University of Brasilia

Institute of Exact Sciences
Department of Computer Science

Unsupervised Domain Adaptation for Real World Person Re-Identification

Tiago de C. G. Pereira

Dissertation presented for conclusion of the Master Degree in Computer Science

Supervisor
Prof. Dr. Teófilo E. de Campos

Brasilia
2022

University of Brasilia

Institute of Exact Sciences
Department of Computer Science

Unsupervised Domain Adaptation for Real World Person Re-Identification

Tiago de C. G. Pereira

Dissertation presented for conclusion of the Master Degree in Computer Science

Prof. Dr. Teófilo E. de Campos (Supervisor)
CIC/UnB

Prof. Dr. Krystian Mikolajczyk
Imperial College London

Prof. Dr. Bruno Macchiavello
CIC/UnB

Dr. Ricardo Pezzuol Jacobi
Computer Science Graduate Program Coordinator

Brasilia, March 11, 2022

Acknowledgments

I’d like to thank my supervisor Teófilo de Campos for always being available to guide me, push me, refine my ideas and encourage me through the course of this masters.

I’d also like to thank my parents Daniel Gallo and Daniela Moraes, my girlfriend Yolanda Ricarte, my siblings Camila Pereira and Pedro Pereira and all my friends for being at my side at all moments and keep my head up towards the final goal.

I’m very appreciated to all researchers that peer-reviewed my paper submissions and to my examiners Dr. Krystian Mikolajczyk and Dr. Bruno Macchiavello for valuable feedbacks that for sure enhanced the quality of this dissertation.

We acknowledge the support of FAPDF (fap.df.gov.br) and CNPq grant PQ 314154/2018-3 (cnpq.br).

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Abstract

In the world where big data reigns and there is plenty of hardware prepared to gather a huge amount of non structured data, data acquisition is no longer a problem. Surveillance cameras are ubiquitous and they capture huge numbers of people walking across different scenes. However, extracting value from this data is challenging, specially for tasks that involve human images, such as face recognition and person re-identification. Annotation of this kind of data is a challenging and expensive task. In this work we propose Unsupervised Domain Adaptation (UDA) methods for person Re-Identification (Re-ID) that rely on target domain samples to model the marginal distribution of the data. To deal with the lack of target domain labels, UDA methods leverage information from labelled source samples and unlabelled target samples.

Firstly, we propose a baseline method that may use Resnet-50 or AlignedReID++ as backbone, trained using a Triplet loss with batch hard. The domain adaptation is done in two phases: **1)** using a GAN generated intermediate dataset that leverages from the source domain labels and approximate the source samples appearance to be similar to the target domain samples, and **2)** using pseudo-labels generated with an unsupervised learning strategy.

Next, we realised that the quality of the clusters clearly plays a major role in the method's performance, however this point has been overlooked by the majority of methods, including our first approach. Therefore, we propose a multi-step pseudo-label refinement method to select the best possible clusters and keep improving them so that these clusters become closer to the class divisions without knowledge of the class labels. Our refinement method includes a cluster selection strategy and a camera-based normalisation method which reduces the within-domain variations caused by the use of multiple cameras in person Re-ID. This allows our method to reach state-of-the-art UDA results on DukeMTMC → Market1501 (source → target). We surpass state-of-the-art for UDA Re-ID by 1.6% on Market1501 → DukeMTMC datasets, which is a more challenging adaptation setup because the target domain (DukeMTMC) has eight distinct cameras. Furthermore, the camera-based normalisation method causes a significant reduction in the number of iterations required for training convergence.

Our results show that domain adaptation techniques really improve the model performance when applied in the target domain. Also, these techniques unlock the person Re-ID use for real world problems, once they may be automated to adapt a model for new unseen scenarios while maintaining its original performance.

Keywords: Computer Vision, Deep Learning, Person Re-Identification, Metric Learning, Domain Adaptation

Resumo

Os avanços da tecnologia e a globalização da industrialização democratizaram o acesso a equipamentos de alta qualidade. Câmeras de segurança seguem essa tendência e se um dia elas foram consideradas um equipamento de luxo utilizado apenas por grandes empreendimentos ou condomínios, hoje não é mais assim. Qualquer pequeno comércio ou residência já possuem um conjuto de câmeras para monitorar os seus arredores.

No entanto, as câmeras por si só não conseguem prover um monitoramento inteligente, elas apenas geram dados que podem ser analisados, em tempo real ou posteriormente. Uma vez que alocar pessoas para monitorar as câmeras em tempo real é custoso, algoritmos de visão computacional são a solução para extrair informações em tempo real dos dados coletados.

Métodos de visão computacional como re-identificação de pessoas, reconhecimento de ações suspeitas e reconhecimento facial são fundamentais para auxiliar nesse monitoramento inteligente de ambientes. Em específico, a re-identificação de pessoas é um método que visa indicar se duas imagens são da mesma pessoa ou não. Dessa forma, esse é um método extremamente valioso para grandes empreendimentos como shoppings ou aeroportos, pois ele permite manter um histórico da movimentação de cada pessoa dentro da área monitorada. Caso houvesse alguma ocorrência de segurança, o responsável pelo monitoramento do ambiente não precisaria rever os vídeos de todas as câmeras para entender o ocorrido, ele poderia apenas verificar a movimentação do infrator.

A grande maioria dos métodos propostos para esses algoritmos não visa a utilização desses em ambientes reais, mas sim em otimizar os resultados em bases de dados criadas para fazer *benchmarks*. Logo, quando esses algoritmos são utilizados em situações reais, eles apresentam performance muito inferiores às apresentadas nos testes. Há três caminhos possíveis para resolver essa diferença de performance: **a)** criar uma base de dados do ambiente real e especializar o algoritmo nessa base de dados, **b)** criar algortimos robustos a variações de ambiente ou **c)** criar métodos que adaptem esses algoritmos para novos ambientes de forma automatizada. Independente do caminho escolhido para solucionar esse problema, o insumo necessário para criar tal solução são imagens de pessoas passando em frente a câmeras de segurança.

Num mundo dominado pelo *Big Data* a aquisição de dados não é mais um problema, pois há inúmeros equipamentos preparados para captar uma grande quantidade de dados não estruturados. Câmeras de segurança são onipresentes e capturam várias imagens de pessoas andando pelos mais diversos cenários. No entanto, extrair valor de dados não estruturados é desafiador, especialmente para tarefas que envolvem imagens de pessoas. A anotação desses dados é um processo extremamente complexo e caro, portanto a criação de bases de dados específicas para cada ambiente não é vista com bons olhos.

A criação de algoritmos robustos a variações de ambiente seria a solução ideal, no entanto as pesquisas desse tema apontam que ainda estamos muito distantes de alcançar tal feito. Logo, técnicas de adaptação de domínio que permitam adaptar os algoritmos para novos cenários de forma automatizada têm sido muito estudadas tanto na academia quanto na indústria.

Nesse trabalho, propomos técnicas não supervisionadas de adaptação de domínio para a re-identificação de pessoas, visando reduzir a lacuna de performance entre a pesquisa de re-identificação de pessoas e as aplicações reais. Essas técnicas buscam modelar a distribuição dos dados do domínio alvo (ambiente de aplicação), utilizando apenas imagens provenientes desse novo cenário, sem ter acesso as anotações dessas imagens. Para lidar com essa falta de anotações no domínio alvo, os métodos de adaptação de domínio também utilizam imagens e anotações de um domínio fonte (base de dados anotada) para auxiliar no aprendizado dos algoritmos.

Os métodos de re-identificação de pessoas utilizados nesse trabalho usam redes neurais convolucionais para extrair *features* das imagens das pessoas. O treinamento dessas redes neurais é realizado de forma que as *features* extraídas das imagens pertençam a um espaço vetorial Euclidiano, onde *features* provenientes de imagens de uma mesma pessoa estão próximas e *features* provenientes de imagens de pessoas distintas estão distantes.

Ao treinar a rede neural em uma base de dados, ela aprende características específicas daquela base de dados para resolver o problema em questão, por isso ao aplicar essas redes em novas bases a performance decai. No caso específico da re-identificação de pessoas, uma das principais características que a rede neural precisa ter é a capacidade de diferenciar o que é o fundo da imagem do que é uma pessoa. Por exemplo, uma base de dados pode ter várias imagens que apresentam grama no fundo, logo a rede neural aprende a diferenciar grama de pessoas. Ao aplicar essa rede neural em um ambiente onde o fundo das imagens apresenta paredes, essa rede pode ter problemas de diferenciar o que é informação de parede do que é informação de pessoas. O reflexo disso na re-identificação de pessoas é que o espaço Euclidiano da saída da rede tenderá a agrupar *features* de imagens provenínete da mesma câmera, ao invés de *features* provenientes de imagens da mesma pessoa.

Em nossa primeira abordagem, propomos um método agnostico a arquitetura de redes neurais utilizada como base. Portanto, utilizamos a arquitetura clássica *Resnet-50* e a arquitetura *AlignedReID++* proposta por Luo et al. em nossos experimentos para analisar como diferentes arquiteturas se comportam frente ao nosso método. Em ambos os casos realizamos o treinamento utilizando a função de custo *Triplet* com a estratégia *batch hard* para gerarmos esse espaço vetorial Euclidiano com a *features* de saída das redes neurais. A adaptação de domínio proposta é feita em duas etapas:

- **1)** Uma GAN (rede neural especializada em gerar imagens) é utilizada para alterar a aparência das imagens do domínio fonte de forma que elas se aparentem com as imagens do domínio alvo. Desta forma criamos um domínio intermediário que contém as anotações do domínio fonte e imagens com aparências próximas as do domínio alvo;
- **2)** Métodos de *clusterização* não supervisionados são utilizados para gerar *pseudo* anotações (*clusters*) no domínio alvo. A partir dessas *pseudo* anotações somos capazes de retreinar a nossa rede neural nas imagens reais do domínio alvo.

Com essa primeira abordagem conseguimos melhorar a performance dos algoritmos ao aplicarmos em novos domínios. No entanto, não nos atentamos a qualidade das pseudo anotações (*clusters*) gerada. Portanto, não fomos capazes de extrair todo o potencial do método e atingirmos resultados que se aproximassem do estado da arte.

Ao percebermos que a qualidade dos *clusters* são cruciais para a performance do método, por mais que esse fator tenha sido subestimado pela maioria dos métodos existentes. Nós propomos um novo método para refinar as *pseudo* anotações utilizando múltiplas etapas, que consistem em selecionar os melhores *clusters* possíveis e continuar melhorando a qualidade deles para que eles se aproximem da real anotação dos dados. Nosso método de refinamento consiste em uma estratégia de seleção de *clusters* e em uma normalização guiada pelas câmeras que reduz a variância intra-domínio causada pelo uso de múltiplas câmeras na re-identificação de pessoas.

Esse novo método elevou nossos resultados a um novo patamar, com ele alcançamos o estado da arte da adaptação de domínio não supervisionada para re-identificação de pessoas nas bases de dados DukeMTMC → Market1501 (fonte → alvo). Para as bases de dados Market1501 → DukeMTMC nós ultrapassamos o estado da arte em 1.6%, essa combinação de bases de dados representa um desafio maior de adaptação, pois o domínio alvo (DukeMTMC) conta com oito câmeras distintas. Além do mais, nossa normalização guiada por câmeras gera uma redução significante na quantidade de iterações necessárias para atingir a convergência durante o treinamento.

Nossos resultados mostram que as técnicas de adaptação de domínio são capazes de melhorar significativamente a performance dos modelos quando aplicados no domínio alvo. Ademais, essas técnicas permitem que a re-identificação de pessoas possa ser usada em casos reais, pois elas automatizam o processo de adaptação do modelo para novos cenários enquanto mantém a performance muito próxima a do original do modelo.

Palavras-chave: Visão Computacional, Aprendizado Profundo, Re-Identificação de Pessoas, Aprendizado de Métricas, Adaptação de Domínio

Contents

List of Acronyms and Abbreviations

AI Artificial Intelligence

BN Batch Normalisation

CCTV Closed-Circuit-Television

CMC Cumulative Matching Characteristics

CNN Convolutional Neural Network

GAN Generative Adversarial Network

GDPR General Data Protection Regulation

LGPD *Lei Geral de Proteção de Dados*

mAP Mean Average Precision

ML Machine Learning

MLP Multilayer Perceptron

MSE Mean Squared Error

NLP Natural Language Processing

Re-ID Re-Identification

RPN Region Proposal Networks

UDA Unsupervised Domain Adaptation

Notation

γ	Batch Size
c	Image Channels
V	Set of Cameras
ν	Camera View
α_{ij}	A Clustering Solution
ζ	Completeness
$\mathcal{D}$	Domain
D	Euclidean Distance
$\mathbb{E}$	Expectation
$\mathcal{E}$	Euclidean Feature Space
Φ	GAN Discriminator
$\mathbf{f}$	Feature Vector
G	GAN Generator
g	Scale of Random Fluctuations
h	Image Height
ξ	Homogeneity
k	Number of Clusters in k-means
$\mathcal{L}$	Loss Function
ε	Learning Rate
m	Margin
ϵ	DBSCAN Maximum Distance
$\boldsymbol{\mu}$	Mean Vector
ω	DBSCAN Minimum Samples at Dense Region
$\mathcal{N}$	Training Set Size
n	Number of classes
ρ	Identities per Batch
ς	Examples per Person
P	Marginal Probability Distribution
$\mathbb{R}$	Set of Real Numbers

τ	CMC Rank
softmax	Softmax function
$\boldsymbol{\sigma}$	Standard Deviation Vector
$\mathcal{T}$	Task
Λ	V-Measure
w	Image Width
$\mathcal{X}$	Feature Space
$\mathcal{Y}$	Label Space

Bold lower case letters are used to represent vectors ($\mathbf{x}$), while bold upper case letters are employed to indicate matrices ($\mathbf{X}$) and italic lower case letters are used for scalars (x). Italic upper case letters are used to denote both sets and sequences (X).

Chapter 1

Introduction

1.1 Problem

Person re-identification (Re-ID) is an image retrieval task which aims at matching person images from different non-overlapping cameras views (Figure 1.1). This is an essential feature for diverse real word challenges, such as smart cities [86], intelligent video surveillance [76], suspicious action recognition [78] and pedestrian retrieval [72].

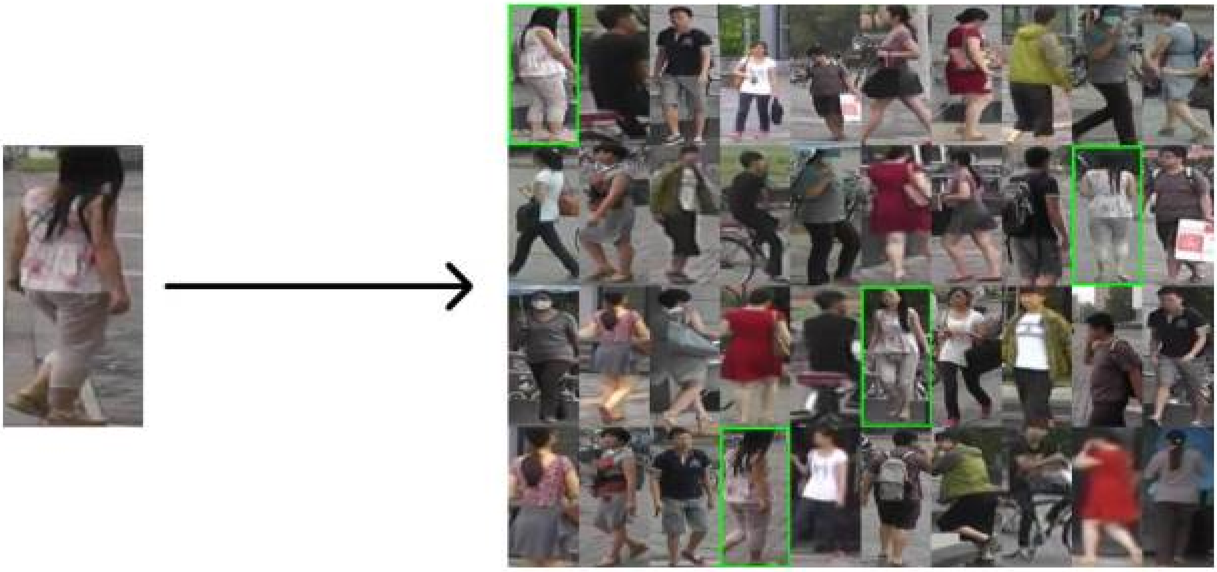


Figure 1.1: Person Re-Identification is an image retrieval task. Given a query image, Person Re-ID's objective is to find images from the same person in a gallery.

Although person Re-ID and face recognition are similar problems, they have a crucial difference. Face recognition requires images with high quality and from a frontal face view, while person Re-ID works with images from CCTV (closed-circuit-television) systems that have low resolution and varied viewpoints.

Furthermore, personal data protection and privacy are mainstream topics with regulations like GDPR (general data protection regulation) in Europe and LGPD (*lei geral de proteção de dados*) in Brazil. As person Re-ID systems do not necessarily recognise a person, only re-identifies a previously seen person (inside a restricted time difference), they have way less friction to be used in real-world systems.

With all these popular possible applications and advantages, there is a clear demand for robust Re-ID systems in the industry. Academic research groups have achieved remarkable in-domain results on popular person Re-ID datasets such as Market1501 [89] and DukeMTMC-reID [91]. Despite these advances, there is still a dichotomy between the success in academic results versus the industrial application. This is because the best academic results [51, 75, 97] are based on supervised methods that require a huge amount of annotated data for their training.

The use of pre-trained state-of-the-art Re-ID models in new scenarios usually leads to disappointing results because each group of cameras has distinct characteristics, such as illumination, resolution, noise level, orientation, pose, distance, focal length, amount of people's motion as well as factors that influence the appearance of people, such as ethnicity, type of location (e.g. leisure vs. work places) and weather conditions.

Therefore, we have a scenario where a lot of non-annotated data (from CCTV systems) is available and we have some pre-trained models that are specialised on specific domains. Our main research question is how to leverage from the pre-trained model knowledge to perform well in data from new scenes?

1.2 Objectives

Our main objective is to propose a person Re-ID framework that is capable to learn good representations from non-annotated data. Then, we set some auxiliary goals to help us achieve our main objective and answer the research question. Our auxiliary goals are:

- Implement a baseline domain adaptation method to have a baseline to start from;
- Identify the flaws in our baseline domain adaptation method and propose techniques to undermine them;
- Compare our proposed methods with the state-of-the-art algorithms.

1.3 Publications

While working towards our goals, we proposed some techniques that generated the following publications:

- Pereira, T. and de Campos, T. Domain Adaptation for Person Re-identification on New Unlabeled Data[59] (best student paper award winner at VISAPP 2020)
- Pereira, T. and de Campos, T. Domain Adaptation for Person Re-Identification with Part Alignment and Progressive Pseudo-Labeling[58]
- Pereira, T. and de Campos, T. Learn by Guessing: Multi-Step Pseudo-Label Refinement for Person Re-Identification [60]

1.4 Outline

The reminder of this dissertation is organised as follows:

- **Chapter 2 - Background:** we present the theoretical background acquired while researching about person Re-ID, the background discussed in this chapter will be the base knowledge to the proposed methods in chapters 4 and 5.
- **Chapter 3 - Datasets and data augmentation:** we discuss the datasets used in our work, also we go through some data processing techniques that are useful for our task.
- **Chapter 4 - Domain adaptation on new unlabelled data:** we present the method for our first approach where we used a two phase domain adaptation framework with ResNet-50 and AlignedReID++ as backbones.
- **Chapter 5 - Multi-Step Pseudo-Label Refinement:** although promising, the results of previous Chapter highlight some deficiencies of several Unsupervised Domain Adaptation (UDA) person Re-ID methods. In this chapter, we propose a combination of techniques that addresses those limitations.
- **Chapter 6 - Proposal:** we conclude this dissertation and present a schedule with next steps to improve our work.

Chapter 2

Background

Person Re-ID is a recent challenge, the first works to use this term were published in 2005 [82]. Initially, this problem was approached extracting hand-crafted features which led to poor results and generalisation ability. Then, circa 2014, with the deep neural networks success, this challenge got more popular and diverse methods relying on Convolutional Neural Network (CNN) have been proposed. In this Chapter we present some background about CNN architectures, loss functions and transfer learning techniques that are the building blocks for a robust person Re-ID model.

2.1 Machine Learning

Machine Learning (ML) is a sub-field of Artificial Intelligence (AI) that involves computers learning to perform a task ($\mathcal{T}$) by itself, without being explicitly programmed for it. In this Section we will present some foundation concepts.

2.1.1 Definitions

The definitions and notations used in this work are based on Csurka, Pan and Yang works [11, 55].

Feature Representation and Label Space

A ML model is an algorithm able to perform a task without being previously programmed for it. The learning process of a model requires two main components, a feature space $\mathcal{X}$ and a label space $\mathcal{Y}$. In a generic manner, we can say that a ML model is a mapping function from the feature space $\mathcal{X}$ into the label space $\mathcal{Y}$.

Domain

A domain $\mathcal{D}$ is composed of a d-dimensional feature space $\mathcal{X} \subset \mathbb{R}^d$ and a marginal probability distribution $P(\mathbf{X})$, where $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathcal{X}$. For the person Re-ID challenge, a domain $\mathcal{D}$ may contain a single camera view ν, or even a set of camera views $V = \{\nu_1, \cdots, \nu_i\}$ with i cameras. As each camera view have its own characteristics as illumination, resolution, noise level, orientation, focal length, a sample set $\hat{\mathbf{X}}$ from a new camera view $\nu_\alpha \notin V$ will represent a new domain, because $P(\mathbf{X}) \neq P(\hat{\mathbf{X}})$.

Task

A task $\mathcal{T}$ is defined by a label space $\mathcal{Y}$ and the conditional probability distribution $P(\mathbf{Y}|\mathbf{X})$, where $\mathbf{X}$ and $\mathbf{Y}$ are sets of random variables (which usually are multivariate).

The person re-identification task $\mathcal{T}$ consists in learning a projection from $\mathbf{x} \in \mathcal{X}$ to a feature $\mathbf{f}$ in a Euclidean space $\mathcal{E}$ where $\mathbf{f}$ is closer to other vectors if they originated from the same person, more distant to vectors from other people. The set of labels can be thought of as the space of all possible person identities in the world, which impractical.

Alternatively, the person re-ID problem can be seen as a binary problem that takes two samples as input, indicating whether or not they come from the same person. Therefore, each person re-ID dataset (or indeed each surveillance camera environment) can be seen as a different domain, however the task is always the same, i.e., telling if two images contain the same person or not.

2.1.2 Supervised Learning

Given a particular sample set $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathcal{X}$, with corresponding labels $\mathbf{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_n\} \in \mathcal{Y}$, $P(\mathbf{Y}|\mathbf{X})$ in general can be learned in a supervised manner from these feature-label pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$.

Classification

The classification task is a classical ML challenge where the model objective is to classify an input into a desired class. For example, an image classification problem may need to distinguish different types of beverages, then the sample set $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathcal{X}$ would be composed by beverages images and the corresponing labels would be the specific beverage class (i.e. $\mathbf{Y} = \{\text{“}beer\text{”}, \text{“}soda\text{”}, \cdots, \text{“}juice\text{”}\}$). Therefore, each feature-label pair $\{\mathbf{x}_i, \mathbf{y}_i\}$ would have a beverage image and class, then the model would use this information to learn how to distinguish images from the desired classes.

Regression

The regression task aim to predict a value given an input, therefore its label space $\mathcal{Y}$ is a continuous set instead of the discrete label space from classification tasks. For example, house price prediction is a regression task where the input may contain relevant informations as neighbourhood, number of rooms, construction date, type of foundation, and the output would be the actual house price.

2.1.3 Unsupervised Learning

The unsupervised learning is a setup where there is no informations from the label space $\mathcal{Y}$, therefore $P(\mathbf{Y}|\mathbf{X})$ can not be learned. In this scenario, the alternative is to learn some patterns from the sample set $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathcal{X}$, this can be useful to group data with similar characteristics (i.e. clustering) or detect outliers (i.e. anomaly detection).

2.1.4 Metric Learning

There are tasks where the label space $\mathcal{Y}$ is mutable along the time. For example, as we said in the task definition, person Re-ID and face recognition may want to consider every human in the world as a class, then every time a person is born or dies the label space mutate. In this case, it is unfeasible to define a closed set of labels and train a model on them, therefore we define it as an open set label space.

In an open set label space there is a need to design a model able to perform the same way for seen and unseen samples/labels. To achieve this desired goal we can use a model that maps a sample from $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathcal{X}$ into a Euclidean feature space $\mathcal{E}$ where the proximity of vectors $\mathbf{f}$ will determine if they belong to the same class or not. Specifically, in the result space $\mathcal{E} = \{\mathbf{f}_1, \cdots, \mathbf{f}_n\}$ a feature pair $\{\mathbf{f}_i, \mathbf{f}_j\}$ where both features lay near each other will output $y = 1$ (same class), while distant features will output $y = 0$ (different class). Therefore, using metric learning, the problem is reduced to a binary task with $\mathcal{Y} = \{0, 1\}$.

2.1.5 Neural Networks

Neural Networks are a set of ML models inspired in the human brain, because it mimicks how the human brain process and propagate informations. Nowadays, it is a trending ML research topic and usually the first approach for several problems, however it have not always been like this.

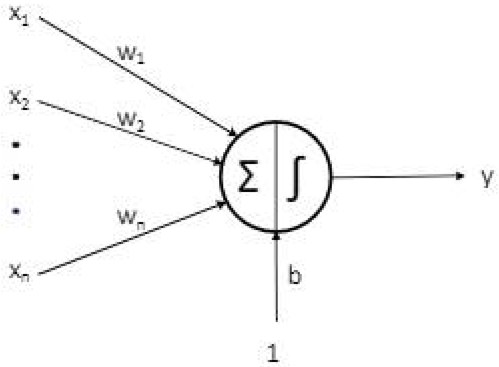


Figure 2.1: The basic Perceptron structure.

Perceptron

The Perceptron proposed by Rosenblatt [67] is the first neural network to ever exist. It simulates the operation of a single neuron and works as a linear binary classifier. The Perceptron (see Figure 2.1) have the ability to learn a weights vector $\mathbf{w} = \{w_1, w_2, \cdots, w_n, b\}$ that will be used to perform a weighted sum of an input vector $\mathbf{x} = \{x_1, x_2, \cdots, x_n, 1\}$ and the output y will be given by

$$y = \begin{cases} 1 & , \sum_{i=1}^{n} w_i x_i + b > 0 \\ 0 & , \sum_{i=1}^{n} w_i x_i + b < 0 \end{cases}. \tag{2.1}$$

The weight b stands for bias and will always have an input value of 1, this weight allow the Perceptron to learn a boundary away from the origin. Although the Percpetron is able to learn some patterns, it is restricted to linear problems only. Therefore, the Perceptron is not capable of solving real-world problems.

Multilayer Perceptron

The Multilayer Perceptron (MLP) [68] create a network of Perceptrons organized in multiple layers. As we said before, the Perceptron can be compared to a single neuron, therefore a MLP can be seen as a network of neurons and this is the origin of the name neural networks.

The input vector $\mathbf{x} = \{x_1, x_2, \cdots, x_n, 1\}$ is the same for all Perceptrons in the first hidden layer (see Figure 2.2). The outputs of each Perceptron in the first hidden leayer compose an intemediate vector $\hat{\mathbf{x}} = \{\hat{x_1}, \hat{x_2}, \cdots, \hat{x_n}, 1\}$ that is then used as input for the second hidden layer. This process of using the output of a layer as input for the next one continues throughout the entire network, which can consist of multiple hidden layers.

The composition of multiple hidden layers create non linearities in the model, therefore MLPs are capable of solving non linear problems. However, this architecture have a high

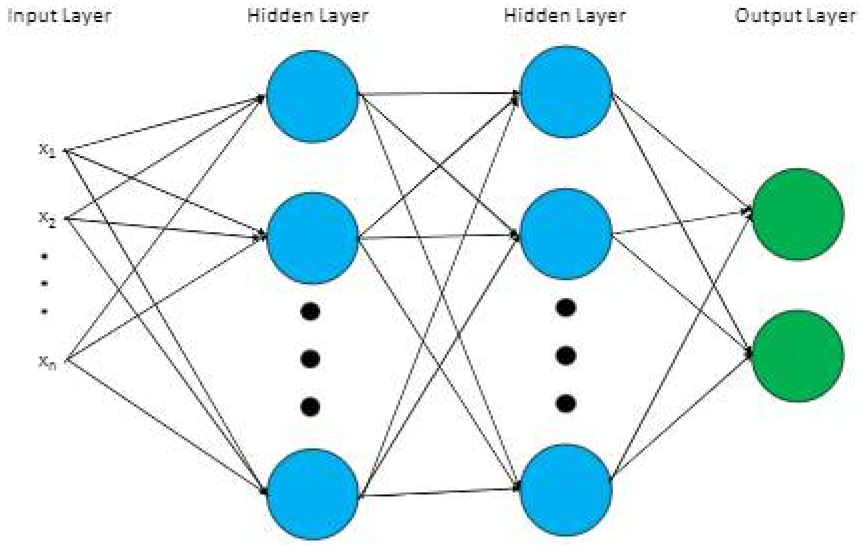


Figure 2.2: An example from a possible Multilayer Perceptron (MLP) architecture. There is a weight associated with each arrow and the activations from the Output Layer Perceptrons compose an output vector **y**.

computational cost related with the ammount of weight vectors (parameters) necessary to learn different equations for all those Perceptrons.

Convolutional Neural Networks

Prior, we generalised the input vector as $\mathbf{x} = \{x_1, x_2, \cdots, x_n, 1\}$, however "what this input vector represent?". In the regression example we pictured an input vector $\mathbf{x} = \{$*"n° rooms"*, *"date"*, $\cdots$, *"type of foundation"*$\}$ and the desired output would be $\mathbf{y} =$ *"house price"* $\in \mathbb{R}$. In this scenario, the use of a single Perceptron or a MLP is direct, however "how do we input an image into a neural network?"

An initial approach could be to reduce the 3 image channels (RGB) into a single grayscale channel and then flatten the pixels values to create a 1D vector with size $w \times h$, where w and h are the image width and height respectively. This 1D vector then would be used as the input layer of a MLP. There are two main problems with this flattened image approach:

- Each Perceptron from the first hidden layer would need to learn a weight vector with size $(w \times h) + 1$. As even small images have sizes in the order of 200×200 pixels, each Perceptron would need to learn 40001 parameters. As we said before, all this parameters require a high computational cost to be learned.
- When the image is flattened, 2D patterns are lost. Although it is possible to learn these 2D patterns from the 1D flattened array, it increase the problem complexity.

In image processing the convolution operation is used to exploit 2D patterns on an image, for example there are some convolutional kernels designed to detect edges on images or blur/sharpen them. Therefore, Fukushima and Miyake proposed the Neocognitron [22] with the first ideas of learning convolutional kernels to solve a specific problem. However, their work did not have a globally supervised learning procedure and was limited by it, then LeCun et al. [38] proposed a supervised learning approach using convolutional kernels to exploit the image 2D patterns.

Although the Convolutional Neural Network (CNN) popularity really boomed around 2010, LeCun et al.'s work was a breakthrough in the computer vision field. Their method allowed the use of diverse ML techniques for images. In addition, the CNN solved image problems, once it was capable of exploiting 2D information and needed a lot less parameters.

The CNN were the go to method to diverse applications in the 2010s, a lot of different architectures were proposed (person Re-ID architectures will be further discussed in next Section). However, recently, transformers [74] started to gain attention, primarly in the Natural Language Processing (NLP) field and now entering the computer vision field with some promising results (e.g. Vision Transformer[16]).

2.2 Neural Network Architectures

Recently, CNNs are the go to technique for several computer vision tasks. The CNN ability to extract robust features is essential for its success. There is a huge variety of CNN architectures, each one better for some tasks than others, e.g. U-Net[65] for image segmentation, faster R-CNN[63] for object detection, Inception[73] for image classification.

For the person Re-ID challenge, we need a feature extractor that can encode person information while disregarding camera variations and background noise. Therefore, we do not need Region Proposal Networks (RPN) that are present in the faster R-CNN for object detection or the contracting path present in the U-Net. We need an architecture capable of producing a strong image encoder, then architectures designed for image classification are an excellent starting point.

2.2.1 Residual Networks

The CNNs popularity boom in the first half of the 2010s lead researchers to pursue better architectures. Firstly, they thought that increasing the number of layers and creating deeper architectures would be sufficient. Although, they quickly hit a performance wall and noticed that deep architectures were overfitting quickly and got worse results than shallower architectures.

He et al. [29] noticed this dimensional problem with deeper architectures and proposed a residual block to unlock the use of very deep architectures as shown in Figure 2.3.

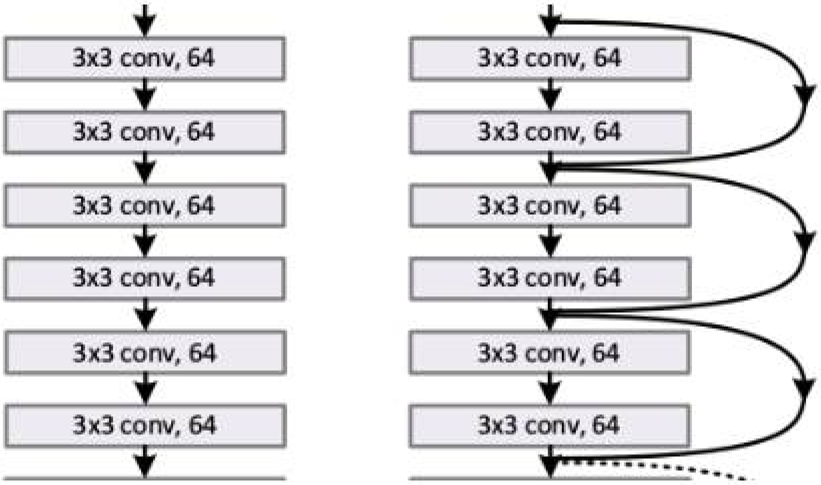


Figure 2.3: Difference between a classical (left image) architecture and a residual (right image) one. Reproduced from He et al. [29]. ©2016 IEEE.

The classical convolution blocks aimed to learn a function $H(\mathbf{X})$ to map the input $\mathbf{X}$ directly to the output $\mathbf{Y}$, while the residual block defines a function $F(\mathbf{X}) = H(\mathbf{X}) - \mathbf{X}$ which can be rewritten as $H(\mathbf{X}) = F(\mathbf{X}) + \mathbf{X}$. Their hypothesis is that for some tasks an identity mapping may be the optimal solution and the CNN should learn $F(\mathbf{X}) = 0$ and so $H(\mathbf{X}) = \mathbf{X}$. Therefore, the optimal function is mapping the input ($\mathbf{X}$) in the output ($\mathbf{Y}$), although it is a trivial equation, the classical architectures have problems to do it while the residual blocks can easily feedfoward this information. A deeper look into the residual blocks is illustrated in Figure 2.4.

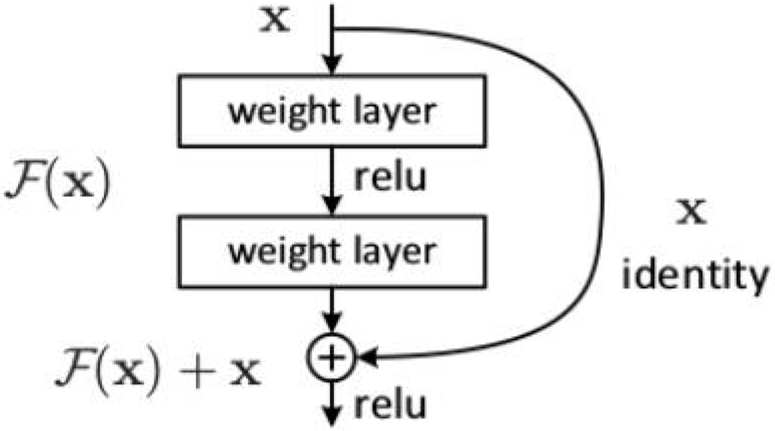

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.4: Residual block structure. Reproduced from He et al.[29]. ©2016 IEEE

Usually, the initial (shallow) layers from CNNs extract features from simple characteristics of the images and deeper layers will be responsible to extract information from more complex characteristics. To illustrate this behaviour in the person Re-ID problem, one can imagine that the initial layers will learn to identify what is background/foreground and extract information like cloth colours and bags, while deeper layers will extract informations like age, hairstyle and gender.

Although this is just an example to ilustrate the idea, it is clear that a robust person Re-ID system need to integrate informations from all those complexity levels. Therefore, the information extracted on the initial layers have to be efficiently propagated to the output, then we believe that the residual blocks from ResNet are a great tool to achieve our goal.

2.2.2 Factorization Networks

To deal with these features from multiple semantic levels, person Re-ID researchers started to work on factorization networks [1, 6]. These networks have mechanisms to ease the information propagation from multiple layers into a final fusion block. Then, they directly propagated middle layers information to deeper layers with the hypothesis that this information is important to guide the learning process.

An ideal factorization network could be illustrated as a first block which identifies what in the image is foreground and what is background, therefore creating an attention map which will be used to filter the final feature. Then, it could have subsequent blocks responsible to extract information about the person itself, like gender, age, clothes, hairstyle. Finnaly, all these simple and complex information sources would be fused by the last block to create a robust vector for that person.

Chang et al. [6] proposed a highly factorized architecture (see Figure 2.5) which rely on various intermediate layers to generate the person feature. They achieved state-of-the-art in multiple datasets to prove their architecture effectiveness.

2.2.3 AlignedReID++

Although Resnet-50 is a great neural network architecture, relying only on its generalisation capacity to perform in such a challenging task as person Re-ID is naive. One of the complications is the amount of pose variation present in this scenario, as each camera will capture the person's image from a different point of view. To deal with this kind of pose variation, some works proposed a pose-guided person Re-ID [46, 54, 62, 71] where they used an algorithm to detect the person pose (e.g. OpenPose [5]) and used this information to undermine the pose variation problem.

Even though it sounds like a great idea to identify the body parts and align the images to reduce the pose variation, it is an expensive step added to your pre-process. Therefore,

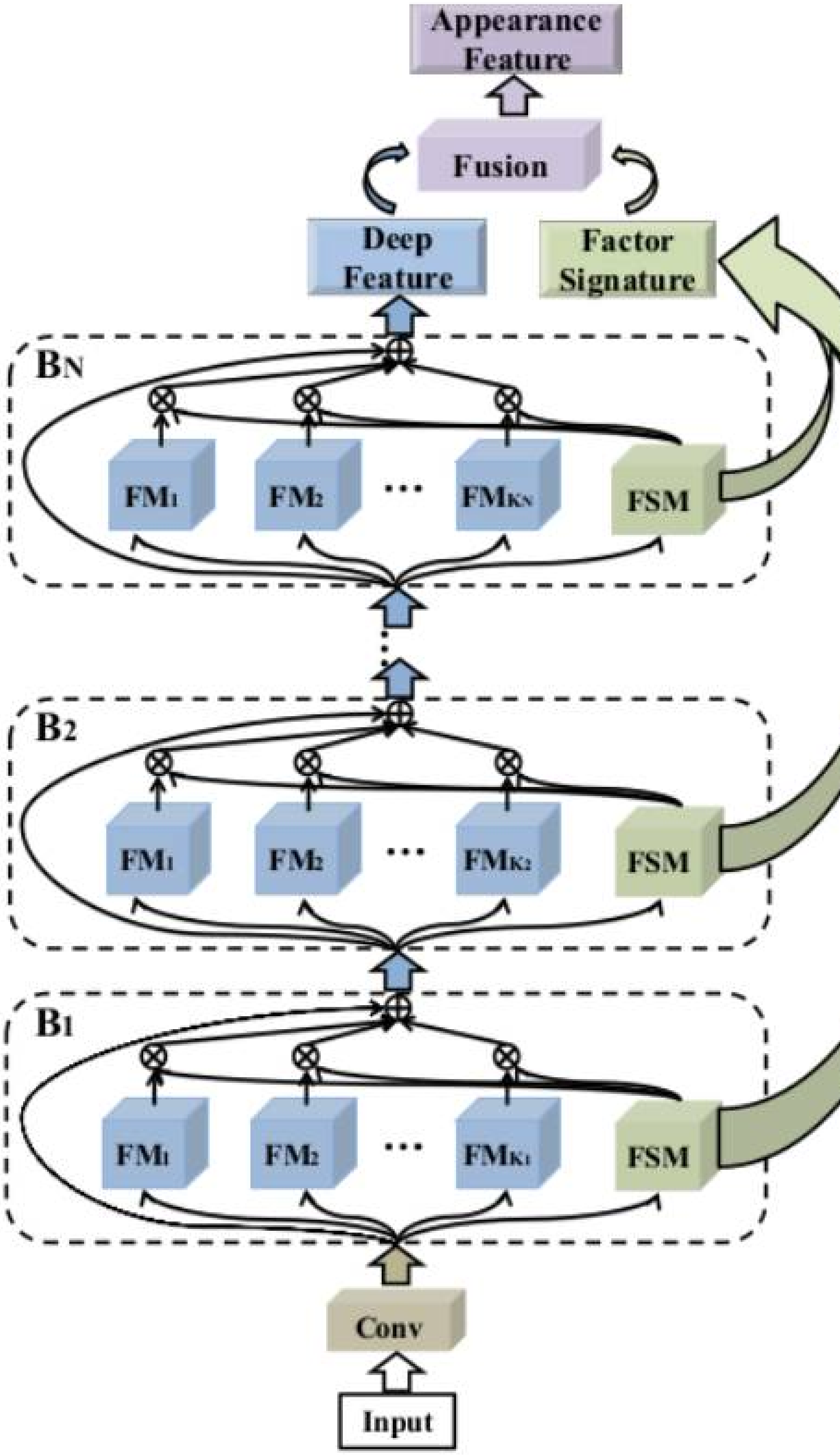


Figure 2.5: *Multi-Level Factorisation Net* (MLFN). Reproduced from Chang et al. [6]. ©2018 IEEE.

AlignedReID++ [53] proposed an architecture (see Figure 2.6) that is able to learn how to align two images without the need of a body part detection. The AlignedReID++ is not as lightweight as using only a classification network (e.g. Resnet-50), but it allow us to do an end-to-end training without detecting body parts and undermining the negative effects of the pose variations.

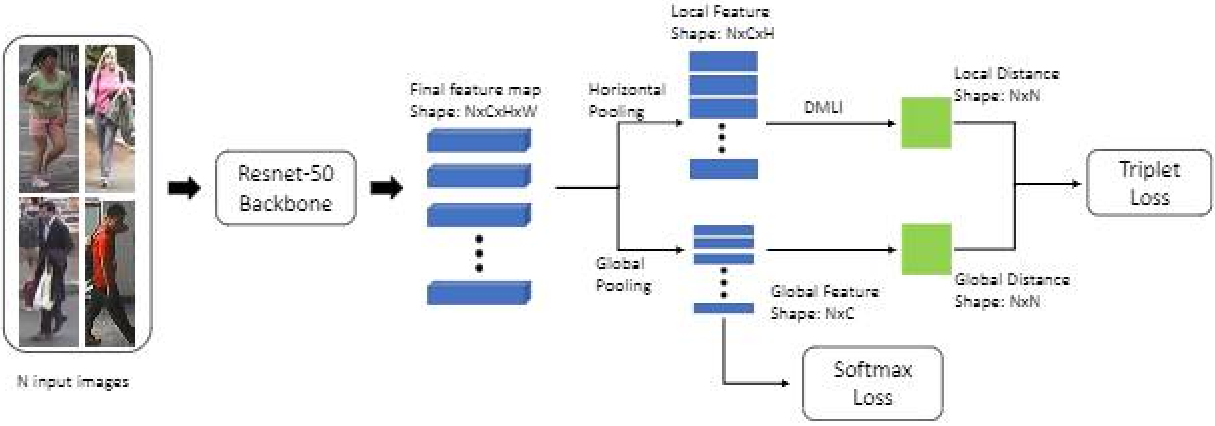


Figure 2.6: AlignedReID++ Pipeline. Adapted from Luo et al.[53].

The AlignedReID++ [53] uses Resnet-50 as a feature extractor and propagates its output to two branches, local and global. The final convolutional layer produces a feature map with dimensions $c \times h \times w$ (c is the number of channels and $h \times w$ is the spatial size). This feature map is the information that is propagated to both branches.

For the global branch, a global average pooling is used to reduce the feature map into a global feature vector $\mathbf{f}$ with size $c \times 1$. Then, this global feature vector $\mathbf{f}$ is used to calculate a cross-entropy Loss ($\mathcal{L}_{cross}$) and to calculate the global distances that will be used by the global triplet loss ($\mathcal{L}^{g}_{Tri}$). Further details about loss functions will be discussed in Section 2.3.

The local branch uses a horizontal max pooling to reduce the feature map into a $c \times h \times 1$ local feature map, which is further reshaped into the size of $h \times c$. The local feature maps are then split into horizontal regions (stripes) and compared with all the horizontal stripes from other image to calculate a distance matrix. This distance matrix has the size $h \times h$ and is used to calculate the shortest path from (1×1) to $(h \times h)$. This method is called Dynamically Matching Local Information (DMLI) and provides a local distance (shortest path) between two local feature maps. The local distances are then used to calculate the local triplet loss ($\mathcal{L}^{l}_{Tri}$).

The local branch is able to align parts of the image that may be displaced because of the camera view (Fig. 2.7). The global branch is able to extract the global image context and a class biased information (cross-entropy loss). Finally, the AlignedReID++ loss is a

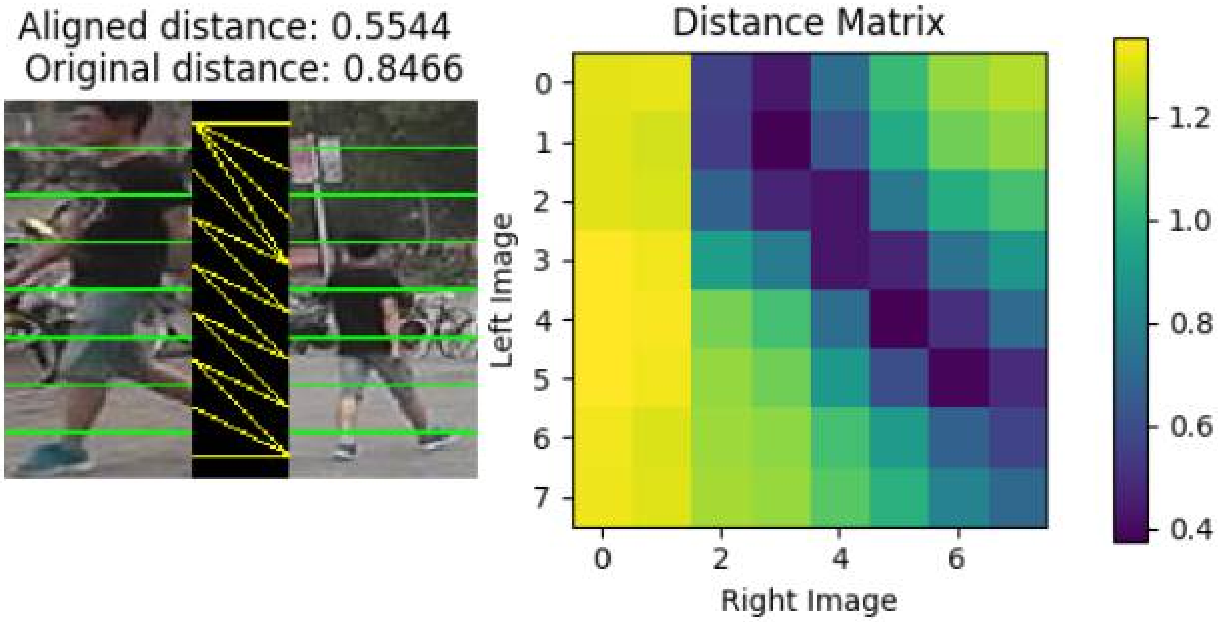


Figure 2.7: Example of how AlignedReID++'s Dynamically Matching Local Information (DMLI) is able to align two pictures that were displaced because of the camera views. The distance matrix on the right is computed by comparing stripes of the two images and their minimum path on that matrix generates the alignment shown on the left. As expected, the aligned distance is smaller than the global distance. The code used to generate this image is available from `https://github.com/michuanhaohao/AlignedReID`

combination of these 3 losses given by Eq. 2.2.

$$\mathcal{L}_{Aligned} = \mathcal{L}_{cross} + \mathcal{L}_{Tri}^{l} + \mathcal{L}_{Tri}^{g} \tag{2.2}$$

2.2.4 IBN-Net

A typical Re-ID system relies on ResNet [29] as their backbone (usually the ResNet-50 model), which is a safe choice, because Re-ID is a task that requires multiple semantic levels to produce robust embeddings and the residual blocks help to propagate these multiple semantic levels to deeper layers. Also, ResNet is a well studied CNN that leads to a step change in the performance on the ImageNet dataset [12].

However, the vanilla ResNet has its generalisation compromised because it does not include instance-batch normalisation. To deal with that, Pan et al. [56] proposed the IBN-Net50, which replaces Batch Normalisation (BN) layers with instance batch normalisation (IBN) layers. The IBN-Net carefully integrates IN and BN as building blocks, significantly increasing its generalisation ability.

2.3 Loss Functions

Loss functions are the base foundation for model optimisation, once they are the mathematical functions that guide the model update. They measure the dissimilarity between the model output and the ground truth, then the optimisation method uses it to update the model and reduce this distance. A well designed loss function guarantees that lowering its value will result in the optimisation of a model for a given data for a desired task.

2.3.1 Metric Learning vs Classification

The most popular computer vision applications have their method designed as a classification system. In a classification scenario it is important to learn how to classify the input as one of the desired classes (e.g. classify animals, beverages, objects). Then, the model will learn decision boundaries which will segment the output space into n different regions, where n is the number of classes.

A typical approach to a classification task is to design the model so it outputs a n-dimensional vector $\mathbf{f}$ where each dimension will represent a class. Then, the ground truth $\mathbf{y}$ is encoded using a one-hot encoding scheme, that is a vector of 0's with a 1 in the dimension of the corresponding class. Normally, the model output will be activated by a softmax function [35] (see Equation 2.3), therefore the sum of the output vector dimensions will be equal to 1.

$$\text{softmax}(\mathbf{f}_i) = \frac{e^{\mathbf{f}_i}}{\sum_{j=1}^{n} e^{\mathbf{f}_j}} \tag{2.3}$$

Once the experiment has been designed for a classification task, the ground truth $\mathbf{y}$ uses a one-hot encoding and the model output is activated by the softmax function, the cross-entropy loss [3]

$$\mathcal{L}_{cross} = \mathbf{y} \ln[\text{softmax}(\mathbf{f})] + (1 - \mathbf{y}) \ln[1 - \text{softmax}(\mathbf{f})] \tag{2.4}$$

is used to space the distributions as much as possible.

As follows, the model will learn a feature space specialised in separating the desired classes given the usual input.

Although a classification model is very good to classify inputs to a determined class, it may have an unexpected behaviour for an input from a new unseen class. As person Re-ID aims to re identify people one could design the model with an output dimension equal to the number of people on Earth. However, that is impossible because the number of people in the world is mutable and the computational cost to train a model of this size is immeasurable.

Therefore, we define the person Re-ID as a open set challenge where we do not have a specific number of classes. Then, we cannot use the classical softmax and cross-entropy approach as the classification task. Because, we do not want to classify a image as "person 1", what we really want is to convert an open set problem to a binary problem, where we are able to say that two images are similar enough to classify then as the same person, or not.

This idea of similar images and measuring a distance between images is the base of metric learning. Now, our goal is not to guide the output space to segment n classes and learn decision boundaries, it is to produce an output vector that belongs in a feature space where features from the same person have a small distance and features from different people have a greater distance.

2.3.2 Siamese Loss

The training strategy is fundamental to enhance the model capacity on learning a specific task, as we discussed in the prior Section the softmax combined with the cross entropy loss is ideal for classification tasks. For metric learning, the equivalent combination relies on comparation networks.

The first idea for a comparation network was the siamese network presented by Bromley et al. [4]. The original siamese network architecture received two images as input and extracted the output feature vector from each image using the same weights. Then, it compared the two output vectors using cosine similarity, expecting a high similarity ($cosine \sim 1.0$) for images from the same class and a low similarity ($cosine \sim -1.0$) for images from different classes.

The siamese architecture allow the training of metric learning models, however it still needs to be paired with a loss function to guide the weight adaptations in the right direction.

Unlike cross-entropy loss that compute its value over samples, a metric learning loss runs over pairs of samples. Therefore, a distance function is necessary to measure the distance between two output vectors $\mathbf{f}_1$ and $\mathbf{f}_2$. In Bromley et al.'s work, they used the cosine similarity

$$cos(\mathbf{f}_1, \mathbf{f}_2) = \frac{\sum_{i=1}^{n} \mathbf{f}_{1i}\mathbf{f}_{2i}}{\|\mathbf{f}_1\| \, \|\mathbf{f}_2\|} \tag{2.5}$$

as their distance function.

Although they did not inform which loss function was used to guide the model update, they said that training was carried out using a modified version of the backpropagation proposed by Lecun [37]. Therefore, they probably designed their experiment with the

Mean Squared Error (MSE) loss given by

$$\mathcal{L}_{MSE} = [cos(\mathbf{f}_1, \mathbf{f}_2) - y]^2 \tag{2.6}$$

where y is set to 1.0 when the features originated from the same class and -1.0 otherwise.

2.3.3 Contrastive Loss

The MSE loss function used by the original siamese networks is able to approximate features from the same class and push away features form different classes. However, when pushing away features from different classes, the MSE loss always push it to the maximum disance possible ($cos(\mathbf{f}_1, \mathbf{f}_2) \sim -1.0$) and by doing it the loss may approximate $\mathbf{f}_2$ from a third feature $\mathbf{f}_3$ that belongs to a third class.

The problem is then "How much to push a feature from a different class?". To deal with this problem Hardsell et al. [26] proposed the contrastive loss, which introduced a margin (m) parameter that defines a radius around $\mathbf{f}$. Therefore, dissimilar pairs only contribute to the loss if their distance is within this radius.

They decided to use the Euclidean distance

$$D(\mathbf{f}_1, \mathbf{f}_2) = \sqrt{\sum_{i=1}^{n} (\mathbf{f}_{1i} - \mathbf{f}_{2i})^2}. \tag{2.7}$$

as their distance function. Then, their contrastive loss is defined by

$$\mathcal{L}_{contrastive} = \frac{(1-y)}{2} D(\mathbf{f}_1, \mathbf{f}_2)^2 + \frac{y}{2} \{\max(0, m - D(\mathbf{f}_1, \mathbf{f}_2))\}^2, \tag{2.8}$$

where $y = 1$ for dissimilar pairs and $y = 0$ for similar pairs.

The contrastive loss is ideal when trying to learn a metric because it allows one to perform an end-to-end learning from a dataset to an embedding space. The contrastive loss receives as input a pair of feature vectors and tries to approximate them if they are from the same person or set them apart if they are from different people. This generates an embedding space where feature vectors from the same person tend to lie near each other.

2.3.4 Triplet Loss and Batch Hard

The triplet loss is an upgrade from the contrastive loss which instead of using a pair of samples as input, it uses an anchor, a positive sample and a negative sample. Therefore, the triplet loss approximates feature vectors from the same person while it also separates features of different people, according to Equation 2.9. This way, one can expect better samples separation in the embedding space:

$$\mathcal{L}_{Tri} = \max\Bigg(0\ ,\ m + D\Big(\mathbf{f}_a, \mathbf{f}_p\Big) - D\Big(\mathbf{f}_a, \mathbf{f}_n\Big)\Bigg), \tag{2.9}$$

where m is a margin that defines how much we want to push the classes away (similar to the constrative loss), $\mathbf{f}$ is the CNN output(sub indexes a, p and n mean anchor, positive and negative, respectively) and $D(\cdot)$ is the Euclidean distance (Equation 2.7).

A question that arises from the triplet loss use is "how to choose the positive/negative examples?" Hermans et al. [30] investigated this problem and came to a conclusion that the best learning is This approach was coined *batch hard* and it works as follows: for each anchor sample $\mathbf{x}_a$ from the achieved when using the hardest positive/negative samples during training. batch, the choice of positive sample $\mathbf{x}_p$ is done as the one that maximises $D(\mathbf{f}_a, \mathbf{f}_p)$ and the negative sample $\mathbf{x}_n$ is chosen as the one that minimises $D(\mathbf{f}_a, \mathbf{f}_n)$. Using this strategy, Equation 2.9 can be rewritten as

$$\begin{aligned}\mathcal{L}_{Tri} = \max\Bigg(0\ ,\ m + \max_{p} D\Big(\mathbf{f}_a, \mathbf{f}_p\Big) \\ - \min_{n} D\Big(\mathbf{f}_a, \mathbf{f}_n\Big)\Bigg),\end{aligned} \tag{2.10}$$

where positive and negative samples are chosen within each batch and the losses across all anchors in a batch are averaged out.

Figure 2.8 illustrates how samples are chosen for a batch. All the rectangles at the top represent samples from a person and the rectangles at the bottom represent samples of another person. The triplet will choose each rectangle as anchor at a time, calculate the loss for it and in the final sum all the losses. From the green rectangle as an anchor, the numbered arrows indicate the distance $D(\cdot)$ from it to the samples, where $\mathbf{f}_{p_i}$, $i = \{1, 2, 3\}$, are possible positive samples and $\mathbf{f}_{n_j}$, $j = \{1, 2, 3, 4\}$, are the possible negative samples. In a batch hard approach, $\mathbf{f}_{p_2}$ is selected as positive sample, $\mathbf{f}_{n_3}$ as negative sample and $\mathcal{L}_{Tri} = m + 0.361 - 0.490$.

2.3.5 Centre Loss

The triplet loss is responsible for grouping features from the same class and move away features from different classes, however as datasets nowadays have millions of samples it is unfeasible to apply this pull/push for all possible triplets. Then, as we discussed before, we use the batch hard strategy to make the loss more robust by using the most difficult

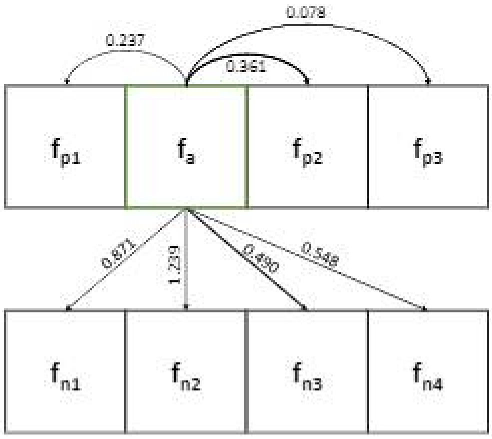


Figure 2.8: Example of how the batch hard triplet loss is computed for an anchor. The rectangles at top represent features from a person while the ones at the bottom represent features from another person. Numbers on arrows show the distance between two features ($D(\mathbf{f}_a, \mathbf{f})$), the green rectangle is the anchor and the bold arrows represent the distances selected by the batch hard.

triplets. However, as person Re-ID is an open set challenge, the testing set may have examples that would generate features extremely close to a training class.

Therefore, it is also important to minimise the intra-class feature variation, so each class is defined by a compact region in the output space. This way, we increase the model robustness, once the region of interest for each class would be more compact.

The centre loss [79] proposed by Wen et al. does that. It penalises the distance between features $\mathbf{f}_i$, $i = \{1, \cdots, n\}$ and the centroid of the class $\boldsymbol{\mu}$, so the model is guided to produce more compact clusters. The centre loss function is defined by

$$\mathcal{L}_{centre} = \frac{1}{2} \sum_{i=1}^{n} \|\mathbf{f}_i - \boldsymbol{\mu}\|^2 . \tag{2.11}$$

2.3.6 Combined losses

As we saw, each loss function presented has its own characteristics and objectives. These objectives are not opposite to each other and can work together to deliver more robust features. Therefore, a joint supervision using more than one of the losses presented may be beneficial for the problem.

Usually, the person Re-ID research is inspired by the face recognition research, because face recognition is an older and more developed challenge. Also, the recent state-of-the-art algorithms for face recognition do not rely anymore in metric learning based losses (i.e. triplet loss, centre loss and contrastive loss). This happened because the number of batches and iterations to space features vectors with a triplet based loss from million IDs would be enormous, as stated by Deng et al. [13]. Therefore, they proposed an Additive

Angular Margin Loss (ArcFace) that introduces a margin loss in the classical softmax, then they can rely on the softmax for the inter-class dispersion and in the angular margin for the intra-class compactness.

Person Re-ID datasets do not have same magnitude as face recognition ones, therefore using margin based softmax losses have not been really necessary yet. Although, using joint supervision with more than one loss is well seen. An interesting approach for the person Re-ID loss used by Luo et al. [51] is to define the loss as

$$\mathcal{L} = \mathcal{L}_{cross} + \mathcal{L}_{Tri} + \beta \mathcal{L}_{centre} \tag{2.12}$$

where $\mathcal{L}_{cross}$ is the softmax loss, $\mathcal{L}_{Tri}$ is the triplet loss and $\mathcal{L}_{centre}$ is the centre loss.

This joint supervision leverages from the softmax capacity of creating inter-class dispersion, the triplet loss capacity of create robust features to the most difficult scenarios and the centre loss ability to increase intra-class compactness. The β term used in the $\mathcal{L}_{centre}$ is to maintain all three losses in the same magnitude order, normally $\beta = 0.0005$.

2.4 Generative Adversarial Network (GAN)

Generative Adversarial Network (GAN) is a class of neural networks proposed by Goodfellow et al. [24], which aims to produce new images as output. The GAN framework have been used in a wide range of applications including person image generation [34], synthesising images from text [85], increasing image resolution [39], blending images [80] and deblurring images [36].

The GAN framework consists of training, simultaneously, two neural networks: a generator G and a discriminator Φ. The generator G is responsible to learn the database probability distribution $P(\mathbf{X})$ while the discriminator Φ predict if the image is from the original database or if it was generated by G. It is an adversarial process because G's objective is to maximise Φ's error rate, therefore G's goal is to learn how to map a white noise $\mathbf{z}$ in a way that $P(G(\mathbf{z})) \sim P(\mathbf{X})$. In this case, Φ would not be able to distinguish between an original image $\mathbf{x} \in \mathbf{X}$ and a generated image $G(\mathbf{z})$.

The GAN input then is a white noise $\mathbf{z}$ with a probability distribution $P(\mathbf{z})$ and G is a mapping function $G(\mathbf{z})$ that maps $P(\mathbf{z})$ into $P(G(\mathbf{z}))$ while approximating $P(G(\mathbf{z}))$ to the original data distribution $P(\mathbf{X})$. While $\Phi(\mathbf{x})$, $\mathbf{x} \in \{\mathbf{X} \cup G(\mathbf{z})\}$ is a classifier to predict if $\mathbf{x}$ is a real image or a generated one. Therefore, the GAN training is given by a minmax game with a dual objective: **a)** minimise the shift between $P(G(\mathbf{z}))$ and $P(\mathbf{X})$, and **b)** maximise $\Phi(\mathbf{x})$ error rate.

To achieve its goal, the GAN loss function is given by

$$\mathcal{L}_{GAN} = \min_{G} \max_{\Phi} \Big[\mathbb{E}_{\mathbf{x} \sim P(\mathbf{X})} \; \log \Phi(\mathbf{x}) + \mathbb{E}_{\mathbf{x} \sim G(\mathbf{z})} \log \Big(1 - \Phi \big(G(\mathbf{z}) \big) \Big) \Big], \tag{2.13}$$

although this is the full representation of the GAN loss, it is not feasible to maximise Φ error at the same time as minimising G error. Therefore, the training have the following two steps and keep alternating between them:

- A Φ error maximisation step using gradient ascent in Equation 2.14;
- A G error minimisation step using gradient descent in Equation 2.15.

$$\mathcal{L} = \max_{\Phi} \Big[\mathbb{E}_{\mathbf{x} \sim P(\mathbf{X})} \; \log \Phi(\mathbf{x}) + \mathbb{E}_{\mathbf{z} \sim G(\mathbf{z})} \log \Big(1 - \Phi \big(G(\mathbf{z}) \big) \Big) \Big] \tag{2.14}$$

$$\mathcal{L} = \min_{G} \Big[\mathbb{E}_{\mathbf{z} \sim G(\mathbf{z})} \; \log \Big(1 - \Phi \big(G(\mathbf{z}) \big) \Big) \Big] \tag{2.15}$$

2.4.1 CycleGAN

Since the proposal of GANs, various interesting methods using GANs have been published. In between all GAN applications, Isola et al. [33] proposed a method to translate an image from a source domain $\mathcal{D}^s$ (e.g. sketchs) to a target domain $\mathcal{D}^t$ (e.g. real objects), as illustrated in Figure 2.9. However, in real-world situation it is rare to have paired images from $\mathcal{D}^s$ and $\mathcal{D}^t$. Usually, there is plenty of data available from both domains, but the inter-domain data have no relations between them. Therefore, the challenge is to learn a mapping function $G : \mathcal{D}^s \rightarrow \mathcal{D}^t$, where $P(G(\mathbf{X}^s)) \sim P(\mathbf{X}^t)$, without examples of how a specific $\mathbf{x}^s \in \mathbf{X}^s$ would look like in $\mathbf{X}^t$.

The fact that there are not paired images available increase the challenge complexity, because there are infinite mapping functions G that may present the ideal quantitative result during training, but this does not guarantee the expected qualitative results.

To deal with this problem, Zhu et al. [98] proposed a method capable of producing the ideal quantitative and qualitative results. Their proposed method is called CycleGAN, because besides learning a generator $G : \mathcal{D}^s \rightarrow \mathcal{D}^t$ there is also a generator $\hat{G} : \mathcal{D}^t \rightarrow \mathcal{D}^s$ and an extra term in the loss function to approximate $\hat{G}(G(\mathbf{X}^s)) \approx \mathbf{X}^s$.

The cycleGAN loss function is then given by Equation 2.16, where $\mathcal{L}_{GAN}$ is the classical GAN loss defined by Equation 2.13, λ is a hyperparameter to control the influence of the cycle loss ($\mathcal{L}_{cyc}$) given by Equation 2.17:

$$\mathcal{L} = \mathcal{L}_{GAN}(G, \Phi_{\mathbf{X}^t}) + \mathcal{L}_{GAN}(\hat{G}, \Phi_{\mathbf{X}^s}) + \lambda \mathcal{L}_{cyc}(G, \hat{G}), \tag{2.16}$$

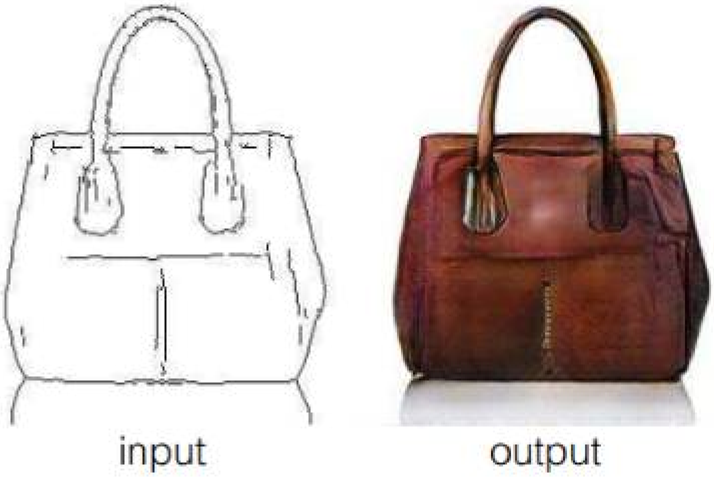


Figure 2.9: Example of paired images to auxiliate the training of and image translation GAN. In this example, the GANs objective is to learn the mapping between sketchs and real object images. Reproduced from [33]. ©2017 IEEE.

where

$$\mathcal{L}_{cyc}(G,\hat{G}) = \mathbb{E}_{\mathbf{x}^s\sim P(\mathbf{X}^s)}\Big[\left\|\hat{G}(G(\mathbf{x}^s)) - \mathbf{x}^s\right\|_1\Big] + \mathbb{E}_{\mathbf{x}^t\sim P(\mathbf{X}^t)}\Big[\left\|G(\hat{G}(\mathbf{x}^t)) - \mathbf{x}^t\right\|_1\Big]. \qquad (2.17)$$

2.5 Transfer Learning

It is well known that deep neural networks need a huge amount of clean and annotated data in order to learn good metrics for a determined problem. Also, it is assumed that testing data will be in the same feature space and have the same distribution as training data, which often not true.

There are some strategies to deal with this data shift, as using data normalisation to reduce this variance in data, or using data augmentation to build a more robust model, or using transfer learning [55] to leverage the previous knowledge and ease the process of learning in a new feature space.

In a deep feedfoward neural network, the deeper the layer it will have more abstract representations from the input. Then, a model trained with a large dataset (e.g. ImageNet [12]) in a fully supervised manner will learn a wide range of abstract representations that may be useful for several tasks, even if they are a little different from the original one. Donahue et al. [15] proved that leveraging the knowledge from these large datasets is beneficial, therefore it is naive to not use these technique.

As one can see in Figure 2.10 there are multiple types of transfer learning, whether the task is maintained, if labelled data are available for source and target domains, or even if there are not labelled data at all, etc.

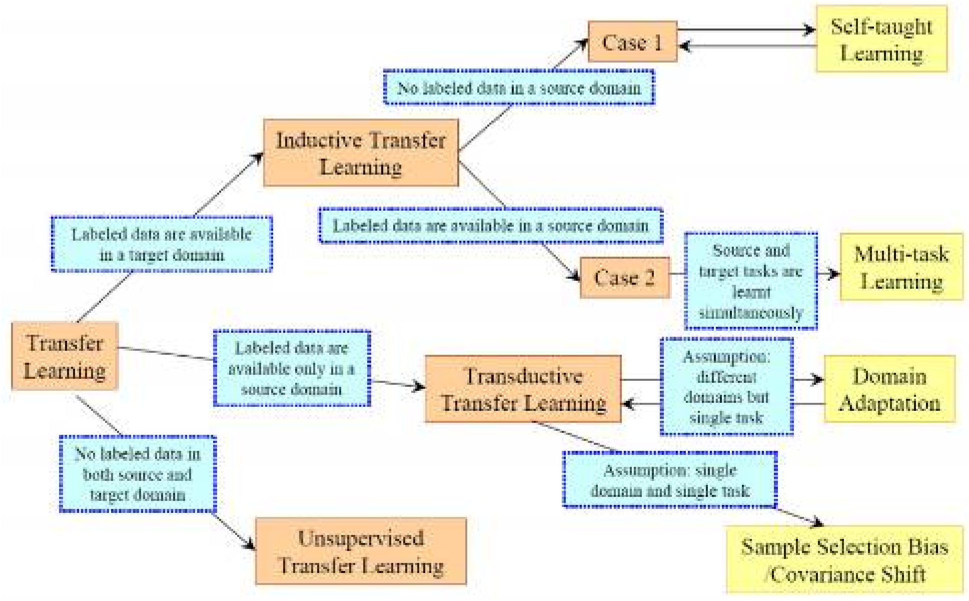

Case 1
Self-taught Learning
No labeled data in a source domain
Inductive Transfer Learning
Labeled data are available in a target domain
Labeled data are available in a source domain
Case 2
Source and target tasks are learnt simultaneously
Multi-task Learning
Transfer Learning
Labeled data are available only in a source domain
Transductive Transfer Learning
Assumption: different domains but single task
Domain Adaptation
No labeled data in both source and target domain
Assumption: single domain and single task
Unsupervised Transfer Learning
Sample Selection Bias /Covariance Shift

Figure 2.10: Transfer Learning taxonomy based on availability of data and maintenance of task. The real-world person Re-ID challenge is classified as a Transductive Transfer Learning, once there are available data only in a source domain, however it is a single task. Reproduced from Pan and Yang [55]. ©2010 IEEE.

Furthermore, Person Re-ID usually uses CCTV cameras, therefore there is a high variance factor in this challenge, once each camera have its own characteristics as illumination, angle, distance from people, saturation, resolution, distortion, etc. We consider that each camera view is a domain and people will have different appearance in different domains. So, domain adaptation techniques are very important for person Re-ID.

2.5.1 Domain Adaptation

As discussed before, each camera view can be seen as a domain because of its characteristics. However, the typical person Re-ID dataset have images from multiple cameras annotated, therefore one could train a model using examples from the same person in multiple cameras views. This way we are capable of reducing the multiple feature spaces of each camera to a single feature space that belongs to that group of cameras.

For simplicity, let us assume that there are two domains: a source domain $\mathcal{D}^s = \{\mathcal{X}^s, P(\mathbf{X}^s)\}$ with $\mathcal{T}^s = \{\mathcal{Y}^s, P(\mathbf{Y}^s|\mathbf{X}^s)\}$ and a target domain $\mathcal{D}^t = \{\mathcal{X}^t, P(\mathbf{X}^t)\}$ with $\mathcal{T}^t = \{\mathcal{Y}^t, P(\mathbf{Y}^t|\mathbf{X}^t)\}$. Those domains are different $\mathcal{D}^s \neq \mathcal{D}^t$, because $P(\mathbf{X}^s) \neq P(\mathbf{X}^t)$ due to domain shift. Also, we do not have the target domain labels $\mathbf{Y}^t$, so we do not have the feature-label pairs $\{\mathbf{x}_i, y_i\}$ to learn $P(\mathbf{Y}^t|\mathbf{X}^t)$ in a supervised manner.

As one can see in Figure 2.10, Pan and Yang [55] defined domain adaptation as the situation where there is a single task ($\mathcal{T}^s = \mathcal{T}^t$), but labelled data is only available in source domain. We find ourselves in the same situation, however once $\mathcal{Y}^t$ is not available the Person Re-ID research community named this as Unsupervised Domain Adaptation (UDA), so we will use this term from now onwards.

The UDA setup usually start training a model in the source domain to learn $P(\mathbf{Y}^s|\mathbf{X}^s)$ and use the related information to learn $P(\mathbf{Y}^t|\mathbf{X}^t)$ without annotating target domain images. Recent research on unsupervised domain adaptation (UDA) for person Re-ID has two leads:

GAN based methods [14, 47, 77, 84, 95, 100]

The ability of GANs to generate images that follows the distribution of a given feature space is perfect for UDA, because one can approximate images from two domains. Zhu et al. [98] proposed a cycle-consistent GAN that is able to generate images from a new domain without the need of paired images during training which opened a lot of opportunities for the person Re-ID community.

Deng et al. [14] used the cycleGAN to generate an intermediate dataset where the source domain images have been transformed to appear similar to those from the target domain. This way they produced a dataset that leveraged from the source domain annotations and had similar characteristics to the target domain. Their work beat the state-of-the-art at the time.

Zhai et al. [84] used GANs to augment the target domain training data, so they could create images that preserved the person ID and that simulates other camera views at the same time. This strategy maximises the inter-class distance with a more diverse sample space and minimises the intra-class distance with more diversity on the person image.

Methods Based on Pseudo-Labels [18, 21, 23, 45, 59, 83, 84, 100]

The idea behind the pseudo-labels model is to use an unsupervised method to create labels for the dataset. The classical way to generate those pseudo-labels is to use a pre-trained model to extract the features from target domain, use this features to predict the label space for this unlabelled target domain, assume those predictions are correct and use then to fine-tune a model previously trained on source domain.

This method is commonly used for person Re-ID unsupervised domain adaptation because it allows the model to train with the real images from the target domain without the need of manual annotation. As we discussed, each CCTV cameras normally used for the person Re-ID task have a lot of unique characteristics and even can be seen as a domain, then using the actual images generated by the camera is useful to learn robust features.

This approach has shown remarkable results and is the idea behind current state-of-the-art UDA Re-ID methods. The drawback with pseudo-labels is that if the domains are not similar enough, they can lead to negative transfer, because the labelling noise might be too high. To deal with that, Ge et al. [23] propose a soft softmax-triplet loss to leverage from pseudo-labels without overfitting their model. Zeng et al. [83] propose a hierarchical clustering method to reduce the influence of outliers and use a batch hard triplet loss to bring outliers closer to interesting regions so they could be used later on.

2.6 Data Augmentation

As ML models learn from the given data, one can imagine that more data presented will result in a better model. Although that is usually true, there are three common problems in real-world applications:

- Not enough data is available;
- The data is not a good representation from the application Domain;
- The data is too repetitive, there are few variations.

These three problems may harm the model learning process and result in a model with poor generalisation for unseen images. It is then essential to deal with them before the training stage.

Data augmentation is a set of techniques used to increase the ammount of available data. These techniques will slightly change the data to create new examples with more variation, then it directly solves the problem of few and repetitive data. For the case where the data is not a good representation of the application Domain, there are some advanced data augmentation techniques to undermine it as we shall present in this Section.

2.6.1 GAN domain approximation

In this method, we have images from source domain $\mathbf{X}^s$ and target domain $\mathbf{X}^t$, but we do not have the labels from target domain $\mathcal{Y}^t$. So, we approximate data from images of a known source domain to images of a target domain generating an intermediate dataset.

An unsupervised domain adaptation method can be used to generate an intermediate dataset $\mathcal{D}^i$ that leverages the source domain annotation $\mathcal{Y}^s$ and is similar to the target domain. For that, we follow an approach based on GANs [24]. More specifically, we use a CycleGAN using the method proposed by Zhu et al. [98] and applied to person re-identification by Deng et al. [14] and by us in [59].

The idea is to use images from the source domain ($\mathbf{X}^s$) as input and train a GAN to generate outputs which are similar to the images from the target domain ($\mathbf{X}^t$). However, once we have no paired images between domains the problem has a high complexity. Zhu et al. proposed to train two generators G and $\hat{G}$ where $G : \mathcal{X}^s \rightarrow \mathcal{X}^t$ is a mapping from the source domain to the target and $\hat{G} : \mathcal{X}^t \rightarrow \mathcal{X}^s$ is a mapping from the target domain to the source. Also, the cyclic component presented in Equation 2.17 is added to the loss.

The cyclic component is there to do an identity match between source domain images $\mathbf{X}^s$ and their double transformed pairing images $\hat{G}(G(\mathbf{X}^s))$, and vice-versa. By minimising this cyclic loss we expect to have transformations that can map both domains.

Therefore, we use the generator $G : \mathcal{X}^s \rightarrow \mathcal{X}^t$ in all images of our source domain to generate an intermediate dataset. That is, we create a dataset that leverages from the labelled data of the source domain and have similar characteristics to the target domain. This way we can expect that a training on the intermediate dataset will perform well in the target domain.

2.6.2 Camera style adaptation

GAN based methods have been widely used to produce more data for training. These methods are able to fulfil gaps that are left on the dataset, as disbalance of images per camera or absence of images from the same person in a specific pair of cameras. The lack of data which cause these problems can have a negative effect in the training phase.

To deal with those problems, Zhong et al. [95] proposed a camera style adaptation method. They trained a cycleGAN for each camera pair and used these models in every dataset image to translate it to all other cameras views. Therefore, given a source domain $\mathcal{D}^s$ with 6 cameras views $V = \{\nu_1, \nu_2, \nu_3, \nu_4, \nu_5, \nu_6\}$, each image $\mathbf{x}_i \in \mathcal{X}^s$ from camera ν_i will be translated to all the other 5 views.

The generated images preserve the person identity and simulate the view from other cameras, therefore facilitating the process of learning how the person appearance is transformed between every camera pair. This method is used for data augmentation during the training phase and the ratio between original images and GAN generated images is $1 : 3$.

Although the GAN generated images give them more munition to learn specific camera transformations, the GAN images are noisy. Then, there is a need to use label smoothing regularisation in the generated images.

2.6.3 Random Erasing

In real world person Re-ID systems there are problems that are not always present in public datasets, like the problems related with errors in the pedestrian detector and occlusions [52]. The second one is very common in the real world, primarily when the system is applied in locals with a high volume of persons, e.g. shopping malls, airports, subways.

The occlusion problem generate partial images of the person, where the person may be occluded by other person or by an object. To deal with the occlusions, some works relied in part detection systems [10, 40, 88, 96], therefore they knew which parts from the person were visible and learned how to weight the visible and the occluded parts from the person to compare the images. Although this strategy led to good results, it includes one more algorithm that is subject to errors thus increasing the risk of error propagation and increasing the computational cost.

We believe that the best way to deal with occluded images is to have examples of them in the training dataset, so the CNN will learn robust features against the occlusion by itself. We therefore use Zhong et al. [93] random erasing method for data augmentation. At each training batch we select 50% of the images to be randomly cropped, therefore simulating occlusions.

Although the use of random erasing method is very interesting to deal with occlusions, Luo et al. [51] showed that this data augmentation method is not beneficial for domain adaptation. Because the method erases part of the image, the CNNs end up relying on other domain-guided characteristics to learn the person identity and therefore do not generalise very well for other domains.

2.7 Evaluation Metrics

In order to evaluate a model performance and compare it with other methods, evaluation metrics are needed. The main evaluation metrics used for the person Re-ID challenge are the Mean Average Precision (mAP) and the Cumulative Matching Characteristics (CMC). In addition, some person Re-ID domain adaptation techniques rely on pseudo-labels which are generated by clustering methods. In this section we provide a brief explanation of the above mentioned person Re-ID metrics as well as clustering evaluation metrics.

2.7.1 Precision and Recall

Precision and recall are two of the most used machine learning metrics. Precision measures the proportion of positive predictions which are correct. On the other hand, given a set of positive class samples, recall measures what proportion of them are correctly predicted as positive.

There are four possible outcomes for every prediction:

- **false positive (FP):** when the model predicts false, but it was a true example;
- **false negative (FN):** when the model predicts true, but it was a false example;
- **true positive (TP):** when the model predicts true and it was a true example;
- **true negative (TN):** when the model predicts false and it was a false example.

with these possible outcomes, we can define precision and recall as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.18}$$

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{2.19}$$

2.7.2 Mean Average Precision ($\boldsymbol{mAP}$)

Usually, the model output represents the probability or a likelihood of the input belonging to a given class (classification learning), or the distance between two inputs (metric learning). Therefore, a threshold value is needed to determine if the model answer is considered positive or negative. For each threshold value used, the model will present different precision and recall values (e.g. a threshold of 0% will classify every example as positive and will result in a recall of 100%, but with a considerable drop in the precision).

As a threshold variation will modify the precision and recall values, we may plot a precision × recall graph and use it to help determine an optimal threshold. The precision × recall graph is a great tool to get a global picture of the performance of the model under all thresholds. The average precision (AP) is given by the area under the curve (AUC) of this plot and works as a robust measure to summarise it. A perfect system generates a graph with an AUC of 1, which indicates precision of 100% for all nonzero recall values.

In the person Re-ID challenge, we define each person ID as a class. Then, we are able to plot a precision × recall graph and calculate a value of average precision (AP) for each person ID. We therefore, may summarize our models performance on a given test dataset using the mean average precision (mAP) metric for the entire dataset (see Equation 2.20).

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{2.20}$$

2.7.3 *Cumulative Matching Characteristics* (CMC)

The most used metric in the person Re-ID challenge is the cumulative matching characteristics (CMC). As the person Re-ID is basically an image retrieval task, this metric evaluates how good the model is to retrieve images from the correct class.

We can eval the CMC metric for different ranks, where the CMC metric for a rank τ indicate the percentage of cases where the correct prediction was in between the τ most similar images given a rank prediction. Therefore, if we set $\tau = 1$, for each image from our testing set we will check if the first image retrieved image by the model is from the same class, if so we consider it correct for the CMC Rank-1, otherwise it is wrong. After doing it for all the images, we see the percentage of the correct predictions and this value indicates the CMC Rank-1.

The Figure 2.11 illustrates the difference between the CMC and the mAP metrics. In this figure, each square indicates an image, where the green square is an image from the same class as the query and the red ones are from other classes. In this example we would have a CMC Rank-1 of 100% that could mislead us and indicate that the model is perfect, however if we check the mAP we would have a value of 90.33%.

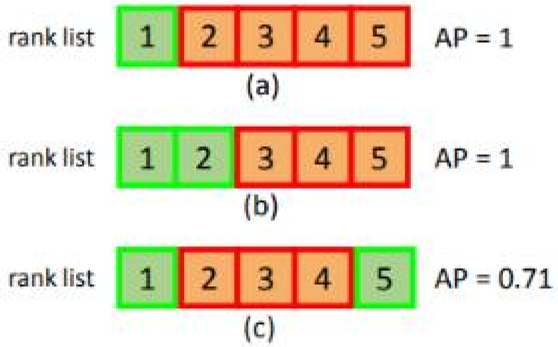


Figure 2.11: Comparision between mAP and CMC. For this example, we have the CMC Rank-1 as 100%, but this do not happen for all AP values and, therefore, for the mAP. Reproduced from [89]. ©2015 IEEE.

2.7.4 Cluster Evaluation Metrics

A person Re-ID model trained as a metric learning system yields a feature vector for each image. In a scenario where we do not have labels from a person Re-ID dataset, we may extract all image vectors with our model and cluster those vectors. We then assume that each cluster represents a person ID and generates pseudo-labels for this previously unlabelled dataset. There are many different clustering techniques to do this and in this case we have errors associated with both: the person Re-ID model and the clustering technique used, then it is important to have some metrics that enable us to evaluate the quality of the generated clusters.

For a scenario where we have access to the real dataset labels, we may use the V-measure (Λ) [66], which is an entropy-based metric to evaluate the clustering quality. The V-measure consists of two criteria that must be satisfied to achieve an optimal cluster assignment, these criteria are:

- **Completeness** (ζ): the proportion of samples from a given class which are in the same cluster;
- **Homogeneity** (ξ): for each cluster, measures the proportion of samples which belong to the same class.

To understand how the completeness and the homogeneity metrics are calculated, let us assume a dataset with $\mathcal{N}$ data points, and two partitions of these: a set of classes, $C = \{\delta_i | i = 1, \ldots, n\}$ and a set of clusters $K = \{\kappa_j | j = 1, \ldots, m\}$. Let $\mathit{\Lambda} = \{\alpha_{ij}\}$ be a clustering solution, where α_{ij} is the number of data points from the class δ_i that are in cluster κ_j.

To achieve a good homogeneity metric, a clustering solution must assign only data points from a single class to a single cluster. This could be done by assigning only one data point per cluster, however it would present a poor completeness score unless each class has only one example. To achieve a perfect homogeneity score, the conditional entropy of the class distribution given the proposed clustering $H(C|K)$ must be 0. As the size of $H(C|K)$ varies with $\mathcal{N}$ and C, it is normalised by $H(C)$. Therefore the homogeneity score may be calculated as:

$$\xi = \begin{cases} 1 & \text{, if } \; H(C|K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{, else} \end{cases} \tag{2.21}$$

where

$$\begin{aligned} H(C|K) &= -\sum_{\kappa=1}^{|K|}\sum_{\delta=1}^{|C|}\frac{\alpha_{\delta\kappa}}{\mathcal{N}}\log\frac{\alpha_{\delta\kappa}}{\sum_{\delta=1}^{|C|}\alpha_{\delta\kappa}} \\ H(C) &= -\sum_{\delta=1}^{|C|}\frac{\sum_{\kappa=1}^{|K|}\alpha_{\delta\kappa}}{n}\log\frac{\sum_{\kappa=1}^{|K|}\alpha_{\delta\kappa}}{n}. \end{aligned} \tag{2.22}$$

The completeness criteria is symmetrical to homogeneity. To satisfy this criteria, a clustering solution must assign all data points from a single class in a single cluster. Therefore, in a perfect case $H(K|C) = 0$, however in the worst case scenario, each class have example in every clusters with a distribution proportional to the distribution of cluster sizes, then the max possible $H(K|C)$ equals $H(K)$. Therefore, symmetric to the homogeneity calculation, completeness is given by:

$$\zeta = \begin{cases} 1 & \text{, if } \; H(K|C) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{, else} \end{cases} \tag{2.23}$$

where

$$\begin{aligned} H(K|C) &= -\sum_{\kappa=1}^{|K|}\sum_{\delta=1}^{|C|}\frac{\alpha_{\delta\kappa}}{\mathcal{N}}\log\frac{\alpha_{\delta\kappa}}{\sum_{\kappa=1}^{|K|}\alpha_{\delta\kappa}} \\ H(K) &= -\sum_{\kappa=1}^{|K|}\frac{\sum_{\delta=1}^{|C|}\alpha_{\delta\kappa}}{n}\log\frac{\sum_{\delta=1}^{|C|}\alpha_{\delta\kappa}}{n}. \end{aligned} \tag{2.24}$$

Finnaly, the V-Measure (Λ) metric is given by the harmonic average between completeness and homogeneity as follows:

$$\Lambda = \frac{(1+\lambda)\cdot\xi\cdot\zeta}{(\lambda\cdot\xi)+\zeta}, \tag{2.25}$$

where λ is the weight attributed to homogeneity.

2.8 Final Remarks

In this Chapter we reviewed some concepts about CNN architectures, loss functions, transfer learning and metrics to evaluate classification and clustering results. They are the pillars to real world person Re-ID models and even small changes on them can result in improvements. In Chapters 4 and 5 we are going to present how we deal with each of these pillars to achieve state-of-the-art.

Chapter 3

Datasets

As the Machine Learning (ML) methods learn from the data presented, it is important to analyse the datasets and understand what kind of data you are using. Specially, CNNs demand a huge amount of data to obtain great results, but public datasets are limited. Therefore, we need to use a strong set of data augmentation techniques to produce more data without losing critical informations.

During the course of our work, we used four publicly available datasets for training and validation of our models.

Each dataset has its own characteristics and can be seen as a domain. Identifying those specific characteristics is essential to understand the complexity behind the person Re-ID domain adaptation challenge. Therefore, in this Chapter we analyse each one of them.

3.1 Viper

The Viper dataset released by Gray et al. [25] in 2007 is the oldest dataset used in our work. It only contains 1264 images from 632 different persons, therefore for each person only one image was captured per camera. To capture these images, the authors used 2 distinct cameras, however they were not fixed cameras, so their position have been relocated multiple times during the data collection. These moving cameras generate a high variance on the images characteristics, so even with only 2 cameras they have challenging setup.

The authors forced some kind of variations to increase even more the diversity of images. The main variation applied by the authors was an angle variation, capturing images with the camera angle varying from 45 to 180 degrees. Also, the data collection process lasted a few days, therefore illumination changes are present.

They recorded all the scenes and processed the video files to create the dataset. All images from the dataset have a resolution of 48×128 pixels, although some of these images have clearly been distorted to achieve these dimensions. The Figure 3.1 show images from the same person in different views from this dataset.

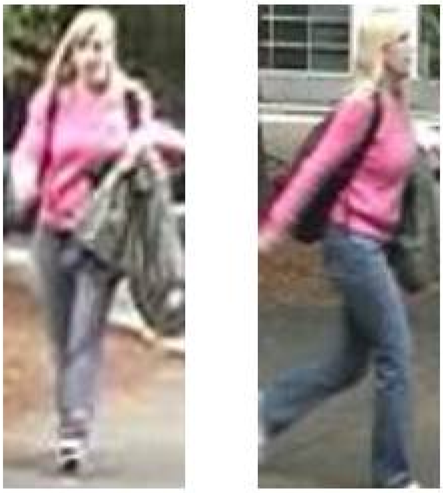


Figure 3.1: Example images from the same person in different cameras from Viper dataset[25]. ©2007 IEEE.

3.2 CUHK03

The CUHK03 dataset released by Li et al. [42] in 2014 has 13164 images from 1360 different persons (an average of 4.8 images per person per camera). To create this dataset, they used up to 6 different cameras, however each person appears in only 2 from these 6 cameras. Also, there are 2 released setups available for this dataset, the first relied on an object detection algorithm to annotate the person bounding box, while the other setup produced the person images using manual annotations. In our work, we choose the manually annotated setup.

All the 6 cameras used are security cameras from CCTV systems, therefore illumination variability and occlusions are common (this problem is minimised for the manually annotated setup). The image resolution also has some variation because of the different cameras used, however the average image resolution is 100 × 300 pixels. In Figure 3.2 we present an example of how the same person look in different views from CUHK03.

3.3 Market1501

The Market1501 dataset released by Zheng et al. [89] in 2015 is one of the most used dataset for person Re-ID research nowadays. This dataset is great to train CNNs, because

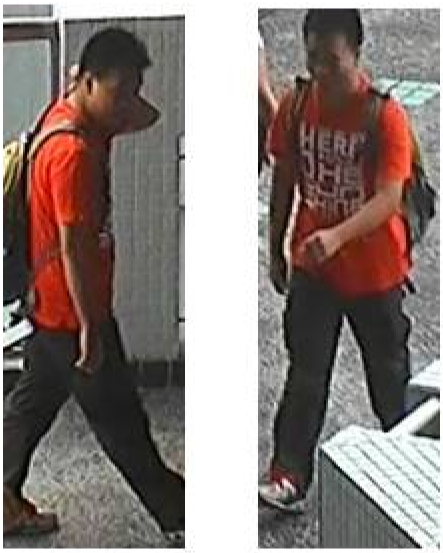


Figure 3.2: Example images from the same person in different views from CUHK03 dataset[42]. ©2014 IEEE.

of the amount of available data. There are more than 32 thousand images from 1501 different people in 6 distinct camera views, averaging 3.6 images per person per camera. Also, in this dataset they have some examples where the person was seen in all 6 cameras, which is great for the person Re-ID learning once you have examples that show how someone appears in all different views.

While creating this dataset, the authors listed 3 problems they wanted to solve:

- The other available datasets did not have enough data to train deep CNNs;
- The images from other datasets were manually annotated, reducing the real world factor from the challenge;
- There were few example images per person in other datasets.

To solve these problems, the authors gathered more than 32 thousand images from people in a real market and merged these images with around 3 thousand distractor images (a group of images where no person is seen). All these images were acquired and annotated using the using the Deformable Part Model (DPM) [20] as pedestrian detector. Finally, these images were collected in an uncontrolled open space and they were able to collect examples from the same person in multiple views (for some persons all views were available).

All the images from this dataset have been resized to 64×128 pixels, Figure 3.3 shows example images from the same person in different views from this dataset.

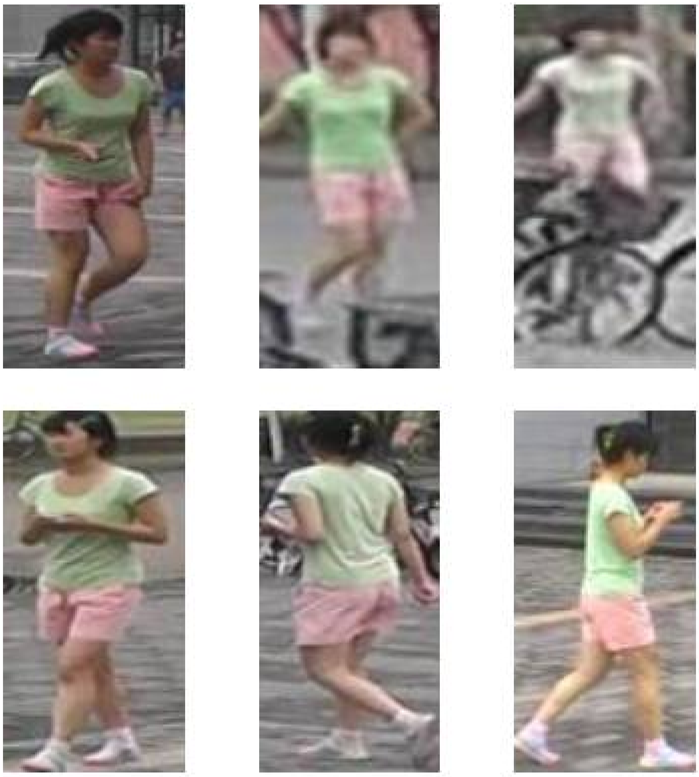


Figure 3.3: Example images from the same person in different cameras from Market1501 dataset [89]. ©2015 IEEE.

3.4 DukeMTMC

Originally, the DukeMTMC dataset [64] was created to help accelerate the progress in multi-target, multi-camera (MTMC) tracking systems. Ristani et al. recorded 85 minutes videos from 8 distinct high resolution cameras at Duke University campus. Therefore, they had 8×85 minutes of video recorded at 1080p, 60fps with more than 2800 identities to perform multi-camera tracking.

Then, in 2017, Zheng et al. [91] processed the DukeMTMC dataset to create the DukeMTMC-reID dataset, which is a subset of the DukeMTMC specific for image-based person Re-ID. For that, they followed the format of Market1501 and cropped pedestrian images every 2s of each video, leading to a total of 36411 images from 1812 identities, where 1404 identities appear in, at least, two cameras and the other 408 identities only appeared in one camera (these IDs were considered distractors). The 1404 identities that appear in more than one camera were randomly separated in training/testing groups, so the dataset have 16522 images from 702 IDs for training and 19889 images from 702 + 408 (distractors) IDs for testing.

The identity which appear in most distinct cameras is in the training set and have the ID 0071, this man appeared 42 times in 6 different cameras. Figure 3.4 show example images from him in different views from this dataset.

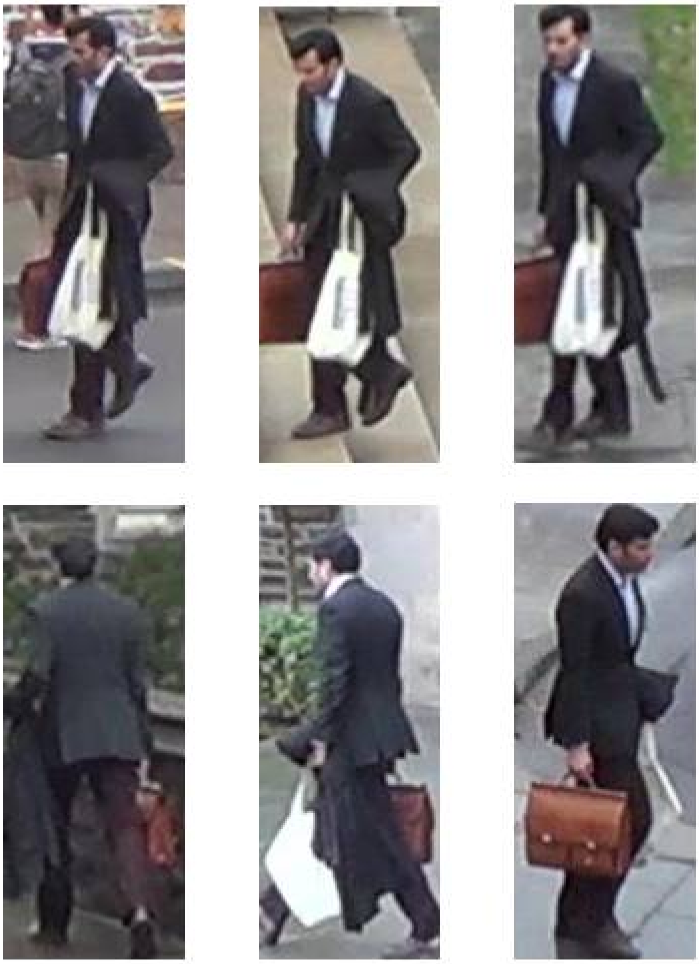


Figure 3.4: Example images from the person 0071 in 6 different cameras from DukeMTMC-reID dataset [91].

3.5 Final Observations

As our work proposes a person Re-ID model that is ready for the real world challenges, we need to simulate it in our experiments. Therefore, having multiple datasets that were captured in real scenarios is excellent for us. Also, the unsupervised domain adaptation techniques that we are going to propose in the next two chapters do not rely in the data annotation which is the exactly situation that someone would face in the real world.

In Table 3.1 we present the statistics for all used datasets[1]. It is interesting to notice how the ammount of samples in a dataset increased over time, this is direct related with

[1] We are aware of other Person Re-ID datasets as the Person30K [2], MSMT17 [77], CUHK02 [41] and PRID [31]. We have not conducted experiments in these datasets because PRID and CUHK02 were too small to train deep neural networks, Person30K was released after we performed our experiments and we had technical difficults obtaining MSMT17 dataset.

the CNNs popularity and need for more training data. In addition, the variety of scenes, indoor for CUHK03 and outdoor for the others, will be an important factor to validate our models generalisation capacity. However, the Viper dataset is really small and their cameras were not fixed during the process of gathering data, these problems will reflect in our models performance.

Table 3.1: An overview of the statistics from each dataset used in this work. This table was inspired in Bai et al.'s work [2]

	Viper [25]	CUHK03 [42]	Market1501 [89]	DukeMTMC [64]
Release Year	2007	2014	2015	2016
Samples	1264	28192	32668	36411
Identities	632	1467	1501	1812
Cameras	2	2	6	8
Avg Number of Cameras Passed per Identity	2	2	4.42	2.67
Scene	outdoor	indoor	outdoor	outdoor

Chapter 4

Domain adaptation on new unlabelled data

4.1 Overview

Person Re-ID models are usually applied on surveillance systems, such as CCTV images. Therefore, there is no clear pattern for the images, once each camera has it own characteristics as illumination, angle, saturation, resolution, distance from people, etc. Then, we define each camera, or group of cameras, as a domain and the addition of a new camera or the modification (hardware or position) of an existent camera will change the domain.

With all this diversity it is challenging to create a model robust to domain variations. The person Re-ID challenge then has diverse possible setups, each one trying to solve a different case. These setups can be divided in three main groups: fully supervised (in-domain) person Re-ID, generalisable person Re-ID and UDA person Re-ID.

In our work we aim to create a person Re-ID algorithm that is feasible for industrial use. As creating the perfect generalisable model sounds impossible, our research focuses on UDA methods that are able to leverage information from a public annotated dataset and adapt its knowledge to perform well in a new unlabeled dataset (domain).

In Chapter 1 we set three auxiliary goals to achieve our main objective, Chapters 2 and 3 presented the theme knowledge we acquired. In this Chapter, we address the first two auxiliary goals of creating a baseline method and tackling its problems.

Firstly, we adopt a simple approach with a basic Resnet-50 backbone as baseline to perform UDA in person Re-ID. Then, we improve our backbone with the AlignedReID++ and analyse how a more robust model is key for domain adaptation.

4.2 Methodology

4.2.1 Training Strategies

To have an initial boost [15], we start with a ResNet-50 CNN pre-trained on ImageNet [12]. We then transfer learn it to the problem of person Re-ID using a public dataset. This is done by replacing the last fully connect layer by a new fully connected layer with 128 features which are used as an embedding for metric learning. We use Adam optimiser and the triplet loss.

As we know, person Re-ID is typically approached as a metric learning problem, then a siamese-like loss is the ideal choice, which allows one to perform an end-to-end learning from a dataset to an embedding space. Therefore, we choose the triplet loss which uses a triplet anchor against the siamese pair. This way, one can expect better samples separation in the embedding space. Also, as we saw in Section 2.3.4, it is important to use the batch hard strategy alongside the triplet loss.

However, the batch hard will always work the worst case scenario and this decision substantially increases the training complexity. Then, to take advantage from batch hard while controlling the training complexity we propose a batch scheduler algorithm to decrease the number of negative samples and lower the training complexity.

4.2.2 Batch Scheduler

Our batch scheduler algorithm (see Algorithm 1) was designed to ease the training convergence, and once the training is converging we slowly increase the batch size γ (and therefore its complexity, having an impact in the loss). This enables us to learn step by step and converge the training even with a noisy dataset.

Algorithm 1 Batch Scheduler

```
$\gamma = 2 \times \varsigma$
for $i = 0$ to $epochs$ do
    $loss = train(i, \gamma)$
    if $loss < (0.8 \times m)$ then
        $\gamma = \gamma \times 2$
    end if
end for
```

While training with the triplet loss, the goal is to make $D(\mathbf{f}_a, \mathbf{f}_p) < D(\mathbf{f}_a, \mathbf{f}_n)$ ($D(\cdot)$ is the Euclidean distance). However, if the batch is big, the number of negative examples is way bigger than the number of positive examples, particularly in the case of person Re-ID. It is therefore possible to have a negative sample that is nearer to the anchor than the hardest positive sample. This way the loss will always be greater than the margin ($\mathcal{L}_{Tri} > m$), then the optimiser learns that outputting vectors of 0s will reduce the loss to the margin, i.e., ($\mathcal{L}_{Tri} = m$).

In Algorithm 1, m is the loss margin of Eq. 2.9 and ς is the number of samples for each person ID, we used $\varsigma = 4$. The training start with samples from 2 person IDs per batch. When $\mathcal{L}_{Tri} < m$ the training converged, because this is only possible if the CNN can distinguish the person IDs, as shown in Eq. 4.1. In line 5 of the algorithm we used a 0.8 factor to ensure this convergence.

$$\mathcal{L}_{Tri} < m \Leftrightarrow D(\mathbf{f}_a, \mathbf{f}_p) < D(\mathbf{f}_a, \mathbf{f}_n) \tag{4.1}$$

Once the convergence is ensured, we can go one step further and increase the training complexity. Then, we double the batch size, doubling the number of person IDs per batch. This process is repeated until we reach the final epoch or the maximum GPU memory.

For this work, we used a NVDIA GTX 1070 Ti GPU with 8 GB of VRAM, so the maximum batch we could reach had 88 images (22 person IDs). We recognise this still is a small batch and recommend experiments to use up to 256 images per batch.

Smith et al. [69] argue that increasing the batch size instead of decreasing the learning rate results in a faster training convergence. This argument is based in the scale of random fluctuations in the optimiser given by

$$g = \varepsilon\left(\frac{\mathcal{N}}{\gamma} - 1\right). \tag{4.2}$$

Where $\mathcal{N}$ is the training set size, γ represents the batch size and ε is the learning rate.

Assuming a big training set $\mathcal{N}$, Eq. 4.2 can be approximated by $g \approx \varepsilon\mathcal{N}/\gamma$. Therefore, increasing the batch size or decreasing the learning rate should have the same impact in the noise scale. However, increasing the batch size leads to a significantly reduction in the number of parameter updates needed, speeding up the training.

Also, the initial high noise scale allows us to explore a larger fraction of the loss function without becoming trapped in local minima. This way, we believe that the slow increase in the training complexity may lead us to a better region in the parameter space. Therefore, we reduce the noise scale and fine-tune the parameters to find a promising local minimum.

4.2.3 Domain Adaptation Strategies

In Subsection 4.2.1 we presented our training strategy that is initially used to train a CNN in the source domain, which is our baseline and will be evaluated in the target domain as the direct transfer method. To improve our model performance in the target domain we will use the two domain adaptation strategies discussed above.

Intermediate Dataset Generation

As discussed in Section 2.6.1, a CycleGAN may be used to generate an intermediate dataset that leverages from the source domain labels and approximates the images to the target domain appearance. We use this intermediate dataset to fine-tune our model trained in the source domain and, hopefully, improve its performance in the target domain.

We are aware that image-to-image translation has become an incredibly active research field in the last couple of years[1], and that CycleGAN is no longer the state-of-the-art for unpaired translation. Even CycleGAN's authors have published a more recent method that not only improves over the original one, but it is also much faster for training [57].

However, at the time we proposed our first method [59], CycleGAN was the state-of-the-art for unpaired translation. Therefore, we stick to it in [58] to ensure our benchmarks are compatible and focus the comparisons on other aspects.

Pseudo-Labels Generation

For this method, we use the CNN to extract all features $\mathbf{f}_i^t$ from target domain images $\mathbf{X}^t$ and these features belong to an Euclidean vector space. Then, we used a clustering algorithm to group these features, using the obtained group identifications as target domain with pseudo-labels $\mathbf{Y}^t$. In addition, we fine tune the CNN using the feature-label pairs $\{\mathbf{x}_i, y_i\}$ with the real images from target domain and the pseudo-labels generated by the clustering algorithm.

Even though the pseudo labels generated may contain some errors, this next training step uses the real images from target domain $\mathbf{X}^t$. Therefore, the CNN is able to learn more robust features for the target domain, because it learns the exact characteristics of the target domain.

We choose the k-means [27] clustering algorithm to group the features in the Euclidean vector space. The value of k was chosen as a proportion of the size of each target dataset. Table 4.1 indicates the values used in this work. However, the naive assignment of samples to clusters is a flawed strategy to annotate the data, because a simple look at the data may cluster viewpoints rather than people. In other words, features from different people taken from the same camera view are often more similar to each other than features from the same person from different camera views.

[1] See e.g. `https://paperswithcode.com/task/image-to-image-translation`

Table 4.1: The chosen k for each dataset when using k-means algorithm.

Dataset	k
CUHK03	2000
Market1501	1600
Viper	632

Our solution is to use k-means algorithm to generate k clusters for each camera view, then use a nearest neighbour algorithm to associate these clusters across the camera views. This way, we guarantee that every person from our pseudo-labels space has images from each camera. That results in a noisy annotation, because that assumption is not a true in the real label space of the dataset. However, using this approach we ease the CNN task of learning features robust for multiple camera views and achieve better results in validation.

Progressive Learning

The first pseudo-labels generated in target domain are often inaccurate and may not lead our method to a significant improvement. However, even these inaccurate and noisy pseudo-labels allow our method to learn some features from the target domain, such as a person appearance in new viewpoints. Learning these features help our model to disregard camera-specific information and focus on the people.

High quality pseudo-labels are key to unlock our framework's full potential and prevent negative transfer, hence the need to keep improving the pseudo-labels quality. Hehe et al. [18] then proposed progressive learning, which is an iterative technique composed of two parts:

- generating target domain pseudo-labels to train the model without labeled data;
- fine-tuning the model with previously generated pseudo-labels;

After each iteration, the model is expected to become better suited to generate new pseudo-labels in the target domain as it learns from it. Such approach has been used in shallow domain adaptation methods in the past as well [19] and [50] for standard classification tasks.

Therefore, in [58] we improve our previously proposed framework [59] with the progressive learning strategy to update the pseudo-labels in target domain. We keep iterating over the progressive learning loop until our model achieves convergence. We will demonstrate this method effectiveness with results in Subsection 4.4.3.

4.3 Qualitative Results

4.3.1 Intermediate Dataset Generation

As said in section 4.2.3 our method tries to approximate the source domain to the target domain. This is done training a cycleGAN between both domains and using the generator to create an intermediate dataset that shifts the source domain samples so that they become more similar to the target domain data. The idea is to generate images that preserve the person morphology, but are visually adapted to the target domain. While there is no guarantee that a GAN preserves person morphology, the cyclic loss contributes towards this goal, as it has an identity match component.

Figure 4.1 presents examples of transformation results between all domains. It is interesting to note that the person morphology have been well preserved and the changes have been more in the colours, texture and background. That means we could produce a great approximation of how a person would appear in the view of another dataset.

The CUHK03 dataset was created using surveillance cameras from a university in Hong Kong with an elevated viewpoint, so normally the background of their images consists in a granular floor. While the Market1501 dataset was created with cameras in a park, so the images usually have grass in the background of their views. Viper is the oldest dataset used in this work, it was published in 2007 and is composed of low resolution outdoor images.

These characteristics of the datasets make it easy to understand the effects seen in Figure 4.1. When using CUHK03 as the target domain, the transformed images tend to have a granular background to approximate the floor texture in CUHK03 images. When using Market1501 as target domain, images from CUHK03 had a background transformation from the granular floor to grass, and images from Viper had just a colour transformation, because both datasets are from outdoor images. When using Viper as target domain, images from Market1501 had a colour transformation and images from CUHK03 had a texture background transformation and a brightness enhancement.

4.3.2 Pseudo-Labels Method

Although the cycleGAN did a great job shifting images between domains, when using the pseudo-labels method we can achieve even better results. This is because the training is now performed with the actual target domain images and estimated pseudo-labels. So, there is no longer the problem of images in which the person morphology was not preserved. The target dataset characteristics are better represented.

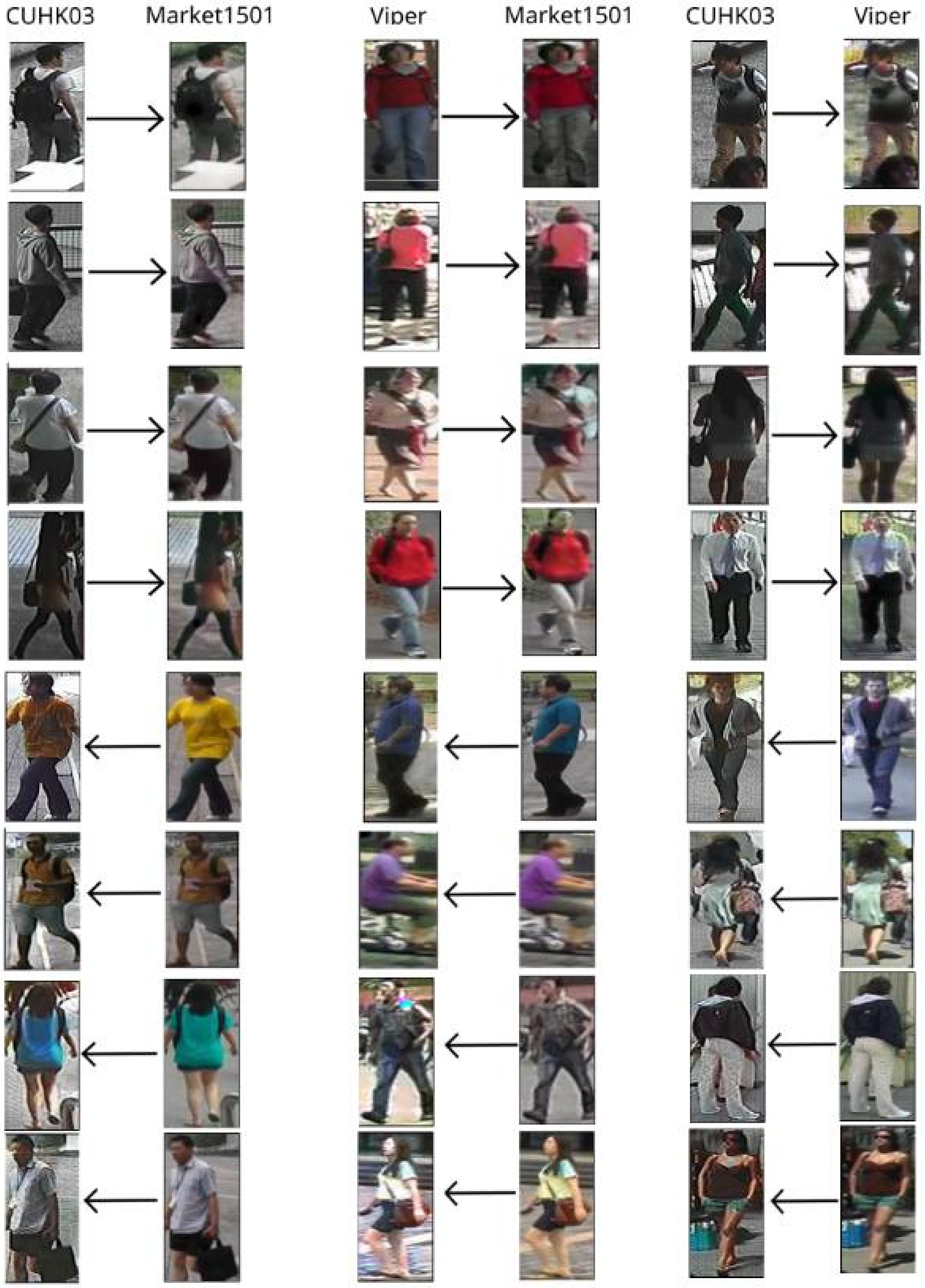


Figure 4.1: Examples of the cycleGAN transformations between domains.

Table 4.2: CMC accuracy results (in %) using Rank-1, Rank-5 and Rank-10, obtained using one dataset as source domain and another as target. The Backbone used to obtain these results was the Resnet-50. As for the methods, Direct refers to application without transfer and Ours is the combination of CycleGAN and pseudo-labels (without progressive learning) for domain adaptation.

			CMC Accuracy(in %)		
Source	**Target**	**Method**	**Rank-1**	**Rank-5**	**Rank-10**
Market	**Viper**	Direct	12.5	25.0	33.1
		CycleGAN	9.8	26.9	36.4
		Ours	**13.9**	**29.0**	**40.7**
	CUHK03	Direct	19.9	49.4	63.2
		CycleGAN	34.8	66.7	79.1
		Ours	**38.2**	**69.7**	**81.6**
CUHK03	**Viper**	Direct	10.1	22.5	29.0
		CycleGAN	11.6	25.5	34.7
		Ours	**13.6**	**33.9**	**46.0**
	Market	Direct	26.8	45.9	55.1
		CycleGAN	35.8	56.5	65.7
		Ours	**37.3**	**60.4**	**70.4**
Viper	**CUHK03**	Direct	5.9	18.1	29.0
		CycleGAN	31.9	64.4	77.5
		Ours	**36.1**	**69.2**	**81.3**
	Market	Direct	5.7	15.5	22.2
		CycleGAN	6.7	17.0	23.7
		Ours	**8.6**	**20.5**	**28.4**

Figure 4.2 illustrates the dataset created using pseudo-labels – as one can see the estimated labels are not perfect, but the grouped images show a strong colour similarity. Also, the effectiveness of progressive learning is clearly visible in Figure 4.2, as the cluster obtained without using this method (left cluster in Figure) clearly has multiple person IDs with some variety. On the other hand, the cluster obtained using progressive learning have a stronger clothes similarity (e.g. everyone wearing shorts) besides the colour one.

4.4 Quantitative Results

4.4.1 Resnet-50 as Backbone

The cycleGAN method was compared with the direct transfer method, where the direct transfer method consists in evaluating in the target domain a CNN trained in the source domain without further training. The direct transfer method therefore shows how different the domains are and is used as a baseline.

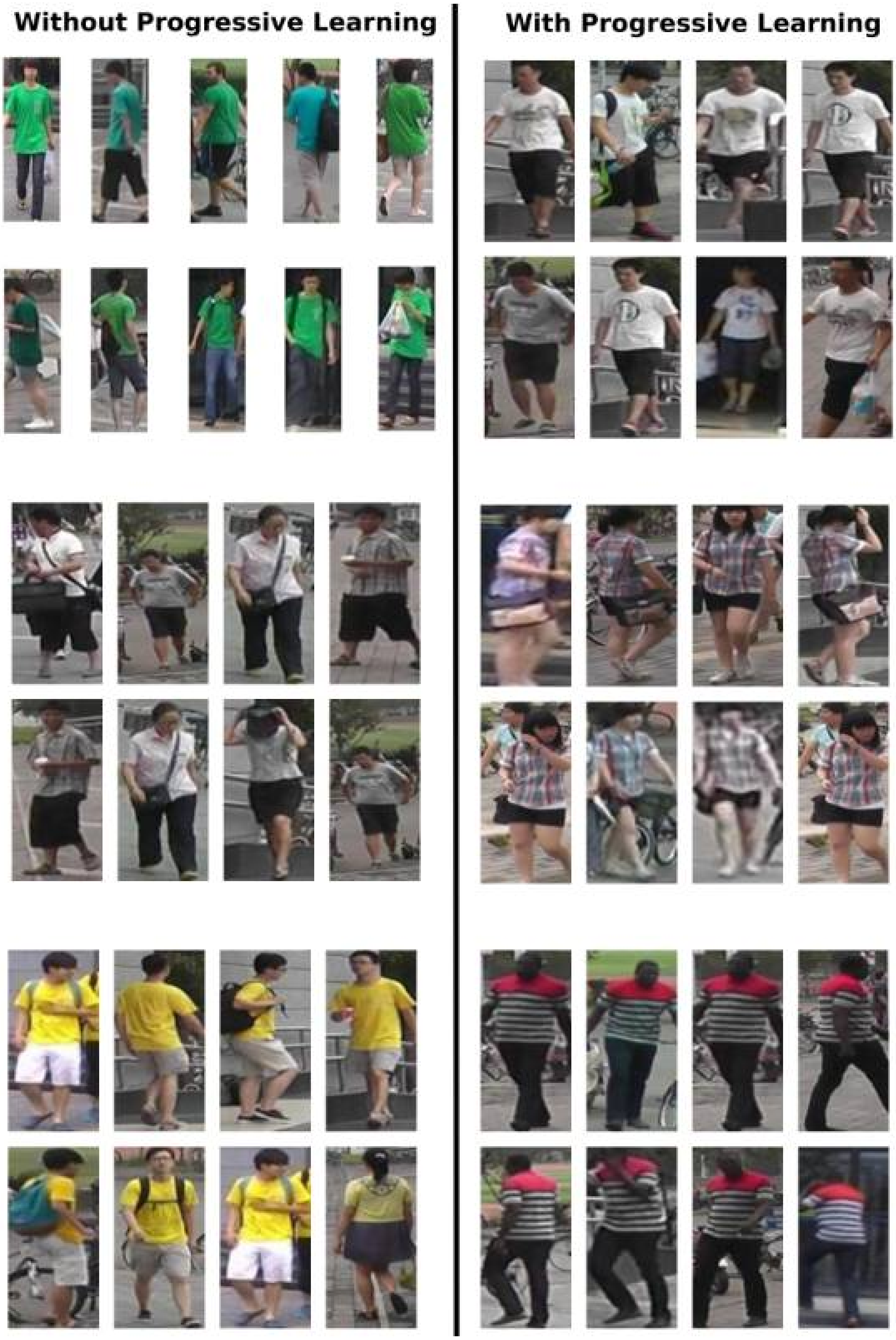


Figure 4.2: Images from two final clusters when using the pseudo-labels method. The clusters were obtained using Viper as source dataset and Market1501 as target dataset. Although different people were included in this cluster, the attributes of their clothing are similar. Furthermore, the left cluster was obtained without using progressive learning, while the right one was obtained after the use of progressive learning.

As one can see in Table 4.2 the cycleGAN method presents huge rank-1 improvements when using CUHK03 as target domain (26% improvement for Viper as source domain and 14.9% improvement for Market1501 as source domain). This happens because the CUHK03 images have granular background texture as a strong characteristic that was easily learned by our cycleGAN.

A great rank-1 improvement was also obtained for Market1501 as target domain and CUHK03 as source domain, where the cycleGAN method achieved a 9% improvement compared with the baseline. Furthermore, for Market1501 as target and Viper as source domain our method achieved 1% improvement, meaning that the colour transformation helped to approximate these domains, but this was not as significant as texture changes that occurred when working with CUHK03 images.

For Viper as a target domain the cycleGAN method achieved 1.5% rank-1 improvement using CUHK03 as source domain and 1.9% rank-5 improvement for Market1501 as source domain. Again, this means that texture transformations are more significant than colour transformations. Although those are not our best results, they are very significant because Viper is an old dataset it has far less images than the others (only 1264 images), so learning to create the intermediate dataset in a unsupervised manner without much data is extremely hard.

As one can see in Table 4.2, our pseudo-labels method showed great improvements in all test cases. Even when using the Viper dataset as target domain our method could improve the cycleGAN results in 2% or more. For the Market1501 dataset the rank-1 improvement was around 2% also and for the CUHK03 our method achieved improvements of 4% in rank-1 accuracy.

It is important to notice that the pseudo-labels have a stronger positive impact on smaller target datasets. This is because small datasets require fewer clusters to annotate the data. This was a significant factor for the improvements we obtained for the Viper dataset as target domain.

In summary our method is significantly better than direct transfer without adaptation. It is important to emphasise that our method does not make use of any label from the target domain, completely removing the burden of annotating new data when the application domain changes.

4.4.2 AlignedReID++ as Backbone

As one can see in Table 4.3 the pseudo-labels method always give the best CMC Rank-1 results. This is the same case as in the Table 4.2 and proves the effectiveness of our domain adaptation model and the advatage of using the original images to train the model, even though they are not with the perfect labels.

Table 4.3: CMC accuracy results (in %) using Rank-1, Rank-5 and Rank-10, obtained using one dataset as source domain and another as target. The Backbone used to obtain these results was the AlignedReID++. As for the methods, Direct refers to application without transfer and Ours is the combination of CycleGAN and pseudo-labels (without progressive learning) for domain adaptation.

			CMC Accuracy (in %)		
Source	**Target**	**Method**	**Rank-1**	**Rank-5**	**Rank-10**
Market	**Viper**	Direct	22.9	**41.8**	50.0
		CycleGAN	21.4	40.2	50.3
		Ours	**23.7**	41.5	**50.8**
	CUHK03	Direct	22.5	45.0	58.0
		CycleGAN	37.0	69.1	80.9
		Ours	**42.9**	**72.5**	**81.2**
CUHK03	**Viper**	Direct	20.6	38.0	47.2
		CycleGAN	21.8	43.2	52.2
		Ours	**22.5**	**43.2**	**54.1**
	Market	Direct	38.7	55.1	62.6
		CycleGAN	42.7	59.7	67.3
		Ours	**46.8**	**65.9**	**73.6**
Viper	**CUHK03**	Direct	9.9	27.9	40.1
		CycleGAN	17.1	41.6	55.8
		Ours	**20.4**	**43.9**	**58.5**
	Market	Direct	15.9	28.2	35.4
		CycleGAN	23.1	37.9	45.8
		Ours	**28.4**	**46.4**	**55.2**

It is interesting to compare Tables 4.3 [58] and 4.2 [59], because the difference between the reported methods is the backbone architecture used, and they highlight that the contribution of AlignedReID++ is clear.

Using this state-of-art method as a feature extractor allowed us to achieve improvements from 4.1% up to 16.4% in CycleGAN method. The only domain combination that did not give a better result was using the Viper as source domain and CUHK03 as target domain.

When it comes to our full method, AlignedReID++ brings an improvement of up to 19.8%. Although the result with Viper as source domain and CUHK03 as target domain was not the expected, this is not our method's fault. If we analyse the Table 4.3 with this domain combination the CycleGAN method could not provide the same results as when only the Resnet-50 was used as backbone, then even with a 3.3% improvement with our method, the result still is bellow expected for the AlignedReID++.

4.4.3 Ablation Studies

In this Subsection we perform ablation studies to show the influence of each component of our method separately and the impact that the progressive learning has in our method. All the results reported in this Subsection were obtained using the AlignedReID++ as the backbone.

Batch Scheduler

In order to analyze the batch scheduler contribution, we performed experiments with and without the batch scheduler algorithm using the new method (based on AlignedReID++) and domain adaptation with CycleGAN and CycleGAN&pseudo-labels (Ours). We have not performed experiments with the batch scheduler for direct transfer because for the Market1501 and CUHK03 datasets we used pre-trained weights from the AlignedReID++ paper [53] and the Viper dataset does not have enough data to profit from the batch scheduler algorithm.

Considering only the rank-1 results shown in Table 4.4, we have 8 test cases where it was better not to use the batch scheduler and 4 test cases that indicate the opposite. Although the majority of test cases indicates that the batch scheduler does not help, 4 of these 8 cases use the Viper images for training (adapted or not). The problem is that the Viper dataset has only 1264 images, then the assumption that we made in Eq. 4.2 when we said that N was big enough to approximate the Equation to $g \approx \varepsilon N/B$ does not hold for this dataset. Because of that, the -1 factor in Eq. 4.2 has a strong contribution and

Table 4.4: CMC accuracy results (in %) using Rank-1, Rank-5 and Rank-10, obtained using one dataset as source domain (Src.) and another as target (Trg.). The column BS stands for batch scheduler and indicates whether the batch scheduler algorithm have been used or not. As for the methods, Ours is the combination of CycleGAN and pseudo-labels (without progressive learning) for domain adaptation. All the results were achieved using AlignedReID++ as backbone.

				CMC Accuracy (in %)		
Src.	**Trg.**	**Method**	**BS.**	**Rank-1**	**Rank-5**	**Rank-10**
Market	**Viper**	CycleGAN	**X**	21.4	**40.2**	**50.3**
			✓	**22.8**	39.1	48.9
		Ours	**X**	**23.7**	41.5	50.8
			✓	21.5	**41.9**	**51.3**
	CUHK03	CycleGAN	**X**	37.0	69.1	80.9
			✓	**38.9**	**69.2**	**81.1**
		Ours	**X**	42.9	72.5	81.2
			✓	**43.1**	**72.7**	**84.2**
CUHK03	**Viper**	CycleGAN	**X**	**21.8**	**43.2**	**52.2**
			✓	17.9	39.9	50.9
		Ours	**X**	**22.5**	**43.2**	**54.1**
			✓	18.5	38.0	50.2
	Market	CycleGAN	**X**	**42.7**	**59.7**	**67.3**
			✓	38.4	57.2	65.5
		Ours	**X**	46.8	65.9	73.6
			✓	**50.1**	**68.2**	**75.6**
Viper	**CUHK03**	CycleGAN	**X**	**17.1**	**41.6**	**55.8**
			✓	14.5	33.5	45.7
		Ours	**X**	**20.4**	43.9	58.5
			✓	17.5	**44.5**	**59.5**
	Market	CycleGAN	**X**	**23.1**	**37.9**	**45.8**
			✓	11.2	22.6	29.2
		Ours	**X**	**28.4**	**46.4**	**55.2**
			✓	27.6	43.9	52.4

the assumption $g \propto 1/B$ is not valid, but $g \propto \varepsilon$ is correct. Therefore, a classical learning rate decay scheduler works better in these cases.

Having this limitation of the Viper dataset in mind, we can focus our analysis on the experiments that did not involve that dataset. However, that still gives a draw of 4 cases in favour and 4 cases against the batch scheduler.

Our results are therefore inconslusive regarding the batch scheduler. We hypothesise that a major factor for that is that we used a GPU with an amount of memory that was too small (8GB) to be effective for this strategy, allowing a maximum batch size of 88 samples.

CycleGAN as intermediate step

To analyse CycleGAN's contribution to our framework, we compared the pseudo-labels results when applied directly with the source domain model against the pseudo-labels results when applied with the model trained on the intermediate dataset. To simplify this experiment, we did not use the progressive learning in the pseudo-labels step. All these experiments were performed using the AlignedReID++ as the backbone and are summarised in Table 4.5.

Table 4.5: CMC accuracy results (in %) using Rank-1, Rank-5 and Rank-10, for the pseudo-labels (without progressive learning) method using the CycleGAN intermediate step or not. All the results were achieved using AlignedReID++ as backbone.

			CMC Accuracy (in %)		
Source	**Target**	CycleGAN	**Rank-1**	**Rank-5**	**Rank-10**
Market	**Viper**	**X**	21.5	38.3	46.5
		✓	**23.7**	**41.5**	**50.8**
	CUHK03	**X**	31.6	58.5	70.5
		✓	**43.1**	**72.7**	**84.2**
CUHK03	**Viper**	**X**	19.5	41.0	70.5
		✓	**22.5**	**43.2**	**54.1**
	Market	**X**	45.7	61.5	68.3
		✓	**50.1**	**68.2**	**75.6**
Viper	**CUHK03**	**X**	18.0	40.8	53.6
		✓	**20.4**	**43.9**	**58.5**
	Market	**X**	23.0	37.6	44.9
		✓	**28.4**	**46.4**	**55.2**

The CycleGAN step creates an intermediate dataset that has the source domain labels and the target domain style, therefore reducing the domain shift between source and target domains. We expect that a model trained on this intermediate dataset outperforms a model trained only on the source domain.

As one can see in Table 4.5 the model pre-trained on the intermediate dataset was able to generate better pseudo-labels than the model pre-trained in the source domain in all cases. Even when using Market1501 as source domain and Viper as target domain, where the CycleGAN achieved worse results than direct transfer, the CycleGAN step was helpful for the framework.

Progressive Learning

The pseudo-labels quality plays a crucial role in the target domain performance, but to avoid negative transfer, it is important that the pseudo-labels be as close as possible to the real labels.

As discussed before, the pseudo-labels generated by a model that has not been updated may be noisy once the model has never seen target domain images and some camera features may mislead the clustering function. However, even with noisy pseudo-labels we are able to improve our model performance in the target domain. We therefore believe that if we create new pseudo-labels using a model that has been fine-tuned with previous pseudo-labels, we will be able to improve the model performance on target domain.

In Table 4.6, we can see that this hypothesis is indeed true, once the use of progressive learning improved the model in all cases, except for those where the Viper dataset was used as target domain. This happened because Viper dataset only has a pair of samples for each person ID, therefore the clustering complexity is too high once there is only one positive sample for each image. In bigger datasets even if we select multiple person IDs within a cluster, there is the possibility of having correct sample pairs to balance that.

In addition, it is important to notice the difference when Viper is used as the source domain. For the case where we have CUHK03 as target domain, we needed 14 progressive learning steps when Viper was used as source domain. On the other hand, only 3 steps were needed for Market1501 as source domain. Also, for Market1501 as target domain, we needed 14 progressive learning steps with Viper as source domain, versus 9 steps when CUHK03 was the source domain. These results show that having fewer images to learn from the source domain does not allow the model to easily learn features that are robust against domain variations.

4.5 Conclusion

In person re-identification, each type of environment (e.g. airport, shopping centre, university campus, etc.) has its own typical appearance, so a system that is trained in one environment is unlikely to perform well in another environment. This observation was confirmed by our cross-dataset (direct transfer) experiments, indicating that each dataset

Table 4.6: CMC accuracy results (in %) using Rank-1, Rank-5 and Rank-10, for the pseudo-labels method using, or not, the progressive learning strategy. The iterations column show how many progressive learning steps were needed to achieve convergence. The rows with value of 1 indicate that no progressive learning was used. All these results were achieved using AlignedReID++ as backbone.

			CMC Accuracy (in %)		
Source	**Target**	**Iterations**	**Rank-1**	**Rank-5**	**Rank-10**
Market	**Viper**	1	**23.7**	**41.5**	**50.8**
		2	18.2	36.9	46.0
	CUHK03	1	43.1	72.7	84.2
		3	**47.8**	**75.9**	**84.2**
CUHK03	**Viper**	1	**22.5**	**43.2**	**54.1**
		2	20.7	40.8	50.6
	Market	1	50.1	68.2	75.6
		9	**64.3**	**81.5**	**87.5**
Viper	**CUHK03**	1	20.4	43.9	58.5
		14	**51.2**	**76.2**	**83.8**
	Market	1	28.4	46.4	55.2
		14	**55.2**	**73.9**	**81.0**

can be treated as a domain. Therefore, we showed that a domain adaptation method based on cycleGAN can be applied to transform the marginal distribution of samples from a source dataset to a target dataset. This enables us to retrain a triplet CNN on adapted samples so that their performance is improved on the target dataset without using a single labeled sample from the target set. Furthermore, we showed that using this CNN and a clustering algorithm to generate pseudo-labels and retrain the triplet CNN leads to a significant performance boost on the target dataset. Finally, we presented an iterative strategy to keep improving the pseudo-labels and retraining the CNN to achieve the best possible performance on target domain. This opens doors for the deployment of person Re-ID software to real applications, as it completely removes the burden of annotating new data.

Further to proposing a domain adaptation technique for this problem, we also presented the use of a batch scheduler which increases the batch size as training starts to converge. However, the hardware limitations and the lack of data in Viper dataset prevented us to perform a deep analysis of this method's effectiveness. In addition, this Chapter proved that our method can be applied with state-of-art person re-identification methods as backbone (AlignedReID++). Also, it was clear that the better the backbone method, the better are the results achieved with our workflow.

Chapter 5

Multi-Step Pseudo-Label Refinement

5.1 Overview

In this Chapter, we dive deep in the UDA Re-ID setup relying only on the pseudo-labels to enhance models performance in target domain. The quality of pseudo-labels clearly is essential for the performance of this kind of method. However, pseudo-labels are expected to be noisy in this scenario. Many methods used soft cost functions to deal with this noise, however we believe that cleaning and improving pseudo-labels is key to achieve high performance. We therefore focus in two main points: camera-based normalisation, which we observed to be key to reduce domain variance; and a novel clusters selection strategy. The latter removes outlying clusters and generate pseudo-labels with important characteristics to help model convergence. This strategy aims to generate clusters which are dense and each contain samples of one person captured from the view of multiple cameras.

Enhancing cluster quality has been overlooked by methods based on pseudo-labels and this has certainly held them back. To evaluate this proposal we work with the most popular cross-domain dataset in unsupervised Re-ID works: Market1501 and DukeMTMC. Our multi-step pseudo-label refinement keeps cleaning and improving the predicted target domain label space to enhance model performance without the burden of annotating data (Figure 5.1). Further, we introduce strategies to build and select clusters in a way that maximises the model's generalisation ability and its potential to transfer learning to new Re-ID datasets where the labels are unknown.

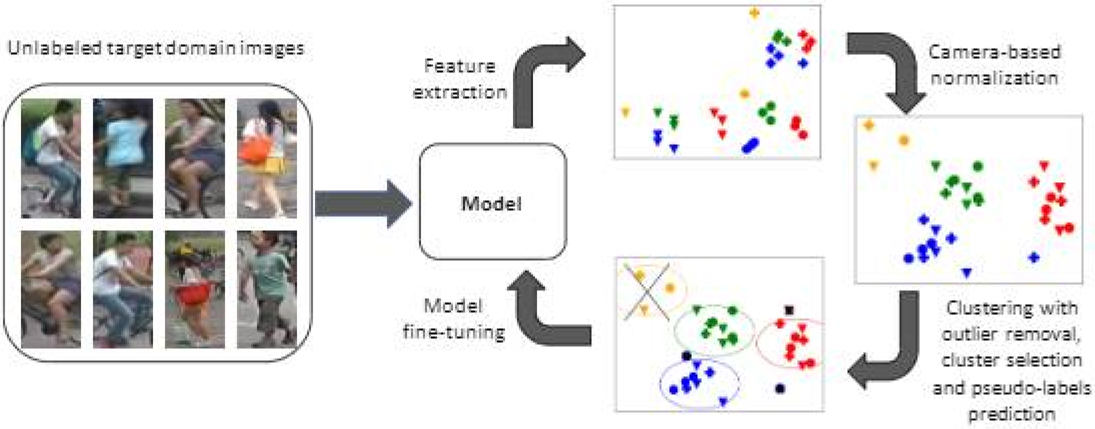

Unlabeled target domain images
Model
Feature extraction
Camera-based normalization
Clustering with outlier removal, cluster selection and pseudo-labels prediction
Model fine-tuning

Figure 5.1: The Multi-Step Pseudo-Label Refinement pipeline. The proposed method consists of four components: extraction of features from unlabelled target domain images, camera-based normalisation, prediction of pseudo-labels with a density-based clustering algorithm, selection of reliable clusters and fine-tuning of the model. The pipeline is cyclical, because at each step it predicts more robust pseudo-labels that offer new information for the model. In the feature space panels, each shape (e.g. triangle, plus signal and circle) represents a camera view and each colour represents a person ID.

5.2 Methodology

5.2.1 Training Protocol

First of all, we train our model in the source domain $\mathcal{D}^s$ as a baseline. All our models use the IBN-Net50-a as backbone and outputs a feature vector $\mathbf{f}$ and a logit prediction vectors $\mathbf{p}$.

Our loss function has three components:

- A batch hard triplet loss ($\mathcal{L}_{Tri}$) [30] that maps $\mathbf{f}$ in an Euclidean vector space;
- A centre loss ($\mathcal{L}_{centre}$) [79] to guarantee cluster compactness;
- A cross entropy label smooth loss ($\mathcal{L}_{ID}$) [90] that uses the logit vectors $\mathbf{p}$ to learn a person ID classifier.

The smoothed person ID component has been proved to help Re-ID systems [51] even though the training IDs are disjoint from the testing IDs. Furthermore, its soft labels has shown interesting features for UDA Re-ID [23]. Our loss function is thus given by Equation 5.1.

$$\mathcal{L} = \mathcal{L}_{Tri} + \mathcal{L}_{ID} + 0.005\mathcal{L}_{centre} \tag{5.1}$$

The weight given to the centre loss is the same that was used in [51].

We start our training with pre-trained weights from ImageNet [15] and use the Adam optimiser for 90 epochs with a warm-up learning rate scheduler defined by Equation 5.2, which is based on [51].

$$lr = \begin{cases} 3.5 \times 10^{-5} \times \frac{epoch}{10} & , epoch \leq 10 \\ 3.5 \times 10^{-4} & , 10 < epoch \leq 40 \\ 3.5 \times 10^{-5} & , 40 < epoch \leq 70 \\ 3.5 \times 10^{-6} & , epoch > 70 \end{cases} \tag{5.2}$$

For data augmentation we use random erase [93], resize images to 256 × 128 and apply a random colour transformation that could be a 20% brightness variation or a 15% contrast variation. We also use random horizontal flipping.

5.2.2 Progressive Learning

As we discussed in Section 4.2.3, the key for good domain adaptation results while working with pseudo-labels is to continuously improve them. Therefore, we adopt the progressive learning strategy, such that in each step the new pseudo-labels get closer of the real labels. However, if the initial model is not good enough, this leads to negative transfer [55] and the performance of the system actually degrades as it iterates. However, since target labels are unknown, it is not possible to predict negative transfer.

For this reason, we argue that progressive learning must be coupled with other techniques, such as the method we describe in the next sections, particularly in §5.2.4 and §5.2.5. In those sections, we propose to evaluate the reliability of samples and their pseudo-labels based on the confidence of the model. If only reliable samples and their pseudo-labels are used, the model should progressively improve and generate more robust pseudo-labels in the consecutive iterations.

5.2.3 Clustering techniques

For standard classification tasks, pseudo-labels generation is direct: it is assumed that the predictions obtained are usually correct and these predictions on the target set are used as pseudo class labels. However, as mentioned earlier due to the lack of control on the number of classes, person Re-ID is usually approached as a metric learning task. The model prediction is therefore not a label, but a feature vector in a space where samples of the same person are expected to lie closer to each other (and further to samples of different people). Therefore, it is necessary to use clustering algorithms and define each cluster as a pseudo-label (or pseudo person ID).

Given a target domain $\mathcal{D}^t$ with $\mathcal{N}$ images $\{\mathbf{x}_i\}_{i=1}^{\mathcal{N}}$ we need to predict their labels $\{y_i\}_{i=1}^{\mathcal{N}}$. We use a model pre-trained on source domain $\mathcal{D}^s$ to extract the features for each image $\{\mathbf{x}_i\}_{i=1}^{\mathcal{N}}$ from $\mathcal{D}^t$ and then use a clustering algorithm to group/predict each image label.

K-means

As a first choice, we used the k-means [27] algorithm to cluster our data. The only parameter k-means needs is the number of clusters k. For our experiments, we choose k using this heuristic:

$$k = \left\lfloor \frac{\mathcal{N}}{15} \right\rfloor \ , \tag{5.3}$$

where $\mathcal{N}$ is the total number of training images in target domain $\mathcal{D}^t$. If all clusters have a balanced number of features (images) this would mean that we are assuming that each person ID in the target domain contains about 15 samples.

There are two problems with k-means for Re-ID: **a)** how to define k without information about $\mathcal{D}^t$ and **b)** as stated by Zeng et al. [83] k-means does not have an outlier detector, so the outliers may drag the centroids away from denser regions, causing the decision boundaries to shift, potentially cutting through sets of samples that actually belong to the same people.

DBSCAN

As discussed above, k-means is not recommended to generate robust pseudo-labels for UDA Re-ID methods. Therefore, we propose the usage of DBSCAN [17] which is a density-based clustering algorithm designed to deal with large and noisy databases.

In a Domain Adaptation Re-ID scenario we can say that the hard samples are actually noise, so a clustering algorithm that identifies them as outliers is fundamental to improve results. Furthermore, when applying Progressive Learning we can leave hard samples out for some iterations and bring then to the pseudo-labelled dataset in later iterations where our model is stronger and the level of confidence in those hard samples is higher.

One important point is that DBSCAN does not require a pre-defined number of clusters (as in k-means), but it requires two parameters: the maximum distance between two samples to determine them as neighbours (ϵ) and the minimum number of samples to consider a region as dense (ω).

In our experiments, we set $\omega = 4$. As for the parameter ϵ, its value depends on the spread of the data. We performed a simple search in an early training step determine a value that would balance the number of clusters selected and the number of outliers. This lead to $\epsilon = 0.35$ when DukeMTMC is the target domain and $\epsilon = 0.42$ when Market1501 is the target domain.

5.2.4 Cluster Selection

Re-ID datasets have disjoint label spaces, that is given a source domain $\mathcal{D}^s$ and a target domain $\mathcal{D}^t$ their labels space do not share the same classes, i.e.

$$\{\mathbf{y}_i\}^s \neq \{\mathbf{y}_j\}^t \ \forall \ i,j. \tag{5.4}$$

Therefore, Re-ID methods typically use triplet loss with batch hard [30] and $\rho\varsigma$ sampling. The $\rho\varsigma$ sampling method consist in selecting ρ identities with ς samples from each identity to form a mini-batch γ in training stage, which leads to the following:

$$\gamma = \rho \times \varsigma. \tag{5.5}$$

In this work we used the triplet loss and $\rho\varsigma$ sampling to train our models, so we expect that every person ID has at least ς images. This clustering step therefore ignores clusters with less than ς images.

An important factor for Re-ID models is to learn features that are robust to camera view variations. For that we guarantee that, in the training stage, our model is fed with samples of the same person ID in different cameras. Therefore, we also prune clusters that had images from only one camera view.

5.2.5 Camera-Guided Feature normalisation

The high variance present in Re-ID datasets is mainly caused by the different camera views, as each view has its own characteristics. This is why a model trained in a source dataset presents poor results when evaluated in a target dataset (or domain). Normally, Re-ID models learn robust features for known views, but lack the ability to generalise for new unseen views.

In Chapter 4 and in [59] we realised that this lack of generalisation power has a negative impact in pseudo-labels generation. We observed that the main reason for that is the fact that, in new unseen cameras, the model tends to cluster images by cameras rather than clustering images from the same person in different views. The majority of clusters would therefore be ignored in the Cluster Selection step.

Zhuang et al. [99] replaced all batch normalisation layers by camera batch normalisation layers. Although this helped them to reduce the data variance between camera views, they normalise the data only on the source domain. We propose to run this camera feature normalisation step before the pseudo-labels step on the target domain training set. By generating pseudo-labels that are normalised by camera information, our method guides the model to learn robust features in the target domain space without the need of changing the model architecture or having additional cost functions.

Camera-guided feature normalisation therefore aims to reduce the target domain variance, enhance the model capacity in the target domain and create better pseudo-labels that further will result in a more robust model.

To apply camera guided feature normalisation, we first divide all target domain training images $\{\mathbf{x}_i\}^t$ in n groups where $V = \{\nu_1, \cdots, \nu_n\}$ are the camera viewpoints in the dataset. Then we extract their features $\mathbf{f}_{\nu_j}$ with our model and calculate, for each camera ν_j, its mean $\boldsymbol{\mu}_{\nu_j}$ and its standard deviation $\boldsymbol{\sigma}_{\nu_j}$ (i.e., these statistics are computed over the activations that they operate in the neural net, in a per camera basis, rather than per batch). Finally, each feature is normalised by

$$\overline{\mathbf{f}}_{\nu_j} = \frac{\mathbf{f}_{\nu_j} - \boldsymbol{\mu}_{\nu_j}}{\boldsymbol{\sigma}_{\nu_j}}. \tag{5.6}$$

The normalised features $\overline{\mathbf{f}}_{\nu_j}$ are then used to generate the pseudo-labels.

5.2.6 Unsupervised Domain Adaptation

For unsupervised domain adaptation, we start with the model pre-trained in $\mathcal{D}^s$ and use it to extract all the features $\mathbf{f}$ from $\mathcal{D}^t$ training images. Once we have all these features extracted, we separate them by camera and use Equation 5.6 to normalise them. Then, we use DBSCAN to create general clusters in $\mathcal{D}^t$ and finally apply our cluster selection strategy of Section §5.2.4 to keep only the clusters which are potentially the the most reliable ones.

From the selected clusters we create our pseudo-labeled dataset and use it to fine-tune our previous model. Since the domains are different datasets, the person IDs on the pseudo-labeled dataset are always different from those of the source dataset. Additionally, as our progressive learning strategy iterates, pseudo-labels are expected to change. Therefore, it is expected that the cross-entropy loss $\mathcal{L}_{ID}$ spikes in first iterations, which can destabilise the training process and lead to catastrophic forgetting. To prevent that, we follow the transfer learning strategy of freezing the body of our model for 20 epochs and let the last fully connected layer learn a good enough $\mathbf{p}$. Then, we unfreeze our model and complete the fine-tuning following the procedure described in 5.2.1.

After the fine-tuning we evaluate our model on $\mathcal{D}^t$ and iterate the whole process, according to the progressive learning strategy.

5.2.7 Post-Processing

Considering the person Re-ID challenge as an image retrieval task, for each query feature $\mathbf{f}_q$ there is a set $N(q,\tau) = \{\mathbf{f}^0_{g1}, \mathbf{f}^0_{g2}, \cdots, \mathbf{f}^0_{g\tau}\}, |N(q,\tau)| = \tau$ containing the nearest gallery features. Our main objective is that the top-τ gallery features are from the same person ID as the probe feature. However, due to variations in illuminations, poses, views or occlusions some gallery features from the same person ID may not be present in the top-τ features.

To tackle this problem, Zhong et al. [92] proposed a re-ranking technique that uses the τ-reciprocal neighbours from a query feature $\mathbf{f}_q$ to calculate its Jaccard distance to a gallery feature $\mathbf{f}_g$. The τ-reciprocal neighbours $\mathcal{R}(q,\tau)$ can be defined by

$$\mathcal{R}(q,\tau) = \{g_i | (g_i \in N(q,\tau) \wedge (q \in N(g_i,\tau))\}. \tag{5.7}$$

These τ-reciprocal nearest neighbours are more related to the query feature $\mathbf{f}_q$ than the τ-nearest neighbours. However, they propose an even more robust reciprocal neighbours set $\mathcal{R}^*(q,\tau)$ by incrementally adding the $\frac{1}{2}\tau$-reciprocal nearest neighbours of each candidate in $\mathcal{R}(q,\tau)$ into this new set, according to the following condition

$$\begin{aligned} &\mathcal{R}^*(q,\tau) \leftarrow \mathcal{R}(q,\tau) \cup \mathcal{R}(p,\tfrac{1}{2}\tau) \\ &s.t.\ |\mathcal{R}(q,\tau) \cap \mathcal{R}(p,\tfrac{1}{2}\tau)| \geq \tfrac{2}{3}|\mathcal{R}(p,\tfrac{1}{2}\tau)|, \\ &\forall p \in \mathcal{R}(q,\tau) \end{aligned} \tag{5.8}$$

Finally, the Jaccard distance from a query feature $\mathbf{f}_q$ and a gallery feature $\mathbf{f}_g$ can be calculated as

$$D_J(q,g) = 1 - \frac{|\mathcal{R}^*(q,\tau) \cap \mathcal{R}^*(g_i,\tau)|}{|\mathcal{R}^*(q,\tau) \cup \mathcal{R}^*(g_i,\tau)|}, \tag{5.9}$$

and the final distance between these features is measured by jointly aggregating the Jaccard distance with the Euclidean distance as

$$\mathcal{D}^*(q,g) = (1-\lambda)\mathcal{D}_J(q,g) + \lambda\mathcal{D}(q,g). \tag{5.10}$$

5.3 Results

In this Section we present the experimental results for our proposed method. Firstly, in 5.3.1 we compare our results with supervised methods, then in 5.3.2 we compare our method against other state-of-the-art methods. Finnaly, in 5.3.3 we perform some ablation studies.

Table 5.1: Comparison of our results with results using supervised learning on the target domain, using supervised learning on both source and target domain at the same time (which is expected to give the best results) and direct transfer results, i.e. the use of a model trained on source directly applied to the target domain, without domain adaptation (which is expected to be a lower bound).

	Supervised	**Market1501 → DukeMTMC**				**DukeMTMC → Market1501**			
	Training	Rank-1	Rank-5	Rank-10	mAP	Rank-1	Rank-5	Rank-10	mAP
1	Source	44.7	60.7	66.4	27.3	58.9	74.3	80.1	29.0
2	Target	82.7	92.1	94.6	68.6	92.5	97.6	98.7	81.5
3	Source and Target	83.9	92.5	94.8	71.1	92.6	97.7	98.6	81.2
4	Source (**Ours**)	82.7	90.5	93.5	69.3	89.1	95.8	97.2	73.6

5.3.1 Comparison with Baseline and Upper Bound

In table 5.1, row 1 is a naive domain adaptation strategy which is expected to be a baseline, as the model trained on the source domain is directly applied to the target domain with no adaptation - this is also known as "direct transfer"; row 2 is the traditional single domain supervised learning scenario, where training samples are available in the application domain; row 3 is expected to be an uppper bound in performance; row 4 is our method, which only uses labeled samples in the source domain but benefits from unlabeled target samples.

In method 3 of Table 5.1, we evaluate our base method (without transfer) in a multi-source domain scenario. For this experiment, we merged the training samples and labels from Market1501 and DukeMTMC. Then, we used this multi-source domain to train our baseline model and further evaluated the results in both domains separated. This experiment is set to be a better upper bound for our study, because when we apply our UDA method, we expose our model to training images from both domains (even if it has no labels from one of them), therefore we believe that it would be fair to compare the results with a supervised method that also had all these information available.

The Direct transfer method (method 1 of Table 5.1) is used to evaluate the domain shift and the model generalisation power. It is expected that this setting gives results that are worse than the domain adaptation setting, because no knowledge of the target set is used in the training process. Our method does not focus on being generalisable, we aim to use the source domain knowledge as start and enhance the model's performance in target domain without any labels. We found it important to present direct transfer results in order to show how much our method enhances over it.

As one can see, our method reaches remarkable results for DukeMTMC as a target dataset. It can be surprising to see that we matched the supervised result for CMC rank-1 and even surpasses it in 0.5% for mAP. DukeMTMC is a dataset with a high intra-variance caused by its eight distinct camera views. We believe that the camera-guided normalisation applied before the clustering step provided pseudo-labels that were more robust to camera view variations. Therefore, the method was able to learn camera invariant features. It is also likely that by transferring from one dataset to another, our method was less prone to over-fitting than the supervised learning setting.

For Market1501 as a target, our method performed equally well enhancing the direct transfer result in 30.2% and 44.6% for CMC rank-1 and mAP, respectively. However, with lower intra-variance in Market1501 the supervised result is already saturated. Therefore, even tough labels from the target set were not used, our methods gives results which are not far below those of the supervised setting.

The multi-source domain setting (method 3 of Table 5.1) shows some improvements against the baseline supervised result. As this setup used samples from both datasets, we believe that it is a better upper bound for our UDA method. Although these results are better than ours, it is important to remind that in our method, nearly half of the samples are unlabelled.

5.3.2 Comparison with state-of-art UDA results

Table 5.2: Comparison of our results with state-of-art methods in UDA. We highlighted in bold, underline and italic the first, second and third best results, respectively. RR stands for Re-Ranking.

Methods	**Market1501 → DukeMTMC**				**DukeMTMC → Market1501**			
	Rank-1	Rank-5	Rank-10	mAP	Rank-1	Rank-5	Rank-10	mAP
SPGAN [14]	46.9	62.6	68.5	26.4	58.1	76.0	82.7	26.9
UCDA-CCE [61]	55.4	-	-	36.7	64.3	-	-	34.5
ARN [44]	60.2	73.9	79.5	33.4	70.3	80.4	86.3	39.4
MAR [81]	67.1	79.8	-	48.0	67.7	81.9	-	40.0
ECN [94]	63.3	75.8	80.4	40.4	75.1	87.6	91.6	43.0
PDA-Net [43]	63.2	77.0	82.5	45.1	75.2	86.3	90.2	47.6
EANet [32]	67.7	-	-	48.0	78.0	-	-	51.6
CBN [99] + ECN	68.0	80.0	83.9	44.9	81.7	91.9	94.7	52.0
Theory [70]	68.4	80.1	83.5	49.0	75.8	89.5	93.2	53.7
CR-GAN [9]	68.9	80.2	84.7	48.6	77.7	89.7	92.7	54.0
PCB-PAST [87]	72.4	-	-	54.3	78.4	-	-	54.6
AD Cluster [84]	72.6	82.5	85.5	54.1	86.7	94.4	96.5	68.3
SSG [21]	76.0	85.8	89.3	60.3	86.2	94.6	96.5	68.7
DG-Net++ [100]	78.9	87.8	90.4	63.8	82.1	90.2	92.7	61.7
MMT [23]	*79.3*	*89.1*	*92.4*	*65.7*	90.9	**96.4**	**97.9**	76.5
Ours	82.7	90.5	**93.5**	69.3	*89.1*	95.8	97.2	*73.6*
Ours + RR [92]	**84.8**	**90.8**	93.2	**81.2**	**92.0**	*95.3*	*96.6*	**88.1**

In Table 5.2 we compare our multi-step pseudo-label refinement method with multiple state-of-the-art Re-ID UDA methods. As one can see, we beat all other methods in DukeMTMC target dataset and push the state-of-the-art by 3.4% and 3.6% for CMC rank-1 and mAP, respectively. For Market1501 we are able to reach second place with a noticeable gap to the third place with an improvement of 2.4% and 4.9% for CMC rank-1 and mAP, respectively.

In addition, our framework has a lightweight architecture when compared to other frameworks that achieve state-of-the-art. MMT [23] uses two CNNs so that one generates soft labels for the other. DG-Net++ [100] uses a extremely complex framework with GANs and multiple encoders and decoders.

As we approach Re-ID as a metric learning task, re-ranking algorithms have a great impact in the results. We therefore evaluated our model using k-reciprocal encoding re-ranking [92] which combines the original distance with the Jaccard distance in an unsupervised manner. The importance to use a ranking system is shown by CMC Rank-1 improvement of 2.1% on DukeMTMC and 2.9% on Market1501 when compared to our raw method. Furthermore, re-ranking significantly pushes the mAP performance in 11.9% for DukeMTMC and 14.5% for Market1501.

5.3.3 Ablation study

Table 5.3: The contribution of each method in the model performance evaluated on Market1501 and DukeMTMC-reID datasets. CN stands for Camera Guided normalisation.

Methods	**Market1501 → DukeMTMC**		**DukeMTMC → Market1501**	
	Rank-1	mAP	Rank-1	mAP
ResNet 50 [51]	41.4	25.7	54.3	25.5
+ IBN-Net50-a	44.7	27.3	58.9	29.0
+ Domain Adaptation	52.2	37.1	60.1	34.8
+ Progressive Learning	52.2	37.1	61.4	35.5
+ Cluster Selection	77.2	61.8	86.5	66.0
+ CN	82.7	69.3	89.1	73.6

Table 5.3 shows how each technique contributes to our final method performance.

IBN-Net50-a: the difference between the original Resnet-50 and the IBN-Net50-a is that the IBN-Net50-a modifies all batch normalisation layers so they also take advantage of instance normalisation. This modification enhances the model's generalisation power, leading to an improvement on the CMC rank-1 performance improvement of 3.3% in DukeMTMC and 4.6% in Market1501.

Domain Adaptation: in Table 5.3 we call domain adaptation the use of pseudo-labels from target domain for training. This clustering-guided DA method allows our model to train using actual images from target domain, which helps the model to learn various aspects of the domain, such as illumination, camera angles, person pose. Learning the characteristics from target domain is a major factor for domain adaptation which becomes evident by the CMC rank-1 improvement of 7.5% in DukeMTMC and 1.2% in Market1501.

Progressive Learning: this technique has a great potential to keep improving the model's performance with new pseudo-labels. However, as we said in Section 5.2.2 to get full advantage of this technique one needs to guarantee that the pseudo-labels are close to the class divisions. Therefore, this step only gives a significant improvement if associated with the proposed cluster selection technique. In Table 5.3, the progressive learning results were obtained using the raw clusters defined by the clustering algorithm. That model used all the available information in target domain and overfitted to these pseudo-labels. In the next step these clusters tend to be the same and the model does not have a stimulus to learn better features. This is why the progressive learning results on their own do not seem to help for Market1501 → DukeMTMC.

Cluster Selection: this method relies on a continuous improvement on the pseudo-labels. For that, we remove clusters that are unlikely to help improve the model, such as small clusters with less than 4 images and clusters that had images from only one camera view. Using this strategy we can get full advantage of progressive learning and push the model to learn camera view invariant features, since all our pseudo-labels have samples from multiple camera views.

The real contribution of the progressive learning technique is shown alongside the contribution of the cluster selection strategy, because they are complementary techniques. This is certainly the most relevant element of our pipeline, as it leads to a step change in the performance, enhancing the rank-1 CMC performance by 25.0% for DukeMTMC and 25.1% for Market1501.

Camera Guided Normalisation: learning camera invariant features is essential for person Re-ID, because the person appearance may vary on different cameras. Since target labels are unknown, when the model extracts features from the target domain, instead of grouping images by the person that appears in them, the feature vectors tend to cluster camera viewpoints. The camera guided normalisation helps to reduce this camera shift and align the features from different cameras. Our cluster selection method thus selects more clusters to be part of the pseudo-label dataset (this can be seen in Figures 5.2d, 5.2e, 5.3d and 5.3e). With this richer and camera invariant pseudo-label dataset, our model has better samples to learn from and its mAP is improved by 7.5% for DukeMTMC and 7.6% for Market1501.

Also, Figures 5.2c and 5.3c allow us to see that using the camera guided normalisation significantly enhances the completeness of our pseudo-labels. This happens because when the normalisation is applied, it facilitates the clustering method to group images from the same person in different camera views.

Although the camera normalisation step sometimes does not imply in better homogeneity and thus V-Measure scores, this is not a problem. As we stated before, when using the normalisation step, a higher percentage of available images are chosen and the total number of clusters slightly increases (DBSCAN only as *k*-means have a fixed number of clusters). Therefore, without normalisation we end up generating smaller clusters with just a few samples from one person, while the methods using normalisation are able to produce bigger (richer) clusters with almost all samples from a person as shown for the completeness graph.

The more populated clusters combined with great completeness score and average homogeneity score are probably clustering images from people with similar appearance. As the model is trying to learn how to group images from the same people, a cluster with multiple people with similar appearance is less harmful than having multiple clusters with images from the same person (which is the case of the methods without normalisation that show high homogeneity and medium completeness). In addition, as the test and training sets are disjoint, learning to cluster images by strong similarities at the person level may be more adequate to train a generalisable model than learning to identify specific identities that will not be present in the test set.

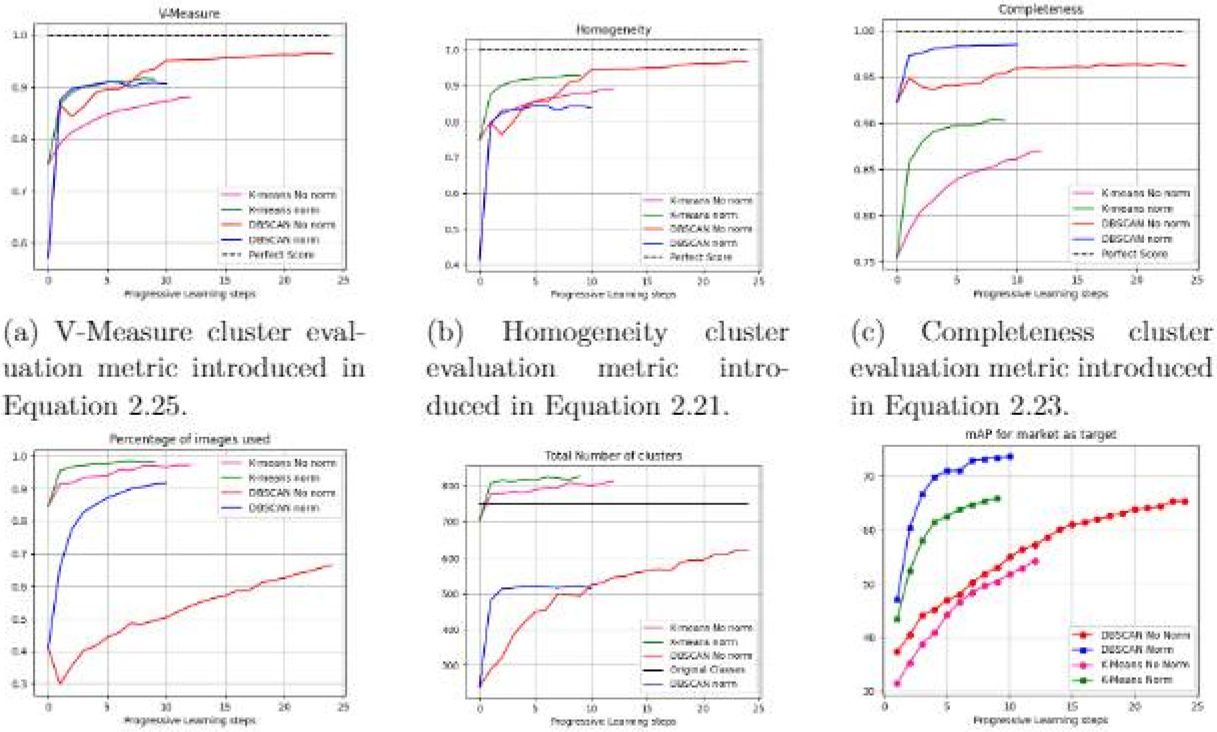
Training efficiency: the better pseudo-labels which are obtained when applying camera guided normalisation speeds up the model convergence, independently of the clustering method used. Figures 5.2f and 5.3f show how many progressive learning steps were needed to reach convergence with or without camera guided normalisation.

Table 5.4: Comparison between DBSCAN and k-means as the clustering algorithm. After cluster selection, different quantities of samples were kept for each clustering method. This portion is shown in the last columns.

Method	**Market1501 → DukeMTMC**				
	Rank-1	Rank-5	Rank-10	mAP	Portion (in %)
k-means	77.7	87.5	90.8	63.1	85.5
DBSCAN	82.7	90.5	93.5	69.3	69.7

Method	**DukeMTMC → Market1501**				
k-means	87.0	94.7	96.9	65.9	95.9
DBSCAN	89.1	95.8	97.2	73.6	79.9

Clustering methods: we ran our multi-step pseudo-label refinement method with two different clustering algorithms in its pipeline: k-means and DBSCAN. Table 5.4 presents the results achieved using each of them and the portion of training data that was selected for use as pseudo-labels after the cluster selection phases. This is shown in Figures 5.2d and 5.3d, which show that amount for each progressive learning step. DBSCAN does not need a fixed number of clusters and has a built-in outlier detector, so it can deal with outliers better than k-means. For k-means, all samples count, then the outliers have a negative impact in the quality of the pseudo-labels. The results in Table 5.4 confirm our hypothesis that it is better to use fewer and less noisy samples, therefore DBCAN is better suited for us.



(a) V-Measure cluster evaluation metric introduced in Equation 2.25.

(b) Homogeneity cluster evaluation metric introduced in Equation 2.21.

(c) Completeness cluster evaluation metric introduced in Equation 2.23.

(d) Percentage of images used at each progressive learning step.

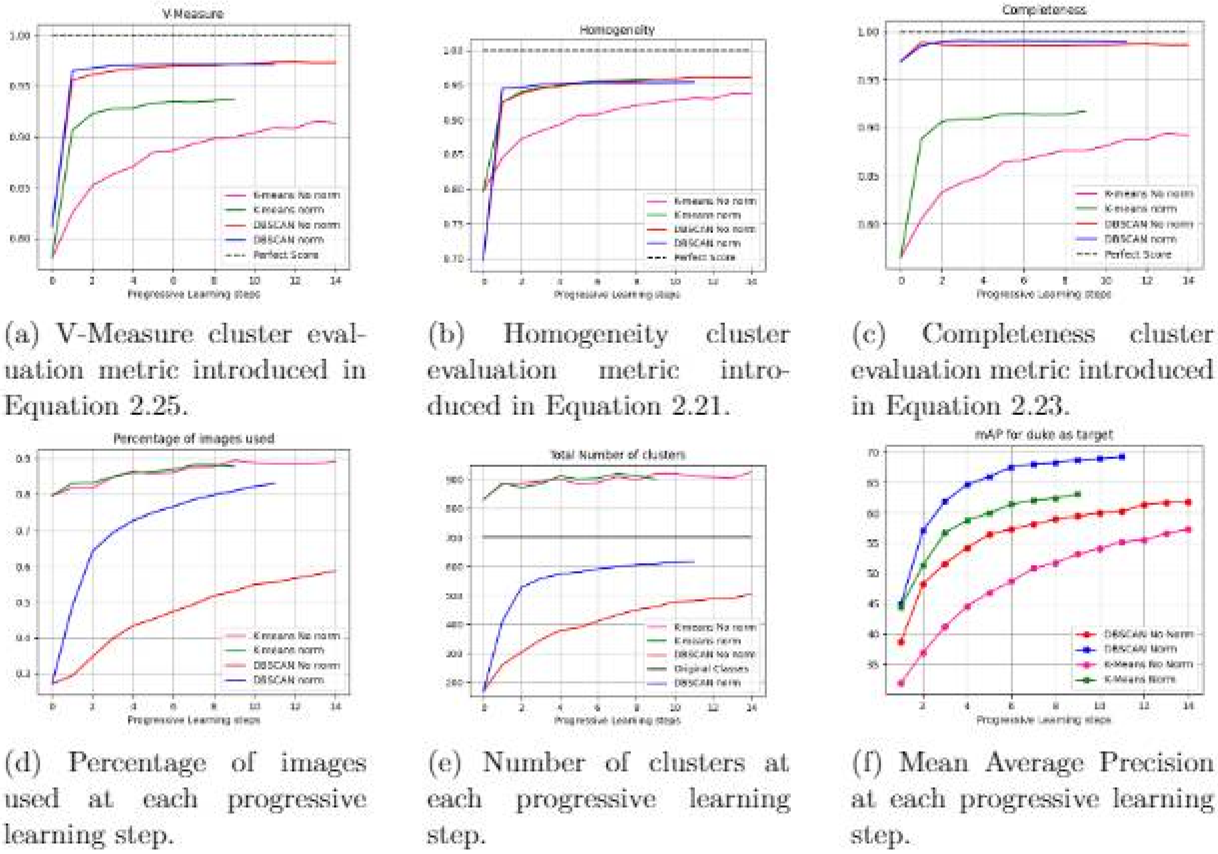
(e) Number of clusters at each progressive learning step.

(f) Mean Average Precision at each progressive learning step.

Figure 5.2: General cluster evaluation metrics for Market1501 dataset as target domain. The curves are plotted up to their convergence point. These metrics helps us to understand why the k-means and DBSCAN differ and how the camera guided normalisation help the pseudo label generation.

5.4 Conclusions

In this Chapter we propose a multi-step pseudo-label refinement method to improve results on Unsupervised Domain Adaptation for Person Re-Identification. We focus on tackling the problem of having noisy pseudo-labels in this task and proposed a pipeline that reduces the shift caused by camera changes as well as techniques for outlier removal and



(a) V-Measure cluster evaluation metric introduced in Equation 2.25.

(b) Homogeneity cluster evaluation metric introduced in Equation 2.21.

(c) Completeness cluster evaluation metric introduced in Equation 2.23.

(d) Percentage of images used at each progressive learning step.

(e) Number of clusters at each progressive learning step.

(f) Mean Average Precision at each progressive learning step.

Figure 5.3: General cluster evaluation metrics for DukeMTMC dataset as target domain. The curves are plotted up to their convergence point. These metrics helps us to understand why the k-means and DBSCAN differ and how the camera guided normalisation help the pseudo label generation.

cluster selection. Our method includes DBSCAN clustering algorithm that was designed to perform well in large and noisy databases; a camera-guided normalisation step to align features from multiple camera views and allow samples from different cameras to be included in the same clusters; and a smart cluster selection method that improves pseudo-labels for our training setup. These steps are iterated until convergence giving better results.

Our method generates a strong label space for target domain without any supervision. We reach state-of-the-art performance on Market1501 as a target dataset and push the state-of-the-art on the challenging DukeMTMC target dataset by 5.5% (or 3.4% without re-ranking). Our work highlights the importance of pseudo-labels refinement with strong normalisation techniques. We evaluated the impact of two different clusterring techniques for pseudo-label generation and concluded that DBSCAN is better suited for this task, as it has a built-in outlier detecton and do not specify the number of clusters. It also takes advantage of a metric learning process and re-ranking [92, 97].

Chapter 6

Conclusion and Further Work

6.1 Final Considerations

In this work, we focused in the real world person Re-ID challenge and could have followed two paths: **a)** Creating a generalisable model capable of achieving a good performance in any dataset or **b)** Using domain adaptation techniques to enhance our (or any) model performance in new domains without the need of further supervision.

A generalisable model that perform the same way despite the domain would be the person Re-ID holy grail, once it could be directly deployed in any scenario. However, this is still far from happening. We are able to affirm that, because even face recognition which nowadays appears to be a less challenging problem, still has not been solved in a generalised way as we can see diverse scandals because methods do not perform the same way for people of different ethnicity.

Therefore, we focused our efforts on a more targeted and realistic path that is the domain adaptation. Even though the domain adaptation does not allow the direct deployment of a model to a new scenario, it can automate that process without the cost of labelling samples. We have in fact shown that using our proposed domain adaptation pipeline, the model performance is comparable to in-domain training strategies where the target domain labels are available.

With the presented results, we believe that our main goal was achieved, as we could propose a person Re-ID framework that starts from a publicly available dataset and performs well in a new dataset without the need of annotated data. This framework can be used in the real world to adapt existent person Re-ID models into new domains in a scalable manner, unlocking this feature to become a product that can be deployed in diverse real applications.

More specifically, we have shown that

- we are able to learn from a new set of images without the need of labels;

- using the camera information allow us to better align the features in a new domain;
- the more samples is not always the merrier: *k*-means does not remove outliers so it uses more images than DBSCAN, but the latter usually performs better; it is better to use more reliable pseudo-labels.

6.2 Future Works

In this work, we proposed two domain adaptation frameworks for the person Re-ID challenge that are able to improve the model's performance in a new domain without the burden of annotating new data. Our Multi-Step Pseudo-Label Refinement method presented in chapter 5 pushed the state-of-the-art and achieved a performance comparable with the supervised models trained in the target domain.

Although our results are very satisfactory, there is still room for improvement in this area. As we saw in Chapter 4, not all domain combinations allow us to achieve remarkable results using domain adaptation, it must be because some domains are too distinct. Therefore, relying on a source domain to train a first supervised model and then adapting it to a new domain may not be the best move always. Also, in Chapters 4 and 5 we saw that the most significant improvement in the pseudo-labels method was because it uses actual target domain images. We therefore believe that using some self-supervised methods (e.g. contrastive unsupervised learning [7, 8, 28]) to warm up the model and then using the warmed model as a starting point to our Chapter 5 method may lead to superior results.

In addition, our main research focus was on the domain adaptation techniques, although the neural network architecture is also crucial to achieve great results. Since the introduction of Vision Transforms (ViTs) [16], transformers have been used to solve several computer vision tasks and have evolved to hierarchical Transformers (e.g. Swin Transformers [48]) that better suits a wide variety of vision tasks. Also, ConvNets are evolving to keep up with the Transformers and new architectures have shown remarkable ImageNet results (e.g. ConvNeXt [49]). We therefore believe that this work could benefit from the use of newer and better neural network architectures.

References

[1] An, H., Hu, H.M., Guo, Y., Zhou, Q., and Li, B.: *Hierarchical Reasoning Network for Pedestrian Attribute Recognition*. IEEE Transactions on Multimedia, 23:268–280, 2021.

[2] Bai, Y., Jiao, J., Ce, W., Liu, J., Lou, Y., Feng, X., and Duan, L.Y.: *Person30K: A Dual-Meta Generalization Network for Person Re-Identification*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2123–2132, June 2021.

[3] Boer, P.T. de, Kroese, D.P., Mannor, S., and Rubinstein, R.Y.: *A tutorial on the cross-entropy method*. Annals of operations research, 134, 2004.

[4] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R.: *Signature Verification using a "Siamese" Time Delay Neural Network*. In Cowan, J., Tesauro, G., and Alspector, J. (eds.): *Advances in Neural Information Processing Systems*, vol. 6. Morgan-Kaufmann, 1994.

[5] Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., and Sheikh, Y.: *OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(1):172–186, 2021.

[6] Chang, X., Hospedales, T.M., and Xiang, T.: *Multi-Level Factorisation Net for Person Re-Identification*. In *Proc of the IEEE Conf on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[7] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G.: *A Simple Framework for Contrastive Learning of Visual Representations*. In III, H.D. and Singh, A. (eds.): *Proceedings of the 37th International Conference on Machine Learning*, vol. 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020. `https://proceedings.mlr.press/v119/chen20j.html`.

[8] Chen, X. and He, K.: *Exploring Simple Siamese Representation Learning*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15750–15758, June 2021.

[9] Chen, Y., Zhu, X., and Gong, S.: *Instance-Guided Context Rendering for Cross-Domain Person Re-Identification*. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[10] Cheng, D., Gong, Y., Zhou, S., Wang, J., and Zheng, N.: *Person Re-Identification by Multi-Channel Parts-Based CNN With Improved Triplet Loss Function.* In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] Csurka, G.: *A Comprehensive Survey on Domain Adaptation for Visual Applications.* In Csurka, G. (ed.): *Domain Adaptation in Computer Vision Applications*, pp. 1–35. Springer International Publishing, Cham, 2017, ISBN 978-3-319-58347-1.

[12] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., and Fei-Fei, L.: *ImageNet: A Large-Scale Hierarchical Image Database.* In *CVPR09*, 2009.

[13] Deng, J., Guo, J., Niannan, X., and Zafeiriou, S.: *ArcFace: Additive Angular Margin Loss for Deep Face Recognition.* In *CVPR*, 2019.

[14] Deng, W., Zheng, L., Ye, Q., Kang, G., Yang, Y., and Jiao, J.: *Image-Image Domain Adaptation With Preserved Self-Similarity and Domain-Dissimilarity for Person Re-Identification.* In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[15] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T.: *DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition.* In Xing, E.P. and Jebara, T. (eds.): *Proceedings of the 31st International Conference on Machine Learning*, Bejing, China, 2014. PMLR. `http://proceedings.mlr.press/v32/donahue14.html`.

[16] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N.: *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.* ICLR, 2021.

[17] Ester, M., Kriegel, H.P., Sander, J., and Xu, X.: *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.* In *KDD*, pp. 226–231, 1996. `http://www.aaai.org/Library/KDD/1996/kdd96-037.php`.

[18] Fan, H., Zheng, L., Yan, C., and Yang, Y.: *Unsupervised Person Re-identification: Clustering and Fine-tuning.* ACM Transactions on Multimedia Computing, Communications, and Applications TOMM, 14(4):83:1–83:18, 2018.

[19] FarajiDavar, N., de Campos, T., and Kittler, J.: *Adaptive Transductive Transfer Machines: A Pipeline for Unsupervised Domain Adaptation.* In *Domain Adaptation in Computer Vision Applications*, Advances in Computer Vision and Pattern Recognition, pp. 115–132. Springer International, 2017. DOI:10.1007/978-3-319-58347-1_6.

[20] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., and Ramanan, D.: *Object Detection with Discriminatively Trained Part-Based Models.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 32(9):1627–1645, 2010.

[21] Fu, Y., Wei, Y., Wang, G., Zhou, Y., Shi, H., and Huang, T.S.: *Self-Similarity Grouping: A Simple Unsupervised Cross Domain Adaptation Approach for Person Re-Identification.* In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[22] Fukushima, K. and Miyake, S.: *Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition.* In Amari, S.i. and Arbib, M.A. (eds.): *Competition and Cooperation in Neural Nets*, pp. 267–285, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg, ISBN 978-3-642-46466-9.

[23] Ge, Y., Chen, D., and Li, H.: *Mutual Mean-Teaching: Pseudo Label Refinery for Unsupervised Domain Adaptation on Person Re-identification.* In *International Conference on Learning Representations*, 2020. `https://openreview.net/forum?id=rJlnOhVYPS`.

[24] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y.: *Generative Adversarial Nets.* In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., and Weinberger, K.Q. (eds.): *Advances in Neural Information Processing Systems (NIPS) 27*, pp. 2672–2680. Curran Associates, Inc., 2014. `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

[25] Gray, D., Brennan, S., and Tao, H.: *Evaluating appearance models for recognition, reacquisition, and tracking.* In *In IEEE International Workshop on Performance Evaluation for Tracking and Surveillance, Rio de Janeiro*, 2007.

[26] Hadsell, R., Chopra, S., and LeCun, Y.: *Dimensionality Reduction by Learning an Invariant Mapping.* In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1735–1742, 2006.

[27] Hartigan, J.A. and Wong, M.A.: *Algorithm AS 136: A K-Means Clustering Algorithm.* Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1):100–108, 1979. `http://www.jstor.org/stable/2346830`.

[28] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R.: *Momentum Contrast for Unsupervised Visual Representation Learning.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[29] He, K., Zhang, X., Ren, S., and Sun, J.: *Deep Residual Learning for Image Recognition.* In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[30] Hermans, A., Beyer, L., and Leibe, B.: *In Defense of the Triplet Loss for Person Re-Identification.* Tech. Rep. arXiv:1703.07737, Cornell University Library, 2017. http://arxiv.org/abs/1703.07737.

[31] Hirzer, M., Beleznai, C., Roth, P.M., and Bischof, H.: *Person Re-identification by Descriptive and Discriminative Classification.* In Heyden, A. and Kahl, F. (eds.): *Image Analysis*, pp. 91–102, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg, ISBN 978-3-642-21227-7.

[32] Huang, H., Yang, W., Chen, X., Zhao, X., Huang, K., Lin, J., Huang, G., and Du, D.: *EANet: Enhancing Alignment for Cross-Domain Person Re-identification.* CoRR, abs/1812.11369, 2018. `http://arxiv.org/abs/1812.11369`.

[33] Isola, P., Zhu, J.Y., Zhou, T., and Efros, A.A.: *Image-To-Image Translation With Conditional Adversarial Networks.* In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T.: *Analyzing and Improving the Image Quality of StyleGAN.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[35] Krizhevsky, A., Sutskever, I., and Hinton, G.E.: *ImageNet Classification with Deep Convolutional Neural Networks.* In Pereira, F., Burges, C.J.C., Bottou, L., and Weinberger, K.Q. (eds.): *Advances in Neural Information Processing Systems*, vol. 25. Curran Associates, Inc., 2012.

[36] Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., and Matas, J.: *DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[37] Lecun, Y.: *Generalization and network design strategies.* Elsevier, 1989.

[38] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P.: *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 86(11):2278–2324, 1998.

[39] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W.: *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[40] Li, D., Chen, X., Zhang, Z., and Huang, K.: *Learning Deep Context-Aware Features over Body and Latent Parts for Person Re-identification.* In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7398–7407, July 2017.

[41] Li, W. and Wang, X.: *Locally Aligned Feature Transforms across Views.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[42] Li, W., Zhao, R., Xiao, T., and Wang, X.: *DeepReID: Deep Filter Pairing Neural Network for Person Re-identification.* In *Proc of the IEEE Conf on Computer Vision and Pattern Recognition (CVPR)*, pp. 152–159, June 2014.

[43] Li, Y.J., Lin, C.S., Lin, Y.B., and Wang, Y.C.F.: *Cross-Dataset Person Re-Identification via Unsupervised Pose Disentanglement and Adaptation.* In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[44] Li, Y.J., Yang, F.E., Liu, Y.C., Yeh, Y.Y., Du, X., and Frank Wang, Y.C.: *Adaptation and Re-Identification Network: An Unsupervised Deep Transfer Learning Approach to Person Re-Identification.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[45] Lin, Y., Xie, L., Wu, Y., Yan, C., and Tian, Q.: *Unsupervised Person Re-Identification via Softened Similarity Learning.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[46] Liu, J., Ni, B., Yan, Y., Zhou, P., Cheng, S., and Hu, J.: *Pose Transferrable Person Re-Identification.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[47] Liu, J., Zha, Z.J., Chen, D., Hong, R., and Wang, M.: *Adaptive Transfer Network for Cross-Domain Person Re-Identification.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[48] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B.: *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows.* CoRR, abs/2103.14030, 2021. `https://arxiv.org/abs/2103.14030`.

[49] Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., and Xie, S.: *A ConvNet for the 2020s*, 2022.

[50] Long, M., Wang, J., Ding, G., Sun, J., and Yu, P.S.: *Transfer Feature Learning with Joint Distribution Adaptation.* In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[51] Luo, H., Gu, Y., Liao, X., Lai, S., and Jiang, W.: *Bag of Tricks and a Strong Baseline for Deep Person Re-Identification.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[52] Luo, H., Jiang, W., Fan, X., and Zhang, C.: *STNReID: Deep Convolutional Networks With Pairwise Spatial Transformer Networks for Partial Person Re-Identification.* IEEE Transactions on Multimedia, 22(11):2905–2913, 2020.

[53] Luo, H., Jiang, W., Zhang, X., Fan, X., Qian, J., and Zhang, C.: *AlignedReID++: Dynamically matching local information for person re-identification.* Pattern Recognition, 94:53–61, 2019.

[54] Miao, J., Wu, Y., Liu, P., Ding, Y., and Yang, Y.: *Pose-Guided Feature Alignment for Occluded Person Re-Identification.* In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[55] Pan, S.J. and Yang, Q.: *A Survey on Transfer Learning.* IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, Oct 2010, ISSN 1041-4347.

[56] Pan, X., Luo, P., Shi, J., and Tang, X.: *Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net.* In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[57] Park, T., Efros, A.A., Zhang, R., and Zhu, J.Y.: *Contrastive learning for unpaired image-to-image translation*. In *European Conference on Computer Vision (ECCV)*, pp. 319–345. Springer, 2020.

[58] Pereira, T. and de Campos, T.E.: *Domain Adaptation for Person Re-Identification with Part Alignment and Progressive Pseudo-Labeling*. International Journal of Pattern Recognition and Artificial Intelligence, 0(0):2160014, 0. `https://doi.org/10.1142/S0218001421600144`.

[59] Pereira, T. and de Campos, T.E.: *Domain Adaptation for Person Re-identification on New Unlabeled Data*. In 15^{th} *International Conference on Computer Vision Theory and Applications (VISAPP) - part of VISIGRAPP*, vol. 4: VISAPP, pp. 695–703, February 27-29 2020.

[60] Pereira, T. and de Campos, T.E.: *Learn by Guessing: Multi-step Pseudo-label Refinement for Person Re-Identification*. In 17^{th} *International Conference on Computer Vision Theory and Applications (VISAPP) - part of VISIGRAPP*, February 6-8 2022.

[61] Qi, L., Wang, L., Huo, J., Zhou, L., Shi, Y., and Gao, Y.: *A Novel Unsupervised Camera-Aware Domain Adaptation Framework for Person Re-Identification*. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[62] Qian, X., Fu, Y., Xiang, T., Wang, W., Qiu, J., Wu, Y., Jiang, Y.G., and Xue, X.: *Pose-Normalized Image Generation for Person Re-identification*. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[63] Ren, S., He, K., Girshick, R., and Sun, J.: *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. In Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., and Garnett, R. (eds.): *Advances in Neural Information Processing Systems (NIPS) 28*, pp. 91–99. Curran Associates, Inc., 2015.

[64] Ristani, E., Solera, F., Zou, R., Cucchiara, R., and Tomasi, C.: *Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking*. In *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016.

[65] Ronneberger, O., Fischer, P., and Brox, T.: *U-Net: Convolutional Networks for Biomedical Image Segmentation*. In Navab, N., Hornegger, J., Wells, W.M., and Frangi, A.F. (eds.): *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing, ISBN 978-3-319-24574-4.

[66] Rosenberg, A. and Hirschberg, J.: *V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure*. In *EMNLP-CoNLL*, pp. 410–420, 2007. `http://www.aclweb.org/anthology/D07-1043`.

[67] Rosenblatt, F.: *The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain*. Psychological Review, pp. 65–386, 1958.

[68] Rosenblatt, F.: *Principles of Neurodynamics.* Perceptrons and the Theory of Brain Mechanisms, 1962. https://ci.nii.ac.jp/naid/10004317109/en/.

[69] Smith, S.L., Kindermans, P.J., and Le, Q.V.: *Don't Decay the Learning Rate, Increase the Batch Size.* In *International Conference on Learning Representations*, 2018. https://openreview.net/forum?id=B1Yy1BxCZ.

[70] Song, L., Wang, C., Zhang, L., Du, B., Zhang, Q., Huang, C., and Wang, X.: *Unsupervised domain adaptive re-identification: Theory and practice.* Pattern Recognition, 102:107173, 2020, ISSN 0031-3203. http://www.sciencedirect.com/science/article/pii/S003132031930473X.

[71] Suh, Y., Wang, J., Tang, S., Mei, T., and Lee, K.M.: *Part-Aligned Bilinear Representations for Person Re-Identification.* In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[72] Sun, Y., Zheng, L., Deng, W., and Wang, S.: *SVDNet for Pedestrian Retrieval.* In *IEEE International Conference on Computer Vision (ICCV)*, pp. 3820–3828, Oct 2017.

[73] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A.: *Going Deeper With Convolutions.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[74] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., and Polosukhin, I.: *Attention is All you Need.* In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.): *Advances in Neural Information Processing Systems (NIPS)*, vol. 30. Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[75] Wang, G., Lai, J., Huang, P., and Xie, X.: *Spatial-Temporal Person Re-identification.* In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8933–8940, 2019.

[76] Wang, X.: *Intelligent multi-camera video surveillance: A review.* Pattern Recognition Letters, 34(1):3 – 19, 2013, ISSN 0167-8655. http://www.sciencedirect.com/science/article/pii/S016786551200219X, Extracting Semantics from Multi-Spectrum Video.

[77] Wei, L., Zhang, S., Gao, W., and Tian, Q.: *Person Transfer GAN to Bridge Domain Gap for Person Re-Identification.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[78] Wei Niu, Jiao Long, Dan Han, and Yuan-Fang Wang: *Human activity detection and recognition for video surveillance.* In *IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, vol. 1, pp. 719–722 Vol.1, June 2004.

[79] Wen, Y., Zhang, K., Li, Z., and Qiao, Y.: *A Discriminative Feature Learning Approach for Deep Face Recognition.* In *ECCV (7)*, pp. 499–515, 2016. `https://doi.org/10.1007/978-3-319-46478-7_31`.

[80] Wu, H., Zheng, S., Zhang, J., and Huang, K.: *GP-GAN: Towards Realistic High-Resolution Image Blending.* ACMMM, 2019.

[81] Yu, H.X., Zheng, W.S., Wu, A., Guo, X., Gong, S., and Lai, J.H.: *Unsupervised Person Re-Identification by Soft Multilabel Learning.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[82] Zajdel, W., Zivkovic, Z., and Krose, B.J.A.: *Keeping Track of Humans: Have I Seen This Person Before?* In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2081–2086, April 2005.

[83] Zeng, K., Ning, M., Wang, Y., and Guo, Y.: *Hierarchical Clustering With Hard-Batch Triplet Loss for Person Re-Identification.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[84] Zhai, Y., Lu, S., Ye, Q., Shan, X., Chen, J., Ji, R., and Tian, Y.: *AD-Cluster: Augmented Discriminative Clustering for Domain Adaptive Person Re-Identification.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[85] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D.N.: *StackGAN: Text to Photo-Realistic Image Synthesis With Stacked Generative Adversarial Networks.* In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[86] Zhang, S. and Yu, H.: *Person Re-Identification by Multi-Camera Networks for Internet of Things in Smart Cities.* IEEE Access, 6:76111–76117, 2018.

[87] Zhang, X., Cao, J., Shen, C., and You, M.: *Self-Training With Progressive Augmentation for Unsupervised Cross-Domain Person Re-Identification.* In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[88] Zhao, H., Tian, M., Sun, S., Shao, J., Yan, J., Yi, S., Wang, X., and Tang, X.: *Spindle Net: Person Re-Identification With Human Body Region Guided Feature Decomposition and Fusion.* In *Proc 30th IEEE Conf on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, July 21-26*, July 2017.

[89] Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., and Tian, Q.: *Scalable Person Re-identification: A Benchmark.* In *IEEE International Conference on Computer Vision*, 2015.

[90] Zheng, Z., Zheng, L., and Yang, Y.: *A Discriminatively Learned CNN Embedding for Person Reidentification.* ACM Trans. Multimedia Comput. Commun. Appl., 14(1), 2017, ISSN 1551-6857. `https://doi.org/10.1145/3159171`.

[91] Zheng, Z., Zheng, L., and Yang, Y.: *Unlabeled Samples Generated by GAN Improve the Person Re-identification Baseline in vitro.* In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[92] Zhong, Z., Zheng, L., Cao, D., and Li, S.: *Re-Ranking Person Re-Identification With k-Reciprocal Encoding.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[93] Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y.: *Random Erasing Data Augmentation.* CoRR, abs/1708.04896, 2017. `http://arxiv.org/abs/1708.04896`.

[94] Zhong, Z., Zheng, L., Luo, Z., Li, S., and Yang, Y.: *Invariance Matters: Exemplar Memory for Domain Adaptive Person Re-Identification.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[95] Zhong, Z., Zheng, L., Zheng, Z., Li, S., and Yang, Y.: *Camera Style Adaptation for Person Re-Identification.* In *Proc of the IEEE Conf on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[96] Zhou, C. and Yuan, J.: *Multi-label learning of part detectors for occluded pedestrian detection.* Pattern Recognition, 86:99 – 111, 2019, ISSN 0031-3203. `http://www.sciencedirect.com/science/article/pii/S0031320318303170`.

[97] Zhou, J., Su, B., and Wu, Y.: *Online Joint Multi-Metric Adaptation From Frequent Sharing-Subset Mining for Person Re-Identification.* In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[98] Zhu, J.Y., Park, T., Isola, P., and Efros, A.A.: *Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks.* In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[99] Zhuang, Z., Wei, L., Xie, L., Zhang, T., Zhang, H., Wu, H., Ai, H., and Tian, Q.: *Rethinking the Distribution Gap of Person Re-identification with Camera-based Batch Normalization.* In *ECCV*, 2020.

[100] Zou, Y., Yang, X., Yu, Z., Kumar, B.V.K.V., and Kautz, J.: *Joint Disentangling and Adaptation for Cross-Domain Person Re-Identification.* In *ECCV*, 2020.