UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

MASTER THESIS

OBJECT DETECTION AND ASSOCIATION IN MULTIVIEW SCENARIOS BASED ON DEEP LEARNING

Master's Degree in ICT Research and Innovation

Image Processing and Computer Vision Program

Paula Moral de Eusebio
Director: Álvaro García Martín
Supervisor: Juan Carlos San Miguel Avedillo

July 2019

OBJECT DETECTION AND ASSOCIATION IN MULTIVIEW SCENARIOS BASED ON DEEP LEARNING

Paula Moral de Eusebio
Director: Álvaro García Martín
Supervisor: Juan Carlos San Miguel Avedillo

Video Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
July 2019

This work has been partially supported by the Spanish Government
TEC2017-88169-R (MobiNetVideo)

Resumen

La detección y asociación de objetos en escenarios multivista es un área de investigación dentro de la Visión Artificial que resulta de gran utilidad en tareas como la de videovigilancia, por ejemplo en el caso de querer identificar en distintas escenas a una persona que ha realizado algún tipo de anomalía. Este proyecto se va a centrar en la re-identificación de vehículos, ya que es un problema de plena actualidad en los Sistemas de Transporte Inteligente (ITS, por sus siglas en inglés) y podemos ver nuestro rendimiento en comparación con algoritmos del estado del arte participando en el 2019 NVIDIA AI City Challenge .

El objetivo principal de este Trabajo Fin de Máster es el desarrollo de un sistema de detección y asociación de múltiples objetos en escenarios multivista basado en *deep learning*. Para ello, se ha realizado el estudio de distintos algoritmos ya existentes en el estado del arte y se ha implementado un método que, usando métricas de aprendizaje y las características extraídas de las imágenes, devuelve una lista con las posibles coincidencias entre el objeto a buscar y las imágenes de la galería ordenadas según la distancia entre ambos. Se ha creado un nuevo *dataset* reorganizando la parte de *train* del *dataset* CityFlow-ReID. Para mejorar los resultados, se ha introducido la combinación de distancias y *ranks* obtenidos con distintos métodos para extraer características y métricas de aprendizaje. Finalmente se ha desarrollado un entorno de evaluación para analizar el rendimiento del sistema propuesto.

Palabras clave

Re-identificación de vehículos, extracción de características, métricas de aprendizaje, redes neuronales convolucionales, marco de evaluación.

Abstract

The detection and association of objects in multiview scenarios is an area of research within the Computer Vision that is very useful in tasks such as video surveillance, for example when identifying in different scenes a person who has carried out any kind of anomaly. This project is going to focus on vehicle re-identification, due to it has been a critical problem in the Intelligent Transportation System (ITS) for the recent years, and we can see our performance compared with the state of the art participating in the 2019 NVIDIA AI City Challenge.

The main objective of this Master Thesis is the development of a system that detects and associates multiple objects in multiview scenarios based on deep learning. For this task, different algorithms from the state of the art have been studied. Furthermore, the method implemented uses feature extraction methods and metric learning techniques and it returns a list with all the matches between the query object and the images from the gallery set, sorted according to their distance. A new dataset has been reorganized from the train part of the CityFlow-ReID dataset. In order to improve the results, it is included a metric network combination at distances and ranks level from the different feature extraction methods and metric learning techniques. Finally, we have developed our own experimental setup.

Keywords

Vehicle re-identification, features extraction, metric learning, convolutional neural networks, experimental setup.

Acknowledgements

Quería aprovechar este aparatado para dar las gracias a mi tutor, Álvaro García, por ofrecerme la oportunidad de realizar este trabajo y prestarme toda su ayuda y su tiempo durante estos meses.

Quería también agradecer a mi familia el apoyo y las facilidades que me han brindado siempre, sin los cuales no estaría aquí ahora. Por último, darles las gracias a todas las personas que me han acompañado durante estos dos años de máster sin los cuales no habrían sido lo mismo.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Motivation

Object detection and association is a relevant task in Computer Vision such that allows to re-identificate the same object identity across different cameras. This problem aims to compare an identity from a query camera view versus all the gallery test candidates from different camera views. If there is a match to the query in the test gallery, it should have a higher rank compared to other candidates on the test set.

There are several algorithms in the literature that deal with person re-identification problem. Some of these method are studied in depth in Chapter 2.

This thesis is focused in vehicle re-identification because it is one of the challenging tasks in order to make transportation system smarter and safer. Some of the challenges present in this area are the small variability between different vehicles model taken by the same camera view against the large variability of the same vehicle identity from different viewing angles. Also the the lack of annotations, the illumination conditions and the low video resolution are issues present in re-identification task. Contrary to the vast majority of vehicle re-identification algorithms, we are going to avoid using the license plate for privacy reasons although this information would be very useful for vehicle re-identification. The dataset used is provided by the 2019 NVIDIA AI City Challenge [1], that focus on Intelligent Transportation System (ITS) problems, including vehicle re-identification. We are going to participate in this challenge in order to see our performance compared with the actual state of the art.

A baseline re-identification system with different algorithms from the state of the art is proposed. The system is developed with the motivation of introduce different improvements and compare them with the baseline methods included.

In this project we use some different feature extraction methods and metric learning algorithms from the state of the art that work with the bounding boxes of the different vehicles. Then, it is necessary to obtain the distances and rank the matches between the query vehicles with the test gallery. Moreover, it is also proposed some deep learning methods as feature embedding representation, with the improvement that they are adapted to the specific task of vehicle re-identification.

1.2. Objectives

This Master Thesis is focused on object detection and association in multiview scenarios based on deep learning. The main objective is to develop a system which re-identificates objects, specifically vehicles, across multiple cameras.

The work can be divided in the following goals:

1. Review the relevant work in the literature related with Multi-camera object detection and association.
2. Integrate the system Person Re-identification Benchmark [2] in order to adapt it to different datasets of object re-identification. This Benchmark proposes different feature extraction and metric learning techniques.
3. Include feature extraction techniques adapted to the specific vehicle re-identification task.
4. The improvement of including a metric network combination at distances level, rank aggregation and adding video tracking information.
5. The participation in the 2019 NVIDIA AI City Challenge, that pursue perform vehicle re-identification based on vehicle images from multiple cameras from urban environments.
6. Propose an experimental setup in order to compare the different methods used.

1.3. Document Structure

The structure of the document is the following:

- Chapter 1: Introduction. This chapter introduces the work and presents the motivation and the objectives of this Master Thesis.
- Chapter 2: State Of The Art. This chapter presents an overview of the literature related to the work presented in this Master Thesis.

- Chapter 3: Design and Development of the proposed method. This chapter presents the development of the system and the improvements included.
- Chapter 4: Evaluation. This chapter presents the comparative evaluation results.
- Chapter 5: Conclusions and Future Work. This chapter summarizes the main achievements of the work, discusses the obtained results and provides suggestions for future work.
- Bibliography.

Chapter 2

State Of The Art

2.1. Introduction

In this chapter, the literature related to re-identification and the different feature extraction and metric learning techniques are studied. It is divided into the following sections: an introduction of object detection and association tasks, the study of the Person Re-identification Benchmark [2], the different features extraction methods and metric learning techniques, the rank methods present in the state of the art and lastly, the description of the 2019 NVIDIA AI City Challenge [1].

Object re-identification aims to detect and associate the same object which appears in different camera views. Re-identification has some challenging tasks, for example, the entire process is not always performed under the same illumination, viewpoint and background conditions. The problem seeks to compare from a query camera view a specific identity with all the test candidates from different camera views, i.e., test gallery. If the system identifies a match to the query in the gallery, it should have a higher rank compared to other candidates on the test set that are not the same object.

Person re-identification has become one of the most studied fields in the re-identification area. The large number of researches working in person re-identification problem implies a great development of those methods which could be generalized to any object re-identification task.

The re-identification problem needs the previous object of interest detection. Some datasets include the annotated objects detection. On the other hand, there are datasets where the given information are the entire scene, and it is necessary to apply an automatic detection algorithm. This project is focused on the re-identification task, so the dataset used have the hand-crafted bounding boxes of the objects of

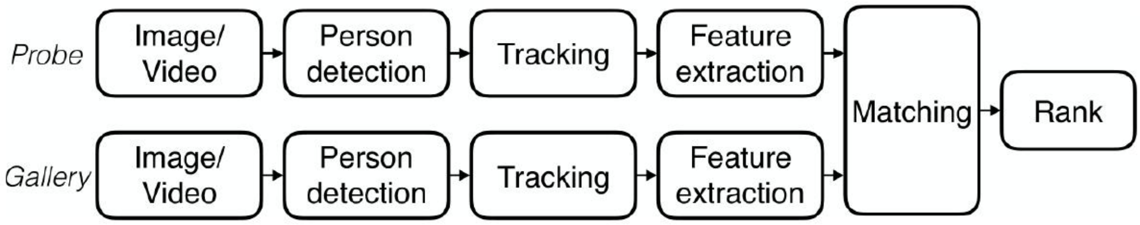


Figure 2.1: End-to-end re-identification pipeline from [2].

interest, and it will be not necessary to apply a detection algorithm.

As it is mentioned in the Introduction 1, this work is focused on vehicle re-identification. Some of the challenges present in this task are the small inter-class variability between different vehicles of the same model and the same point of view, the large intra-class variability of the same car from different points of view and the low video resolution [1].

2.2. Person Re-identification Benchmark

The Person Re-identification Benchmark [2] presents a systematic evaluation of the most popular person re-identification algorithms from the literature with the most recent advances. The re-identification system is divided in the feature extraction step, the metric learning technique and rank algorithm applied. The benchmark is evaluated on a wide variety of datasets. It gives us a huge previous knowledge due to the large number of techniques for feature extraction and metric learning that they use.

It compares a person of interest seen in a “probe” camera view (also known as query) to a “gallery” set of candidates. To consider as a re-identification, it is necessary that the probe camera and the gallery camera are not the same one. If there is a true match to the probe in the gallery, it should have a higher matching score compared to the incorrect candidates.

In Figure 2.1 the end-to-end of a person re-identification pipeline is shown. In this scheme, the person detection and tracking algorithms are extracted on-the-fly, and it may result errors in bounding boxes that not represent a person. The authors include a large-scale dataset with images recorded in a surveillance camera network from an airport. They use the ACF [3] detector and a combination of FAST corner features [4] and the KLT [5] tracker to extract the bounding boxes with the person.

The public available datasets that the system uses in its evaluation are: iLIDSVID [6], VIPeR [7], HDA+ [8],CAVIAR [9], WARD [10], 3DPeS [11], PRID [12], SAIVT-

SoftBio [13], CUHK01 [14], CHUK02 [15], CUHK03 [16], DukeMTMC4ReID [17], GRID [18], V47 [19], RAiD [20] and Market1501 [21]. These are person re-identification datasets with different characteristics and organization structures. For instance, VIPeR contains 632 image pairs (632 pedestrians identities), in which each pair of the same person was taken from different viewpoint and illumination conditions.

The feature extraction schemes used in this benchmark are: multi-scale biologically-inspired features encoded using covariance descriptors(gBiCov) [22], Ensemble of Localized Features (ELF) [7], color histograms and SIFT features extracted from each patch (DenseColorSIFT) [23], Local Descriptors encoded by Fisher Vectors (LDFV) [24], color and texture histograms from local binary patterns (HistLBP) [25], local maximal occurrence (LOMO) [26], Weighted Histograms of Overlapping Strips (WHOS) [27], hierarchical gaussian descriptor (GOG) [28] and the convolutional neural networks trained for this classification objective: AlexNet [29], ResNet [30] and VGGNet [31]. According to the results provided by the benchmark [2], we study from all the aforementioned feature extraction methods only those with better results. In Section 2.3 the methods used in this system are explained more in detail.

Furthermore, the benchmark [2] incorporates supervision using training data to learn a feature space. In this space, the feature vectors of a person are closer than those from different people. The metric learning methods used are: Cross-view Quadratic Discriminant Analysis (XQDA) [26], Fisher Discriminant Analysis (FDA) [32], Local Fisher Discriminant Analysis (LFDA) [33] and its kernelize version (KLFDA) [25], discriminative null space learning (NFST) [34], Marginal Fisher Analysis (MFA) [35] and its kernelize variant (KMFA) [25], Large Margin Nearest Neighbour (LMNN) [36], Information-Theoric Metric Learning (ITML) [37], Probabilistic Relative Distance Comparison (PRDC) [38], Keep-it-simple-and-straightforward (KISSME) [39] and finally, Pairwise Constrained Components Analysis (PCCA) [40] and its kernelize version (KPCCA) [40]. Due to the benchmark [2] includes a detailed results comparison, we only have selected and studied the metrics which return higher performances. In Section 2.4 the metrics used in our system will be explained in more detail.

2.3. Feature extraction methods

The Person Re-identification Benchmark includes the evaluation of all the feature extraction methods mentioned in Section 2.2. According to these results, the features extraction methods with higher performance have been selected for this project.

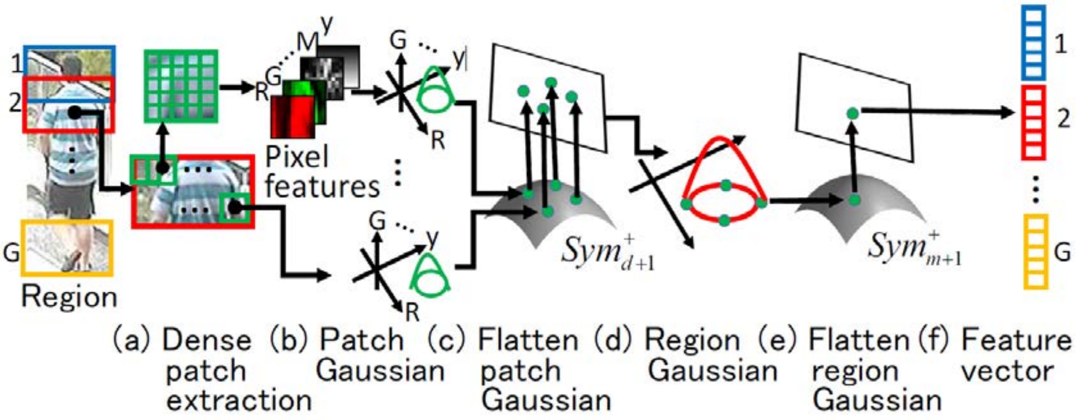


Figure 2.2: GOG descriptor, from [28]: (a) Local patches for each region. (b) Each local patch is described with a Gaussian distribution of pixel features. (c) Each patch Gaussian is flattened and vectorized by considering the underlining geometry of Gaussians. (d) The patch Gaussians inside a region are summarized into a region Gaussian. (e) Flatten the region Gaussian and extract the feature vector. (f) Finally, the feature vectors extracted from all regions are concatenated into one vector.

2.3.1. Gaussian of Gaussian (GOG)

Gaussian of Gaussian descriptor [28] provides a simple and consistent hierarchical method to generate robust and discriminative features. These features describe information related with texture and color.

The images are modeled as a set of multiple Gaussian distributions. Each Gaussian represents the appearance of a local patch (horizontal strips) of the image. Then, the local patches are organized by sets of patch Gaussians. The characteristics of this sets are described by another Gaussian distribution. Then, in order to represent a region, it is used the parameters of this Gaussian of Gaussian as a feature vector. The graphic explanation of the descriptor performance is shown in Figure 2.2.

2.3.2. Weighted Histograms of Overlapping Strips (WHOS)

Weighted Histograms of Overlapping Strips [27] is a descriptor of appearance based on coarse and striped pooling of local features. It segments foreground from background using a center support kernel. The descriptor process is explain in Figure 2.3. First, the image is scaled to a fixed size. Then, a spatial pyramid is built by splitting the image into overlapping horizontal stripes. The RGB histograms and Hue-Saturation (HS) are extracted from each region. To calculate the contribution of each pixel to its corresponding histogram bin, it is used the Epanechnikov kernel centred on the image. A Histogram of Oriented Gradient (HOG) [41] descriptor is

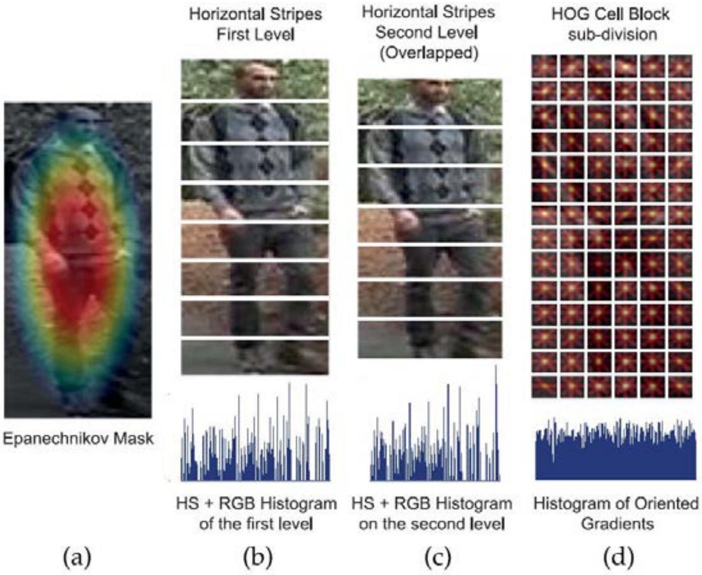


Figure 2.3: WHOS descriptor, from [27]: (a) An Epanechnikov kernel weights the contribution of each pixel to HS and RGB histograms computed on overlapping stripes (b) and (c). Overlapping HOG descriptors are concatenated with these (d).

concatenated to the HS and RGB histograms.

The advantages that the method provides are, among others: pose invariance (due to the horizontal and overlapping strips), illumination changes invariance (due to color HS histograms) and discriminative color information (due to RGB histograms).

2.3.3. Convolutional neural networks

In Person Re-identification Benchmark [2], the Convolutional Neural Networks (CNNs) treat each person as a class. The architectures used in the Benchmark are AlexNet [29], Resnet [30] and VGGNet [31], that are pre-trained on ImageNet [42] dataset and fine-tuned using the person datasets mentioned in Section 2.2. In our project, the architectures chosen because of their relevance in scene and object classification are AlexNet [29], ResNet-18 [30], ResNet-50 [30], ResNet-101 [30], DenseNet-201 [43] and Inception-ResNet-v2 [44].

AlexNet [29]: This deep convolutional neural network was developed and trained in 1.2 million high-resolution images of ImageNet [42] in order to classify 1000 classes. The architecture contains 60 million parameters and 650,000 neurons. It also has eight layers with weights, five are convolutional and the other three are fully-connected layers. The final fully-connected layer is followed by a 1000-way softmax, that gives the distribution of the 1000 classes. The architecture scheme is shown in Figure 2.4.

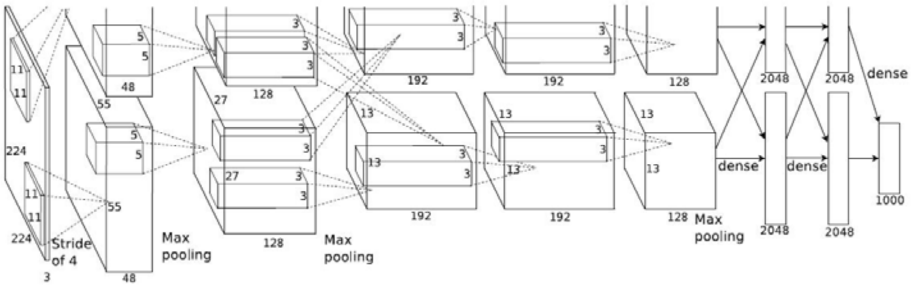


Figure 2.4: AlexNet architecture from [29].

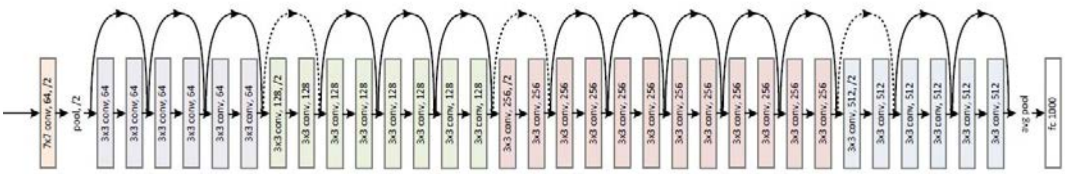


Figure 2.5: ResNet architecture from [30] for a ResNet-34.

ResNet [30]: The residual network was created in order to facilitate the training step of deep networks. ResNet uses the layers as learning residual functions with references to the first layers, instead of learning functions without reference. Each layer has 3x3 filters and follow two rules. The first rule says that if the feature map size is the same, then the number of filters in the layer is the equal. The second rule says that if the number of filters increase, the feature map size has to decrease in order to preserve the time complexity per layer. According to the number of weighted layers, we have ResNet-18, ResNet-50 and Resnet-101. The architecture scheme is shown for the specific case of ResNet-34 in Figure 2.5.

DenseNet-201 [43]: Dense Convolutional networks use the fact that if CNNs contain feed-forward connections between layers close to the input and close to the output, they can be more efficient, accurate and deeper. This network proposes a connectivity model that adds direct connections between any layer to all the following layers. The DenseNet architecture starts with convolution and pooling layer followed by four dense blocks and transition layers. After a final dense block, it is included a global average pooling followed by a softmax classifier. The architecture scheme is shown in Figure 2.6.

Inception-ResNet-v2 [44]: Inception-ResNets are deep convolutional networks that achieve a good performance with a low computational cost. The architecture of

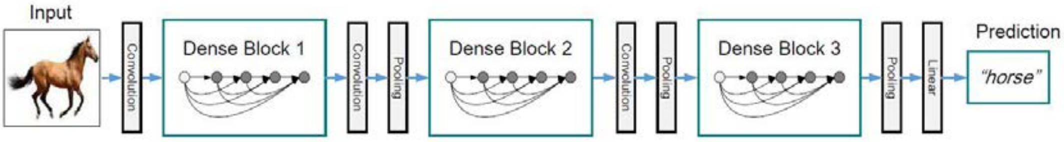


Figure 2.6: DenseNet architecture from [43].

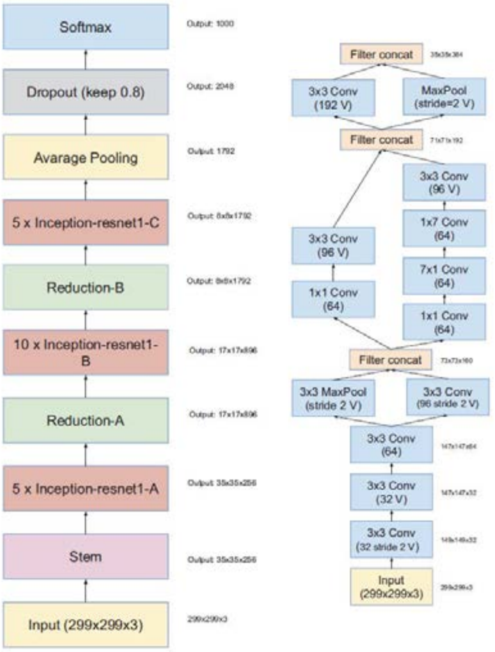


Figure 2.7: Inception-ResNet-v2 architecture from [44]. On the left is the scheme of the specific stem of Inception-ResNet-v2.

these CNNs allows many variants and tunable parameters. In the case of Inception-ResNet-v2 the different layers that appear are the stem, the convolutional layers, max pooling layers, inceptions layers, reduction layers, average pooling, droput and a final softmax. The architecture scheme is shown in Figure 2.7.

2.4. Metric learning techniques

In this section, the metrics with higher performance in the benchmark will be explained. The metric learning that allows us to learn the feature spaces should be discriminant enough to match various object images. The objective is to learn the optimal distance metric (see Figure 2.8), that means small values to the targets of the same person and large values to those of different people.

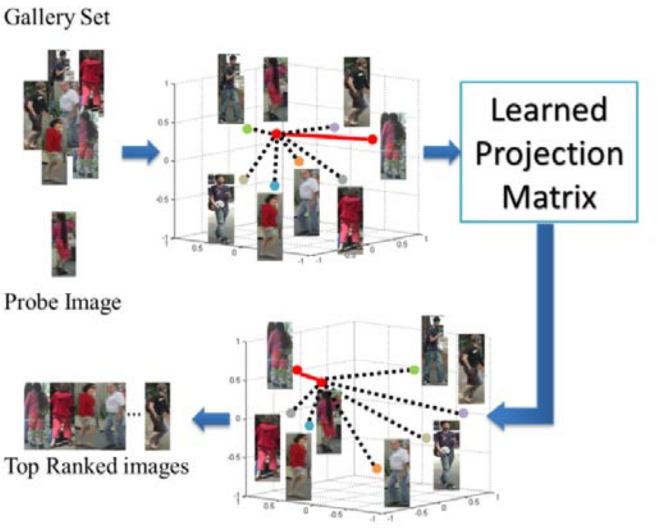


Figure 2.8: Learned projection matrix extracted from [25] where the features from the same individual are closer than features from different ones.

One of the problems present in distance metric learning is the small number of training samples compared with the high number of feature representation. It is necessary a huge number of feature representation in order to be robust against condition changes between images from different cameras views, like illumination changes, pose variation, view angle and background clutter. But the number of training samples is smaller due to the difficulties in the process of obtain the matched training images. This is called the Small Sample Size problem (SSS) [45].

Metric learning methods aim to minimize the intra-class variance whilst maximizing the inter-class variance. With a small sample size, the inter-class scatter matrix becomes singular and to avoid it, it is neccesary to reduce the dimensionality.

2.4.1. XQDA

This metric learning method, called Cross-view Quadratic Discriminant Analysis (XQDA) [26], is derived from KISSME and Bayesian Face methods used in cross view metric learning. It is based on learn a discriminant subspace with low dimension using cross-view quadratic discriminant analysis and, at the same time, learn a distance function using QDA metric on the subspace.

2.4.2. NFST

A discriminative null space learning presented in [34] is an improvement of the original null Foley-Sammon transfer method [46], also known as the null space methods. NFST [34] proposes that images of a same person must be projected to a single point in a new space by a transform, instead of minimizing the intra-class variance.

NFST raises to learn a discriminative null space of the training data. The authors also developed a semi-supervised learning method in the null space to take advantage of the high number of unlabelled data to deal with the effects of the small sample size problem.

2.4.3. KLFDA

Kernel Local Fisher Discriminant Analysis [25] is a feature kernel-based distance learning approach. It improves the classification accuracy when the data space is undersampled. KLFDA uses a kernel to deal with feature vectors of high dimension at the same time that maximizes a Fischer optimization criteria.

The principal problem that tends to avoid KLFDA using kernel approach based on supervised dimensionality reduction is obtaining features that overfit the data when the dimensionality reduction step is performed, due to the small size of datasets.

The method calculates a projection matrix that maximizes the inter-class variance while minimizes the intra-class variance using the Fisher discriminant objective. This method allows to choose different kernels in order to increase the accuracy.

2.5. Ranking

Person Re-identification Benchmark [2] proposes two evaluation schemes, single-shot and multi-shot.

Single-shot problem extracts the features from a single probe image of the identity person, while multi-shot problem extracts the features from a set of probe images, and not only one. In order to deal with multi-shot data, the average feature vector for each person would be computed, but also several algorithms like AHISD [47] and RNP [48]. This algorithms could be used to treat the image set and compute the distance between the probe and all the gallery set. In baseline single-shot evaluation scheme, the Euclidean distance is computed to determine rank, and for superior performance, the metric learnings techniques are included. In our system the single-shot scheme will be the one used according to the dataset used for the vehicle re-identification task.

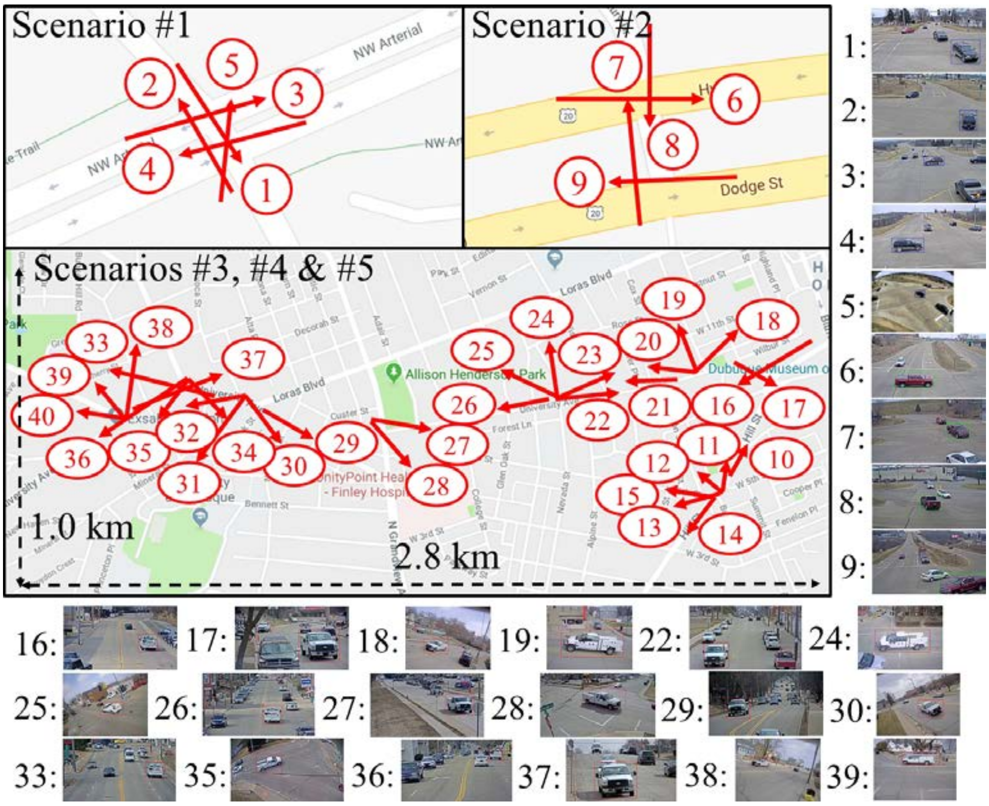


Figure 2.9: Scheme extracted from [1]. It shows the camera distribution and the street scenario of the CityFlow-ReID dataset. The arrows indicate the direction and location of the cameras. Some examples of the images taken are shown.

2.6. 2019 NVIDIA AI City Challenge

The 2019 NVIDIA AI City Challenge [1] was created to accelerate intelligent video analysis that makes cities smarter and safer. It is an important challenge that focus on Intelligent Transportation Systems problems. The first edition in 2017 was aimed to annotate the dataset provided and develop artificial intelligent city applications related to safety and congestion in urban environments. The second edition, in 2018, included the tracks of traffic flow analysis, anomaly detection and multi-sensor vehicle detection and re-identification. In particular, the issues to solve in current 2019 NVIDIA AI City Challenge are traffic anomaly detection, city-scale multi-camera vehicle tracking and city-scale multi-camera vehicle re-identification.

Figure 2.9 shows the scheme of the dataset given by the challenge in order to perform the task of vehicle re-identification. We can see the cameras present in the different scenarios and their direction and location. In the example images we can see two differentiated urban areas, scenario 1 and 2. Then, for scenarios 3, 4 and 5 we can see that are correlated since the same vehicle is captured in its trajectory by all the cameras.

Chapter 3

Design and Development of the proposed method

3.1. Introduction

In this chapter the details of the object detection and association proposed method are exposed. First, a brief overview of our system is showed, followed by the explanation in detail of the features extraction methods and the metric learning techniques used. Further on, the improvement proposals to enhance the results obtained are presented and all the necessaries developments for the 2019 NVIDIA AI City Challenge submission.

3.2. Proposed method review

This section includes the summary of the techniques used to develop the proposed multi-camera vehicle re-identification approach. In Figure 3.1 we have the flow diagram of the approach, first we apply the features extraction methods using the query, train and test sets. Then, we learn the metric in order to get the projection matrix with the features map. The objective of using metric learning is to learn a feature space where features metrics that belongs to the same person are closer than those of different persons. Finally we obtain the distances between each query and all the test set.

We use an unsupervised dimensionality feature reduction scheme, Principal Component Analysis [49]. This reduces the computational cost of the system execution. Based on Person Re-identification Benchmark [2], the selected dimension space is 100.

Depending on the object dataset structure and its annotations, different number of

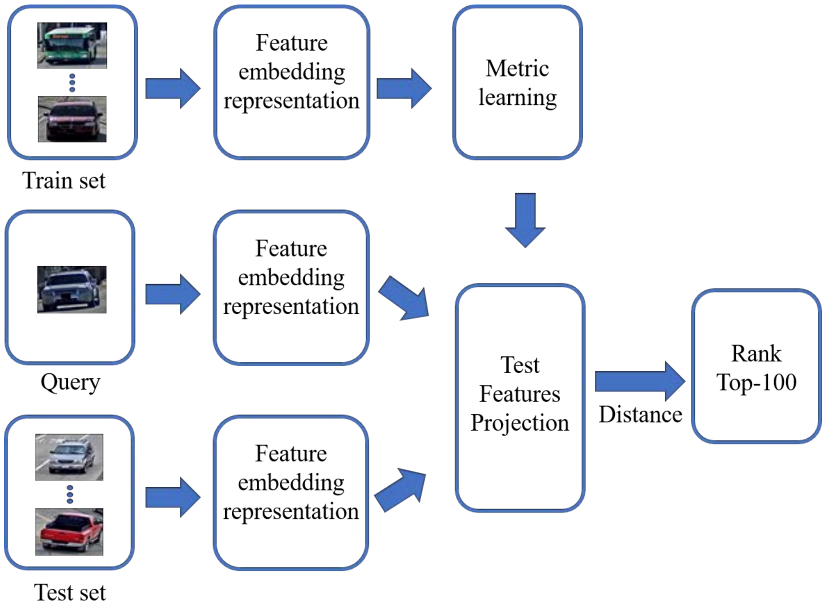

Train set
Feature embedding representation
Metric learning
Query
Feature embedding representation
Test Features Projection
Distance
Rank Top-100
Test set
Feature embedding representation

Figure 3.1: Flow diagram of the system approach.

camera views and number of images per vehicle, some different reorganization would be necessary in order to have a generic environment to evaluate and train the system. In section 4.3.1, the necessary organization of the dataset proposed (CityFlow-ReID-SubSet) and the provided by the challenge (CityFlow-ReID) is presented. We have focused on vehicle images as objects, so all the method and the evaluation will be performed with the specific CityFlow-ReID-Subset dataset. It is not necessary to use detection algorithms because the dataset already includes bounding boxes of the objects.

We are going to use 3 metric learning algorithms in combination with the 14 feature (baselines CNNs GOG and WHOS, and fine-tuned CNNs). In total we evaluate 42 different algorithm combinations and then, for the fine-tuned CNNs, we are going to probe distances and ranks combinations.

3.3. Feature extraction methods

In previous literature work [2] a comparison between the last state of the art methods are presented. With this information, we study in depth the features with higher performance and include them in our system.

The Feature extraction methods can be separated in two groups: one group is the methods that use hand-crafted feature extractions and the other are those methods that use convolutional neural networks.

GOG [28] and WHOS [27] belongs to the first group of hand-crafted methods from the Person Re-identification benchmark [2]. To adapt them to vehicles instead of people, first we calculate the aspect ratio of all the vehicle images. With this value, we resize the images due to these methods work with the same input size. In general, vehicles used to be horizontal, in contrast to person target that used to be vertical. GOG and WHOS extract horizontal strips of the input images (see Figures 2.3 and 2.2), but having the vehicle aspect ratio horizontal, we choose vertical strips which should give us more characteristic information.

The Convolutional Neural Networks used in this system are AlexNet [29], ResNet-18 [30], ResNet-50 [30], ResNet-101 [30], DenseNet-201 [43] and Inception-ResNet-v2 [44]. These are the baseline methods. We choose these networks because of their importance in the state of the art and the competitions they have won as ILSVRC [50] or COCO [51] in the case of ResNet-101. All the CNNs are pre-trained on a generic object dataset (ImageNet [42]). To adapt the feature extraction technique to vehicle model, it is necessary to fine-tune each network with the train set of the dataset CityFlow-ReID [1].

The networks used in Person Re-identification Benchmark [2] were AlexNet, ResNet and VGGNet [31]. We do not use VGGNet and, instead, we include the different variations aforementioned of ResNet, DenseNet-201 and Inception-ResNet-v2.

3.3.1. Deep Learning features generation

Each network used in this system is pre-trained on ImageNet [42] dataset since it is a generic enough object database, so it is not necessary to retrain again the entire network when a new object appears. In order to obtain a vehicles object feature embedding representation for each CNN, we adapt these networks by freezing the weights of the initial layers pre-trained in generic objects, and then adapting the remaining weights during the training (fine-tunning).

To fine-tune the networks, the earlier layers are frozen and the other layers are retrain for the specific task of vehicle re-identification. The classes used are the 166 vehicles from the 18,886 images of the CityFlow-ReID-Subset train part. We have based on [52] to decide the frozen parts of the networks. We freeze before the CNN *block3* except for AlexNet, that we freeze before the *pool1* layer. All the remaining parts of the networks that are not frozen, adapt their weights when we retrain on the vehicle images.

The input images of the CNNs are resize to 227x227. The parameters used for the transfer learning of the not frozen layers are a learning rate of 3e-4 and a batch

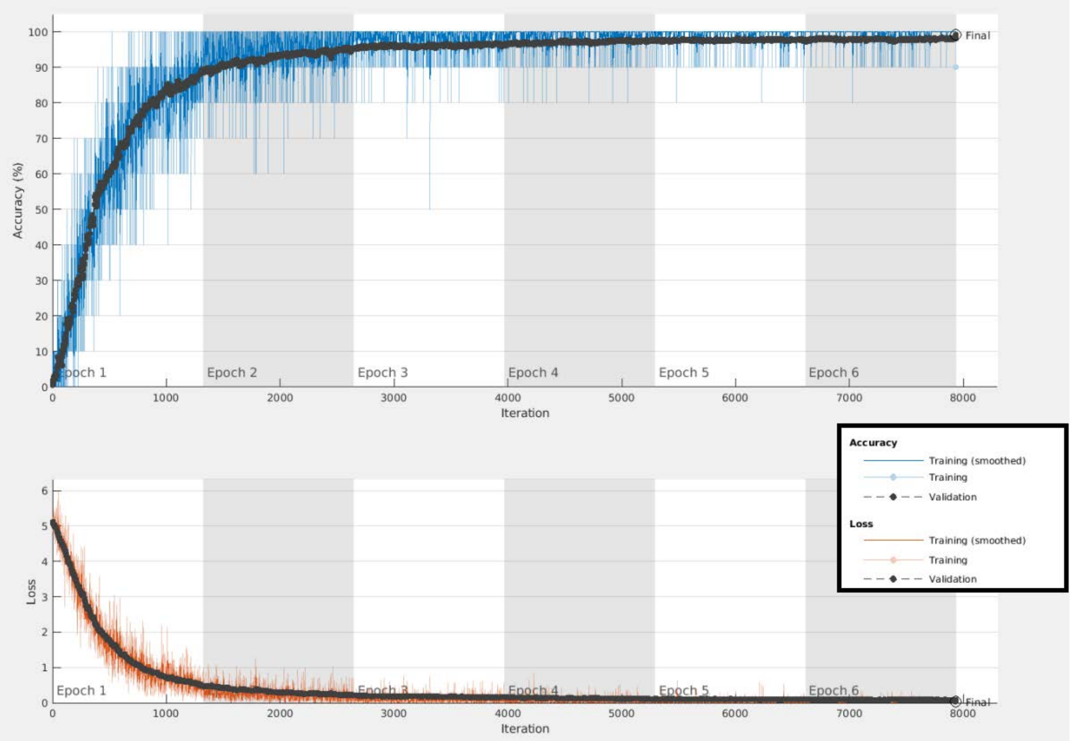


Figure 3.2: ResNet-101 fine-tuned progress. In black dots is represented the validation process. The upper figure represent the accuracy of the training process and the lower figure the loss.

size of 10. We have trained for 6 epochs and use Stochastic Gradient Descent with Momentum optimizer [53].

Figure 3.2 shows the fine-tuning process of the network ResNet-101. For this case, the accuracy reach a 95% in epoch 3 (1322 iterations per epoch). It has a validation accuracy of 99.08% and a validation frequency of 3 iterations. In order to see the graphics for the other networks, see Appendix B.

The retrained AlexNet architecture give us 4096-dimensional feature vector at the output of fc7 layer before the final fully connected and softmax layers. In case of ResNet101 we obtain a 2048-dimensional vector at pool5, and for ResNet50 also a 2048-dimensional vector at average pooling layer. ResNet18 gives a 512-dimension vector at pool5, and finally for DenseNet201 and Inception-ResNet-v2 networks a 1920-dimensional and 1536-dimensional vector respectively at last average pooling layer before the final fully connected and softmax layers.

3.4. Metric learning techniques

Instead of using the combination of a feature embedding representation and the Euclidean distance (l_2) to rank the test candidates, we improve the performance of the system introducing supervision using the training data. In particular, the metric learning allows to learn a feature space where the feature vectors of the same vehicle ID are closer than those features from different vehicles. We consider different metrics according to the results they obtain in benchmark [2]: XQDA [26] , NFST [34] and KLFDA [25].

In NFST we use an exponential kernel due to in [34] the authors show that distance metric learning methods for non-linearity objects benefit from kernelisation. NFST method does not need to perform a reduction before learning and neither a regularisation term. There are no parameters to tune.

The parameters fixed for XQDA are also chosen according to the original paper [26]. The regularizer lambda value, necessary in order to make the estimation covariance matrix smoother and robust, is equals to 0.001. This method learns a subspace with cross-view data, and also a distance function in the subspace for the cross-view similarity measure. The dimension that fixes the XQDA paper of the subspace projection matrix equals to 100.

In case of KLFDA, the set parameters are based on those used in [25]. The dimensionality of the learning feature space is fixed to 40 and the regularizing weight is 0.01. We use a linear kernel according to [25].

3.5. Improvement proposals

All the improvements included are explained in detail in this section in order to obtain better results than those obtained with the baseline method.

3.5.1. Distance combination

To increase the performance of our system, we develop a metric network combination at distances level. As we can see in Figure 3.3, we choose which feature embedding representation and metric learning techniques we want to include (in the left part of the Figure 3.3 the feature spaces are represented). For the task of choosing the methods that would return better results, we performed an evaluation

Then, we extract the distances for each method applying the proposed system. Distances are matrices with a number of rows equal to the number of query image and a number of columns equal to the number of per image from the gallery test

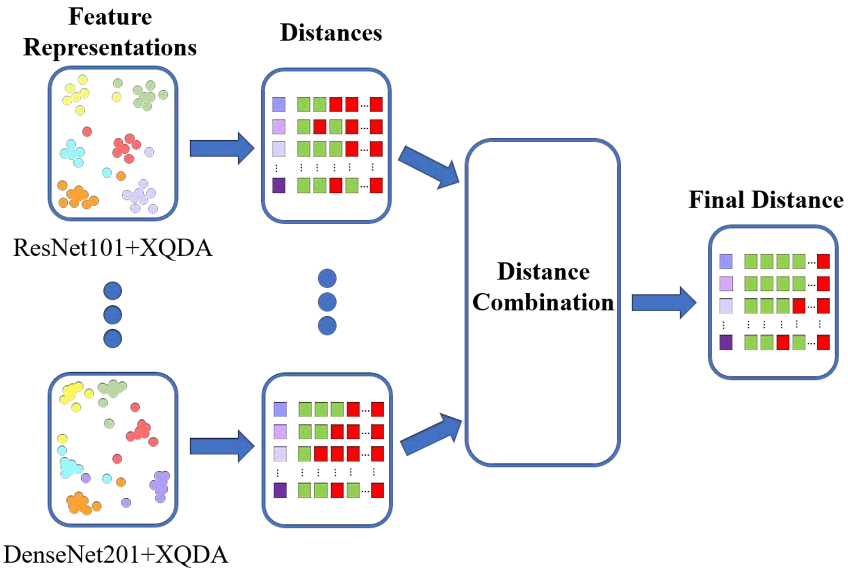


Figure 3.3: Metric network combination at distances level. The Feature representations show the feature spaces obtained with different feature embedding representation and metric learning techniques. Then, for each method the obtained distances are represented, in purple hues for each query image, the true positives matches in green, and the false positives matches in red. After the distance combination, the final distances is shown.

with the distances values sorted in ascendant form. In Figure 3.3 we can see in the distances the representation of the query images in different purple colors, the true positives matches in green, and the false positives matches in red. Finally, we combine different feature extraction and metric learning techniques by normalizing the ranked distances obtained for each one between 0 and 1 and averaging them.

In the Evaluation chapter we will see the combinations of the different features and metric learning techniques, and their results obtained, more specifically in Section 4.4.3.

3.5.2. Rank aggregation

CityFlow-ReID dataset follows the single-shot scheme due to there are only one image per query identity. In this case we are not going to apply the rank algorithms seen in SoA Section 2.5 for multi-shot schemes. Once the metric learning is applied, the rank list is calculated doing the distances between the gallery and the query projected features. Mahalanobis distance [54] is calculated for the XQDA metric learning and for the other techniques, the Euclidean distance (l_2).

As a proposal to increase the performance of our algorithm, we include three rank aggregation methods, in particular Robust Rank Aggregation, Stuart-Aerts method and Geometric mean. These methods combines the different ranks results obtained from different features and metric techniques, into a single rank list.

Robust Rank Aggregation described in [55] is robust to noise and it allows to calculate the relevant probabilities for all the items in the final ranking.

Stuart-Aerts method [56] compares the expected behavior of uncorrelated rankings with the rank list.It ranks the elements and gives significance scores. In comparison with Robust Rank Aggregation, this method does not support incomplete rank lists.

For geometric mean, the lower ranks are penalized while the higher ones are enhanced. An advantage of this method is the low computational cost, also for large rank lists.

3.6. 2019 NVIDIA AI City Challenge development

All the necessary requirements and specific techniques for the participation in the 2019 NVIDIA AI City Challenge are explained in this section.

3.6.1. Dataset organization

The main drawback of this dataset (and also one of the objectives of the 2019 NVIDIA AI City Challenge) is the lack of annotation, due to the camera ID and vehicle ID of the test and query sets are unknown. The absence of ground truth do not allow us to evaluate the results. It is necessary to perform two data reorganizations: one for the 2019 NVIDIA AI City Challenge submission, and another one for the system evaluation. Both of them will be explain in detail in section 4.3.1.1.

For the participation in the 2019 NVIDIA AI City Challenge, we use the entire dataset CityFlow-ReID[1].

3.6.2. Trajectory information

We are not including this improvement in Section 3.5 due to the trajectory information is provided only by this particular dataset, so we can not generalize.

Note that the testing data of this challenge is not used in any way during the training part (neither in feature nor metric learning), so we are going to use the test track information to increase the performance modifying the queries top-100 matches. This dataset includes a script with video sequences information. Each test track sequence indicates the images of a vehicle identity recorded by the same camera,

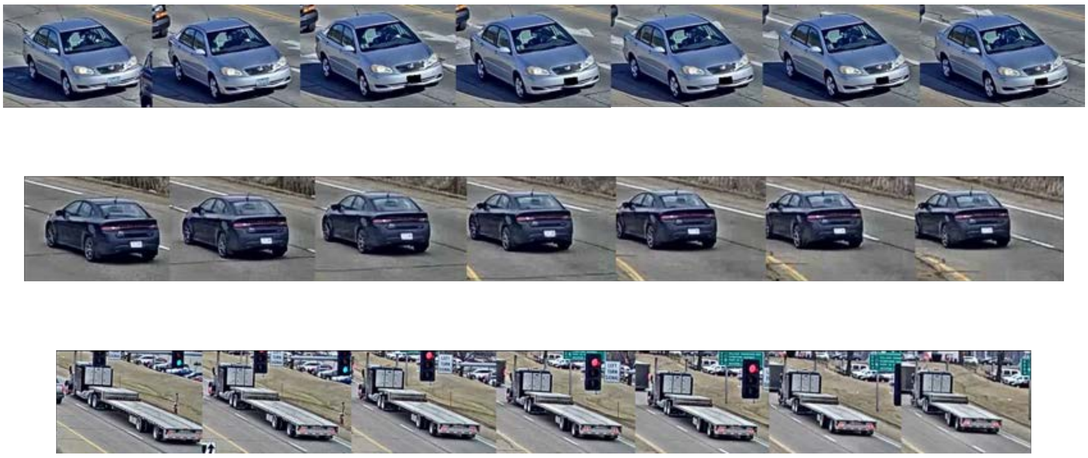


Figure 3.4: Example of track sequences of the City-Flow ReID dataset. The given information is all the images that belongs to a same sequence. The vehicleID and cameraID is unknown.

but it does not specify which vehicle or camera identity is. Then, according to the ranked distance between the query and the test gallery, we can assume that if there are enough images of the same test track and they have small distances, the rest of the track must be presented in the matched list of vehicle re-identification . In Figure 3.4 we have three examples of the given track test information.

We can not assume that all the re-identifications with small distances are true positives, and neither delete all the re-identifications with high distance, because we can loss accuracy. In order to manage the track information, we have followed three methods. We take into account the dataset composition information provided in [1] . On average, each vehicle appears in 4.55 cameras from the total 40. The average of the total number of images per vehicle is 84.50. All of the methods proposed rearrange each query top-100 matches separately.

- **First method:** We check all the tracks that appear in each query top-100 list, and sort them according to the ratio between the number of images of each track that appear in the query list and the total number of images of the track. Once we have all the tracks sorted, we delete one third of the less representative and add all the images of the tracks with higher ratio.

- **Second method:** This method also sorts the tracks according to their ratio between the number of images of each track that appear in the query list and the total number of images of the track. But unlike the first method, here we add all the images from the tracks with higher ratio until we achieve the 100 images.

- **Third method:** We sort the tracks that appear in each query top-100 matches according to their first occurrence in the top-100 list. We add all the images of the sorted tracks until we achieve the 100 matches.

Chapter 4

Evaluation

4.1. Introduction

This chapter covers the evaluation process in order to analyze the obtained results and see the performance of the system. For this purpose, the explanation of the metrics used is included. After that, the baseline feature extraction methods with each metric learning technique is studied. Then, we include the evaluation of the CNNs adapted to vehicle object using fine-tunning and also all the improvements of the system. Finally, the results obtained in the 2019 NVIDIA AI City Challenge are exposed.

4.2. Metrics

In order to evaluate the performance of the developed system, the metrics used are the mean Average Precision (mAP) and the Cumulative Match Characteristic (CMC), using the same version as the used in [21].

Before seeing more in detail the mAP and CMC metrics, it is necessary to see the measures involved:

- **Precision:** Precision indicates the relation between the true matches (True Positives) among all the re-identified matches (True Positives and False Positives).

$$Precision = \frac{True\,Positives}{True\,Positives + False\,Positives} \tag{4.1}$$

- **Recall:** Recall is the relation between all the true matches (True Positives) among all the real matches (True Positives and False Negatives).

$$Recall = \frac{True\,Positives}{True\,Positives + False\,Negatives} \tag{4.2}$$

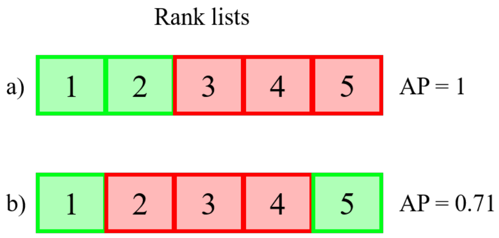


Figure 4.1: Example of the difference between AP and CMC. True Positives are represented by green boxes an False positives in red. For a) and b) the CMC remains 1, but AP is 1 and 0.71 respectively.

4.2.1. Cumulative Match Characteristic curve (CMC)

The Cumulative Match Characteristic curve displays the re-identification rate (or matching rate) as a function of the rank. This curve represents the cumulative probability of a query first match occurrence in the gallery list.

The CMC is accurate if there is only one ground truth per query. This is because only the first match of the query occurred in the gallery list is counted for the CMC.

To avoid the problems of multiple ground truths, the authors in [21] include the mAP, that take into account the recall. So, in the case of two systems which are good enough matching the ground truth but with different recall ability, the CMC is not going to give different enough results while mAP does.

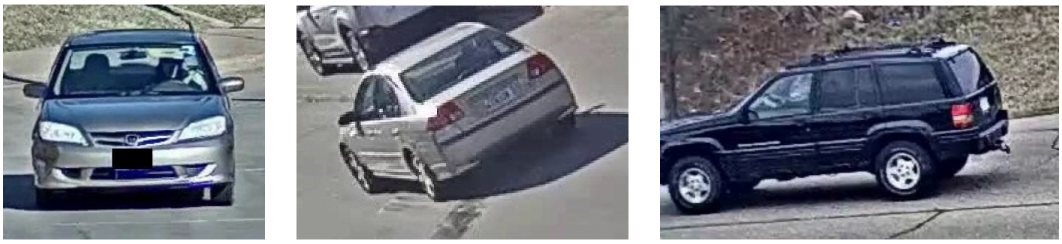
CMC ranked list is performed ranking each query distance with all the gallery test targets. If the vehicle from the query and the gallery are recorded by the same camera, it is not a re-identificaiton. In this case, the re-identification match is deleted in order to not take it into account in the CMC. Once all the queries distances are calculated, the cumulative sum of the first occurrence normalized to the total number of queries is done obtained in this way the CMC.

4.2.2. Mean Average Precision (mAP)

Mean Average Precision is the average of the precision value across all queries average precision. This metric is used in re-identification due to the object can appear in multiple cameras, so the model must be represented by rank-1 to rank-n, and not only by rank-1. The mAP calculates for each query the area under the Precision-Recall curve (AP), and then calculates the mean value of the APs of all the queries.

The intuitive formula used for each Average Precision is:

$$AP = AP_{old} + (recall - recall_{old}) \times \left(\frac{precision_{old} + precision}{2} \right) \tag{4.3}$$



Query Gallery Train

Figure 4.2: Example of the bounding boxes from the query, gallery and train sets present in the dataset CityFlow-ReID [1].The query and gallery vehicle identity are the same recorded by different cameras.

In Figure 4.1, we have an easy example of the requirement of use the mAP in addition to the CMC metrics. We have two ground truth cases b) and c) where CMC is equals to 1, while Average Precision is 1 when the true matches appear at the beginning of the top-rank, and 0.71 when one of the matches appear at the end. So, for obtain good results, the true re-identifications must be at the beginning of the rank list.

4.3. Experimental setup

4.3.1. Dataset

As we have mentioned in Section 3.6.1, it is necessary to reorganize the CityFlow-ReID dataset in order to alleviate the lack of annotated data and be able to make our own experimental setup independently of the challenge. In this section we see in detail the used dataset and our proposed subset.

4.3.1.1. CityFlow-ReID

The dataset CityFlow-ReID used in this algorithm is a subset of the CityFlow [1] dataset. It contains 3.25 hours of synchronized HD from 10 intersections in an U.S. city videos recorded by 40 cameras. It includes different scenes, like city streets, highways and roads from residential areas [1]. It has a huge number of bounding boxes with the annotated data from different cameras, vehicle models and urban traffic flow conditions.

CityFlow-ReID consists of 56,277 bounding boxes with vehicles images of different sizes. From this images, 36,935 belong to the training set with half of the total vehicle identities (333 of the total 666), and the other half belongs to the test set with 18,290 bounding boxes. The 1052 remaining images are the queries bounding boxes. An

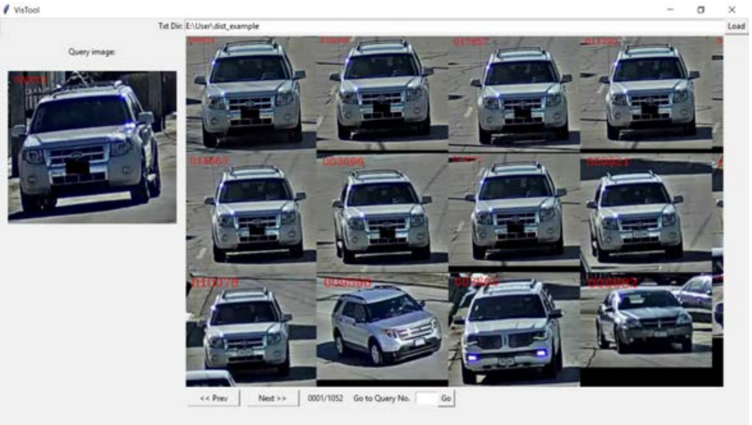


Figure 4.3: Example of a vehicle ID from the query and some of its matches from the test set obtained with the visual tool provided by CityFlow-ReID [1].

example of the query, test and train set is shown in Figure 4.2. The dataset also provides the train and test track information, that consists of the images of the same vehicle ID captured by the same camera. The train labels with the vehicles and cameras IDs are given in a file. Finally, CityFlow-ReID contains a Python tool for visualizing the results of vehicle ReID as we can see in Figure 4.3.

4.3.1.2. Parsing CityFlow-ReID Dataset

The necessary format that must follow the dataset's organization is an important issue, due to the system looks for the images in specific paths with a specific structure. There must be two main folders called *test* and *train*.

Inside the *train* folder, camera subfolder per each camera ID must be created with the name format *cameraXXX*, where *XXX* corresponds to the camera ID number. Inside each camera folder, all the objects ID recorded by this camera must have a folder with the name format *XXXX* according to the object ID. Inside each object ID folder, all the images recorded by the specific camera ID that belong to the specific object ID must be saved with the name format *XXXXXX.jpg*.

For the *test* case, a *gallery* and *probe* subfolder must be created, containing the *gallery* all the test set and the *probe* all the queries images. Inside this two subfolder, the same organization as in the *train* folder must be followed (camera ID folders, object ID folders and the images with the same format explained above).

In order to perform the evaluation method, it is necessary to have all the annotations mentioned in this section. The test set has not the vehicle and camera labels, so we are going to reorganize dataset using only the CityFlow-ReID [1] training set to split it into train and test sets. The vehicles ID must be only in one of the sets because we can not train with the same vehicles present in the test part. We divide the original train choosing randomly half of the vehicles ID and given half to the new test and train sets. We are going to call this CityFlow-ReID-Subset.

4.4. Results

This section includes the results of the proposed system and their evaluation. It is important to highlight that the 14 feature extraction methods present in the system and the three metric learning techniques would give us a total of 42 possible combinations. We are going to include only the combinations which provide higher information for each specific case.

The organization of this section is the following:

- Firstly, we show the baselines results that include the neural networks pre-trained on ImageNet [42] and the feature extraction methods with higher performances according to [2]. In this first part we compare the CNNs using the metric XQDA, adding the other metrics in the Appendix A. We compare the metric learning techniques for the baseline methods.

- Secondly, we include a comparison of fine-tuned networks using XQDA and the comparison of the different metric learning techniques with this feature embedding representations.

- Thirdly, we add the comparison of the distance combination obtained from the fine-tuned networks and the hand-crafted methods with each metric learning techniques. Then, we compare this combination with the option of only use for the combination the feature extraction methods which returns highest performances.

- Fourthly, for the three rank aggregation algorithms we perform the same organization as the done in the evaluation of distance combination.

- Fifthly, we include the three methods designed in order to include the trajectory information. To evaluate them, we use the three best results obtained in the evaluations aforementioned.

- Lastly, we present the results obtained in our participation of the 2019 NVIDIA AI City Challenge.

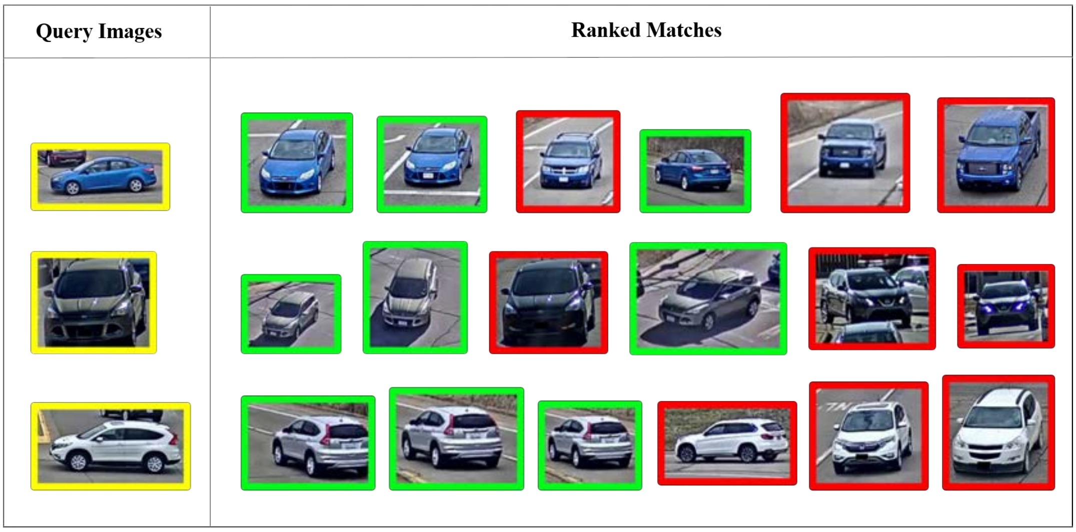


Figure 4.4: Example results of the re-identification output in the CityFlow-ReID-Subset dataset. In yellow is represented the query image, the true matches are in green color and the false matches are in red. The rank position presented are 1, 5, 10, 15, 50, 70 and 90.

Figure 4.4 shows a visual example of the objective pursued in the re-identification task. It represents the top-100 list of the matches from three query images at rank positions 1, 5, 10, 15, 50, 70 and 90. 2019 NVIDIA AI City Challenge works with the top-100 matches, but we can extend it to the total number of images in the CityFlow-ReID-Subset gallery (17128). The list includes an ascendant organization of the distance between the gallery image and the query.

4.4.1. Baselines results

	mAP	Rank-1	Rank-5	Rank-10	Rank-20	Rank-50	Rank-100
AlexNet	6.91%	22.26%	39.20%	46.91%	57.11%	69.16%	76.98%
ResNet-18	5.54%	17.48%	35.07%	42.89%	53.75%	66.12%	74.92%
ResNet-50	8.90%	25.73%	44.52%	53.42%	62.98%	73.40%	79.04%
ResNet-101	8.72%	25.73%	42.56%	51.14%	60.26%	71.23%	78.39%
DenseNet-201	**10.03%**	**28.99%**	**46.58%**	**54.07%**	**63.41%**	**74.92%**	**81.87%**
InceptionResNetv2	6.10%	21.93%	35.94%	44.63%	55.16%	68.95%	76.76%

Table 4.1: Results of the baseline deep learning feature methods obtained in the CityFlow-ReID-Subset, all of them with the metric learning XQDA. The architectures are pre-trained in Imagenet dataset. In bold are the results with the best performance, in particular for DenseNet-201.

In the baseline results we include the GOG and WHOS feature extraction methods, which are hand-crafted techniques, and the convolutional neural networks ResNet-18, ResNet-50, ResNet-101, DenseNet-201 and InceptionResNetv2.

In order to see the different behavior of each CNN, we show in Table 4.1 the results of all the networks for the case of using as metric learning XQDA. The DenseNet-201 returns the highest results for the mAP and CMC rank. For the hand-crafted methods, Table 4.2 includes the results for the three metric learning used in the system. WHOS with XQDA gives the best behavior with a mAP equals to 6.10%, but in comparison with Table 4.1, it is only above the ResNet-18 network.

Table 4.3 includes the comparison of the metric learning techniques in case of CNNs baseline results. It reaffirms what we have seen in Table 4.2, XQDA returns the best results in performance of the baselines techniques.

It is important to emphasize that if the query and the matched image are recorded by the same camera identity, we can not consider a re-identification. In Figure 4.5, we are going to see the impact of the different query cameras and the matched cameras from the gallery. We have a confusion matrix with the gallery cameras in horizontal axis and query cameras in vertical axis. In this matrix the values are represented in gray level colormap and two lined strips. The darker the gray value, the higher the mAP value which represents. The lined strips are drawn in the cameras number that are not provided by the challenge. These missing cameras will be used by the 2019 NVIDIA AI City Challenge to evaluate the participants' algorithms. In the diagonal of the confusion matrix, we can see a miss value for camera 15. The reason of this lack is that, when we performed the parsing step to obtain the subset CityFlow-ReID-Subset, camera 15 only had one vehicle identity and in the division of the train between the new train and test, this identity went to the new train set. For same vehicle identities recorded by the same camera (the diagonal values of the matrix), the re-identification has the highest mAP values, but we are not to consider them as a re-identification as we have mentioned before.

As we have seen in the Figure 2.9, in Section 2.6, there are three differentiated urban areas. One is the scenario 1 recorded by cameras from 1 to 5, then the scenario 2 from cameras 6 to 9 (the missing cameras). The last area is scenario 3, 4 and 5, recorded by cameras from 10 to 40. In all mAP results of the confusion matrix, there are not re-identification matches between scenario 1 and the other ones. There is another diagonal with higher mAP results between the camera pairs 16-21, 17-22, 18-23...22-27. We can see that these camera pairs belong to the same street areas.

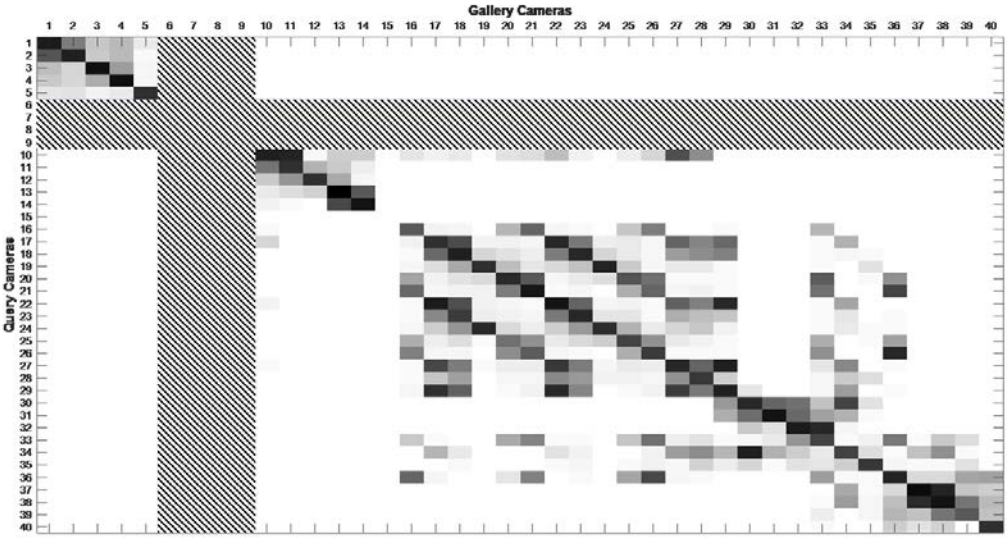


Figure 4.5: Confusion Matrix in case of distance combination of DenseNet-201, ResNet-50, ResNet-101 and metric Learning XQDA in terms of mAP. The absence of cameras between the number 6 and 9 is represented with a striped area. In dark gray it is represented the higher mAP values.

	mAP	Rank-1	Rank-5	Rank-10	Rank-20	Rank-50	Rank-100
GOG XQDA	**5.75%**	17.70%	**32.57%**	**41.37%**	**49.95%**	**64.60%**	**75.24%**
GOG NFST	3.77%	15.31%	26.17%	35.07%	44.73%	61.56%	71.77%
GOG KLFDA	5.21%	**19.54%**	30.62%	38.98%	47.34%	60.15%	70.36%
WHOS XQDA	**6.10%**	**21.82%**	**35.72%**	**43.76%**	**55.16%**	**68.73%**	**77.09%**
WHOS NFST	3.25%	15.64%	26.17%	35.07%	44.73%	61.56%	71.77%
WHOS KLFDA	4.92%	18.89%	31.16%	39.31%	47.45%	61.45%	72.53%

Table 4.2: GOG and WHOS comparisong with XQDA, NFST and KLFDA.

	XQDA	NFST	KLFDA
AlexNet (mAP)	**6.91%**	3.39%	4.16%
ResNet-18 (mAP)	**5.54%**	3.04%	3.85%
ResNet-50 (mAP)	**8.90%**	4.91%	5.37%
ResNet-101 (mAP)	**8.72%**	4.72%	5.59%
DenseNet-201 (mAP)	**10.03%**	6.00%	6.81%
InceptionResNetv2 (mAP)	**6.10%**	3.25%	4.92%

Table 4.3: Metric Learning comparison with baseline CNNs. In bold is the XQDA result with the best performance for all the networks.

	mAP	Rank-1	Rank-5	Rank-10	Rank-20	Rank-50	Rank-100
AlexNet_VPU	12.66%	33.55%	50.38%	58.31%	66.78%	76.44%	85.23%
ResNet18_VPU	23.85%	53.42%	68.73%	73.94%	81.32%	**87.51%**	**92.29%**
ResNet50_VPU	22.75%	55.27%	69.16%	75.14%	79.91%	85.67%	89.47%
ResNet101_VPU	23.43%	56.35%	68.40%	74.59%	80.67%	86.43%	90.66%
DenseNet201_VPU	**30.02%**	**63.19%**	**73.62%**	**78.50%**	**82.74%**	87.30%	91.97%
InceptionResNetv2_VPU	16.39%	39.96%	58.96%	66.99%	74.92%	83.17%	89.90%

Table 4.4: Results of the fine-tuned deep learning feature methods obtained in the CityFlow-ReID-Subset, all of them with the metric learning XQDA. In bold are the results with the best performance, in particular for DenseNet201_VPU and ResNet18_VPU.

4.4.2. Feature embedding representation

In order to see the difference between the fine-tuned feature embedding representation in contrast with using the CNNs trained in ImageNet [42], we show in Table 4.4 the results of the fine-tuned CNNs for the case of using as metric learning XQDA. If we compare these results with the ones of the baseline CNNs in Table 4.1, we realize that using the fine-tuned architectures we obtain more than the double of mAP. For instance, in case of *DenseNet-201* (architecture trained in ImageNet) and *DenseNet-201_ VPU* (architecture fine-tuned in CityFlow-ReID-Subset) the mAP obtained is 10.03% and 30.02% respectively. Also the rank list is significantly higher in case of fine-tuned architectures.

We want to include in Table 4.5 the comparison between the three metrics learning techniques using the fine-tuned CNNs (feature embedding representation) with better performance. We realize that the results are similar, obtaining a better behavior with NFST metric instead of XQDA as happen for the baselines results.

4.4.3. Distance combination results

One of the proposed improvements is a decision combination at distance level, explained in Section 3.5.1. We are going to show the comparison between use the six feature embedding representation and the two hand-crafted methods with each metric learning. Then, we are going to see the difference of use only the three feature embedding representation with the highest performance instead of the eight feature extraction methods.

In Figure 4.6 it is shown the Cumulative Matching Curves comparison for the XQDA, NFST and KLFDA metric learning techniques between the AlexNet_VPU, DenseNet201_VPU, ResNet18_VPU, ResNet50_VPU, ResNet101_VPU, Inception-ResNetv2_VPU, GOG and WHOS feature extraction methods. We can realize

	mAP	Rank-1	Rank-5	Rank-10	Rank-20	Rank-50	Rank-100
				XQDA			
ResNet18_VPU	23.85%	53.42%	68.73%	73.94%	81.32%	87.51%	92.29%
ResNet50_VPU	22.75%	55.27%	69.16%	75.14%	79.91%	85.67%	89.47%
ResNet101_VPU	23.43%	56.35%	68.40%	74.59%	80.67%	86.43%	90.66%
DenseNet201_VPU	30.02%	63.19%	73.62%	78.50%	82.74%	87.30%	91.97%
				NFST			
ResNet18_VPU	27.46%	54.83%	69.82%	76.98%	83.39%	**91.21%**	93.92%
ResNet50_VPU	25.19%	56.46%	70.03%	75.46%	82.08%	87.30%	91.97%
ResNet101_VPU	25.10%	57.65%	**81.55%**	77.20%	82.52%	88.93%	92.62%
DenseNet201_VPU	**33.68%**	**64.06%**	75.46%	**80.67%**	**85.34%**	**91.21%**	**94.03%**
				KLFDA			
ResNet18_VPU	26.15%	54.51%	70.58%	77.09%	82.63%	89.79%	93.38%
ResNet50_VPU	23.83%	56.03%	69.60%	75.35%	79.80%	86.32%	89.36%
ResNet101_VPU	24.43%	56.24%	71.44%	77.31%	82.02%	88.06%	91.64%
DenseNet201_VPU	32.73%	61.56%	75.03%	79.48%	84.15%	89.25%	93.49%

Table 4.5: Results of DenseNet-201_VPU, ResNet101_VPU, ResNet18_VPU and ResNet50_VPU using XQDA, NFST and KLFDA metric learning techniques.

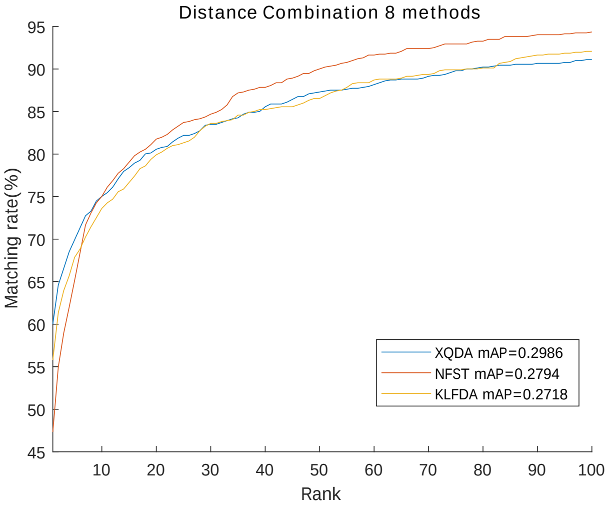


Figure 4.6: Combination at distance level for each metric learning. It combains DenseNet-201_VPU, ResNet101_VPU, ResNet18_VPU, ResNet50_VPU, AlexNet_VPU, InceptionResNetv2_VPU, GOG and WHOS. The mean Average Precision for XQDA is 0.2986, 0.2794 for NFST and 0.2718 for KLFDA. We can see that XQDA gives the higher performance in mAP and CMC-1, lossing accuracy at CMC-100.

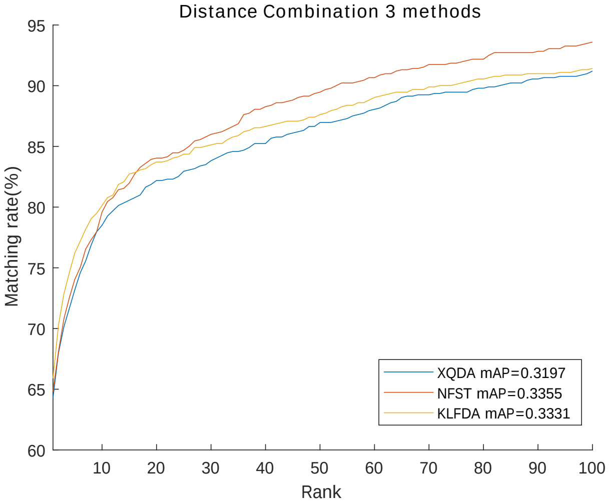


Figure 4.7: Combination at distance level for each metric learning. It combines DenseNet-201_VPU, ResNet101_VPU and ResNet50_VPU. The mean Average Precision for XQDA is 0.3197, 0.3355 for NFST and 0.3331for KLFDA. The combination that uses NFST gives the higher performance in mAP and CMC-1 and CMC-100.

that XQDA gives higher CMC-1 value (60.04%) than the combination using NFST (47.23%) and KLFDA (55.81 %). Conversely, XQDA losses accuracy at CMC-100 (91.10%) and the other metrics increase it (NFST 94.35% and KLFDA 92.07%).

Figure 4.7 shows the distance combination for all the metric learnings and the feature embedding representation which gives the higher performance (DenseNet201_VPU, ResNet50_VPU and ResNet101_VPU). For this case, the combination which returns the higher mAP is the one which uses NFST, giving a mAP equals to 0.3355, a CMC-1 equals to 64.48% and a CMC-100 equals to 93.59%. In case of use for the combination the metric learning KLFDA, it gives a mAP equal to 0.3331, a CMC-1 equals to 65.91% which overcomes the CMC-1 of NFST and a CMC-100 of 91.42%. On the other hand, using XQDA which has given better result in the combination of the eight methods, in the case of using the three best feature embedding representations, it gives us a lower mAP and CMC than NFST and KLFDA. XQDA has a mAP value equals to 0.3197, a CMC-1 of 64.28% and a CMC-100 equal to 91.21%.

It has been confirmed that distance combination improves the results of just using feature extraction methods with metric learning techniques. We also realize that combining the methods which returns higher performance (DenseNet-201_VPU, ResNet101_VPU and ResNet50_VPU using all of them NFST) gives a higher mAP and CMC results than if we use all the algorithms, including the ones which returns the lowest results (GOG, WHOS, InceptionResNetv2_VPU). We include in

Appendix A the combination of the three ResNet networks instead of the three ones combined here. We confirm that the results are better if the combination is done with DenseNet-201_VPU, ResNet101_VPU and ResNet50_VPU.

4.4.4. Rank aggregation results

In this section we are going to see the results of making rank aggregation for different feature extraction and metric learning combinations. The rank aggregation methods, explained in Section 3.5.2, are Stuart-Aerts method, Robust Rank Aggregation and Geometric mean. We are going to compare the three rank methods for each metric learning first, using the six feature embedding representation and the two hand-crafted methods. Then, instead of combine eight feature extraction methods, we only going to select the three feature embedding representation as in Section 4.4.3. In Appendix A we show the combination of the three ResNet networks instead of the three ones combined here. We confirm that the results are better if the combination is done with this one with returns the highest results.

In Figure 4.8 we have the Cumulative Match Characteristic curve for the combination of DenseNet-201_VPU, ResNet101_VPU, ResNet18_VPU, ResNet50_VPU, AlexNet_VPU, InceptionResNetv2_VPU, GOG and WHOS using the three different rank aggregation methods and XQDA, NFST and KLFDA metric learning techniques. We have for each combination the eight feature extraction methods with one metric learning technique and one rank aggregation method. It is represented with continuous lines the XQDA results for each rank aggregation method, with dotted lines the KLFDA and in discontinuous line the NFST. We can observe that the CMC curves of the three rank aggregation which use NFST overcome the other ones. For instance, Geometric mean method using NFST obtains CMC-1 equals to 52.33%, CMC-100 equals to 95.01% and a mAP equals to 0.2932, while for XQDA CMC-1 is equals to 56.24% and CMC-100 is equals to 91.75% and a mAP equals to 0.2875.

As it happens in the distance combination, if instead of work with all the feature extraction methods, we only select the ones which return the best results, we are going to improve the performance of the system. Figure 4.9 collect the CMC curves which belongs to each rank aggregation method and metric learning for the combination of the feature embedding representations DenseNet-201_VPU, ResNet101_VPU and ResNet50_VPU. It has a similar behavior than the previous Figure 4.8. In terms of CMC values the three rank aggregation methods which use NFST overcome KLFDA and XQDA. In terms of mAP values, Geometric mean methods using XQDA has the highest mAP, equals to 0.3322 (0.0447 more than using all the feature extraction methods). In Appendix A we include the rank aggregation for the three ResNet

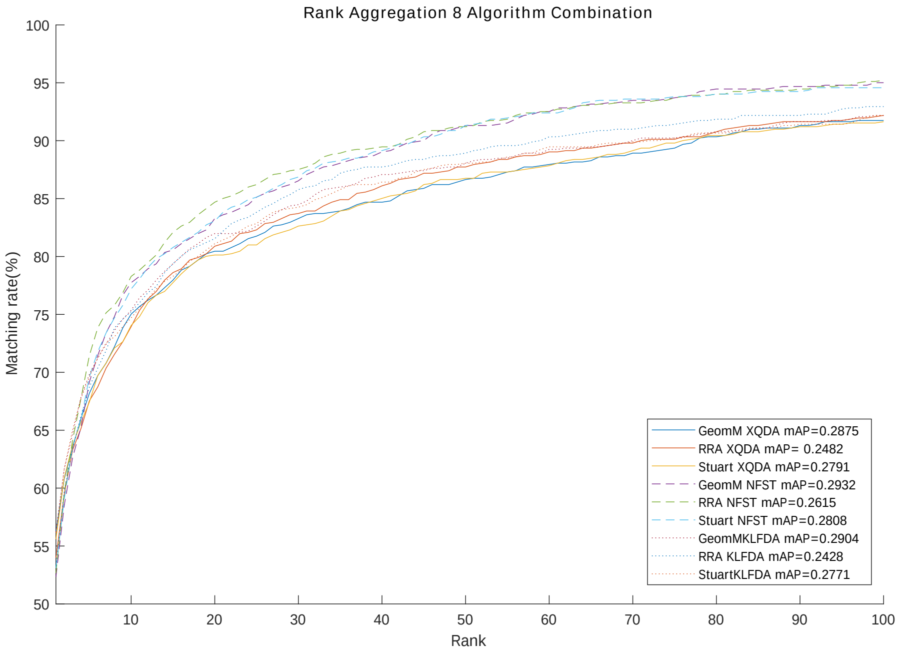


Figure 4.8: Rank aggregation methods (Geometric mean, Robust Rank Aggregation and Stuart-Aerts) for each metric learning. It combines DenseNet-201_VPU, ResNet101_VPU, ResNet18_VPU, ResNet50_VPU, AlexNet_VPU, InceptionResNetv2_VPU, GOG and WHOS. The higher mAP equals to 0.2904 is given by Geometric mean rank aggregation using NFST metric learning.

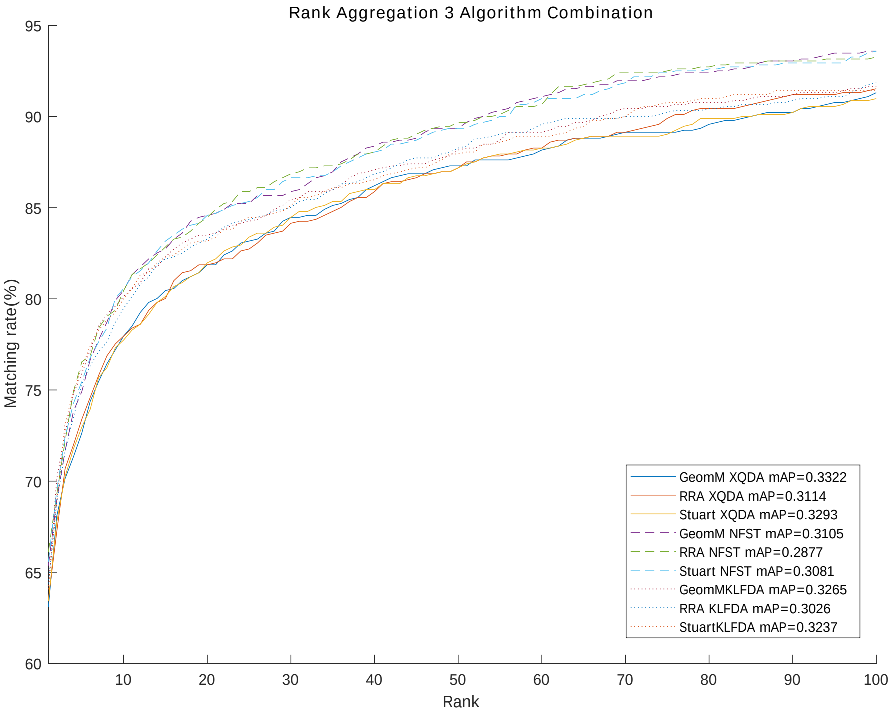


Figure 4.9: Rank aggregation methods (Geometric mean, Robust Rank Aggregation and Stuart-Aerts) for each metric learning. It combines DenseNet-201_VPU, ResNet101_VPU and ResNet50_VPU. The higher mAP equals to 0.3322 is given by Geometric mean rank aggregation using XQDA metric learning.

feature embedding representation.

4.4.5. Trajectory information results

In order to show the improvement of add trajectory information, we are going to show the results of the three different methods explained in Section 3.6.2. As we have seen before, these methods use the track information that provides the challenge (images of the same vehicle recorded by the same camera but without the vehicle ID or camera ID).

As input, we are going to use the ranked images list given by the methods which returns highest results (distance combination and rank aggregation). The list has per row the top-100 images matches between a query identity and all the gallery. We can assume that if there are enough images of the same test track and they have small distances, the rest of the track must be presented in the matched list of vehicle re-identification. In order to compare our results with the ones of the 2019 NVIDIA AI City Challenge, we are going to evaluates the top-100 ranked results as they do. In previous sections we show the results given by the distance between each query and all the gallery.

In Table 4.6 first we show the top-100 best results obtained for each metric learning technique for the best three feature embedding representation (DenseNet-201_VPU, ResNet101_VPU and ResNet18_VPU) combined using distance combination or rank aggregation. In detail, the best results are given by distance combination of the three feature embedding representation using KLFDA (mAP = 56.14%).

We apply the track information with the three methods explained in Section 3.6.2 and we realize that the method 2 and 3 returns better results than method 1. Distance combination of the three feature embedding representation and NFST using the rearrange track information second method gives the highest mAP performance.

4.5. 2019 AI City Challenge Results

The results of the 2019 NVIDIA AI City Challenge have been published on May 2019. There were three tracks with different issues to solve. Fist track was City-scale multi-camera vehicle tracking, second one was the City-scale multi-camera vehicle re-identification (our participation track) and the last one was Traffic anomaly detection. The number of participant to each track were 22, 84 and 23 respectively, being our track the one with more participants. We published our work in paper [57].

The environment given by 2019 NVIDIA AI City Challenge has allowed to submit up 5 results per day, with a total of 20 submissions. The results that have returned

	mAP	Rank-1	Rank-5	Rank-10	Rank-20	Rank-50	Rank-100
Methods without track information							
Dist_NFST	56.11%	71.23%	80.46%	85.99%	90.45%	95.87%	100%
Dist_KLFDA	**56.14%**	**74.48%**	**84.80%**	**88.71%**	**92.29%**	**96.20%**	**100%**
Rank_XQDA	54.47%	71.77%	81.32%	86.64%	90.55%	95.98%	100%
Dist_NFST + track information							
Dist_NFST_Method1	55.57%	73.62%	82.74%	87.95%	92.40%	97.39%	100%
Dist_NFST_Method2	**65.29%**	81.32%	81.87%	83.06%	86.10%	94.35%	100%
Dist_NFST_Method3	64.97%	81.22%	81.65%	82.74%	85.34%	94.03%	100%
Dist_KLFDA+ track information							
Dist_KLFDA_Method1	55.57%	77.09%	**87.19%**	**91.10%**	**94.68%**	**97.94%**	100%
Dist_KLFDA_Method2	65.16%	**82.84%**	83.17%	84.47%	87.19%	94.68%	100%
Dist_KLFDA_Method3	**66.00%**	82.74%	82.95%	83.93%	86.64%	94.35%	100%
Rank_XQDA+ track information							
Rank_XQDA_Method1	53.93%	74.27%	83.71%	89.14%	92.73%	97.39%	100%
Rank_XQDA_Method2	63.94%	81.00%	81.22%	82.41%	84.69%	93.16%	100%
Rank_XQDA_Method3	63.33%	81.98%	82.19%	83.17%	85.34%	94.03%	100%

Table 4.6: Results obtained with the top-100 matches. *“Dist_NFST”* represents the results obtained with distance combination of Densenet-201_VPU, ResNet101_VPU and ResNet50_VPU using NFST metric learning. *“Dist_KLFDA”* represents the results obtained with distance combination of Densenet-201_VPU, ResNet101_VPU and ResNet50_VPU using KLFDA metric learning. *“Rank_XQDA”* represents the results obtained with the geometric mean rank aggregation of Densenet-201_VPU, ResNet101_VPU and ResNet50_VPU using XQDA metric learning. Then we show the results of apply the track information methods each one.

the server until the competition deadline were computed on a 50% subset of the test data. The online server also has provided a leader board with the top 3 results of all the competition and the own best result (in case not to be on the top-3). Once the deadline has been reached, the server shows all the submissions evaluated with all the test set and the entire leader board with all the participants' best result.

In Table 4.7 we can see the results given at the end of the challenge of the different methods that we have developed. First of all, we have the features embedding representation with XQDA as metric learning and the CNNs AlexNet_VPU, ResNet18_VPU, ResNet50_VPU, ResNet101_VPU and DenseNet201_VPU, given ResNet101_VPU and DensNet20_VPU1 the best results in mAP and in Rank-1, and Rank-100 for the case of DenseNet201. Then, we develop the distance combinations with the distance of ResNet101_VPU, ResNet50_VPU and ResNet18_VPU (DisCombResNet) and ResNet101_VPU, DenseNet201_VPU and ResNet50_VPU (DistCombRes-Dense-Net), obtaining similar ranks values and a higher mAP than with each network separately. When we include the information of the tracks files provided in the CityFlow-ReID [1] explained in section 3.6.2, we improve the mAP with the inconvenient that we loss precision. DistCombResNet method1 ,DistCombResNet method2 ,DistCombResNet method3 are the first, second and third method respectively. The best result is given by the third method of the distance combination of ResNet101, DenseNet201 and ResNet50 (DistCombRes-Dense-Net method3) with a mAP value of 25.05%.

We compare the results obtained with our experimental setup included in Table 4.4 with the ones obtained in the 2019 NVIDIA AI City server in Table 4.7. For instance, the value of AlexNet_VPU in our evaluation gives a mAP value of 12.66% while in the 2019 NVIDIA AI City evaluation is 7.04%. The same thing happens with the results of the other feature embedding representations. In our evaluation the results are around double than for the 2019 NVIDIA AI City server. That could be because, our evaluation is done in a reduce subset of the CityFlow-ReID dataset given, and furthermore, the challenge does not provide the entire data in order to make its own evaluation.

The method proposed in this paper has finished the 60 out of the 84 participating teams on the challenge City-Scale Multi-Camera Vehicle Re-Identification. In order to compare our performance in the challenge with the other teams, we show in Table 4.8 the participants that are in the multiples of ten positions in the rank. We can see that the team in position 40^{th} ($TJU0432$), that is in the middle of the ranked results of the challenge, has a mAP score equal to 33.39%, which is only 8.34% more than our mAP result (25.05%). Best mAP result achieved in the challenge is equal to

	Rank-100 mAP	CMC-1	CMC-5	CMC-10	CMC-30	CMC-100
AlexNet_VPU	7.04%	22.91%	33.17%	39.35%	51.52%	59.98%
ResNet18_VPU	10.94%	30.89%	42.02%	50.95%	65.21%	72.15%
ResNet50_VPU	12.05%	33.37%	44.96%	51.33%	64.64%	72.43%
ResNet101_VPU	13.81%	36.79%	47.53%	53.52%	66.83%	74.14%
DenseNet201_VPU	13.63%	36.31%	46.48%	52.85%	68.44%	76.14%
DistCombResNet_VPU	15.54%	39.07%	49.14%	53.23%	67.11%	73.29%
DistCombResNet method1	16.45%	39.07%	49.14%	53.14%	66.25%	71.48%
DistCombResNet method2	23.44%	38.88%	39.26%	39.54%	46.39%	53.04%
DistCombResNet method3	24.25%	39.07%	39.07%	39.35%	45.72%	51.71%
DistCombRes-Dense-Net	16.66%	**40.97%**	**49.81%**	**55.32%**	**69.11%**	**75.86%**
DistCombRes-Dense-Net method3	**25.05%**	**40.97%**	40.97%	41.25%	47.53%	53.52%

Table 4.7: Results obtained in the online evaluation 2019 NVIDIA AI City Challenge [1] server for our different methods, all of them with the metric learning XQDA.

Team Name	Rank in Leader Board	mAP Score
Zero_One	1	85.54%
UWIPL	2	79.17%
ANU AI city tracking and Re-ID team	3	75.89%
flyZJ	10	58.27%
BUPT-MCPRL	20	46.10%
SYSUITS	30	37.69
TJU0432	40	33.39%
Alpha	50	29.65%
VPUTeam	**60**	**25.05%**
NCTUAI	70	20.18%
i-TRACK	80	1.46%

Table 4.8: Results of the leader board in [1] .

85.54%. The teams with the best performance use as baseline the networks trained using triplet loss or cross entropy loss. They also include in the classification step the information of vehicle models and the vehicle orientation.

Chapter 5

Conclusions and future work

5.1. Conclusions

In this work we have proposed an object detection and association system in multiview scenarios, in particular, a vehicle re-identification system in a city-scale mutli-camera scenario. In order to address all the difficulties present in the poor data quality, lack of labelled data, similarities in vehicles models and variability of the same vehicle from different points of view, we have developed a system based on adapted feature embedding representation networks and metric learning techniques.

For this task we have studied the related work in the literature of multi-camera object detection and association methods, highlighting the system Person Re-identification Benchmark [2] because of the large number of methods it compares. From this Benchmark, we have included in our system the feature extraction methods and learning metric techniques which return better results. Then we have improved the methods adapting the convolutional neural networks to the specific task of vehicle re-identification with fine-tunning. Finally, we have increased the accuracy with a metric network combination at distances level, rank aggregation and adding video tracking information that provides the CityFlow-ReID dataset.

We have participated in 2019 NVIDIA AI City Challenge, that aims to perform vehicle re-identification based on vehicle images from multiple cameras placed at multiple intersection. We obtained the 60^{th} position in the participants' Leader Board from the 84 teams and we have published our work in paper [57].

In addition to compare ourselves with the participants of the 2019 NVIDIA AI City Challenge, we also have developed our own experimental setup.

5.2. Future work

The system proposed is the first approximation to the vehicle re-identification problem. Other training strategies could shed light and improve this work in a future. These include, among others, adding hard triplet loss [58] or cross entropy loss [59] in order to optimize the train step of the network for the final task. In [1] the baseline methods proposed combine triplet loss and cross entropy loss, obtaining the highest performance.

Also including the association of the landmarks from different points of view of the same vehicle ID [60], and the information of different vehicle models could improve the results.

Bibliography

[1] Z. Tang, M. Naphade, M.-Y. Liu, X. Yang, S. Birchfield, S. Wang, R. Kumar, D. C. Anastasiu, and J.-N. Hwang, "Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[2] S. Karanam, M. Gou, Z. Wu, A. Rates-Borras, O. Camps, and R. J. Radke, "A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2018.

[3] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[4] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2008.

[5] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," *Proc. of the DARPA Image Understanding Workshop*, pp. 121–130, 1981.

[6] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identification by video ranking," in *Proc. of the European Conference on Computer Vision*, pp. 688–703, Springer, 2014.

[7] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proc. of the European Conference on Computer Vision*, pp. 262–275, Springer, 2008.

[8] D. Figueira, M. Taiana, A. Nambiar, J. Nascimento, and A. Bernardino, "The hda+ data set for research on fully automated re-identification systems," in *Proc. of the European Conference on Computer Vision*, pp. 241–255, Springer, 2014.

[9] D. Seon Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino, "Custom pictorial structures for re-identification," in *Proc. of the BMVC*, pp. 68.1–68.11, 2011.

[10] N. Martinel, C. Micheloni, and C. Piciarelli, "Distributed signature fusion for person re-identification," in *Proc. of the Sixth International Conference on Distributed Smart Cameras*, pp. 1–6, IEEE, 2012.

[11] D. Baltieri, R. Vezzani, and R. Cucchiara, "3dpes: 3d people dataset for surveillance and forensics," in *Proc. of the joint ACM workshop on Human gesture and behavior understanding*, pp. 59–64, ACM, 2011.

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

[13] A. Bialkowski, S. Denman, S. Sridharan, C. Fookes, and P. Lucey, "A database for person re-identification in multi-camera surveillance networks," in *Proc. of the International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pp. 1–8, IEEE, 2012.

[14] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *Proc. of the Asian Conference on ComputerVision*, pp. 31–44, Springer, 2012.

[15] W. Li and X. Wang, "Locally aligned feature transforms across views," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3594–3601, 2013.

[16] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 152–159, 2014.

[17] M. Gou, S. Karanam, W. Liu, O. Camps, and R. J. Radke, "Dukemtmc4reid: A large-scale multi-camera person re-identification dataset," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1425–1434, 2017.

[18] C. C. Loy, T. Xiang, and S. Gong, "Time-delayed correlation analysis for multi-camera activity understanding," *International Journal of Computer Vision*, vol. 90, no. 1, pp. 106–129, 2010.

[19] S. Wang, M. Lewandowski, J. Annesley, and J. Orwell, "Re-identification of pedestrians with variable occlusion and scale," in *Proc. of the IEEE International Conference on Computer Vision Workshops*, pp. 1876–1882, IEEE, 2011.

[20] A. Das, A. Chakraborty, and A. K. Roy-Chowdhury, "Consistent re-identification in a camera network," in *Proc. of the European Conference on Computer Vision*, pp. 330–345, Springer, 2014.

[21] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. of the IEEE International Conference on Computer Vision*, pp. 1116–1124, 2015.

[22] B. Ma, Y. Su, and F. Jurie, "Covariance descriptor based on bio-inspired features for person re-identification and face verification," *Image and Vision Computing*, vol. 32, no. 6-7, pp. 379–390, 2014.

[23] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised salience learning for person re-identification," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3586–3593, 2013.

[24] B. Ma, Y. Su, and F. Jurie, "Local descriptors encoded by fisher vectors for person re-identification," in *Proc. of the European Conference on Computer Vision*, pp. 413–422, Springer, 2012.

[25] F. Xiong, M. Gou, O. Camps, and M. Sznaier, "Person re-identification using kernel-based metric learning methods," in *Proc. of the European Conference on Computer Vision*, pp. 1–16, Springer, 2014.

[26] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2197–2206, 2015.

[27] G. Lisanti, I. Masi, A. D. Bagdanov, and A. Del Bimbo, "Person re-identification by iterative re-weighted sparse ranking," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 8, pp. 1629–1642, 2014.

[28] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato, "Hierarchical gaussian descriptor for person re-identification," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1363–1372, 2016.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, pp. 1097–1105, 2012.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[32] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[33] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian, "Local fisher discriminant analysis for pedestrian re-identification," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3318–3325, 2013.

[34] L. Zhang, T. Xiang, and S. Gong, "Learning a discriminative null space for person re-identification," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1239–1248, 2016.

[35] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 1, pp. 40–51, 2007.

[36] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[37] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. of the 24th International Conference on Machine learning*, pp. 209–216, ACM, 2007.

[38] W.-S. Zheng, S. Gong, and T. Xiang, "Person re-identification by probabilistic relative distance comparison," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 649–656, IEEE, 2011.

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

[40] A. Mignon and F. Jurie, "Pcca: A new approach for distance learning from sparse pairwise constraints," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2666–2672, IEEE, 2012.

[41] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, IEEE Computer Society, 2005.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Ieee, 2009.

[43] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

[44] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. of the AAAI Conference on Artificial Intelligence*, 2017.

[45] K. Fukunaga, "Introduction to statistical pattern recognition, chapter 10." Academic Press, New York, NY, USA, 1990.

[46] Y.-F. Guo, L. Wu, H. Lu, Z. Feng, and X. Xue, "Null foley–sammon transform," *PatternRecognition*, vol. 39, no. 11, pp. 2248–2251, 2006.

[47] H. Cevikalp and B. Triggs, "Face recognition based on image sets," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2567–2573, IEEE, 2010.

[48] M. Yang, P. Zhu, L. Van Gool, and L. Zhang, "Face recognition based on regularized nearest points between image sets," in *Proc. of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pp. 1–7, IEEE, 2013.

[49] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[50] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

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

[52] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige, "On pre-trained image features and synthetic images for deep learning," in *Proc. of the European Conference on Computer Vision*, pp. 0–0, 2018.

[53] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.

[54] P. C. Mahalanobis, "On the generalized distance in statistics," National Institute of Science of India, 1936.

[55] R. Kolde, S. Laur, P. Adler, and J. Vilo, "Robust rank aggregation for gene list integration and meta-analysis," *Bioinformatics*, vol. 28, no. 4, pp. 573–580, 2012.

[56] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim, "A gene-coexpression network for global discovery of conserved genetic modules," *Science*, vol. 302, no. 5643, pp. 249–255, 2003.

[57] E. Luna, P. Moral, J. C. SanMiguel, A. Garcıa-Martın, and J. M. Martınez, "Vpulab participation at ai city challenge 2019," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 343–352, 2019.

[58] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[59] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[60] Z. Wang, L. Tang, X. Liu, Z. Yao, S. Yi, J. Shao, J. Yan, S. Wang, H. Li, and X. Wang, "Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 379–387, 2017.

Appendix A

Result comparison

A.1. Baseline results

	mAP	Rank-1	Rank-5	Rank-10	Rank-20	Rank-50	Rank-100
AlexNet	3.39%	14.01%	25.30 %	33.12 %	45.17%	61.13 %	74.05%
ResNet-18	3.04%	13.57%	24.97%	34.42%	45.28%	63.63%	74.59%
ResNet-50	4.91%	14.22%	28.01%	36.92%	52.01%	69.82%	74.59%
ResNet-101	4.72%	13.14%	27.14%	36.92%	50.92%	71.66%	80.89%
DenseNet-201	**6.00%**	8.24%	**33.44%**	**43.65%**	**57.44%**	**73.83%**	**83.82%**
InceptionResNetv2	3.25%	**15.64%**	26.17%	35.07%	44.73%	61.56%	71.77%

Table A.1: Results of the baseline deep learning feature methods obtained in the CityFlow-ReID-Section, all of them with the metric learning NFST. The architectures are pre-train in Imagenet dataset. In bold are the results with the best performance.

	mAP	Rank-1	Rank-5	Rank-10	Rank-20	Rank-50	Rank-100
AlexNet	4.16%	14.88%	26.98%	35.83%	45.06%	59.17%	69.06%
ResNet-18	3.85%	15.42%	27.36%	34.31%	42.56%	56.13%	67.54%
ResNet-50	5.37%	17.59%	28.88%	34.85%	43.21%	56.89%	69.16%
ResNet-101	5.59%	19.76 %	31.05%	38.11%	49.51%	56.61%	69.38%
DenseNet-201	**6.81%**	**21.50%**	**35.29%**	**43.32%**	**53.20%**	**64.93%**	**73.94%**
InceptionResNetv2	4.92%	18.89%	31.16%	39.31%	47.45%	61.45%	72.53%

Table A.2: Results of the baseline deep learning feature methods obtained in the CityFlow-ReID-Section, all of them with the metric learning KLFDA. The architectures are pre-train in Imagenet dataset. In bold are the results with the best performance.

A.2. Feature embedding results

	mAP	Rank-1	Rank-5	Rank-10	Rank-20	Rank-50	Rank-100
AlexNet_VPU	13.44%	32.90%	50.92%	61.35%	69.16%	79.59%	89.79%
ResNet18_VPU	27.46%	54.83%	69.82%	76.98%	83.39%	91.21%	93.92%
ResNet50_VPU	25.19%	56.46%	70.03%	75.46%	82.08%	87.30%	91.97%
ResNet101_VPU	25.10%	57.65%	81.55%	77.20%	82.52%	88.93%	92.62%
DenseNet201_VPU	**33.68%**	**64.06%**	**75.46%**	**80.67%**	**85.34%**	**91.21%**	**94.03%**
InceptionResNetv2_VPU	16.15%	43.00%	60.80%	68.08%	77.09%	86.54%	92.51%

Table A.3: Results of the finetune deep learning feature methods obtained in the CityFlow-ReID-Section, all of them with the metric learning NFST. In bold are the results with the best performance, in particular for DenseNet201_VPU.

	mAP	Rank-1	Rank-5	Rank-10	Rank-20	Rank-50	Rank-100
AlexNet_VPU	13.17%	37.68%	54.72%	62.76%	71.34%	80.56%	86.32%
ResNet18_VPU	26.15%	54.51%	70.58%	77.09%	82.63%	**89.79%**	93.38%
ResNet50_VPU	23.83%	56.03%	69.60%	75.35%	79.80%	86.32%	89.36%
ResNet101_VPU	24.43%	56.24%	71.44%	77.31%	82.08%	88.06%	91.64%
DenseNet201_VPU	**32.73%**	**61.56%**	**75.03%**	**79.48%**	**84.15%**	89.25%	**93.49%**
InceptionResNetv2_VPU	18.32%	41.26%	61.67%	70.47%	77.52%	86.21%	91.86%

Table A.4: Results of the finetune deep learning feature methods obtained in the CityFlow-ReID-Section, all of them with the metric learning KLFDA. In bold are the results with the best performance, in particular for DenseNet201_VPU and ResNet18_VPU.

A.3. Distance combination

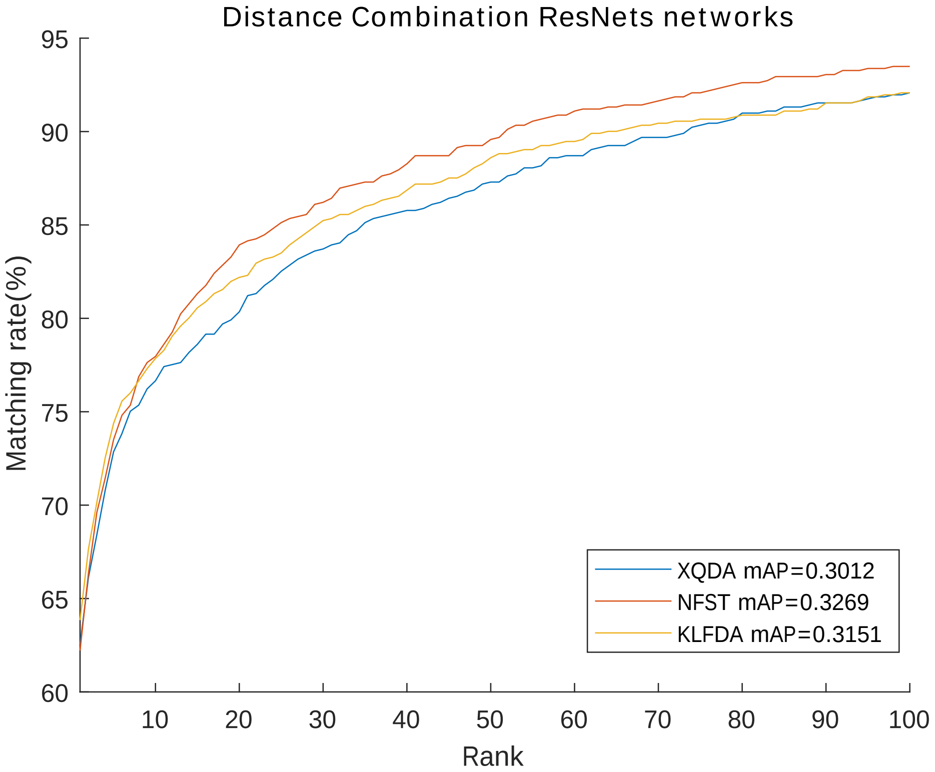


Figure A.1: Combination at distance level for each metric learning. It combines ResNet18_VPU, ResNet101_VPU and ResNet50_VPU. The mean Average Precision for XQDA is 0.3012, 0.3269 for NFST and 0.3151 for KLFDA. The combination that uses NFST gives the higher performance in mAP and CMC-1 and CMC-100.

A.4. Rank aggregation

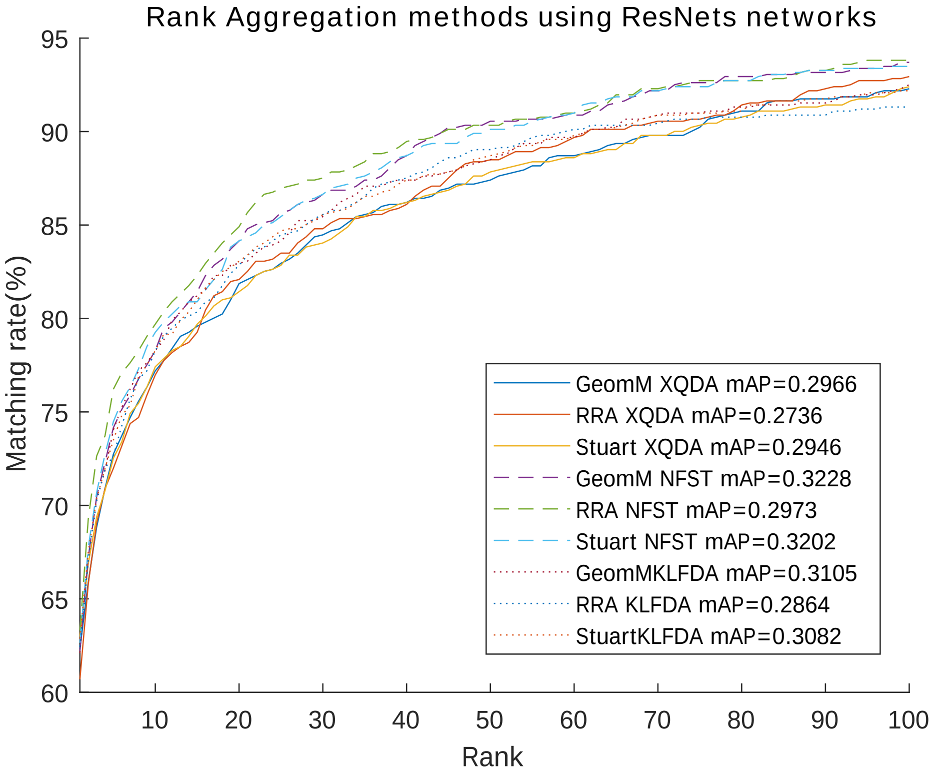
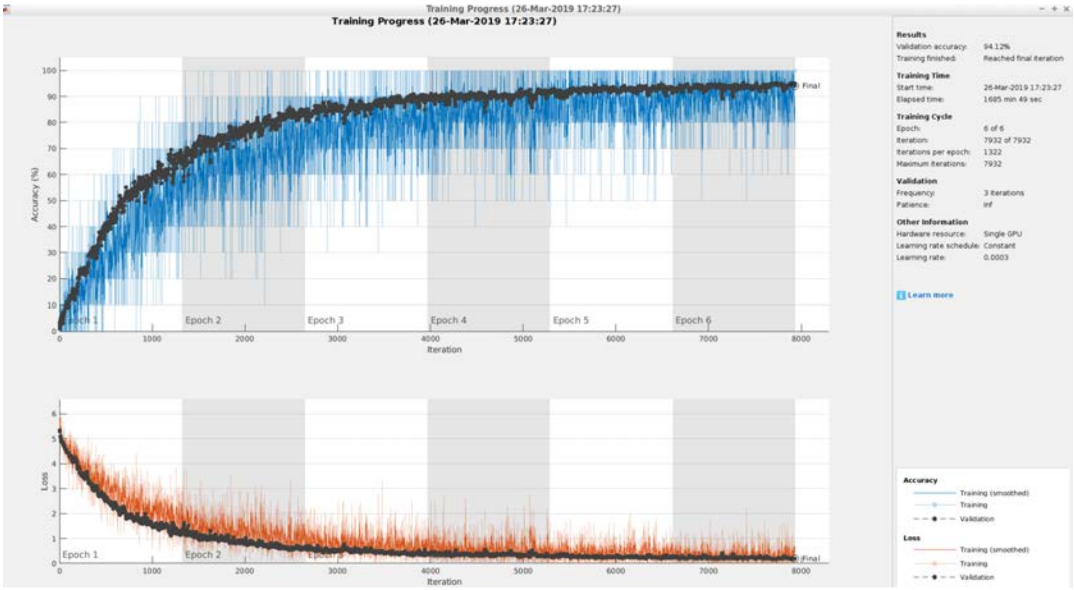


Figure A.2: Rank aggregation methods (Geometric mean, Robust Rank Aggregation and Stuart-Aerts) for each metric learning. It combines ResNet18_VPU, ResNet101_VPU and ResNet50_VPU. The higher mAP equals to 0.3228 is given by Geometric mean rank aggregation using NFST metric learning.

Appendix B

Fine-tunning graphs

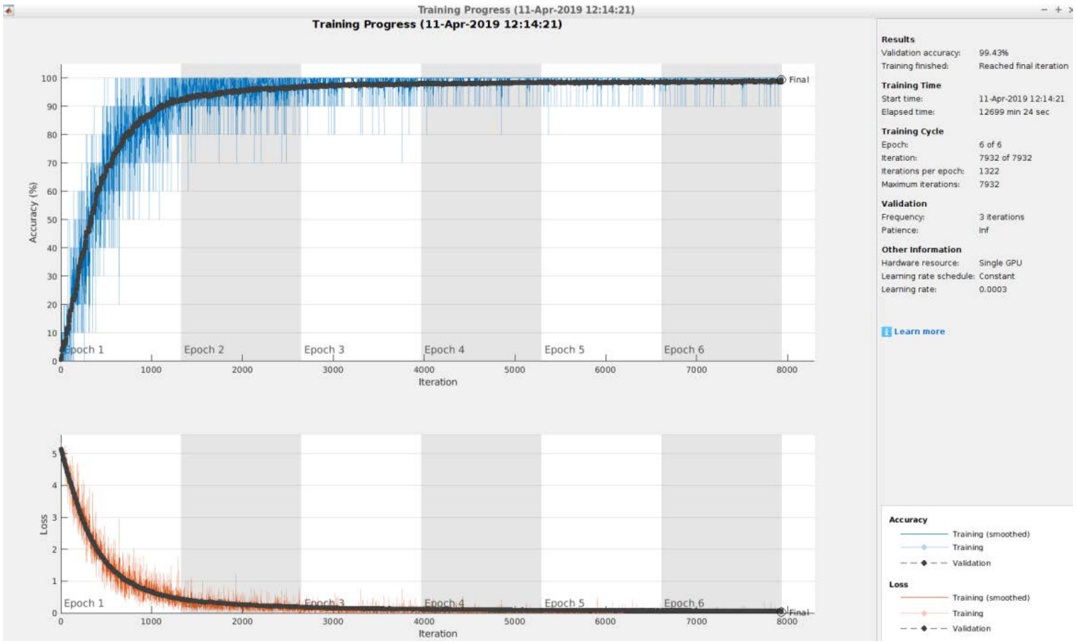
B.1. Training progress



Results

Validation accuracy:	94.12%
Training finished:	Reached final iteration
Training Time	
Start time:	26-Mar-2019 17:23:27
Elapsed time:	1685 min 49 sec
Training Cycle	
Epoch:	6 of 6
Iteration:	7932 of 7932
Iterations per epoch:	1322
Maximum iterations:	7932
Validation	
Frequency:	3 iterations
Patience:	Inf
Other Information	
Hardware resource:	Single GPU
Learning rate schedule:	Constant
Learning rate:	0.0003

Figure B.1: AlexNet fine-tuned progress. In black dots is represented the validation process. The upper figure represent the accuracy of the training process and the lower figure the loss.



Results
Validation accuracy: 99.43%
Training finished: Reached final iteration

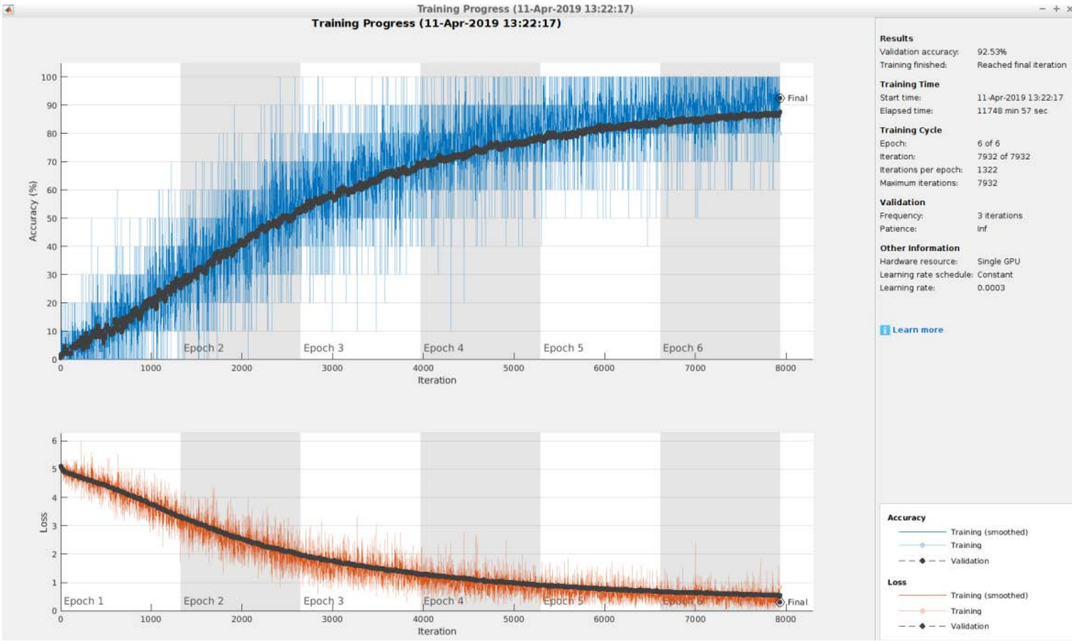
Training Time
Start time: 11-Apr-2019 12:14:21
Elapsed time: 12699 min 24 sec

Training Cycle
Epoch: 6 of 6
Iteration: 7932 of 7932
Iterations per epoch: 1322
Maximum iterations: 7932

Validation
Frequency: 3 iterations
Patience: Inf

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.0003

Figure B.2: DenseNet-201 fine-tuned progress. In black dots is represented the validation process. The upper figure represent the accuracy of the training process and the lower figure the loss.



Results
Validation accuracy: 92.53%
Training finished: Reached final iteration

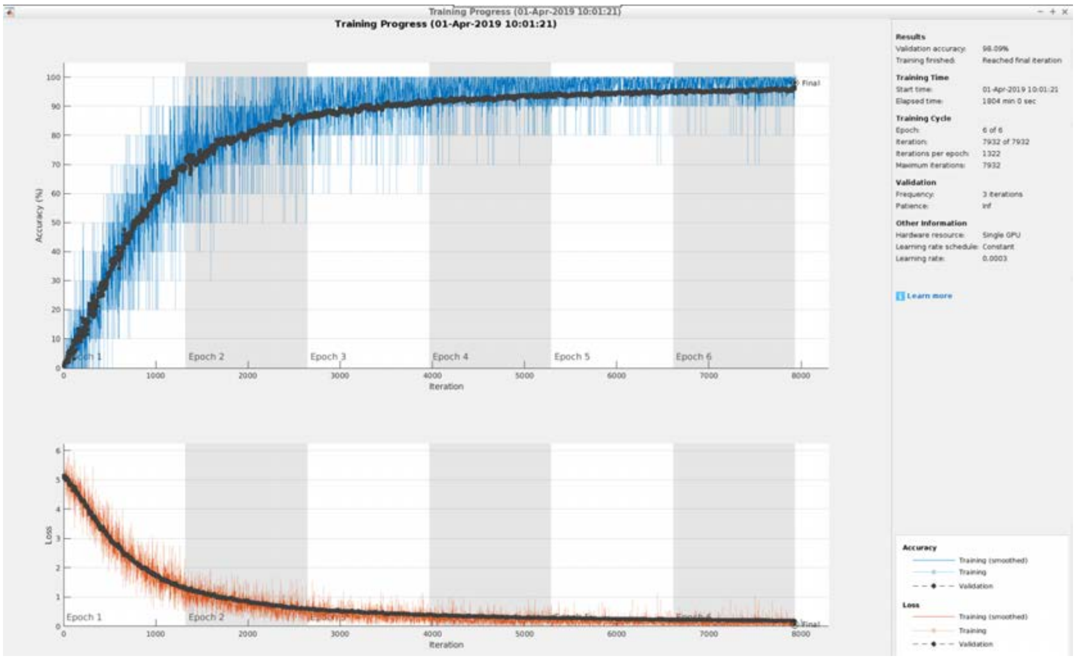
Training Time
Start time: 11-Apr-2019 13:22:17
Elapsed time: 11748 min 57 sec

Training Cycle
Epoch: 6 of 6
Iteration: 7932 of 7932
Iterations per epoch: 1322
Maximum iterations: 7932

Validation
Frequency: 3 iterations
Patience: Inf

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.0003

Figure B.3: Inception-ResNet-v2 fine-tuned progress. In black dots is represented the validation process. The upper figure represent the accuracy of the training process and the lower figure the loss.



Training Progress (01-Apr-2019 10:01:21)

Results
Validation accuracy: 98.09%
Training finished: Reached final iteration

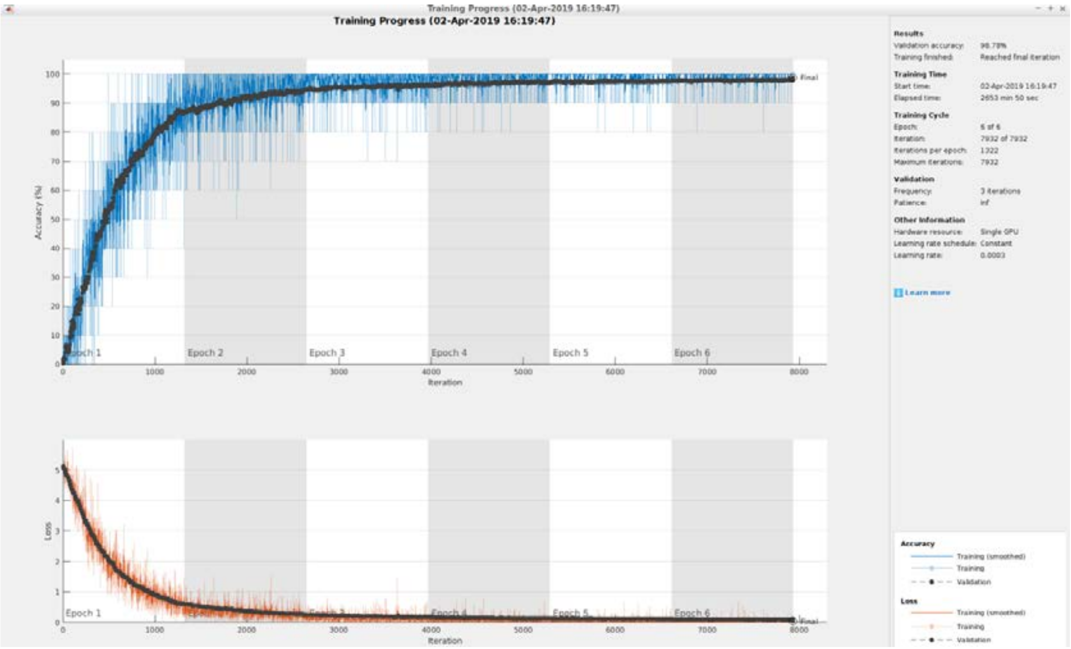
Training Time
Start time: 01-Apr-2019 10:01:21
Elapsed time: 1804 min 0 sec

Training Cycle
Epoch: 6 of 6
Iteration: 7932 of 7932
Iterations per epoch: 1322
Maximum iterations: 7932

Validation
Frequency: 3 iterations
Patience: inf

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.0003

Figure B.4: ResNet-18 fine-tuned progress. In black dots is represented the validation process. The upper figure represent the accuracy of the training process and the lower figure the loss.



Training Progress (02-Apr-2019 16:19:47)

Results
Validation accuracy: 98.78%
Training finished: Reached final iteration

Training Time
Start time: 02-Apr-2019 16:19:47
Elapsed time: 2653 min 50 sec

Training Cycle
Epoch: 6 of 6
Iteration: 7932 of 7932
Iterations per epoch: 1322
Maximum iterations: 7932

Validation
Frequency: 3 iterations
Patience: inf

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.0003

Figure B.5: ResNet-50 fine-tuned progress. In black dots is represented the validation process. The upper figure represent the accuracy of the training process and the lower figure the loss.