Multi-Target Domain Adaptation for Person Re-Identification

by

Félix REMIGEREAU

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT OF A MASTER'S DEGREE
WITH THESIS IN ELECTRICAL ENGINEERING
M.A.Sc.

MONTREAL, AUGUST 8, 2023

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

M. Eric Granger, Memorandum supervisor
Département de génie des systèmes, École de Technologie Supérieure

M. Rafael Menelau-Cruz, co-Supervisor
Département de génie logiciel et des TI, École de Technologie Supérieure

M. Mohammadhadi Shateri, President of the board of examiners
Département de génie des systèmes, École de Technologie Supérieure

M. Marco Pedersoli, External examiner
Département de génie des systèmes, École de Technologie Supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON JULY 10, 2023

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

I would like to thank my advisor Eric Granger for his time and support during the realization of this work.

I would also like to thank Rafael Menelau-Cruz for joining as co-advisor and helping me complete this work.

Thank you to all my colleagues from the LIVIA, especially Djebril Mekhazni, Sajjad Abdoli, Le Thanh Nguyen-Meidine, Arthur Josi, Mahdi Alehdaghi and Madhu Kiran.

Adaptation de domaine à cible multiple pour la réidentification de personnes

Félix REMIGEREAU

RÉSUMÉ

La ré-identification de personnes (ReID) consiste à faire correspondre des images de piétons capturées par un réseau de caméras multiples qui ne partagent pas les conditions de capture et dont les champs de vision ne s'entrecroisent pas. À partir d'une image de requête, dites "query", capturée pour un individu, le système doit automatiquement trouver toutes les autres images du même individu dans une galerie d'images de piétons. Plusieurs facteurs rendent cette tâche difficile, tels que la différence entre les points de vue des caméras, les occlusions visuelles, la pose du piéton et les variations d'éclairage de la scène. Ce domaine de recherche a des applications importantes dans l'automatisation de la surveillance vidéo et de la biométrie. Les progrès récents en matière d'apprentissage profond et la disponibilité de grandes bases de données annotées ont permis aux systèmes modernes de s'entraîner efficacement et d'atteindre un niveau de précision très élevé.

Si ces solutions fonctionnent bien avec suffisamment de données d'entraînement capturées pour un réseau de caméras donné, également connu sous le nom de "domaine", les performances chutent de manière significative lors du traitement d'images provenant d'un autre domaine opérationnel. Dans l'application d'un système ReID, nous considérons deux types de données : les données sources provenant d'un environnement contrôlé et étiqueté, et les données cibles, généralement non étiquetées, provenant du domaine opérationnel dans lequel le système est déployé. En effet, les différents domaines divergent en raison de leurs caractéristiques différentes, telles que la position de la caméra, la résolution de l'image, l'éclairage et l'arrière-plan, pour n'en citer que quelques-unes. Étant donné que le coût de génération d'un nouvel ensemble de données étiquetées pour chaque nouveau domaine opérationnel est très élevé, des techniques d'adaptation de domaine non supervisée (UDA) ont été développées pour exploiter un ensemble de données source étiqueté et des ensembles de données cibles non étiquetés afin de maximiser la performance sur les données opérationnelles. Cependant, un autre problème se pose lorsque plusieurs domaines cibles sont présents. Chaque domaine cible diverge dans ses caractéristiques spécifiques et nécessite une adaptation unique. L'adaptation d'un modèle ReID personnalisé pour chaque domaine cible est une solution simple mais peu pratique dans les applications réelles où les ressources informatiques sont limitées. D'autre part, l'apprentissage d'un modèle unique sur toutes les cibles simultanément peut réduire la précision en raison de la capacité insuffisante du modèle, lorsqu'il s'agit de données très variées. La solution optimale doit donc offrir une grande précision sur chaque cible tout en minimisant la complexité de la mémoire du modèle résultant.

Très peu de recherches s'intéressent aux méthodes d'adaptation de domaines non supervisée multi-cibles (MTDA) pour la ReID. Ces méthodes sont souvent peu précises et ne prennent pas en compte la complexité de la solution lors de l'évaluation. Le succès d'un système dépend des données utilisées, de la capacité du CNN et de la méthode MTDA employée. Il est donc essentiel

de prendre en compte la complexité du CNN entraîné ainsi que les ensembles de données utilisés lors de l'évaluation d'une méthode MTDA.

Ce travail aborde le problème du MTDA pour l'identification des personnes à l'aide de la distillation des connaissances (KD). L'objectif est d'entraîner un modèle CNN compact capable de faire correspondre deux images de piétons capturées par des caméras différentes. Le modèle de Deep Learning (DL) sera capable d'effectuer cette tâche sur des images provenant de plusieurs domaines cibles. Nous évaluons la solution en fonction de (1) la précision du modèle lorsqu'il est entraîné pour de nombreux domaines cibles, et (2) le temps d'inférence et le nombre de paramètres du modèle au moment de l'inférence. Dans cette thèse, nous présentons deux contributions principales liées à ces critères.

La première contribution, présentée au Chapitre 4, est le développement d'une technique MTDA basée sur KD pour la ReID de personnes, intitulée KD-ReID. À l'aide d'une fonction de coût spécialement conçue pour la ReID, nous adaptons un ensemble de modèles CNN "teacher", chacun à un domaine cible spécifique, puis nous distillons les connaissances dans un seul modèle CNN "student". Le modèle résultant est précis pour tous les domaines cibles tout en restant peu coûteux pour une application réelle. Nous montrons que cette approche est plus performante que les approches de pointe existantes en termes de précision et de complexité du modèle. En outre, nous montrons que KD-ReID est très flexible, permettant d'utiliser des modèles d'enseignants d'architecture et de techniques d'apprentissage différentes. Cette flexibilité renforce le potentiel de KD-ReID à être utilisé dans des applications réelles. Cette contribution a été acceptée et publiée dans l'IEEE International Conference on Image Processing 2022 (ICIP2022).

Notre deuxième contribution, présentée au Chapitre 5, est une étude comparative complète des techniques KD-MTDA, afin de maximiser le rapport entre la complexité du système et la précision pour la tâche ReID. À l'aide d'un modèle compact, nous adaptons diverses techniques à notre problème MTDA. Plus précisément, nous analysons et comparons quatre techniques : les couches de BN spécifiques à un domaine, le modèle multi-branches, les adaptateurs de domaine et les adaptateurs de distillation. Les techniques étudiées visent à augmenter la précision sur des cibles multiples au prix d'une augmentation du nombre de paramètres du modèle. En plus de permettre l'optimisation de notre solution dans une situation de ressources limitées, cette étude nous permet de tirer des conclusions importantes sur le compromis entre la précision des méthodes MTDA spécialisées pour notre tâche.

Mots-clés: Video-surveillance, Distillation de connaissance, Adaptation de domaine à cible multiple, Ré-identification de personnes, Adapteur de domaines, Normalisation de Batch

Multi-Target Domain Adaptation for Person Re-Identification

Félix REMIGEREAU

ABSTRACT

Re-identification of individuals (ReID) involves matching images of pedestrians captured by a network of multiple cameras that do not capture conditions or fields of view. Given a query image captured for an individual, the system must automatically find all other images of the same individual in a pedestrian image gallery. Several factors make this task difficult, such as the difference in camera viewpoints, visual occlusions, pose, and illumination variations over time. This research area has important applications in the automation of video surveillance monitoring and biometrics. Recent advances in deep learning and the availability of large annotated databases have enabled modern systems to train efficiently and achieve very high accuracy.

Although these solutions work well with sufficient training data captured for a given camera network, also known as a "domain", performance drops significantly when processing images from another operational domain. In the application of a ReID system, we consider two types of data: source data coming from a controlled environment that is labeled, and target data, generally unlabeled, coming from the operational domain in which the system is deployed. Indeed, the different domains diverge because of their different characteristics, such as camera position, image resolution, lighting, and background, to name a few. Since the cost of generating a new labeled dataset for each new operational domain is very high, unsupervised domain adaptation (UDA) techniques have been developed to leverage data from a labeled source dataset and an unlabeled target dataset to improve performance on the operational data. However, another problem arises when multiple target domains are present. Each target domain diverges in its specific characteristics and requires a unique adaptation. Adapting a customized ReID model for each target domain is a simple but impractical solution in real applications where computing resources are limited. On the other hand, training a single model on all targets simultaneously can reduce accuracy due to insufficient model capacity, when dealing with highly varied data. The optimal solution must therefore have high accuracy on each target while minimizing the memory complexity of the resulting model.

Very few works tackle multi-target unsupervised domain adaptation (MTDA) methods for person ReID. These methods are often not accurate and do not consider the complexity of the solution during evaluation. The success of a system depends on the data used, the capacity of the CNN, and the MTDA method employed. Therefore, it is essential to consider the complexity of the trained CNN as well as the datasets used when evaluating an MTDA method.

This work tackles the MTDA problem for person ReID using Knowledge Distillation (KD). The objective is to train a compact CNN model capable of matching two pedestrian images captured by different cameras. The Deep Learning (DL) model will be able to perform this task on images from several target domains. We evaluate the solution according to (1) the accuracy

of the model when trained for many target domains, and (2) the time and memory complexity during inference. In this dissertation, we present two main contributions related to these criteria.

The first contribution, presented in Chapter 4, is the development of a knowledge distillation-based MTDA technique for person ReID entitled KD-ReID. Using a cost function tailored specifically for ReID, we adapt a set of "Teacher" CNN models, each one to a specific target domain, and then distill the knowledge to a single "Student" CNN model. The resulting model is accurate for all target domains while remaining low cost for a real-world application. We show that this approach outperforms existing state-of-the-art approaches in terms of accuracy and model complexity. Furthermore, we show that KD-ReID is very flexible, allowing us to use Teacher models of different architecture and training techniques. This flexibility reinforces the potential of KD-ReID to be used in real applications. This contribution has been accepted and published in the IEEE International Conference on Image Processing 2022 (ICIP2022).

The second contribution, presented in Chapter 5, is a comprehensive comparative study of KD-MTDA techniques, to maximize the relationship between system complexity and accuracy for the ReID task. Using a compact model, we adapt various techniques to our MTDA problem. More precisely, we analyze and compare four techniques: domain-specific BN layers, multi-branch model, domain adapters, and distillation adapters. The techniques studied aim to increase accuracy on multiple targets at the cost of increasing the number of model parameters. In addition to allowing optimization of our solution in a resource-constrained situation, this study allows us to draw important conclusions on the trade-off between the accuracy of specialized MTDA methods for our task.

Keywords: Video-surveillance, Knowledge Distillation, Multi-target Domain Adaptation, Person Re-identification, Domain Adapters, Batch Normalization

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

CMC	Cumulative Matching Characteristics
CNN	Convolutional Neural Network
DL	Deep Learning
DPM	Deformable Part Model
FLOPs	Floating Point operations
FLOPS	Floating Point operations per second
GAN	Generative Adversarial Network
KD	Knowledge Distillation
mAP	Mean Average Precision
ML	Machine Learning
MSDA	Multi-Source Domain Adaptation
MTDA	Multi-Target Domain Adaptation
ReID	Re-identification
SOTA	State-of-the-Art
STDA	Single-Target Domain Adaptation
UDA	Unsupervised Domain Adaptation
VS	Video Surveillance

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

x	Input sample
x^t	Input sample from target domain t
x^s	Input sample from the source domain
y	Output embedding
w	Neural network weight
b	Neural network bias
σ	Non-linear activation function
$*$	Convolution operation
$\mathcal{L}$	Loss function
Θ	Student model
$\mathbf{\Phi}$	Set of all Teacher models
Φ^t	Teacher model associated with target t
α	Adapter bank of 1x1 filters
β	Adapter FC layer

INTRODUCTION

Context

Public safety is an ever-growing concern in society, and video surveillance (VS) systems are more and more common. A video surveillance system is a set of cameras with non-overlapping fields of vision used to survey a public area such as an airport or a bank. These systems capture a vast amount of video data, which is then stored and analyzed for security purposes. The task of analyzing the captured images is traditionally assigned to human operators. Given the nature of the task, and the high volume of data needing analysis, human operators are prone to error. This prompts the need for an automatic surveillance system that could provide decision support to human operators, freeing them to perform more difficult tasks.

An automatic security system aims to match images of individuals captured by different cameras at different times and locations (Gong, Cristani, Loy & Hospedales, 2014). This allows tracking the whereabouts of a person as they appear over a distributed network of cameras. An automatic security system works in three steps: raw data collection, pedestrian localization, and person re-identification (ReID), as illustrated by Figure 0.1. The raw data collection is simply the accumulation of footage collected by a network of cameras. The pedestrian localization aims to generate bounding boxes around pedestrians in every captured frame from the first part. Finally, the system must match pedestrians based on their identity. Images are continuously stored in a database called a gallery. Given a sample (query) image of a specific identity, the system must identify all occurrences of images belonging to that same identity from the gallery. This final step is what is called person ReID.

While pedestrian detection (Brunetti, Buongiorno, Trotta & Bevilacqua, 2018) and person ReID (Ye, Shen, Lin, Xiang, Shao & Hoi, 2021) are dependent tasks in an automatic surveillance system, they are considered two distinct fields of research in computer vision due to their

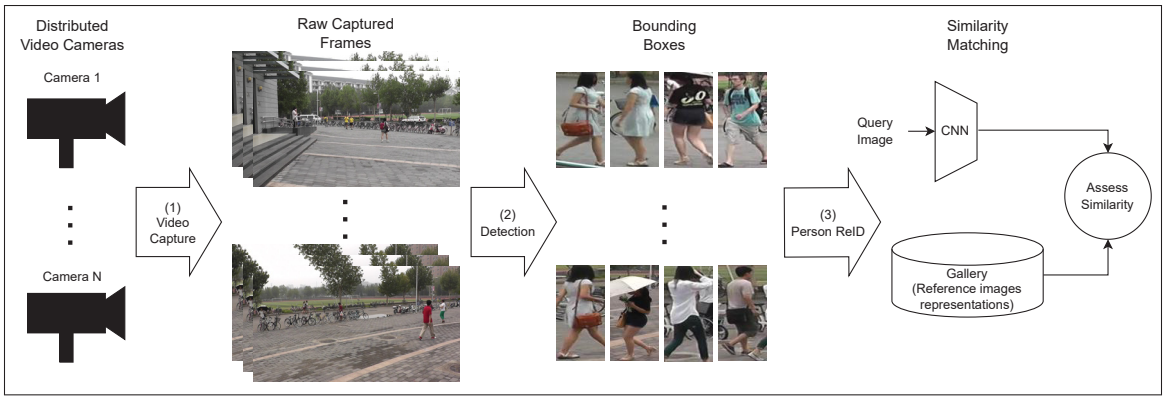

Figure 0.1 Overview of an automated video ReID system. The three main steps illustrated are (1) video capture, where cameras capture frames; (2) pedestrian detection, where the system generates bounding boxes around pedestrians; and (3) person ReID where bounding boxes are transformed into a lower dimension representation and stored in a gallery. Given a new query image, the system computes a similarity score between every stored image representation in the gallery and ranks them based on similarity

individual complexities. To this effect, experimental datasets used to evaluate person ReID methods suppose that a pedestrian detector/tracker has already extracted bounding boxes from the raw data. This project focuses only on the person ReID part of the system.

Person ReID is challenging as images captured by a security network present important variations. Different cameras composing the security system introduce variations of pedestrian pose (Sarfraz, Schumann, Eberle & Stiefelhagen, 2018b), background clutter (Song, Huang, Ouyang & Wang, 2018), and occlusion (Hou, Ma, Chang, Gu, Shan & Chen, 2019a). Variations in capture conditions due to weather and time of day create illumination variations increasing the difficulty of ReID (Huang, Zha, Fu & Zhang, 2019b). Additionally, pedestrian detectors are imperfect and can produce inaccurate bounding boxes. The type of camera used also influences the ReID due to variations in image resolution and scale. Further complications arise when considering infrared cameras, but this study limits its application to visible RGB images captured with cameras. Each security system and its cameras produce images with specific characteristics and can be defined as domains. The divergence or shift between domains reduces matching accuracy

when considering images coming from multiple domains. For this reason, adaptation to specific target domains is important to ReID systems' accuracy. In this thesis, we focus on the case where source and target domains are defined by datasets with images captured over multiple cameras.

In a realistic person ReID scenario, identities present in the training data are not present in the testing data; the model must work well on data from unknown persons. Therefore, this problem cannot be implemented as a classification system where each identity is a separate class. Rather it is modeled as a pairwise matching problem using a DL network, as shown in Figure 0.2. In this figure, the convolutional network attempts to learn an alternate representation space through metric learning where images of the same identity are closer together while images of different identities are far apart. The term embedding refers to images transformed into a feature representation by a backbone CNN. Once the model is trained, it builds a gallery of computed embeddings for a feed of pedestrian images. When the model receives a query image, it transforms it into the embedding space. The distance between embeddings produced for the query image and the gallery images is then measured using a distance or similarity metric, such as Euclidean or cosine distance, or a correlation measure. Finally, images stored in the gallery are ranked based on how close their embedding is to the query embedding. A successful model will produce a ranking where images of the same identity are higher in the ranked list than images of a different identity.

Problem Statement and Objectives

Recent research (Gong, 2021; Wieczorek, Rychalska & Dąbrowski, 2021; Wang, Lai, Huang & Xie, 2019a) has obtained excellent results on challenging ReID datasets in the supervised scenario. However, DL must be adapted for applications and deployments using an unlabeled target dataset. Source labeled datasets are very difficult to obtain due to the high costs of manpower, time to annotate, and they require ethical approval. Unsupervised Domain Adaptation (UDA) (Long, Cao, Wang & Jordan, 2015) is a branch of transfer learning which applies well to person

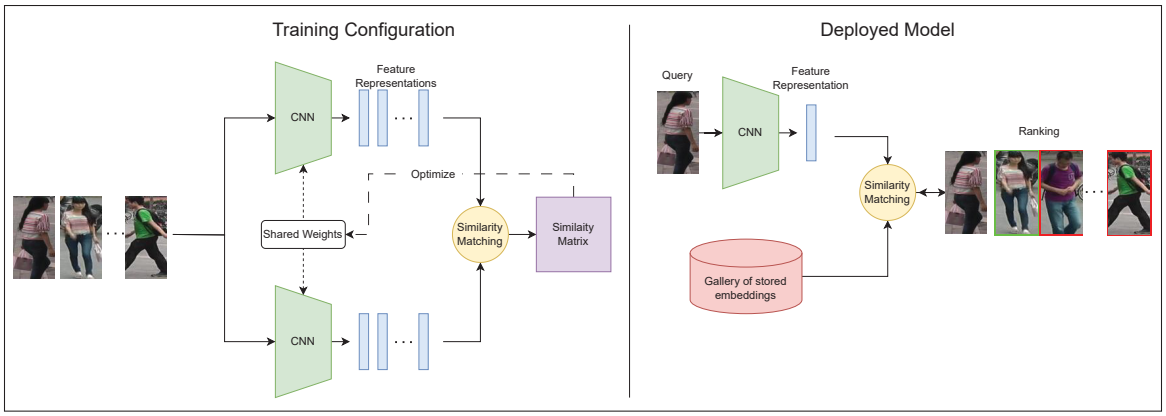

Figure 0.2 DL network used to perform person ReID in training and deployment configurations. In the training configuration, the model builds a similarity matrix from all images and uses it to optimize the CNN weights. In deployment, the model is used to compute the similarity between the feature representation of a query image and stored feature representations from the gallery

ReID. The idea of UDA is to use some labeled data (source dataset) to train and to adapt the resulting model to unlabelled data (target dataset). With this idea, Using UDA, many techniques (Bertocco, Andaló & Rocha, 2021; Wang, Zhao, Liao & Shao, 2022; Sheng, Li, Zheng, Liang, Dong, Huang, Ji & Sun, 2021) were developed for person ReID to bridge the gap in performance between labeled and unlabeled data. With a single labeled source dataset, these techniques can adapt to any new dataset without the need to annotate it.

While these techniques work well when having one source dataset and a single target dataset, in a scenario known as Single Target Domain Adaptation (STDA), a problem arises for multiple target domains. In this thesis, each dataset is referred to as a domain, where data is captured in different capture conditions. They present an important domain shift between them due to various capture conditions (eg. camera position, illumination, background environments, person pose, occlusion, resolution/scale). Training a distinct model for each new domain can become impractical, especially as the number of target domains grows large. DL models require large computational resources to be stored in memory. In ReID applications with limited computational resources minimizing the time and memory complexity of the system becomes

important. Time complexity refers to how fast the model operates while memory complexity refers to how cumbersome the model is to store. While inference time is often considered a more important characteristic to minimize, we argue that model memory complexity is also significant. During inference, we must consider the time required to access a large number of parameters in a hardware memory device. Faster memory devices are generally more expensive and limited in a system compared to slower large memory devices. Therefore minimizing the memory complexity of our model reduces the cost of the deployed system by reducing the need for larger fast memory devices. Simply training a single model with all target domain data combined solves the memory complexity issue but is very challenging as the large domain shifts between datasets impact the accuracy negatively. Current MTDA techniques in person ReID (Mohanty, Banerjee & Velmurugan, 2022; Tian, Tang, Li, Teng, Zhang & Fan, 2021; Wu, Zheng, Guo & Lai, 2019b) have relatively low accuracy and do not consider the model complexity when evaluating their solution. These techniques attempt to directly tackle the adaptation to all target domain data combined, which is an extremely difficult task. This research proposes to break the problem into simple STDA problems and combine the results at a later stage. We argue that model complexity is crucial if the solution aims to be used in real-world VS scenarios.

This research aims to develop Knowledge Distillation (KD) based MTDA methods adapted to the problem of person ReID. We seek to train models that perform well on multiple target domains while being compact enough for real-world applications. To this end, the size of the final model should scale efficiently to the number of target domains considered, as it must apply to situations with large numbers of target domains. Finding a solution with a good trade-off between accuracy and model complexity is essential. KD has proven to be a very cost-effective approach due to its model compression properties (Hinton, Vinyals, Dean et al., 2015).

Main Contributions

This thesis is composed of two main contributions: a new MTDA technique to adapt DL models for person ReID based on KD, which we name KD-ReID and a comparative study of extensions to the proposed KD-ReID approach. Our technique, presented in detail in Chapter 4, uses an existing single-target UDA technique; it trains an individual large teacher model on each target domain separately. These teacher models are then combined into a single smaller student model through KD. This technique yields higher results than other approaches in literature while also producing one compact model, making it more suitable for real-world applications. Furthermore, we show that the technique works independently of the existing UDA technique used to train individual teachers. The technique's versatility means it can easily be implemented and integrated with existing UDA methods, so will keep improving as new UDA techniques are developed.

Related Publication: Remigereau, F., Mekhazni, D., Abdoli, S., Nguyen-Meidine, L. T., Cruz, R. M. O. & Granger,E. (2022). Knowledge Distillation for Multi-Target Domain Adaptation in Real-Time Person Re-Identification. 2022 IEEE International Conference on Image Processing (ICIP), pp. 3853-3557

Second, this thesis provides a comparative study of various architectural modifications that can further improve KD-ReID. These modifications were analyzed to increase ReID accuracy of the student at the cost of increased complexity. The four modifications are domain-specific BN layers, multi-branch models, domain adapters, and distillation adapters. This study identifies a cost-effective model configuration that balances complexity with accuracy. Through experimental analysis with these modifications, this thesis draws conclusions that help understand the trade-off between accuracy and complexity for MTDA for person ReID.

Structure of the thesis

This thesis is divided into 5 chapters. Chapter 1 presents a brief overview of key deep learning concepts necessary to understanding the rest of this thesis. Chapter 2 presents a DL review of the State-of-the-art (SOTA) literature in the field of person ReID and related methods for unsupervised STDA and MTDA, as well as KD techniques. Chapter 3 details the experimental methodology used to validate the methods for MTDA. In Chapter 4 the proposed KD-ReID approach is presented in detail. A comprehensive study of how the method behaves in various MTDA scenarios shows how KD-ReID produces a compact DL model which performs well on multiple target domains simultaneously. A comparison to other MTDA techniques in literature in terms of complexity and accuracy is provided as well. Additionally, the versatility of KD-ReID is demonstrated by testing it using two base STDA techniques, which function very differently. We show that our technique works to extend any person ReID technique and can even combine different training approaches to optimize accuracy. Finally, in Chapter 5, a study and comparison of architectural modifications that can efficiently improve MTDA accuracy are provided. By adapting four techniques to ReID, we determine the optimal model configuration and further improve results from the approach presented in Chapter 4 while keeping the model complexity at a minimum. The techniques studied are: domain-specific BN layers, multi-branch models, domain adapters, and distillation adapters.

CHAPTER 1

BACKGROUND CONCEPTS

This chapter introduces basic notions on deep neural networks that are useful to understand this research. Then networks used for the problem of person ReID are explained. A basic explanation of UDA and KD is also provided.

1.1 Deep Neural Networks

The concept of ML has existed for a long time with algorithms such as SVM (Cortes & Vapnik, 1995), K-NN, or decision trees, to name a few. DL algorithms have become the main approach when tackling complex ML problems only with the recent improvement of computational resources. DL algorithms attempt to define a complex function that maps input data to a latent embedding space where data samples are easier to process. While shallow networks can theoretically approximate any function as demonstrated by (Hornik, Stinchcombe & White, 1989), models become much more powerful with many layers. Indeed, each layer feeding into the next allows for various levels of abstraction to be learned. The following sections present neural networks in more detail.

As their name suggests, artificial NNs are inspired by biological NNs. In essence, an artificial neuron is a weighted sum of elements of an input vector followed by a non-linear activation function that maps the input vector to a single output. Formally the artificial neuron can be defined as:

$$y = \sigma(\sum_{i=1}^{n} \mathbf{w}^T \mathbf{x} + b) \tag{1.1}$$

Where $\mathbf{x} = (x_1, x_2, ..., x_n, 1)$ is an input vector, $\mathbf{w} = (w_1, w_2, ..., w_n, w_{n+1})$ is the vector of weights associated to the neuron, σ is the activation function and y is the output. The activation function is essential as it allows the algorithm to model non-linear relations, which are common in real-world applications. While there exist many choices for the non-linear activation, the ReLU (Agarap, 2018) is the most commonly used. The ReLU function cuts off all negative

activations to 0 and keeps positive activations the same. This function offers some advantages over its more complex counterparts, such as being very efficient in computing and does not constrain the output to a small range.

These artificial neurons can be used to form a perceptron. A single neuron forms a very limited perception, as seen on the left of Figure 1.1, which is insufficient to tackle even simple problems. A multi-layer perceptron (MLP) network, on the middle and right of Figure 1.1, allows the algorithm to model much more complex mappings. We can see that every neuron has a connection to every neuron from the next layer, forming what is called a fully-connected layer. A single hidden layer is already a universal approximator but requires a very high number of nodes to map complex functions (Hornik *et al.*, 1989). A more feasible configuration is having multiple layers that make the relation between variables easier to map. The power of multiple layers feeding into each other is that they each learn more abstract features, which are interpreted by the following layer.

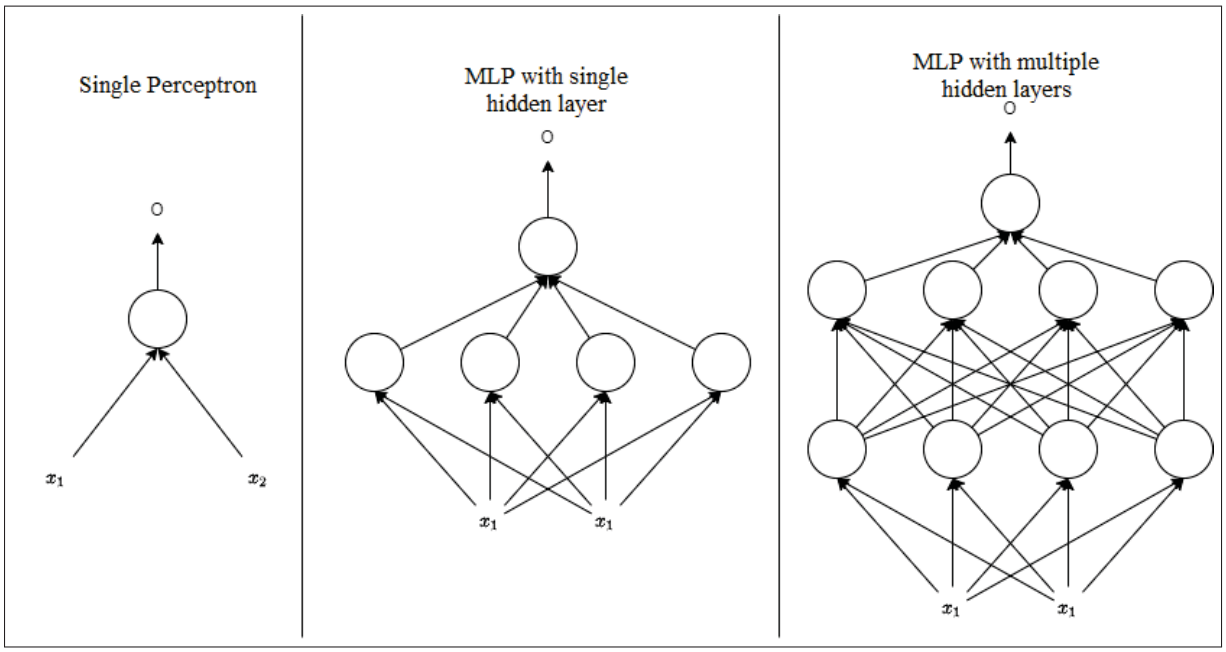

Figure 1.1 (Left) A single layer perceptron. (Middle) MLP with a single hidden layer. (Right) MLP with multiple hidden layers. Each arrow has an associated weight

A NN or ML model must be trained with data collected for the application. Considering a dataset :

$$D = \begin{pmatrix} \mathbf{x}_1; y_1 \\ \mathbf{x}_2; y_2 \\ \dots \\ \mathbf{x}_k; y_k \end{pmatrix} \tag{1.2}$$

of k samples x and associated ground-truth y. The set of weights $\mathbf{W} = w_1, w_2, ..., w_n$ and the bias b of a model Ψ, also called parameters, need to be tuned to obtain a specific output O. Note here the use of a bias b allows the model to shift the result by a constant. To train a model's parameters we input a vector $\mathbf{x}_k = (x_{k1}, x_{k2}, ..., x_{kn})$ and compare the predicted output $\hat{y} = \Psi(\mathbf{W}, \mathbf{x})$ with the ground-truth output y for that specific vector. Formally the optimization problem of training the model is:

$$\min_{\mathbf{W}} \mathcal{L}(\hat{y}, y) \tag{1.3}$$

Where $\mathcal{L}(\cdot, \cdot)$ is a loss function specific to the task being trained. In essence, the cost function defines the task the model is learning. The most common algorithm for training a neural network is called backpropagation (Rumelhart, Hinton & Williams, 1986) and applies the gradient descent optimization technique to a neural network. The idea behind the gradient descent is to move each parameter along its gradient with respect to the cost function $\mathcal{L}$ to find a minimum. The backpropagation algorithm provides a way to compute the gradients with respect to $\mathcal{L}$ for every parameter by propagating the error at the output backwards through the model using the chain rule. Weights are then updated by moving them along the gradient, minimizing the error. This is repeated for each training sample available until the model's parameters converge. An approach is to compute the gradients for every sample in the training set and update the parameters optimally. Computational limitations make this impossible for very deep architectures with

many parameters. Instead, we form subsets of samples called mini-batches and compute only the gradients for those samples. This approach is called Stochastic Gradient Descent (SGD) and has some benefits, such as avoiding local minima due to its stochastic nature.

A deep NN is simply a NN with many hidden layers. A limit of the backpropagation is that gradients become zero rapidly as they are propagated to deeper layers, and the chain rule is applied repeatedly. We call this problem the vanishing gradient problem. Once the gradients reach zero, the weights are no longer updated, and there is no training of the earlier layers of a deep network. We will see techniques that address this problem in later sections. While shallow models will often use a feature extractor to reduce the raw data to a more manageable format, deep NNs are usually trained end-to-end, meaning the model takes as input the raw data and outputs the desired response for the task directly. End-to-end training is quite demanding when the raw data is complex, for example, a three-dimensional image. The next section explores an efficient concept to solve this issue.

1.2 Convolutional Neural Networks

MLPs and NNs are a powerful tool with many applications but have some important drawbacks when it comes to computer vision tasks. In computer vision we deal with images which can be described as a large matrix of pixel values. These matrices can be vectorized and passed as input to a standard MLP. The first problem with this approach is that even a low resolution image can have several thousands of pixels. An MLP would require a very large number of parameters to process such a large input which can lead to training problems such as overfitting. Another disadvantage of the MLP is that it does not account for the spatial correlation of pixels. Indeed, pixels close to each other are more related than pixels far away and that information is lost once an image is vectorized. To address both these issues the convolutional neural network (CNN) was introduced. The idea of the CNN is to keep the shape of the input i.e. image matrix and to use localised filters to extract localised features. Similarly to equation 1.1, we formally define a convolutional layer as:

$$y = \sigma(\mathbf{W} * \mathbf{X} + b) \tag{1.4}$$

Where $\mathbf{W}$ is a small localized matrix of weights called a filter or kernel, $\mathbf{X}$ is the input matrix to that layer, b is a bias term, $*$ represents the convolution operation and σ is a non-linear activation function. The convolution operation, illustrated in Figure 1.2, can be intuitively understood as sliding a small kernel over the image and computing a dot product at at each location. The resulting matrix is called a feature map. The weights of the kernel are learned by backpropagation, similar to weights in an MLP. A convolutional layer will generally use multiple kernels to recover more varied information from the input. The output feature map has dimensions $H \times W \times C$, where H and W are the height and width of the input, respectively, and C is the number of channels that corresponds to the number of kernels used. The advantage of this method is that the model only stores a few small weight filters in each convolutional layer which greatly reduces the number of parameters learned by the model. Furthermore, the filters take into account the spatial proximity of pixels which are more likely to be correlated. The activation functions used after the convolution operation are the same as those used for MLPs and are applied element-wise to the output feature maps. There are many parameters to consider when designing a convolutional layer. The dimension and number of kernels of the layer can vary from one layer to another. The stride of a layer corresponds to the size of the steps during convolution. A stride of 1 means the convolution is evaluated at every spatial coordinate, whereas a stride of 2 means the convolution skips a spatial coordinate when moving the filter. The result of having a stride bigger than 1 is a reduction of the dimensions of the feature maps.

An important type of layer used in CNNs is a pooling layer. A pooling layer is similar to a regular convolutional layer in the sense that it uses a filter and the convolution operation to process an input matrix. The difference is that the pooling layer uses a filter that performs a predefined operation, such as an average or a maximum of all values within the filter range, as seen in Figure 1.3. This layer is often used to downsample the image, meaning to reduce the dimensions of the feature maps. Downsampling reduces the resolution of the image, which

removes small details of the image while keeping the larger structural information, This allows the model to learn a hierarchy of features. Models often reduce the dimensions of the feature maps as the image is forwarded deeper throughout the model. Conversely, the number of filters in a convolutional layer increases as the image dimension is reduced.

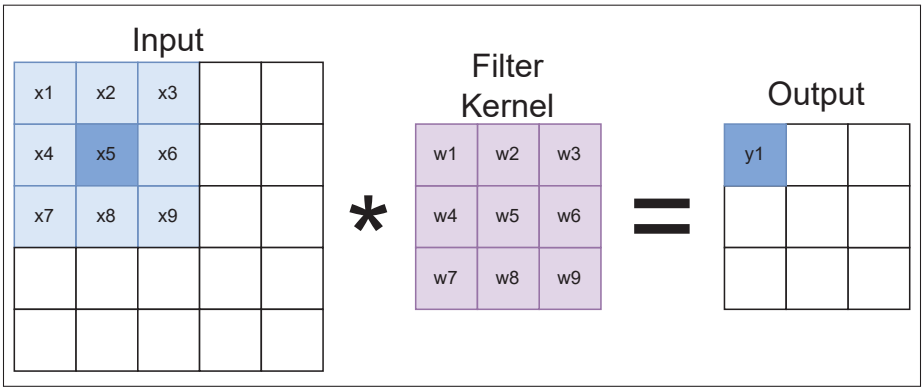

Figure 1.2 Convolution operation of an image and a kernel

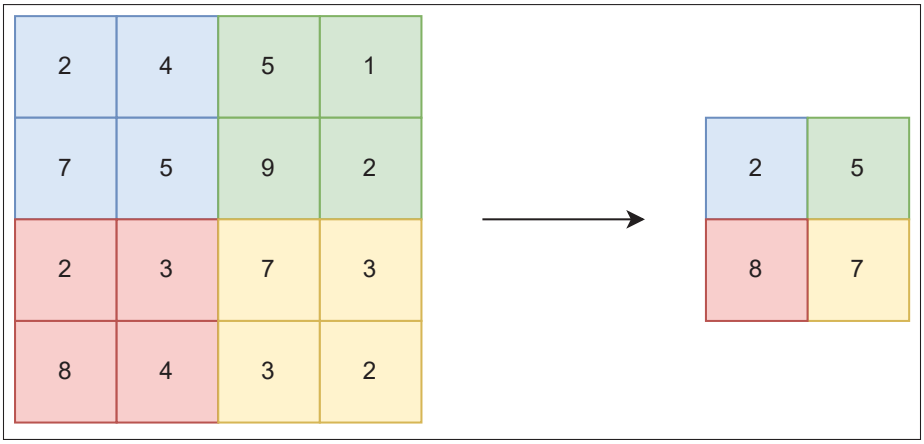

Figure 1.3 Max Pool Layer with a 2x2 filter

While deeper networks allow the model to learn more complex tasks, they are also more difficult to train efficiently. One risk of having a very deep model is that the model can overfit the training data meaning the model learns to perform perfectly on training data without learning to generalize well for new unseen data. There exist many techniques aiming to regularize the model and improve its generalization capabilities. One of the most common is the use of a batch normalization layer (Ioffe & Szegedy, 2015). This layer stores statistics about the batch data

being processed and normalizes the data using the stored mean and variance. This type of layer is very common and prevents the Vanishing Gradient effect. The normalization prevents values from becoming too small. This, in turn, speeds up the training and trains early network layers efficiently. A side benefit of this layer is that it can sometimes reduce overfitting and improve generalization.

CNNs are often combined with fully-connected layers to optimize performance. The model starts with multiple convolutional layers between which pooling layers gradually downsample the input. Once the feature maps' dimensions become small enough, they are vectorized and fed to a fully-connected layer. The following section presents a few common CNN architectures relevant to our work.

1.2.1 Overview of Backbone CNN Architectures

In video surveillance applications like person ReID, we need to train a visual backbone for similarity matching that is both accurate and cost-effective. While it is possible to design a neural network architecture from scratch, it is much more common and efficient to use time-tested architectures from the literature. Many CNN architectures have been developed over the years for either general machine learning (Alexnet (Krizhevsky, Sutskever & Hinton, 2012), Resnet (He, Zhang, Ren & Sun, 2015), VGG-16 (Simonyan & Zisserman, 2014), MobileNetV2 (Sandler, Howard, Zhu, Zhmoginov & Chen, 2018)) or designed specifically for the person ReID task (OSNet (Zhou, Yang, Cavallaro & Xiang, 2019), PCB (Sun, Zheng, Yang, Tian & Wang, 2018), MLFN (Chang, Hospedales & Xiang, 2018b)). In the following sections, we present the two main backbones we use in our experiments in more detail.

1.2.1.1 A description of Resnet

The Resnet architecture was introduced during the ImageNet Large Scale Visual Recognition Challenge in 2015 where it won the first place (He *et al.*, 2015). This architecture is used in many research areas and is the most common in person ReID (Mekhazni, Bhuiyan, Ekladious & Granger,

2020; Ge, Zhu, Chen, Zhao & Li, 2020; Mohanty *et al.*, 2022; Tian *et al.*, 2021; Isobe, Jia, Chen, He, Shi, Liu, Lu & Wang, 2021). The Resnet architecture follows a very standard CNN architecture but it makes use of a novel convolutional block illustrated in Figure 1.4. This basic block makes use of a shortcut connection which allows the input to skip a layer completely. This configuration resolves the vanishing gradient problem by allowing the gradients to propagate backward through the shortcut connections. This way, the gradient does not go through as many convolutional layers and does not shrink because of multiple chain rule applications. The Resnet architecture allows very deep models to be trained efficiently and works well for many ML applications. Multiple versions of the Resnet architecture exist where the number of layers varies. Heavier versions of the model have higher performance but require significantly more parameters stored in memory and longer inference times. Model variants are characterized by their complexity, namely Resnet18, Resnet34, Resnet50, Resnet101, and Resnet152, where the number represents the number of convolutional layers used. This architecture is very commonly used in literature (Mekhazni *et al.*, 2020; Ge *et al.*, 2020; Mohanty *et al.*, 2022; Tian *et al.*, 2021; Isobe *et al.*, 2021), and we, therefore, use it to present a better comparison to other SOTA techniques.

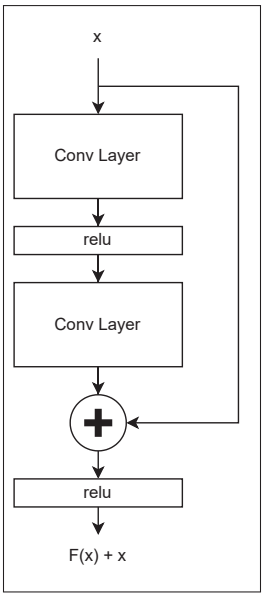

Figure 1.4 Residual block introduced in the ResNet architecture

1.2.1.2 A description of OSNet

OSNet (Zhou *et al.*, 2019) is an architecture specialized for the problem of person ReID. OSNet is based on the residual block presented in the previous section, as seen in figure 1.5. The main idea behind OSNet is to use features of the variable scale. Intuitively, when matching to images of pedestrians, it is important to consider large-scale features (like clothes and hair) as well as smaller features (like shoes, glasses, and accessories). The four branches of the OSNet residual block seen in figure 1.5 correspond to various scales of features. Features from multiple scales are then combined using a learned gating mechanism, giving weight to each scale's feature. The OSNet architecture is very light as it uses depth-wise separable convolutional layers and performs very well on the person ReID problem. We notice that the residual connection is still present to ease training as in Resnet. The model uses depth-wise convolutional layers, which are much more lightweight. Classical convolutional layers are separated into a point-wise convolution followed by a depthwise convolution resulting in fewer parameters. The gating mechanism is in itself a small MLP followed by a sigmoid function. This MLP learns to weigh each branch's output to maximize accuracy. The MLP produces a channel-wise vector of weights allowing for a more fine-grained gating mechanism than a scalar value for each branch. Similarly to the ResNet architecture, OSNet has versions of the model with different complexity. While this model offers strong performance relative to its complexity, it is not very common in the literature. This model was designed and tested specifically for person ReID problems with an emphasis on being lightweight, making the perfect model to reach our goals of accuracy and cost-efficiency. We use this model in our experiments mainly in Chapter 5 where we try to optimize model complexity.

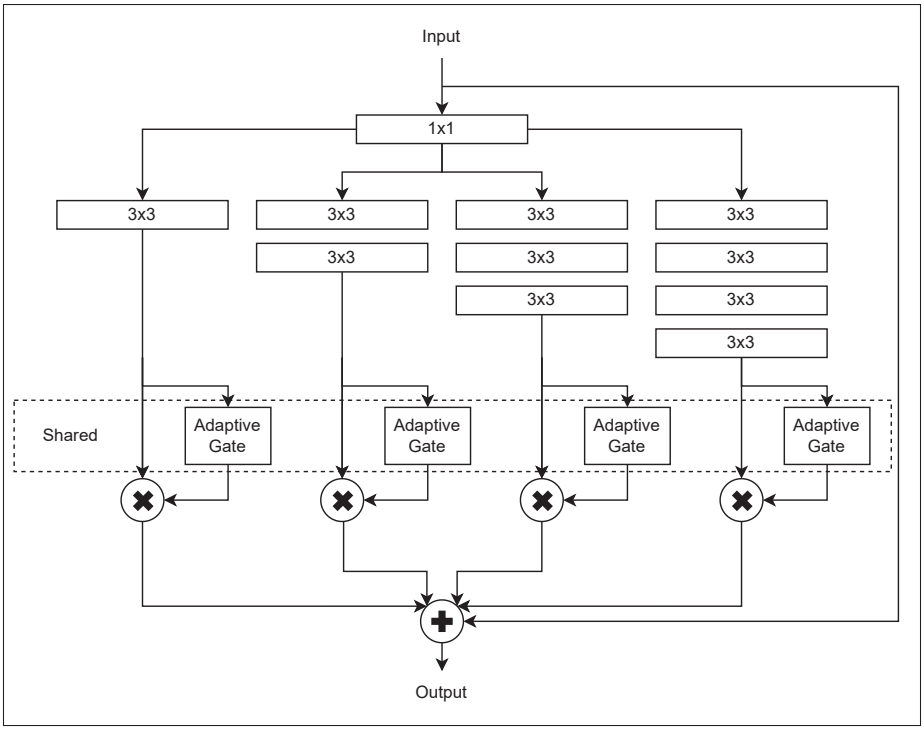

Figure 1.5 The OSNet residual block (b)

CHAPTER 2

LITERATURE REVIEW

2.1 Supervised Person ReID

In a typical person ReID system, samples are transformed into embeddings and used to form a similarity matrix. Each entry in the matrix corresponds to the distance between each pair of samples. Using knowledge about the identity of each sample, the model parameters are tuned to bring embeddings of the same identity closer while pushing away negative pairs. Once deployed, the model stores embeddings in a gallery as it receives new data. Whenever a query image is received, the system transforms it into the embedding space learned in the previous step and compares it to every embedding stored in the gallery. For each embedding, a similarity score is computed. Images in the gallery are then ranked according to their level of similarity to the query. The closer the embedding, the more likely it is of the same identity as the query. The problem of ReID can be defined in an image-based context or a video-based context. For the image-based context, images of pedestrians are captured individually with no tracking of person movement over time. For video-based ReID, images are regrouped into tracklets of images of the same identity captured over consecutive frames. Different datasets are used to evaluate works from both areas. In this work we focus on image-based ReID.

2.1.1 Conventional Machine Learning Approaches

This thesis focuses on the task of similarity matching in person ReID. The task of person ReID appeared before the popularization of DL models (Porikli, 2003; Zajdel, Zivkovic & Krose, 2005; Xu, Ma, Huang & Lin, 2014; Gheissari, Sebastian & Hartley, 2006; Liao & Li, 2015; Matsukawa, Okabe, Suzuki & Sato, 2016; Yu, Wu & Zheng, 2020). Initial efforts in the field produced feature representations of images that were hand-designed. Early works relied on color histograms to produce discriminant features. Porikli (2003) proposes to model pairwise color variations for every pair of cameras to learn a transformation that reduces illumination variations.

Zajdel *et al.* (2005) proposes to model person trajectories using a Bayesian Generative model. Gheissari *et al.* (2006) proposes to combine color histogram features with a spatiotemporal edge detector to make the features invariant to movement. Xu *et al.* (2014) proposes to split images into separate parts to represent better specific areas of the pedestrian that are discriminant such as the head, torso, legs, or feet. Liao & Li (2015) proposes to use a positive semi-definite constraint with asymmetric sample weighting to solve the imbalance between positive and negative samples. Matsukawa *et al.* (2016) proposes the use of a hierarchical Gaussian distribution descriptor to encode the image. A set of Gaussian distributions describes local patches of the image, which in turn is described by a Gaussian distribution. This repeats until we obtain high-level features of the image.Yu *et al.* (2020) proposes an unsupervised asymmetric distance metric based on cross-view clustering. This technique combines camera-specific distance metric learning with a DL model. Xiong, Gou, Camps & Sznaier (2014) proposes four new kernel-based approaches showing how multiple traditional ML techniques work on the ReID problem.

Video-based approaches also existed before DL models were commonly used (Hamdoun, Moutarde, Stanciulescu & Steux, 2008; You, Wu, Li & Zheng, 2016; Zhu, Jing, You, Zhang & Zhang, 2018; Karaman & Bagdanov, 2012; Gao, Wang, Liu, Yu & Sang, 2016) using hand-crafted features and traditional ML techniques. Hamdoun *et al.* (2008) uses SURF key points captured over multiple time-spaced images to gather information on the appearance change of identities. You *et al.* (2016) proposes a top-push distance learning which constrains the learning towards top-ranked samples yielding more discriminative features. Zhu *et al.* (2018) proposes a dual inter-video and intra-video learning metric, which makes videos more compact while also bringing together matching sequences. Karaman & Bagdanov (2012) uses a CRF to model identities over video sequences. Gao *et al.* (2016) proposes a gait-based descriptor that selects the most representative walking sequences to create more robust feature representations.

2.1.2 Deep Learning Approaches

Advances in DL have allowed us to depart from hand-crafted features towards a much more complex feature space that is learned by the CNN algorithm. The availability of larger person

ReID datasets (Zheng, Shen, Tian, Wang, Wang & Tian, 2015; Wei, Zhang, Gao & Tian, 2018; Ristani, Solera, Zou, Cucchiara & Tomasi, 2016; Li, Zhao, Xiao & Wang, 2014) also increased consequently. With these improvements, the accuracy of techniques has improved considerably. The main idea is to use a Deep Siamese network with a strong classification backbone architecture such as Resnet (He *et al.*, 2015). The three most common losses used to train the Siamese network are cross-entropy, triplet, and contrastive loss. The cross-entropy loss for person ReID aims to train a classifier where each identity in the training set is considered a class. It is not a metric learning loss. At test time, the classifier layer is discarded, and the output from the previous layer is the feature representation used by the Siamese network. The idea of the triplet loss introduced by Wang, Song, Leung, Rosenberg, Wang, Philbin, Chen & Wu (2014) is to consider three samples when computing the loss: an anchor, a positive sample of the same identity, and a negative sample of a different identity. The goal of this loss is to move the anchor representation closer to the positive representation while moving away from the negative sample. Hermans, Beyer & Leibe (2017) demonstrated that the triplet loss was a highly effective loss for the task of person ReID. They propose variants of the loss which avoid the hard sample mining, ultimately reducing the cost of computing the loss. The contrastive loss aims to bring clusters of positive matches closer together while increasing the distance to other identity clusters by optimizing the pairwise relation of samples.

Yi, Lei, Liao & Li (2014) was the first work using a Deep Siamese network to learn a feature representation in person ReID. This work, however, is limited by the size of available datasets (less than 5000 images). Wu, Chen, Li, Wu, You & Zheng (2016) proposes to combine hand-crafted features with learned features. We then transition to more recent approaches, which use learned features exclusively as they are much more reliable. We denote two main approaches to generating features from a bounding box: global feature representation and local feature representation. Additionally, some approaches consider auxiliary information about the images to enhance the representation. Recent approaches use large video-based datasets to incorporate temporal information in their representation further improving robustness to occlusions and changes of background.

Global features represent the pedestrian image as a whole, and the DL model is trained end-to-end to produce this discriminant representation. Many works use attention mechanisms to enhance their feature representation (Li, Zhu & Gong, 2018; Wieczorek *et al.*, 2021; Si, Zhang, Li, Kuen, Kong, Kot & Wang, 2018; Yang, Yan, Lu, Jia, Xie & Gao, 2019b). Other approaches (Liu, Ni, Yan, Zhou, Cheng & Hu, 2018a; Zhong, Zheng, Zheng, Li & Yang, 2018b; Zheng, Yang, Yu, Zheng, Yang & Kautz, 2019) use Generative Adversarial Networks (GAN) to enhance the training data. Zhong *et al.* (2018b) uses a GAN to generate images as if they were captured from another camera. Transferring the camera styles makes the features more invariant to viewpoint variations. Liu *et al.* (2018a) attempts to generate pose-aligned images to reduce the impact of viewpoint variations and pedestrians changing position.

On the other hand, local features consist of multiple representations corresponding to different body parts of a pedestrian, which are then aggregated to form a complete representation. Sun *et al.* (2018) separates the image into multiple horizontal slices and learns multiple specialized classifiers to generate features more robust to misalignment. Bhuiyan, Liu, Siva, Javan, Ayed & Granger (2020) proposes the use of a pose estimator to guide a gating mechanism. More relevant body parts are given more importance based on the estimated pose information. Yang *et al.* (2019b) proposes a multi-branch network that considers a global feature representation computed using the full pedestrian image as well as body parts-specific feature representations. An attention scheme determines the most discriminant representations and weighs them accordingly.

A common approach is to leverage additional information collected by the security system to improve the feature representation. Some works leverage semantic information to improve the supervision signal in supervised training (Su, Zhang, Xing, Gao & Tian, 2016; Lin, Zheng, Zheng, Wu, Hu, Yan & Yang, 2019b; Tay, Roy & Yap, 2019). Su *et al.* (2016) incorporates semantic attributes prediction in their training identifying the color of clothes and accessories in the supervision signal. Lin *et al.* (2019b) also use semantic annotations to investigate how the task of attribute detection and person ReID correlate. Tay *et al.* (2019) uses an attribute detection model t generate attribute attention maps which in turn are used to improve a parts detection model. Other works use camera information as an auxiliary supervision signal (Lin,

Ren, Lu, Feng & Zhou, 2017; Liu & Zhang, 2019; Chang, Hospedales & Xiang, 2018a; Sarfraz, Schumann, Eberle & Stiefelhagen, 2018a; Zhong, Zheng, Zheng, Li & Yang, 2018a). Lin *et al.* (2017) proposes a camera consistency condition during training to enforce more similar feature representations across cameras. Liu & Zhang (2019) uses a view confusion model to make feature representations more similar between viewpoints. Chang *et al.* (2018a) combines semantic attributes with camera information to find an optimal common feature space. Sarfraz *et al.* (2018a) proposes a re-ranking algorithm based on the camera information and a viewpoint detector. Zhong *et al.* (2018a) uses a GAN to transform images from a certain camera into a synthetic sample from another camera allowing more efficient training. Other approaches use GANs to produce auxiliary supervision signals related to pose information (Liu, Ni, Yan, Zhou, Cheng & Hu, 2018b; Qian, Fu, Xiang, Wang, Qiu, Wu, Jiang & Xue, 2018). Liu *et al.* (2018b) uses pose information to generate new pose-rich samples. A GAN is then used to confuse the pose-generated samples and real samples. Qian *et al.* (2018) uses pose information to generate a synthetic sample as well. The sample is then processed by a pose-invariant model and fused back with the representation of the real sample.

Most previously mentioned techniques are image-based and treat images of pedestrians one at a time. Video-based person ReID has gained increased traction in recent years (Aich, Zheng, Karanam, Chen, Roy-Chowdhury & Wu, 2021; Wang, Zhang, Gao, Geng, Lu & Wang, 2021b; Zang, Li & Gao, 2022) as video sequences offer additional information unavailable in single images. Video-based ReID allows the use of temporal information as multiple subsequent frames are available and are called tracklets. They can give valuable information, such as a person's gait. McLaughlin, Del Rincon & Miller (2016) propose the use of an RNN to aggregate information from multiple frames. Xu, Cheng, Gu, Yang, Chang & Zhou (2017) employs a two-stream approach where a gating mechanism combines spatial and temporal information. Hou, Ma, Chang, Gu, Shan & Chen (2019b) uses the subsequent video frames to complete occluded regions improving the model's resilience. Zhao, Shen, Jin, Lu & Hua (2019) propose a frame re-weighting scheme based on pedestrian attributes to maximize spatiotemporal information.

2.2 Unsupervised Domain Adaptation

While all the techniques mentioned in the previous section yield impressive results, they are not realistic as they require large amounts of annotated data to perform well. A more suitable approach to real-world applications is an unsupervised approach leveraging unlabeled data to train. UDA aims to use labeled data from a source domain in conjunction with unlabeled target domain data to adapt the model to the target domain. This branch of transfer learning is very popular for person ReID. In this section, we present UDA applied specifically to person ReID. We denote five main common families of approaches: pseudo-labeling, reconstruction, discrepancy, adversarial, and attention-based. Pseudo-labeling approaches aim to use source domain information as a basis to assign reliable pseudo-labels to target domain samples. Reconstruction approaches map both domains to a new common domain to optimize the training process. This can be done by transferring the style of source and target data between each other. Discrepancy-based approaches attempt to match the data distribution between the source and target data. Adversarial approaches use multiple competing networks to identify domain-invariant features. Attention-based approaches use source domain training to find ROIs that maintain shared information between domains.

First, pseudo-labeling techniques (Ge *et al.*, 2020; Lin, Dong, Zheng, Yan & Yang, 2019a) attempt to use the source data to mine target pseudo-labels generally through clustering algorithms. The pseudo-labels are then used in a supervised training scheme. Supervised training on the source data is also included to avoid the collapse of the representation. The main drawback of these approaches is that the quality of pseudo-labels has an important effect on performance as false positives and negatives are learned by the model. Similarly to the supervised scenario, some techniques use camera information to enhance the representations (Liang, Wang, Lai & Zhu, 2018; Xuan & Zhang, 2021; Li, Zhu & Gong, 2019). Liang *et al.* (2018) considers each domain as a combination of sub-domains corresponding to the different cameras allowing for a more precise image translation. Xuan & Zhang (2021) proposes splitting the training into intra-camera and inter-camera training. By doing so, the model learns to differentiate identities within a

camera as well as over the whole dataset. Li *et al.* (2019) propose a similar approach where they additionally use tracklet information to form the cluster for the intra-camera setting.

Reconstruction-based techniques aim to generate images that bridge the gap between source and target domains (Wei *et al.*, 2018; Deng, Zheng, Ye, Kang, Yang & Jiao, 2018; Huang, Wu, Xu & Zhong, 2019a; Chen, Zhu & Gong, 2019b; Bak, Carr & Lalonde, 2018). Wei *et al.* (2018); Deng *et al.* (2018) propose GAN-based approaches to transform images from the source dataset into images with the style of the target dataset. This makes training on the labeled transformed source dataset much more valuable when performing ReID on the target dataset. (Huang *et al.*, 2019a) generate images where the background is suppressed. The goal is to reduce the domain shift caused by varying backgrounds. Similarly, Chen *et al.* (2019b) transforms source images by placing pedestrians in a target domain background to enrich the target dataset with labeled source identities. The results of these techniques are highly dependent on the quality of the generated images.

Discrepancy-based approaches are popular in visual recognition UDA. Mekhazni *et al.* (2020) applies a discrepancy-based approach to person ReID in the dissimilarity space. Rather than matching feature distribution directly, they use tracklet information to match pairwise distance distributions between the source and target domains. They also evaluate the approach with image-based datasets using images from the same camera and identity as tracklets.

Adversarial approaches use two competing networks to find an invariant common representation space. Delorme, Xu, Lathuilière, Horaud & Alameda-Pineda (2021) uses an adversarial approach combined with a clustering approach to generate domain-invariant features. They find that straightforward adversarial training results in a negative transfer and so introduce a conditional adversarial training scheme based on cluster centroids from the clustering module. Fu & Lai (2021) proposes a cross-view adversarial network that matches distributions across different viewpoints within the datasets.

Attention-based approaches use mechanisms to direct the model towards regions of interest that maintain shared information between source and target. Wang, Liu, Raychaudhuri, Paul,

Wang & Roy-Chowdhury (2021a) proposes a multi-level attention mechanism based on weak video annotations to reduce the domain gap between source and target samples. Nikhal & Riggan (2021) use several trainable attention modules in their model which learn important regions of interest on source data which transfers well to target data. Wu, Yang & Wang (2022) introduces a multi-context attention architecture that generates attention maps for both global and local contexts to enhance the representation.

The next two subsections present more detail on two representative SOTA STDA methods which we use as parts of our experiments: Dissimilarity-based Maximum Mean Discrepancy (D-MMD) (Mekhazni *et al.*, 2020) and Self-Paced Contrastive Learning (SPCL) (Ge *et al.*, 2020). Both techniques have been developed specifically for UDA in-person ReID, producing some of the best results in the field. We chose these techniques as they are the most performing at the moment and function very differently. This will allow us to show that our research is up-to-date and compatible with various techniques. Additionally, we note that while the D-MMD approach uses tracklet information during training it is still evaluated on image-based person ReID datasets. The technique considers images from the same identity and camera to belong to the same tracklet. Therefore the technique is considered an image-based ReID technique.

2.2.1 Dissimilarity-based Maximum Mean Discrepancy

The first UDA technique we use as part of our experiments is the one proposed by Mekhazni *et al.* (2020). This technique aims to align the pair-wise dissimilarity between domains. The technique relies on a loss with two parts: a supervised loss and a Dissimilarity-Based Minimum Mean Discrepancy (D-MMD) loss. The supervised loss is as cross-entropy combined with a triplet loss and is computed on the labeled source data. This loss maintains the integrity of the previous pre-training while the models adapt to a new domain.

To compute the D-MMD, the distance distributions must be first computed. The Euclidean distances between features for every pair of images belonging to the same identity are computed

as follows:

$$d_i^{wc}(M(x_i^u), M(x_i^v)) = ||M(x_i^u) - M(x_i^v)||, u \neq v \tag{2.1}$$

where M is the model being adapted and x_i^u is the u-th image of class i. Note that we use tracklet information to determine the images with the same identity on the unlabeled target dataset. After that, the distances between features for samples from different identities are computed as:

$$d_{i,j}^{bc}(M(x_i^u), M(x_j^z)) = ||M(x_i^u) - M(x_j^z)||, i \neq j, u \neq v \tag{2.2}$$

We then define $\mathbf{d^{wc}}$ and $\mathbf{d^{bc}}$ as the distributions of distance values d_i^{wc} and $d_{i,j}^{bc}$ respectively. These distributions characterize the features in the dissimilarity space.

MMD (Gretton, Borgwardt, Rasch, Schölkopf & Smola, 2012) is a metric used to evaluate the distance between two distributions and takes the form:

$$MMD(P(A), Q(B)) = \frac{1}{I^2}\sum_{i=1}^{I}\sum_{j=1}^{J} k(a_i, a_j) + \frac{1}{J^2}\sum_{i=1}^{I}\sum_{j=1}^{J} k(b_i, b_j) - \frac{2}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J} k(a_i, b_j) \tag{2.3}$$

where $P(A)$ is the distribution of the source domain A and $Q(B)$ is the distribution of target domain B. $k(\cdot, \cdot)$ is a kernel. a_i is the i-th sample for distribution $P(A)$ and b_i is the i-th sample from distribution $Q(B)$. I and J are the total number of samples in distributions $P(A)$ and $Q(B)$ respectively. The goal is to minimize this metric between the source and target domain. Therefore we optimize by minimizing the MMD metric in the dissimilarity space as well as the feature space. The D-MMD loss is defined as follows:

$$\mathcal{L}_{D-MMD} = MMD(\mathbf{d_s^{wc}}, \mathbf{d_t^{wc}}) + MMD(\mathbf{d_s^{bc}}, \mathbf{d_t^{bc}}) + MMD(S, T), \tag{2.4}$$

where S and T are the source and target distributions of features, respectively. The subscripts s and t indicate whether the distance distributions were computed on source or target samples. The overall loss for domain adaptation can be expressed as:

$$\mathcal{L}_{DA}(x_s, x_t) = \mathcal{L}_{D-MMD}(x_s, x_t) + \mathcal{L}_{ces}(x_s) + \mathcal{L}_{tri}(x_s) \tag{2.5}$$

By training the model by the use of this loss function, the distances of features produced by the model for the source and target domains are aligned, which causes the model to perform well on the target domain as well.

2.2.2 Self-paced Contrastive Learning with Hybrid Memory

SPCL (Ge *et al.*, 2020) is another effective technique based on clustering to produce strong teacher models. Since it is assumed that the data samples from the target domains are not annotated, the initial step is to generate pseudo-labels for them. Clustering could be performed using a standard clustering algorithm such as DBSCAN (Ester, Kriegel, Sander, Xu et al., 1996).

The method relies on the criteria of compactness and independence to determine which clusters are reliable. Compactness is the measure of how close samples are within a cluster, and Independence is the measure of how close a cluster is to other samples not in the cluster. Samples in clusters deemed reliable are assigned a pseudo-label matching their cluster, and samples that are not in a reliable cluster is considered un-labeled samples. The feature vectors for every sample are then kept in a hybrid memory module under three categories: *labeled* source samples, *pseudo-labeled* target samples, and *unlabeled* target samples.

For each label in the source samples category, a class centroid corresponding to the mean feature vector of the label is generated. Similarly, class centroids are generated using the pseudo-labels of the clustered target samples. Once all the data is properly categorized and class centroids are determined, a unified contrastive loss function is used to adapt the teacher models

to corresponding target datasets. The domain adaptation loss, $\mathcal{L}_{DA}$ defined as:

$$\mathcal{L}_{DA} = -log\frac{exp(\langle \mathbf{f}, (z^{+})\rangle/\tau)}{\sum_{k=1}^{n^s} exp(\langle \mathbf{f}, (w_k)\rangle/\tau) + \sum_{k=1}^{n_c^t} exp(\langle \mathbf{f}, (c_k)\rangle/\tau) + \sum_{k=1}^{n_o^t} exp(\langle \mathbf{f}, (v_k)\rangle/\tau)} \quad (2.6)$$

Where $\mathbf{f}$ is the feature vector being used to compute the loss, τ is a temperature hyperparameter, $\mathbf{w_k}$ is the class centroid for the source label k, $\mathbf{c_k}$ is the class centroid for target pseudo-label k and $\mathbf{v_k}$ is the feature vector of un-labeled target sample k. $\mathbf{z}^{+}$ corresponds to the positive category for sample $\mathbf{f}$. n^s, n_c^t, and n_o^t are the number of labeled source samples, number of pseudo-labeled target samples, and number of un-labeled samples, respectively, within the mini-batch. In essence, this loss brings every sample closer to its positive centroid while pushing it away from negative centroids. The clusters are re-evaluated every epoch, and centroids are updated each epoch accordingly until convergence. To summarize, SPCL loss brings every sample closer to its positive centroid while pushing it away from negative centroids. The clusters are re-evaluated every epoch, and centroids are updated each epoch accordingly until convergence.

2.3 Multi-target Domain Adaptation

MTDA is a generally less explored field of research than STDA, with only a few papers treating the subject (Gholami, Sahu, Rudovic, Bousmalis & Pavlovic, 2020; Nguyen-Meidine, Bela, Kiran, Dolz, Blais-Morin & Granger, 2020; Isobe *et al.*, 2021; Yu, Hu & Chen, 2018). The goal of MTDA is to have a system which performs well on multiple datasets simultaneously. The main challenge is that the multiple target datasets can present a significant domain shift. While treating the problem as multiple STDA problems with multiple models would be simple, this solution becomes rapidly unmanageable when the number of target domains is large. Nguyen-Meidine *et al.* (2020) proposes a knowledge distillation-based approach that transfers knowledge from multiple teacher networks to a single student network. The student attempts to match features extracted at multiple points in the teacher models to produce representations similar to that of the teacher. Gholami *et al.* (2020) attempts to find a shared feature space for all target domains while disentangling domain-specific characteristics by using an information theoretic approach. Isobe *et al.* (2021) jointly trains a common model with target-specific expert models. All models

are encouraged to collaborate through the use of bridging loss terms and regularization. Yu *et al.* (2018) produces shared model parameters in a pairwise manner between the source and target domain and between the target domain. Chen, Zhuang, Liang & Lin (2019c) proposes an alternative problem statement where different domains are considered sub-domains of a large blended dataset. An adversarial approach is then used to learn to distinguish the sub-domains and improve accuracy. Rebuffi, Bilen & Vedaldi (2018) proposes the addition of target-specific parameters to an existing model. The model then uses the appropriate parameters when facing a sample from a given target.

To the best of our knowledge, only two works currently deal with MTDA in person ReID (Tian *et al.*, 2021; Mohanty *et al.*, 2022). Tian *et al.* (2021) proposes a Camera Identity-guided Distribution Consistency method that attempts to align the target domains. Different types of consistency, such as scene consistency and domain consistency, are enforced during training to ensure invariance between target domain features. Mohanty *et al.* (2022) is a clustering approach that aims to generate domain-invariant features. It does so using the DVIB (Alemi, Fischer, Dillon & Murphy, 2016) loss and ECW (Kirkpatrick, Pascanu, Rabinowitz, Veness, Desjardins, Rusu, Milan, Quan, Ramalho, Grabska-Barwinska et al., 2017) regularization. We should also mention a multi-source UDA (MSDA) technique (Wu, Zheng, Guo & Lai, 2019a) based on KD that we also use as a comparison to our proposed approach. The main idea of this work is to dynamically determine the relevance of each labeled source dataset using a very small subset of labeled target data. A gating mechanism then controls the distillation giving more importance to more relevant datasets. This technique would be considered weakly supervised DA but is still related to our work due to the use of KD.

2.4 Knowledge Distillation

KD is a ML concept first introduced by Hinton *et al.* (2015) as a technique to compress ensembles of models. The idea is to use the output of a large trained teacher model to train a smaller model for the same task. A loss is computed between the outputs of both models with the goal of optimizing the smaller model so its output resembles the larger model's output, as seen in figure

2.1. This approach allows smaller models to benefit from the training of larger models on the same data. Since then, it has become a popular field of research with many varied algorithms (Chen, Choi, Yu, Han & Chandraker, 2017; Zhang, Zhu & Ye, 2019a; Meng, Li, Zhao & Gong, 2019). Chen *et al.* (2017) proposes new losses adapted for the object detection problem. Zhang *et al.* (2019a) propose to use KD to produce a lightweight and fast pose estimator suitable for real applications. Meng *et al.* (2019) proposes a smart student configuration in which the student conditionally learns from the teacher or the ground truth based on teacher performance.

Some work found that using only the output of a DL model was insufficient to capture the knowledge of a large teacher (Romero, Ballas, Kahou, Chassang, Gatta & Bengio, 2014). Many works (Passban, Wu, Rezagholizadeh & Liu, 2021; Wang, Hu, Lai, Zhang & Zheng, 2019b; Chen, Mei, Zhang, Wang, Wang, Feng & Chen, 2021) use feature distillation where feature maps for various intermediate layers of the models are extracted to compute distillation losses allowing the student model to follow more closely the teacher throughout its architecture. Romero *et al.* (2014) proposes the use of "hints" from intermediate layers to complement the output-based KD. Passban *et al.* (2021) proposes to combine multiple teacher layer feature maps with an attention scheme based on layer relevance. These combined feature maps offer a better correspondence to the student feature maps. Wang *et al.* (2019b) propose to use KD to train a student for action prediction. The teacher model is trained using full video sequences while the student only gets partial sequences. The student learns to predict actions from the missing part of the video through the teacher. Chen *et al.* (2021) uses a learned attention module to find the optimal correspondence between layers from the teacher and student model.

In the most basic form of KD, the teacher model is frozen once trained and the student is adapted to match it; this is called offline distillation. In online distillation, the teacher is trained at the same time as the student to provide a better learning curve for the student. Walawalkar, Shen & Savvides (2020) proposes an ensemble of students that learn from a pseudo-teacher model. Wu & Gong (2021) uses an ensemble teacher with many branches called peers. KD is used between the peers and the ensemble teacher so they both improve iteratively. Zhang, Hu, Qin, Xu & Wang (2021) uses a GAN to generate divergent examples, which are then used to

train an ensemble of students. Another type of distillation is self-distillation, where a single model is both the teacher and the student. Zhang, Song, Gao, Chen, Bao & Ma (2019b) uses information from later layers to guide the earlier layers. Yang, Xie, Su & Yuille (2019a) proposes to distill from a snapshot of an earlier epoch to enforce consistency in training. Yuan, Tay, Li, Wang & Feng (2020) proposes a teacher-free system that uses KD as a label-smoothing regularization mechanism.

Previously mentioned techniques use a standard single teacher and student configuration, but an alternative approach is to use a combination of multiple teachers to train a single student. Hinton *et al.* (2015) already proposed to use the average of the output of multiple teachers to strengthen the distillation. Since then, more sophisticated approaches have been developed (Chen, Su & Zhang, 2019a; Shen, Xue, Wang, Song, Sun & Song, 2019; Luo, Pan, Wang, Wang, Tang & Song, 2020). Chen *et al.* (2019a) uses two teachers: one to transfer output-based knowledge and the other to transfer feature-based knowledge. Shen *et al.* (2019) proposes a novel Knowledge amalgamation module to combine the output between two models. By cascading the modules, they obtain a single-student model with amalgamated knowledge from all initial teachers. Luo *et al.* (2020) proposes an approach to train a multi-task student from multiple teachers specialized for different tasks.

There are multiple examples of KD applied to the problem of ReID (Ren & Li, 2018; PENG, KUANG, LI & GU, 2019; Yu, Liu, Lu, Chu & Yu, 2022; Wu *et al.*, 2019a). Ren & Li (2018) train multiple partial ReID models and combine them using KD. Each teacher model is specialized for a specific type of view (focused on the faces, feet, ...). This technique employs feature distillation, where feature maps from the various partial models are factorized and compared to the features from the student model. PENG *et al.* (2019) proposes a simple joint loss configuration where feature distillation and classical ReID losses are combined to train a student efficiently. Yu *et al.* (2022) tackle the problem of pedestrians changing clothes through 3D representations. This technique uses a 3D reconstructed body mesh of pedestrians to extract ReID-specific geometry representations. The 3D representation extractor is considered a teacher network to the standard ReID student model. The distillation between these two models guides the student

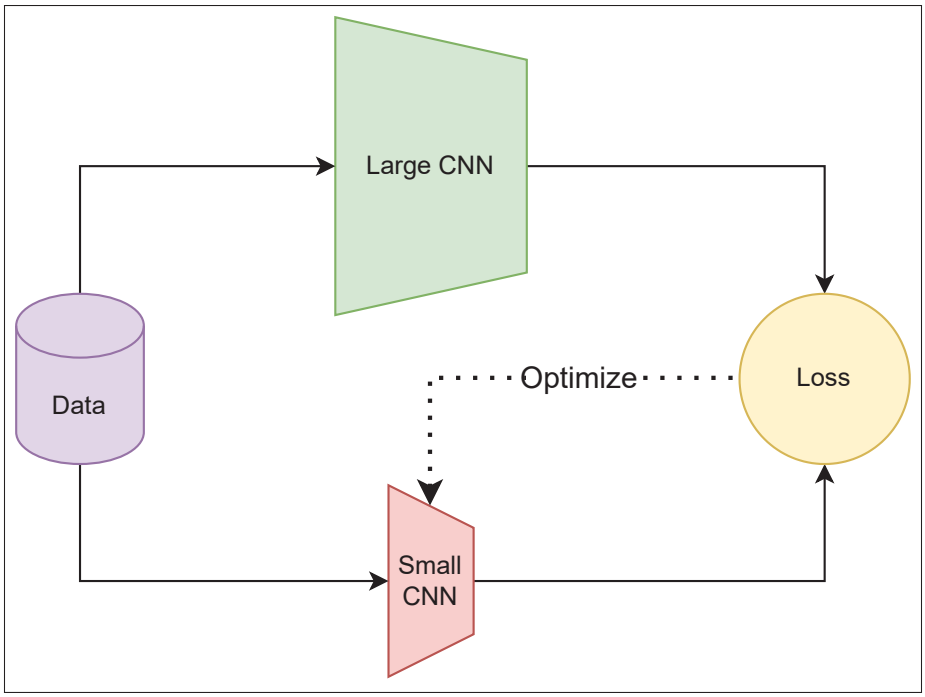

Figure 2.1 Basic Knowledge Distillation

toward producing more clothes-invariant representations. Wu *et al.* (2019a) tackles the MSDA problem configuration for ReID. Their goal is to leverage multiple source datasets to adapt to an unlabeled target dataset. Their technique performs a weighted multi-teacher KD based on dataset relevance. They use a small subset of labeled data to determine dataset importance, making this technique a Semi-Unsupervised DA technique.

2.5 Critical Analysis

We observe two major issues in the state-of-the-art. First, we notice the limited research in MTDA for person ReID contrary to the STDA problem, which is studied extensively. MTDA is challenging as it is necessary for real-world systems to integrate several target domains into a single DL model, each with its own set of variations (cameras, illumination, occlusions). Existing MTDA techniques in ReID fail separate the problem into simpler parts allowing the use of existing STDA techniques

Second, we notice that very few works focus on model complexity, even though producing a compact model is essential in real-world applications where computational resources are limited. Most techniques presented in this section use large architectures, which would be unusable in real-time video surveillance applications. This work presents a flexible approach that extends existing STDA work to MTDA while optimizing model complexity to accuracy relation. We propose the use of KD to transfer knowledge from trained STDA models to a student model. The knowledge compression property of KD allows us to minimize model complexity at the same time.

CHAPTER 3

EXPERIMENTAL METHODOLOGY

3.1 Person ReID Datasets

The choice of datasets is important when evaluating MTDA because the domain shift between datasets has an important impact on results. The older person ReID datasets such as VIPeR (Gray, Brennan & Tao (2007)) , RAiD (Das, Chakraborty & Roy-Chowdhury (2014)), and PRiD (Hirzer, Beleznai, Roth & Bischof (2011)) are very limited and have been mostly replaced by more modern and larger datasets Zheng *et al.* (2015); Li *et al.* (2014); Ristani *et al.* (2016); Wei *et al.* (2018). We conduct our experiments using commonly used datasets with a larger number of identities, cameras, and bounding boxes, namely : Market1501, DukeMTMC-ReID, CUHK03, and MSMT17. Table 3.1 shows a brief overview of the main dataset characteristics. The following sections will present each dataset used in more detail.

Table 3.1 Properties of the four challenging datasets used in our experiments. Annotations are produced either by hand, using a Deformable Parts Model (DPM) or a Faster R-CNN

Datasets	# cameras	# images (IDs)	# train (IDs) gallery (IDs) query (IDs)	Annotation method
Market-1501	6	32668 (1501)	12936 (750) 15913 (751) 3368 (751)	DPM
CUHK03	5	13164 (1360)	7365 (767) 5332 (700) 1400 (700)	Hand / DPM
Duke-MTMC	8	36441 (1404)	16522 (702) 17661 (702) 2228 (702)	manual
MSMT17	15	126441 (4101)	32624 (1041) 82161 (3060) 11659 (3060)	Faster R-CNN

3.1.1 Market1501

Market1501 (Zheng *et al.* (2015)) is composed of images captured at the exit of a campus supermarket and is the most commonly used dataset for person ReID. The dataset contains 32,668 bounding boxes belonging to 1501 identities. The images are captured by 8 cameras as illustrated in Figure 3.1 with some overlap in their fields of vision, which explains why this dataset is often the one yielding the highest accuracy. Each identity has images from at least 2 cameras to ensure cross-camera search is possible. All bounding boxes are 64x128 pixels. The bounding boxes are generated using an automatic pedestrian detection, namely Deformable Part Model (DPM) (Felzenszwalb, Girshick, McAllester & Ramanan (2010)).

For testing, the query set contains at most 1 bounding box per identity per camera, so a specific person can have at most 6 query samples. Samples from the gallery which are from the same identity and same camera as the query are not considered when ranking as we evaluate for cross-camera matching.

Additionally, this dataset includes a set of distractors, i.e., images of identities that are not part of the 1501 base identities, images of the background, or bounding boxes that contain less than 20% of a person. This set contains 500,000 images and forms a larger dataset named Market1501+500k. We do not use this dataset in our experiments.

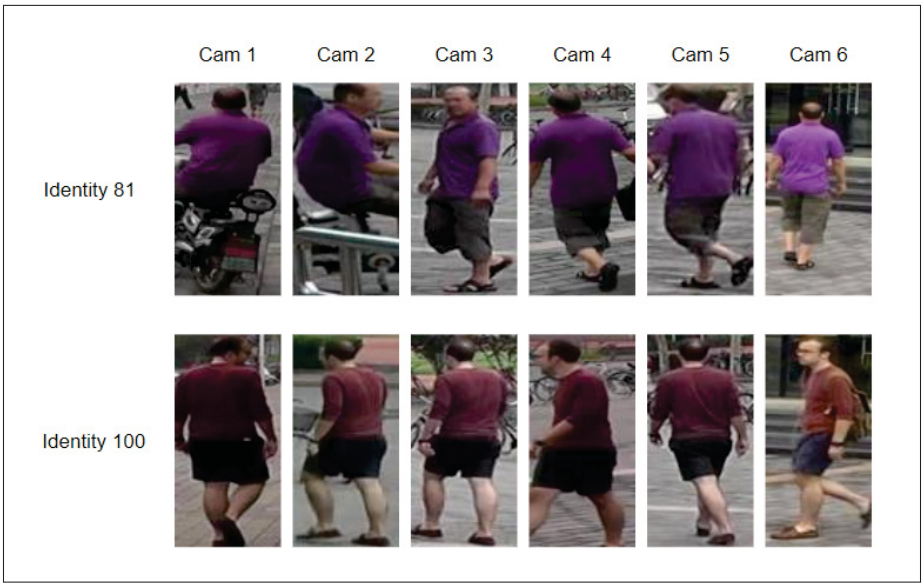

Figure 3.1 Cross-camera images from the Market1501 Dataset

3.1.2 DukeMTMC-ReID

DukeMTMC-ReID (Ristani *et al.* (2016)) is another frequently used dataset composed from frames recorded on the campus of Duke University. The dataset contains 36441 bounding boxes of 1404 identities. The frames were captured by 8 cameras, with some cameras having some overlap in their fields of vision. This dataset was initially developed with person tracking in mind and contains annotations of person trajectories made by hand. Images from the dataset have varying dimensions, as shown in Figure 3.2, but are all resized to 64x128 pixels during training. This dataset also has an alternative version named DukeMTMC4ReID which uses the Doppia person detector (Benenson, Omran, Hosang & Schiele (2014)). This dataset is a tracking dataset that often has many bounding boxes from the same camera of the same identity with very little variation between images.

Figure 3.2 Cross-camera images from the DukeMTMC-ReID Dataset

3.1.3 MSMT17

MSMT17 (Wei *et al.* (2018)) is by far the largest and the most recent dataset we use in our experiments. The dataset contains 126441 bounding boxes of 4101 identities. The footage is captured by 15 cameras which are deployed both indoors and outdoors. The quality of images in this dataset is much higher than for

other datasets and bounding box dimensions have a high variance, as shown by Figure 3.3. Furthermore, images are captured with different weather conditions and lighting. This dataset favors a much larger test set compared to the training set with almost 75% of identities, compared to the more common 50% in other datasets. This, with a large number of bounding boxes, makes it the most challenging dataset we use. Bounding boxes are generated by the Faster R-CNN detector (Ren, He, Girshick & Sun (2015)), a more reliable detector than the DPM detector commonly used by Market1501 and CUHK03.

Figure 3.3 Cross-camera images from the MSMT17 Dataset

3.1.4 CUHK03

CUHK03 (Li *et al.* (2014)) is the smallest and oldest dataset we consider in our experiments. It is the third dataset published by the Chinese University of Hong Kong after CUHK02 (Li & Wang (2013)) and CUHK01 (Li, Zhao & Wang (2012)). The dataset contains 13164 bounding boxes of 1360 identities. This dataset has a very small test set compared to to the training set with only 100 identities used for testing.

Images are captured by 6 cameras. Cameras are divided into pairs of cameras with disjointed views and each identity appears in one pair as seen in Figure 3.4. The dataset has bounding boxes generated by hand as well as by a person detector, DPM (Felzenszwalb *et al.* (2010)). In our experiments, we use the labels automatically generated as they represent a much more realistic scenario. The recordings were made over months which causes important illumination changes. We note that Zhou *et al.* (2019) proposes an alternative split of the dataset where half the identities are used for training and the other half for training which is more in line with the splits of the more modern datasets previously mentioned. We choose to conduct our experiments using the older split as it allows a better comparison with results from literature which in the majority use the old split. The different splits yield very different results as the number of training samples varies significantly as shown in Table 3.2.

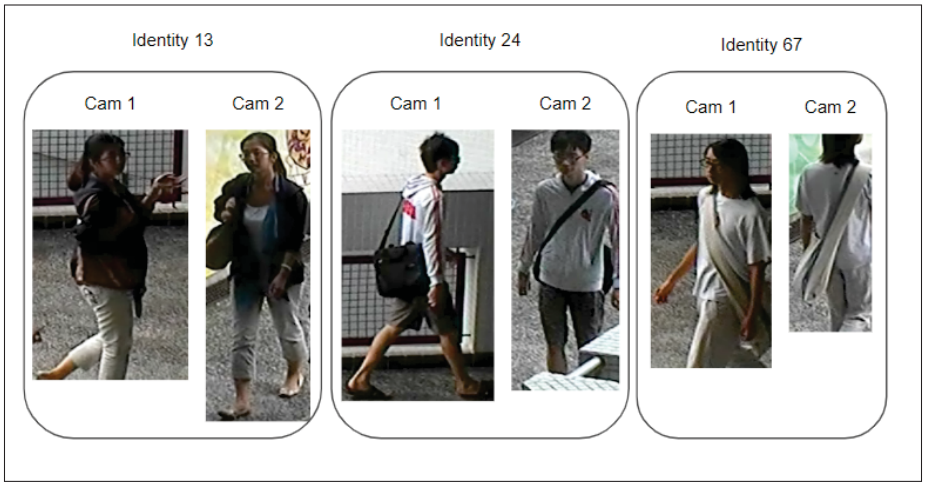

Figure 3.4 Cross-camera images from the CUHK03 Dataset

Table 3.2 D-MMD using MSMT as source and CUHK03 as target for the two CUHK03 splits

Split (train/test)	mAP (%)	rank-1 (%)
Old (1260/100)	60.5	64.0
New (680/680)	29.3	29.8

3.2 Performance Metrics

3.2.1 Accuracy Metrics

The main aspect to consider when evaluating a person ReID approach is accuracy. There exist multiple ways to measure accuracy in ReID problems. The two most commonly used metrics are the Cumulative Matching Characteristics (CMC) curve and the mean Average Precision (mAP).

CMC curves are the older method of reporting accuracy in person ReID. When evaluating a model, the embedding it generates for a query image is compared to the embedding generated for all gallery images. Gallery images are then ranked from most similar (having the smallest distance to the query embedding) to least similar (having a high distance). From this ranking, we derive the rank-n accuracy. For a single query sample, the rank-n accuracy is 1 if there is a positive identity match in the n first gallery samples from the ranked list and is 0 otherwise. This value is averaged over all query samples in the test to give the overall rank-n accuracy. The CMC curve is simply the curve when plotting the rank-n accuracy over all values of n. In literature, the CMC curve is often presented by only reporting a few rank-n accuracy values i.e. rank-1, rank-5, rank-10. While this metric is appropriate in a single-gallery-shot setting, meaning there is only 1 positive match in the gallery for each query sample, it can be imprecise in a multi-gallery-shot setting. Formally, rank-n accuracy can be defined as:

$$rank_n = \begin{cases} 1 & \text{if top-}n\text{ ranked gallery samples contain the query identity} \\ 0 & \text{otherwise} \end{cases}$$

Where this represents the rank-n accuracy for a single sample. The reported value in this work is the average rank-n accuracy over the whole test set.

mAP is a metric that has become more and more standard when reporting the accuracy of person ReID methods. While the CMC curve fails to take into account the recall, the mAP is a representative metric even in the multi-gallery-shot setting. The Average Precision (AP) is defined as the area under the precision-recall curve for a single query, and the mAP is simply the average of the AP of all query samples. Precision can be computed as:

$$\frac{T_p}{T_p + F_p}$$

Where T_p represents true positives and F_p represents false positives. This can be seen as the fraction of true matches accurately detected over all detections. Recall is computed as:

$$\frac{T_p}{T_p + F_n}$$

Where T_p represents true positives and F_n represents false negatives. This can be interpreted as the fraction of images successfully matched to the query over every possible positive match. With these 2 values computed, we find the area under the precision-recall curve to get the Average Precision. The mAP is simply the average of all APs for every query image.

Figure 3.5 shows various cases of ranking lists for a single sample and how they would be evaluated. Cases a) and b) are in the single-gallery-shot settings, and c) and d) are in the muli-gallery-shot setting. We can see that case b) is a worse ranking as the positive sample is ranked last, while in case a), the positive sample is ranked first. The CMC curve reflects that in one case, the rank-1 is 0 while it is 1 in the other. In cases c) and d), however, we see that case d) is better than c) since the positive samples are at the start of the ranking. The CMC curve, however, does not reflect that since both cases get the same rank-1 and rank-5. Looking a the AP now, we see that it differentiates both a) from b) and c) from d). Even though mAP is a better metric overall, we use both the CMC curve and the mAP to evaluate the accuracy of our approaches to allow a better comparison with SOTA techniques in the literature.

3.2.2 Complexity Metrics

We consider two metrics when evaluating the complexity of a DL model which relate to memory footprint and time complexity. First, we consider the number of parameters required to store the model in memory. This metric indicates the required storage capacity and computational resources to integrate the model into a real-world application. It is important to minimize this aspect of our solution as large models can require an important amount of reading and writing in slower memory while a small model can be conveniently stored in a cache. For this measurement, we consider all parameters which characterize

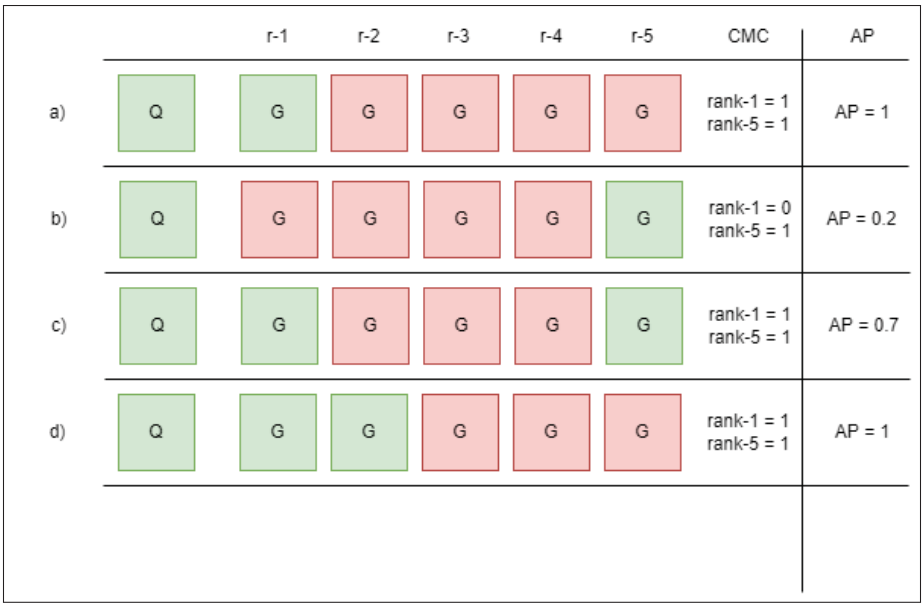

Figure 3.5 Simplified example of AP and CMC values for various ranking cases for a single query. Green boxes represent the positive samples, and red boxes are the negative samples

the various layers of a CNN. The main sources of learnable parameters in a model are the convolutional layers and fully-connected layers. For the convolutional layer, the number of parameters is proportional to the number of input feature maps l, the number of output feature maps k and the height h and width w of the learned filters. It follows the following equation:

$$(h \times w \times l + 1) \times k$$

Note that the 1 is added to take into account the bias term. The heaviest layer is the fully-connected layer due to the fact that every neuron in the layer n has a connection to every other neuron in the previous layer m. The memory complexity can be computed as:

$$(n + 1) \times m$$

Again, the 1 accounts for the bias term. Pooling layers have no learnable parameters and batch normalization layers account for a negligible number of parameters.

The second metric we consider relates to the number of operations required by the model to process data. We use Floating Point OPerations (FLOPs) as a measure of how many operations are required to process a single data sample. We use this metric to give a hardware-independent measurement of how fast a model can be. This metric is particularly important in real-time applications which require fast analysis of data. This work mostly minimizes the memory complexity of our solution, which we argue is important due to the high cost of fast memory devices. Each layer the input data goes through adds to the number of operations for inference. Convolutional layers are the most expensive time complexity wise and their FLOPs can be computed as follows:

$$2 \times (h \times w \times l \times k)$$

Conversely, fully-connected layers are faster to compute and can be described as follows:

$$2 \times (n \times m)$$

Note here that we compute here the number of multiply-and-combine operations which count as two floating point operations which is why the term is multiplied by 2. Pooling layers are also fairly expensive though by reducing the dimensions of the feature maps they reduce complexity down the line. Their FLOPs can be expressed as:

$$\frac{h \times w}{s} \times d$$

Where h and w are the height and width of the input data, s is the stride of the layer, and d is the depth of the input data.

3.2.3 Domain Shift Metrics

An important aspect of the datasets in an MTDA setting is to evaluate how they relate to each other. Training a model for MTDA means the model should perform well on all targets simultaneously. It is intuitive to think that it is easier for the model to perform well on datasets that resemble each other. This work looks into a few ways to measure domain shifts between datasets. Rather than comparing the datasets themselves, we consider the embeddings of a model which was trained on a source domain. This way, we compute the shift going from a source domain to a target domain as shown in Figure 3.6. We repeat this process with every dataset as a source to obtain the shift for every source/target combination.

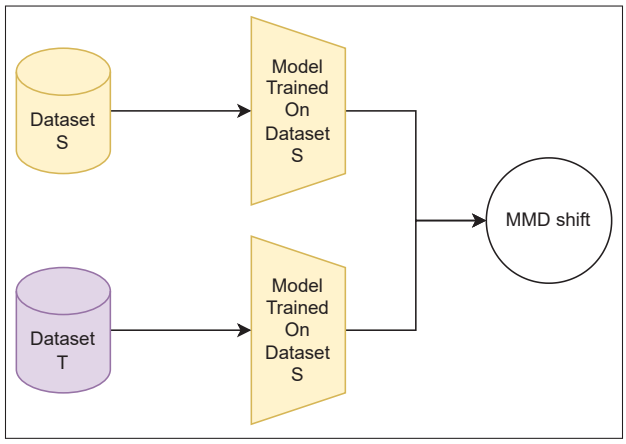

Figure 3.6 Configuration to compute the shift going from a source dataset (S) to a target dataset (T) using S as a basis

A common practice in Domain Adaptation is to attempt to align the data distribution statistics to reduce the performance drop when going from source to target data. Following this idea, we measure domain shift by measuring the distance between the distribution of embeddings coming from a source dataset and those coming from a target dataset. We compute the MMD as a metric of similarity between distributions of data points. We see in Table 3.3 that the MMD value is low when the source is less complex than the target, for instance when going from CUHK03 to MSMT17. The inverse is also true when going from a complex dataset to a simpler one, for example, DukeMTMC-ReID to CUHK03. These results a then a good metric of how hard it is to go from a source dataset to a target dataset.

Table 3.3 MMD between source and target data distributions encoded by a Resnet-50 trained in a supervised way on source

MMD Shift		Target			
		MSMT17	Market1501	DukeMTMCReID	CUHK03
Source	MSMT17	-	2.5297	2.4176	2.5446
	Market1501	1.1067	-	2.1497	2.7605
	DukeMTMCReID	1.3738	2.7525	-	2.8044
	CUHK03	0.9638	2.4370	1.9275	-

3.3 Training Protocol

The training setting described in this section is common to all our experiments unless specified otherwise. The code is developed by extending an existing Pytorch person ReID framework, namely Torchreid (Zhou & Xiang (2019)). We study two main architectures that were presented in Section 1.2.1.1 and Section 1.2.1.2, the Resnet architecture and the OSNet architecture. Both these architectures come in versions with varying complexity. We choose the Resnet50 architecture for all teacher networks throughout our experiments. In Chapter 4, we use only Resnet architectures for the student models as we aim to fully understand the method and compare it to the literature. In Chapter 5, we use the OSNet architectures which are specifically designed for person ReID as we are studying ways to balance good accuracy with low model complexity. A common variant to CNN architectures is to add a fully-connected layer between the classification layer and the convolutional blocks to further transform the features. We employ two consecutive fully-connected layers of dimensions 512 for all models. All bounding boxes are resized to 256x128 pixels so we can make mini-batches and have a constant output feature dimension. We use the following data augmentations for training:

- Random horizontal flip of the image
- Random crop of a region of the image
- Random erasing of a section of the image

A fully connected layer is added at the end of the CNN which sets the output feature dimension to 2048 for any Resnet architecture. We use two base STDA techniques to evaluate the MTDA approaches: SPCL (Ge *et al.* (2020)) and D-MMD (Mekhazni *et al.* (2020)). The optimizer and and scheduler parameters for each step of training are listed in Table 3.4.

Table 3.4 Experimental configuration for optimizer and scheduler at different steps of the algorithm

Process	optimizer	learning rate	scheduler	step	gamma
Supervised pre-training	adam	0.0003	single_step	50	0.1
UDA with D-MMD	adam	0.0003	single_step	50	0.1
UDA with SPCL	adam	0.00035	single_step	20	0.1
KD	SGD	0.01	single_step	5	0.1

The hyperparameters specific to the SPCL and D-MMD techniques are kept the same as in their original papers (Ge *et al.* (2020), Mekhazni *et al.* (2020)). Some steps of the training require both sources and target data simultaneously. We use a batch size of 64 where half the samples are from the source domain, and the other half are from the target domain. Each batch is composed of groups of four images with the same identity corresponding to tracklets. For every step of the algorithm, the training lasts until the model's accuracy converges, meaning the average accuracy over all target domains does not increase in a window of 5 epochs.

3.4 Baseline MTDA Approaches

MTDA is a mostly unexplored problem in literature, especially in person ReID. To measure the effectiveness of our proposed method we compare two baseline naive approaches: Multiple STDA Models and STDA on Blended Targets. The first one, illustrated in Figure 3.7 a., is the most straightforward approach. We simply apply the base STDA techniques N times for the N targets and we save every resulting model. Each model is used to test its corresponding dataset. This approach has a high complexity due to saving many models but good results. The second baseline, seen in Figure 3.7 b., is to combine all the target datasets into a single big dataset and perform an STDA technique on the resulting data. No information about which dataset the data come from is used during training. The test set, however, remain separated and the common backbone is used for all tests. This approach has low complexity but at the cost of accuracy. We will use these baselines in our evaluation of our proposed approach in the next chapter.

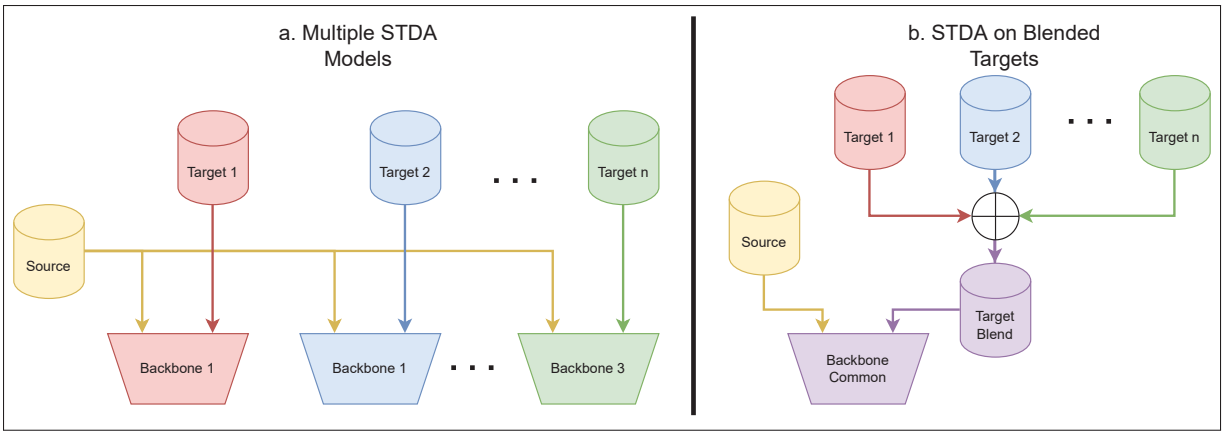

Figure 3.7 Overview of MTDA approaches. a) A model is adapted for each for each target using an STDA technique. b) The target domain datasets are first combined to form a single dataset and a model is adapted using an STDA technique

CHAPTER 4

MULTI-TARGET DOMAIN ADAPTATION FOR PERSON RE-IDENTIFICATION

Multi-target domain adaptation is a relatively unexplored field of research in computer vision and there are no techniques developed specifically for the problem of person ReID. In the following sections, we will present a proposed technique to tackle this problem. We will then provide a complete study of the proposed approach and how it compares to the MTDA baselines we presented in the previous chapter.

4.1 Proposed Knowledge Distillation Approach

Our proposed approach is inspired by the KD-MTDA technique presented by (Nguyen-Meidine *et al.* (2020)). That technique, however, was developed for classification problems and requires some modifications to perform well on ReID tasks. Because ReID tasks require a Siamese network rather than a single branch network we opt for a different knowledge distillation loss function. (Nguyen-Meidine *et al.* (2020)) uses a feature-based distillation approach (initially developed by (Heo, Kim, Yun, Park, Kwak & Choi (2019))). We opt for a similarity matrix distillation which is more appropriate in our case, similar to (Wu *et al.* (2019a)). (Nguyen-Meidine *et al.* (2020)) also distills information from source data which makes sense as the source, in their case, shares some classes with the targets. In our case, however, there are no shared identities between datasets. Finally, (Nguyen-Meidine *et al.* (2020)) jointly train the teacher and distill to the student. Our experiments show that this does not improve the distillation so we chose to perform the two steps one after the other. A side-effect of this change is that we can easily swap teachers and even have teachers trained with different STDA techniques. The approach is illustrated in Figure 4.1

The approach is a three-step process:

- Supervised pre-training of Student and Teacher models
- Training of the teacher models using a STDA technique
- Distillation of the knowledge from the fully adapted teacher models to the student model

The full algorithm is described in more detail in algorithm 4.1

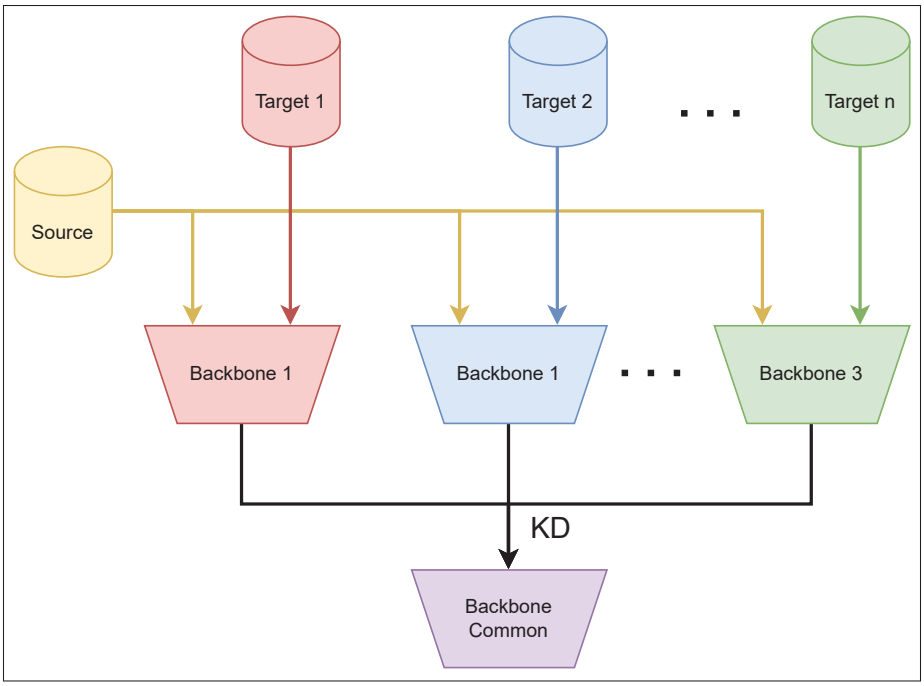

Figure 4.1 The proposed approach named KD-ReID combines multiple specialized teachers in a single common model

Considering labeled samples from the source domain, $\mathbf{x}^s \subset \mathbf{X^s}$, and unlabeled samples from the target domain, $\mathbf{x}^t \subset \mathbf{X^t}$, we define a single student model as Θ and a set of teacher models, adapted to each target domain, as $\mathbf{\Phi} = \{\Phi^1, \Phi^2, ..., \Phi^t\}$ where t is the number of target domains. For the first step, we train both the student and teacher models on source data $\mathbf{x}^s \subset \mathbf{X^s}$ using two supervised losses: $\mathcal{L}_{ces}$ which is a cross-entropy loss with a label smoothing regularizer (Szegedy, Vanhoucke, Ioffe, Shlens & Wojna (2016)) and $\mathcal{L}_{tri}$ which is a triplet loss with hard positive/negative sample mining as proposed by Hermans *et al.* (2017). More formally, $\mathcal{L}_{ces}$ is defined as:

$$\mathcal{L}_{ces} = (1 - \epsilon) \cdot \mathcal{L}_{ce} + \frac{\epsilon}{C}$$

Where C is the number of classes, $\epsilon \in [0, 1]$ is a hyper-parameter to control the label smoothing and $\mathcal{L}_{ce}$ is the classical cross-entropy loss. For each sample γ in the batch, the triplet loss is defined as:

$$\begin{aligned}\mathcal{L}_{tri} = \sum_{i=1}^{P}\sum_{\gamma=1}^{K}[m + \max_{p=1\ldots K} d(\Pi(\mathbf{x}_i^{\gamma}, \mathbf{x}_i^{p})) \\ - \min_{\substack{j=1\ldots P \\ n=1\ldots K \\ j\neq i}} d(\Pi(\mathbf{x}_i^{\gamma}, \mathbf{x}_j^{n}))]_+\end{aligned} \tag{4.1}$$

where, $\Pi(.)$ to be the features extracted right before the fully connected layers of model Π, m is a hyperparameter margin, and $d(\cdot, \cdot)$ is the euclidean distance. γ is the anchor sample while p and n indicate a positive, same-identity, sample or a negative, different-identity, sample, respectively.

The teacher models are then adapted to their respective target domains using an STDA technique. In our case, we use teachers trained with the D-MMD and SPCL techniques that were detailed in Sections 2.2.1 and 2.2.2.

Finally, once the teacher models are fully trained, we distill the knowledge from the teacher models to the student. The target which is being distilled alternates every mini-batch to ensure that the model does not forget information from datasets trained earlier in the process. The order in which the targets are distilled is chosen before each round of N batches, where N is the number of target domains.

Given $\mathbf{A_s}$ and $\mathbf{A_t}$ which are the cosine similarity matrices computed using the output vectors of the student and teacher model, respectively, the loss function is as follows:

$$\mathcal{L}_{KD} = ||\mathbf{A_s} - \mathbf{A_t}|| \tag{4.2}$$

Where $||\cdot, \cdot||$ is the matrix Frobenius norm. Minimizing this loss allows the student model to produce distance matrices similar to that of the teacher.

4.2 Experiments and Discussion

In this section, we present the experiments allowing us to evaluate the proposed approach compared to the baseline presented in Section 3.4. First, we present the main results, which give the clearest

Algorithm 4.1 DA and KD procedure

Require: labeled source data $\mathbf{x}^s$, unlabeled target data $\mathbf{x}^t$,
Pre-train a set of teachers models $\mathbf{\Phi} = \{\Phi^1, \Phi^2, ..., \Phi^T\}$ on source data
Pre-train a student model Θ on source data
for $\Phi^t \in \mathbf{\Phi}$ **do**
  while Φ *not converged* **do**
    for *each mini-batch* $B^s \subset \mathbf{x}^s$ *and each mini-batch* $B^t \subset \mathbf{x}^t$ **do**
      Compute $\mathcal{L}_{DA}$ and optimize the teacher model Φ
    end for
  end while
end for
while Θ *not converged* **do**
  for $\Phi^t \in \mathbf{\Phi}$ **do**
    for *each mini-batch* $B^s \subset \mathbf{x}^s$ *and each mini-batch* $B^t \subset \mathbf{x}^t$ **do**
      Compute $\mathcal{L}_{KD}$ and optimize the student model Θ
    end for
  end for
end while

overview of which approach performs best in a general case. Secondly, we present results when the MTDA configuration is changed. These results show which approach performs best for specific more difficult MTDA problems (limited model capacity, small source domain, etc...). Thirdly we will go into more detail about how different configurations of our proposed approach affect performance. Then we will present the results of an interesting use case of our approach allowing the use of teacher models trained with different STDA techniques simultaneously.

4.2.1 Overall Performance

This section presents a broad overview of how the proposed approach KD-ReID compares with the baselines (methods a. and b. in Figure 3.7). Two main aspects of the approaches are evaluated: how lightweight the final model is and how accurate the model is on the target datasets.

An important aspect to consider when comparing the proposed approach to baselines is the inference time for a single sample and the number of weights to be kept in memory. We evaluate these measures of complexity for the final models required to perform person ReID once the training is complete. For

method a. we consider the model for each target while we only consider the common backbone for methods b. and c.. Table 4.1 shows the number of parameters required to store the model in memory as well as the FLoating point Operations (FLOPs) required to process a single sample for the various backbone architecture we use in the experiments of this chapter. We refer to those statistics as memory complexity and inference time respectively.

Table 4.1 Memory complexity and inference time of the final model(s) for each model architecture

Model Architecture	# of parameters	FLOPs / sample
Resnet50	27.7M	2.70G
Resnet34	22.3M	2.39G
Resnet18	12.2M	1.19G
MobileNetV2	8.0M	0.4G
OSNet_x1_0	3.0M	1.00G
OSNet_x0_75	1.9M	0.6G
OSNet_x0_5	1.0M	0.3G
OSNet_x0_25	0.4M	0.08G

The first thing to note when evaluating the complexity of the approaches is that method a.'s memory complexity grows with the number of target domains. Because method a. requires a complete backbone for each target, the number of parameters held in memory increases with the number of target domains. The inference time however is unaffected by the number of targets. Methods b. and c.'s complexities only depend on the backbone architecture chosen. Note however that for method c. the teacher backbones can be very large without increasing the approach's complexity because only the common student backbone is held in memory for inference. Another thing to notice is that the Resnet18 architecture has under half the parameters and FLOPs as the Resnet50 model. Because we value the computing resources required in evaluating the best approach, we chose to use the Resnet18 as the base common backbone architecture for our main experiments even if it might yield slightly lower performance. For method a. however, we use Resnet50s to allow a fair accuracy comparison with method c. which uses Resnet50s as its teacher backbones. Notice that the OSNet architectures have significantly lower complexity. In this chapter, we choose to focus on the Resnet architecture as it is a general-purpose architecture used in many applications and research materials. This allows a fair evaluation of our method when compared to approaches in the literature.

For accuracy, we consider the performance metrics presented in Section 3.2.1 on each dataset as well as the average for each metric. While averaging accuracy metrics over all datasets is not a very indicative way to measure how good an approach is, it does provide a clear indication when comparing two MTDA approaches. All three MTDA approaches studied are extensions of STDA techniques. We reproduce our experiments using the two base STDA techniques that were presented in Section 3.4 to show how they interact with the MTDA approaches. Table 4.2 shows the main results of our experiments. For KD-ReID, ResNet50 implements the target-specific CNN backbones, and Resnet18 implements the common student CNN backbones. The lower bound performance is obtained through supervised training of Resnet18 on the labeled source dataset only, and the upper bound after supervised fine-tuning on blended target datasets. FLOPs are related to the extraction CNN features for one image sample

Table 4.2 Performance of MTDA methods when **MSMT17** is used as the source dataset, and **Market1501**, **DukeMTMC**, and **CUHK03** as target datasets (T = 3 targets), with 2 STDA techniques – D-MMD and SPCL.

MTDA Method – Base STDA Method	Accuracy on Target Data (%)								Complexity	
	Market1501		DukeMTMC		CUHK03		Average			
	mAP	R1	mAP	R1	mAP	R1	mAP	R1	# Parameters	FLOPs
Lower Bound: Superv. on Source Only	27.7	54.6	30.1	49.5	27.8	32.0	28.5	45.3	12.2 M	1.19 G
One Model per Target – D-MMD (Teachers)	51.4	74.9	51.4	69.3	**61.8**	**65.9**	**54.9**	**70.0**	T x 27.7 M	2.70 G
Blending Targets – D-MMD	40.3	64.5	42.2	61.8	54.2	58.0	45.6	61.4	12.2 M	1.19 G
KD-ReID – D-MMD (Ours)	48.9	71.9	48.9	66.9	58.0	61.7	51.9	66.5	12.2 M	1.19 G
One Model per Target – SPCL (Teachers)	54.2	75.3	**52.0**	69.6	33.4	34.8	45.9	59.9	T x 27.7 M	2.70 G
Blending Targets – SPCL	54.8	76.2	51.6	**70.2**	38.2	42.0	48.2	62.8	12.2 M	1.19 G
KD-ReID – SPCL (Ours)	54.1	75.2	46.5	65.7	38.1	41.1	46.2	60.7	12.2 M	1.19 G
KD-ReID – Mixed D-MMD & SPCL (Ours)	**55.2**	**76.3**	50.5	68.8	53.5	57.8	53.1	67.6	12.2 M	1.19 G
Upper Bound: Superv. Fine-Tuning on Targets	65.7	86.1	60.5	77.2	65.9	68.5	64.0	77.3	12.2 M	1.19 G

We use lower and upper bound experiments to give more context to our results. The lower bound consists of training a student model in a supervised way on source data and testing directly on data from the target domains. For the upper bound experiment we train the model on a blend of target labeled target data. We use the supervised loss to train the upper bound model. As expected the lower bound has lower accuracy than all approaches considered and the upper bound outperforms all other results. One thing to note is that the results for "One Model per Target" correspond to the performance of teacher models used in the KD-ReID approach. This accuracy acts as a soft upper bound for the KD-ReID method as the student model attempts to mimic the output of teacher models.

The results for one model per target represent the most naive approach where we train multiple models which are specialized for data from a specific domain. We see that while the accuracy is quite good, the

model complexity is much higher than for the other two approaches considered. This gap in complexity grows larger as we increase the number of target T.

The second approach (Blending Targets) interacts differently with the two base STDA techniques. Looking at the D-MMD results first, we notice that the blending approach gives the lowest accuracy of all considered approaches. It does however have the benefit of having a significantly lower complexity than the one model per target approach which is constant for any number of targets. In the case of SPCL however, the Blending approach yields the highest accuracy of all three approach. This can be due to the high compatibility of the SPCL technique with blended datasets. the SPCL technique is a clustering algorithm that naturally separates samples from different datasets into different clusters. This could allow the technique to actually benefit from a larger amount of different clusters to better separate different identities. Still, the SPCL has very poor performance on the CUHK03 dataset. The technique was not tested on this dataset originally and would probably require some tuning of hyperparameters to perform well on this dataset. This low accuracy is consistent for all three approaches.

Finally, our proposed approach KD-ReID performs very well when using D-MMD as the base STDA technique. The complexity is the same as for the blending approach as we use a single Resnet18 common model. The accuracy is much higher than for the blending approach and is relatively close to the teacher's accuracy. For SPCL however, the accuracy is slightly lower on all targets than the blending approach. Again we notice that the weak CUHK03 SPCL teacher transfers weak knowledge to the student which translates into a weak accuracy on CUHK03 specifically. This motivates the final experiment we consider: Mixed Teachers KD-ReID. For this experiment, we use the best teacher possible for each dataset. In this case, we use SPCL-trained models for the Market1501 and DukeMTMCReID datasets and a D-MMD trained model for the CUHK03 dataset. This yields the highest accuracy of all experiments. This showcases our approach as the most flexible and therefore the easiest to use in real-world situations. Instead of calibrating a single technique to perform well on all datasets simultaneously, we can simply take the best model with the best hyperparameters for each target and combine them using the KD-ReID technique. This avoids the need for costly parameter searches and allows techniques to be deployed significantly faster. It is interesting that while SPCL (Clustering) and D-MMD (Distribution Matching) function very differently, the knowledge of the resulting teachers can still be combined through distillation. Table 4.3 shows in more detail different teacher configurations for mixed teachers.

Table 4.3 Result of using a mix of teachers adapted using D-MMD and teacher using SPCL

Teachers (Student: Resnet18)	Market1501		DukeMTMCReID		CUHK03		Average	
	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Market1501: Resnet50 D-MMD DukeMTMCReID: Resnet50 D-MMD CUHK03: Resnet50 D-MMD	48.9	71.9	48.9	66.9	**58.0**	**61.7**	51.9	66.5
Market1501: Resnet50 SPCL DukeMTMCReID: Resnet50 SPCL CUHK03: Resnet50 SPCL	54.1	75.2	46.5	65.7	38.1	41.1	46.2	60.7
Market1501: Resnet50 SPCL DukeMTMCReID: Resnet50 SPCL CUHK03: Resnet50 D-MMD	**55.2**	**76.3**	**50.5**	**68.8**	53.5	57.8	**53.1**	**67.6**

We also compare our technique with the most relevant work in the literature. No other work presents results using all four mainstream datasets at once so we present results matching their configuration. Table 4.4 shows that our approach outperforms both current techniques tackling MTDA for person ReID. Our technique also uses a smaller backbone with a Resnet18 while the other techniques use a Resnet50.

Table 4.4 Comparison of KD-ReID with related works from literature

MTDA Technique	Configuration (Source -> Targets)	Target1		Target 2		Complexity	
		mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	# Parameters	FLOPs
Tian *et al.* (2021)	MSMT17 -> Market1501 & DukeMTMCReID	35.9	70.4	33.6	57.2	23.5M	2.70G
KD-ReID (Ours)	MSMT17 -> Market1501 & DukeMTMCReID	49.8	73.6	49.2	67.0	11.2M	1.2G
Mohanty *et al.* (2022)	Market1501 -> DukeMTMCReID & CUHK03	35.4	55.0	32.7	33.5	23.5M	2.70G
KD-ReID (Ours)	Market1501 -> DukeMTMCReID & CUHK03	41.6	60.0	58.8	63.6	11.2M	1.2G

To summarize, KD-ReID has three main advantages over the other techniques and baselines that perform MTDA for person ReID. First, KD-ReID produces the model with the highest accuracy of all techniques excluding the one model per target baseline. Second, KD-ReID produces a solution that has less memory complexity than all other approaches and is significantly more practical than the one model per target baseline. Finally, KD-ReID is a very versatile technique that can be used to extend any STDA technique and can even combine different techniques to optimize average accuracy.

4.2.2 MTDA Configuration

In this section, we show how the approaches compare when the MTDA problem parameters vary. First, we study how Student model complexity affects accuracy. Then we evaluate the scalability of the approaches, meaning how well they perform when the number of target domains increases. Finally, we change the source dataset to evaluate the impact of more labeled samples on final model accuracy.

Table 4.5 and 4.6 show the detailed accuracy results of the common model when using different architectures for the Blending and KD-ReID approaches respectively. A more concise summary of these results can be seen in Figure 4.2 and 4.3. In Figure 4.2, we observe that KD-ReID outperforms the blending for every architecture. Furthermore, KD-ReID reaches peak performance when using the Resnet50 architecture. This can be explained by the fact that Resnet18 and Resnet34 have too few parameters to fully contain knowledge from all teachers. On the other end, Resnet101 might have too many parameters making it too difficult to train efficiently. Looking at Figure 4.3, we notice that the blending performance degrades significantly as the models become less complex while KD-ReID is relatively less affected. This shows our technique would scale better in a resource-limited application.

Table 4.5 Detailed accuracy variations due to model complexity for the Blending approach. The teachers are Resnet50 models trained with the D-MMD base STDA technique

Architecture of Common Model	**Accuracy on Target Data (%)**								**Complexity**	
Blending of Targets	**Market1501**		**DukeMTMC**		**CUHK03**		**Average**			
Base STDA Method: D-MMD	**mAP**	**R1**	**mAP**	**R1**	**mAP**	**R1**	**mAP**	**R1**	**# Parameters**	**FLOPs**
Resnet101	44.9	69.4	45.7	64.0	55.9	60.2	48.8	64.5	44.5M	7.6G
Resnet50	43.0	67.1	45.8	64.9	54.6	59.7	47.8	63.9	27.7M	2.4G
Resnet34	42.9	66.6	43.1	62.4	54.4	59.6	48.5	62.9	21.3M	2.4G
Resnet18	40.3	64.5	42.2	61.8	54.2	58.0	45.6	61.4	11.2M	1.2G
MobileNetV2	42.1	65.8	43.1	63.0	52.4	56.3	45.9	61.7	4.3M	0.4G
OSNet_x1_0	52.3	77.2	52.4	70.0	57.6	62.3	54.1	69.8	3.0M	1.0G
OSNet_x0_5	45.0	71.4	46.2	64.0	49.9	53.6	47.0	63.0	1.0M	0.3G
OSNet_x0_25	32.5	59.5	35.5	55.6	35.8	40.5	34.6	51.9	0.4M	0.08G

Table 4.6 Detailed accuracy variations due to model complexity for the KD-ReID approach. The teachers are Resnet50 models trained with the D-MMD base STDA technique

Architecture of Common Model	**Accuracy on Target Data (%)**								**Complexity**	
KD-ReID	**Market1501**		**DukeMTMC**		**CUHK03**		**Average**			
Base STDA Method: D-MMD	**mAP**	**R1**	**mAP**	**R1**	**mAP**	**R1**	**mAP**	**R1**	**# Parameters**	**FLOPs**
Resnet101	51.6	73.6	52.1	69.7	58.9	62.7	54.2	68.6	44.5M	7.6G
Resnet50	52.8	75.8	53.9	71.7	61.0	64.0	55.9	70.5	27.7M	2.4G
Resnet34	51.7	75.2	51.7	69.8	58.2	62.0	53.9	69.0	21.3M	2.4G
Resnet18	48.9	71.9	48.9	66.9	58.0	61.7	51.9	66.5	11.2M	1.2G
MobileNetV2	49.3	72.8	49.5	67.2	58.0	61.8	52.3	67.3	4.3M	0.4G
OSNet_x1_0	54.4	76.9	54.8	71.5	61.1	65.2	56.8	71.2	3.0M	1.0G
OSNet_x0_5	52.7	75.6	53.6	71.4	57.7	61.7	54.7	69.4	1.0M	0.3G
OSNet_x0_25	47.8	71.5	49.3	68.3	53.3	57.4	50.1	65.7	0.4M	0.08G

Next, we study the effect of varying the number of target domains considered during training. A good solution should have minimal loss in performance as we add more targets. Table 4.7 shows minimal

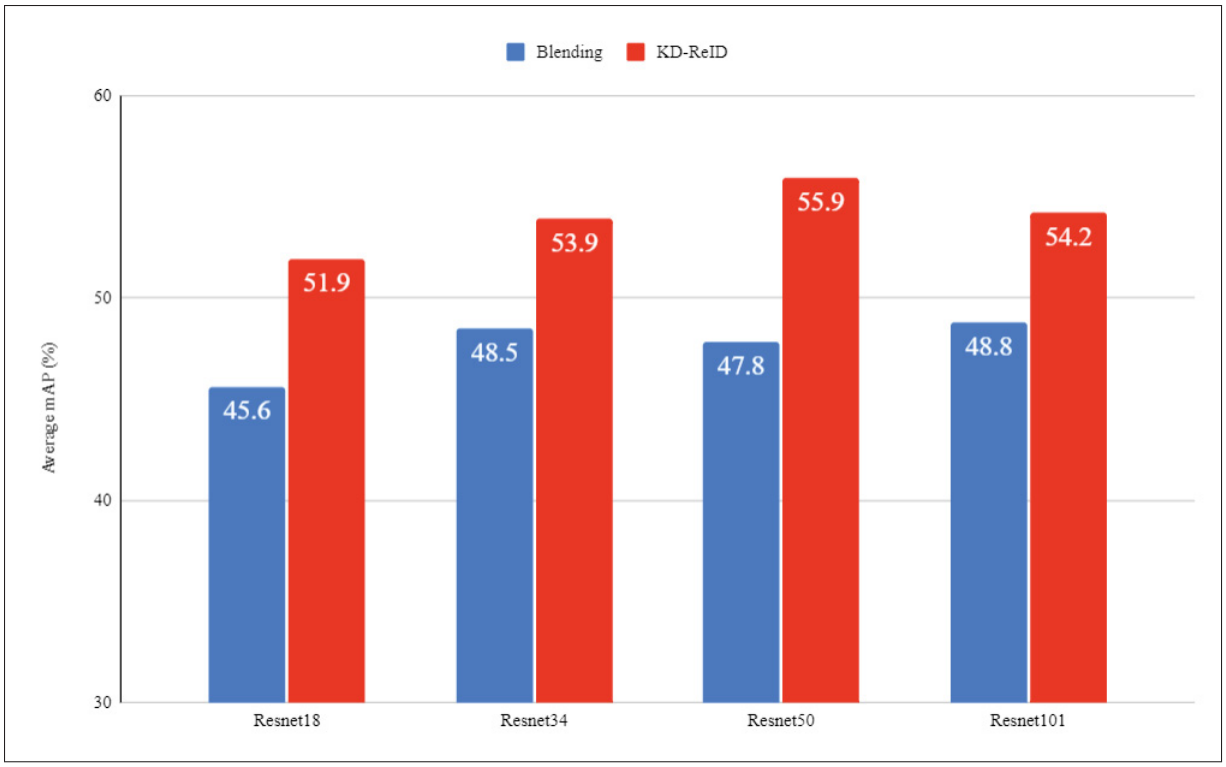

Figure 4.2 Variations of Average performance for Resnet architecture variants

performance drops across the board as the number of targets increases for KD-ReID. In comparison, the blending results presented in Table 4.8 show a larger performance drop when going from 1 target to 2. This shows KD-ReID scales better for problems with more target domains.

Table 4.7 The impact of varying the number of target dataset the model is trying to adapt to (M = Market1501, D = DukeMTMCReID, C = CUHK03) with KD-ReID. These experiments are conducted using a resnet18 student and resnet50 teachers trained with D-MMD

Targets	Market1501		DukeMTMCReID		CUHK03	
(Source:MSMT17)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Teachers	51.4	74.9	51.4	69.3	61.8	65.9
M	49.2	71.7	-	-	-	-
D	-	-	**50.0**	67.8	-	-
C	-	-	-	-	**60.1**	**64.2**
M + D	**49.8**	**73.6**	49.2	67.0	-	-
M + C	47.9	70.9	-	-	59.7	**64.0**
D + C	-	-	49.5	**68.2**	58.1	62.0
M+D+C	48.9	71.9	48.9	66.9	58.0	61.7

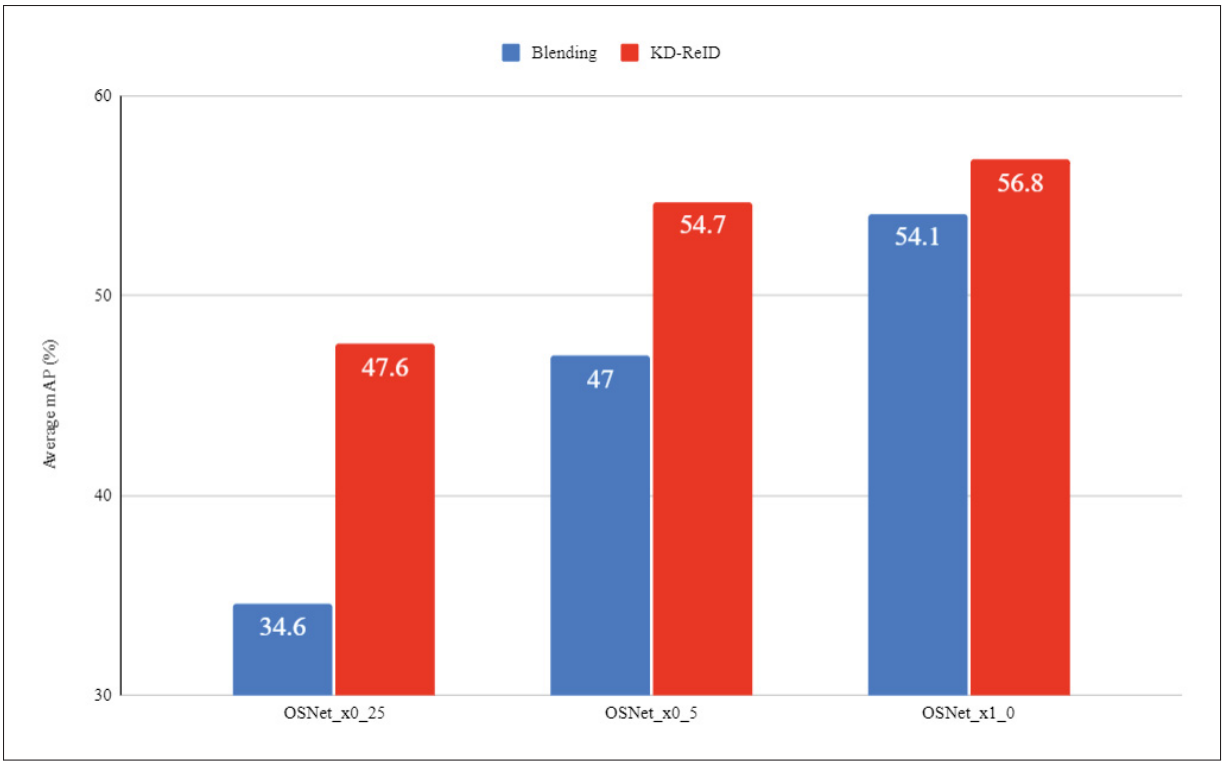

Figure 4.3 Variations of Average performance for OSNet architecture variants

Table 4.8 The impact of varying the number of target datasets the model is trying to adapt to (M = Market1501, D = DukeMTMCReID, C = CUHK03) with Blending. These experiments are conducted using a Resnet18 trained on the blended datasets with the D-MMD technique

Targets	Market1501		DukeMTMCReID		CUHK03	
(Source:MSMT17)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Teachers	51.4	74.9	51.4	69.3	61.8	65.9
M	**49.2**	**71.7**	-	-	-	-
D	-	-	**50.0**	**67.8**	-	-
C	-	-	-	-	**60.1**	**64.2**
M + D	40.5	65.1	43.1	62.9	-	-
M + C	39.9	64.5	-	-	55.7	61.9
D + C	-	-	42.9	62.7	54.6	60.1
M+D+C	40.3	64.5	42.2	61.8	54.2	59.6

Table 4.9 shows results when we use a smaller dataset as the source domain, namely Market1501 as the source domain in this case. As expected, results on MSMT17 are very weak as it is a much more complex

dataset than the other three. We do note however that results on DukeMTMCReID and CUHK03 do not degrade significantly when using a smaller source dataset. This shows that our technique is suitable for use with more limited amounts of labeled data. It is also interesting to note that the very weak performance on MSMT17 does not degrade the performance on the other target datasets. On the contrary, adding MSMT17 to the targets improves the performance on the less complex datasets. A link can be made to the domain shift measures presented in the previous chapter. We noted that going from a simpler source dataset (Market1501) to a more complex one (MSMT17) yielded a low MMD measure. We see here that it translates into a low accuracy on the complex dataset.

Table 4.9 The impact of varying the number of target datasets the model is trying to adapt to (Ms = MSMT17, D = DukeMTMCReID, C = CUHK03). These experiments are conducted using a resnet18 student and a resnet50 teacher trained with D-MMD

Targets	MSMT17		DukeMTMCReID		CUHK03	
(Source: Market1501)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Teachers	15.8	34.1	51.3	69.3	68.5	72.7
Ms	**12.5**	**28.6**	-	-	-	-
D	-	-	**48.3**	**65.3**	-	-
C	-	-	-	-	**61.0**	**65.1**
Ms + D	12.1	27.0	47.3	64.4	-	-
Ms + C	11.6	24.8	-	-	60.4	64.8
D + C	-	-	41.6	60.0	58.8	63.6
Ms+D+C	11.2	25.2	47.6	65.5	59.8	64.1

4.2.3 KD-ReID Configuration

In this section, we present how different configurations of the proposed KD-ReID method perform and how we chose the optimal methodology to obtain the best accuracy.

First, we explored how ordering the target datasets during the distillation process would affect performance. We study three types of data ordering: fixed, random, and based on a domain shift metric. The fixed ordering simply keeps the same order throughout the training. The random ordering changes the order between each mini-batch. For the domain shift-based order we compute a domain shift measure as explained in Section 3.2.3. Using this measure between datasets we order them from easiest (low domain shift) to hardest (high domain shift). Looking at Table 4.10, we see that order has very little effect on performance. We choose to use the random order in our approach as the other two ordering methods have

some drawbacks. The fixed order skews the performance slightly towards the dataset seen last in the order which yields a more uneven performance. The domain shift-based method takes a long time to compute between batches and slows down training significantly.

Table 4.10 Impact of data ordering method. Fixed means the order stays the same throughout training with the indicated order (M = Market1501, D = DukeMTMCReID, C = CUHK03). Random means the order is chosen randomly at the start of every epoch. Domain shift-based uses domain shift to order the datasets from easiest to hardest at the start of every epoch. Results are obtained with a Resnet18 student and Resnet50 teachers trained with D-MMD

Data ordering method	Market1501		DukeMTMCReID		CUHK03		Average	
	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Fixed: M - D - C	48.4	71.1	49.2	67.8	56.9	61.1	51.5	66.7
Fixed: D - M - C	49.0	71.6	48.4	67.0	58.1	61.6	51.8	66.7
Fixed: C - M - D	48.5	71.4	49.1	68.9	56.2	60.0	51.3	66.8
Random	48.9	71.9	48.9	66.9	58.0	61.7	51.9	66.5
Domain Shift Based	49.3	72.4	48.4	65.8	58.2	61.5	52.0	66.5

Next, we study we alternate the datasets during distillation. Either we change the dataset every mini-batch or we go through a full dataset before switching to another. When alternating every batch, we revisit samples from smaller datasets so all samples from the largest dataset are seen. Alternating every batch proves the optimal configuration as it allows a more even performance across all targets.

Table 4.11 Performance for different target alternation schemes. The dataset being distilled switches every mini-batch or whenever all its samples are seen by the model

Target Alternation	Market1501		DukeMTMCReID		CUHK03		Average	
	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Every Batch	**48.9**	**71.9**	**48.9**	**66.9**	58.0	61.7	**51.9**	**66.5**
At the end of Dataset	47.3	71.3	45.9	64.5	**59.1**	**62.2**	50.8	66.0

While Nguyen-Meidine *et al.* (2020) use feature distillation for their technique we show that similarity matrix distillation is much better suited for person ReID. We see in Table 4.12 a much higher accuracy when distilling similarity matrices rather than intermediate features from the CNN or even a combination of both feature and similarity matrices.

In essence, we have designed KD-ReID specifically for the problem of person ReID by choosing the appropriate KD approach and optimizing how the data is used during the training of the student.

Table 4.12 Performance of different distillation techniques. The trachers are Resnet50s trained with D-MMD and the student is a Resnet18

Distillation Technique	Market1501		DukeMTMCReID		CUHK03		Average	
	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Teachers	51.4	74.9	51.4	69.3	61.8	65.9	54.9	70.0
Similarity Matrix Distillation	**48.9**	**71.9**	**48.9**	**66.9**	**58.0**	**61.7**	**51.9**	**66.5**
Feature Distillation	42.6	66.4	47.4	65.0	56.6	58.9	48.9	63.4
Both Combined	43.7	65.8	41.3	58.9	48.1	52.2	44.4	59.0

4.3 Conclusion

In this chapter, we presented our approach based on Knowledge Distillation to the MTDA problem for person ReID. We showed that our approach produces a more accurate model than baseline approaches as well as relevant techniques from the literature.

We also showed that our technique scales better for problems with more target domains. Our technique is also more effective when using smaller models which are better suited for security applications where computational resources are limited.

We also demonstrated the flexibility of our approach by using it to extend two different STDA person ReID techniques: D-MMD Mekhazni *et al.* (2020) and SPCL Ge *et al.* (2020). We further demonstrated the versatility of the approach by combining knowledge from teachers trained using different base STDA techniques. In real-world scenarios, this means we can simply take any combination of teacher models without having to fine-tune a technique for each target domain.

CHAPTER 5

DOMAIN-SPECIFIC MODELS FOR MULTI-TARGET DOMAIN ADAPTATION IN PERSON REID

In this chapter, we study alternative model architectures to improve performance on the multi-target problem on compact DL models. The technique used to train the models is based on KD, as studied in the previous chapter. The models we explore are domain-specific batch normalization (DSBN), multi-branch networks, parameterized networks, and distillation adapters. Each variation increases the complexity of the solution to a certain degree. The goal is to identify solutions that provide the best trade-off between accuracy and complexity.

5.1 Domain-specific Batch Normalization (DSBN)

This model variation was first proposed by Chang, You, Seo, Kwak & Han (2019) and is very common in the field of domain adaptation to separate source and target running statistics. BN layers capture important domain-specific information through their running statistics. Separating BN layers for each domain allows the model to produce more domain-invariant features by removing the domain-specific information through normalization. In our case, we separate every batch normalization layer in the model so data from each target domain goes through its target-specific batch-norm layers, as seen in Figure 5.1. The idea is to separate the running statistics of each domain. Formally, we can define Domain-Specific Batch Normalization as:

$$BN(x_i) = \frac{x_i - \mu_i}{\sqrt{\sigma_i^2}} \tag{5.1}$$

Where μ_i and σ_i are the mean and variance of a batch and x_i represents a sample from the batch, and i indicates the domain of the data. A mean and variance are computed for each domain separately. The separation of the batch normalization layers occurs right after the student model is pre-trained on the source. This model modification is simple and compatible with all of the other model variations. We want to determine if this simple modification can be effectively applied to the multi-target scenario.

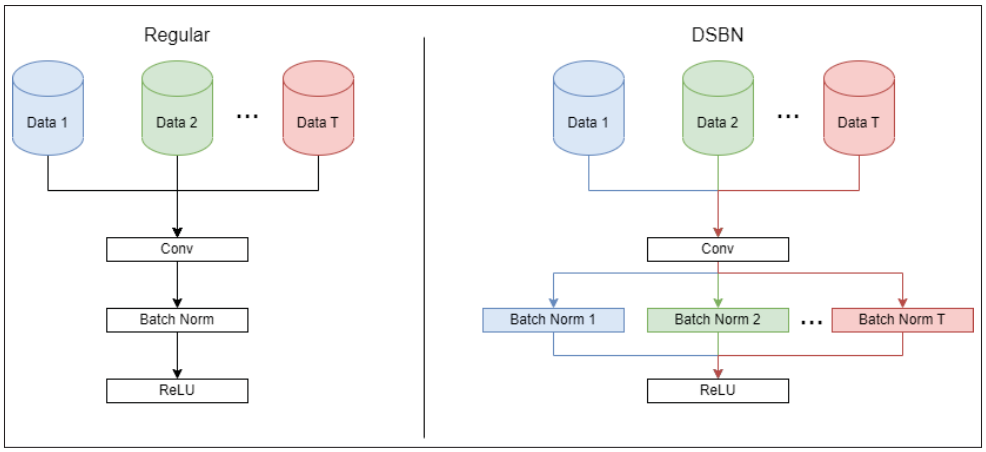

Figure 5.1 Regular BN vs DSBN

5.2 Multi-Branch Networks

These types of models have been used for multi-task and multi-target applications Vandenhende, Georgoulis, De Brabandere & Van Gool (2019); Zeng, Li, Li, Lu, Xu, Gu & Chen (2022); Neven, De Brabandere, Georgoulis, Proesmans & Van Gool (2017); Kokkinos (2017). It is a similar modification to DSBN, but instead, entire sections of the models are separated into domain-specific branches. When a part of the model is separated, it only sees samples from its corresponding target domain. Going forward, we will refer to the process of duplicating and training with specific target data as specializing a module.

We consider two main multi-branch configurations: multi-head and multi-tail. Multi-head is a common approach in multi-task and multi-target learning where the fully-connected head of the model is specialized to have one head per target, as seen in Figure 5.2b. The main idea is that a common features extractor extracts low-level features which are more general while multiple decoder heads learn domain-specific information allowing the model to perform well on each domain. We also consider the less common multi-tail configuration where the base layers of the network are specialized while the heads are common to all domains, as seen in Figure 5.2c. The idea behind this approach is that variations between the domains (illumination, resolution, weather) are expressed in the lower-level features, so earlier layers should be specialized. On the other hand, the domains are very similar at a higher level (they all contain images of pedestrians) and so the head of the model does not require to be specialized to a domain. The main downside of this type of model is that its number of parameters increases with the number of specialized modules and the number of target domains. The student architecture used in our experiments

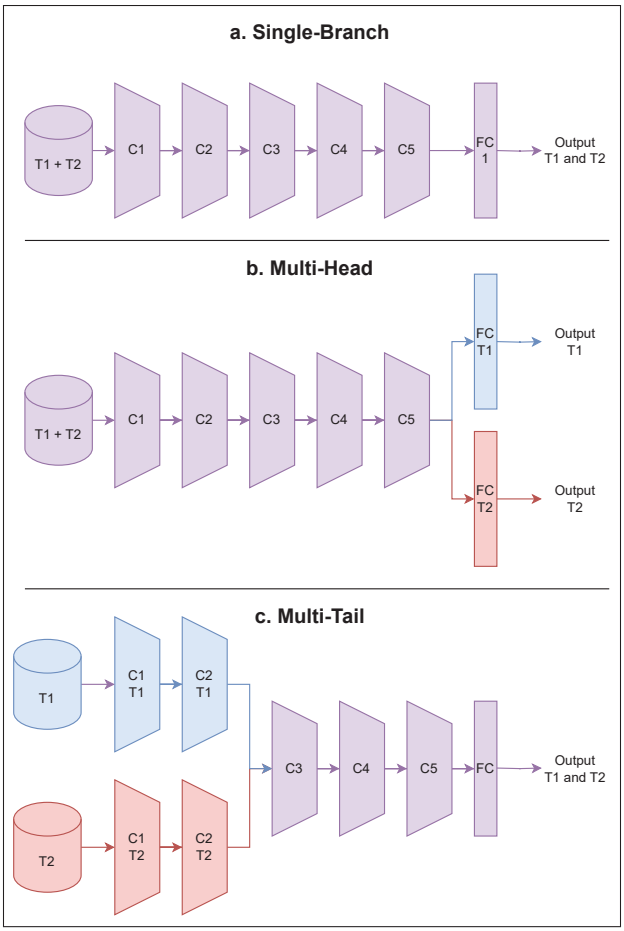

Figure 5.2 Multi-branch configurations

is OSNet_x0_25 which was described in detail in Section 1.2.1.2. We consider the model as 6 modules: 5 convolutional blocks denoted $C1$, $C2$ $C3$, $C4$, $C5$, and a fully-connected head denoted FC, as seen in Figure 5.2a, which can all be specialized. Formally we can decompose the full model Θ into consecutive blocks as:

$$\Theta(x^i) = \Theta^i_{FC}(\Theta^i_{C5}(\Theta^i_{C4}(\Theta^i_{C3}(\Theta^i_{C2}(\Theta^i_{C1}(x)))))) \tag{5.2}$$

Where Θ_{XX} is the specific module XX from the model Θ and i indicates the domain to which the data point x^i belongs. Note that for non-specialized modules, the parameters are the same regardless of i. The goal of these experiments is to determine which model layers benefit most from being target specific and,

in turn, determine the optimal level of the model branch by having a single model for all targets at one extreme and a separate model for each model at the other.

5.3 Parameterized networks

Parameterized networks are networks that use a small number of domain-specific parameters. Those parameters allow the model to specialize to specific data while keeping a large majority of parameters common to all domains. These networks use a smaller number of specialized parameters than the multi-head and multi-tail approach by adding specialized adapters in key parts of the model. We base our experiments on the technique presented in Rebuffi *et al.* (2018), which proposes two variants in terms of adapter placement. Formally, the series adapter is a 1x1 filter bank α in parallel to a skip connection, as seen in Figure 5.3, which can be expressed as:

$$\mathbf{y} = \mathbf{x} + BN(\alpha * \mathbf{x}) \tag{5.3}$$

where $*$ represents the convolution operation and BN the batch normalization layer application. x represents the input data to the adapter and y the output. It is interesting to note that if $\alpha = 0$, we recover the original signal $\mathbf{x}$. This allows controlling the impact of the adapters on the model in cases where adapters are not needed. The parallel configuration is placed in parallel to an existing convolutional layer $\mathbf{f}$ and can be written as:

$$\mathbf{y} = \mathbf{f} * \mathbf{x} + \alpha * \mathbf{x} \tag{5.4}$$

This second configuration has the same property. If $\alpha = 0$, we retrieve the original signal. Furthermore, this configuration does not require an additional BN layer as it uses the same BN layer as $\mathbf{f}$.

Note that we apply this technique on a model architecture (OSNet) that is different from the original model architecture presented in Rebuffi *et al.* (2018) (Resnet-26). We test three positions for the output of the adapters: after the conv layers, after the gating mechanism, and after the final residual connection, as seen in Figure 5.4. Note that there are actually four adapters for a residual OSNet block, one for

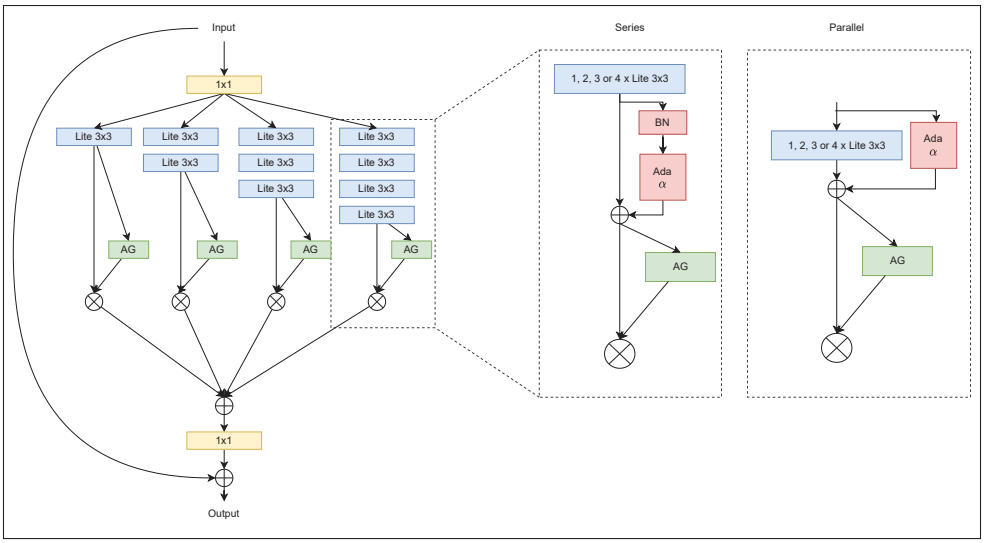

Figure 5.3 Two adapter types presented in this research applied to the OSNet residual block. The series adapter requires an additional BN layer while the parallel adapter leverages existing BN layers from the model.

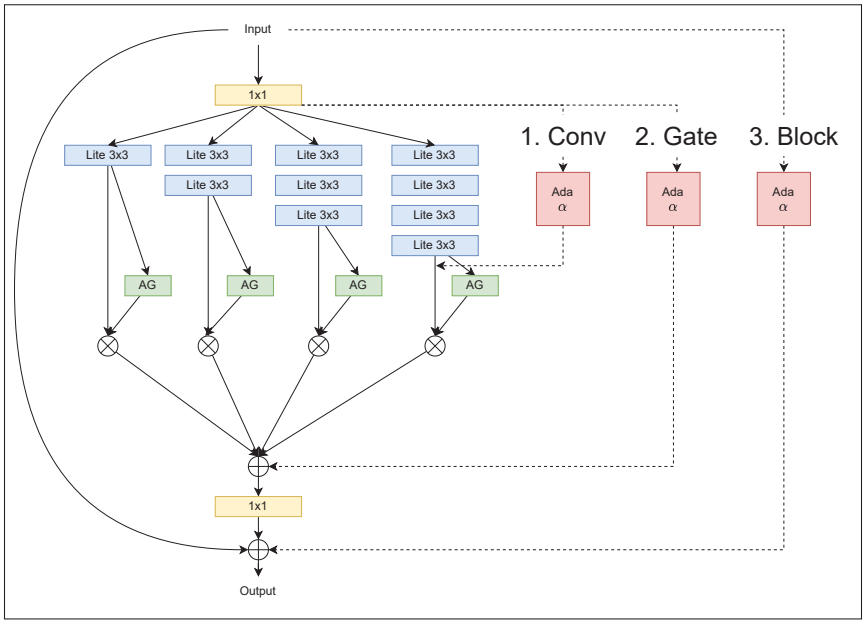

Figure 5.4 Adapter placement in the OSNet architecture

each branch. The adapters are trained with backpropagation like other model weights. The adapters are domain-specific and are only trained with data from their respective target domain. Unlike the technique presented in Rebuffi *et al.* (2018), the adapters are trained jointly with the rest of the model. The idea

is that the parameters learn domain-specific knowledge while the rest of the model focuses more on general knowledge. For this variation, we want to adapt the technique to the OSNet_x0_25 architecture by determining the best adapter placement and type. We then evaluate if the specialization of smaller adapter modules can be more efficient than specializing entire layer blocks as described in Section 5.2.

5.4 Distillation Adapters

Distillation adapters are a concept first developed for multi-task learning Li, Liu & Bilen (2022). The idea is to use a small adapter network to transform the output of the student model to ease the distillation process. This technique employs a training process similar to KD-ReID. We study two distillation adapter positioning illustrated as positions A and B in Figure 5.5. For position A, first individual teachers are trained separately using $\mathcal{L}_{DA}$ which is the domain adaptation loss, the DMMD loss in our case. The encoder portion of the model Φ^t_{Enc} corresponding to the convolutional layer portion of the model is separated from the decoder portion Φ^t_{Dec} which corresponds to the FC layers at the end of the model. Then using a distillation loss $\mathcal{L}_{KD-A}$, a student model encoder Θ_{Enc} is trained to mimic the output from Φ^t_{Enc} for all targets t. For $\mathcal{L}_{KD-A}$, we use a Euclidean distance between the outputs of both encoders. A domain-specific adapter β consisting of an FC layer is placed right before $\mathcal{L}_{KD-A}$ are used concurrently to train the student decoder Θ_{Dec}. The overall loss term for the student model can be expressed as:

$$\mathcal{L}_{PosA} = \mathcal{L}_{DA}[\Theta_{Dec}(\beta(\Theta_{Enc}(x^t)))] + \mathcal{L}_{KD-A}[\beta(\Theta_{Enc}(x^t)), \Phi^t_{Enc}(x^t)] \tag{5.5}$$

The adapter β is discarded during testing resulting in no additional parameters to the model.

The second position B has a similar training process. However, the adapter β is placed after the student decoder Θ_{Dec}, and the distillation loss $\mathcal{L}_{KD-B}$ is computed after the decoders. The loss becomes:

$$\mathcal{L}_{PosB} = \mathcal{L}_{DA}[\beta(\Theta_{Dec}(\Theta_{Enc}(x^t)))] + \mathcal{L}_{KD-B}[\beta(\Theta_{Dec}(\Theta_{Enc}(x^t))), \Phi^t_{Dec}(\Phi^t_{Enc}(x^t))] \tag{5.6}$$

Note that we may choose to keep the adapters during testing. Doing so, however, increases memory complexity significantly. Also, KD losses use only target domain data, while the DA loss uses both source and target data.

For this technique, we want to determine if we can adapt this multi-task technique to the multi-domain problem. We will evaluate if easing the knowledge distillation process is useful in the case where target domains are closer than task domains.

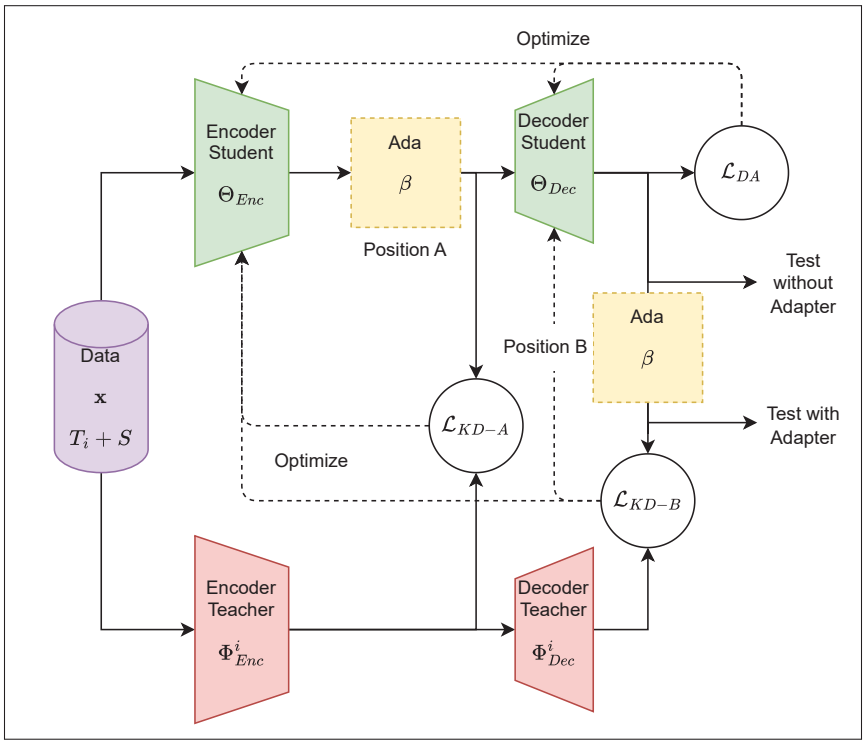

Figure 5.5 Shows our use of distillation adapters to align the student embeddings with a specific teacher's embeddings to improve the KD. Source data is only used when computing the DA loss.

5.5 Experiments and methodology

The main goal of this chapter is to provide a comparative study of all techniques presented in the previous sections. We also answer technique-specific research questions:

- Is DSBN an effective technique when applied to MTDA rather than the classical STDA scenario?
- Which layers are most important when training a multi-branch model?
- Can we effectively adapt the parameterized networks proposed by Rebuffi *et al.* (2018) to the OSNet_x0_25 architecture?

- Are the MTDA and multi-task problems similar enough that we can efficiently use a multi-task distillation technique Li *et al.* (2022) for our MTDA person ReID problem?

We experiment using the same experimental methodology from the previous chapter unless specified otherwise. To reduce the number of experiments, we focus on teacher models trained using the D-MMD approach. Since the focus of this chapter is to minimize the complexity of our solution, we experiment with the OSNet architecture for the student model, specifically OSNet_x0_25, as it is the most compact architecture we have studied in the previous section. Model variants presented in this chapter include domain-specific weights and weights common to all targets. When presenting model complexity, we will express the number of parameters as $(AxT) + B$ where A is the number of domain-specific parameters, T is the number of target domains considered and B is the number of parameters common to all domains. A higher value of A signifies the model does not scale well with the number of targets. The next section will provide the main results of our comparative study, while subsequent sections aim to answer the technique-specific questions in more detail.

5.6 Results and Discussion

5.6.1 Overall Comparison

In this section, we compare the studied modifications to determine the most cost-effective solution. We discuss in the following sections the results for each variation in more detail.

The first observation is that DSBN improves accuracy significantly for a very low increase in memory footprint. Due to the simplicity and effectiveness of this modification, we include it for all other modifications.

Multi-head has the worst complexity scaling of all model modifications. Duplicating the FC layer for each target increases the memory footprint dramatically. The accuracy of this method is also lower than using only DSBN. There is no advantage to using this modification.

Multi-tail has the highest average accuracy of all modifications being very slightly superior to the parameterized model. The complexity scaling is mediocre, being only better than the Multi-head modification.

Parameterized models offer an excellent compromise between memory footprint and accuracy. It has the second-best scaling behind DSBN and the second-best accuracy behind Multi-tail.

Finally, the use of distillation adapters degrades performance dramatically. While the complexity scaling is good, the accuracy makes it unusable for person ReID.

Overall, DSBN offers the best accuracy-to-complexity ratio increasing the number of domain-specific parameters by only 6K while increasing average accuracy by 4%. If accuracy is the main priority, a parameterized model is a good option, only increasing domain-specific parameters by 20K and increasing average accuracy by almost 5%. Multi-tail and Multi-head modifications are much heavier and probably not well-suited for real-world applications.

Table 5.1 Performance of each MTDA model considered in this study. The teachers are always Resnet50 models trained using the D-MMD method. The student is an OSNet_x0_25 with a fully connected head composed of two consecutive layers with 512 neurons. In the case of multi-branch approaches, the values in parenthesis represent which parts of the model are duplicated

Model Configuration	**Accuracy on Target Data (%)**								**Complexity**	
	Market1501		**DukeMTMC**		**CUHK03**		**Average**			
	mAP	**R1**	**mAP**	**R1**	**mAP**	**R1**	**mAP**	**R1**	**# Parameters**	**FLOPs**
Simple	47.8	71.5	49.3	68.3	53.3	57.4	50.1	65.7	467 k	82.6 M
DSBN	51.7	**75.1**	**51.4**	70.0	59.3	62.6	54.1	69.2	(6 x T) + 462 K	82.6 M
Multi-head (C45 + FC)	50.5	74.2	51.1	69.5	57.6	61.5	53.1	68.4	(409 x T) + 57 K	82.6 M
Multi-tail (C1234)	51.1	74	51.2	**70.4**	**61.6**	**64.9**	**55.0**	**69.8**	(126 x T) + 340 K	82.6 M
Parameterized model	**52.1**	**75.0**	**51.6**	69.8	61.1	63.6	**54.9**	**69.5**	(20 x T) + 462 K	82.6 M
Distillation Adapters	42.5	65.7	41.2	60.3	50.0	53.8	44.6	59.9	(6 x T) + 462 K	82.6 M

5.6.2 Multi-Branch Networks

In this section, we present more in-depth results for the multi-branch configurations. We start by looking at the impact of specializing individual parts of the model. That is, we duplicate a specific module and train that block using only data that belongs to its assigned target domain. Note that for this experiment, we do not use domain-specific batch normalization to evaluate the impact of individual modules better. In the experiments of this section, we indicate which modules are specialized i.e. C45 + FC means convolutional blocks 4 and 5 as well as the fully-connected head are all specialized while the rest of the model is common. Figure 5.6 and Table 5.2 report results for individual specialized modules. From Figure 5.6 we observe that mid-level blocks have more impact on results when specialized than early and

late modules. This can be explained by the fact that C234 are larger convolutional blocks in the OSNet architecture than C15. As for FC, we notice that specializing it has no impact on performance even though it represents a large portion of the parameters of the model. This could indicate that the fully-connected layer does not benefit from specializing to a specific domain and that high-level information between domains is similar.

Table 5.2 Performance when individual parts of the model are duplicated and specialized

Multi-Branch Configuration	Accuracy on Target Data (%)								Complexity	
	Market1501		DukeMTMC		CUHK03		Average			
	mAP	R1	mAP	R1	mAP	R1	mAP	R1	# Parameters	FLOPs
Simple	47.8	71.5	49.3	68.3	53.3	57.4	50.1	65.7	467 K	82.6 M
C1	46.4	70.4	49.2	66.7	55.6	58.4	50.4	65.2	(3 x T) + 463 K	82.6 M
C2	49.4	**73.2**	**50.6**	**69.1**	**56.9**	**59.9**	**52.3**	**67.4**	(18 x T) + 449 K	82.6 M
C3	49.4	72.7	50.3	**68.9**	**56.7**	**59.8**	**52.1**	67.1	(42 x T) + 425 K	82.6 M
C4	**49.6**	72.4	49.8	68.4	56.0	60.2	51.8	67.0	(57 x T) + 410 K	82.6 M
C5	49.3	72.8	49.8	68.8	54.1	57.7	51.1	66.4	(17 x T) + 450 K	82.6 M
FC	48.5	72.1	48.8	66.9	53.1	56.3	50.1	65.1	(329 x T) + 137 K	82.6 M

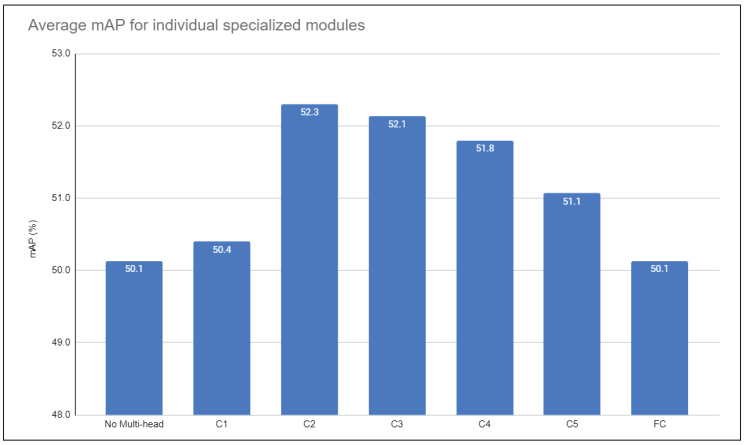

Figure 5.6 Impact of having single modules of the model duplicated and specialized.

Next, we experiment with the Multi-head configurations where the last layers of the model are specialized. Table 5.3 presents results for multi-head configurations with an increasing number of modules being included in the multi-head. Note that Simple means no parts of the model are specialized, while C12345 + FC means all modules are specialized (equivalent to having a separate model per target). We notice that performance increases as we add more modules to the multi-head. Notice that specializing the FC module has no impact on performance, while adding more and more convolutional blocks to the specialized heads has a positive impact on the results. These results are coherent with the results from Table 5.2 where

specializing convolutional blocks C234 has a greater impact. Notice that specializing the FC module increases model complexity dramatically while not improving accuracy. These results motivate the use of multi-tail configurations which start specializing in early modules first.

Table 5.3 Performance for multi-head configurations. These results are obtained without using domain-specific batch normalization

Multi-Branch Configuration	**Accuracy on Target Data (%)**								**Complexity**	
	Market1501		**DukeMTMC**		**CUHK03**		**Average**			
	mAP	**R1**	**mAP**	**R1**	**mAP**	**R1**	**mAP**	**R1**	**# Parameters**	**FLOPs**
Simple	47.8	71.5	49.3	68.3	53.3	57.4	50.1	65.7	467 K	82.6 M
FC	48.5	72.1	48.8	66.9	53.1	56.3	50.1	65.1	(329 x T) + 137 K	82.6 M
C5 + FC	49.7	**73.7**	49.5	68.5	53.4	57.1	50.9	66.4	(346 x T) + 120 K	82.6 M
C45 + FC	49.0	72.4	49.3	68.0	55.4	59.2	51.2	66.5	(403 x T) + 63 K	82.6 M
C345 + FC	49.6	72.5	50.5	68.5	56.4	60.8	52.2	67.3	(445 x T) + 21 K	82.6 M
C2345 + FC	**50.4**	**73.5**	**51.1**	68.6	56.5	60.6	52,7	67.6	(463 x T) + 3 K	82.6 M
C12345 + FC	**50.2**	72.4	**51.2**	**69.3**	**58.6**	**62.1**	**53.3**	**67.9**	(467 x T) K	82.6 M

In Table 5.4, we present the results of multi-tail configurations where an increasing number of modules are included in the specialized tails. We observe that results peak with the C1234 configuration. The main advantage of the multi-head configurations is that we do not specialize the FC module and therefore reduce significantly the complexity of the specialized model. Again we notice that adding C5 and FC to the specialized tail does not improve performance indicating that these modules do not benefit from learning domain-specific information. Furthermore, configuration C1234 has a higher average accuracy than C12345+FC meaning sharing some layers can actually be beneficial for the model.

The higher success of duplicating earlier convolutional blocks gives interesting information about the MTDA person ReID problem. Earlier layers in models are often thought to deal with low-level features such as shapes and colors while later layers take care of the higher levels of abstraction more related to the task, identifying pedestrians in this case. The fact that earlier layers are more useful in MTDA person ReID could indicate that the domain shift between target domains is mainly due to low-level differences between datasets. A parallel can be made with the great performance of DSBN. BN layers store statistics that aim to reduce low level-variations within a data distribution. If target domains have important low-level differences, it is logical to think that storing different statistics for each dataset allows a much better normalization.

Therefore, we conclude that earlier layers benefit more from target specialization than later layers. These results suggest that datasets differ at a lower level while they are similar at a higher task level. In turn, multi-tail configurations are more effective at producing domain-invariant features than multi-head configurations. This result is coherent with the success of DSBN which also affects lower-level features. We determine the configuration with the best compromise of memory footprint to accuracy to be C1234.

Table 5.4 Performance of the multi-tail configurations. These results are obtained without domain-specific batch normalization.

Multi-Branch Configuration	**Accuracy on Target Data (%)**								**Complexity**	
	Market1501		**DukeMTMC**		**CUHK03**		**Average**			
	mAP	**R1**	**mAP**	**R1**	**mAP**	**R1**	**mAP**	**R1**	**# Parameters**	**FLOPs**
Simple	47.8	71.5	49.3	68.3	53.3	57.4	50.1	65.7	467 K	82.6 M
C1	46.4	70.4	49.2	66.7	55.6	58.4	50.4	65.2	(3 x T) + 463 K	82.6 M
C12	49.8	73.2	50.5	69.0	57.8	61.2	52.7	67.8	(21 x T) + 445 K	82.6 M
C123	**50.5**	73.2	**51.3**	**69.4**	57.9	61.4	53.3	68.2	(63 x T) + 403 K	82.6 M
C1234	50.4	**73.8**	**51.2**	68.8	**59.4**	**63.2**	**53.7**	**68.6**	(120 x T) + 346 K	82.6 M
C12345	50.1	73.0	**51.2**	68.6	58.6	62.0	53.3	67.9	(137 x T) + 329 K	82.6 M
C12345 + FC	50.2	72.4	**51.2**	**69.3**	58.6	62.1	53.3	67.9	(467 x T) K	82.6 M

5.6.3 Parameterized Networks

In this section we present and analyse results regarding variations to the parameterized network modification. First, Rebuffi *et al.* (2018) proposes two versions of the network adapters: series adapters and parallel adapters as shown in Figure 5.3. Our results match those from Rebuffi *et al.* (2018), the parallel adapter outperform the series adapters by 2% on the average accuracy as shown in Table 5.5.

Table 5.5 Performance when using different adapter types

Adapter Type	**Market1501**		**DukeMTMCReID**		**CUHK03**		**Average**	
	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Series	49.7	73.2	49.8	68.2	57.4	61.2	52.3	67.5
Parallel	**52.1**	**75.0**	**51.6**	**69.8**	**61.1**	**63.6**	**54.9**	**69.5**

We then explore the optimal position to include these adapters in the OSNet architecture. Table 5.6 shows the result for various positioning shown in Figure 5.4. We notice that performance drops significantly as the adapter includes larger parts of the model. This could be explained by the fact that the adapter is not complex enough to improve the complex OSNet residual block as a whole. Instead, it can easily improve the simple convolutional layers found in each branch.

Table 5.6 Performance when adapters are placed at different points in the model

Adapter Position	Market1501		DukeMTMCReID		CUHK03		Average	
	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Conv	**52.1**	**75.0**	**51.6**	**69.8**	**61.1**	**63.6**	**54.9**	**69.5**
Gate	48.4	71.9	48.5	68.5	55.6	59.1	50.8	66.5
Block	44.0	68.0	44.0	64.6	51.0	54.9	46.3	62.5

Rebuffi *et al.* (2018) proposes to freeze the whole model except the adapters and find tune their parameters separately from the common parameters. As shown in Table 5.7, in our case jointly training the domain-specific adapters with the common parameters yields higher accuracy.

Table 5.7 Performance when adapters are trained jointly with the common model or separately while the model is frozen

Training	Market1501		DukeMTMCReID		CUHK03		Average	
	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Jointly	**52.1**	**75.0**	**51.6**	**69.8**	**61.1**	**63.6**	**54.9**	**69.5**
Freeze common model	50.4	73.8	50.3	68.8	58.4	62.3	53.0	68.3

To summarize, parameterized networks offer an interesting alternative to multi-branch networks. We have successfully adapted this technique to the OSNet_x0_25 architecture by choosing the appropriate adapters and modifying the training protocol. This technique offers an accuracy close to the highest multi-branch configuration accuracy while having a much lower memory footprint.

5.6.4 Distillation Adapters

In this section, we present our experiments with Distillation Adapters in an attempt to determine why it yields such a poor performance. We first try different adapters and distillation positioning. We try placing the adapter before the decoder heads and distilling using the features to follow closely the implementation by Li *et al.* (2022). We compare it with placing adapters after the decoder head and distilling using the similarity matrix distillation in Table 5.8. We note that the accuracy of the model is significantly higher when the distillation adapter is placed after the decoder head. This result is expected as we previously determined that similarity matrix distillation is more effective than feature distillation for person ReID. We note also that placing the distillation adapters at the very end of the model means there are no domain-specific layers kept after training. Indeed, keeping a domain-specific head as proposed by Li *et al.* (2022) is very costly memory-wise. The addition of distillation adapters however yields worse

results than having no adapters at all. We first theorized that the adapters learn valuable information to perform ReID and that discarding these layers at test time impacts the results negatively.

Table 5.8 Performance when distillation adapters are placed before and after the decoder

Adapter Position	Market1501		DukeMTMCReID		CUHK03		Average	
	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Position A	18.9	39.5	16.0	28.5	21.0	23.5	18.6	30.5
Position B	42.3	65.5	39.7	59.4	47.5	50.8	43.2	58.6
No Distillation Adapters	**51.7**	**75.1**	**51.4**	**70.0**	**59.3**	**62.6**	**54.1**	**69.2**

We see in Table 5.9 that keeping the adapters at test time does improve accuracy marginally indicating that some useful information is lost when discarding the adapters, The accuracy is still much lower than when not using any adapters.

Table 5.9 Performance when distillation adapters are used at test time or discarded after training

Test Configuration	Market1501		DukeMTMCReID		CUHK03		Average	
	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Without Adapters	42.3	65.5	39.7	59.4	47.5	50.8	43.2	58.6
With Adapters	42.3	65.6	42.1	62.0	49.6	54.3	44.7	60.6
No Distillation Adapters	**51.7**	**75.1**	**51.4**	**70.0**	**59.3**	**62.6**	**54.1**	**69.2**

In conclusion, this technique does not seem to be easily adaptable to the MTDA scenario. We theorize that the domain gap between datasets is smaller than between tasks and therefore the distillation does not benefit from the adapters. The adapters are meant to ease a very difficult distillation. In our case, the task-level information is similar, and therefore having common layers at the head of the model might even be beneficial.

5.7 Discussion

This chapter focuses on MTDA models to improve the trade-off between accuracy and complexity. We present results using the OSNet_x0_25 architecture, a very compact model that is well-suited for real-world applications. We then propose multiple model modifications which increase accuracy at the cost of increased complexity.

The simple DSBN modification yields the best gain in accuracy for the cost. While this modification is very efficient, we push further to maximize accuracy. We find that using a parameterized network in

addition to DSBN gives a higher accuracy for a reasonable cost. While this gain is not as efficient as the gain due to DSBN, it is a very interesting solution in applications that value accuracy more.

Multi-head and multi-tail models are then studied, where the accuracy is very good but the cost in complexity is high. These models are outclassed by the parameterized networks, which match the accuracy for a significantly lower complexity increase. These experiments allow us to understand the multi-target person ReID problem better. The importance of various stages of the model gives information about what causes domain shifts between target datasets, namely low-level variations.

Finally, our experiments with distillation adapters show that the multi-task problem is not completely analogous to the multi-target problem. While multi-task deals with completely different tasks, multi-target deals with datasets that represent the same high-level problem but with domain shifts. This would explain why model modifications which aim to attenuate low-level feature differences are more effective, eg, target-specific BN layers or even specializing early convolutional blocks. Distillation adapters are placed a the end of the pipeline, where high-level features are processed.

In this chapter, we have studied multiple techniques which ultimately attempt to make features target-invariant. Our experiments show that techniques that affect earlier features in the model are more effective than techniques that affect the later features. Early features are often thought of as more local low-level features. In relation to MTDA for ReID, we can say that variations between datasets are more low-level (changes in illumination and viewpoint), while at a higher level, they are very similar (images of pedestrians). We determine through our experiments that using specialized sets of parameters to reduce domain variations in low-level features is more important than attempting to reduce the domain variations at a high level.

CONCLUSION AND RECOMMENDATIONS

This thesis provides a comprehensive study of the MTDA problem for person ReID. We tackle the problem with a focus on real-world applications. We do so by optimizing accuracy on multiple targets simultaneously while minimizing the computational resource requirements of the solution.

We first provide a short presentation of key machine learning concepts relevant to this work. We then present an overview of the SOTA deep learning techniques used for person ReID with an emphasis on STDA and MTDA techniques.

For our first contribution, we propose a new MTDA technique adapted for person ReID. We produce a compact model capable of performing ReID on multiple datasets simultaneously with high accuracy. The technique uses KD to combine knowledge from multiple large specialized teacher models into a single small standard student model. We provide a comprehensive study of the technique's performance in multiple problem settings. We show our technique outperforms relevant SOTA techniques from the literature both in terms of accuracy and model complexity. Additionally, we demonstrate the flexibility of this technique by extending two SOTA STDA techniques to the MTDA scenario successfully. The technique even allows the use of models trained with different techniques for individual target domains, making it extremely easy to deploy in real-world applications.

For our second contribution, we study model modifications to maximize accuracy at the cost of higher complexity. The main goal here is to maximize the ratio of accuracy to the complexity of our solution. Our experiments allow us to find the best model configuration for a real-world application and allow us to draw interesting conclusions about the MTDA problem for person ReID. Indeed, these experiments allow us to observe the importance of low-level specific parameters, while later-stage parameters benefit from being common to all target domains. Intuitively we note that the variations between re-id datasets are mostly of a low-level nature (illumination, viewpoint, and resolution), while the high-level task remains the same (identifying images of pedestrians).

Looking forward, the following ideas need to be further explored:

- **Improve the domain-shift measurement techniques and involve it more in the pipeline:** Some datasets are much further from each other than others in a domain-shift sense. We have observed that the order in which the model sees these datasets can influence the performance of the model. A better understanding of how the different datasets relate could improve the training procedure.
- **Study how the technique scales with a large number of targets:** The experiments presented in this work are limited to four-person ReID datasets due to the limited availability of high-quality datasets. It would be interesting to identify the limits of KD-ReID when applied to a situation with a high number of targets or with more challenging datasets.
- **Adapt KD-ReID as an STDA where each camera within a single dataset is a target:** Cameras can be thought of as domains which their specific viewpoint, illumination, and image quality. KD-ReID could be used to reduce the domain gap between each camera. The nature of the person ReID task makes it so that a model could not specialize for a single camera. In fact, the images of pedestrians that are being compared are always from different cameras; therefore, the difficulty of person ReID. One way to adapt KD-ReID would be to have teachers specialized for each camera pair and distill all teachers into a strong student model. The potential advantage of this approach is that each model learns very precisely the relation between images coming from a specific pair of images. This would improve the final model's ability to deal with domain shifts due to different camera captures. A strong example of this type of work can be found in Mekhazni, Dufau, Desrosiers, Pedersoli & Granger (2023) where a domain discriminator is used with cameras as domains. This method yields good improvements showing the potential of treating different camera views as different domains.

BIBLIOGRAPHY

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Aich, A., Zheng, M., Karanam, S., Chen, T., Roy-Chowdhury, A. K. & Wu, Z. (2021). Spatio-temporal representation factorization for video-based person re-identification. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 152–162.

Alemi, A. A., Fischer, I., Dillon, J. V. & Murphy, K. (2016). Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

Bak, S., Carr, P. & Lalonde, J.-F. (2018). Domain adaptation through synthesis for unsupervised person re-identification. *Proceedings of the European conference on computer vision (ECCV)*, pp. 189–205.

Bazzani, L., Cristani, M., Perina, A., Farenzena, M. & Murino, V. (2010). Multiple-shot person re-identification by hpe signature. *2010 20th international conference on pattern recognition*, pp. 1413–1416.

Benenson, R., Omran, M., Hosang, J. & Schiele, B. (2014). Ten years of pedestrian detection, what have we learned? *European Conference on Computer Vision*, pp. 613–627.

Bertocco, G. C., Andaló, F. & Rocha, A. (2021). Unsupervised and self-adaptative techniques for cross-domain person re-identification. *IEEE Transactions on Information Forensics and Security*, 16, 4419–4434.

Bhuiyan, A., Liu, Y., Siva, P., Javan, M., Ayed, I. B. & Granger, E. (2020). Pose guided gated fusion for person re-identification. *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2675–2684.

Brunetti, A., Buongiorno, D., Trotta, G. F. & Bevilacqua, V. (2018). Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, 300, 17–33.

Chang, W.-G., You, T., Seo, S., Kwak, S. & Han, B. (2019). Domain-specific batch normalization for unsupervised domain adaptation. *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 7354–7362.

Chang, X., Hospedales, T. M. & Xiang, T. (2018a). Multi-level factorisation net for person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2109–2118.

Chang, X., Hospedales, T. M. & Xiang, T. (2018b). Multi-level factorisation net for person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2109–2118.

Chen, D., Mei, J.-P., Zhang, Y., Wang, C., Wang, Z., Feng, Y. & Chen, C. (2021). Cross-layer distillation with semantic calibration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8), 7028–7036.

Chen, G., Choi, W., Yu, X., Han, T. & Chandraker, M. (2017). Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30.

Chen, X., Su, J. & Zhang, J. (2019a). A two-teacher framework for knowledge distillation. *Advances in Neural Networks–ISNN 2019: 16th International Symposium on Neural Networks, ISNN 2019, Moscow, Russia, July 10–12, 2019, Proceedings, Part I 16*, pp. 58–66.

Chen, Y., Zhu, X. & Gong, S. (2019b). Instance-guided context rendering for cross-domain person re-identification. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 232–242.

Chen, Z., Zhuang, J., Liang, X. & Lin, L. (2019c). Blending-target domain adaptation by adversarial meta-adaptation networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2248–2257.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273–297.

Crammer, K., Kearns, M. & Wortman, J. (2008). Learning from Multiple Sources. *Journal of Machine Learning Research*, 9(8).

Das, A., Chakraborty, A. & Roy-Chowdhury, A. K. (2014). Consistent Re-identification in a Camera Network. *European Conference on Computer Vision*, 8690(Lecture Notes in Computer Science), 330–345.

Delorme, G., Xu, Y., Lathuilière, S., Horaud, R. & Alameda-Pineda, X. (2021). CANU-ReID: a conditional adversarial network for unsupervised person re-identification. *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4428–4435.

Deng, W., Zheng, L., Ye, Q., Kang, G., Yang, Y. & Jiao, J. (2018). Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 994–1003.

Ester, M., Kriegel, H., Sander, J., Xu, X. et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *kdd*, 96(34).

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. & Ramanan, D. (2010). Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 1627-1645. doi: 10.1109/TPAMI.2009.167.

Fu, X. & Lai, X. (2021). Unsupervised person re-identification via multi-order cross-view graph adversarial network. *IEEE Access*, 9, 22264–22273.

Gao, C., Wang, J., Liu, L., Yu, J.-G. & Sang, N. (2016). Temporally aligned pooling representation for video-based person re-identification. *2016 IEEE international conference on image processing (ICIP)*, pp. 4284–4288.

Ge, Y., Zhu, F., Chen, D., Zhao, R. & Li, H. (2020). Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. *arXiv preprint arXiv:2006.02713*.

Gheissari, N., Sebastian, T. B. & Hartley, R. (2006). Person reidentification using spatiotemporal appearance. *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, 2, 1528–1535.

Gholami, B., Sahu, P., Rudovic, O., Bousmalis, K. & Pavlovic, V. (2020). Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing*, 29, 3993–4002.

Gong, S., Cristani, M., Loy, C. C. & Hospedales, T. M. (2014). The re-identification challenge. In *Person re-identification* (pp. 1–20). Springer.

Gong, Y. (2021). A general multi-modal data learning method for person re-identification. *arXiv preprint arXiv:2101.08533*.

Gray, D., Brennan, S. & Tao, H. (2007). Evaluating appearance models for recognition, reacquisition, and tracking. *Proc. IEEE international workshop on performance evaluation for tracking and surveillance (PETS)*, 3(5), 1–7.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. & Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1), 723–773.

Hamdoun, O., Moutarde, F., Stanciulescu, B. & Steux, B. (2008). Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. *2008 Second ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 1-6. doi: 10.1109/ICDSC.2008.4635689.

Hasan, I., Liao, S., Li, J., Akram, S. U. & Shao, L. (2021). Generalizable pedestrian detection: The elephant in the room. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11328–11337.

He, K., Zhang, X., Ren, S. & Sun, J. (2015). Deep Residual Learning for Image Recognition.

Heo, B., Kim, J., Yun, S., Park, H., Kwak, N. & Choi, J. Y. (2019). A comprehensive overhaul of feature distillation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1921–1930.

Hermans, A., Beyer, L. & Leibe, B. (2017). In Defense of the Triplet Loss for Person Re-Identification.

Hinton, G., Vinyals, O., Dean, J. et al. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Hirzer, M., Beleznai, C., Roth, P. M. & Bischof, H. (2011). Person Re-Identification by Descriptive and Discriminative Classification. *Proc. Scandinavian Conference on Image Analysis (SCIA)*.

Hornik, K., Stinchcombe, M. & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5), 359–366.

Hou, R., Ma, B., Chang, H., Gu, X., Shan, S. & Chen, X. (2019a). Vrstc: Occlusion-free video person re-identification. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7183–7192.

Hou, R., Ma, B., Chang, H., Gu, X., Shan, S. & Chen, X. (2019b). Vrstc: Occlusion-free video person re-identification. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7183–7192.

Huang, Y., Wu, Q., Xu, J. & Zhong, Y. (2019a). SBSGAN: Suppression of inter-domain background shift for person re-identification. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9527–9536.

Huang, Y., Zha, Z.-J., Fu, X. & Zhang, W. (2019b). Illumination-invariant person re-identification. *Proceedings of the 27th ACM international conference on multimedia*, pp. 365–373.

Ioffe, S. & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, pp. 448–456.

Isobe, T., Jia, X., Chen, S., He, J., Shi, Y., Liu, J., Lu, H. & Wang, S. (2021). Multi-target domain adaptation with collaborative consistency learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8187–8196.

Karaman, S. & Bagdanov, A. D. (2012). Identity inference: generalizing person re-identification scenarios. *Computer Vision–ECCV 2012. Workshops and Demonstrations: Florence, Italy, October 7-13, 2012, Proceedings, Part I 12*, pp. 443–452.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A. et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13), 3521–3526.

Kokkinos, I. (2017). Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6129–6138.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Li, M., Zhu, X. & Gong, S. (2019). Unsupervised tracklet person re-identification. *IEEE transactions on pattern analysis and machine intelligence*, 42(7), 1770–1782.

Li, W. & Wang, X. (2013). Locally aligned feature transforms across views. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3594–3601.

Li, W., Zhao, R. & Wang, X. (2012). Human reidentification with transferred metric learning. *Asian conference on computer vision*, pp. 31–44.

Li, W., Zhao, R., Xiao, T. & Wang, X. (2014). Deepreid: Deep filter pairing neural network for person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 152–159.

Li, W., Zhu, X. & Gong, S. (2018). Harmonious attention network for person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2285–2294.

Li, W.-H., Liu, X. & Bilen, H. (2022). Universal Representations: A Unified Look at Multiple Task and Domain Learning. *arXiv preprint arXiv:2204.02744*.

Liang, W., Wang, G., Lai, J. & Zhu, J. (2018). M2m-gan: Many-to-many generative adversarial transfer learning for person re-identification. *arXiv preprint arXiv:1811.03768*.

Liao, S. & Li, S. Z. (2015). Efficient PSD Constrained Asymmetric Metric Learning for Person Re-Identification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3685-3693. doi: 10.1109/ICCV.2015.420.

Lin, J., Ren, L., Lu, J., Feng, J. & Zhou, J. (2017). Consistent-aware deep learning for person re-identification in a camera network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5771–5780.

Lin, Y., Dong, X., Zheng, L., Yan, Y. & Yang, Y. (2019a). A bottom-up clustering approach to unsupervised person re-identification. *Proceedings of the AAAI conference on artificial intelligence*, 33(01), 8738–8745.

Lin, Y., Zheng, L., Zheng, Z., Wu, Y., Hu, Z., Yan, C. & Yang, Y. (2019b). Improving person re-identification by attribute and identity learning. *Pattern recognition*, 95, 151–161.

Liu, F. & Zhang, L. (2019). View confusion feature learning for person re-identification. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6639–6648.

Liu, J., Ni, B., Yan, Y., Zhou, P., Cheng, S. & Hu, J. (2018a). Pose transferrable person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4099–4108.

Liu, J., Ni, B., Yan, Y., Zhou, P., Cheng, S. & Hu, J. (2018b). Pose transferrable person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4099–4108.

Long, M., Cao, Y., Wang, J. & Jordan, M. (2015). Learning transferable features with deep adaptation networks. *International conference on machine learning*, pp. 97–105.

Luo, S., Pan, W., Wang, X., Wang, D., Tang, H. & Song, M. (2020). Collaboration by competition: Self-coordinated knowledge amalgamation for multi-talent student learning. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*,

pp. 631–646.

Matsukawa, T., Okabe, T., Suzuki, E. & Sato, Y. (2016). Hierarchical gaussian descriptor for person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1363–1372.

McLaughlin, N., Del Rincon, J. M. & Miller, P. (2016). Recurrent convolutional network for video-based person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1325–1334.

Mekhazni, D., Bhuiyan, A., Ekladious, G. & Granger, É. (2020). Unsupervised Domain Adaptation in the Dissimilarity Space for Person Re-identification. *ECCV*.

Mekhazni, D., Dufau, M., Desrosiers, C., Pedersoli, M. & Granger, E. (2023). Camera alignment and weighted contrastive learning for domain adaptation in video person ReID. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1624–1633.

Meng, Z., Li, J., Zhao, Y. & Gong, Y. (2019). Conditional teacher-student learning. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6445–6449.

Mohanty, A., Banerjee, B. & Velmurugan, R. (2022). SSMTReID-Net: Multi-Target Unsupervised Domain Adaptation for Person Re-Identification. *Pattern Recognition Letters*, 163, 40–46.

Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M. & Van Gool, L. (2017). Fast scene understanding for autonomous driving. *arXiv preprint arXiv:1708.02550*.

Nguyen-Meidine, L. T., Bela, A., Kiran, M., Dolz, J., Blais-Morin, L.-A. & Granger, E. (2020). Unsupervised Multi-Target Domain Adaptation Through Knowledge Distillation.

Nikhal, K. & Riggan, B. S. (2021). Unsupervised attention based instance discriminative learning for person re-identification. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2422–2431.

Passban, P., Wu, Y., Rezagholizadeh, M. & Liu, Q. (2021). Alp-kd: Attention-based layer projection for knowledge distillation. *Proceedings of the AAAI Conference on artificial intelligence*, 35(15), 13657–13665.

PENG, L., KUANG, P., LI, F. & GU, X. (2019). An Efficient Person Reid Methed Based on Knowledge Distillation. *2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing*, pp. 13–16.

Porikli, F. (2003). Inter-camera color calibration by correlation model function. *Proceedings 2003 international conference on image processing (cat. No. 03CH37429)*, 2, II–133.

Qian, X., Fu, Y., Xiang, T., Wang, W., Qiu, J., Wu, Y., Jiang, Y.-G. & Xue, X. (2018). Pose-normalized image generation for person re-identification. *Proceedings of the European conference on computer*

vision (ECCV), pp. 650–667.

Rebuffi, S.-A., Bilen, H. & Vedaldi, A. (2018). Efficient parametrization of multi-domain deep neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8119–8127.

Remigereau, F., Mekhazni, D., Abdoli, S., Nguyen-Meidine, L. T., Cruz, R. M. O. & Granger, E. (2022). Knowledge Distillation for Multi-Target Domain Adaptation in Real-Time Person Re-Identification. *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 3853-3557. doi: 10.1109/ICIP46576.2022.9897730.

Ren, P. & Li, J. (2018). Factorized distillation: Training holistic person re-identification model by distilling an ensemble of partial ReID models. *arXiv preprint arXiv:1811.08073*.

Ren, S., He, K., Girshick, R. & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 91–99.

Ristani, E., Solera, F., Zou, R., Cucchiara, R. & Tomasi, C. (2016). Performance measures and a data set for multi-target, multi-camera tracking. *European conference on computer vision*, pp. 17–35.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C. & Bengio, Y. (2014). Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533–536.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520.

Sarfraz, M. S., Schumann, A., Eberle, A. & Stiefelhagen, R. (2018a). A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 420–429.

Sarfraz, M. S., Schumann, A., Eberle, A. & Stiefelhagen, R. (2018b). A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 420–429.

Shen, C., Xue, M., Wang, X., Song, J., Sun, L. & Song, M. (2019). Customizing student networks from heterogeneous teachers via adaptive knowledge amalgamation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3504–3513.

Sheng, K., Li, K., Zheng, X., Liang, J., Dong, W., Huang, F., Ji, R. & Sun, X. (2021). On evolving attention towards domain adaptation. *arXiv preprint arXiv:2103.13561*.

Si, J., Zhang, H., Li, C.-G., Kuen, J., Kong, X., Kot, A. C. & Wang, G. (2018). Dual attention matching network for context-aware feature sequence based person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5363–5372.

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Song, C., Huang, Y., Ouyang, W. & Wang, L. (2018). Mask-guided contrastive attention model for person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1179–1188.

Su, C., Zhang, S., Xing, J., Gao, W. & Tian, Q. (2016). Deep attributes driven multi-camera person re-identification. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pp. 475–491.

Sun, Y., Zheng, L., Yang, Y., Tian, Q. & Wang, S. (2018). Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). *Proceedings of the European conference on computer vision (ECCV)*, pp. 480–496.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016, June). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tay, C.-P., Roy, S. & Yap, K.-H. (2019). Aanet: Attribute attention network for person re-identifications. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7134–7143.

Tian, J., Tang, Q., Li, R., Teng, Z., Zhang, B. & Fan, J. (2021). A Camera Identity-guided Distribution Consistency Method for Unsupervised Multi-target Domain Person reID. *ACM Trans. Intelligent Systems and Technology*, 12(4), 1–18.

Vandenhende, S., Georgoulis, S., De Brabandere, B. & Van Gool, L. (2019). Branched Multi-Task Networks: Deciding What Layers To Share. arXiv. doi: 10.48550/ARXIV.1904.02920.

Walawalkar, D., Shen, Z. & Savvides, M. (2020). Online ensemble model compression using knowledge distillation. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, pp. 18–35.

Wang, D. & Zhang, S. (2020). Unsupervised person re-identification via multi-label classification. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10981–10990.

Wang, G., Lai, J., Huang, P. & Xie, X. (2019a). Spatial-temporal person re-identification. *Proceedings of the AAAI conference on artificial intelligence*, 33(01), 8933–8940.

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B. & Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. *Proceedings of the IEEE conference on computer*

vision and pattern recognition, pp. 1386–1393.

Wang, W. (2020). Adapted Center and Scale Prediction: More Stable and More Accurate. *arXiv preprint arXiv:2002.09053*.

Wang, W., Zhao, F., Liao, S. & Shao, L. (2022). Attentive waveblock: complementarity-enhanced mutual networks for unsupervised domain adaptation in person re-identification and beyond. *IEEE Transactions on Image Processing*, 31, 1532–1544.

Wang, X., Hu, J.-F., Lai, J.-H., Zhang, J. & Zheng, W.-S. (2019b). Progressive teacher-student learning for early action prediction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3556–3565.

Wang, X., Liu, M., Raychaudhuri, D. S., Paul, S., Wang, Y. & Roy-Chowdhury, A. K. (2021a). Learning person re-identification models from videos with weak supervision. *IEEE Transactions on Image Processing*, 30, 3017–3028.

Wang, Y., Zhang, P., Gao, S., Geng, X., Lu, H. & Wang, D. (2021b). Pyramid spatial-temporal aggregation for video-based person re-identification. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12026–12035.

Wei, L., Zhang, S., Gao, W. & Tian, Q. (2018). Person transfer gan to bridge domain gap for person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 79–88.

Wieczorek, M., Rychalska, B. & Dąbrowski, J. (2021). On the unreasonable effectiveness of centroids in image retrieval. *International Conference on Neural Information Processing*, pp. 212–223.

Wu, A., Zheng, W.-S., Guo, X. & Lai, J.-H. (2019a). Distilled person re-identification: Towards a more scalable system. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1187–1196.

Wu, A., Zheng, W.-S., Guo, X. & Lai, J.-H. (2019b). Distilled person re-identification: Towards a more scalable system. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1187–1196.

Wu, G. & Gong, S. (2021). Peer collaborative learning for online knowledge distillation. *Proceedings of the AAAI Conference on artificial intelligence*, 35(12), 10302–10310.

Wu, S., Chen, Y.-C., Li, X., Wu, A.-C., You, J.-J. & Zheng, W.-S. (2016). An enhanced deep feature representation for person re-identification. *2016 IEEE winter conference on applications of computer vision (WACV)*, pp. 1–8.

Wu, Y., Yang, W. & Wang, M. (2022). Unsupervised Person Re-Identification with Attention-Guided Fine-Grained Features and Symmetric Contrast Learning. *Sensors*, 22(18), 6978.

Xiong, F., Gou, M., Camps, O. & Sznaier, M. (2014). Person re-identification using kernel-based metric learning methods. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, pp. 1–16.

Xu, S., Cheng, Y., Gu, K., Yang, Y., Chang, S. & Zhou, P. (2017). Jointly attentive spatial-temporal pooling networks for video-based person re-identification. *Proceedings of the IEEE international conference on computer vision*, pp. 4733–4742.

Xu, Y., Ma, B., Huang, R. & Lin, L. (2014). Person search in a scene by jointly modeling people commonness and person uniqueness. *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 937–940.

Xu, Z., Li, B., Yuan, Y. & Dang, A. (2020). Beta r-cnn: Looking into pedestrian detection from another perspective. *Advances in Neural Information Processing Systems*.

Xuan, S. & Zhang, S. (2021). Intra-Inter Camera Similarity for Unsupervised Person Re-Identification. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11926–11935.

Yang, C., Xie, L., Su, C. & Yuille, A. L. (2019a). Snapshot distillation: Teacher-student optimization in one generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2859–2868.

Yang, F., Yan, K., Lu, S., Jia, H., Xie, X. & Gao, W. (2019b). Attention driven person re-identification. *Pattern Recognition*, 86, 143–155.

Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L. & Hoi, S. C. (2021). Deep learning for person re-identification: A survey and outlook. *IEEE transactions on pattern analysis and machine intelligence*, 44(6), 2872–2893.

Yi, D., Lei, Z., Liao, S. & Li, S. Z. (2014). Deep metric learning for person re-identification. *2014 22nd international conference on pattern recognition*, pp. 34–39.

You, J., Wu, A., Li, X. & Zheng, W.-S. (2016). Top-push video-based person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1345–1353.

Yu, H., Liu, B., Lu, Y., Chu, Q. & Yu, N. (2022). Multi-view Geometry Distillation for Cloth-Changing Person ReID. *Pattern Recognition and Computer Vision: 5th Chinese Conference, PRCV 2022, Shenzhen, China, November 4–7, 2022, Proceedings, Part I*, pp. 29–41.

Yu, H.-X., Wu, A. & Zheng, W.-S. (2020). Unsupervised Person Re-Identification by Deep Asymmetric Metric Embedding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4), 956-973. doi: 10.1109/TPAMI.2018.2886878.

Yu, H., Hu, M. & Chen, S. (2018). Multi-target unsupervised domain adaptation without exactly shared categories. *arXiv preprint arXiv:1809.00852*.

Yuan, L., Tay, F. E., Li, G., Wang, T. & Feng, J. (2020). Revisiting knowledge distillation via label smoothing regularization. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3903–3911.

Zajdel, W., Zivkovic, Z. & Krose, B. J. (2005). Keeping track of humans: Have I seen this person before? *Proceedings of the 2005 IEEE international conference on robotics and automation*, pp. 2081–2086.

Zang, X., Li, G. & Gao, W. (2022). Multi-direction and Multi-scale Pyramid in Transformer for Video-based Pedestrian Retrieval. *IEEE Transactions on Industrial Informatics*.

Zeng, M., Li, S., Li, R., Lu, J., Xu, K., Gu, J. & Chen, Y. (2022). A Multi-target Domain Adaptive Method for Intelligent Transfer Fault Diagnosis. *Measurement*, 112352.

Zhang, F., Zhu, X. & Ye, M. (2019a). Fast human pose estimation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3517–3526.

Zhang, H., Hu, Z., Qin, W., Xu, M. & Wang, M. (2021). Adversarial co-distillation learning for image recognition. *Pattern Recognition*, 111, 107659.

Zhang, L., Song, J., Gao, A., Chen, J., Bao, C. & Ma, K. (2019b). Be your own teacher: Improve the performance of convolutional neural networks via self distillation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3713–3722.

Zhao, Y., Shen, X., Jin, Z., Lu, H. & Hua, X.-s. (2019). Attribute-driven feature disentangling and temporal aggregation for video person re-identification. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4913–4922.

Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J. & Tian, Q. (2015). Scalable person re-identification: A benchmark. *Proceedings of the IEEE international conference on computer vision*, pp. 1116–1124.

Zheng, Z., Yang, X., Yu, Z., Zheng, L., Yang, Y. & Kautz, J. (2019). Joint discriminative and generative learning for person re-identification. *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2138–2147.

Zhong, Z., Zheng, L., Zheng, Z., Li, S. & Yang, Y. (2018a). Camstyle: A novel data augmentation method for person re-identification. *IEEE Transactions on Image Processing*, 28(3), 1176–1190.

Zhong, Z., Zheng, L., Zheng, Z., Li, S. & Yang, Y. (2018b). Camera style adaptation for person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5157–5166.

Zhou, K. & Xiang, T. (2019). Torchreid: A Library for Deep Learning Person Re-Identification in Pytorch. *arXiv preprint arXiv:1910.10093*.

Zhou, K., Yang, Y., Cavallaro, A. & Xiang, T. (2019). Omni-scale feature learning for person re-identification. *Proceedings of the IEEE/CVF International Conference on Computer Vision*,

pp. 3702–3712.

Zhu, X., Jing, X.-Y., You, X., Zhang, X. & Zhang, T. (2018). Video-based person re-identification by simultaneously learning intra-video and inter-video distance metrics. *IEEE Transactions on Image Processing*, 27(11), 5683–5695.