lan, S. Shan, and A. Hauptmann, "Self-paced learning with diversity," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2014, pp. 2078–2086. 57

[127] F. Ma, D. Meng, Q. Xie, Z. Li, and X. Dong, "Self-paced co-training," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2275–2284. 57

[128] Y. Lin, X. Dong, L. Zheng, Y. Yan, and Y. Yang, "A bottom-up clustering approach to unsupervised person re-identification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8738–8745. 57, 67, 68

[129] Y. Ding, H. Fan, M. Xu, and Y. Yang, "Adaptive exploration for unsupervised person re-identification," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 1, pp. 1–19, 2020. 57, 67, 68

[130] H.-X. Yu, W.-S. Zheng, A. Wu, X. Guo, S. Gong, and J.-H. Lai, "Unsupervised person re-identification by soft multilabel learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2148–2157. 57, 67, 68, 82, 90

[131] Y. Fu, Y. Wei, G. Wang, Y. Zhou, H. Shi, and T. S. Huang, "Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 6112–6121. 57, 67, 68, 72, 79, 83, 90, 91

doi:10.6342/NTU202202005

[132] Z. Zhong, L. Zheng, S. Li, and Y. Yang, "Generalizing a person retrieval model hetero- and homogeneously," in *Proceedings of European Conference on Computer Vision (ECCV)*, September 2018. 59, 82, 90

[133] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." 60, 72, 83

[134] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 17–35. 66, 180

[135] Y. Ge, D. Chen, and H. Li, "Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=rJlnOhVYPS 67, 68

[136] P. Peng, T. Xiang, Y. Wang, M. Pontil, S. Gong, T. Huang, and Y. Tian, "Unsupervised cross-dataset transfer learning for person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1306–1315. 67, 68, 82

[137] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, "Camera style adaptation for person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5157–5166. 68

[138] X. Liu, S. Zhang, Q. Huang, and W. Gao, "Ram: a region-aware deep model for vehicle re-identification," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2018, pp. 1–6. 75

[139] X. Liu, S. Zhang, X. Wang, R. Hong, and Q. Tian, "Group-group loss-based global-regional feature learning for vehicle re-identification," *IEEE Transactions on Image Processing (TIP)*, vol. 29, pp. 2638–2652, 2019. 75

doi:10.6342/NTU202202005

[140] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge university press, 2008. 75

[141] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 360–368. 75, 76

[142] B. Harwood, B. Kumar, G. Carneiro, I. Reid, T. Drummond *et al.*, "Smart mining for deep metric learning," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2821–2829. 75, 76

[143] H.-X. Yu, A. Wu, and W.-S. Zheng, "Cross-view asymmetric metric learning for unsupervised person re-identification," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 994–1002. 79, 83, 90

[144] X. Jin, C. Lan, W. Zeng, and Z. Chen, "Global distance-distributions separation for unsupervised person re-identification," *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 79, 90

[145] W. Deng, L. Zheng, G. Kang, Y. Yang, Q. Ye, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 82, 90

[146] X. Zhang, J. Cao, C. Shen, and M. You, "Self-training with progressive augmentation for unsupervised cross-domain person re-identification," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 8222–8231. 83, 86, 90, 91

[147] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W. T. Freeman, "Best-buddies similarity for robust template matching," in *Proceedings of IEEE*

doi:10.6342/NTU202202005

Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2021–2029. 86

[148] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987. 87

[149] G. Wang, J.-H. Lai, W. Liang, and G. Wang, "Smoothing adversarial domain attack and p-memory reconsolidation for cross-domain person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 568–10 577. 90

[150] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008. 93

[151] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007, pp. 410–420. 94

[152] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 95, 120, 140, 143

[153] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. 95, 96, 107, 142

[154] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440. 95

doi:10.6342/NTU202202005

[155] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1646–1654. 95, 107

[156] Y. LeCun, J. S. Denker, S. A. Solla, R. E. Howard, and L. D. Jackel, "Optimal brain damage." in *Proceedings of Neural Information Processing Systems (NIPS)*, vol. 2, 1989, pp. 598–605. 97

[157] B. Hassibi, D. G. Stork *et al.*, "Second order derivatives for network pruning: Optimal brain surgeon," *Proceedings of Neural Information Processing Systems (NIPS)*, pp. 164–164, 1993. 97

[158] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," *arXiv preprint arXiv:1405.3866*, 2014. 98

[159] V. Lebedev and V. Lempitsky, "Fast convnets using group-wise brain damage," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2554–2564. 98

[160] S. Anwar, K. Hwang, and W. Sung, "Structured pruning of deep convolutional neural networks," *Proc. ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 13, no. 3, p. 32, 2017. 98

[161] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, Nov 1998. 102

[162] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing (TIP)*, vol. 9, no. 7, pp. 1158–1170, Jul 2000. 102

doi:10.6342/NTU202202005

[163] S. Anwar, K. Hwang, and W. Sung, "Fixed point optimization of deep convolutional neural networks for object recognition," in *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 1131–1135. 103

[164] Wikipedia contributors, "Binary search algorithm — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Binary_search_algorithm&oldid=838321018, 2018, [Online; accessed 5-May-2018]. 104

[165] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017. 107

[166] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)." [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html 107, 140, 143

[167] ARM-Software, "Systolic cnn accelerator simulator (scale sim)," https://github.com/ARM-software/SCALE-Sim, 2018. 108

[168] J. K. Lee, "A caffe-based implementation of very deep convolution network for image super-resolution," https://github.com/huangzehao/caffe-vdsr, 2016. 109

[169] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," 2012. 109

[170] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. International conference on curves and surfaces*. Springer, 2010, pp. 711–730. 109

doi:10.6342/NTU202202005

[171] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017. 121

[172] Z. You, K. Yan, J. Ye, M. Ma, and P. Wang, "Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2019, pp. 2130–2141. 126, 129, 131, 141, 144

[173] Y. He, G. Kang, X. Dong, Y. Fu, and Y. Yang, "Soft filter pruning for accelerating deep convolutional neural networks," *arXiv preprint arXiv:1808.06866*, 2018. 128

[174] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4340–4349. 128, 144

[175] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. 128

[176] R. Yu, A. Li, C.-F. Chen, J.-H. Lai, V. I. Morariu, X. Han, M. Gao, C.-Y. Lin, and L. S. Davis, "Nisp: Pruning networks using neuron importance score propagation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9194–9203. 129, 144

[177] J. Ye, X. Lu, Z. Lin, and J. Z. Wang, "Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers," *arXiv preprint arXiv:1802.00124*, 2018. 129

[178] T.-W. Chin, C. Zhang, and D. Marculescu, "Layer-compensated pruning for resource-constrained convolutional neural networks," *arXiv preprint arXiv:1810.00518*, 2018. 129

doi:10.6342/NTU202202005

[179] A. Gordon, E. Eban, O. Nachum, B. Chen, H. Wu, T.-J. Yang, and E. Choi, "Morphnet: Fast & simple resource-constrained structure learning of deep networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1586–1595. 129

[180] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520. 141

[181] X. Ding, G. Ding, Y. Guo, and J. Han, "Centripetal sgd for pruning very deep convolutional networks with complicated structure," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4943–4953. 141

[182] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019. 149, 151

[183] D. Aggarwal, J. Zhou, and A. K. Jain, "Fedface: Collaborative learning of face recognition model," *arXiv preprint arXiv:2104.03008*, 2021. 149, 152, 165

[184] F. Yu, A. S. Rawat, A. Menon, and S. Kumar, "Federated learning with only positive labels," in *ICML*, 2020. 149, 152

[185] C. N. Duong, T.-D. Truong, K. Luu, K. G. Quach, H. Bui, and K. Roy, "Vec2face: Unveil human faces from their blackbox features in face recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 149

doi:10.6342/NTU202202005

[186] Q. Meng, F. Zhou, H. Ren, T. Feng, G. Liu, and Y. Lin, "Improving federated learning face recognition via privacy-agnostic clusters," *arXiv preprint arXiv:2201.12467*, 2022. 149

[187] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2020. 150, 152

[188] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney *et al.*, "Iarpa janus benchmark-c: Face dataset and protocol," in *International Conference on Biometrics (ICB)*, 2018. 150, 161

[189] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *ECCV*, 2016. 150, 151, 161

[190] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, 2018. 151

[191] Z. Zhu, G. Huang, J. Deng, Y. Ye, J. Huang, X. Chen, J. Zhu, T. Yang, J. Lu, D. Du *et al.*, "Webface260m: A benchmark unveiling the power of million-scale deep face recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 492–10 502. 151

[192] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015. 151

[193] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *CVPR*, 2019. 151, 162

doi:10.6342/NTU202202005

[194] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *CVPR*, 2020, pp. 6398–6407. 151

[195] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data *et al.*, "A field guide to federated optimization," *arXiv preprint arXiv:2107.06917*, 2021. 151

[196] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," *arXiv preprint arXiv:1910.14425*, 2019. 152

[197] A. Khaled, K. Mishchenko, and P. Richtárik, "Tighter theory for local sgd on identical and heterogeneous data," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020. 152

[198] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019. 152

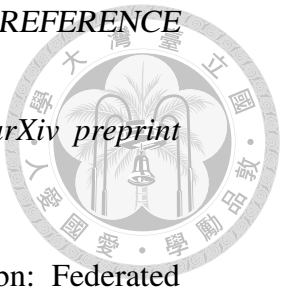
[199] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016. 152

[200] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019. 152

[201] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *CVPR*, 2021. 152, 159

[202] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Fed-

doi:10.6342/NTU202202005



erated learning with local and global representations," *arXiv preprint arXiv:2001.01523*, 2020. 152

[203] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," *arXiv preprint arXiv:2102.07623*, 2021. 152

[204] H.-Y. Chen and W.-L. Chao, "On bridging generic and personalized federated learning," *arXiv preprint arXiv:2107.00778*, 2021. 152

[205] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2017. 152

[206] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," *arXiv preprint arXiv:1802.07876*, 2018. 152

[207] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," *arXiv preprint arXiv:2002.07948*, 2020. 152

[208] T. Yu, E. Bagdasaryan, and V. Shmatikov, "Salvaging federated learning by local adaptation," *arXiv preprint arXiv:2002.04758*, 2020. 152, 166

[209] W. Zhuang, Y. Wen, X. Zhang, X. Gan, D. Yin, D. Zhou, S. Zhang, and S. Yi, "Performance optimization of federated person re-identification via benchmark analysis," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020. 155

[210] C. Li, D. Niu, B. Jiang, X. Zuo, and J. Yang, "Meta-har: Federated representation learning for human activity recognition," *arXiv preprint arXiv:2106.00615*, 2021. 155

[211] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018. 157

doi:10.6342/NTU202202005

[212] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *arXiv preprint arXiv:2006.07242*, 2020. 157

[213] C.-Y. Wang, Y.-L. Chang, S.-T. Yang, D. Chen, and S.-H. Lai, "Unified representation learning for cross model compatibility," *arXiv preprint arXiv:2008.04821*, 2020. 159

[214] Y. Wen, W. Liu, A. Weller, B. Raj, and R. Singh, "Sphereface2: Binary classification is all you need for deep face recognition," *arXiv preprint arXiv:2108.01513*, 2021. 159

[215] G. Wu and S. Gong, "Decentralised learning from independent multi-domain labels for person re-identification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, 2021, pp. 2898–2906. 170

[216] C. C. Loy, C. Liu, and S. Gong, "Person re-identification by manifold ranking," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 3567–3571. 170

[217] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. 174

[218] M. Grinberg, *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.", 2018. 175

[219] N. TensorRT. [Online]. Available: Available:https://developer.nvidia.com/tensorrt/ 176, 180

[220] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3645–3649. 177

[221] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018. 177

doi:10.6342/NTU202202005

[222] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019. 177

[223] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *Proceedings of IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6. 177

[224] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomammana, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Yu, changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4154370 178

[225] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9627–9636. 178

[226] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942 [cs]*, Apr. 2015, arXiv: 1504.01942. [Online]. Available: http://arxiv.org/abs/1504.01942 179

[227] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *arXiv:1603.00831 [cs]*, Mar. 2016, arXiv: 1603.00831. [Online]. Available: http://arxiv.org/abs/1603.00831 179

[228] [Online]. Available: http://tdr.lib.ntu.edu.tw/jspui/handle/123456789/71450 186

doi:10.6342/NTU202202005