UNIVERSIDADE D
COIMBRA

Ricardo Martins

URBAN SAFETY WITH VIDEO ANALYTICS

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Professor Nuno António Marques Lourenço and Engineer João Garcia and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

July of 2022

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

Ricardo Martins

Urban Safety with video analytics

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Prof. Nuno Lourenço and Engineer João Garcia presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

July 2022

Acknowledgements

I would like to thank Ubiwhere for the opportunity to develop my work. Thank you for support and guidance that allowed me not only to complete this work, but also made me a better professional.

I would also like to thank João Garcia and Professor Nuno Lourenço for the crucial and much appreciated guidance. I am extremely grateful for the efforts they made throughout the year to help me in the development of this work and for the availability to answer my questions at all times.

To my friends, I need to thank you for always standing beside me, pushing me forward and helping me whenever I needed. You are a big reason I could reach where I am. I am extremely grateful for being able to share this journey with all of you.

A special thanks to my family. To my parents, you are outstanding role models and a big part of who I am today. Thank you for your support, not only in the past year, but for always going above and beyond. To my brother and sister, thank you for giving me the strength and encouragement to pursue what I want and for always being there for me.

Abstract

With the population concentrating on urban areas, crime rates increase and cities cannot provide a reliable answer since human resources and infrastructures are limited, creating several problems related to security.

With this work, we propose a platform that, with the use of machine learning, is capable of finding individuals across different video feeds to be used in a surveillance system for smart cities. This document details all the steps to build a solution capable of performing this task. We provide an analysis of the state of the art on the subjects of person detection and person re-identification. Based on the analysis of the current methods available we propose a solution considering two critical steps, person detection and image extraction from videos and person re-identification.

Our approach consists of transforming video into images by sampling one frame per second and performing person detection on those frames. The person re-identification algorithms then go through these detections to find the intended person. We implemented different methods for each of the mentioned steps of our solution. We perform tests to find the best person detection and person re-identification algorithms for our specific solution.

The experiment results show the solution's viability by achieving the objectives set for person re-identification (70% mAP and 80% CMC). We also leave a suggestion of a possible path for future developments regarding the person re-identification performance and new features that can be implemented to improve our solution.

Keywords

Computer vision, Unsupervised Learning, Unsupervised Domain Adaptation, Person Re-Identification, Person Detection

Resumo

Com a população concentrada nas áreas urbanas, as taxas de criminalidade aumentam e as cidades deixam de conseguir fornecer uma resposta fiável, uma vez que os recursos humanos e as infraestruturas são limitados, criando vários problemas relacionados com a segurança da população.

Com este trabalho, propomos uma plataforma que, com a utilização da machine learning, é capaz de encontrar indivíduos através de diferentes alimentações de vídeo para serem utilizados num sistema de vigilância para cidades inteligentes. Este documento detalha todos os passos dados para construir uma solução capaz de realizar esta tarefa. Fornecemos uma análise do estado da arte sobre os temas da deteção e re-identificação de pessoas. Com base na análise dos métodos atualmente disponíveis, propomos uma solução considerando duas etapas críticas, a deteção de pessoas e a extração de imagens de vídeos e a re-identificação de pessoas.

A nossa abordagem consiste em transformar vídeo em imagens através da seleção de um frame a cada segundo de vídeo e realizar a deteção de pessoas nesses frames. Os algoritmos de re-identificação da pessoa passam então por estas deteções para encontrar a pessoa pretendida. Implementamos métodos diferentes para cada uma das etapas mencionadas da nossa solução. Realizamos testes para encontrar a melhor deteção de pessoas e algoritmos de re-identificação de pessoas para a nossa solução específica.

Os resultados das experiências mostram a viabilidade da solução ao atingir os objetivos estabelecidos para a re-identificação de pessoas (70% mAP e 80% CMC). Deixamos também uma sugestão de um possível caminho para futuros desenvolvimentos no que respeita ao desempenho da re-identificação de pessoas e novas características que podem ser implementadas para melhorar a nossa solução.

Contents

Acronyms

AI Artificial Intelligence.

CMC Cumulative Match Characteristics.

CNN Convolutional Neural Networks.

GPU Graphics Processing unit.

IoT Internet of Things.

mAP mean Average Precision.

MEB-net Multiple Expert Brainstorming network.

ML Machine Learning.

NMS Non-Maximum Suppression.

OpenCV Open Source Computer Vision Library.

RPN Region Proposal Networks.

SpCL Self-paced Contrastive Learning.

SSD Single Shot Detection.

UDA Unsupervied Domain Adaptation.

YOLO You Only Look Once.

List of Figures

List of Tables

Chapter 1

Introduction

More than half the world's population live in urban settlements [1]. As cities are growing more rapidly, the demand for things like education, employment and health is growing exponentially, creating challenges for most cities to build infrastructures that can respond to the increase of the population. This has several consequences, one of them being the lack of public security, which leads to an increase in crime, creating an unstable social environment. Some challenges that cities face regarding security include thefts, harassment, gang violence, homicides, unsafe mobility and terrorism. While cities have specific authorities, such as police and judicial courts, to deal with these issues and provide safety, it is not possible to control the entire territory due to their large population density.

Nowadays, modern cities are heavily monitored with security cameras, aiming to improve overall security [1]. Surveillance footage can help monitor large areas without the mass mobilization of human resources. However, public entities do not have the access to all the gathered footage. Even if the footage was available manually, analysing hours of videos is a task that requires a large amount of time and focus.

To tackle these problems we propose a solution using machine learning and person re-identification methods to identify a given person-of-interest in images or videos using a previously captured image as a query. The solution will also use video footage gathered through street surveillance and security cameras.

1.1 Context and Motivation

Ubiwhere was established in 2007 in Aveiro. They work in the fields of telecommunications and Smart Cities. Focusing on solving problems using solutions based on Internet of Things (IoT), 5G, edge computing and Artificial Intelligence (AI).

A recent report [2] shows that, in 2019, over one in four Europeans were victims of harassment and at least 22 million were physically attacked. Many of these crimes are not even reported since victims are afraid of retaliation or are intimidated by

Figure 1.1: Example of an image captured by the Smart Lamppost.

the aggressors. With the use of technology, we can help improve the lives of the citizens. This project aims to tackle these issues, using surveillance footage and machine learning. The implementation of this project will be supported by technology already developed by Ubiwhere, such as the Smart Lamppost and the Urban Platform.

1.1.1 Smart Lamppost

The Smart Lamppost follows a simple modular approach, with scalability in mind, making it easy to add different modules. It offers a highly available infrastructure, replacing traditional lampposts, to provide a number of features to citizens and improve the city's layout. It brings value to counties that want to future-proof their smart city and Mobile Network Operators to cost-effectively deploy their 5G solution. This product offers multiple advantages, including Smart Lighting, an integrated surveillance camera (Figure 1.1 has an image captured by this camera), a LED-based solution that allows an improvement in energy efficiency through remote management and scheduling of lights. It offers an EV charging module that will be effectively transforming light poles into e-charging stations. The smart lamppost also has support for neutral hosting, with no added visual impact. It uses an innovative Neutral Hosting Web Platform, used to speed up the whole deployment procedure, powered this smart shared infrastructure. Finally, the Smart Lamppost also offers a solution for Edge computing. Different service providers can use it to deploy and storage network resources in a distributed way.

Figure 1.2: Example of the Urban platform.

1.1.2 Urban Platform

To take full advantage of all the data gathered and allow for cities to take proper action to manage and improve the life of its citizens, Ubiwhere created the Urban Platform[1]. The goal is to provide a holistic (example shown in Figure 1.2) view of the city's environment through a dashboard that contains all the data gathered by several devices around the city. This product helps city councils to make informed choices to solve the many demanding challenges that modern cities face (environmental monitoring, energy efficiency, mobility, sustainability, among others), while also helping them meet the sustainability development goals defined in [3].

The Urban Platform is a single integrated system that allows the centralised collection and processing of different data from different sources for a single city. It uses real time indicators gathered by the deployed sensors and has relevant features for the work being developed like an occurrence management system that gives integrated and customizable workflows for a more efficient and coordinated response to incidents.

1.2 Objectives

The main objective of this work is to develop a solution that can identify people in surveillance video from multiple non-overlapping cameras. The system will receive a query in the form of an image of a person. Then our solution will look for the occurrences of that same person in the available video feed or in the images available. Another goal is to tune the product in order to keep its performance within the defined constraints of identification capabilities (measured with CMC and mAP) and time.

Since Ubiwhere's Urban Platform already has a service that can detect dangerous situations and manage occurrences, this project is meant to add the ability to find a person in video and the platform to interact with the ML model.

[1] https://urbanplatform.city

In concrete we aim to answer the following research questions:

- What is the most suitable algorithm for person re-identification;
- How can we extract people images from videos;

1.3 Challenges

The problem of person re-identification raises several issues and obstacles, namely:

- Ethics;
- Multiple Cameras and different angles;
- Scalability.

While developing machine learning applications, caution with sensitive data is always needed, especially when dealing with personal data. Since this project focus on person re-identification, using images of different people is inevitable. Using these types of datasets carries some ethical dilemmas, since the images used are of real people. There are cases of some datasets that were gathered from cameras in the street where the participants were not aware they were being filmed and that their images would be used later [4]. Even though we are aware of this problem, it is out of the scope of this project, therefore we will not provide an answer during the development of our solution. When designing solutions for person re-identification problems we need to take into account the use of different cameras. This means that there will be different angles in images from different cameras and image quality is not guaranteed to be the same across all available cameras.

In addition to the problems already mentioned there are others that are specific to our approach to the problem and the final goal of the solution such as:

- Transforming Video to Image;
- Integration in the Urban Platform.

After analysing the literature on person re-identification we opted by using methods that handle images instead of videos.

1.3.1 Transforming video to image

Most state of the art algorithms for this purpose were designed for still image datasets. This may add another step to the solution, regarding the transformation of video into images. There is a need to use a fast method to sample essential video frames to apply image algorithms to videos.

1.3.2 Multiple Cameras

Implementing this solution needs to consider the need to handle input from different cameras. This means that there will be different angles in images from different cameras and image quality is not guaranteed to be the same across all available cameras. These factors present a hurdle to overcome in order to achieve the desired system performance.

1.3.3 Scalability

When considering the use of this solution in a real scenario, scalability is another problem that needs to be addressed. Here, scalability is related to the effects caused by introducing additional cameras for a scenario. To allow the system to scale properly will be challenging for the development of this solution. Since computer vision algorithms are usually complex and demand time to execute, adding more entries will inevitably increase the execution time.

1.3.4 Ethics

While developing machine learning applications, caution with sensitive data is always needed, especially when dealing with personal data. Since this project focus on person re-identification, using images of different people is inevitable. Using these types of datasets carries some ethical dilemmas, since the images used are of real people. There are cases of some datasets that were gathered from cameras in the street where the participants were not aware they were being filmed and that their images would be used later [4]. Even though we are aware of this problem, it is out of the scope of this project, therefore we will not provide an answer during the development of our solution.

1.4 Contributions

With this work we contributed to the development of a solution that allows to find people across multiple non-overlapping cameras. Our main contributions were:

- We make an analysis of the most relevant state of the art algorithms in the fields of object detection and person re-identification.
- We provide a reliable method to extract images of people from videos using different object detection algorithms.
- We provide a study on the effectiveness and performance of the selected methods for person detection and person re-identification. We evaluate these algorithms on performance using different metrics such as mAP, CMC and time required for execution.

- We propose a framework that allows to perform person re-identification on video. The proposed solution uses an image of a person as query, and searches the available instances in the gallery to find and show the positive matches. This framework provides a solution for the problem we set out to solve re-identify people in non-overlapping cameras within the set constraints for performance.

1.5 Document Structure

The rest of the document is structured as follows.

- **Chapter 2**: This chapter contains some important concepts for the development of this project. It also contains the results of the research done on previous work in the field of object detection and person re-identification.
- **Chapter 3**: Here we show the adopted development methodology as well as the planning for the second semester.
- **Chapter 4:** This chapter shows a detailed description of the implementation of our solution.
- **Chapter 5:** In this chapter we show the experiments we used to compare the implemented configurations of the algorithms for person detection and person re-identification.
- **Chapter 6** presents the main conclusion and lessons learned from this work. We also present a path for future developments to expand this work and improve the results achieved.

Chapter 2

Background and Related Work

This chapter contains a brief introduction to the main topics that will be relevant for this work. We start by introducing the concepts of Machine Learning (ML) and computer vision, then discussing methods and other concepts that will be used for the rest of the work. After that we will look into an overview of relevant work from literature, that is closely related to our project. Focusing on object detection and person re-identification.

2.1 Machine Learning

With the ever increasing amounts of complex data gathered everyday through the internet, phones, sensors and others, there was a need to create automated processes that could analyse the data and uncover knowledge from the discovered patterns. This is what ML algorithms provide [5].

Machine Learning is a sub-area of Artificial Intelligence (AI) that is focused on the development and study of methods to allow computer systems to learn to solve certain problems without the need for explicit instructions. Computer systems empowered with this methods are able to learn what they need from available data by automatically detecting hidden patterns. The system then applies these patterns in order to achieve a solution for the proposed problem [5].

The algorithms in the field of ML can be divided into the following three categories: [5].

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

The first two, supervised and unsupervised learning algorithms take data as inputs to find the function for the specified task. This two categories are differentiated by the existence (supervised) or not (unsupervised) of labelled input. The

other type of algorithms is reinforced learning. The distinctive feature of these algorithms is that they learn through interaction with the environment, penalties and rewards.

Even though the selected model is very important for a system using ML there are other important factors when tackling these types of problems. Usually ML projects follow distinct steps to improve the overall performance. The normal pipeline for these projects can be divided in six steps as it is shown in figure 2.1. After clarifying the problem we want to solve we need to find a way to gather meaningful data that will help us solve the problem. After that we need to process the data in order to make usable in ML models (discussed in section 2.1.1). Considering the problem and the data we have we then select an appropriate model to use and evaluate its performance. If the model achieves the required performance it is ready for deployment, if not we need to change the model used.

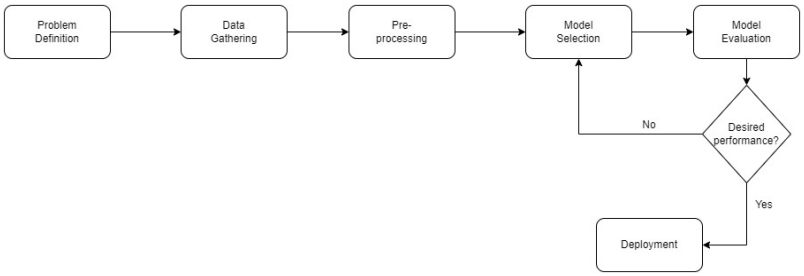


Figure 2.1: Standard ML project pipeline

Supervised learning implies using a dataset with input-output pairs to reach a function which maps input to output [6]. The expected output in the dataset is referred to as labels. Supervised methods are usually applied in classification problems, where the goal is to find the correct label for a given input, and the number of outputs is known as a priori. These methods are also used in regression where instead of labels we have continuous values.

Using supervised learning can become a problem when considering real-world scenarios, where the data gathered is usually unlabelled [7]. Since, manual labelling is extremely time consuming, we can use unsupervised learning to find patterns without the need for labels. Among the different uses of unsupervised learning methods the most relevant to this work is clustering. These methods are used to group instances based on their similarities [8]. Instead of making predictions these algorithms aim to extract structures from data samples and separate different instances based on their characteristics, without specifically giving it a class or label [8].

An alternative availabel for some problems is also Reinforcement Learning. This type of machine learning algorithms do not use datasets. Instead, these methods consider an agent that has interactions with an environment. Sometimes, it is possible to have multiple agents and they can interact with each other as well. The interactions affect the state of both the agent and the environment. When this happens, a penalty or a reward will be given to the agent [6]. The goal of this type of learning is to maximize the reward for the agent, since that allows improvement in the desired task.

2.1.1 Data Preparation and Pre-processing

One of the first steps in a machine learning project is the pre-processing of input data. It is common that the gathered data is not ready to be used by a machine learning model. Generally, for supervised and unsupervised learning, we want to use complete datasets (no missing values), the different features should be represented with numerical values and have similar magnitude. We use the data pre-processing stage to apply these transformations to the available data. During this stage, not only we can have a better understanding of the data but in most cases it helps improving the results achieved by the applied machine learning models.

Most machine learning algorithms cannot handle entries with missing values in the dataset. For this reason, resolving missing values in our data is an important part of pre-processing [9]. There are two ways of dealing with missing data, either delete the incomplete samples or find a solution to fill missing values. When deleting values, we need to be careful about the repercussions. For example, if there are many entries with missing values, we risk discarding many important sources of information. Deleting entries may also add bias to the model when the missing values follow a pattern. For example, if only one class has a missing feature, those values cannot be deleted [9]. If we decide to replace the missing values, we can use machine learning methods like the K-Nearest Neighbors to predict the feature values. However, this becomes a problem in large datasets. The other option is to use a formula to fill the missing values, for example, the feature average [9].

After these steps, we need to evaluate and understand the dataset in order to proceed. In case of classification problem, we need to check if the target output is discrete. If the original data was continuous, creating *bins* (intervals of values) is an option. If the target values are in text, it is possible to assign numerical values to represent each class. When making the analysis of the dataset, we need to make sure that all classes are evenly represented, otherwise the results obtained by the trained model might be misleading [9]. To solve this problem, we can use sampling strategies in pre-processing or adjusting the used metrics to account for the imbalanced dataset.

Feature selection is another important step in the machine pipeline. It consists of methods that are used to find the most informative features in a dataset. To do this, we look at the correlation between all the features and the target or we can use measures like the chi-square statistic [9]. There are also methods to create new features from the ones that are available, while reducing the problem dimension. Algorithms like the Linear Discriminant Analysis and the Principal Component Analysis are used for these situations. Feature engineering is used when the available features are not informative enough. We use this process to create new features with the information already available.

Finally we need to to transform the input data to assure that all the data is in the same order of magnitude. This is crucial because many of the commonly used machine learning algorithms assume features on the same scale [9]. Different methods can be used in order to transform and rescale the features on the pro-

vided dataset. Even though there are many ways to re-scale a dataset, we can use stardardisation or normalization. With standardisation, we change the data to make it approximately normally distributed. After being transformed (using formula 2.1), the data has a mean of 0 and a standard deviation of 1. In normalization the values are shifted and re-scaled and changed to a range of values between 0 and 1 (using formula 2.2) [10].

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2.1}$$

$$X' = \frac{X - \mu}{\theta} \tag{2.2}$$

For this project, in particular, the datasets contain videos or images, therefore there is a need to apply methods particular to these types of data. With videos, we will transform them into images, since processing videos are computationally more expensive. For all the images used, there is a need to keep consistency in image size, both in the training and target dataset. We also want to keep as much information as possible so we won't remove the colors.

2.1.2 Model Selection and Evaluation

This section will describe the methods used to evaluate the performance of ML models, both the most commonly used metrics in machine learning and the performance measures used in the domain of Person Re-Identification.

When applying ML models, there is the need to account for the bias-variance trade-off. Bias is the difference between the obtained model and the true model of the data [11]. Since models cannot learn the true function we are seeking to approximate bias is inevitable. Variance measures the errors introduced in the model during the training step because of sampling noise [11]. We consider a trade-off between bias and variance since when we lower the bias of the model the variance will increase and vice-versa. The aim of controlling the bias-variance trade-off is to find the model that achieves the best balance between both for the project. This is a hard balance to find since usually, models with low variance have high bias and models with low bias have high variance. The complexity of the model can be used as an indicator of these values. Complex models tend to have small bias and high variance since they become too complex for the training data [11]. The bias-variance trade-off relates to the concepts of underfitting and overfitting. Models that are too complex have the risk of overfitting. This happens when the model gets too close to the training data, causing the loss of generalization capabilities. Simpler models might not learn significant patterns, meaning the model cannot extract valuable information from the data (underfitting).

In order to select the appropriate model for a given problem we need to evaluate its generalization ability. In other words we need to evaluate how it performs when presented with unseen data. Several approaches can be taken at the training step. When we have sufficient data, the most effective method to evaluate the

		Predicted Class	
		Positive	Negative
True Class	Positive	TP	FN
	Negative	FP	TN

Table 2.1: Confusion Matrix example.

model, is to divide the dataset into three parts: training, validation and testing. Even though there is no metric to determine how to split the dataset, a typical split is 50% for training, 25% for validation and 25% for testing. The testing partition should be unknown at training time and is used to evaluate the generalization ability of the model. When the data is not enough to split without hurting the training step, we can use cross validation. This method consists of dividing the dataset into K folds. Train the model in K-1 partitions and use the other to evaluate the model. Repeat the process K times, always changing the partition used for testing and calculate the average of prediction error to get an estimate of the generalization error [12].

We can evaluate the machine learning methods with different methods and metrics. In a classification problem we can present the testing results in a confusion matrix. This contains the count of positive and negative samples that were correctly identified, true positives (TP) and true negatives (TN) respectively, and the count of positive and negative samples that were mis-classified false positives (FP) and false negatives (FN) respectively. An example is shown in Table 2.1.

With the values of the confusion matrix we are able to calculate metrics used to evaluate ML solutions. Common measures in classification problems include the F1-score, accuracy, precision and recall defined in [5].

While accuracy is the fraction of the instances were predicted correctly (calculated using equation 2.3), precision measures the fraction of our predictions that are actually positive (calculated using equation 2.4) and recall measures what fraction of the positives we actually predict (calculated using equation 2.5). The F1 score is calculated using equation 2.6 and is simply the harmonic mean of precision and recall.

$$accuracy = \frac{TP + TN}{Numberof predictions} \tag{2.3}$$

$$precision = \frac{TP}{TP + FP} \tag{2.4}$$

$$recall = \frac{TP}{TP + FN} \tag{2.5}$$

$$F1score = \frac{2 * precision * recall}{precision + recall} \tag{2.6}$$

The solutions presented in the project will be evaluated by the metrics usually considered in person re-identification work (mean Average Precision (mAP) and

the Cumulative Match Characteristics (CMC)). Besides these measures, the solutions will also be evaluated by the required time to perform the task.

Mean Average Precision

The mAP is a widely used metric in image retrieval [13]. It is calculated as the average area under the curve of precision-recall (average precision) of all considered classes. As it is shown in [14], to calculate the average precision, it is normal to consider 11 points in the precision-recall curve and calculate the average of maximum precision value for these 11 recall values. Formula 2.7 shows a more precise definition for average precision. To calculate the mAP value we consider the average of all classes.

$$AP = \frac{1}{11} \sum_{r \epsilon \{0, 0.1, \ldots, 1\}} AP_r \tag{2.7}$$

$$AP = \frac{1}{11} \sum_{r \epsilon \{0, 0.1, \ldots, 1\}} p_{interp}(r) \tag{2.8}$$

$$p_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r}) \tag{2.9}$$

Cumulative Match Characteristic

This performance measure, also referred as rank-k accuracy in some works, is usually used to evaluate solutions in image classification (facial recognition, for example). It represents the probability of finding the correct match over the first n ranks [13]. Meaning that, by evaluating a model's performance this way, we are not considering how often is the best match the right one but how often the right image is in the top n matches. When this measure is referred as rank-k accuracy, the k shows how many of the top matches are considered. If we consider $k = 1$ this measure will show the accuracy of the model.

2.2 Computer Vision

Computer vision is a field of study focused on helping computers to understand and process visual content. The human's ability to see inspires most methods used and developed for problems related to this area, since the main goal is to replicate that capability for machines [15]. It is a multidisciplinary area involving fields of engineering, computer science and bio-sciences, which may include the use of advanced methods, like complex statistical approaches for specific cases or with the use of different and generalized machine learning algorithms.

In order to apply machine learning to problems such as image classification, object detection and tracking, it was crucial to extract information from visual data. Computer vision is a field of artificial intelligence that enables that. Usually with

the use of neural networks. There are many approaches that use in the field of computer vision however we will focus on Convolutional Neural Networks (CNN) since these networks are the most relevant for this work. Solutions using CNNs are widely used in the field of computer vision [15], because given enough quality data, these models can "learn" how to perform the desired task.

2.2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are widely used for computer vision tasks, since they have shown great performance in image classification problems (GoogleNet [16] and ResNet [17] for example). These type of networks usually take images as inputs and are composed of single or multiple blocks of convolution (convolution layers) and sub-sampling layers (pooling layers), after that one or more fully connected layers and an output layer. The design of CNN allows for the neurons to learn local feature maps from the inputs to the outputs [6]. A example of how these networks function is shown in figure 2.2.

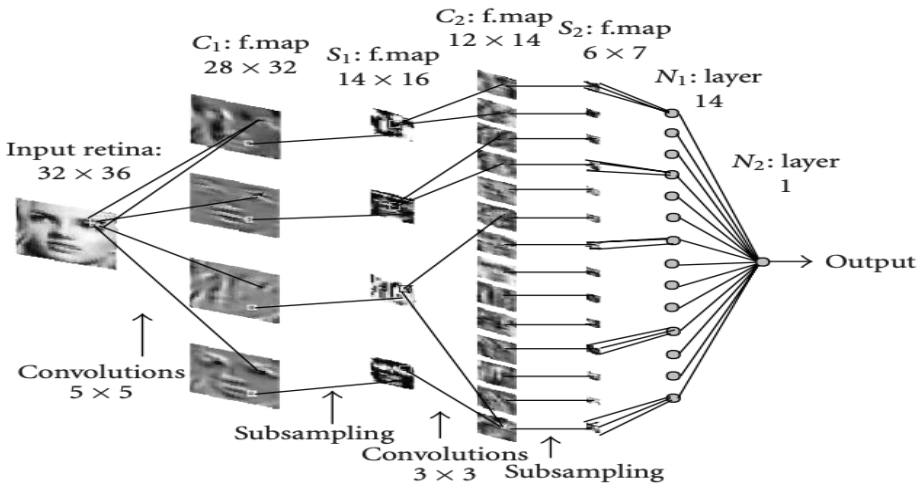


Figure 2.2: Example of Convolutional Neural Network with two convolution layers, two pooling layers and two fully-connected layers from [15]

Convolution Layers

In the convolutional layers, a CNN uses various kernels as filters to convolve the input data. The kernels is a matrix of integers that is used on a subset of the input image. Each pixel is multiplied by the corresponding value in the kernel, then the result is summed up for a single value representing a grid cell in the output feature map [18]. The used filters are smaller than the input dimensions. The kernel filter goes through the data, estimating the dot product between the weights of the kernel and the value of the input image at each step. This creates an activation map with a dimension equal to the number of filters applied [19]. Figure 2.3 shows a simple example of how a convolution layer works. Using multiple

convolution layers usually grants the ability to learn more abstract features to a CNN.

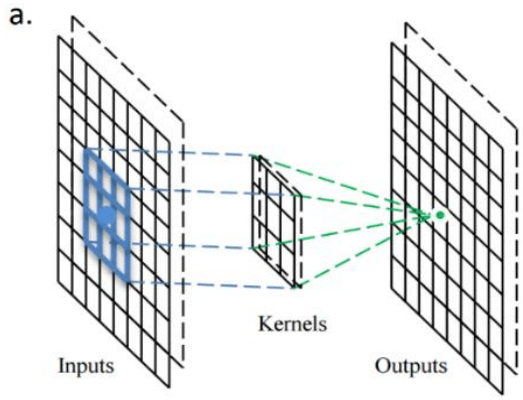


Figure 2.3: Example of Convolutional layer from [20].

Pooling Layers

Usually, a pooling layer follows a convolutional layer. These layers are used to reduce the dimensions of the feature maps and network parameters calculated in the convolutional layers [21]. This process is done by taking the values of several neighborhood feature maps and extracting one representative value [22]. For this step, the most common strategies used are average pooling, where the chosen value is the average of the selected segment from the feature map and max pooling where the output value is the maximum of the selected segment [9].

Fully Connected Layers

Fully-connected layers perform like traditional neural networks. These layers allow us to feed forward the neural network into a vector with a pre-defined length. This vector can create an output layer by selecting the desired number of categories for classification or to create a feature vector for follow-up processing [21].

2.2.2 Transfer Leaning

Most ML models follow the assumption that the training and target data follow the same distribution. This means that when the distribution changes, the model has to be retrained from scratch to accommodate the new data [23]. However, finding complete datasets to represent each situation is often impossible. Transfer learning consists of training a model in a source domain, and then apply or “re-train" the model in a related target domain [23]. By using transfer learning, we can reduce the number of required examples from the target domain, which also results in less computational effort and improved generalization.

The usual transfer learning approach is to train a base model and then use its first n layers on the target network. The remaining layers of the target network are then randomly initialized and trained toward the target task. Depending on the size of the target dataset, the layers trained in the source domain can stay unchanged or can be fine-tuned by backpropagating the error. When backpropagating the errors we need to be careful. If the target dataset is small and the number of parameters is large, fine-tuning may cause overfitting, so the weights of the layers are often left unchanged from the training in the source domain. If the target dataset is large or the number of parameters is small, so that overfitting is not a problem, then the base features can be fine-tuned to the new task to improve performance [24].

For the subject of person re-identification among the several pre-trained networks used as backbones in different CNN architecture the most common is the ResNet [17] and for this type of problem it is pre-trained a variety of datasets, including the ImageNet [25] and other specific datasets mentioned in section 2.5.1.

2.3 Object Detection

To be able to re-identify people we will need to get images from them. To find an adequate solution for this problem we decided to study the most used object detection algorithms. Object detection is one of the main problems in computer vision and a crucial part of people tracking and identification. This means that there is extensive work and research done on the subject. We will look into two variations of these methods: 1) "One-shot“ detectors (You Only Look Once (YOLO) and Single Shot Detection (SSD)) that only need to go through the image once to detect and classify the objects; 2) Methods that separate the detection from the classification (Faster R-CNN, Mask R-CNN).

2.3.1 You Only Look Once

YOLO is an object detection framework that has been developed and updated over time. The first version was published in [26]. This method approached object detection as a regression problem and proposed a single CNN architecture to find the bounding boxes and calculate class probabilities. In [27], there were several improvements made in order to solve some shortcomings of the previous version. Some changes include batch normalization and the implementation of anchor boxes. For the third version of this method, only a few changes were made in order to increase the accuracy of the bounding box and class predictions. The changes are documented in [28]. The most recent developments on this algorithm for object detection are presented in [29]. With the change of author, a variety of features and a new architecture are proposed in order to improve the results of the CNN.

2.3.2 Single Shot detection

Similar to the YOLO, the SSD only need to "look“ at the image once to detect multiple objects [30]. This algorithm divides the image using a grid, where each grid cell is responsible for detecting objects in that region of the image. If an object is found it predicts it's class, if nothing is found that region is labeled as background and is ignored. Each grid cell used in SSD can be assigned with multiple pre-defined anchor boxes. Each of these boxes is responsible for a size and shape. The anchor box that overlaps the most with an object will be used to predict that object's class and location [31]. Since every anchor box is defined by an aspect ratio and a zoom level this algorithm is capable of dealing with objects of different sizes and shapes. There are usually two different models considered: SSD300(faster but with lower resolution) and SSD512 (more accurate but slower).

2.3.3 Faster R-CNN

This region-based CNN (R-CNN) is an object detection framework proposed in [32]. It is composed of two different modules (one for Region Proposal Networks (RPN) and one for object classification) and builds on the what was achieved in Fast R-CNN [33]. The Faster R-CNN first module is a deep fully convolutional network that receives a feature map generated by the backbone layer as input and after applying a sliding window on those inputs, the network outputs the anchors. With these anchors we can represent the most important regions of the input image. The second module, used for classification, is the same used in the predecessor of this network (Fast R-CNN [33]). This classifier considers uses two layers for classification, one is a binary classifier that is used to decide if a region is background or an image, the other is a softmax layer to predict the class scores for the objects found. Because of the use of the extra network, the RPN, this algorithm becomes more complex and slower compared to YOLO and SSD. However it shows a significant increase in performance.

2.3.4 Mask R-CNN

This approach, presented in [34], applies bounding-box classification and regression in parallel. Even though it is used mainly for image segmentation it can also be used in object detection. Similar to Faster R-CNN, previously described, Mask R-CNN also adopts a two-stage method. The first stage is used for finding the bounding boxes RPN. In the second stage, in parallel to predicting the class and box offset, Mask R-CNN also outputs a binary mask for each RoI (Region of Interest). This mask encodes an input object's spatial layout. The spatial structure of masks can be addressed naturally by the pixel-to-pixel correspondence provided by convolutions.

2.4 Person re-identification

Person re-identification corresponds to the problem of finding someone given an image of that person as query. This query is usually an image or video feed and the goal is to find if the person in the query reappears in another place or time [35]. To construct solutions in the domain of person re-identification we need to consider the different characteristics of the problem. We can divide the problem in sections to be able to narrow our focus on the better solutions. We can consider the following dimensions [36]:

- **Query-type:** This property is used to analyse a model regarding the data it will be performing re-identification and how the data is aquired.
- **Data modality:** Defines if the data used for the query and the gallery have the same modality (Homogeneous re-identification) or not (Heterogenous re-identification).
- **strategy:** This property can be used to group the different algorithms by the answers they provide to scalability, pre-processing, architecture, post-processing and noise robustness.
- **Setting:** Describes if a model is used for a closed-world or an open-world setting.
- **Learning type:** Shows if the algorithm uses supervised or unsupervised learning.
- **Approach:** Describes how the features are extracted.
- **Context:** Whether the data available has other information besides the image itself (i.e camera position).

Figure 2.4 shows an overview of the different variations of re-identification solutions. We can make decisions based on our problem definition, for example the query type will be rgb to rgb, we have non-contextual data since there are only videos or images and it is not provided extra information and will use homogeneous data since the type of data used for the query is similar to the data in the gallery. We will focus on deep learning methods for our solution since these methods get better results compared to hand-crafted systems [37], and will separate them regarding the setting they focus (open world or closed world).

2.4.1 Closed-World Person Re-Identification

Most of the work done in person re-identification has been made in closed-world. This type of scenario has a specific set of assumptions. Closed world scenarios assume that each captured image of a person is represented by a bounding box. The training dataset has enough annotated data to allow for the application of

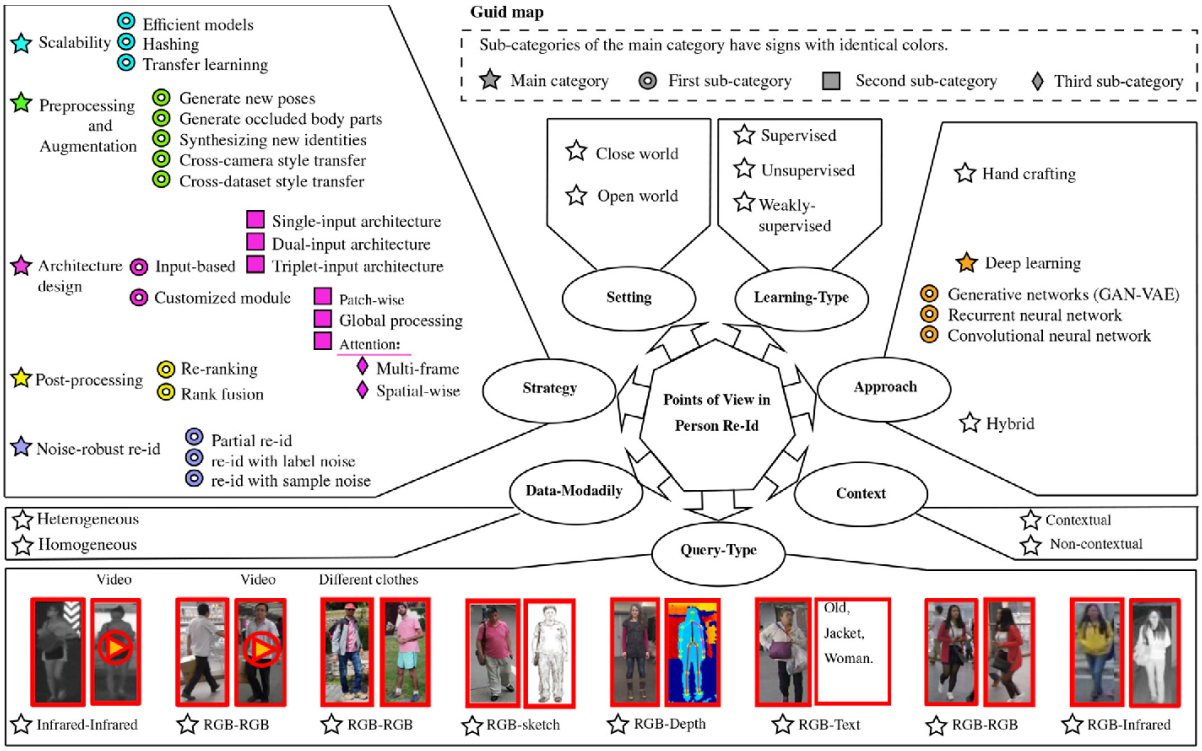


Figure 2.4: Dimensions of a person re-identification problem [36].

a supervised discriminative person re-identification model. The annotations are correct and the queried identity must appear in the image set [13].

A standard algorithm designed for closed-world person re-identification can usually be deconstructed in three key components. Feature Representation learning that focuses on the feature construction strategies, Deep Metric learning that aims to define training objectives with different loss functions or sampling strategies, and Ranking Optimization which concentrates on improving the retrieval performance in the testing stage [13].

Even though several algorithms already surpassed human classification, the closed world scenario is not very representative of real-world situations, since most of the required assumptions cannot be guaranteed in most scenarios.

2.4.2 Open-World Person Re-Identification

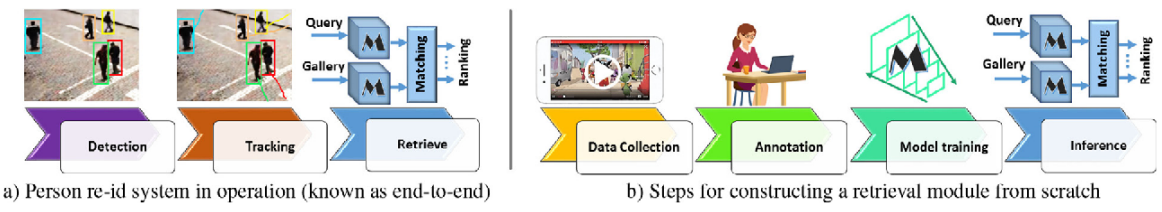
Compared to closed-world re-identification, this is a more realistic scenario, since most of the assumptions of the closed world setting (section 2.4.1) cannot always be met. For example, real-world scenarios are subject to the lack of labels or incorrect labels, and the queried identity is not guaranteed to appear in the image set. To overcome these problems, different algorithms are used for this approach, they can be separated into different categories [13].

Heterogeneous Person Re-Identification

Heterogeneous Re-Identification methods aim for challenges that are not in the scope of our project. These methods propose solutions to deal with problems beyond in intra-modality discrepancies, such as resolution and color discrepancies and some more complex problems for example match photos with sketches or with text descriptions [38]. The Heterogeneous methods in person re-identification often rely on the different camera settings and specifications or inputs, making them good for specific scenarios. For example, some solutions ([39] and [40]) use depth images to gather information on discriminative local regions of the human body. Other methods rely on the use of infrared cameras in low-light environments. Since we want to provide a more generic solution to this problem,the types of inputs used in these methods do are not similar to the situations we want to solve. Therefore taking into account the specifications of the cameras installed in the Smart Lamppost it was decided not to pursue this type of algorithms.

End-to-end

End-to-end solutions are also applied to the open-world person re-identification problem. These algorithms aim to reduce the reliance on an additional step to generate bounding boxes. Figure 2.5 shows a comparison of how these models work when compared to the most commonly used techniques for re-identification. In [41] is proposed a single convolution neural network to handle both detection and re-identification. The neural person search machines shown in [42] performs a recursive person localization in an image by shrinking the target image in each of the recursion steps. A similar approach was used in [43] by using deep reinforcement learning to accurately select re-identification attention to increase the accuracy auto-detected bounding boxes. In order to take full advantage of the information gathered [44] applied graph learning in order to use the context of the query. While this approach to the re-identification problem seems to be more efficient in a production environment these algorithms tend to fall behind in performance especially in a open world setting.



a) Person re-id system in operation (known as end-to-end)

b) Steps for constructing a retrieval module from scratch

An end-to-end re-id model detects and tracks the individuals in a video, and then retrieves the query person, while a typical re-id model focuses on the re

Figure 2.5: End-to-end person re-identification systems compared to standard re-identification techniques [36].

Unsupervised Methods

One challenge in person re-identification is the need for large labeled datasets. This poses a problem, since data is not always available and even when it is, the cost of annotation is high. Using Unsupervied Domain Adaptation (UDA) is a common approach for unsupervised person re-identification methods. This method consists of transferring the knowledge from a labeled dataset (source domain) to the unlabeled dataset (target domain) [45].

Using GANs to generate target domain images is a popular approach among the unsupervised learning methods used in person re-identification [13]. By using this type of methods, it is possible to use supervised learning on the generated images. Several algorithms use this kind of approach. Other algorithms take advantage of UDA by learning a transferable deep person re-identification model using both the labeled source domain and unlabeled target domain. The Table 2.2 shown in [13] is a good summary of the state-of-the-art for unsupervised learning methods used in person re-identification problems.

As we can see in Table 2.2, Self-paced Contrastive Learning (SpCL) [46] and Multiple Expert Brainstorming network (MEB-net) [47] show great results in both benchmark datasets when compared to the other unsupervised methods. The MEB-net explores a different path regarding unsupervised domain adaptation. This algorithm applies ensemble learning in unsupervised domain adaptation for person re-id. This approach consists of having multiple networks with different architectures pre-trained with a source domain. Using different architectures and mutual learning enforces the adaptability and enhances the discrimination capability of the model. This model uses two stages. In the first stage, supervised learning is used to train the “expert" models on the source dataset. In the second step, the pre-trained experts are adapted to the target domain iteratively (unsupervised learning) [47]. When applied to the target domain, the algorithm uses a clustering approach to generate pseudo-labels and extracts features. By averaging the features extracted by every model, we get the ensemble features. In each epoch, each of the expert models extracts the features of image samples from the dataset. The selected samples are grouped in different clusters using k-means clustering and the cluster ids are used as pseudo-labels. To take advantage of the knowledge extracted by the different models, the MEB-net implements a mechanism to transfer knowledge among the experts. In order to achieve this collaboration, the class predictions from one model are used by the others as labels in training. To avoid error amplification, the models are temporaly averaged [47]. Since the expert models have different architectures, the learned features in training will be different and the discrimination capability will also be different. To solve the discrepancies, an authority regularization scheme is proposed where the authority of each of the experts is measured by the inter and intra-cluster scatter of each cluster. Figure 2.6 shows a representation of how this model works.

SpCL does not ignore the data from the source domain and the un-clustered instances of the target domain. Instead, consider all the data (source domain classes, target domain clusters and unclustered instances), taking advantage of all the information using the unified contrastive loss. The data is stored and dynam-

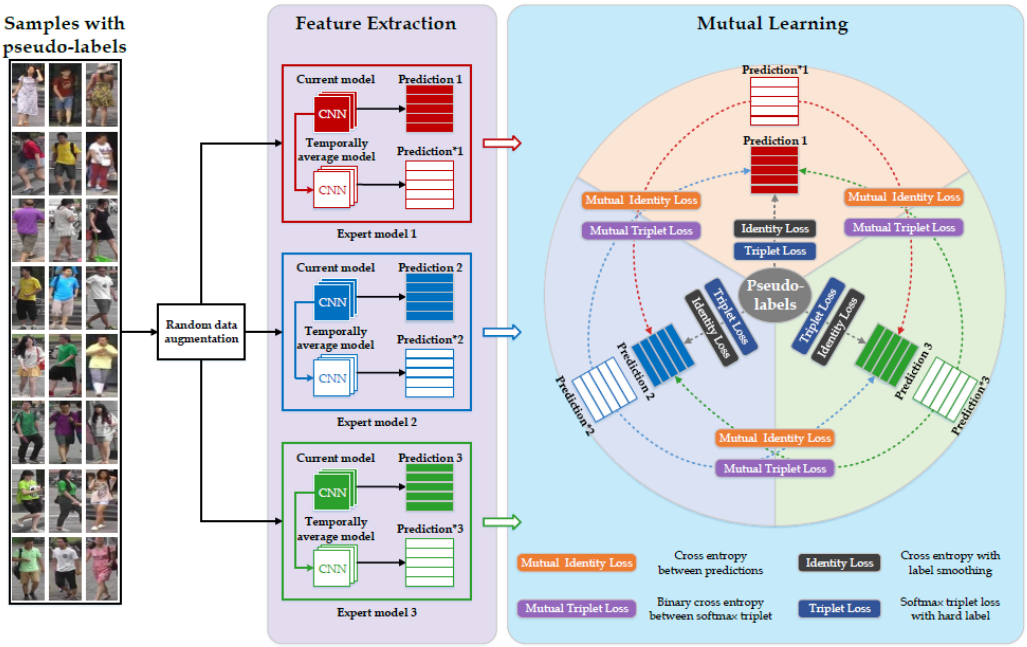


Figure 2.6: Representation of the MEB-net model shown in [47] and available with the code in https://github.com/YunpengZhai/MEB-Net.

ically updated using a hybrid memory. The self-paced learning strategy avoids the amplification by using reliable clusters [46].

This unified contrastive learning uses the class centroids instead of "weight" vectors. It differs from other contrastive loss functions, since it distinguishes between the different class points above mentioned (source classes and clustered and unclustered target instances) instead of only separating instances. The hybrid memory starts with the source-domain class centroids {w} and the initial target-domain, encoded, instance features {v}. After that, the target-domain cluster centroids {c}, are initialized with the mean of the encoded target features {v}. Every iteration, the memory is updated, changing the feature vectors and the class centroids. The self-paced learning strategy consists of adding a re-clustering step before each epoch. For the re-clustering step, only the most reliable clusters are selected, the unreliable clusters are disassembled back to un-clustered instances. The reliability of a cluster is measured by its independence and compactness. Since clusters should be independent of each other and the inter-sample distances should be small inside a cluster, a reliable cluster means high independence and low compactness [46]. Figure 2.7 shows a representation of how this model works.

Noise Related Re-Identification

Person re-identification presents several challenges to match images of people. To accurately perform this task we need to take into account the possibility of changes the lighting, view angle and pose of each image and also the occurrence of occlusions (partial body images) [52]. The occlusion is problem is particularly

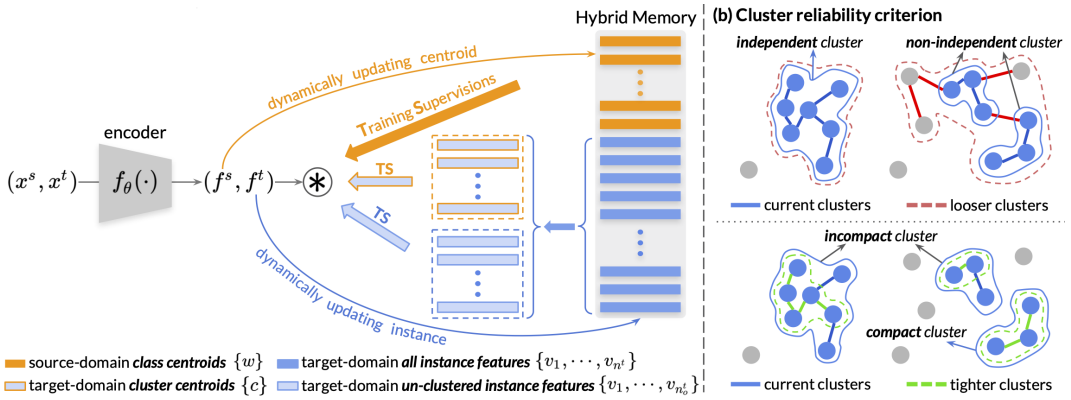


Figure 2.7: Representation of the SpCL model shown in [46] and available with the code in https://github.com/yxgeee/SpCL.

Unsupervised Learing Methods in Person Re-ID						
			Market-1501		DukeMTMC	
Method	Source	Gen	mAP	Rank-1	mAP	Rank-1
SpCL [46]	Data	No	90.3	76.7	82.9	68.8
MEB-net [47]	Data	No	89.9	76.0	79.6	66.1
MMT [48]	Data	No	87.7	71.2	78.0	65.1
SSG [49]	Model	No	80.0	58.3	73.0	53.4
HCT [50]	Model	No	80.0	56.4	69.6	50.7
CR-GAN [51]	Data	Yes	77.7	54.0	68.9	48.6

Table 2.2: Performance of different State of the art unsupervised person Re-Identification algorithms on two datasets. "Source" represents if it utilizes the source annotated data in training the target model. "Gen." indicates if it contains an image generation process [13].

difficult since the information loss is irreversible. Some algorithms like [52], [53] and [54] are focused on these particular situations. The sample noise problem refer to poorly detected images, outlying regions or background clutter within the person image. Other types of noise in data include label noise that is unavoidable due to annotation error [13] and sample noise that refers to poor person detection or the problem causing outlying regions [13].

Video-Based Re-Identification

Video specific methods in person re-identification have received less attention when compared to image based re-id. Despite of this several improvements have been made over the years on the most used datasets [13]. One of the advantages of using video is the possibility to use multiple frames within the video sequence. This allows the mitigation of the problem created by occluded regions. Current state of the art methods like [55] take advantage of the spatial and temporal features gathered in with videos. Using a determined number of frames, the STA model [55] generates a 2-D matrix which assigns an attention weight for each spatial region of each frame.

2.4.3 Loss Functions

The loss function is used to compute the distance between the current output of the algorithm and the expected output. These functions are used to optimize the proposed solutions and select the best candidates. In machine learning problems, loss functions are mainly used to minimize the error in the selected problem. Different loss functions can be used, however, for the specific problem of person re-identification among the different loss function used are the Triplet loss (section 2.4.3) and contrastive loss (section 2.4.3). We will focus on these functions since they are the most relevant to this work.

Triplet Loss

The triplet loss function takes advantages of similarities and dissimilarities between all the examples. This is achieved by considering three different data points, the anchor image (a), a positive example (x_p) that is like the anchor point and a negative example (x_n) that is dissimilar. When considered labeled data, we usually define similarity with class labels.

$$L_{tri}(\theta) = max(\sum_{y_a=y_p\neq y_n} [M + D_{a,p} - D_{a,n}], 0) \tag{2.10}$$

By using the formula 2.10 it is guaranteed that the anchor point is closer to its projection in the positive class (y_p) than to any point belonging to the negative class (y_n) by at least a margin M [56]. When optimizing this function over the dataset, the points belonging to the same class will eventually become closer amongst themselves and clusters will appear. This is important for person re-identification since over time all the images belonging to the same person will stay in one cluster. However, a big problem in using this loss function is that as the dataset grows, the number of triplets will also increase to the point of making the training in very large datasets impractical [56].

Contrastive Loss

The contrastive loss function is used to optimize the network parameters so that the learned discriminative features can successfully bring the positive pairs together and push apart the negative pairs [57]. Even though there are different implementations of contrastive learning, they all follow the same principle. The general formula for this loss function defined in [58] is shown in equation 2.11.

$$L_{con}(i, x_1, x_2) = (1 - i) * max(0, m - d)^2 + i * d^2 \tag{2.11}$$

Considering x_1 and x_2 a pair of input vector, d is the euclidean distance between the pair and i a binary value that allows the distinction between matching and unmatching pairs ($i = 1$ means the considered pair is positively matched and $i = 0$ the pair is negatively matched).

Name	Number of Identities	Number of cameras	Number of Images
CUHK03 [60]	1467	10(5 pairs)	13164
DukeMTMC-reID [61]	1812	8	36441
Market1501 [62]	1501	6	32217
MSMT17 [63]	4101	15	126441
Airport [64]	9651	6	39902

Table 2.3: Different datasets used in Person Re-Identfication.

2.5 Resources and tools

In this section the tools that will be used as well as the available datasets that will be used to train and evaluate the proposed solutions. All the resources and used frameworks are open-source.

2.5.1 Datasets

There are several datasets used as benchmarks in person re-identification problems [1]. Table 2.3 shows the most common datasets used in research. Besides the datasets shown in table 2.3 we can also consider the PersonX [59], which is a tool that allows to generate synthetic images of people in diferent backgrounds and viewpoints. Most of the research done using these datasets use the same performance metrics decided for this work. This is advantageous since it gives an overview of the expected results of the implemented models. It is important to note that some of this datasets were retracted due to privacy issues and are no longer available to use, however we can still use the results obtained with these datasets as references for the expected results. Based on availability and similarity to the images captured by ubiwhere's smart lamppost we decided to use the PersonX dataset (figure 2.8c) and the CUHCK03 (figure 2.8b) dataset for training and the Market1501 (figure 2.8b) dataset for testing and performance benchmark.

2.5.2 Tools

For the development of this project we consider the most common open-source frameworks used in deep learning projects. For simplicity and to take advantage of previous experience we will only look into the most common tools that can be used with python. We opted for to focus on the technologies based in python to because it is the technology adopted by the company, it is one on the most used programming languages in ML development and the familiarity with the tool was also a factor. Among the available frameworks we selected the following:

[1] https://github.com/NEU-Gou/awesome-reid-dataset/blob/master/README.md

(a) Example from the Market1501 dataset.

(b) Example from the CUHK03 dataset.

(c) Example from the PersonX dataset.

Figure 2.8: Examples of the images available from each of the used datasets.

- **Open Source Computer Vision Library (OpenCV)** [65]: This is an open source sofware library developed to provide support and accelerate computer vision applications. It is used by large companies like Google, Microsoft, Intel and IBM in multiple products. It can be used with Python, C++, MATLAB and is compatible with almost every operating system. This module will be used to handle data and to implement some methods methods if necessary (i.e the RPN methods shown in section 2.3). The OpenCV software itself is available under Apache 2 license and the python module that will be used (OpenCV-python) is available under MIT license.

- **TensorFlow** [66]: An open-source library used to develop ML models. It is compatible with different programming languages like C++, JavaScript, Python and others. It is compatible with the keras API and allows for parallelism (using multiple GPUs).

- **PyTorch** [67]: An open-source deep learning framework compatible with python, it is commonly used in image recognition and language processing. It is common choice for fast prototyping and experimentation.

- **Scikit-learn**: SciPy is an open source software project. It was developed with the goal of creating an open source software for scientific computing in Python. It was released under the BSD (or similar) open source license, developed openly and hosted on public GitHub repositories under the SciPy GitHub organization. This python framework simplifies the use of ML algorithms by providing easy to use implementations of the most common algorithms in various fields like supervised and unsupervised learning, clus-

tering, feature extraction, among others. This framework is also capable of delivering great performance since it works as a wrapper for optimized code written low-level languages (usually faster).

2.6 Discussion

We intend to provide a solution to perform re-identification on non-overlapping cameras. To provide a reliable solution we need to take into account both time and performance constraints. In this chapter we studied the different options available for the implementation regarding person detection and person re-identification. In addition we also describe the most used datasets for the task of re-identification as well as the most relevant technologies for the problem at hands.

Regarding person detection, we selected some of the most common object detection algorithms such as YOLO and Faster R-CNN. When analysing both algorithms we concluded that because Faster R-CNN providing a more thorough scanning of the images when compared to YOLO which leads to Faster R-CNN provides a more reliable way to find all the instances of people in a given image. However, because of the repeated scans in the images Faster R-CNN will be much slower. For this reason in the next phases of this project, both algorithms will be implemented, allowing us to study speed and performance in order to be able to choose the best option. We decided to not use the SSD algorithm since it was shown in [29] that YOLO performs much better.

Among the different algorithms found for person re-identification, we focused on the unsupervised algorithms because of the lack of labels in the real world scenario. Since we cannot guarantee the assumptions for the closed world-scenario established in section 2.4.1 we focused on open-world person re-identification algorithms: the ones using UDA techniques. This allows us to train our model using large, labelled, public datasets and then apply these models on the target domain. Of the available algorithms we decided to implement the SpCL and MEB-net. These algorithms were chosen not only for the performance shown in benchmark datasets but also because there are open source examples of the implementation of these algorithms.

When deciding on the tools we wanted to use to build our solution we first looked into what the authors of the selected algorithms were using, both authors used PyTorch to build the algorithms we want to implement (MEB-net and SpCL). Because of this and the fact that PyTorch is used for fast prototyping and experimenting and also allowed Graphics Processing unit (GPU) acceleration we decided it was the indicated tool to implement the solution regarding the person re-identification module. Because of the simplicity and support regarding image manipulation, we chose to use OpenCV to handle the person detection and image extraction.

Chapter 3

Methodology and Planning

In this chapter the methodology used for the development will be explained. The plans for the work in both semesters is also be presented here. This allows us to understand how the internship will function as well as establish clear goals. Considering the goals set it is possible to determine at any point if the work is on schedule or not.

3.1 Methodology

For the internship, we opted to use an agile-based methodology. Since this is a research project, this type of methodology is useful to accommodate the adaptations that might need to be made during the development [68]. Weekly meetings will be held between the intern and the project manager, which in this case is the advisor for the project. This meeting will be used to update the project manager on how the development progressed during the previous week and to devise a more concrete plan on the work for the next week. A representation of the described methodology is shown in figure 3.1. These meetings will also help in clarifying and solving some problems found during the development of the solution. Since the communication between the intern and the advisor is constant, there are several benefits to using this type of methodology. For example, the constant feedback will improve the quality of the final solution and prevent errors to persist throughout the development. The actors used in this adaptation of the agile methodology are listed below:

- **Product Owner:** Ubiwhere
- **Project Manager:** João Garcia
- **Developer:** Ricardo Martins

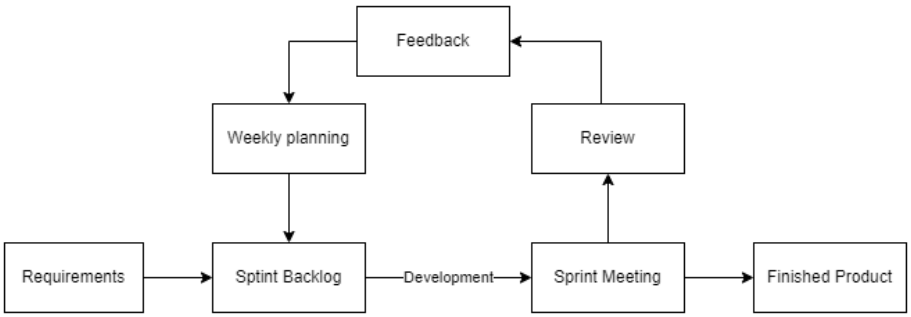


Figure 3.1: Representation of the adopted methodology.

3.2 Planning

Here we show the devised plan for the internship. By planning the tasks to be done for the different stages of the internship we are able to assess the state of the project. With the defined plan we are able to know if we are on schedule or not at any given time.

3.2.1 First Semester

Here we present the plan for the first semester of the internship. The planned tasks for this part of the internship were the literature in the subject of regarding person re-identification, the specification for the requirements of the final solution. We also defined the work methodology and planned the work for the second semester.

- **Report writing:** Document all the findings and relevant information for the development of the solution.
- **Literature review:** Study available solutions regarding the subject of person re-identification, the technologies used and available resources.
- **Requirement Specification:** Present the functional and non-functional requirements for the platform. This requirements need to be approved by Ubiwhere. This also includes the user stories.
- **Risk Analysis:** Make an analysis regarding internal and external factors that can compromise the development of the project and present solutions to mitigate these risks.
- **Plan for the second semester:** Present a plan for the work to be carried in the second semester.

Figure 3.2 shows the Gantt chart with the comparison of what was planned and what actually happened during the semester. Looking at the chart we can see that the main issue is related to the literature review task that was started much

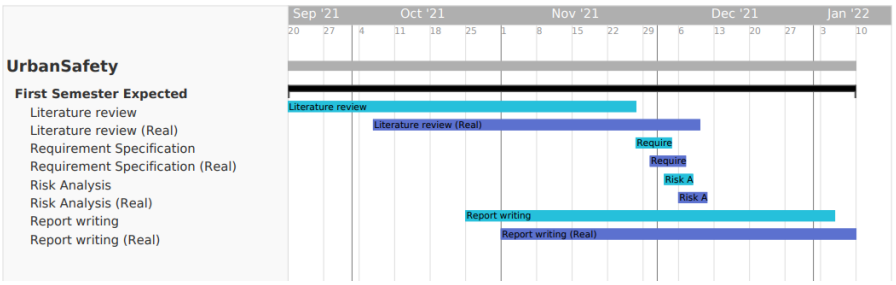


Figure 3.2: Gantt chart with the planned schedule and the real time frame for each task in the second semester.

later than predicted which resulted in the disruption of the rest of the plan. This delay was due to the several problems we encountered when trying to define our problem and goals for this project.

3.2.2 Second Semester

The work required for the second semester can be divided in two parts. The machine learning model development and testing and the writing of the final report.

- **Machine Learning:** In this step of the development different machine learning models will be applied. They have to be tested in order to find if they comply with the required performance defined in section 6.1, and if not apply changes or develop different models to achieve the goals previously set.
  - **Apply the models to the problem:** Apply the MEB-net and SpCL models to the solution.
  - **Evaluate the solutions:** Compare the performance of the different solutions in order to choose the best possible.
- **Report:** During the second semester we will need add eventual changes and include the details of the implementation and the tests made to the intermediate report. This task is reserved for the last month of the project.

Looking at Figure 3.3 we can Gantt chart with the planned schedule planed as well the real completion timeframe for each task. By looking at the chart we can see that two of the tasks stand out by failing the planned schedule. The dataset preparation and the experiment design and analysis. The dataset preparation took longer than expected because some of the datasets that we were planning on using were retracted for privacy issues and we took some time to find viable alternatives. This error in planning propagated through the rest of the tasks and was aggravated during the experiment design and analysis because training the models using 10 different seeds took longer than expected.

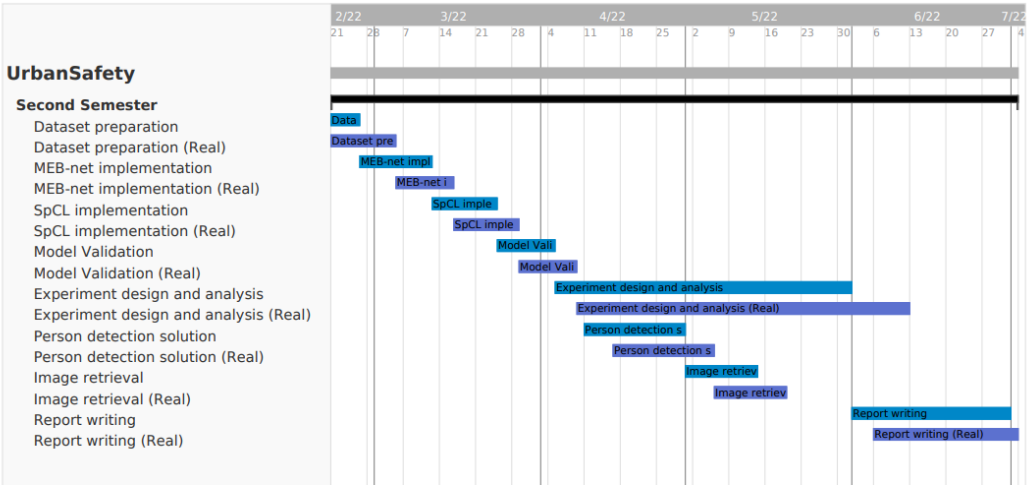


Figure 3.3: Gantt chart with the planned schedule and the real time frame for each task in the second semester.

Chapter 4

Approach

To tackle the problem of person re-identification in video footage we need to consider two critical steps: 1) the gathering of images from video surveillance and 2) re-identification of people considering a query. In this chapter it will be further discussed the options taken to the implementation of each step. All the parts of this project were built using Python and the different frameworks mentioned in section 2.5. Image 4.1**a)** shows how the implemented solution will work from the query selection to the system output regarding the person re-identification module, while image 4.1**b)** shows how we plan to extract the images of people from videos.

We start by selecting a query (an image of a person) then, using a person re-identification algorithm, our solution will search the gallery of images and find the positive matches available. The system will then output all the images that match the query. For the person re-identification task, we used the selected person re-identification algorithms SpCL (Self-paced Contrastive Learning) and MEB-net (Multiple Expert Brainstorming network). However, because of our goal was to perform person re-identification in videos, and these algorithm process images we had to find a solution to extract images of people from videos. To do this, we selected a frame for each second of the video, to minimize the time and resources required, and then use an object detection algorithm (YOLO or Faster R-CNN) to isolate all people present in each frame. In the following sections, some of the more complex steps will be further detailed.

4.1 Person detection and Image extraction

The main goal of this step to find and isolate the images of different people in the available video. A representative scheme of how we implement this step in our solution is shown in figure 4.1**a)**. To avoid creating massive datasets with long videos we decide to sample 1 frame from each second of the video. Then we apply the object detection algorithms on each of the extracted frames. The object detection algorithms give us a bounding box, a label and a confidence level for each of the objects found in the frame. With this we can then crop the bounding

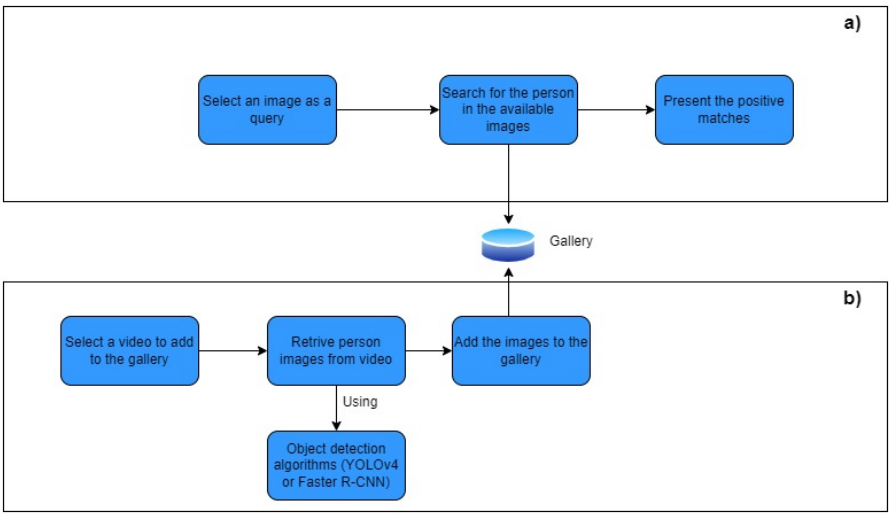


Figure 4.1: Diagram of the intended workflow of for the implemented the solution.

box from the image, resize it to 64x128 in order to match the input size of the re-identification models and save to the gallery all the detections labeled as person with a confidence level of 60% or higher. To restrain the amount of detected images even further we decide to apply Non-Maximum Suppression (NMS) to the detection algorithms. By using this we are able to reject bounding box proposals that severely overlap with each other. This allows us to keep the proposition with the highest confidence for each object and discard the others. Figure 4.2 shows an example of how this step of the solution works using real data. As it was shown in figure 4.1b) we receive a video file, select individual frames from that file and from each frame extract the instances of people that are detected using an object detection algorithm (In the example of figure 4.2).

For this task we implemented the YOLO and Faster R-CNN algorithms using PyTorch and OpenCV. Since these are well known algorithms for this task we were able to use configurations and weights of networks that were previously trained in the COCO dataset [69]. This allowed us to save the time required to training the models and still get a good performance. Later in this work we will compare both algorithms in the context of the problem considering the time of execution as the most relevant measure.

4.2 Person Re-Identification

As it was discussed in section 2.4 for person re-identification we selected the MEB-net and the SpCL algorithms based not only on their capabilities to handle data that is unknown at training time but also because of their performance, both in terms of CMC and mAP. We used the implementations provided by the authors, built using PyTorch and SciPy. In the case of the SpCL algorithm we used resnet50_ibn as a backbone since it provided the best results according to

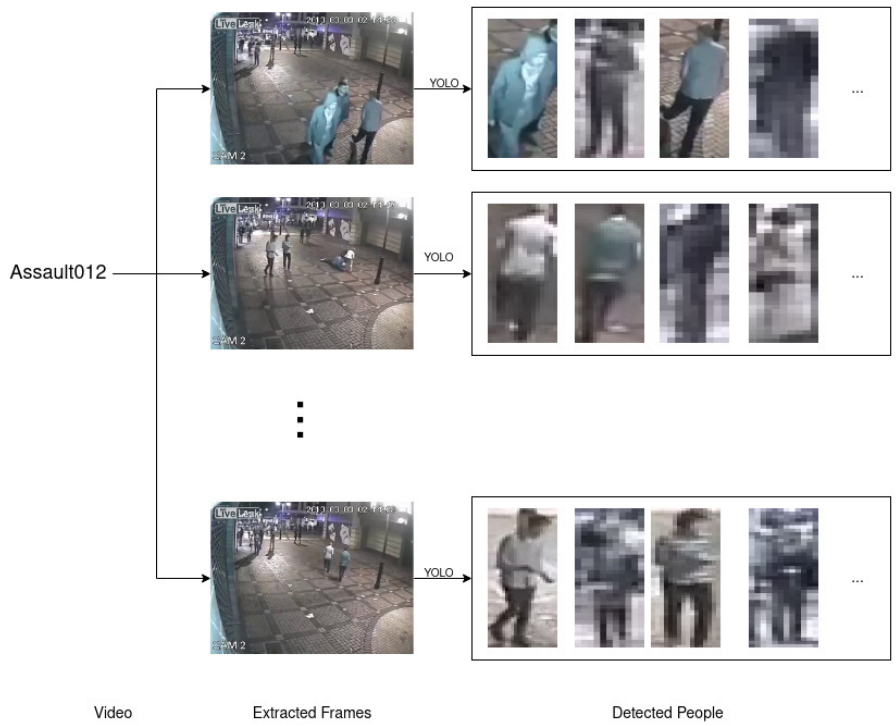
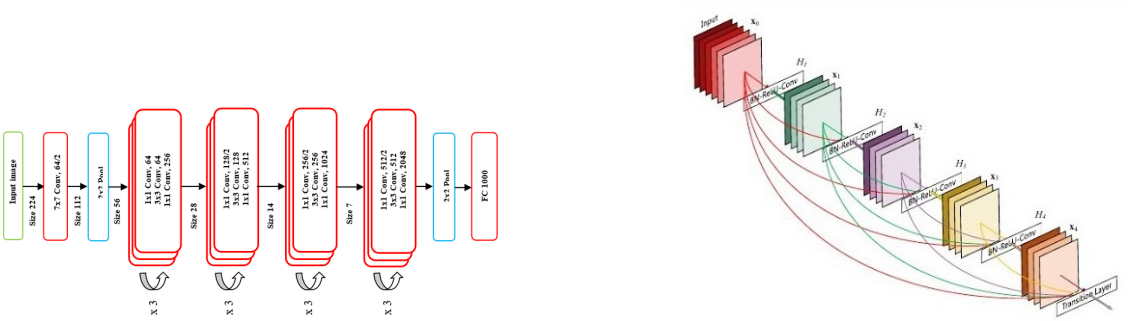


Figure 4.2: Example of the workflow of the platform to extract people images from videos using YOLO.

the original author [46]. In the MEB-net case we experimented different combinations using densenet and resnet50.

Resnet and densenet are usually used as backbones for other networks, which means they are used to perform feature extraction on the input and then feed that information to the rest of the network. This type of network introduced the concept of skip connections to tackle the problem of the vanishing gradient (performance deteoration when using several layers). The skip connections work as a shortcut between groups of layers (residual blocks) [17]. Resnet50 is a variation of these networks, where the number 50 means it has fifty layers following the architecture shown in figure 4.3a. We discuss the results of the different options in detail in section 5.4. Densenet also tries to alleviate the problem of vanishing gradient by strengthening feature propagation and encouraging feature reuse. This is done with the use of the dense blocks. Each dense block contains multiple layers that share the same feature map size, and each layer output is connected to the subsequent layers input. To ensure that the feature map size is reduced along the network, the downsampling of the feature maps is done in the transition layers (name given to the layers that connect the dense blocks) [70]. Figure 4.3b shows an example a densenet architecture.

The SpCL algorithm uses an CNN based approach that considers all the instances from both the source and target domain to improve the quality of the learning targets. This is done by using a hybrid memory that keeps the class centroids, and the unclustered instances. Every iteration the target data is updated while the source domain class centroids remain the same. Using a clustering algorithm,



(a) Architecture of resnet50 network [71]

(b) Architecture of densenet network [70]

Figure 4.3: Architectures of the different backbones used to train the person re-identification models. Resnet50 in 4.3a and Densenet in 4.3b.

e.g DBSCAN, allows us to update the class centroids and unclustered instances every iteration. This clustering algorithm that allows to control the maximum distances between instances in a cluster (we use the jaccard distance). With this we can use a constant *eps* to control the definition of reliable clusters by how close to each other their instances are. We then keep only the clusters that are below threshold and dismantle the others.

The MEB-net also has a CNN based architecture, however uses ensemble learning to perform person re-identification on the target domain. Because of this, we need to pre-train the "expert“ models in the source domain, here we used the densenet or resnet50. Each model performs feature extraction on the target dataset separately and organizes the data in clusters using the K-means algorithm and the euclidean distance to compute distances, where each cluster contains the images of a person. Then, after weight normalization, the models adapt to the target domain by iteratively going through the unlabelled images of the dataset. In each iteration, each of the models predicts pseudo-labels for the target samples by using the formed clusters. Then these pseudo-labels are used to fine-tune the expert networks by mutual learning, which allows the experts to share what they learned amongst each other. MEB-net also uses an algorithm to regulate the impact that each of the models has on the decision in re-identification. This algorithm gives "scores“ to each pre-trained model based on it’s capabilities at training time.

Figure 4.4 shows the representation of one execution of the person re-identification module. Figure 4.4a) shows the query sent to the system and 4.4b) and 4.4c) the output with the instances that were positively matched with the query. Because we wanted to see the discriminative capabilities of the algorithm we manually separated the results of the algorithm in the correct classifications 4.4b) and the incorrect classifications 4.4c). By doing this we are able to verify what the model is missing and what type of errors it is making.

Since our main goal was to retrieve the different images of the person used as query, we had to implement a function that allowed us to get the images that correspond to the positive matches. To be able to do this we simply kept a mapping between the inputs and the images in the gallery to be able to correctly identify

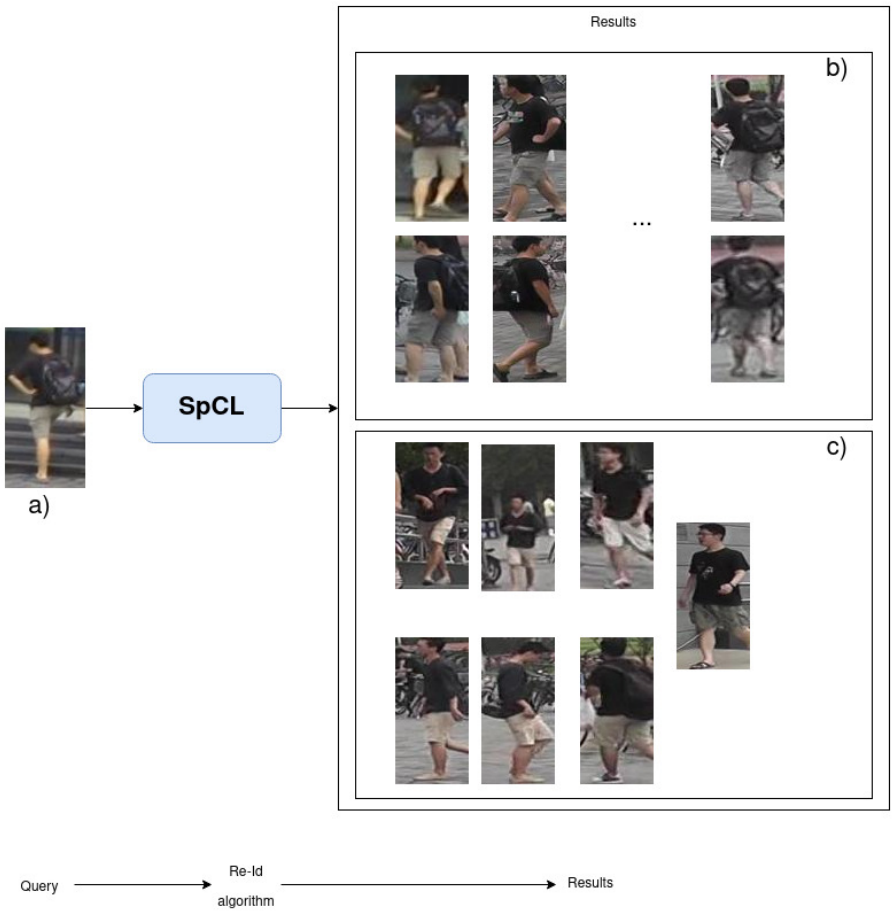


Figure 4.4: Example of the workflow of the module for person re-identification using SpCL.

the outputs. This functionality proved to be very useful to perform the validation of the model and to understand the results of the different models and their capabilities in person re-identification.

Chapter 5

Experimental Study

In this chapter we will perform an empirical study to compare the different options that were implemented for each part of the solution (person detection, image extraction and person re-identification) to select the most suitable configuration for the final solution. We also want to evaluate if these solutions can match the performance requirements established in section 6.1.

5.1 Datasets

To perform the analysis we selected four different public datasets. We use the UCF-crime dataset [72] to test the image extraction from video and the person detection. This dataset contains a total of 28 hours of recorded video captured at 30 frames per second from surveillance cameras, it is composed of 1900 shorter videos, captured in different situations. These clips are classified in 13 different classes related to public safety including abuse, arrest, assault, fighting, robbery, shooting, vandalism, among others. Since we only need the videos to test the capabilities of our solution regarding person detection and image extraction we can ignore the labels. We select 5 videos with varying lengths from this dataset (Assault008,Assault012,Assault022,Assault027 and Assault041). We use these videos to compare not only the performance in image extraction but also to understand the scalability of the implemented algorithms for the person detection and image extraction from video.

For the experiments regarding person re-identification we use the CUHK03, Market1501 and PersonX datasets. Table 5.1 shows how the different datasets are divided (training and testing).

Because we are using UDA, we use different datasets for training and testing. We selected Market1501 for testing, since it is more complex than the others and is the closest to the real world scenario, as it was shown previously, and the CUHK03 and PersonX are used for training. This means that at the training step we use CUHK03 and PersonX and the performance of the algorithm is measured on the testing split of the Market1501 dataset.

Name	Training Images	Training Identities	Testing Images	Testing Identities	Number of queries
CUHK03	7368	767	5328	700	1400
Market1501	12936	751	19732	751	3368
PersonX	14760	410	30816	856	856

Table 5.1: Details of the datasets used for the Person Re-Identfication.

5.2 Experimental setup

In order to decrease the time needed to execute the models, we used a GPU for training and testing, namely using a Nvidia RTX 2060. Since the experimental setup in [47] and [46] is different from the one we are using, we were unable to reproduce the results discussed described in precious works. We were using a single GPU, the memory available is limited therefore it was necessary to adjust batch size and to try to compensate by increasing the iterations and epochs. Another problem we faced was that some datasets that were used in other instances were retracted because of privacy issues.

To compare the methods for person detection and image extraction we use as performance measures the time the algorithm needs to extract all the detected people from the video frames, the number of instances identified for each video and we will also go manually verify the quality of the predictions since the dataset is not labelled. For these experiments we will not use statistical tests since the architectures of the used algorithms will remain constant as well as the frames selected, which means the result regarding the number of instances found will be the same, and the only expected variances will be small and will only happen when looking at the execution time of the algorithms. The default value for the confidence threshold is 60% and 40% for the NMS.

For the experiments of the person re-identification algorithms we validate the results by training all the models using 10 different random seeds. The results obtained are analysed using statistical tests to compare the different configurations of each algorithm and the different algorithms against each other. The training parameters for both algorithms YOLO and MEB-net are shown in Table 5.2. Regarding these algorithms we are interested in comparing the different implementations in person re-identification performance (mAP and CMC), the time each different architecture needs to perform the re-identification task and with the goals established in section 6.1.

For the statistical tests we only use non-parametric tests (Wilcoxon Signed Rank test to compare two samples and Kruskall-Wallis test to compare three or more samples), even when the data follows a normal distribution. This helps in mitigating the fact that the sample size is small and using these tests also enables us to have more conservative results. Since all the tests were done using the same set of random seeds we always used paired tests. We selected a significance level of $\alpha = 0.05$ for all the tests, and when multiple pairs are compared we use the bonferroni correction to ajust the p_{value}. Our main goal is to find if there are sig-

Default Parameters	value
batch-size	64
dropout	0
learning rate	0.00035
weight decay	$5e^{-4}$
iterations	500
step-size	20
training dataset	CUHK03
target dataset	market1501
SpCL parameters	
epochs	60
k1 (jaccard distance hyperparameter)	30
k2 (jaccard distance hyperparameter)	6
maximum neighbor distance	0.6
momentum	0.2
eps	0.02
MEB-net parameters	
number of clusters	500
alpha	0.999
momentum	0.9
epochs	80(pre-train)/40(target train)
iterations	200(pre-train)/8000(target train)
soft-ce-weight	0.5
soft-tri-weight	0.8

Table 5.2: Training parameters for the SpCL and MEB-net models.

nificant statistical differences in the performance of the algorithms used for person re-identification. For each statistical test we need to consider two hypothesis. The null hypothesis(H0) that states that there are no differences in the samples.

5.3 Person detection and Image Extraction

In this experiment we compare the two most common methods for object detection (YOLO and Faster R-CNN). Since we are not training the networks from scratch we will not perform hyperparameter tuning and will only compare the results of both algorithms to extract the images of each person in the video. Since we were unable to gather video using Ubiwhere's smart lamppost we used a video sample available from the dataset [72]. As it was detailed in section 4.1 we will retrieve sample frames from each second of the video, use the different algorithms on all the selected frames and compare the times, number of people found and the quality of the images with visual inspection. We selected 5 videos of different lengths to be able to understand how our solution scale with the increase in the number of images to process when considering the two different algorithms.

We compared the time each of the implemented algorithms needed to extract the

images from all the frames. In this experiment we use the default parameters for NMS and for the confidence interval in classification established in section 5.2. Figure 5.1 shows a representation of the results, we can also see with this experiment how each algorithm's performance reacts to the increase in the number of frames.

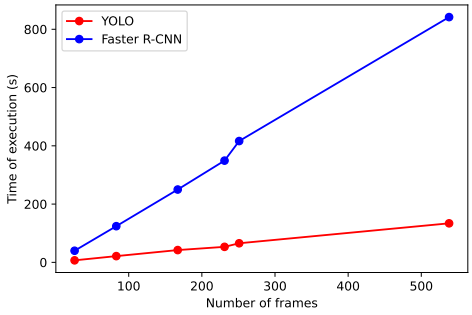


Figure 5.1: Diagram showing how the models are trained for this problem.

Considering the results shown in Figure 5.1 we can confirm that, using YOLO to extract people from images is not only much faster than using the Faster R-CNN but it also scales much better with the increase of images we need to scan.

In addition to the algorithm's execution time, we also compared the YOLO and Faster R-CNN on how many instances each algorithm can extract from the provided video frames. For a more complete comparison we also study how changing of the values for the confidence threshold for classification affects the performance of the different algorithms, regarding the number of instances identified. By increasing the confidence threshold we are expecting the number of identities to decrease since images with poor quality will end up being discarded. Figure 5.3 shows the comparison of the algorithms using different values for the different confidence thresholds. We were expecting the decrease in instances found to be more accentuated when using the Faster R-CNN, because this algorithm does a more thorough scan of the image, which can result in the identification of more instances at low quality than when using YOLO. However if we look at Figure 5.3 we see that when we increase the confidence level both algorithms seem to decrease the detected instances at a similar rate. Despite of the clear superiority of the Faster R-CNN when it comes detecting people, we confirmed through visual inspection that a lot of the extra instances detected by the Faster R-CNN algorithm are people in the background. Which becomes a problem since their detection box is very small. When transformed to the required size of the re-identification algorithms (64x128) it loses quality, making it unusable for the task of person re-identification.

5.4 Person Re-identification

Both SpCL and MEB-net models for person re-identification were implemented and improved through hyperparameter tuning. We want to test the different configurations of each algorithm and then compare them. With these experiments

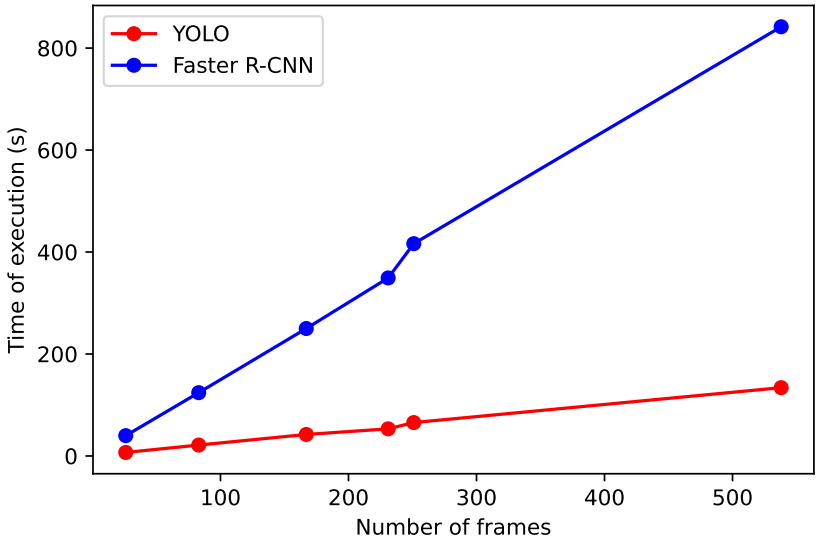


Figure 5.2: Execution time for the person detection algorithms considering the number of frames.

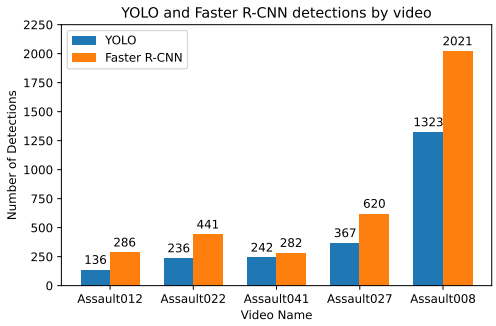
we will be able to make a decision on which algorithm is the most suitable for the problem. Figure 5.4 shows the pipeline used to train the models for these experiments. We use CHUCK03 to train the models and the perform the validation and tests using Market1501.

5.4.1 Self-paced Contrastive Learning

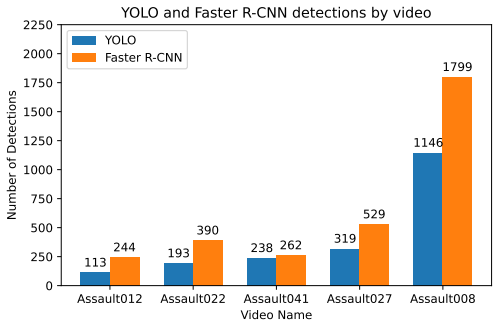
Most of the training hyperparameters for the SpCL model were previously studied in the work where the approach was originally proposed [46]. Because of the time constraints we decided to keep the parameters that showed the best performance in the scenarios tested in [46]. The training parameters for these experiments are described in Table 5.2.

The first experiment we did with this algorithm to study the influence of the chosen dataset used to train the model. To evaluate the impact of the chosen dataset we compared the performance of the algorithm when trained with an artificial dataset (PersonX [59]) and a real dataset (CUHCK03 [60]) and then evaluated on the target dataset (Market1501 [62]). We want to compare the results achieved with both datasets with regards to the CMC and the mAP, shown in Figure 5.5.

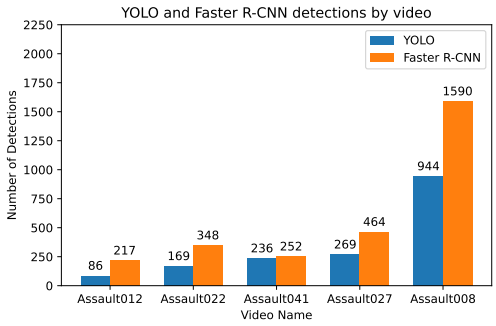
With this experiment we want to see if there are significant differences in the performance of SpCL with the two different datasets. To perform the statistical test described in section 5.2 we define the null hypothesis (H0) as: "The median of the population of differences between the paired data is zero“. We expect the performance of the SpCL to be better when using CUHK03 since it is a larger dataset and is closer to the used target dataset (Market1501). Therefore we formulate our hypothesis (H1) as: "The median of the population of differences between using CHUK03 and PersonX is greater than zero“.



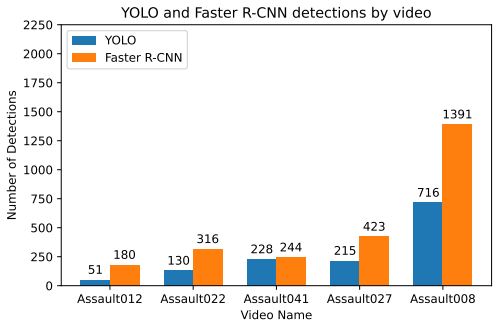
(a) Confidence threshold of 50%.



(b) Confidence threshold of 60%.



(c) Confidence threshold of 70%.



(d) Confidence threshold of 80%.

Figure 5.3: Number of person instances each algorithm detected in the selected videos considering different confidence thresholds.

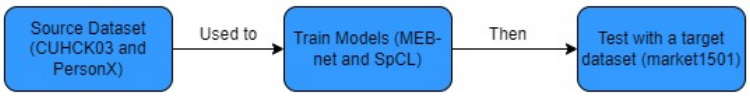
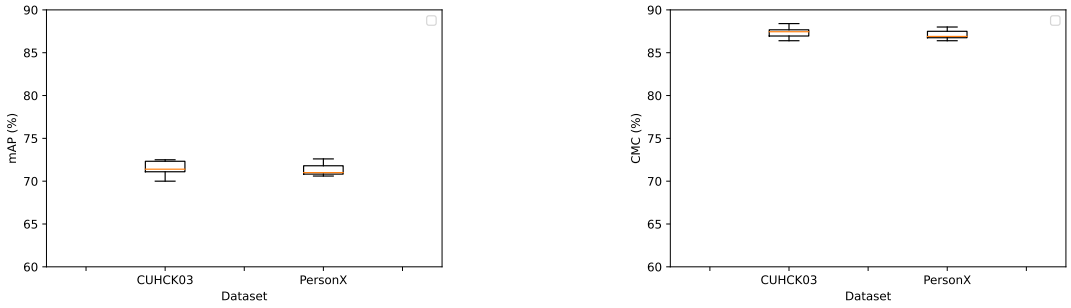


Figure 5.4: Diagram showing how the models are trained for this problem.

When applying the Wilcoxon test to compare the distributions of the results regarding mAP we get a $p_{value} = 0.278$. Since $p_{value} > \alpha$ we cannot reject the null hypothesis, meaning that both samples follow similar distributions and there are no significant statistical differences between using either dataset. When comparing the results regarding the CMC we followed the same procedure getting from the wilcoxon test a $p_{pvalue} = 0.200$, and since $p_{value} > \alpha$ we cannot reject the null hypothesis either. In conclusion, with this experiment we were able to verify that there are no significant differences in performance when using either of the selected datasets as source domain.

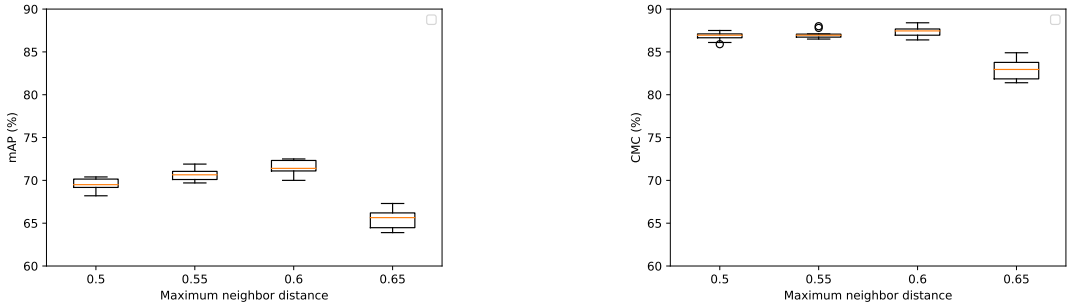
Even though we were able to achieve similar results, considering the similarity of the CUHCK03 dataset with the dataset that would be used in the real scenario and the fact that the average of both mAP and CMC is slightly higher we will be using this dataset as source domain for the rest of the experiments. Another reason to use CUHK03 is that the PersonX can only be used for research purposes.

For the second experiment performed on the parameters of the SpCL algorithm we changed the parameters of the clustering algorithm. More specifically, we



(a) mAP results. (b) CMC results.

Figure 5.5: Performance comparison of SpCL when using different datasets for training, regarding mAP and CMC.



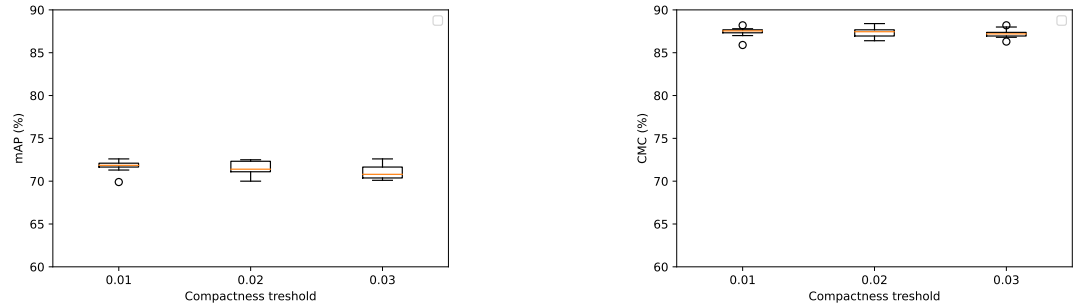
(a) mAP results. (b) CMC results.

Figure 5.6: Performance comparison of SpCL considering different maximum neighbor distances in clusters for the DBSCAN algorithm.

studied the influence that maximum neighbor distance has on the performance of the algorithm, considering mAP and CMC as performance metrics. Changing this parameter allows us to have control on how inclusive a cluster can be. While decreasing the maximum distance can limit the number of false positives the false negatives will likely increase, since we make the clusters more inclusive. So with this experiment we want to optimize the trade-off in classification by adjusting the value for the maximum distance.

We compare the performance of the SpCL algorithm using different values for the maximum cluster distance (0.5, 0.55, 0.6 and 0,65). Figure 5.6 shows the results from the experiments conducted done with each different configuration.

We want to verify if there are statistically significant differences in the algorithm's performance when using different values for maximum neighbour distance. Because of this we have H0: "The population medians are equal.“, and H1: "The population medians are different". When we perform the Kruskal-Wallis test to compare the performance in mAP we get a $p_{value} = 0.0000007$, and when we compare the results in CMC we get $p_{value} = 0.000023$. In both cases $p_{value} < \alpha$ so we reject the null hypothesis, i.e, there are statistical significant differences in the performance of the SpCL algorithm when using different values for maximum neighbor distance.



(a) mAP results.

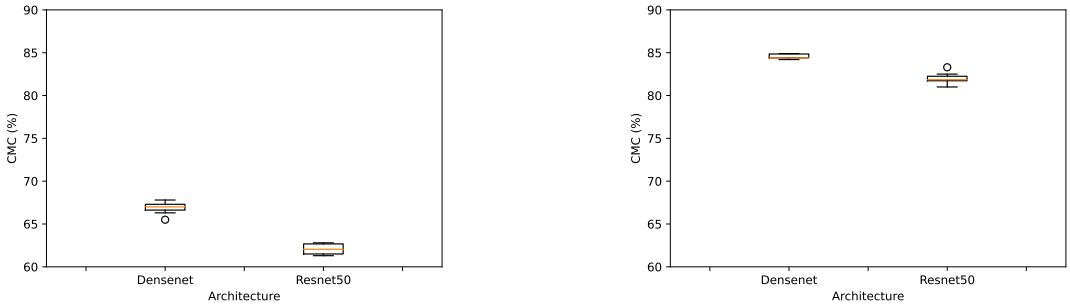
(b) CMC results.

Figure 5.7: Performance comparison of SpCL considering different thresholds to consider a cluster compact.

To find the best value for the maximum neighbor distance we will have to compare every combination of values (0.5, 0.55, 0.6, 0.65) individually. Using the bonferroni correction we got a new $p_{value} = 0.008$. With the results of the pairwise comparison, regarding the performance measured for mAP, we were able to conclude that there are no real statistical differences when using 0.55 or 0.6 but there are differences among other pairs. We found that using a maximum neighbor distance of 0.5 provided better results than using a distance of 0.65, however using 0.55 or 0.6 creates the better results. When comparing the performance in CMC we concluded that there is no statistical difference in results when using a maximum neighbor distance of 0.5, 0.55 or 0.6. However, using 0.65 for this parameter yields worse results than the other values tested.

Another property of the SpCL is that it creates a measure for cluster reliability. One of the parameters that is considered for reliability is the compactness of a cluster. So in this experiment we want to evaluate how changing the threshold of distance to consider a cluster compactness influences the performance of the algorithm in the re-identification task. For this we set a value d that is used to control to loosen or tighten the clustering criterion. This is done by subtracting (tighter) or adding (looser) the value of d to the max distance between neighbors set for the cluster. In this experiment we experiment with following values for d: 0.01,0.02 and 0.03. The results of the tests performed can be seen in Figure 5.7

Just like in the previous experiment, we use the Kruskal-Wallis test to verify if there are significant statistical differences in the medians of the performance results when using different values for d. When applying the test regarding the mAP we get $p_{value} = 0.245$, and when applying the Kruskal-Wallis test to the results regarding the performance in CMC we have a $p_{value} = 0.477$. Since in both cases we have $p_{value} < \alpha$ we cannot reject H0, meaning that there are no significant statistical differences in the medians of the populations regarding either performance metric. This result is surprising especially when considering that the results shown in Figure 5.7a, where the differences between $d = 0.01$ and $d = 0.03$ seem accentuated, if we discard the outlier.



(a) mAP results.

(b) CMC results.

Figure 5.8: Performance comparison of SpCL considering different maximum neighbor distances in clusters for the DBSCAN algorithm.

5.4.2 Multiple Expert Brainstorming network

To test this algorithm we also used the CUHCH03 dataset. For the hyperparmeters for the training step we used the values established in [47] and shown in Table 5.2. Since this method relies on pre-trained networks on the source dataset we studied the impact of using different architectures. This influences the time required for execution as well as the performance. The architectures we tested as backbones are Densenet and Resnet50. Figure 5.8 shows the results achieved on all the tests using different random seeds.

To compare the performances of the different architectures we use the Wilcoxon Signed Rank Test. With this test we will confirm if there are any significant statistical differences in the medians of the population. So we consider H0: "There are no statistical differences in the medians of the population“. Because we expect densenet to outperform the resnet50 [73], we consider H1: "The median of the population of differences between using CHUK03 and PersonX is greater than zero“.

By performing the Wilcoxon test we get a $p_{value} = 0.0009$ when comparing the results regarding mAP and a $p_{value} = 0.001$ when comparing the CMC. Since in both comparisons we get a $p_{value} < 0.05$ we reject H0 on a significance level of $\alpha = 0.05$. Meaning we can say that the differences in the medians of the population are significant and using the MEB-net with densenets as backbone should yield better results than when using the resnet50.

We also wanted to compare the time for execution required by the different architectures (MEB-net with resnet50, MEB-net with densenet and SpCL). To perform this comparison we evaluated the time each architecture took to perform the re-identification task, considering a query size of 1, and a varying number of images in the gallery. Figure 5.9 shows the results of this test. We can see that all three architectures have a similar perform this re-identification task, however the SpCL seems to be slightly faster when the gallery size reaches higher values. We can also see that the time for execution seems to show a linear correlation with the gallery size for all the architectures. This is due to the fact that in the re-identification task most of the time is spent in feature extraction from the images

available in the gallery.

Since the performance of the MEB-net is inferior when compared to the SpCL, we decided not to pursue this algorithm any further.

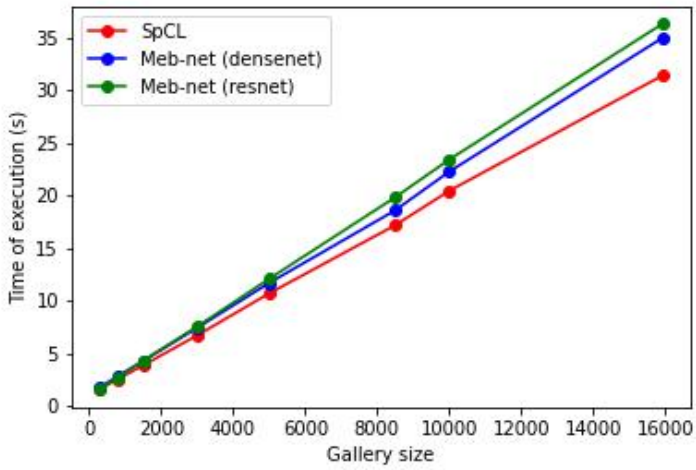


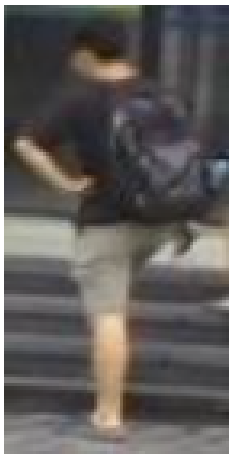
Figure 5.9: Execution time for the person re-identification algorithms considering the size of the gallery.

Finally to better understand our person re-identification algorithm, we wanted to find out what instances were being classified correctly and incorrectly. More specifically we focused on the true positives, false positives and false negatives. To visualize this we sent one query to our algorithm and then manually analyzed the results. This experiment is shown in Figure 5.10. We divided the results into true positives Figure 5.10b and false positives Figure 5.10c. As we can see the false positives are very similar to the query, similar clothes and carrying a backpack. However in addition to this we decided to search for the false negatives, (Figure 5.11 shows some examples). If we look at Figure 5.11a we see that it is a cropped image, Figure 5.11b shows a completely different perspective from the query. When comparing these false negatives to the query (Figure 4.4a)) we can see that our algorithm still has room for improvement for example handling images with occluded body parts Figure 5.11a and slight perspective changes Figure 5.11c. In the case of Figure 5.11a it is hard to handle since we have a completely different perspective of the person, the backpack is not seen and the pose is different making it difficult to accurately classify as a positive match.

5.4.3 Summary

With the experiments we have done in this chapter we were able to take several conclusions that allow us to choose the best algorithms with the best parameters to propose a proper solution for the problem of person re-identification in video surveillance.

Regarding the person detection algorithm we were able to verify that while the YOLO is much faster (Figure 5.1), the Faster R-CNN is much more capable since

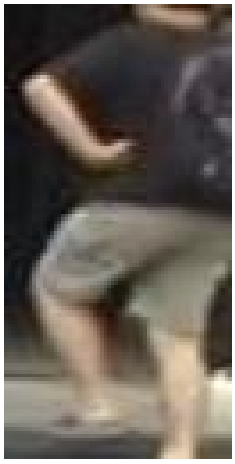


(a) Query used for the person re-identification algorithm.

(b) Examples of true positives in the results with SpCL.

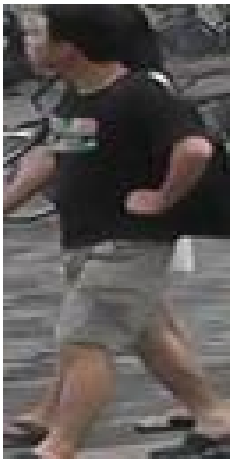
(c) False positives in the results with SpCL.

Figure 5.10: Results using SpCL for person re-identification using a single query.



(a)

(b)



(c)

Figure 5.11: False negative examples from a test using SpCL.

it is able to detect more people than the YOLO in the same set of images (Figure 5.3). Because of this we had to choose what was our priority speed or detection performance. We decided to prioritize speed in this case to be able to comply with the time requirements, and also because the difference in the number of detections is not accentuated in all situations. We also studied the impact of the confidence threshold for the classification of a detection as a person. We saw that both algorithms decreased the number of detections when we increase the value for the classification. We were expecting that by increasing the value of this threshold we could reduce the difference of the number of people detected by both algorithms, that was not the case. To gather more information we manually verified the outputs of both algorithms manually. By doing this we were able to understand that the Faster R-CNN was identifying people that were further away from the camera that the YOLO was not. However our goal is to use the extracted images of people for person re-identification, and some of those extra images gathered with Faster R-CNN will not have the necessary quality to be used in the person re-identification task. This detail paired with the extreme advantage that the YOLO algorithm provides in execution time we opted to use YOLO in our final solution. We were also able to verify that using a confidence threshold between 60% and 70% for YOLO would be a compromise between the number of identities detected and the quality of the images.

When it comes to the person re-identification we performed several experiments on the implemented algorithms in order to extract the maximum performance from them. We trained the SpCL model with an artificial dataset (PersonX) and a regular dataset (CUHK03), and found no significant advantages in either one (Figure 5.5). Then we changed the properties of the clustering algorithm of the SpCL (DBSCAN). With this we were able to understand the impact that the distance between the cluster instances has on performance by changing the values of the maximum neighbor distance on the clusters. We concluded, from the results shown in Figure 5.6, that we could optimize the algorithm to avoid false positives by decreasing the maximum neighbor distance allowed and decrease the false negatives by increasing the maximum distance allowed. After testing we ultimately considered the trade-off between false positives and false negatives and decided that based on the results of the tests the best value for our setup would be 0.6. We also changed the cluster compactness during the training phase to control the definition of a reliable cluster. This was done by using different values for d. Considering the results shown in Figure 5.7 we decided the bet value would be $d = 0.01$. Finally we tried to use the MEB-net, we tested the backbone architectures that provided the better results in [47] (densenet and resnet50). The results shown in figure 5.8 show that using the densenet provides much better results, however it is still too far from the results we were able to get from SpCL. Finally we compared performance regarding time of execution on the three implemented architectures for person re-identification (MEB-net with resnet50 and densenet and the SpCL). The results in Figure 5.9 show us that the required time for each of the architectures is very similar, however, when the size of the gallery starts to increase to large numbers the SpCL starts to have a small advantage.

In conclusion, based on the results of the tests that were performed in this chapter we propose a solution that uses YOLO for person detection and image extraction

with a confidence threshold between 60% and 70%. For person re-identification we will use SpCL trained with CUHK03, considering a maximum neighbor distance of 0.6 and the interval for compactness $d = 0.01$. With this solution we achieved, on average, 71.6% in mAP and 87.5% in CMC, while classifying 8500 images in under 20s. This means we were able to satisfy the performance requirements for both person re-identification (70% in mAP and 80% in CMC) and time (7500 images in less than 30s). Even though we were not able to provide a clear answer to the scalability of the system, we are expecting that the resources required are proportional to the increase in images and cameras based on the results shown in Figure 5.1 and Figure 5.9.

Chapter 6

Conclusion

Based on the increasing number of challenges that public security faces, especially in an urban environment, we provide a module that can be used in Ubiwhere's solutions to leverage the capabilities of ML to provide help to safety authorities using person re-identification methods. We designed and analysed different possible solutions solutions and chose the best one to be integrated with Ubiwhere's Urban Platform and Smart Lamppost.

During the research made on state-of-the-art methods used in person re-identification, we directed our focus on the characteristics of our specific problem. The most important constraints we have are: people are not marked with bounding boxes, we have unlabelled data and we cannot guarantee that every person appears more than once in the dataset. Because of this we use open-world unsupervised methods, more specifically UDA algorithms. From the methods that we analysed, we selected two algorithms (SpCL and MEB-net) that stood out from the rest in terms of classification performance. Because these methods are applied on images, and our product should work with video we also performed a study on relevant object detection algorithms to be able to detect people.

With this work we propose a framework that allows us to use person re-identification algorithms that focus on images, using videos as inputs. To be able to do this we divided our work in two critical steps: person identification and image extraction, and person re-identification. To extract the images of the people from video we created a function to select one frame from each second from the video, and then apply the object detection algorithms to find the people present in each frame. This allows us to get the desired result while optimizing the required space, since it is likely that if one person is in one frame it will also appear in the other frames of the same second.

We were able to test how each part of the implemented system works, however, because we were unable to use video from the smart lamppost we could not validate the solution on a fully real scenario and were only able to test the different components isolated from each other using different public datasets for testing (UCF-crime for person detection and Market1501 for person re-identification).

We performed experiments and comparisons between the selected algorithms implemented for both person detection and person re-identification. Regarding person detection we confirmed that using YOLO was much faster than using Faster R-CNN, however, YOLO falls short on detection performance when compared to the Faster R-CNN. However considering the performance/speed trade-off and the quality of the images extracted we decided that for our problem using YOLO will be the best solution. Through our experiments we also saw that the SpCL has better performance in terms of person re-identification and execution speed when compared to the MEB-net. In conclusion we provide a solution for the problem of person re-identification in video surveillance using YOLO for person detection and image extraction with a confidence threshold between 60% and 70%, and using SpCL considering a maximum neighbor distance of 0.6 and the interval for compactness $d = 0.01$. With this solution we were able to get, on average, 71.6% in mAP and 87.5% in CMC, while classifying 8500 images in under 20s.

6.1 Future Work

As for future work we can divide our aspirations in two different categories, the improvement of the re-identification algorithm and the improvement of the framework as a product.

When it comes to the person re-identification task, several approaches are left to experiment, test and improve. One of the paths we can explore is the improvement of training data through the use pre-processing and data augmentation techniques such as using Generative Adversarial Networks to improve the datasets by creating, new poses, improving the quality of the images or simply by creating more instances to the dataset. We can also improve the classification abilities by adding allowing a human to intervene through Active learning. Another development for the future is the creation and use of a dataset gathered with Ubiwhere's smart lamppost. By creating this dataset we can guarantee the training, and testing on the intended environment. With this we can truly optimize the algorithms for the data and guarantee the quality of the training data.

To further develop the framework developed in this work we can add several adjustments, such as improving usability in query selection allowing for a easier use of the platform. We can also improve this solution so that it can analyse video on near real-time. Transforming this solution to allow it to work on real-time will not only make the tool more useful but will also allow us to optimize the resources required, since we would no longer need to store the images captured permanently.

References

[1] DCAF Geneva Centre for Security Sector Governance. Urban safety and security. 2019.

[2] Violence and harassment across europe much higher than official records. *European Union Agency for Fundamental Rights*, Feb 2019.

[3] The 17 goals | sustainable development.

[4] Richard Van Noorden. The ethical questions that haunt facial-recognition research. *Nature*, 587(7834):354–358, November 2020.

[5] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[6] Titus Neupert, Mark H Fischer, Eliska Greplova, Kenny Choo, and Michael Denner. Introduction to machine learning for the sciences. 2 2021.

[7] Iqbal H. Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2, 5 2021.

[8] Taiwo Oladipupo Ayodele. Types of machine learning algorithms. 2010.

[9] Chris Albon. *Machine learning with python cookbook: Practical solutions from preprocessing to deep learning*. " O'Reilly Media, Inc.", 2018.

[10] Feature scaling: Standardization vs normalization, Aug 2021.

[11] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre GR Day, Clint Richardson, Charles K Fisher, and David J Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics reports*, 810:1–124, 2019.

[12] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Model Assessment and Selection*, pages 219–259. Springer New York, New York, NY, 2009.

[13] Mang Ye, Jianbing Shen, Senior Member, Gaojie Lin, Tao Xiang, Ling Shao, and Steven C H Hoi. Deep Learning for Person Re-identification : A Survey and Outlook.

[14] Jonathan Hui. Map (mean average precision) for object detection, Apr 2019.

[15] Rishabh Chandaliya. Tele Stroke System for Stroke Detection TeleStrokeSystem for Stroke Detection by Rishabh Ajitkumar Chandaliya This thesis has been submitted in partial fulfillment for the. (July), 2020.

[16] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:770–778, 2016.

[18] Christopher Thomas BSc Hons. MIAP. An introduction to convolutional neural networks, May 2019.

[19] Sinam Ajitkumar Singh, Takhellambam Gautam Meitei, and Swanirbhar Majumder. 6 - short pcg classification based on deep learning. In Basant Agarwal, Valentina Emilia Balas, Lakhmi C. Jain, Ramesh Chandra Poonia, and Manisha, editors, *Deep Learning Techniques for Biomedical and Health Informatics*, pages 141–164. Academic Press, 2020.

[20] I. Poletaev, Konstantin Pervunin, and M. Tokarev. Artificial neural network for bubbles pattern recognition on the images. *Journal of Physics Conference Series*, 754:(072002)–13, 10 2016.

[21] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016.

[22] B. Mehlig. Machine learning with neural networks. jan 2019.

[23] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[24] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*, 2014.

[25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[26] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. YOLOv1. *Cvpr*, 2016-Decem:779–788, 2016.

[27] Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:6517–6525, 2017.

[28] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. 2018.

[29] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020.

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[31] How single-shot detector (ssd) works? `https://developers.arcgis.com/python/guide/how-ssd-works/`. Accessed: 2022-06-20.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[33] Ross Girshick. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 International Conference on Computer Vision, ICCV 2015:1440–1448, 2015.

[34] Ross Girshic Georgia Gkioxari, Piotr Dollar. Mask R-CNN slides. *GEO: connexion*, 15(1):28–29, 2018.

[35] Xuan Zhang, Hao Luo, Xing Fan, Weilai Xiang, Yixiao Sun, Qiqi Xiao, Wei Jiang, Chi Zhang, and Jian Sun. AlignedReID: Surpassing Human-Level Performance in Person Re-Identification. 2017.

[36] Ehsan Yaghoubi, Aruna Kumar, and Hugo Proença. Sss-pr: A short survey of surveys in person re-identification. *Pattern Recognit. Lett.*, 143:50–57, 2021.

[37] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.

[38] Zheng Wang, Zhixiang Wang, Yinqiang Zheng, Yang Wu, Wenjun Zeng, and Shin'ichi Satoh. Beyond intra-modality: A survey of heterogeneous person re-identification. *arXiv preprint arXiv:1905.10048*, 2019.

[39] Albert Haque, Alexandre Alahi, and Li Fei-Fei. Recurrent attention models for depth-based person identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1229–1238, 2016.

[40] Nikolaos Karianakis, Zicheng Liu, Yinpeng Chen, and Stefano Soatto. Reinforced temporal attention and split-rate transfer for depth-based person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 715–733, 2018.

[41] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. 4 2016.

[42] Hao Liu, Jiashi Feng, Zequn Jie, Karlekar Jayashree, Bo Zhao, Meibin Qi, Jianguo Jiang, and Shuicheng Yan. Neural person search machines. 7 2017.

[43] Xu Lan, Hanxiao Wang, Shaogang Gong, and Xiatian Zhu. Deep reinforcement learning attention selection for person re-identification. *British Machine Vision Conference 2017, BMVC 2017*, (Table 1):1–14, 2017.

[44] Yichao Yan, Qiang Zhang, Bingbing Ni, Wendong Zhang, Minghao Xu, and Xiaokang Yang. Learning context graph for person search. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:2153–2162, 2019.

[45] Andy J. Ma, Pong C. Yuen, and Jiawei Li.

[46] Yixiao Ge, Feng Zhu, Dapeng Chen, Rui Zhao, and Hongsheng Li. Self-paced contrastive learning with hybrid memory for domain adaptive object re-id.

[47] Yunpeng Zhai, Qixiang Ye, Shijian Lu, Mengxi Jia, Rongrong Ji, and Yonghong Tian. Multiple expert brainstorming for domain adaptive person re-identification.

[48] Yixiao Ge, Dapeng Chen, and Hongsheng Li. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. *arXiv preprint arXiv:2001.01526*, 2020.

[49] Yang Fu, Yunchao Wei, Guanshuo Wang, Yuqian Zhou, Honghui Shi, and Thomas S. Huang. 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.

[50] Kaiwei Zeng, Munan Ning, Yaohua Wang, and Yang Guo. Hierarchical clustering with hard-batch triplet loss for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13657–13665, 2020.

[51] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Instance-guided context rendering for cross-domain person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 232–242, 2019.

[52] Wei-Shi Zheng, Xiang Li, Tao Xiang, Shengcai Liao, Jianhuang Lai, and Shaogang Gong. Partial person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4678–4686, 2015.

[53] Lingxiao He, Jian Liang, Haiqing Li, and Zhenan Sun. Deep spatial feature reconstruction for partial person re-identification: Alignment-free approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7073–7082, 2018.

[54] Yifan Sun, Qin Xu, Yali Li, Chi Zhang, Yikang Li, Shengjin Wang, and Jian Sun. Perceive where to focus: Learning visibility-aware part-level features for partial person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 393–402, 2019.

[55] Yang Fu, Xiaoyang Wang, Yunchao Wei, and Thomas Huang. Sta: Spatial-temporal attention for large-scale video-based person re-identification. 11 2018.

[56] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In Defense of the Triplet Loss for Person Re-Identification. 2017.

[57] Rahul Rama Varior, Bing Shuai, Jiwen Lu, Dong Xu, and Gang Wang. A siamese long short-term memory architecture for human re-identification. In *European conference on computer vision*, pages 135–153. Springer, 2016.

[58] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 994–1003, 2018.

[59] Xiaoxiao Sun and Liang Zheng. Dissecting person re-identification from the viewpoint of viewpoint. In *CVPR*, 2019.

[60] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 152–159, 2014.

[61] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE international conference on computer vision*, pages 3754–3762, 2017.

[62] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.

[63] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 79–88, 2018.

[64] Mengran Gou, Ziyan Wu, Angels Rates-Borras, Octavia Camps, Richard J Radke, et al. A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets. *IEEE transactions on pattern analysis and machine intelligence*, 41(3):523–536, 2018.

[65] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[66] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[67] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style,

high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[68] Laura Pirro. How agile project management can work for your research. *Nature*, April 2019.

[69] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[70] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[71] Bendjillali Ridha Ilyas, Moh Beladgham, Khaled Merit, and Abdelmalik taleb ahmed. Illumination-robust face recognition based on deep convolutional neural networks architectures. Vol 18:1015 1027, 12 2019.

[72] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488, 2018.

[73] Chaoning Zhang, Philipp Benz, Dawit Mureja Argaw, Seokju Lee, Junsik Kim, Francois Rameau, Jean-Charles Bazin, and In So Kweon. Resnet or densenet? introducing dense shortcuts to resnet. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3550–3559, 2021.

Appendices

Requirements

In this chapter, we present the functional and non-functional requirements specified for the solution. Even though these requirements are aligned with the needs and expectations of the company, they are not final and can change over the course of the development.

Functional Requirements

For this project, it was decided to use user stories in order to define the requirements of the solution. By using small sentences with non-technical language, user stories are great for defining requirements which will help the developer know what features to implement and to what end.

The system only has one type of actor. We define this actor as the **user**. It acts as an administrator, who interacts with the service through the Urban Platform. This user has to provide the query for the system.

Since the main goal of the project is not the platform but the algorithm itself, only the essential were features for the platform were specified as it follows.

User stories

User Story 1

- **Description:** As a user I want to be able to manually select the different queries for the system.
- **Acceptance Criteria:**
  - **Scenario:** The user creates a bounding box.
    - **Given:** The video is currently stopped.
    - **When:** The bounding box is drawn.
    - **Then:** The user gets a message acknowledging that the system is searching.
- **Dependencies:** User story 1, User story 2;
- **Priority:** Must have.

This requirement is essential since we need to be able to choose the person or people we want to search for.

User Story 2

- **Description:** As a user I want to get the results of the selected query to see the desired information.

- **Acceptance Criteria:**
  - **Scenario:** Occurrences for the selected query were found.
    - **Given:** A valid query was provided.
    - **When:** The user searches for occurrences in the system.
    - **Then:** The user receives the images the occurrences of the selected query in the system and the cameras that spotted them.
  - **Scenario:** There are no results for the selected query.
    - **Given:** A valid was provided.
    - **When:** The user searches for occurrences in the system.
    - **Then:** The user gets an message with the information of no results.
- **Dependencies:** User story 4;
- **Priority:** Must have.

By showing the images found by the re-identification algorithm we allow the user to confirm the results, reducing the impact of false positives.

User Story 3

- **Description:** As a user I want to be able to consider only a specific time interval of the videos to be able to narrow the search.
- **Acceptance Criteria:**
  - **Scenario:** The user chooses the time interval to search.
    - **Given:** The interval is valid.
    - **When:** The system is queried.
    - **Then:** The user gets a message that the time is valid.
  - **Scenario:** The user chooses the time interval to search.
    - **Given:** The interval is invalid.
    - **When:** The system is queried.
    - **Then:** The user gets a message showing that the selected time is invalid.
- **Dependencies:** User story 1, User story 2;
- **Priority:** Nice to have.

If we have knowledge of the time frame that we are interested, we could narrow the gallery size improving the speed of the execution for he person re-identification algorithm.

Non-functional requirements

The non-functional requirements are used to specify the quality of the implemented solution desired by the company. In this case, the specified requirements concern mainly performance. We consider the system scalability as a "nice-to-have" requirement, therefore it is not crucial for the success of the project.

- **Performance** - This specifies how well the solution must perform in different scenarios with different measures like mAP, accuracy and running time.
- **Scalability** - This solution is to be implemented on different scenarios where the camera setup will vary, specially in number. Considering this, if possible, make a solution that adapts to scenarios with a high number of cameras.

To evaluate performance, there are two main components. The classification precision and accuracy measured by the mean average precision and the rank-1 accuracy respectively, and the time needed to perform the search and classification. Regarding classification performance, it is expected that the solution achieves at least a 70% mean average precision and 80% rank-1 accuracy. As for how long the classification task should take, in a scenario with two cameras and a time interval of 2 minutes (7500 examples) the classification should take no longer than 30 seconds.

For the scalability factor, it is required that the system can accommodate dynamically added cameras and, by adding cameras, the developed algorithm execution time will not increase exponentially.

Risk Analysis

There are risk associated with the development of a software product. These risks can threaten the success of the project, so by analysing and planning on how to mitigate them beforehand, we are able to create contingency plans and have a better chance to avoid the certain consequences. After identifying the risks, Table 1 shows an analysis for probability and impact for each of the risks taken into consideration.

Risk 1

Lack of computational resources and time to train properly and test all the desired models properly. This would cause not being able to test all the proposed algorithms and variations.

Mitigation plan: Start developing the machine learning algorithms before the rest of the work. This would give more time to train and test different models. Assess with Ubiwhere available resources as soon as possible.

Risk Analysis		
Risk	Impact	Probability
1	Catastrophic	Moderate
2	Catastrophic	Low
3	Moderate	Low
4	High	High

Table 1: Analysis of every risk according to probability and impact.

Risk 2

The datasets could not be representative enough of the desired situations for testing, and the desired test dataset might not be ready on time. This would make it difficult to evaluate the proposed solutions.

Mitigation plan: Find alternative public datasets that both the developer and the company accept as valid representations of the desired scenarios.

Risk 3

The performance requirements can be too ambitious, since the dataset quality has a heavy influence on the performance of the algorithms.

Mitigation plan: Do proper research on the results achieved by the methods that will be used in order to get a general estimation of the probable outcome of the experiments.

Risk 4

Even though there are multiple studies on how the algorithms perform in terms of quality, there is not enough information to have accurate estimations on expected times of execution of the different algorithms in the testing scenarios.

Mitigation plan: To mitigate this risk, we need to be prepared to adjust the requirements in order to accommodate the trade-off between speed and quality of classification.