Deep Representation Learning for Vehicle Re-Identification

H. Aytaç Kanacı

Submitted for the degree of Doctor of Philosophy

School of Electronic Engineering and Computer Science

Queen Mary University of London

2021

Deep Representation Learning for Vehicle Re-Identification

H. Aytaç Kanacı

Abstract

With the widespread use of surveillance cameras in cities and on motorways, computer vision based intelligent systems are becoming a standard in the industry. Vehicle related problems such as Automatic License Plate Recognition have been addressed by computer vision systems, albeit in controlled settings (*e.g.*cameras installed at toll gates). Due to the freely available research data becoming available in the last few years, surveillance footage analysis for vehicle related problems are being studied with a computer vision focus. In this thesis, vision-based approaches for the problem of vehicle re-identification are investigated and original approaches are presented for various challenges of the problem.

Computer vision based systems have advanced considerably in the last decade due to rapid improvements in machine learning with the advent of deep learning and convolutional neural networks (CNNs). At the core of the paradigm shift that has arrived with deep learning in machine learning is feature learning by multiple stacked neural network layers. Compared to traditional machine learning methods that utilise hand-crafted feature extraction and shallow model learning, deep neural networks can learn hierarchical feature representations as input data transform from low-level to high-level representation through consecutive neural network layers. Furthermore, machine learning tasks are trained in an end-to-end fashion that integrates feature extraction and machine learning methods into a combined framework using neural networks.

This thesis focuses on visual feature learning with deep convolutional neural networks for the vehicle re-identification problem. The problem of re-identification has attracted attention from the computer vision community, especially for the person re-identification domain, whereas vehicle re-identification is relatively understudied. Re-identification is the problem of matching identities of subjects in images. The images come from non-overlapping viewing angles captured at varying locations, illuminations, etc. Compared to person re-identification, vehicle re-identification is particularly challenging as vehicles are manufactured to have the same visual appearance and shape that makes different instances visually indistinguishable. This thesis investigates solutions for the aforementioned challenges and makes the following contributions, improving accuracy and robustness of recent approaches. The contributions are the following: (1) Exploring the man-made nature of vehicles, that is, their hierarchical categories such as type (*e.g.*sedan, SUV) and model (*e.g.*Audi-2011-A4) and its usefulness in identity matching when identity pairwise labelling is not present (2) A new vehicle re-identification benchmark, Vehicle Re-Identification in Context (VRIC), is introduced to enable the design and evaluation of vehicle re-id methods to more closely reflect real-world application conditions compared to existing benchmarks. VRIC is uniquely characterised by unconstrained vehicle images in low resolution; from wide field of view traffic scene videos exhibiting variations of illumination, motion blur,

and occlusion. (3) We evaluate the advantages of Multi-Scale Visual Representation (MSVR) in multi-scale cross-camera matching performance by training a multi-branch CNN model for vehicle re-identification enabled by the availability of low resolution images in VRIC. Experimental results indicate that this approach is useful in real-world settings where image resolution is low and varying across cameras. (4) With Multi-Task Mutual Learning (MTML) we propose a multi-modal learning representation *e.g.*using *orientation* as well as *identity* labels in training. We utilise deep convolutional neural networks with multiple branches to facilitate the learning of multi-modal and multi-scale deep features that increase re-identification performance, as well as orientation invariant feature learning.

Submitted for the degree of Doctor of Philosophy

School of Electronic Engineering and Computer Science

Queen Mary University of London

2021

Declaration

I hereby declare that this thesis has been composed by myself and that it describes my own work. It has not been submitted, either in the same or different form, to this or any other university for a degree. All verbatim extracts are distinguished by quotation marks, and all sources of information have been acknowledged. Some works have been published or under review:

- A. Kanaci, X. Zhu and S. Gong. Vehicle Re-Identification by Fine-Grained Cross-Level Deep Learning. *In Proc. 5th Activity Monitoring by Multiple Distributed Sensing Workshop, British Machine Vision Conference, London, September 2017* (Chapter 3)

- A. Kanaci, X. Zhu, S. Gong. Vehicle Re-Identification in Context. *In Proc. German Conference on Pattern Recognition, Stuttgart, Germany, October 2018* (Chapter 4,5)

- G. Rajamanoharan, A. Kanaci, M. Li, S. Gong. Multi-Task Mutual Learning for Vehicle Re-Identification. *In Proc. IEEE Conference on Computer Vision and Pattern Recognition AI City Challenge Workshop, Long Beach, California, USA, June 2019* (Chapter 6)

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Prof. Shaogang Gong for his continued encouragement and enthusiastic supervision, especially during the witting up of this thesis. Secondly I specially send my tanks to my collaborators Dr. Xiatian Zhu and Dr. Minxian Li for valuable technical discussions.

I would like to thank my colleagues in the Vision Group for their companion and friendship during the PhD, especially; Wei Li, Umar Muhammed, Jifei Song, Qi Dong, Yanbei Chen and Zhiyi Cheng.

Finally I would thank my friends I made in London for their friendship and making living in London enjoyable during the stressful periods in the last 4 years. I am most thankful for the support of my family throughout the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Re-Identification

In Computer Vision understanding and interpreting an image, or a collection of images is the fundamental task at hand. These fundamental tasks are shaped by, informally, basic questions that can be asked to that image such as *what, where* and *who*. Respectively these questions lead to problems we have come to define as *classification, segmentation* and *recognition*. Re-Identification is an example of such a problem, where we are interested in interpreting an image of a certain known subject, such as person or face, and determining if the identity of the subject is known to the system.

The ubiquitousness of large-scale distributed multi-camera systems capturing continuously in public spaces (*e.g.*squares, public transport, highways) and in private spaces (*e.g.*buildings, factories, and supermarkets) in modern cities as well as digital imaging devices such as digital cameras (*e.g.*smartphones) in our everyday life create vast amounts of digital imaging data. Specifically in the case of surveillance setting, how the captured data can be used and utilized is an open research question and the need for analysing this visual data has become an interest in computer vision.

Let us first introduce the *re-identification* problem briefly to give an overview of the common characteristics of the problem and its setting. Re-identification (Re-ID) aims to identify identities among various camera views. The goal is to recognize identities, using face or the whole-body images, namely, *face* recognition and *person (whole-body)* re-identification.

The problem is usually set as the following; given a *probe* image, can we find the matching image with the same identity among the images in the *gallery* set? A *gallery* is a collection of images with a predefined(known, non-arbitrary) set of identities, and a *probe* is the query image input. Main challenges of this setting are building robust systems to discriminate identities under varying conditions *e.g.*different camera angles and different lighting conditions.

In the surveillance setting, recently, re-identification is being applied to *vehicle* data captured in cities and motorways. Next section will give more detail on vehicle re-identification.

1.1.1 Vehicle Re-Identification

Figure 1.1: Real world surveillance footage showcasing challenging vehicle re-identification scenario. Illumination differences make cross-camera matching extremely difficult. Resolution and viewing angles make number plate recognition infeasible.

In this thesis, we will focus on *vehicle* re-ID in the surveillance (CCTV) setting. See Figure 1.1 for example images showcasing typical traffic scenes. With the recent developments of autonomous driving and smart city applications, the need to accurately analyse vehicles on urban streets captured by distributed multi-camera systems with computer vision is ever-increasing. Specially, vehicle re-identification, solving the problem of associating vehicle identities across camera views at different locations and time, has attracted increasing attention in the research community [43, 44, 45, 62, 62, 75], as it can play an important role in intelligent transportation systems and public safety.

Vehicle re-identification aims to search a specific vehicle(s) instance(s) visually; across non-overlapping cameras (at different locations), captured at diverse camera views in uncontrolled environments potentially with various lighting and pose differences over a period of time. Particular instances can appear and disappear in a certain view and need to be associated in another view, with large differences in lighting conditions and pose at a different location and time, while

also needing to be differentiated visually from similar candidates.

With the change in subject to vehicles in the re-ID setting, domain specific challenges need to be taken into consideration. Similar to person re-ID, vehicle re-ID is also challenging because different vehicle identities have *very* similar visual appearance (low inter-class variance). Unlike person re-ID, the pose/orientation of vehicles results in occlusion and drastic visual geometry changes, since the vehicle is a rigid body (high intra-class variance). This means that it is difficult to infer the same identity from any given pose/orientation of a vehicle. Even in the same orientation, vehicles of different identities may look very similar due to being of the same, or similar, vehicle make/model. This requires vehicle re-ID models to have a more discriminative fine-grained recognition ability. In fact in vehicle re-id, the same vehicle model manufactured the same year would have identical visual appearance. This makes vehicle re-identification more challenging compared to person re-ID. See Figure 1.2. Hence vehicle re-identification is not a trivial task when we are concerned with matching images visually under these circumstances. Using number plates for matching is also problematic since only a limited percentage of images in surveillance settings can provide detectable and recognisable number plate information due to low resolution, non-frontal camera angles, and occlusions.

Figure 1.2: Inherent qualities of vehicles make vehicle re-ID an ill defined supervised learning problem where visually identical instances has different labels and same vehicle has captured at different location by different camera angle has dissimilar appearance. Examples images from VeRi-776[45] benchmark showcasing high *intra-class* variance and low *inter-class* variance.

1.2 Deep Learning and Convolutional Neural Networks

In machine learning *Deep Learning*[35, 16], a powerful set of techniques for learning in neural networks, revolutionized the usage of neural networks in many areas. Particularly in computer vision, fundamental problems such as object classification [32], object detection [14, 13] and semantic segmentation [49] have shifted the attention of researchers and has been shown to achieve significant in leaps state-of-the-art performance even surpassing human level performance for large scale classification tasks [19].

Practically, convolutional neural networks have been shown to recognize handwritten digits[34] in 90s however only recently novel machine learning techniques [48, 30, 26] allowed neural networks to be trained efficiently with improved generalization properties. CNNs are designed to be efficient for computer vision tasks compared to fully-connected networks with fewer parameters and fit well to the 2D structure of images with useful properties such as translation invariance.

Using CNNs and deep learning, an exiting possibility is to use end-to-end framework for feature learning[4]. Previous approaches in computer vision relied on hand crafted detectors and descriptors for feature extraction which then were used as input for machine learning algorithms. On the other hand end-to-end learning provides a framework where these two stages are combined and tightly coupled, meaning feature learning and the machine learning tasks are trained in the same framework at the same time.

1.3 Contributions

Firstly, we propose "Cross-Level Vehicle Recognition" (CLVR) method that uses a vehicle model classification approach and repurposes the classification model for the vehicle re-identification task. This approach learns a fine-grained classification model using vehicle model labels (*e.g.*Ford Focus, Audi A4) and the learned model is tested with the identity labels for vehicle re-identification. *Finer*-grained vehicle instance re-identification task requires expensive and time-consuming cross-camera identity pairwise labelling, and this approach uses easier to obtain model labels to perform the instance re-identification task. This cross-level matching scheme is significantly different from existing methods that typically rely on the availability of identity instance annotations/labels for the re-ID problem and showcases an alternative approach with usable re-identification accuracy where identity annotations are not available.

With current vehicle re-ID benchmarks, VehicleID [43] and VeRi-776 [45], being limited

in their real world applicability, we introduce a new benchmark dataset called **Vehicle Re-Identification in Context** (VRIC[1]) for more realistic and challenging vehicle re-identification. VRIC dataset aims to provide a more realistic vehicle re-ID evaluation benchmark that contains vehicle images of *unconstrained* visual appearances with variations in resolution, motion blur, weather, and occlusion. See Figure 1.1 for video frame samples showing realistic settings with challenging aforementioned difficulties.

Following our focus on challenging inputs for vehicle re-identification with VRIC benchmark, especially with low resolution images, we further investigate a Multi-Scale Vehicle Representation (MSVR) learning model to address re-identification performance on low resolution input. MSVR is a multi-branch CNN architecture for re-identification that aims to learn resolution invariant models focusing performance on low resolution images. This is done with a augmenting the learning for each branch with a consensus signal during the training process. Extensive comparative evaluations demonstrate the effectiveness of the proposed MSVR method in comparison to the state-of-the-art vehicle re-ID techniques on the two existing benchmarks (VehicleID [43] and VeRi-776 [45]) and the newly introduced VRIC benchmark. The inherent low resolution input captured from wide-view traffic scenes can be better evaluated in VRIC benchmarks as current available datasets contains mostly high resolution images.

Lastly, we propose an original *Multi-Task Mutual Learning* (MTML) method that improves upon the MSVR model by addressing the high intra-class variance matching that arises from vehicle orientation at the time of capture. This approach is motivated by the observation that low inter-class variance between identities results in poor performance when similar vehicles captured with the same orientation are present in the gallery set. This scenario hinders the re-ID performance as incorrect identities are ranked higher in the results. MTML uses additional orientation labels to learn orientation aware model weights by utilising multi-modal learning from identity and orientation supervisory signals together. MTML is similar to MSVR in its use of multi-branch CNN architecture that simultaneously learns the re-identification task from different supervisory signals, plus a consensus loss function, to build an improved representation for the purpose of vehicle re-identification.

[1]Available at `http://qmul-vric.github.io`

1.4 Thesis Structure

The thesis is organised as the following. *Chapter 2* describes the re-identification problem as a similarity learning problem and details the state-of-the-art literature in both person and vehicle re-identification as well as current benchmarks. *Chapter 3* discusses fine-grained vehicle model classification, *Chapter 4* introduces Vehicle Re-Identification in Context (VRIC) benchmark, *Chapter 5* proposes the Multi-Scale Vehicle Representation (MSVR) approach. *Chapter 6* further improves the multi-branch learning with *Multi-Task Mutual Learning* (MTML) model to incorporate vehicle orientation information with resolution invariant learning. *Chapter 7* concludes the thesis.

Chapter 2

Literature Review

2.1 Computer Vision and Machine Learning

In recent years, machine learning techniques have advanced significantly due to prevalent *deep learning* research. Deep Learning[35] showed that *Artificial Neural Networks* are powerful high level feature extractors from raw input when neurons are organised as consecutive layers, transforming the input to high-level features as the network becomes *deeper.* Currently, modern neural network *models* can contain hundreds of layers [20] and such models are called *Deep Neural Networks* giving rise to the popularity of the term Deep Learning. Deep learning has proven to be shown the most performant approach not only in computer vision [32, 53, 57, 20] but many other areas of research domains such as machine translation [65] and voice recognition [22]. Such *deep* neural networks take advantage of the capabilities offered by modern graphical processing units (GPU) to train large neural networks with millions of parameters.

Image understanding is the core interest in Computer Vision and recognition(identification) is one of the most studied to in modern computer vision. Researchers have formulated typical scenarios where determining an identity of humans in different modalities such as face *detection, verification* and person *re-identification*. The topics have attracted interest not only because they propose useful for applications such as security and behavioural analysis but also they help arrive at better understandings of problems that are at the core of computer vision. Particularly in computer vision, fundamental problems closely related to classification[32], such as object detection [14, 13] and semantic segmentation [49] as well as low level image processing tasks e.g. edge de-

tection and optical flow have attracted attention from researchers and has been shown to achieve significant in leaps state of the art performance, even surpassing human level performance for large scale classification tasks[19], when using modern neural networks.

2.1.1 Supervised Learning for Computer Vision

Classification

Classification is a process of categorizing an input as one of the given set of classes. For example, handwritten digit recognition classifies given input as one of the digits, 0-9. LeCun *et al.* [34] introduced a convolutional neural network(LeNet5) and showed neural networks can be used for classification handwritten digits successfully. Modern CNNs use almost identical frameworks as this seminal work, adding modern modifications that make learning more efficient with better representation learning capabilities. ImageNet[10] visual recognition challenge provided the first large scale dataset for computer vision tasks. For training, ImageNet provides 1.2 million images with 1000 category labels. At this scale, the computer vision models trained on this dataset can be used as a general purpose computer vision model for visual classification tasks. Krizhevsky *et al.* [32] introduced the AlexNet model with significantly more parameters than previously used. This model performed significantly better than traditional computer vision methods by winning 2012 ImageNet[10] challenge, with 16% error surpassing the closest competitor at 26% error by a large margin, and cementing deep learning approach in computer vision. This modern model successfully uses pooling and strided convolutions with ReLU activations, compared to sigmoid activations in LeNet5. Following the success of AlexNet, VGG[6] models improved the performance by utilising 3×3 kernels in all convolutional layers of the network, showcasing simpler uniform structure can provide state-of-the-art performance. Following this, more complex convolutional layer structures with parallel data processing as in Inception[56], residual connections and bottleneck layers in ResNet[20] as well as BatchNorm[57] layers that allow deeper CNN models with significantly more layers to be trained quickly and efficiently.

Fine Grained Visual Classification

When the provided label for a dataset identifies subcategories of a certain type of subject of interest *e.g.*species of dogs[29], birds [60] etc. we are dealing with a similar task but with finer granularity of information provided by labels. Convolutional neural networks have been proving to be quite performant in this setting without requiring the learning framework to be changed.

This is the power of neural networks in supervised learning approaches. In fine grained visual classification, standard classification network models can be utilised with good performance however task-specific modifications [41, 27] do increase performance.

One closely related problem to vehicle re-identification is vehicle model classification [39, 18, 42, 69, 54, 25]. But, the two problems are usually studied independently. For example, Yang et al. [69] propose a part attributes driven vehicle model recognition. They also contribute a large comprehensive car dataset named "CompCars" with model class labels but without vehicle identity labels.

Verification

A facial recognition system is a technology capable of matching a human face from a digital image or a video frame against a database of face. As the faces are coming from a predefined set, this problem has almost the same setting of classification in terms of machine learning approach, but the granularity of information provided by the dataset labels are identities rather than categories or subcategories. The ground truth should come in pairs of images denoting their identity match or otherwise. Then the verification score is given by the percentage of pairs that correctly predicts if the image pair has the same identity or not. Both re-identification and verification uses very same loss functions during model training [51, 64] as the approaches are quite similar with just a small difference in test setting.

2.2 Re-Identification

2.2.1 Problem Definition

In computer vision *Re-Identification* is problem of matching identities visually. The matching is usually done between different viewing angles, and/or lighting conditions of the identities, e.g. matching of images that are captured from different surveillance cameras in a building. In a sense, it can be considered as a case of classification problem where classes are instances(identities). However having all identities available to us is neither feasible nor preferred for tackling this problem. For this reason re-identification problem is formulated in the following way. Similar to classification, it is a supervised learning problem. We have multiple examples of data for each identity, preferably from various capture conditions and angles, in the training set. Each example is labelled with identity information. However the train set and test set are disjoint *i.e.*contain

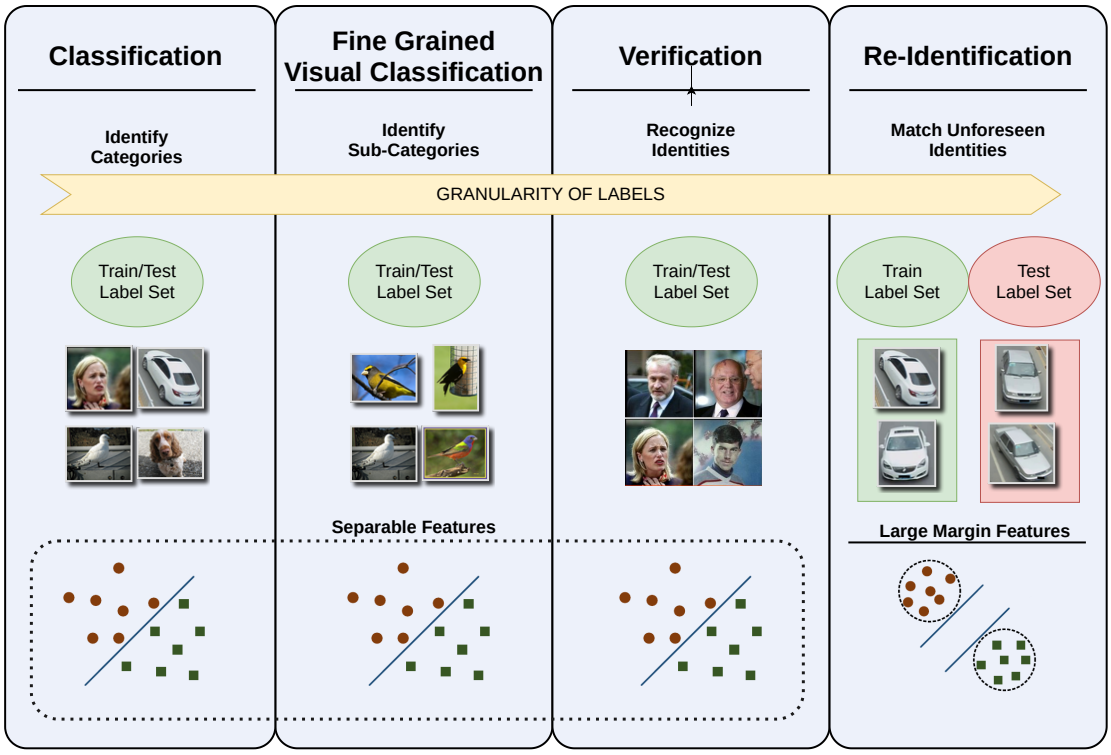

Figure 2.1: Overview of the supervised learning tasks and key differences in problem settings regarding data collection and inherent goal of the visual representation learning.

different set of identities. Meaning during testing time, test classes are unseen compared to training time. Figure 2.1 highlights the differences between re-identification and related problems and shows their general characteristics in terms of the images in the datasets and the provided labels.

2.2.2 Testing

Traditionally, for person and face re-ID each dataset had slightly different evaluation settings. This is still valid but in time, as the field matured common characteristics emerged. Images in the test classes are separated in two categories: probe and gallery. Note that all test identities are present in probe and gallery sets. Probe images are query images *i.e.*images that are to be matched. For each probe image, the images in the gallery set are ranked(sorted) according to a similarity metric.

The re-ID score is given by the percentage of the *rank-n* images (first *n* images after sorting) that have the same label as the probe image e.g. *rank-1* gives the score by the most similarly sorted image in the gallery set. This score is reported as the percentage of the all probe images that satisfy *rank-n*, where *n* is usually set as 1 and 5. Moreover a Cumulative Matching Char-

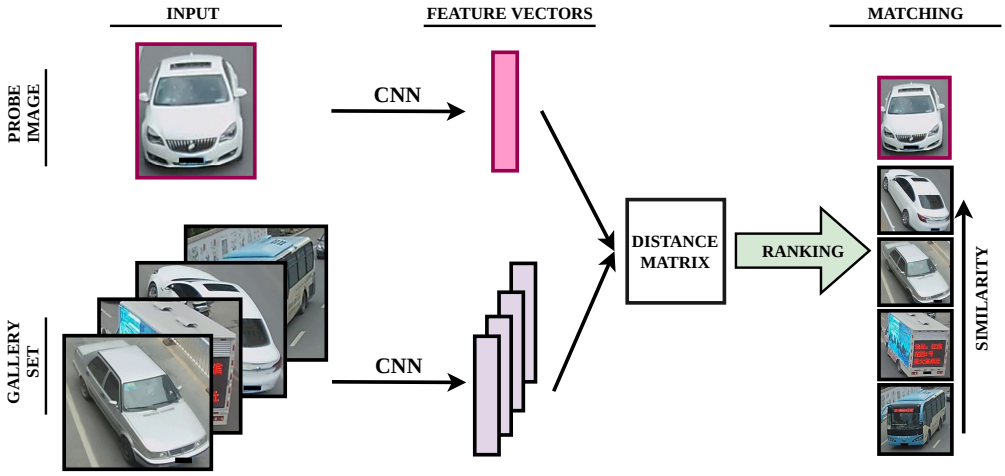

Figure 2.2: Overview of the re-identification evaluation using learned CNN model to extract feature vector representation and ranking.

acteristic(CMC) can be reported showing all the percentages of *rank-1* to *rank-N*, where *N* is a relatively large natural number i.e. 50. Figure 2.2 shows the general framework of the testing in re-identification.

2.3 Vehicle Re-Identification

Whist vehicle re-ID is less studied than person re-id [15, 36, 2, 66, 37, 55, 38, 74], there are a handful of existing methods. Notably, Feris *et al.*[12] proposed an attribute-based re-ID method. The vehicles are firstly classified by different attributes like vehicle model and colours. The re-ID matching is then conducted in the attribute space. Dominik *et al.*[70] used 3D bounding boxes for rectifying car images and then concatenate colour histogram features of vehicle image pairs. A binary linear SVM model is then trained to verify whether a pair of images have the same identity. Both methods rely heavily on hand-crafted visual features and consecutive machine learning training that result in poor performance with weak model generalisation.

More recently, deep learning techniques have been used for vehicle re-identification. Liu *et al.*[45] explored a deep neural network to estimate the visual similarities between vehicle images. Liu *et al.*[43] also designed a Coupled Cluster Loss (CCL) to boost a multi-branch CNN model for vehicle re-id. All these methods utilise the global appearance features of vehicle images and ignore local discriminative regions. To explore local information and motivated by the idea of landmark alignment [71] in both face recognition [58] and human body pose estimation [47], Wang *et al.*[62] considered 20 vehicle keypoints for learning and aligning local regions of a

vehicle for re-id. Clearly, this approach comes with extra cost of exhaustively labelling these keypoints in a large number of vehicle images, and the implicit assumption of having sufficient image resolution/details for computing these keypoints.

Additionally, space-time contextual knowledge has also been exploited for vehicle re-ID subject to structured scenes [45, 52]. Liu *et al.*[45] proposed a spatio-temporal affinity approach for quantifying every pair of images. Shen *et al.*[52] further incorporated spatio-temporal path information of vehicles. Whilst this method improves the re-ID performance on the VeRi-776 dataset, it may not generalise to complex scene structures when the number of visual spatio-temporal path proposals is very large with only weak contextual knowledge available to facilitate model decision.

2.4 Datasets

Liu *et al.* [43] introduced the "VehicleID" benchmark with a total of 221,763 images from 26,267 IDs. In parallel, Liu *et al.* [44] created "VeRi-776", a smaller scale re-ID dataset (51,035 images of 776 IDs) but with space-time annotations among 20 cameras in a road network. Recently, Yan *et al.*[68] presented two large scale datasets (846,358 images of 141,756 IDs in "VD1", 690,518 images of 79,763 IDs in "VD2") with constrained viewing angle visual characteristics similar to VehicleID. Finally, CityFlow[59] with 56,277 bounding boxes in total, where 36,935 of them from 333 object identities form the training set, and the test set consists of 18,290 bounding boxes from the other 333 identities. CityFlow is the most challenging addition to vehicle re-identification datasets as it was captured from from 40 cameras across 10 intersections, with the longest distance between two simultaneous cameras being 2.5 km.

Whilst existing benchmarks have contributed significantly to the development of vehicle re-ID methods, they only represent *constrained* test scenarios due to the rather artificial assumption of having high-quality images of constant resolution. This makes them limited for testing the true robustness of re-ID matching algorithms in typically *unconstrained* wide-view traffic scene imaging conditions. The VRIC benchmark introduced in this work addresses this limitation by providing a vehicle re-ID dataset with more realistic imaging quality from unconstrained viewing conditions giving rise to changes in resolution, motion blur, weather, illumination, and occlusion. A summary of the dataset statistics are available in Table 2.1.

VehicleID

Figure 2.3: Sample images of collected images from the VehicleID dataset.

The VehicleID [43] dataset contains data captured during daytime by multiple real-world surveillance cameras. It contains 221,763 images of 26,267 vehicles in total and 78,982 images, which have been used for model training, are labelled with three attributes: colour, vehicle model and vehicle ID. There are three gallery sets containing 800, 1600 and 2400 unique identities. Colour and vehicle model can be used as coarse attributes which can be useful for 7 distinct colours, 250 distinct models in total. The train and test split identities are disjoint sets, however each identity is only captured from single viewpoint. In this regard, it is different from usual person re-ID datasets and it is not suitable for *multi*-camera re-identification evaluation. See Figure 2.3 for example images.

VeRi-776

Introduced in [45], Veri-776 contains over 50,000 images of 776 vehicles captured by 20 cameras covering a 1 km^2 area in 24 hour period. Images are also labelled with attributes, namely, vehicle type and colour. Each vehicle is captured by 2 to 18 cameras at different locations. The database also provides 9000 vehicle tracks.

The training set has 576 vehicles with 37,778 images and the testing set has 200 vehicles with 11,579 images. Also in [45] a different evaluation method is introduced other than the image-to-image ranking. The database provides 1678 query images and 2021 testing tracks for evaluating

Table 2.1: Summary of Characteristics of existing vehicle re-identification datasets.

Dataset	Images	IDs	Cameras	Motion Blur	Illumination	Occlusion
VehicleID [43]	113,123	15,524	-	No	Limited	No
VeRi-776 [45]	51,034	776	20	No	Limited	No
CityFlow [59]	56,277	666	40	No	Limited	No
VRIC	60,434	5626	120	Unconstrained	Unconstrained	Unconstrained

image-to-track scenario which authors say is closer to the real world scenario. Compared to VehicleID this dataset is relatively small but has images that are captured from multiple cameras of the same identity, thus it is the first true vehicle re-ID dataset.

CityFlow

The most recent vehicle re-identification benchmark, CityFlow [59], a city-scale traffic camera dataset consisting of more than 3 hours of synchronised HD videos from 40 cameras across 10 intersections, with the longest distance between two simultaneous cameras being 2.5 km. CityFlow has large-scale spatial coverage and with large number of cameras/videos in an urban environment. The dataset contains more than 56,277 annotated bounding boxes covering a wide range of scenes, viewing angles, vehicle models, and urban traffic flow conditions.

The CityFlow dataset [59] has 36,935 images of 333 IDs in the training set and 333 different IDs in the test set. The standard probe and gallery sets consist of 1,052 and 18,290 images respectively. Baseline benchmarks tested on CityFlow are still lower than 35% in mAP. In comparison, performance of the same method on other public benchmarks, using the same implementations and hyper-parameters, is significantly better, thus verifying that CityFlow is indeed more challenging [59].

2.5 Discussion

Convolutional Neural Networks [32, 20], have proven to be the most effective tool we have in our disposal that can achieve high performance accuracy in most computer vision tasks. This is due to the powerful high-level features learned by *deep* neural networks with many weight parameters. Following recent developments in the state-of-the-art vehicle re-identification literature, this thesis explores many aspects of training and deploying vehicle re-identification models that can

learn discriminative visual representations for challenging input data. Approaches presented here have improved performance in challenging situations such as low resolution images, illumination and pose variance.

Approaches to vehicle re-identification in this thesis are laid out in the following chapters. **Chapter 3** Vehicle re-identification in unconstrained images is a challenging computer vision task due to the subtle visual appearance discrepancy between different identities and large visual appearance changes of the same vehicle instance. Existing methods rely on the availability of pairwise annotations. In this work, we question the necessity of costly pairwise identity labelling by exploiting the inherent hierarchical nature of vehicle identity and vehicle model type so as to eliminate the need of identity level label collection. Specifically, we propose to transfer the fine-grained vehicle model discriminative representation for instance re-ID matching task by leveraging the discriminative capacity of fine-grained classification at the vehicle model level. This realises "Cross-Level Vehicle Recognition" (CLVR). Extensive comparative experiments demonstrate the effectiveness of the proposed CLVR method compared to state-of-the-art approaches of using fine-grained identity pairwise labels on the largest vehicle re-ID benchmark dataset.

Chapter 4 Existing vehicle re-identification (re-id) evaluation benchmarks consider strongly artificial test scenarios by assuming the availability of high quality images and fine-grained appearance at an almost constant image scale, reminiscent to images required for Automatic Number Plate Recognition, *e.g.*VehicleID benchmark. Such assumptions are often invalid in real-world vehicle re-id scenarios where arbitrarily changing image resolutions (scales) are the norm. This makes the existing vehicle re-ID benchmarks limited for testing the true performance of a re-ID method. We introduce a more realistic and challenging vehicle re-ID benchmark, called Vehicle Re-Identification in Context (VRIC). In contrast to existing datasets, VRIC is uniquely characterised by vehicle images subject to more realistic and unconstrained variations in scale (resolution), motion blur, illumination, occlusion, and viewpoint. It contains 60,434 images of 5,626 vehicle identities captured by 60 different cameras at heterogeneous road traffic scenes in both day-time and night-time.

Chapter 5 Given the nature of this new benchmark VRIC, we further investigate a matching approach to vehicle re-ID by learning a multi-scale discriminative feature representations from varying resolution images. Compared to existing datasets, resolution of images in VRIC are more

than 50% smaller in width and height on average. Specifically, we design a consensus learning loss function by fusing features from multiple convolutional network branches to get more robust features for low resolution image matching. In our experiments, we will showcase the efficacy of using multi-branch multi-scale convolutional neural network design to learn models that can perform better on vehicle re-identification, increasing the matching performance for image pairs of varying resolution.

Chapter 6 Another prominent challenge of vehicle re-ID is that the visual appearance of vehicles may drastically change according to diverse viewpoints and illumination. Most existing vehicle re-ID models cannot make full use of various complementary vehicle information, e.g. vehicle type and orientation. In this paper, we propose a *Multi-Task Mutual Learning* (MTML) deep model to learn discriminative features, simultaneously from multiple branches similar to the previous chapter but also using additional label supervision, *i.e.*combining identity level label supervision at multiple scales with extra vehicle orientation information. Extensive comparative evaluations demonstrate the effectiveness of our proposed MTML method in comparison to the state-of-the-art vehicle Re-ID techniques on a large-scale benchmark dataset, VeRi-776 and CityFlow.

Chapter 3

Fine-grained Vehicle Model Classification for Vehicle Instance Re-ID

3.1 Introduction

Motivated by the extensive work on person re-ID and the capacity of deep neural network models learning from large scale training data, recent vehicle re-ID methods are typically designed to learn a discriminative deep feature representation [43, 45] where images have identity labels available. Existing methods often rely on the availability of cross-camera identity annotations collected by exhaustive human labelling. This approach is not scalable to many real-world conditions as cross-camera labelling of vehicle reappearance is costly as well as time consuming. In this work, we investigate the usefulness of fine-grained vehicle model classification for even more *finer*-grained vehicle instance search and re-identification without the need of cross-camera vehicle identity label supervision in training. The motivation for this approach comes from the observations: **(1)** Vehicle identity is intrinsically associated with the vehicle model classes, in other words, the model and identity labels form a hierarchical relation. **(2)** Relative to the identity labels, the vehicle model labelling is much *easier to collect* without the need for cross-camera vehicle annotation, although less fine-grained with weaker supervision information. **(3)** There are many different vehicle models with small visual appearance differences *e.g.*different models from the same manufacturer or slight styling changes that are made every year for the same model. This means that model class labels are already *fine-grained* and potentially provide notable discriminative information relevant to vehicle re-ID task.

Figure 3.1: Illustration of vehicle re-identification challenges: **(First Row)**: Stark visual similarities between different vehicle instances with same model (low inter-class variance); **(Second Row)**: Significant variations in visual appearance due to variation of illumination, view-angle for a given identity label (high intra-class variance).

3.2 Modelling Framework

3.2.1 Deep Convolutional Neural Networks

This section will take a closer look at the CNN architectures in similarity learning for re-identification problem. As mentioned previously, state-of-the-art re-identification architectures mostly take inspiration from tried and tested deep neural networks for visual image classification tasks benchmarked with ImageNet challenge [10]. Architectures like [32, 53] provided very successful incarnations of CNN architectures for classification and these types of architectures are almost always used as-is with minor modifications in re-ID setting. These modifications are what allows researchers to derive the learned features of similar and dissimilar examples in the feature space. This, in practice, means choosing appropriate loss functions to derive the learning procedure to end up with features that are robust to low inter-class variance and high intra-class variance. For an illustration of this challenge see Figure 3.1.

Formally, a CNN learns a mapping f. Each input (image) $\boldsymbol{x}$ can be transformed in to feature vector $\boldsymbol{z}$ and $\boldsymbol{W}$ is the weights of the network:

$$\boldsymbol{z} = f(\boldsymbol{x}|\boldsymbol{W}) \tag{3.1}$$

CNN models consist of layers that are stacked up one after the other. State-of-the-art architectures consist of two types of layers in two distinct stages. The first stage after the input consists of *convolutional* layers and the second stage consists of *fully-connected* layers. The first stage is analogous to a feature extractor for the image with mainly convolutional layers resulting

in feature vector $\boldsymbol{z}$ and that is fed to the second stage, a linear classifier made of fully-connected layers. This approach is referred as *end-to-end* learning, as starting from raw input including the last classification layer, the network parameters are updated in end-to-end fashion, providing updates to both feature extraction and classification parameters at each step in the training procedure. In other words previously distinct steps in computer vision, namely, feature extraction and classifier training are unified in and end-to-end framework.

3.2.2 Loss Functions for Re-Identification

Cross-Entropy Loss

In a classification problem with C classes, we can use the *cross-entropy* between ground-truth label and the model's estimated predictions as the cost function, thus we minimise the *negative log-likelihood* of the predictions of the model f. This is the standard approach for learning discriminative models in classification. We can use the same approach in re-identification by treating identity labels as the classes when training re-ID models.

Formally, given a training image $\boldsymbol{x}$, and ground truth one-hot label vector $\boldsymbol{y} \in \mathbb{R}^C$, we define the estimated posterior probability distribution vector $\hat{\boldsymbol{p}} \in \mathbb{R}^C$ of $\boldsymbol{x}$ by converting the models output logits $\boldsymbol{z} \in \mathbb{R}^C$ (see Eq 3.1) to a valid probability distribution using the softmax function. Specifically the j-th class posterior probability $\hat{p}_j$ is written as:

$$\hat{p}_j = P(y_j = y|\boldsymbol{x}) = \frac{\exp(\boldsymbol{W}_j^\top \boldsymbol{z})}{\sum_{k=1}^{C} \exp(\boldsymbol{W}_k^\top \boldsymbol{z})} \tag{3.2}$$

where y is the ground truth label and $\boldsymbol{W}_k$ the prediction function parameter of model class k.

The negative log likelihood $-\log P(\mathbf{y} \mid \boldsymbol{x})$ or equivalently the cross-entropy $\mathcal{L}_{CE}$ between one-hot vector label encoding $\mathbf{y}$ and model's prediction $\hat{\boldsymbol{p}}$ over C classes is computed by:

$$\mathcal{L}_{CE} = \sum_{j=1}^{C} -y_j \log \hat{p}_j \tag{3.3}$$

Triplet Loss

Triplet loss is the standard technique in a metric learning system where we know some examples should lie closer to certain examples than others in a feature space. This approach have been used successfully in deep learning applications[51, 21]. Formally, we have the triplets of input selected from the training set, namely $\mathbb{X} = \{< \boldsymbol{x}_a^{(i)}, \boldsymbol{x}_p^{(i)}, \boldsymbol{x}_n^{(i)} >\}_{i=1}^{N}$ namely *anchor, positive* and *negative* input images as denoted by the superscripts; where $\boldsymbol{x}_a^{(i)}$ and $\boldsymbol{x}_p^{(i)}$ belong to the same identity while $\boldsymbol{x}_a^{(i)}$ and $\boldsymbol{x}_n^{(i)}$ belong to different identities. The feature vectors $< \boldsymbol{z}_a^{(i)}, \boldsymbol{z}_p^{(i)}, \boldsymbol{z}_n^{(i)} >$

must satisfy the following constraint:

$$||\boldsymbol{z}_a^{(i)} - \boldsymbol{z}_p^{(i)}||_2^2 + \alpha \geq ||\boldsymbol{z}_a^{(i)} - \boldsymbol{z}_n^{(i)}||_2^2 \tag{3.4}$$

after the training of the model. Loss function is defined as the sum of all triplet's *hinge* loss:

$$\mathcal{L} = \sum_{i=1}^{N} max\{||\boldsymbol{z}_a^{(i)} - \boldsymbol{z}_p^{(i)}||_2^2 + \alpha - ||\boldsymbol{z}_a^{(i)} - \boldsymbol{z}_n^{(i)}||_2^2, 0\} \tag{3.5}$$

Hinge loss penalises the examples that are not similar enough by theamplthe margin α. If the distance to the positive example is bigger than to the negative example plus the margin, we have a positive loss, otherwise the loss is 0 for that triplet.

When using triplet loss, one concern is that which triplets are to be chosen during training, namely *triplet sampling*. Uniformly sampling the triplets from the whole dataset is not desired because the number of triplets are in the order of $O(n^3)$ and most triplets will easily satisfy the triplet constraint. It is essential to choose triplets that are more informative so that the training converges quickly with efficient learning. Two approaches have emerged. First option is pausing the training process and mining the most informative triplets using up-to-date feature vectors as in [51, 61]. Second, choosing the most informative triplets from the current minibatch, a minibatch specifically designed for re-identification as in [21].

Coupled Cluster Loss

This is a variation of the triplet loss idea, it is relevant because it was used recently to achieve best re-ID results on the VehicleID dataset in [43]. Here two sets of positive and negative examples are chosen and the anchor chosen from the positive set. Then the loss is calculated between the mean of the positive examples and the closest negative example to anchor. Figure 3.2 shows a visualisation. For N positive, M negative samples: $\{\boldsymbol{x}_p^{(i)}\}_{i=1}^{N}, \{\boldsymbol{x}_n^{(i)}\}_{i=1}^{M}$ the loss is defined as:

$$\mathcal{L} = \sum_{i=1}^{N} \frac{1}{2} max\{||\boldsymbol{z}_p^{(i)} - \boldsymbol{c}||_2^2 + \alpha - ||\boldsymbol{z}_{n^*}^{(i)} - \boldsymbol{c}||_2^2, 0\} \tag{3.6}$$

where $\boldsymbol{c} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{z}_p^{(i)}$, $\boldsymbol{z}_{n^*}^{(i)}$ is the closest negative sample to $\boldsymbol{c}$, and α the margin. This formulation helps mean feature vector of the positive images in a batch to form clusters and triplet sampling becomes less complicated.

Contrastive Loss

This loss was introduced in [9] and was shown in an application to face verification. Here the network is trained in pairs of inputs instead of triplets and a *binary* label $y \in \{0, 1\}$ denotes if the

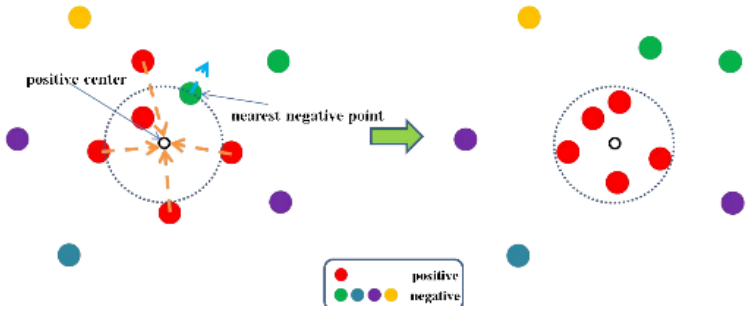

Figure 3.2: Coupled Cluster Loss

pairs are from the same class or not. $D_W(\boldsymbol{x}_1,\boldsymbol{x}_2) = ||\mathbf{z}_1 - \mathbf{z}_2||_2$ is the distance between the feature representations of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. And the loss function for the tuple $\boldsymbol{x}_1,\boldsymbol{x}_2$ and the binary label Y is defined as:

$$L(y,\boldsymbol{x}_1,\boldsymbol{x}_2) = (1-y)L_S + yL_D$$

where L_S and L_D denotes partial losses for similar and dissimilar pairs. The binary label y effectively cancels out either L_S and L_D by multiplication. The exact loss function is:

$$L(y,\boldsymbol{x}_1,\boldsymbol{x}_2) = (1-y)\frac{1}{2}(D_W)^2 + y\frac{1}{2}max(0,m-(D_W)^2)$$

where m is the margin of similarity imposed by the training. As triplet loss inherently enforces a three-branch structure, contrastive loss enforces a two-branch structure conceptually. However, in practice, one branch is sufficient as the branches share the same weights and input pairs can be processed sequentially.

Center Loss

Introduced in [64] center loss is used in conjunction with the standard cross-entropy loss used in classification. Using this loss is as simple as changing the loss function and no complicated triplet sampling is necessary. The loss keeps track of all the classes' feature mean(center) and works to keep these centers separated in the feature space.

$$L_C = \frac{1}{2}\sum_{i=1}^{m}||z_i - c_{y_i}||_2^2$$

The centers c_{y_i} are updated with each minibatch and cross-entropy loss, L_S, and center loss L_C are summed $L = L_S + \lambda L_C$. The scalar λ is used to weight the two loss functions.

Figure 3.3 shows the effect of the hyper parameter λ when center loss is used to classify the MNIST dataset.

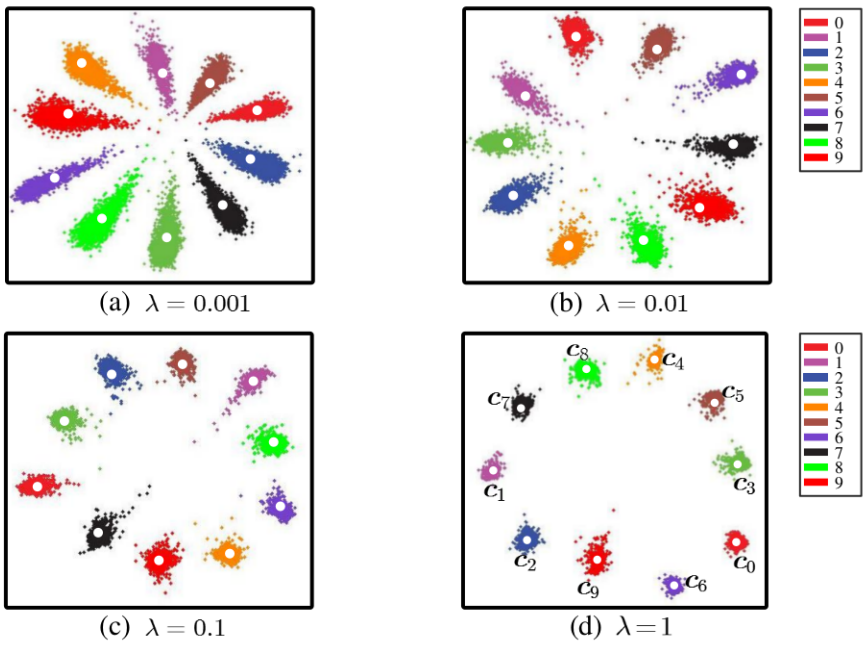

Figure 3.3: The distribution of deeply learned features under the joint supervision of soft- max loss and center loss. The points with different colours denote features from different classes. The white dots $(c_0, \cdots, c_9)$ denote 10 class centers of deep features for each label (digits 0–9) in MNIST dataset.

3.3 Cross-Level Vehicle Recognition

Inspired by the recent work done on vehicle model classification benchmark CompCars [69] and more generally, the application of deep convolutional neural networks [41, 27] on fine-grained recognition tasks [29, 60], we explore a similar learning paradigm for vehicle re-ID, using vehicle model labels instead of identity labels for learning. Specifically, we propose to utilise the vehicle model discriminative representation for more fine-grained identification task through our "Cross-Level Vehicle Recognition" (CLVR) approach. See Figure 3.4 for an overview of CVLR processing pipeline.

The contributions of this work are: **(1)** We propose a vehicle discriminative learning model for fine-grained vehicle instance re-identification task so that expensive and time-consuming cross-camera identity pairwise labelling can be avoided by using fine-grained vehicle model classification models for vehicle re-identification. This cross-level matching scheme is significantly different from existing methods that typically rely on the availability of identity instance annotations for re-ID models. Apart from reducing labelling cost, this approach takes into account that vehicle identity instance labelling can potentially cause a negative impact to re-ID model optimisation due to the strong similarities of different instances of the same vehicle model (low-intra class variance). To our best knowledge, this is the first attempt of exploiting the po-

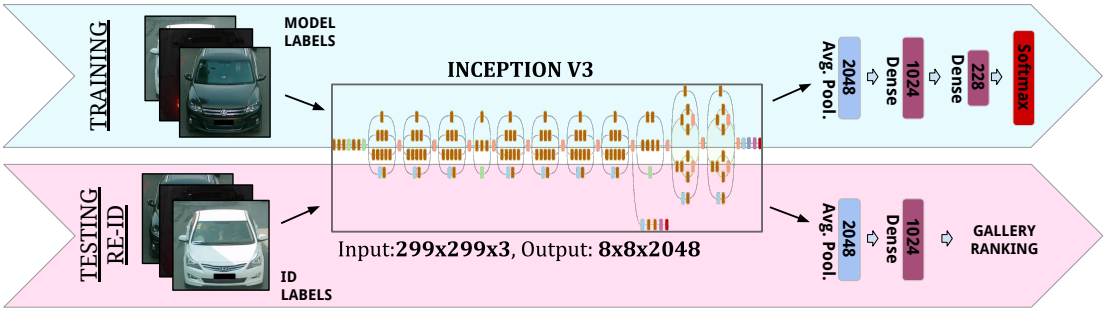

Figure 3.4: Overview of the proposed Cross-Level Vehicle Recognition (CLVR) method for vehicle re-identification: **(1) Training** (vehicle model classification): Learn a less fine-grained vehicle model classification deep model by a customised Inception-V3 [57] CNN network; **(2) Testing** (Vehicle Re-ID Matching): Deploy the learned CLVR model as a feature extractor using the output of the fully-connected feature vector (Dense-1024) layer for instance level vehicle re-identification.

tential of vehicle model information for semantically correlated instance level re-identification tasks. **(2)** We present a simple but effective CLVR model for vehicle re-identification by exploiting state-of-the-art deep Convolutional Neural Network (CNN) models (e.g. Inception-V3 [57]) for achieving not only accurate vehicle model classification but also reliable vehicle instance re-identification beyond just vehicle model-level recognition. Extensive comparative evaluations demonstrate the effectiveness of the proposed CLVR method regarding existing state-of-the-art vehicle re-ID models [43] utilising instance level label supervision on the largest vehicle re-ID benchmark dataset.

3.3.1 Fine-grained Vehicle Model Classification

We aim to learn a deep representation model for a generic distance matching (e.g. L_2) based vehicle re-identification without the need for tedious identity labels in model training, instead only less fine-grained vehicle model labels are exploited. We assume a set of n vehicle bounding box training images $\mathbb{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{n}$ with the corresponding vehicle model class labels as $\mathbb{Y} = \{\boldsymbol{y}^{(i)}\}_{i=1}^{n}$. These training images capture the visual appearance and variation of C_{model} different vehicle model categories where $y^{(i)} \in \mathbb{R}^{C_{\text{model}}}$ is one-hot encoded ground truth vehicle model label.

For model training, we utilise typical classification formulation with the cross-entropy loss function to optimise vehicle *model* discrimination given training labels of multiple vehicle model classes. Formally, we predict the posterior probability $\hat{p}_j$ of training image $\boldsymbol{x}$ over all vehicle model labels y_j:

$$\hat{p}_j = P(y_j = y|\boldsymbol{x}) = \frac{\exp(\boldsymbol{W}_j^\top \boldsymbol{z})}{\sum_{k=1}^{C_{\text{model}}} \exp(\boldsymbol{W}_k^\top \boldsymbol{z})} \tag{3.7}$$

where $\boldsymbol{z}$ refers to the feature vector of $\boldsymbol{x}$ from the CLVR CNN model, and $\boldsymbol{W}_k$ the prediction

function parameter of training model class k.

The model training loss on a minibatch of $\mathbb{X}_{\text{batch}} = \{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}_{i=1}^{n_{\text{batch}}}$ training samples is computed as:

$$\mathcal{L}_{\text{batch}} = -\frac{1}{n_{\text{batch}}} \sum_{i=1}^{n_{\text{batch}}} \log \left(P(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)} \right) \tag{3.8}$$

The negative log likelihood $-\log P(\boldsymbol{y} \mid \boldsymbol{x})$ or equivalently the cross-entropy $\mathcal{L}_{CE}$ between one-hot vector label encoding $\mathbf{y}$ and model's prediction $\hat{\boldsymbol{p}}$ over C classes is computed by:

$$\mathcal{L}_{CE} = \sum_{j=1}^{C} -y_j \log \hat{p}_j \tag{3.9}$$

3.3.2 Implementation Details

Training

We use the 42-layers Inception-V3 CNN architecture design [57] due to its high computational efficiency (higher classification accuracy with fewer parameters on ImageNet benchmark). We modify the network by (1) removing the original 1000-D classification layer (for ImageNet 1,000 class) and (2) adding a fully-connected *feature* layer with 1024 neurons on top of the Inception-V3 average pooling layer, followed by a new *classification* layer for accommodating the 228 vehicle model classes. Other competitive architectures, e.g. ResNet [20] or VGG [53], can be modified in a similar manner for this purpose.

Testing

After the CLVR deep CNN model is trained with vehicle model label annotations, we deploy the last fully connected layer output (1024-D vector) as the feature representation for more fine-grained vehicle re-ID at the instance level. We utilise *only* a generic distance metric *without* camera-pair specific distance metric learning, e.g. L_2 distance. Specifically, given a test probe vehicle image $\mathbf{z}^p$ from one camera view and a set of test gallery images $\{\mathbf{z}_i^g\}$ from other non-overlapping camera views: (1) We first compute their corresponding 1024-D feature vectors by forward-feeding vehicle images into the trained CLVR model, denoted as $\mathbf{z}^p$ and $\{\mathbf{z}_i^g\}$. (2) We then compute the cross-camera similarity score between $\mathbf{z}^p$ and $\mathbf{z}_i^g$ by L_2 distance. (3) We lastly rank all gallery images in ascending order by their matching distances to the probe image.

3.4 Experiments

3.4.1 Fine-grained Vehicle Model Classification

For evaluation, we selected the recent large vehicle re-identification dataset VehicleID [43]. This dataset provides a standard training/test images split: (1) 113,346 images of 13,164 identities for model training (8.61 images per identity); and (2) non-overlapping 108,211 image of 13,164 identities for test evaluation (8.22 images per identity). Of which 90,168 images in the training set are also labelled with vehicle model categories. Table 3.1 summarises of the image statistics.

Table 3.1: Summary of VehicleID Dataset.

Images	Train	Test
With model label	90168	0
Without model label	23178	108221
Total	113346	108221

Note that, only vehicle model labels are required for training fine-grained vehicle model classification. In total, there are 228 vehicle model classes, with many classes presenting only very subtle visual appearance differences. This causes typical fine-grained recognition challenges, further compounded by the uncontrolled appearance variations in illumination, pose, view-angle, and background clutters (see examples in Figure 2.3). Of the 90,168 images with model labels we split 80% for training, 20% for testing.

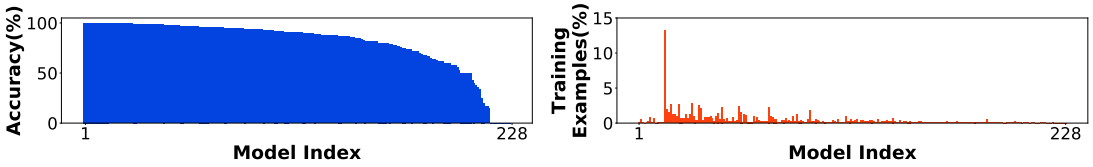

Figure 3.5: **(Left)**: Classification accuracies over all 228 vehicle model classes; **(Right)**: The training image size distribution over the corresponding model classes of (Left).

For vehicle model classification, the common accuracy measure, top-1, is used [10], *e.g.*percentage of correctly classified images in the test set. Overall, the proposed model achieves 94.8% vehicle model classification accuracy over all 228 model classes. This suggests the satisfactory performance of our learned deep features in distinguishing the subtle visual discrepancy between different but very similar vehicle model classes. We further examined the per-class recognition performance. Figure 3.5 (Left) shows that the vast majority classes can be very accurately recog-

nised, while a few instances obtain very low (even 0%) accuracy. This is mainly because of only very sparse corresponding training images available for these poorly detected classes (Figure 3.5 (Right)). For visual evaluation, we show some vehicle model recognition examples in Figure 3.6.

Figure 3.6: Qualitative evaluations of vehicle model classification. **(Left)**: Correctly classified vehicle images with large visual appearance similarity between different model classes. **(Right)**: Misclassified vehicle images due to extreme illumination conditions.

3.4.2 Vehicle Re-Identification

We used the cumulative matching characteristic (CMC) as vehicle re-identification performance measure [43]. The CMC is computed on each individual rank position k as the probe cumulative percentage of truth matches appearing at ranks $\leq k$. Moreover, three different scales of vehicle re-ID test setting are benchmarked: *small* (6,493 images of 800 identities), *medium* (13,377 images of 1,600 identities), and *large* (19,777 images of 2,400 identities). For each case, one image per identity is randomly selected from the test set as the probe image, whilst the remaining images are put into the gallery set. Summary of the test sets are available in Table 3.2.

Table 3.2: Test Gallery Sets in VehicleID.

	Small	Medium	Large
Number of vehicles	800	1600	2400
Number of images	6493	13377	19777

We present the results in Table 3.3. Compared to the state-of-the-art methods, the results are relatively competitive with the state-of-the-art considering that CVLR uses less informative model labels in training. The results show 7.2%,4.4% decrease in rank-1 accuracy in `small` and `large` test sets respectively compared to ID label supervised method with pairwise objective loss function: Coupled Cluster Loss(a variant of triplet loss). Compared to MixedDiff

Table 3.3: Vehicle re-identification performance comparisons. Metric: CMC measure (%).

Method	Pairwise	Label Type	Rank	Small (800)	Medium (1600)	Large (2400)
CCL[43]	✓	ID	1	43.6	37.0	32.9
MixedDiff[43]	✓	Model&ID		49.0	42.8	38.2
CLVR	✗	Model		**36.4**	**32.2**	**28.5**
CCL [43]	✓	ID	5	64.2	57.1	53.3
MixedDiff[43]	✓	Model&ID		73.5	66.8	61.6
CLVR	✗	Model		**52.9**	**48.8**	**45.6**

method which utilises both ID and model labels with two-branch network structure for each type of label supervision and complicated multi-stage training procedure, the difference increases to 12.6%,9.7% in rank-1 matching. Note that the larger (harder) test set becomes the advantage of using instance level label supervision diminishes, showcasing the usefulness of coarser model category labels in realistic larger data setting. These evidences suggest that vehicle model label supervision in deep neural network training can be effectively used for visual feature learning for vehicle re-identification. While expensive pairwise identity labels offer finer-grained information, model optimisation is likely to get confused, due to the subtle, and possibly no distinguishable visual appearance, in training image data. Figure 3.7 show visual examples of probe/gallery ranking using CLVR on vehicle re-identification.

3.5 Discussion

In contrast to existing methods, we uniquely bridge the connection between vehicle model classification and vehicle re-ID, by investigating the discrimination capability of vehicle model sensitive deep features in performing more fine-grained identity matching tasks. To our best knowledge, this is the first systematic attempt of investigating this structural knowledge inherent to man-made vehicles recognition approach in the hope of eliminating the tedious identity-level fine-grained labelling requirement. The state-of-the-art methods achieve their performance with typical identity level label supervision during training. Moreover, pairwise loss functions like triplet loss are often costly as it requires extra steps such as triplet sampling *e.g.*hard-negative

Figure 3.7: Highly successful Top-1 ranking examples. Left-most column is the probe image. Ranked images are ordered in ascending order by their distance from left to right. Green border shows same identity images, red border notes otherwise.(1 gallery image per ID)

triplet mining as in[43, 51]. This is in contrast to the CLVR design of exploiting only the cheaper vehicle model annotation in a much simpler way of training *i.e.*standard CNN training model training for classification with cross-entropy loss.

This type of approaches is largely motivated by the extensive person re-ID methods [15] due to their similar nature in the problem level. We have showed that using vehicle model labels are also useful when we want to match identities of vehicles visually. This is not possible in person re-identification. However our approach is brittle and still suffers under illumination changes and the performance is not on par with state-of-the-art that utilise identity labels. Further research is necessary regarding the use of both labels in conjunction.

In the following chapter we'll introduce a more realistic vehicle re-identification benchmark that tries to collect data points in uncontrolled situations with focus on low resolution images and that doesn't constrain the viewing angle to only front/and back views compared to the current vehicle re-identification benchmark VehicleID[43].

Chapter 4

Constructing a New Benchmark: Vehicle Re-Identification in Context (VRIC)

Figure 4.1: Realistic video frames from UA-DETRAC captured in uncontrolled environment showcasing: (a) common artefacts such as motion blur (b) low resolution with limited number plate visibility (c) different illumination due to time of day and changing weather conditions and time of day *i.e.*sunny, night, cloudy and rainy.

4.1 Introduction

We want to establish a realistic vehicle re-ID evaluation benchmark with natural visual appearance characteristics and matching challenges. Whilst these existing benchmarks VehicleID[43] and VeRi-776[45] have contributed significantly to the development of vehicle re-ID methods, they only represent *constrained* test scenarios due to the rather artificial assumption of having

high quality images of constant resolution (See Table 4.2). This makes them limited for testing the true robustness of re-ID matching algorithms in typically *unconstrained* wide-view traffic scene imaging conditions. To this end, it is necessary to collect a large number of vehicle images/videos from wide-view traffic scenes.

Given highly restricted access permission of typical surveillance video data, we propose to reuse existing vehicle related datasets publicly available in the research community.

In the following, we describe the process of constructing the Vehicle Re-Identification in Context (VRIC) benchmark.

4.2 Methodology

In this chapter we will detail how we repurpose a detection and tracking dataset into re-identification dataset in the spirit of the DukeMTMC4REID [17]. Like DukeMTMC[50] UA-DETRAC is a detection and tracking dataset but the subject matter is vehicles instead of people. However there is a signification difference when it comes to re-identification. Multi-tracking *multi*-camera aspect is quite important because re-identification is a problem of visual matching of identities between *different* camera views hence a multi-camera multi-tracking dataset is perfectly suited for this conversion. Compared to the aforementioned multi-tracking multi-camera detection and tracking datasets, UA-DETRAC is a multi-tracking *single*-camera detection and tracking dataset. This is an important distinction to note and Section 4.2.3 discusses how we overcome this drawback.

4.2.1 Source Dataset: UA-DETRAC

UA-DETRAC[63] is a challenging real-world multi-object detection and multi-object tracking benchmark. The dataset consists of 10 hours of videos captured with a Cannon EOS 550D camera at 24 different locations at Beijing and Tianjin in China. The videos are recorded at 25 frames per seconds (fps), with resolution of 960x540 pixels. There are more than 140 thousand frames in the UA-DETRAC dataset and 8250 vehicles that are manually annotated, leading to a total of 1.21 million labelled bounding boxes of objects.

Based on following considerations: (1) All videos were captured from the real-world traffic scenes (e.g. roads), reflecting realistic context for vehicle re-id. (2) It covers 24 different surveillance locations with diverse environmental conditions, therefore offering a rich spectrum of test scenarios without bias towards particular viewing conditions. (3) It contains rich object

and attribute annotations that can facilitate vehicle re-ID labelling. Samples of the whole scene images are shown in Fig 4.1.

4.2.2 Frame Selection

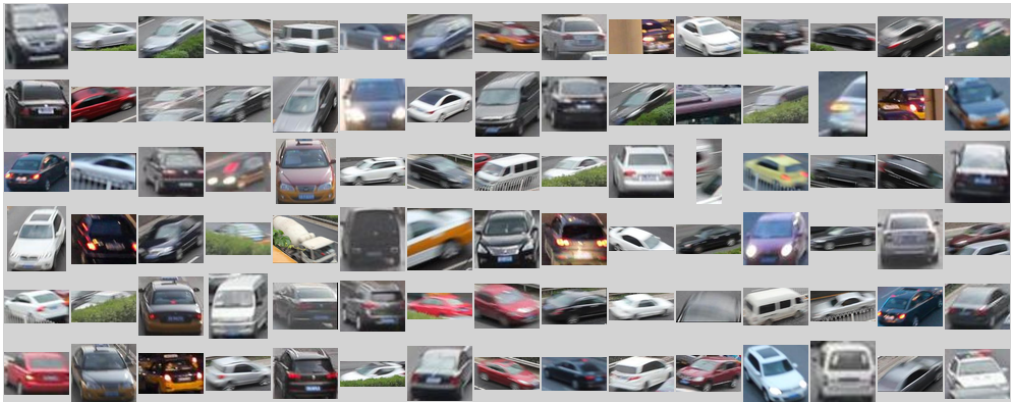

Figure 4.2: Samples of cropped image frames in final dataset of the VRIC benchmark.

To construct a vehicle re-ID dataset, we used 60 UA-DETRAC training videos with object bounding box annotations. For vehicle identity (ID) annotation, we started with assigning a unique label to each vehicle trajectory per UA-DETRAC video and then manually verified the ID duplication cases. Since all raw videos were collected from different scenes and time durations, we found little duplicated trajectories in terms of identity. To ensure sufficient vehicle appearance variation, we throw away short trajectories with less than 20 frames and bounding boxes smaller than 24×24. By doing so, we obtained 5,626 vehicle IDs across all 60 videos.

In terms of vehicle instance resolution, the average image resolution of all 60,434 vehicle bounding-boxes is 69.8×107.5 pixels in width×height, with a variance of 32 to 280 pixels due to the unconstrained distances between vehicles and cameras. This presents low resolution multi-scale re-ID matching benchmark compared to available datasets VeRi-776 and VehicleID. Fig 4.3 shows image width and height distributions afters the frame selection process.

4.2.3 Train and Test Data Split

For model training and testing using the VRIC dataset as a benchmark, we randomly split all 5,626 vehicle IDs into two disjoint sets: 2,813 for `train`, and 2,813 for `test`.

Train Set

To remove data redundancy coming from consecutive frames, we performed random frame-wise

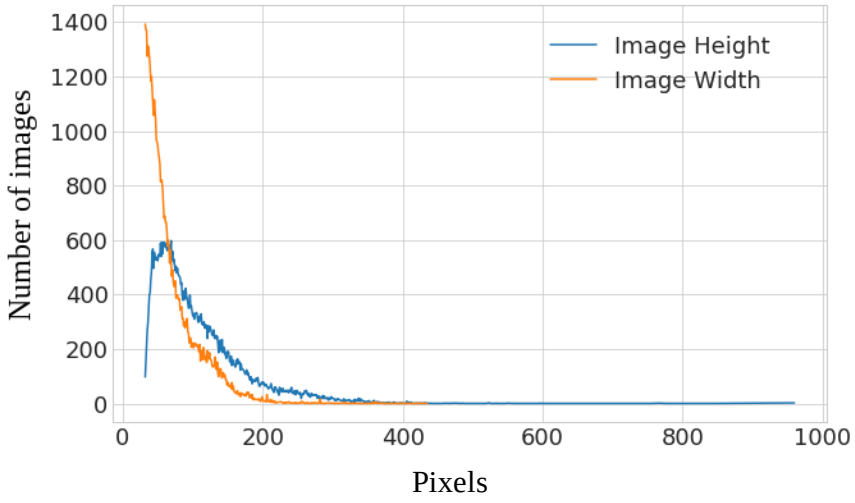

Figure 4.3: Histogram of image pixel width and height distributions in VRIC.

sub-sampling of the training trajectories rather than adding all the frames in each tracklet to the benchmark. Subsampling process is done as the following: for each of the identities selected for the `train` set, k number of frame indexes are randomly selected from a tracklet, where k is randomly selected integer between 1 and 40. This gives us a realistic setting with 54,808 where each identity in the training set has varying number of data points (images).

Test Set

UA-DETRAC is a multi-tracking *single*-camera detection and tracking dataset unlike its counterparts in person tracking where identities observed across cameras. Since there is no cross-camera pairwise ID matches we simulated cross-view variation by distant sampling of probe and gallery images. In particular, we defined two pseudo views, *near* or *far*, for each tracklet and then built the probe/gallery sets from the tracklets in the test set by choosing the first and last frames associated to near and far cameras corresponding to to probe and gallery images in the `test` set. As a result, for each 2813 identities in the test we have chosen one probe-gallery pair and in total test set contains 5,626 images. Fig 4.4 shows that the *near* and *far* views present different viewing conditions and hence allowing for a good simulation of two non-overlapping camera views. In this sense, VRIC contains a total 120 pseudo camera views from the 60 original camera views with unconstrained condition diversity.

From the above, we obtained 54,808/5,626 training/testing images for the VRIC benchmark. The data partition and statistics are summarised in Table 4.1.

Table 4.1: Data statistics and partition in VRIC.

	Total	Training Set	Test Set	
			Probe	Gallery
IDs	5,626	2,813	2,813	2,813
Images	60,434	54,808	2,813	2,813

4.2.4 Evaluation Protocol

Figure 4.4: Examples of probe/gallery test images in the VRIC benchmark. Probe and gallery image pairs are shown in first and second row respectively.

Resulting VRIC gallery test set only has one image for each identity, this is referred as single-shot setting. Another way to think about the test as set each identity has one image per view. We adopted the standard single-shot evaluation setting, i.e. For re-ID performance measure, we used the *Cumulative Matching Characteristic* (CMC) [31]. The CMC is computed for each individual rank k as the cumulative percentage of the truth matches for probes returned at ranks $\leq k$. In practice, the Rank-1 rate is often used as a strong indicator of an algorithm's efficacy.

4.3 Discussion

Figure 4.5: Example images of VehicleID, VeRi-776 and VRIC. Images in each row depict the same vehicle instance. VRIC images exhibit significantly more unconstrained variations in resolution, motion blur, occlusion/truncation and illumination within each vehicle bounding-box images.

The main contribution of VRIC benchmark is its varying low-resolution images compared

Table 4.2: Characteristics of vehicle existing re-id datasets.

Dataset	Images	IDs	Cameras	Resolutions Width×Height (Mean)	Motion Blur	Illumination	Occlusion
VehicleID [43]	113,123	15,524	-	345.4×376.1	No	Limited	No
VeRi-776 [45]	51,034	776	20	376.1×345.4	No	Limited	No
VRIC [28]	60,434	5626	120	65.9×103.0	Unconstrained	Unconstrained	Unconstrained

to existing datasets which contain mostly similar resolution images with higher resolution. Arguably more important aspect is the images' realistic capture conditions with non-constrained viewing angles and Additionally these images are not in any way constrained by the placement of cameras where we only get front/back views. Fig 4.5 provides a qualitative comparison of the collection of images available in competing benchmarks. In Table 4.2 datasets statistics are also shown highlight the image resolution characteristics combated to current datasets. Existing benchmarks[43, 45] have contributed significantly to the development of vehicle re-ID methods, however, they only represent *constrained* test scenarios due to the rather artificial assumption of having high quality images of constant resolution.

Chapter 5

Multi-Scale Vehicle Representation with Multi-Branch Convolutional Neural Network

5.1 Introduction

Existing vehicle re-identification (re-id) evaluation benchmarks consider strongly artificial test scenarios by assuming the availability of high quality images and fine-grained appearance at an almost constant image scale, reminiscent to images required for Automatic Number Plate Recognition, e.g. VeRi-776. Given the nature of this new benchmark VRIC introduced in Chapter 4, we further investigate a multi-scale matching approach to vehicle re-ID by learning more discriminative feature representations from multi-resolution images. Our approach is inspired by using a combination of features at different scales as in image pyramids.

Current vehicle re-ID studies are mainly driven by two benchmark datasets, VehicleID [43] and VeRi-776 [45]. While having achieved significant performance improvement (e.g. from 61.44% by [45] to 92.35% Rank-1 by [62] on VeRi-776), the scalability of existing re-ID algorithms to real-world vehicle re-ID applications remains unclear. All these methods utilise the global appearance features of vehicle images and ignore local discriminative regions. Moreover, existing benchmarks present rather artificial tests using high-quality images of high resolution, no motion blur, limited weather conditions and occlusion. (See Figure 4.5 from previous chapter for a visual comparison). We investigate a Multi-Scale Vehicle Representation (MSVR) learning model to address the inherent and significant multi-scale resolution in vehicle visual appearances from typical wide-view traffic scenes, currently an unaddressed problem in vehicle re-ID due

to the lack of a suitable benchmark dataset. Extensive comparative evaluations demonstrate the effectiveness of the proposed MSVR method in comparison to the state-of-the-art vehicle re-ID techniques on the two existing benchmarks (VehicleID [43] and VeRi-776 [45]) and the newly introduced VRIC benchmark.

In typical surveillance scenes, vehicles are often captured at varying scales (resolutions), which causes a significant inter-view feature representation discrepancy in re-ID matching. Our approach differs notably from existing vehicle re-ID models that typically assume single-scale representation learning.

5.2 Methodology

Our objective is to extract and represent complementary appearance information of vehicle identity from multiple resolution scales concurrently in order to optimise re-ID matching in instances where probe and gallery images differ in resolution. In this work, we investigate this problem in vehicle re-ID by exploring image pyramid [1, 33] inspired multi-branch CNN architectures where each sub-branch extracts a feature representation from different scales. Specifically, we exploit the potential of learning identity discriminative multi-scale representations originally designed for person re-id in [7].

MSVR Overview

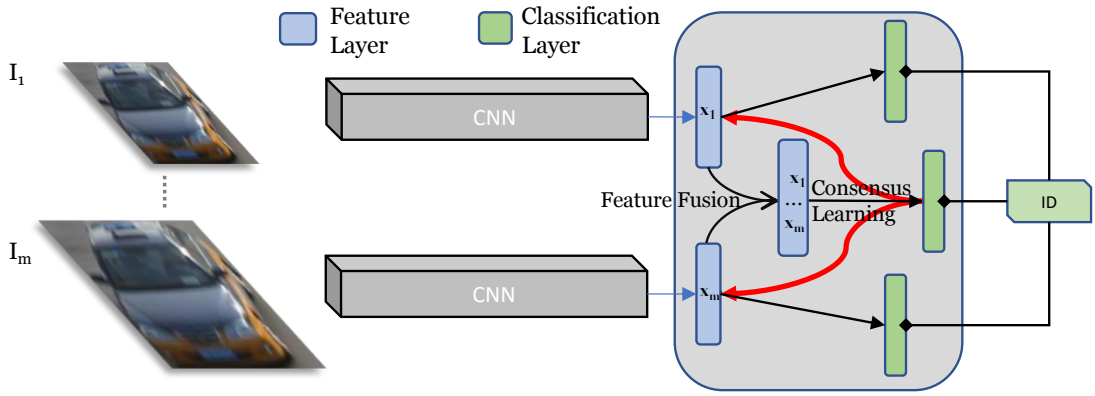

Figure 5.1: Overview of Multi-Scale Vehicle Representation (MSVR) learning for discriminative vehicle re-ID at varying spatial resolutions. MSVR learns vehicle re-ID sensitive feature representations from image pyramid by an network architecture of multiple branches all of which are optimised concurrently (consensus feedback shown in red, see Eq. (5.4)) subject to the same ID label constraints. Importantly, an inter-scale interaction mechanism is enforce to further enhance the scale-generic feature learning.

The overall MSVR network design is depicted in Fig 5.1. Specifically, MSVR consists of $(m+1)$ sub-networks: (1) m branches each for learning discriminative scale-specific visual fea-

tures. Each branch has an identical structure. (2) 1 fusion branch for learning the discriminative integration of m scale-specific representations of the same vehicle image. Low resolution input images are scaled down versions of the original images with bilinear interpolation. To maximise the complementary advantage between different scales of feature representation in learning, we concurrently optimise per-scale discriminative representations with scale-specific and scale-generic (combined) learning subject to the same ID label supervision. Critically, we further propagate multi-scale consensus as feedback to regulate the learning of per-scale branches. Next, we detail three MSVR components: (1) Single-Scale Representation; (2) Multi-Scale Consensus; (3) Feature Regularisation.

(1) Single-Scale Representation We exploit MobileNet [24] to design single-scale branches due to its favourable trade-off between model complexity and learning capability. To train a single-scale branch, we use the softmax cross-entropy loss function to optimise vehicle re-ID sensitive information from ID labels. Assume we have n training samples $\{\mathbb{X},\mathbb{Y}\}$. The images $\mathbb{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{n}$ with the corresponding vehicle ID labels $\mathbb{Y} = \{\boldsymbol{y}^{(i)}\}_{i=1}^{n}$. These training images capture the visual appearance variations of C_{ID} different identities where $\boldsymbol{y}^{(i)} \in \mathbb{R}^{C_{\text{ID}}}$ is one-hot vector encoding ground truth vehicle identity label y. Formally, we predict the posterior probability $\hat{p}_j$ of training image $\boldsymbol{x}$ over all vehicle identity labels y_j:

$$\hat{p}_j = P(y_j = y|\boldsymbol{x}) = \frac{\exp(\boldsymbol{W}_j^\top \boldsymbol{z})}{\sum_{k=1}^{C_{\text{ID}}} \exp(\boldsymbol{W}_k^\top \boldsymbol{z})} \tag{5.1}$$

where $\boldsymbol{z}$ refers to the feature vector of $\boldsymbol{x}$ of the CNN model, and $\boldsymbol{W}_k$ the prediction function parameter of training identity class k.

The negative log likelihood $-\log P(\boldsymbol{y} \mid \boldsymbol{x})$ or equivalently the cross-entropy $\mathcal{L}_{CE}$ between one-hot vector label encoding $\mathbf{y}$ and model's prediction $\hat{\boldsymbol{p}}$ over C classes is computed by:

$$\mathcal{L}_{CE} = \sum_{j=1}^{C} -y_j \log \hat{p}_j \tag{5.2}$$

(2) Multi-Scale Consensus We learn multi-scale consensus on vehicle ID classes between m scale-specific branches. We achieve this using joint-feature based classification. First, we obtain the joint feature vector of different scales by vector fusion. In MobileNet, feature vectors are computed by global average pooling of the last CNN feature maps with a dimension of 1024. Hence, this fusion produces a $1024\times m$-D feature vectors. We then use this combined features to perform classification for providing multi-scale consensus on the ID labels. We again adopt the cross-entropy loss (Eq (5.2)) as in single-scale representation learning.

(3) Feature Regularisation We regularise the single-scale branches by multi-scale consensus for imposing interaction between different scale representations in model learning. Specifically, we propagate the consensus as an auxiliary *feedback* to regularise the learning of each single-scale branch concurrently. We first compute for each training sample a soft probability prediction (i.e. a consensus representation) $\tilde{P} = [\tilde{p}_1, \cdots, \tilde{p}_i, \cdots, \tilde{p}_{C_{\text{ID}}}]$ as:

$$\tilde{p}_i = \tilde{p}(\tilde{y} = y|\boldsymbol{x}) = \frac{\exp(\frac{z_i}{T})}{\sum_{k=1}^{C_{\text{ID}}} \exp(\frac{z_k}{T})}, \quad i \in [1, \cdots, C_{\text{ID}}] \tag{5.3}$$

where z is the logit and T the temperature parameter (higher values lead to a softer probability distribution). We empirically set $T = 1$ in our experiments. Then, we use the consensus probability $\tilde{P}$ as the *teacher* signal to guide the learning process of each single-scale branch (*student*). To quantify the alignment between these predictions, we use the cross-entropy measurement which is defined as:

$$\mathcal{H}(\tilde{P}, P) = -\frac{1}{C_{\text{ID}}} \sum_{i=1}^{C_{\text{ID}}} \big(\tilde{p}_i \log(p_i) + (1 - \tilde{p}_i) \log(1 - p_i)\big) \tag{5.4}$$

The objective loss function for each single-scale branch is then:

$$\mathcal{L}_{\text{Scale}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{H}(\tilde{P}, P) \tag{5.5}$$

where the hyper-parameter λ ($\lambda = 1$ in our experiments) is the weighting between two loss terms. $P = [p_1, \cdots, p_{C_{\text{ID}}}]$ defines the probability prediction over all C_{ID} identity classes by the corresponding single-scale branch (Eq. (3.2)). As such, each single-scale branch learns to correctly predict the true ID label of training sample ($\mathcal{L}_{\text{CE}}$) by the corresponding scale-specific representation and to match the consensus probability estimated based on the scale-generic representation ($\mathcal{H}$).

MSVR Re-Identification Ranking

In testing, we deploy the fusion branch's representation for multi-scale aware vehicle re-ID matching. We use only a generic distance metric without camera-pair specific distance metric learning, e.g. L_2 distance. Based on the pairwise distances, we then return a ranking list as the re-ID results.

5.3 Experiments

Datasets For evaluation, in addition to the newly introduced VRIC dataset, we also utilised two most popular vehicle re-ID benchmarks. The **VehicleID** [43] dataset provides a training set with 113,346 from 13,164 IDs and a test set with 19,777 images from 2,400 identities. It adopts the single-shot re-ID setting, with only one true matching for each probe. Following the standard performance reporting, 10 random probe image selection was done and average performances are reported. in our experiments. The **VeRi-776** dataset [45] has 37,778 images of 576 IDs in training set and 200 IDs in test set. The standard probe and gallery sets consist of 1,678 and 11,579 images, respectively. The data split statistics are summarised in Table 5.1.

Table 5.1: Data split of vehicle re-ID datasets evaluated in our experiments.

Dataset	Training IDs / Images	Probe IDs / Images	Gallery IDs / Images
VehicleID[43]	13,164 / 113,346	2,400 / 17,377	2,400 / 2,400
VeRi-776[45]	576 / 37,778	200 / 1,678	200 / 11,579
(**VRIC**)	2,813 / 54,808	2,813 / 2,813	2,813 / 2,813

Performance Metrics For VehicleID and VRIC, we used the CMC measurement to evaluate re-ID performance. For VeRi-776, we additionally adopted the *mean average precision* (mAP) due to its multi-shot nature in the gallery of the test data. Specifically, for each probe, we compute the area under its Precision-Recall curve, i.e. Average Precision (AP). The mAP is then computed as the mean value of APs for all probes. This metric considers both precision and recall performance, and hence provides a more comprehensive evaluation.

Implementation Details In the MSVR model, we used 2 resolution scales, 224×224 and 160×160. We adopted the ADAM optimiser and set the initial learning rate to 0.0002, the weight decay to 0.0002, β_1 to 0.5, the minibatch size to 8, the max training epoch to 100,000. Model initialisation was done with ImageNet [10] pretrained weights. The MobileNet[24] architecture is used for each branch of different scale input. The data augmentation includes random cropping and horizontal flipping.

Evaluation Table 5.2 compares MSVR with state-of-the-art methods on three benchmarks. We make three main observations:

(1) Under standard visual appearance based evaluation setting (the top part), MSVR outperforms

Table 5.2: Comparative vehicle re-ID results on three benchmarking datasets. Upper part of table lists methods trained with only the images available from the respective datasets for fair comparison of the methods; lower part lists methods trained with additional datasets and/or labels. *: By our reimplementation. **E**: Extra information and annotation, *e.g.* number plates, local key-points, space-time prior knowledge. **M**: Multiple vehicle re-ID and classification datasets are combined for training. †: Result from [62].

Method	Notes	VeRi-776 [45]		VehicleID [43]		VRIC		Publication
		Rank-1	mAP	Rank-1	Rank-5	Rank-1	Rank-5	
LOMO [40]		25.33	9.64	-	-	-	-	CVPR'15
FACT [44]		50.95	18.49	-	-	-	-	ICME'16
Mixed Diff + CCL [43]		-	-	38.20	50.30	-	-	CVPR'16
Siamese-Visual [52]		41.12	29.40	-	-	-	-	ICCV'17
Siamese-Visual [52]	*	64.18	31.54	36.83	57.97	30.55	57.30	ICCV'17
OIFE(Single Branch) [62]	*	60.13	31.81	32.86	52.75	24.62	50.98	ICCV'17
MSVR		**88.56**	**49.30**	**63.02**	**73.05**	**46.61**	**65.58**	**Ours**
KEPLER [46] †	M	68.70	33.53	45.40	68.90	-	-	TIP'15
FACT + Plate + Space-Time [45]	E	61.44	27.77	-	-	-	-	ECCV'16
Siamese-CNN + Path-LSTM [52]	E	83.49	**58.27**	-	-	-	-	ICCV'17
OIFE(Single Branch) [62]	M	88.66	45.50	63.20	80.60	-	-	ICCV'17
OIFE(4Views) [62]	ME	89.43	48.00	**67.00**	**82.90**	-	-	ICCV'17
OIFE(4Views + Space-Time) [62]	ME	**92.35**	51.42	-	-	-	-	ICCV'17

all other competitors with large margins – MSVR surpasses the best competitor in Rank-1 rate by 24.38 % (88.56-64.18) on VeRi-776, 24.82% (62.02-38.20) on VehicleID, and 16.73% (46.61-30.55) on VRIC. This demonstrates the consistent superiority of MSVR over alternative methods in vehicle re-id, showing the importance in modelling multi-scale representation for vehicle re-id.

(2) Benefited from more training data plus space-time contextual knowledge and fine-grained local key-point supervision, the OIFE model achieves the best performance on VeRi-776. However, such advantages from additional data and knowledge representation is generically beneficial to all models including the MSVR.

(3) We carefully reproduced two methods, OIFE(Single-Branch) [62] and Siamese-Visual [52], and obtained inconsistent results compared to the reported performances of these two models. In particular, the performance of OIFE(Single-Branch) decreases on VeRi-776 and VehicleID.

This is mainly due to that the original results are based on a larger multi-source training set with 225,268 training images of 36,108 IDs (from VehicleID [43], VeRi-776 [45], BoxCars [54] and CompCars [69]), *versus* the standard 100,182 training images of 13,164 IDs on VehicleID, *i.e.*2.2 times more training images and 2.7 times more training ID labels, and the standard 37,778 training images of 576 IDs on VeRi-776, *i.e.*6.0 times more training images and 62.7 times more training ID labels, respectively.

Further Analysis

Table 5.3: Comparing single-scale and multi-scale representations of MSVR. Gain is measured as the performance difference of MSVR over the *mean* of single-scale variants.

Dataset	VeRi-776 [45]		VehicleID [43]		VRIC	
Metrics (%)	Rank-1	mAP	Rank-1	Rank-5	Rank-1	Rank-5
Scale-224	88.37	47.37	62.80	72.54	43.55	61.88
Scale-160	87.43	46.81	60.29	71.15	43.62	62.77
MSVR	**88.56**	**49.30**	**63.02**	**73.05**	**46.61**	**65.58**
Gain (%)	+0.76	+2.11	+1.47	+1.20	**+3.02**	**+3.25**

Table 5.3 compares the performances of a single-scale and a multi-scale feature representations of the MSVR model. Results on the VRIC benchmark, where we have significant resolution differences in resolution for the test images, the performance gain between single-scale and multi-scale feature representation is 3.02% in rank-1. However on the other datasets where the resolution of the images in the dataset are more uniform, the performance increase is 0.76% and 1.47% for VeRi-776 and VehicleID respectively for rank-1. It is evident that the multi-scale representation learning with MSVR has performance benefit across all three datasets with different resolution scale changes, however, the performance gain is dependant on the inherent characteristics of the dataset, especially the resolution variance. The model performance gain on VRIC is the largest, which is consistent with the more significant scale variations exhibited in the VRIC vehicle images (See Table 5.1). This shows that the overall effectiveness of MSVR in boosting vehicle re-ID matching performance.

5.4 Discussion

Enabled by the availability of the new datasets with varying resolution images for cross camera matching in vehicle re-identification setting, we proposed a multi-scale visual representation learning paradigm, utilising an image pyramid inspired multi-branch neural network design. Our visual representation learning optimisation design by consensus learning feedback enables us to learn compatible feature extraction from multiple scales leading into resolution invariant feature learning. In other words, we investigated a multi-scale learning representation by exploiting an image pyramid inspired multi-branch CNN. Experimental evaluations demonstrate the effectiveness and performance advantages of our multi-scale learning method over the state-of-the-art vehicle re-ID methods on three benchmarks VeRi-776, VehicleID, and VRIC. The realistic and challenging vehicle re-identification benchmark, Vehicle Re-Identification in Context (VRIC), containing unconstrained vehicle images with varying resolution, from wide-view traffic scenes showed the most performance gains with our method.

Next chapter focuses on expanding multi-branch CNN architecture with grayscale input in conjunction with multi-modal multi-label supervision of vehicle viewpoint information.

Chapter 6

Orientation Invariant Feature Learning With Multi-Modal Training

6.1 Introduction

The main challenge of vehicle re-ID is that the visual appearance of vehicles can be quite different according to diverse camera angles and illumination changes. Most existing vehicle re-ID models cannot make full use of various complementary vehicle information, e.g. vehicle type and orientation. In this work, we propose an original *Multi-Task Mutual Learning* (MTML) deep model to learn discriminative features simultaneously from multiple branches. Specifically, we design a consensus learning loss function by fusing features from the final convolutional feature maps from all branches. Extensive comparative evaluations demonstrate the effectiveness of our proposed MTML method in comparison to the state-of-the-art vehicle re-ID techniques on a large-scale benchmark dataset, VeRi-776. We also yield competitive performance on the NVIDIA 2019 AI City Challenge Track 2.

In this work, we propose an original *Multi-Task Mutual Learning* (MTML) based network architecture, that aims to simultaneously learn a number of recognition tasks from different supervisory signals, plus a consensus loss function, to build an improved representation that is robust in varying resolution input and viewpoint aware for vehicle re-identification.

We make two contributions in this work: (1) we formulate an original *Multi-Task Mutual Learning* (MTML) deep learning model by building multi-branch CNN where each branch learns a feature extractor that specialises on certain type of input and supervised by complementary la-

Index	Orientation	Colour
0	front	red
1	rear	-
2	left	-
3	left front	cyan
4	left rear	yellow
5	right	-
6	right front	green
7	right rear	black

Figure 6.1: Examples from the VeRi-776 dataset with the orientation labels provided in [62] (best viewed in colour).

bel supervision. For vehicle re-ID we are aiming for an invariant feature representation regarding multi-scale and grayscale input and/or pose identification by orientation labels. 2) we introduce a mutual learning mechanism to improving the robustness of re-ID feature fusion by synchronising the discrimination qualities of each branch with consensus feedback. Our model benefits from multiple supervisory signals in order to enhance model learning of more discriminative features for vehicle Re-ID. Our model aims to fuse the complementary information improved re-ID matching. Extensive evaluations demonstrate the effectiveness of the proposed MTML method in comparison to the state-of-the-art vehicle re-ID techniques on the large-scale benchmark VeRi-776 [45]. We also yield competitive performance on the CityFlow [59] benchmark at the NVIDIA 2019 AI City Challenge.

A number of deep learning techniques have been exploited for the purpose of vehicle re-ID. For instance, Liu *et al.*[45] explored a deep neural network to estimate the visual similarities between vehicle images. Liu *et al.*[43] also designed a Coupled Cluster Loss (CCL) to boost a multi-branch CNN model for vehicle Re-ID. All state-of-the-art methods utilise the global appearance features of vehicle images and ignore local discriminative regions. To explore local information, motivated by the idea of landmark alignment [71] in both face recognition [58] and human body pose estimation [47], Wang *et al.*[62] considered 20 vehicle keypoints for learning and aligning local regions of a vehicle images for re-ID. Clearly, this approach comes with extra cost of exhaustively labelling of keypoints in a large number of vehicle images, and the implicit assumption of having sufficient image resolution/details for extracting these keypoints.

Additionally, space-time contextual knowledge has also been exploited for vehicle Re-ID subject to structured scenes [45, 52]. Liu *et al.*[45] proposed a spatio-temporal affinity approach for quantifying every pair of images. Shen *et al.*[52] further incorporated spatio-temporal path information of vehicles. Whilst this method improves the re-ID performance on the VeRi-776 dataset, it may not generalise to complex scene structures when the number of visual spatio-temporal path proposals is very large with only weak contextual knowledge available to facilitate model decision.

Multi-task Learning(MTL) is a machine learning strategy that learns several related tasks simultaneously for their mutual benefits [3]. A good MTL survey with focus on neural networks is provided in [5]. Deep CNNs are well suited for performing multi-task learning as they are inherently designed to learn joint feature representations subject to multiple label objectives concurrently in multi-branch architectures. Joint learning of multiple related tasks has been proven to be effective in solving computer vision problems [11, 72]. Critically, our method is uniquely designed to explore the potential of multi-task learning by combining multiple label supervision of the vehicle images (e.g. ID and orientation) with each label being associated with an individual sub-branch of a single model.

6.2 Methodology

6.2.1 Multi-modal Vehicle Re-identification

In order to perform re-ID of previously unseen query vehicles, the aim of our model is to learn a feature embedding and calculate similarity of vehicle images by distances in the feature embedding space (*e.g.*L_2). We aim to improve vehicle re-ID matching by utilising multi-branch CNN network design with multi-modal label supervision that would result in accurate matching of identities. In order to perform this task, we utilise training data containing a number of different labels: identity class labels as well as vehicle orientation class labels. We assume two sets of training examples $\mathbb{X}_1 = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ and $\mathbb{X}_2 = \{\boldsymbol{x}^{(i)}\}_{i=1}^{M}$, containing N and M training images respectively. Both training sets contain the associated identity class labels $\mathbb{Y}_1 = \{\boldsymbol{y}^{(i)}\}_{i=1}^{N}$ and $\mathbb{Y}_2 = \{\boldsymbol{y}^{(i)}\}_{i=1}^{M}$, where $y_i \in [1,...,C_{ID}]$ for C_{ID} distinct vehicle identities spanning the two training sets. In addition, $\mathbb{X}_1$ also contains orientation labels, $\mathbb{O} = \{\boldsymbol{o}^{(i)}\}_{i=1}^{N}$, where $\boldsymbol{o}_i \in [1,...,C_O]$ is the orientation (for C_O possible orientations).

In order to perform accurate vehicle re-ID, we use this data to build a model constructed

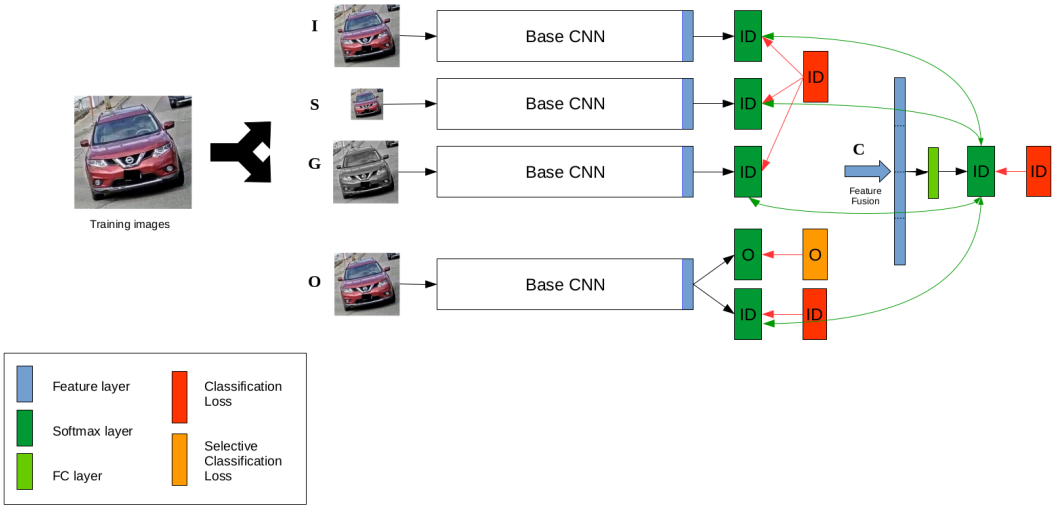

Figure 6.2: An overview of our proposed model (best viewed in colour). (I) Vehicle identity branch (S) Multi-scale analysis branch (G) Grayscale analysis branch (O) Vehicle orientation branch (C) Consensus learning through feature fusion. Feedforward signals shown in black. Hard target (ground-truth) loss propagation shown in red. Soft target consensus feedback loss propagation shown in green.

from multiple branches, each of which is tasked with learning a specific aspect of the data concurrently. The branches of the model are as the following: B^I) Identity classification, B^S) Identity classification from a scaled image, B^G) Identity from grayscale image, B^O) Identity plus the vehicles' orientations. These individual branches then form a consensus prediction on the identity of the training examples, and this consensus is then used as a regularise signal for the individual sub-branches.

6.2.2 Model Structure and Feature Learning

An overview of our proposed model can be seen in Figure 6.2. The model is composed of four sub-branches, each of which is simultaneously learning a representation to solve its own task. In addition, there is a single fusion branch, which allows feature selection to be performed from the entire collection of individual representations. It is the output from this branch that is taken during deployment. Each sub-branch will now be described in detail.

(I) Vehicle Identity The root branch of our model is tasked with learning the best representation for vehicle identity discrimination, for both training sets $\mathbb{X}_1$ and $\mathbb{X}_2$ with C_{ID} different identities where $\boldsymbol{y}^{(i)} \in \mathbb{R}^{C_{\text{ID}}}$ is one-hot vector encoding ground truth vehicle identity label y. Here, we exploit the cross entropy classification loss function in order to train one branch to predict vehicle identity. Thus, we predict the posterior probability p_j^{ID} of training image $\boldsymbol{x}$ over all identity labels

y_j:

$$p_j^{ID} = P(y_j = y|\boldsymbol{x}) = \frac{\exp(\boldsymbol{W}_j^\top \boldsymbol{z})}{\sum_{k=1}^{C_{ID}} \exp(\boldsymbol{W}_k^\top \boldsymbol{z})} \tag{6.1}$$

where $\boldsymbol{z}$ refers to the feature vector of $\boldsymbol{x}$ of the CNN model, and $\boldsymbol{W}_k$ the prediction function parameter of training model class k. The cross-entropy $\mathcal{L}_{ID}$ between one-hot vector label encoding $\boldsymbol{y}$ and model's prediction $\boldsymbol{p}^{ID}$ over C_{ID} classes is computed by:

$$\mathcal{L}_{ID} = \sum_{j=1}^{C_{ID}} -y_j \log p_j^{ID} \tag{6.2}$$

(S) Identity from Scaled Image Here we exploit the multi-scale analysis that has previously been shown to be of benefit for the task of re-identification, both for persons [8] and vehicles [28]. This is done by including a branch that is trained via cross-entropy loss (Eq. 6.4) to predict the class identity from a rescaled version of the input image, in a similar way to branch A. The low resolution image is the result of resizing with bilinear interpolation from the high resolution image.

(G) Identity from Grayscale Image In order to direct the model to learn low frequency lightness changes in vehicles images, we ensure that one branch will not be able to use colour information for distinguishing between these classes. This is done by giving as input only the grayscale image, and again training the branch to predict identity via the cross entropy loss. The grayscale image is the resulting of RGB conversion using Floyd-Steinberg dither to approximate the original image luminosity level.

(O) Vehicle Orientation This branch is tasked with learning a representation to simultaneously predict the identity class and the orientation class when this is known. Both sets of labels are simultaneously employed in a joint loss function in order to optimise the branch for the prediction of both identity and orientation. As orientation labels are not available for all training data, we employ a selective classification subset loss function, that allows the loss to be calculated across only the subset of the batch for which orientation labels are known.

Again, the cross entropy loss is exploited for this task. Hence, the branch calculates both Eq. (6.3), as well as the softmax posterior probability of the orientation label o for the images for which the orientation class is known. Using $\mathbb{O}$ with C_O different orientations where $\boldsymbol{o}^{(i)} \in \mathbb{R}^{C_O}$ is one-hot vector encoding ground truth vehicle identity label o. Here, we use the cross entropy classification loss function in order to train one branch to predict vehicle orientation. Thus, we

predict the posterior probability p_j^O of training image $\boldsymbol{x}$ over all orientation labels o_j:

$$p_j^O = P(o_j = o|\boldsymbol{x}) = \frac{\exp(\boldsymbol{W}_j^\top \boldsymbol{z})}{\sum_{k=1}^{C_O} \exp(\boldsymbol{W}_k^\top \boldsymbol{z})} \tag{6.3}$$

where $\boldsymbol{z}$ refers to the feature vector of $\boldsymbol{x}$ of the CNN model, and $\boldsymbol{W}_k$ the prediction function parameter of orientation label k. The cross-entropy $\mathcal{L}_O$ between one-hot vector label encoding $\boldsymbol{o}$ and model's prediction $\boldsymbol{p}^O$ over C_O orientations is computed by:

$$\mathcal{L}_O = \sum_{j=1}^{C_O} -o_j \log p_j^O \tag{6.4}$$

The total loss for the orientation branch $\mathcal{L}_{BO}$ is calculated as:

$$\mathcal{L}_{BO} = \begin{cases} \mathcal{L}_{ID} + \mathcal{L}_O & \text{if } x^{(i)} \in \mathbb{X}_1 \\ \mathcal{L}_{ID} & \text{if } x^{(i)} \in \mathbb{X}_2 \end{cases} \tag{6.5}$$

(C) Consensus Learning and Feedback In order to harness the benefit of all branches for the purpose of vehicle re-identification, we employ consensus learning as proposed in [8] and previously harnessed for vehicle re-ID in [28]. This is done via feature fusion of the final convolutional feature maps from all branches for consensus learning. As our branches are based on the ResNet50 architecture [20], these feature maps are formed via an average pooling operation that results in feature vectors of length 2048. Hence our fused concatenated features in the consensus branch are of length 8192. We then add one additional fully connected layer, of size 1024, and the output of this is passed to a final identity softmax classification layer, again employed with cross entropy loss. Hence:

$$p_j^C = P(y_j = y|\boldsymbol{x}) = \frac{\exp(\boldsymbol{W}_j^\top \boldsymbol{z})}{\sum_{k=1}^{C_{ID}} \exp(\boldsymbol{W}_k^\top \boldsymbol{z})} \tag{6.6}$$

where $\boldsymbol{z}$ refers to the consensus feature vector of $\boldsymbol{x}$.

Additionally, we also utilise a consensus propagation mechanism, similar to the previously proposed method [8, 28]. Here the consensus output is taken as 'soft targets' (as opposed to the ground truth label 'hard targets') for the training data, and used to feedback information about the predictions made by the entire ensemble of branches. This is done concurrently with the training of the individual branches. The method is inspired by the idea of Knowledge Distillation (KD) [23], but is different in that here we employ the combined predictions from all 'student' branches as a *virtual* teacher model, rather than utilising a pretrained powerful teacher model to provide soft targets.

Algorithm 1 The MTML training algorithm.

Require: Training sets $\mathbb{X}_1$ $\mathbb{X}_2$, labels $\mathbb{Y}_1$ $\mathbb{Y}_2$ $\mathbb{O}_1$, model $\mathcal{M}$

- Initialise network branches with pretrained ImageNet weights
- Initialise output layers of $\mathcal{M}$ randomly

for epoch $e \in (1, E)$, minibatch images $\mathbb{X}_e$ **do**

- Feed-forward images $\mathbb{X}_e$ through model for all branches in $\mathcal{M}$
- Feed-forward to obtain identity classification predictions for images $\mathbb{X}_e$
- Feed-forward to obtain orientation classification predictions for images $\mathbb{X}_e \in \mathbb{X}_1$
- Fuse features and obtain consensus identity classification predictions for images $\mathbb{X}_e$
- Calculate identity losses for each branch and back-propagate to update weights
- Calculate orientation losses for images $\mathbb{X}_e \in \mathbb{X}_1$ and back-propagate to update weights on the orientation branch
- Calculate identity losses for the consensus branch and back-propagate

end for

Specifically, the feedback mechanism employs the consensus probability prediction vector $P^C = \left[p_1^C, \ldots, p_{C_{ID}}^C\right]$ given image $\boldsymbol{x}$, feeding these into the cross entropy loss between the two distributions to provide a consensus regularisation loss for each input sub-branch B^I, B^S, B^G and B^O. The total consensus loss for a particular branch B is then:

$$\mathcal{H}(P^C, P^B) = -\frac{1}{C_{ID}} \sum_{j=1}^{C_{ID}} p_j^C \log p_j^B \tag{6.7}$$

where $P^B = \left[p_1^B, \ldots, p_{C_{ID}}^B\right]$ is the softmax output probability vector. This is added to each individual branch's loss function. In addition, this mechanism provides regularisation of the whole network by propagating all of the consensus losses back through the feature fusion layer, which also boosts the learning of the ensemble. The total loss for a particular branch is then:

$$\mathcal{L} = \mathcal{L}_B + \lambda \mathcal{H}(P^C, P^B) \tag{6.8}$$

6.2.3 Model Training

In order to train our model, we combine both training sets, $\mathbb{X}_1$ and $\mathbb{X}_2$, and employ batches that contain both images with and without orientation labelling. The full training algorithm can be seen in Algorithm 1.

Table 6.1: Details of the datasets employed for train and test.

Dataset	Training			Probe		Gallery	
	#IDs	#Imgs	#Orients	#IDs	#Imgs	#IDs	#Imgs
VeRi-776 [45]	576	37778	8	200	1678	200	11579
CityFlow [59]	333	36935	-	333	1052	333	18290

6.2.4 Vehicle Re-ID deployment

During deployment, we employ the feature fusion layer from our trained model as the full feature representation in order to perform vehicle re-identification matching. As we do not necessarily have camera information about the query or gallery images, or timestamp information, which would allow the use of camera distance or time-based analysis, we use only a generic distance metric - the L_2 metric - in order to match gallery images to the query. Hence, for each of the query image $\boldsymbol{x}_q^{(i)}$, and the gallery images $\{\boldsymbol{x}_g^{(i)}\}$, we compute our 6400 dimension fused feature representations, $\boldsymbol{z}^q$ and $\{\boldsymbol{z}_g^{(i)}\}$ respectively. We then calculate the L_2 distance between the query representation and each of the gallery images, and rank the latter by increasing distance in order to calculate the Rank-1 and mAP performance scores.

6.3 Experiments

We conduct a number of experiments to explore the performance of our method. First, we exploit a number of widely available vehicle ID benchmark datasets in order to assess the benefit of each of the branches of our model independently, and altogether. Then we compare the performance of our model to other current work by looking at our performance in the NVIDIA AI City Challenge 2019 Task 2 (Vehicle Re-identification). As our method includes a branch that predicts vehicle orientation in addition to identity, our model requires data that contains the orientation labels for training. As a result, we include the VeRi-776 [45] dataset in the training set for all our experiments.

6.3.1 Datasets

We employ two vehicle re-ID datasets in our experiments in order to train and test our method extensively. Firstly we conduct experiments on a benchmark dataset, VeRi-776 [45], which has been widely tested by the majority of recent works. And secondly, we employ the new CityFlow

dataset [59], a challenging dataset that has been shown to be more difficult than previous publicly available benchmarks. The VeRi-776 dataset [45] has 37,778 images of 576 IDs in the training set and 200 IDs in the test set. The standard probe and gallery sets consist of 1,678 and 11,579 images, respectively. There are also orientation labels, for 8 possible orientations, available for the VeRi-776 dataset, which were provided by [62]. The CityFlow dataset [59] has 36,935 images of 333 IDs in the training set and 333 different IDs in the test set. The standard probe and gallery sets consist of 1,052 and 18,290 images respectively. The data split statistics of both datasets are summarised in Table 6.1.

6.3.2 Implementation Details

We employ the ResNet50 [20] network architecture as the base of our model. We train the model with minibatches of size 8, using the Adam optimisation technique with a learning rate of 0.0001, exponential decay rates set as the following: $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The two image sizes used were standard 224x224 and small (for the scaled branch) 160x160.

We measure the performance of our vehicle re-identification methods according to the standard Cumulative Matching Characteristic (CMC) and mean Average Precision (mAP). The CMC is computed on each individual rank k as the cumulative percentage of correct matches appearing at ranks $\leq k$. The mAP is calculated as the mean over all query images of the Average Precision, which itself is calculated as the precision cut-off at each correct recalled image position averaged over all possible correct gallery images.

6.3.3 Evaluation on VeRi-776 Dataset

Firstly, we train and test on the VeRi-776 dataset in order to compare with existing state-of-the-art methods with identical settings. In order to experiment the benefit of adding each of the separate branches of our model, we take branch I (vehicle identity) as our central branch, and all models have an C (consensus) branch. We then perform experiments where we include each of the other branches in turn. So MTML-*S* refers to a model built from branches I,C and *S*, MTML-OG has branches I,C and *O and G*, and so on. MTML-OSG (branches I, S, G, O and C) is then our full model, and MTML-OSG (ReRank [73]) the results of the full model after additional re-ranking. Table 6.3 shows the included branches. Table 6.2 shows the experiment results, with all experiments run for 150 epochs of training. As can be seen, even before re-ranking, our full model achieves state-of-the-art mAP and Rank-1 scores on this dataset, of 64.6% and 92.3%

Table 6.2: Trained/tested on VeRi-776

Method	mAP	Rank-1	Rank-5
MSVF [28]	49.3	88.6	-
OIFE [62]	51.4	68.3	89.7
S-CNN+P-LSTM [52]	58.3	83.5	90.0
MTCRO [67]	61.6	87.2	94.2
MTCRO (ReRank) [67]	62.6	88.0	94.6
MTML-S	59.4	89.5	94.9
MTML-O	60.8	90.2	95.4
MTML-G	62.8	91.1	95.8
MTML-SG	63.7	90.6	95.8
MTML-OG	63.5	92.0	**96.4**
MTML-OSG	**64.6**	**92.3**	95.7
MTML-OSG (ReRank)	***68.3***	92.0	94.2

respectively. And after re-ranking the mAP score is increased to 68.3%.

Table 6.3: Tested architectures and the included branches

Method	Branches
MTML-S	I,C,S
MTML-O	I,C,O
MTML-G	I,C,G
MTML-SG	I,C,S,G
MTML-OG	I,C,O,G
MTML-OSG	I,C,O,S,G

The mAP results from the experiments also show how the individual branches contribute to the performance, with orientation (60.8%) improving over the result of scaled analysis alone (59.4%), and grayscale doing even better (62.8%). This demonstrates how allowing the model to learn about the orientation of the vehicle at the same time as identity can strengthen the performance. And that removal of the colour during learning - although obviously a useful indicator

of identity at test time - allows for the model to focus on the more discriminatory features of the identity that ultimately boosts the re-ID performance.

The combinations of three branches all show improvement over only two, with MTML-SG and MTML-OG achieving mAPs of 63.7% and 63.5% respectively. However they are still outperformed by combining all four branches in the MTML-OSG model. These results show that combining all the different signals for multi-task learning does indeed allow for the overall model to perform better in the final task of vehicle re-identification.

6.3.4 Evaluation on CityFlow Dataset

Table 6.4: Trained/tested on CityFlow

Method	mAP	Rank-1	Rank-5
Resnet50 [59]	25.5	41.3	-
MTML-S	17.0	40.4	53.3
MTML-G	19.6	44.5	58.4
MTML-SG	20.6	44.1	55.8
MTML-SG (ReRank)	25.7	43.4	47.2

Table 6.5: Trained/tested on CityFlow+VeRi-776

	VeRi-776			CityFlow		
Method	mAP	Rank-1	Rank-5	mAP	Rank-1	Rank-5
MTML-S	58.4	88.5	94.6	18.9	40.6	53.3
MTML-O	59.2	89.9	94.9	20.3	44.3	56.0
MTML-G	61.6	89.7	95.1	21.6	46.1	57.5
MTML-SG	62.6	90.8	**95.8**	22.1	45.8	56.5
MLML-OG	62.0	**91.2**	95.6	22.9	46.6	58.2
MTML-OSG	**62.6**	90.6	95.5	**23.6**	**48.9**	**59.7**
MTML-OSG (ReRank)	***66.4***	91.5	93.6	***29.2***	48.8	50.7

We participated in Task 2 of the NVIDIA AI City Challenge 2019. This task was benchmarked with a new vehicle re-identification dataset. city-scale multi-camera vehicle re-identification.

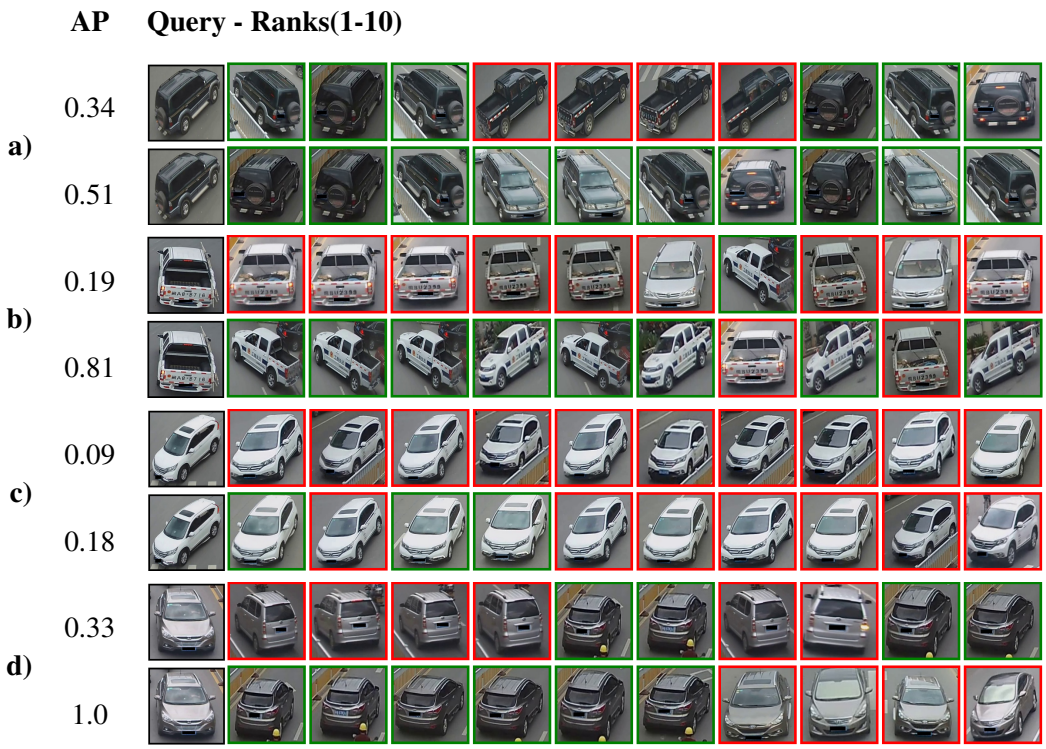

Figure 6.3: Qualitative comparison of example query images between experiments with and without inclusion of the orientation branch. Beginning with the query image in the first column, ranks from Rank-1 to Rank-10 is shown. Each pair (a,b,c,d) compares MTML-SG (upper) to MTML-OSG (lower) trained and tested on the VeRi-776, showcasing the impact on cross-camera matching performance gains by training with additional orientation labels. AP refers to Average Precision of that query. Correct and incorrect identity matches are shown with green and red borders around images, respectively.

Multiple cameras were placed at multiple intersections and no camera information was provided about the images.

Two sets of experiments are conducted on CityFlow benchmark: (1) Training on CityFlow, and (2) Training on CityFlow and VeRi-776. For the first set of experiments, we trained MTML-S (branch A, B and E), MTML-G (branch A, C and E) and MTML-SG model only on CityFlow training data. Table 6.4 shows that: (1) MTML-G branch combination is much better than MTML-S branch combination. The potential reason is that grayscale analysis is more useful than multi-scale in vehicle Re-ID. (2) Joint learning with MTML-SG is better than any individual one of them on mAP evaluation. Another interesting observation is that after re-ranking algorithm, mAP and rank-1 performance improves while rank-5 score drops. The possible reason as the re-ranking moves up an image with correct ID to rank-1, at the same time it also moves up in rank images with different identities with similar visual appearance as well; resulting in a lower rank-5 score as other gallery images with the correct identity is pushed down in rank.

Since the orientation label is only available in VeRi-776, for training the full MTML model including the orientation supervisory signal, we did the second experiment which included both this database plus CityFlow. Table 6.5 shows that: (1) By adding VeRi-776 training data, with MTML-S, MTML-G and MTML-SG, we all obtain a better mAP and CMC performance on CityFlow than the model which was only trained on CityFlow. Meanwhile, the mAP and CMC performance is slightly lower on VeRi-776 than the model only trained on VeRi-776. We suspect this is due to training for a shorter period of time, as this experiment ran for only 100 epochs, compared to 150. (2) By adding the orientation branch for the MLTML-OSG model, we obtain the best mAP performance 62.6% and 23.6% on VeRi-776 and CityFlow respectively. This is improved to 66.4% and 29.2% with re-ranking. This shows that our method of mutual learning between the orientation branch supervised by orientation labels and the other branches supervised by ID label is effective.

Qualitative results showing a comparison of rankings with or without orientation branch are shown in Figure 6.3. The advantage of learning both the orientation and ID signal can be seen in each pair where the MTML-OSG model is able to rank very different views of the same ID vehicle highly, which compares to the MTML-SG model which can only find images containing similar viewpoints, many of which are incorrect IDs (Figure 6.3(a,b,d)). It can also be observed that similar viewpoints are better grouped together in the ranks (Figure 6.3(c,d)).

6.4 Discussion

Vehicle re-ID is a difficult problem due to the fact that the visual appearance of a vehicle instance may drastically change due to diverse viewpoints and illumination, whilst different vehicle instances of the same model type may have a very similar appearance. In this chapter, we propose *Multi-Task Mutual Learning* (MTML) deep model to learn discriminative features simultaneously from multiple branches by combining multi-scale feature fusion in conjunction with additional orientation labels. Following from the findings of the effectiveness of consensus learning by feature fusion in multi-scale input, we explore a similar approach with additional features from grayscale input branch and orientation branches. Orientation labelling provides the information to address large inter-class visual appearance variance from vehicle pose with respect to camera. Consensus learning by feature fusion from all branches including orientation allows for cross-camera matching vehicle identities invariant to pose. Extensive evaluations

show the performance increase in all permutations of the proposed additions to neural network design in matching performance, measured by mAP. Qualitative investigation of re-ID ranking performance shows that different poses of the same identity instances in the gallery set are ranked higher in testing. Additionally, grayscale input branch provides a data augmentation option for improved vehicle re-identification performance. We also yield a competitive performance on the NVIDIA 2019 AI City Challenge Track 2 tested using the CityFlow benchmark.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we have explored several approaches for the vehicle re-identification problem to improve visual matching capability with modern convolutional neural networks. Vehicle re-identification is the matching problem of visual vehicle appearance captured by different cameras in varying pose and illumination.

In computer vision literature, convolutional neural networks are the preferred tools for processing visual data *e.g.*images and videos. CNNs can learn feature representations with high level discriminative capabilities, enabled by the efficient learning implementations on modern hardware as well as large scale labelled data that is currently available. Large scale datasets such as ImageNet[10] provide diverse labels that can be used to learn non-linear CNN models with millions of parameters that can encode discriminative feature extractors. This is achieved by end-to-end learning approach, combining the representation power of deep neural networks and training the network parameters with Stochastic Gradient Descent in conjunction with back-propagation to efficiently update model parameters.

Re-identification aims to match identities by visual information among various camera views. Main challenges of re-identification problem are two-fold: (a) visually matching *an* identity captured at different locations that vary in pose and illumination (high intra-class variance), (b) discrimination between *different* identities at same location and/or pose with similar visual appearance (low inter-class variance). Thus re-identification requires dealing with a two-faceted

problem of discrimination at the identity granularity. When it comes to *vehicle* re-identification, the problem of matching becomes even more challenging by the man-made characteristics of the subject data, that is, the similar (in some cases identical) visual appearance of different identities of certain vehicles with same make, model, and colour. To address these challenging aspects, this thesis investigates the effectiveness of approaches in the following original work:

1. Chapter 3 explores the man-made nature of vehicles, that is, their inherent hierarchical categories such as model (*e.g.*Audi-2011-A) and its usefulness in identity matching when identity pairwise labelling is not present. In contrast to existing methods, we uniquely bridge the connection between vehicle model classification and vehicle re-identification, by investigating the discrimination capability of vehicle model label supervised deep CNN features in performing finer-grained identity matching. Experiments show learning from vehicle model label supervision provide useful information for visual matching that can be used for vehicle re-identification.

2. Chapter 4 introduces a more realistic and challenging vehicle re-ID benchmark, called Vehicle Re-Identification in Context (VRIC). In contrast to existing datasets, VRIC is uniquely characterised by vehicle images subject to more realistic and unconstrained variations in resolution (scale), motion blur, illumination, occlusion, and viewpoint. Motion artefacts and varying resolution vehicle images provide a re-identification matching setting in wide-view traffic scenes "in-the-wild". Moreover, test set construction uniquely pairs high-low resolution probe-gallery images that puts emphasis on matching challenges of varying resolution input.

3. Chapter 5 evaluates image pyramid inspired multi-branch neural network design with multiple input resolution scales. Experiments evaluated on our varying resolution dataset VRIC has shown that visual representation training augmented by consensus feature identification loss feedback improves features compatibility between varying resolution scales with significant performance gains.

4. Chapter 6 proposes *Multi-Task Mutual Learning* (MTML) deep model to learn discriminative features simultaneously from multiple branches by combining multi-scale feature fusion in conjunction with additional orientation labels. Extensive comparative evaluations demonstrate the effectiveness of our proposed MTML method in comparison to the state-

of-the-art vehicle re-ID techniques and performance gains of each branch and labels are investigated with ablation studies. It has been also shown that orientation supervision has an observable affect on intra-class matching of vehicles captured by different pose.

7.2 Discussion and Future Work

In supervised learning, labels provide the *information* to be learned, on the other hand, the quality of the datasets, *e.g.*diversity and representative quality of the collected set of data, determine how successful the learned machine learning model can be. Data collection as well as labelling are time consuming and costly endeavours. Arguably the most important dataset in the recent computer vision literature is the *large scale* image classification benchmark ImageNet[10]. At this scale it was only feasible to be constructed with crawling the internet and very costly labelling process that followed. The scale of the data in training sets is one of the most important factors that enabled the performance gains we have enjoyed in the "deep learning" era. We have discussed how each computer vision task requires different labelling practices and as we move from general classification tasks to finer grained recognition tasks such as re-identification the labelling required is also of finer grain; *e.g.*individual identities rather than generic categories. We see here that there is a structural hierarchy in these tasks regarding the input domain and their labelling. In most specialised computer vision tasks transfer learning have become standard practice. When we train neural network models for narrower *i.e.downstream* computer vision tasks such as re-identification; they are trained by fine turning model parameters with new labels associated with the narrower problem domain. In this thesis *all* experiments, as well as the vast majority of the approaches for re-identification in recent literature, are trained by fine-tuning ImageNet classification model parameters *i.e.*transferring the information that was learned with generic category labels by re-training with labels of the downstream task *e.g.*re-identification. This fact attests to be usefulness of fine-tuning technique as well as the importance of rigorously labelled large scale datasets

Unlike supervised learning, semi-supervised training paradigms try to overcome costly labelling process by combining unlabelled data with smaller scale labelled data to take advantage of the available labelled data. Self-supervised learning is a new unsupervised learning paradigm where an where the structured data (images) itself provides the supervision. Both of these paradigms address the issue of task specific labelling that is required to achieve acceptable per-

formance in a supervised setting. Enabling the usage of the unlabelled data with self-supervised learning in combination with more traditional clustering approaches provides an exciting opportunity to discover the hierarchical relation of the downstream task labels with self-supervised learning, we can imagine CNN architectures that can do discrimination at different levels of this hierarchy with the same network parameters, an *all* purpose feature extractor. This would also change the fine-tuning approach we use in supervised learning for each task, that require learning different network parameters with discriminative characteristics at each label granularity. This of sort of approach that can learn from data itself, that require little or no labelling, is especially interesting for re-identification problem because it is simply not feasible to know/label all identities in real world scenarios.

Bibliography

[1] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41, 1984.

[2] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3908–3916, Boston, MA, USA, 2015.

[3] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems*, pages 41–48, Vancouver, BC, Canada, 2007.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[5] Rich Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[6] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*, Nothingham, UK, 2015.

[7] Yanbei Chen, Xiatian Zhu, Shaogang Gong, et al. Person re-identification by deep learning multi-scale representations. In *Workshop of IEEE International Conference on Computer Vision*, pages 2590–2600, Venice, Italy, 2017.

[8] Ying-Cong Chen, Xiatian Zhu, Wei-Shi Zheng, and Jian-Huang Lai. Person re-identification by camera correlation aware feature augmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):392–408, 2018.

[9] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 539–546 vol. 1, San Diego, CA, USA, 2005.

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

[11] Qi Dong, Shaogang Gong, and Xiatian Zhu. Multi-task curriculum transfer deep learning of clothing attributes. In *IEEE Winter Conference on Applications of Computer Vision*, pages 520–529, Santa Rosa, CA, USA, 2017.

[12] Rogerio Schmidt Feris, Behjat Siddiquie, James Petterson, Yun Zhai, Ankur Datta, Lisa M Brown, and Sharath Pankanti. Large-scale vehicle detection, indexing, and search in urban surveillance videos. *IEEE Transactions on Multimedia*, 14(1):28–42, 2012.

[13] Ross Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision*, pages 1440–1448, Santiago, Chile, 2015.

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, Columbus, OH, USA, 2014.

[15] Shaogang Gong, Marco Cristani, Shuicheng Yan, and Chen Change Loy. *Person re-identification*. Advances in Computer Vision and Pattern Recognition. Springer, January 2014.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[17] Mengran Gou, Srikrishna Karanam, Wenqian Liu, Octavia Camps, and Richard J. Radke. Dukemtmc4reid: A large-scale multi-camera person re-identification dataset. In *Workshop of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1425–1434, Honolulu, HI, USA, 2017.

[18] Hui-Zhen Gu and Suh-Yin Lee. Car model recognition by utilizing symmetric property to overcome severe pose variation. *Machine Vision and Applications*, 24(2):255–274, 2013.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE International Conference on Computer Vision*, volume abs/1502.01852, pages 1026–1034, Santiago, Chile, 2015.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, Las Vegas, NV, 2016.

[21] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv*, 2017.

[22] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[23] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[24] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[25] Qichang Hu, Huibing Wang, Teng Li, and Chunhua Shen. Deep cnns with spatially weighted pooling for fine-grained car recognition. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):3147–3156, 2017.

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine learning*, pages 448–456, 2015.

[27] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks, 2015.

[28] Aytaç Kanacı, Xiatian Zhu, and Shaogang Gong. Vehicle re-identification in context. In *German Conference of Pattern Recognition*, volume 11269 of *Lecture Notes in Computer Science*, pages 377–390, Stuttgart, Germany, 2018. Springer.

[29] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Workshop of IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, USA, 2011.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] Brendan F Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah, and Anil K Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1931–1939, Boston, MA, USA, 2015.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, Lake Tahoe, Nevada, United States, 2012.

[33] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, New York, NY, USA, 2006.

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

[35] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[36] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–159, Columbus, OH, USA, 2014.

[37] Wei Li, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep joint learning of multi-loss classification. In *International Joint Conference of Artificial Intelligence*, pages 2194–2200, Melbourne, Australia, 2017.

[38] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2285–2294, Salt Lake City, UT, USA, 2018.

[39] Liang Liao, Ruimin Hu, Jun Xiao, Qi Wang, Jing Xiao, and Jun Chen. Exploiting effects of parts in fine-grained categorization of vehicles. In *IEEE International Conference on Image Processing*, pages 745–749, Quebec City, QC, Canada, 2015.

[40] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2197–2206, Boston, MA, USA, 2015.

[41] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *IEEE International Conference on Computer Vision*, pages 1449–1457, Santiago, Chile, 2015.

[42] Yen-Liang Lin, Vlad I Morariu, Winston Hsu, and Larry S Davis. Jointly optimizing 3d model fitting and fine-grained classification. In *European Conference on Computer Vision*, pages 466–480, Zurich, Switzerland, 2014.

[43] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2167–2175, Las Vegas, NV, USA, 2016.

[44] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. Large-scale vehicle re-identification in urban surveillance videos. In *IEEE International Conference on Multimedia and Expo*, pages 1–6, Seattle, WA, USA, 2016.

[45] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European Conference on Computer Vision*, pages 869–884, Amsterdam, The Netherlands, 2016.

[46] Niki Martinel, Christian Micheloni, and Gian Luca Foresti. Kernelized saliency-based person re-identification through multiple metric learning. *IEEE Transactions on Image Processing*, 24(12):5645–5658, 2015.

[47] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499, Amsterdam, The Netherlands, 2016.

[48] Srivastava Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[49] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L. Yuille. Weakly- and semi-supervised learning of a DCNN for semantic image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1742–1750, 2015.

[50] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, pages 17–35, Amsterdam, The Netherlands, 2016.

[51] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, Boston, MA, USA, 2015.

[52] Yantao Shen, Tong Xiao, Hongsheng Li, Shuai Yi, and Xiaogang Wang. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In *IEEE International Conference on Computer Vision*, Venice, Italy, 2017.

[53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, San Diego, CA, USA, 2014.

[54] Jakub Sochor, Adam Herout, and Jiri Havel. Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3006–3015, Las Vegas, NV, 2016.

[55] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. In *IEEE International Conference on Computer Vision*, pages 3820–3828, Venice, Italy, 2017.

[56] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, Boston, MA, USA, June 2015.

[57] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, Las Vegas, NV, USA, 2016.

[58] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, Columbus, OH, USA, 2014.

[59] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8797–8806, 2019.

[60] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[61] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, Columbus, OH, USA, 2014.

[62] Zhongdao Wang, Luming Tang, Xihui Liu, Zhuliang Yao, Shuai Yi, Jing Shao, Junjie Yan, Shengjin Wang, Hongsheng Li, and Xiaogang Wang. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *IEEE International Conference on Computer Vision*, pages 379–387, Venice, Italy, 2017.

[63] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *arXiv preprint arXiv:1511.04136*, 2015.

[64] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515, Amsterdam, The Netherlands, 2016.

[65] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff

Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

[66] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1249–1258, Las Vegas, NV, USA, 2016. IEEE.

[67] Dongwu Xu, Congyan Lang, Songhe Feng, and Tao Wang. A framework with a multi-task cnn model joint with a re-ranking method for vehicle re-identification. In *ACM International Conference on Internet Multimedia Computing and Service*, pages 1:1–1:7, Nanjing, China, 2018.

[68] Ke Yan, Yonghong Tian, Yaowei Wang, Wei Zeng, and Tiejun Huang. Exploiting multi-grain ranking constraints for precisely searching visually-similar vehicles. In *IEEE International Conference on Computer Vision*, pages 562–570, Venice, Italy, 2017.

[69] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3973–3981, Boston, MA, USA, 2015.

[70] Dominik Zapletal and Adam Herout. Vehicle re-identification for automatic video traffic surveillance. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1568–1574, Honolulu, HI, USA, 2016.

[71] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108, Zurich, Switzerland, 2014.

[72] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Learning deep representation for face alignment with auxiliary attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(5):918–930, 2016.

[73] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3652–3661, Honolulu, HI, USA, July 2017.

[74] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. Camera style adaptation for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5157–5166, Salt Lake City, UT, USA, 2018.

[75] Yi Zhou and Ling Shao. Viewpoint-aware attentive multi-view inference for vehicle re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6489–6498, Salt Lake City, UT, USA, 2018.

Appendix A

Machine Learning and Artificial Neural Network Fundamentals

A.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are powerful family of neural networks that are designed build efficient models for learning from image data. Modern CNNs have multiple convolutional layers where each layer contains a number of neurons with non-linearities to enable powerful computations. Each hidden layer computes a simple affine transformation combined with a non-linearity. Modern hardware such as Graphical Processing Unit(GPU)s that can compute hundreds of such transformations calculating various input-output relations by each hidden layer. The resulting neural network stacking such layers one after the other with millions of neurons are deep convolutional neural networks that can achieve human like processing capability of said input, extracting higher level relations.

Artificial Neurons

The basic building blocks of a neural network are called *neurons*. Foundations of modern neural networks called *perceptron* were developed in 1950s which could learn linear models of binary input-output relations. Modern neural networks use neurons compute non-linear outputs of it inputs such as the *sigmoid* neuron. This is done by (1) computing an affine transformation of inputs and weights that are continuous - that is a linear weighting of inputs x by weights w and a translation of the output called *bias* b to get the output z (2) the output is fed to a non-linear function *e.g.*sigmoid function. Sigmoid function σ (see Figure A.1) has range $(0,1)$ and defined

as:

$$\sigma(x) \equiv \frac{1}{1+e^{-x}}. \tag{A.1}$$

Formally, when our inputs consist of d features, the affine transformation, *i.e.*output of a given neuron, usually denoted by z is defined as:

$$z = w_1 x_1 + ... + w_d x_d + b.$$

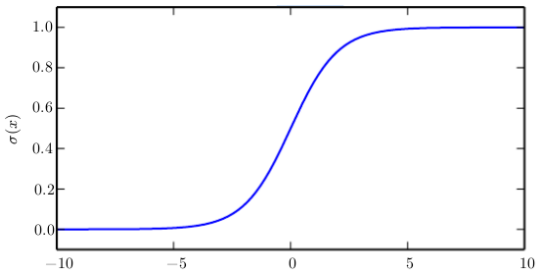

Figure A.1: Sigmoid activation function

Vectorizing the equation for compact representation with dot product:

$$z = \mathbf{w}^\top \mathbf{x} + b$$

where $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{w} \in \mathbb{R}^d$. Finally we can use the non-linear sigmoid function to express an artificial neuron by:

$$z = \sigma(\mathbf{w}^\top \mathbf{x} + b). \tag{A.2}$$

Fully Connected Layers

A layer is a collection of neurons. By stacking and connecting layers of neurons one after the other we can get *artificial neural network* models. See Figure A.2 for an example diagram. Suppose we have m number of neurons, considering the all of neurons in layer, we can formulate all weights in needed for a layer as matrix $\mathbf{W} \in \mathbb{R}^{d \times m}$ and bias vector $\mathbf{b} \in \mathbb{R}^m$ finally defining a layer in matrix notation as:

$$\mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}. \tag{A.3}$$

The layer is said to be *fully-connected* because each of its inputs is connected to each of its outputs by means of a matrix-vector multiplication. First and last layers of a network are called *input* and *output* layers respectively, whereas intermediate layers are called *hidden* layers.

Convolutional Layers

Convolutional layers are the basic building blocks of modern convolutional neural network. For image data with a 2D structure CNNs are computationally efficient because they require fewer parameters than fully-connected architectures where each neuron has input from all of the outputs from the previous layer.

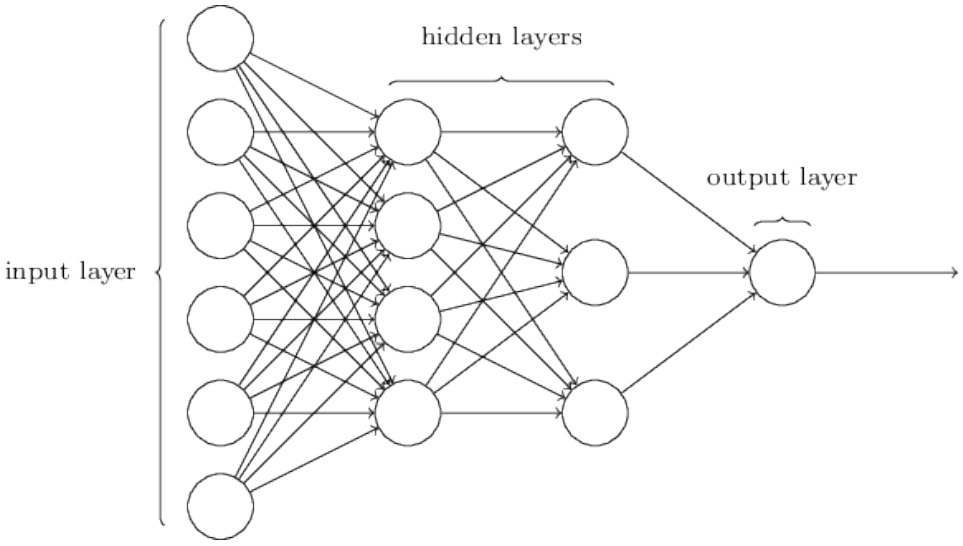

Figure A.2: An example artificial neural network model with input, hidden and output layers. All layers are *fully connected.*

The convolution between two functions, $f, g : \mathbb{R}^d \to \mathbb{R}$ and denoted by $(f * g)$, is defined as:

$$(f * g)(i) = \int f(i)g(i-m)di.$$

For two-dimensional tensors, such as 2D images with discrete values rather than functions, we have a corresponding sum: with indices (m,n) for f and $(i-m, j-n)$ for g, respectively:

$$h(i,j) = (f * g)(i,j) = \sum_m \sum_n f(m,n)g(i-m, j-n). \tag{A.4}$$

In convolutional network terminology, the first argument (the function f) to the convolution is often referred to as the input and the second argument (the function g) as the kernel. For two-dimensional images $\mathbf{X}$ as inputs and their immediate hidden representations $\mathbf{H}$ and neurons now called kernels $\mathbf{K}$ all similarly represented as tensors.

Let $[\mathbf{X}]_{i,j}$ and $[\mathbf{H}]_{i,j}$ denote the pixel at location $(i,\ j)$, in the input image and hidden representation respectively. While $[\mathbf{K}]_{m,n}$ denote kernel weights $\mathbf{K}$ and $\mathbf{B}$ vector denoting biasses b for each kernel.

$$[\mathbf{H}]_{i,j} = b + \sum_{m=-\Delta}^{\Delta} \sum_{n=-\Delta}^{\Delta} [\mathbf{K}]_{m,n}[\mathbf{X}]_{i-m,j-n} \tag{A.5}$$

In this formulation $[\mathbf{X}]$ and $[\mathbf{K}]$ assumed to have the same dimensions however outside the range Δ: $|m| > \Delta$ or $|n| > \Delta$ we should set the $[\mathbf{K}_{m,n} = 0]$. In practice these weights in $[\mathbf{K}]$ outside the range Δ are omitted resulting in desired reduction of parameters in CNNs. Moreover *translation invariance* and *locality* properties which are crucial for processing images are achieved.

Translation invariance in images implies that all patches of an image will be treated in the same manner whereas locality means that only a small neighbourhood of pixels will be used to compute the corresponding hidden representations. Figure A.3 illustrates a 2D convolution example calculation with a single kernel.

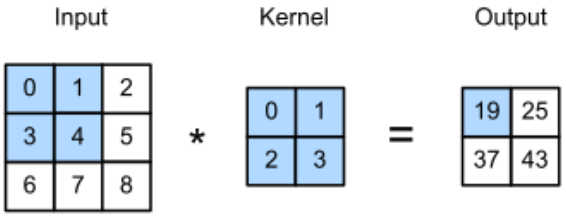

Figure A.3: Illustration of convolution operation with a 2D 3×3 input and a 2×2 kernel with the resulting output, omitting the bias term.

Colour images have three channels: red, green, and blue. Formally they are third-order tensors, characterized by a height, width, and channel, e.g., with shape $256 \times 256 \times 3$ pixels. We thus index X as $[\mathsf{X}]_{i,j,k}$. The convolutional hence instead becomes $[\mathsf{K}]_{m,n,l}$. Hidden representations are third-order tensors H – an entire vector of hidden representations with an extra dimension corresponding to each spatial location. Extra dimension is naturally named *channels* or *feature maps*.

To support multiple channels in both inputs (X) and hidden representations (H), we can add a fourth coordinate to K: $[\mathsf{K}]_{m,n,l,d}$. Hence we have:

$$[\mathsf{H}]_{i,j,d} = \sum_{m=-\Delta}^{\Delta} \sum_{n=-\Delta}^{\Delta} \sum_{l} [\mathsf{K}]_{m,n,l,d} [\mathsf{X}]_{i+m,j+n,l} \tag{A.6}$$

where d indexes the output channels in the hidden representations H. The subsequent convolutional layer will go on to take a third-order tensor, H, as the input.

Non-linear Activations

Other than the sigmoid function (σ) defined in eq A.1 there are two other commonly used non-linear functions, namely *tanh* and *ReLU*. The definitions are the following:

$$\tanh(x) = \frac{1 - \exp(-2x)}{1 + \exp(-2x)} \tag{A.7}$$

$$\mathrm{ReLU}(x) = \max(x, 0) \tag{A.8}$$

Activation functions differentiable operators that provide non-linear transform for signal passing through a neuron. While *sigmoid* and *tanh* outputs a smooth signal in the range $(0, 1)$,

ReLU simply suppresses negative signals, allowing only positive signal to effect later layers, thus activating each neuron or not. In CNNs ReLU activations are much preferred for computer vision tasks as they provide a way to make each neuron *decide* rather than smoothly passing all range of signals for later layers.

Stride and Pooling

CNNs are effectively feature extractors modelled with convolutional layers one after the other. Compact and high level representations of images are achieved by dowmsampling intermediate hidden feature layers without ignoring important information. Input images as well as intermediate feature maps provide possibly redundant information considering commonly used kernel sizes of 3×3 and 5×5 by the locality principle.

During convolution operation starting at the top-left corner of the input tensor, computation progresses by sliding the kernel over all locations both down and to the right. Using a *stride* kernels skip processing at intermediate positions of the input tensor, increasing computational efficiency by effectively reducing the number of affine transforms that would contain relatively similar input that is offset by translation in the small receptive field of the kernels.

Taking the input elements in the pooling window, the maximum pooling operation assigns the maximum value as the output and the average pooling operation assigns the average value as the output. gradually reduce the spatial resolution of our hidden representations, aggregating information so that the higher up we go in the network,

A.2 Gradient-Based Learning

Classification Problem

Classification problem is defined as determining whether an input belongs to a predefined set of classes. Classification is a supervised learning problem where we learn a prediction model from data and their correct labels. A dataset is a collection training samples, *i.e.*pairs input data and their corresponding labels. Formally, a training set $\mathbb{S}$ consists of pairs of vectors such that $\mathbb{S} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{n}$ where n is the number of training samples. The feature vector $\mathbf{x}^{(i)}$ and a one-hot label vector $\mathbf{y}^{(i)}$ have dimensions m and k respectively, where d is the input dimension and k is the number of classes for our problem, *i.e.*$\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^k$.

Cross-Entropy Loss

In machine learning, learning is done through optimisation of a model parameters according to an

objective, the loss function. For a multi-label classification problem, we adopt the *cross-entropy* loss. This means we use the cross-entropy between the training data and the model's predictions as the loss function.

Consider the model $\mathbf{z} = f(\mathbf{x}|\boldsymbol{W})$ where $\mathbf{z} \in \mathbb{R}^k$ is output feature vector, k is the number of classes and $\boldsymbol{W}$ as the model parameters. We require the parametric model f to define a distribution:

$$\hat{\mathbf{y}} = P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{W}) \tag{A.9}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^k$ defines a multi-class probabilistic prediction vector over all classes, as such, we require not only that each element of $\hat{y}_i \in [0,1]$, but also that the entire vector sums to 1 so that it represents a valid probability distribution. This is achieved with the *softmax function*:

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{z}) \quad \text{where} \quad \hat{y}_j = \frac{\exp(z_j)}{\sum_k \exp(z_k)} \tag{A.10}$$

such that $\hat{y}_j$ become non-negative and sum to 1; $\sum_{i=1}^{k} \hat{y}_i = 1$. In other words, the softmax function takes a feature vector $\mathbf{z}$ and maps it into probabilities $\hat{\mathbf{y}}$. The model output $\hat{y}_j$ is to be interpreted as the estimated posterior probability that an input $\mathbf{x}$ belongs to class j.

.

We will optimise the model parameters $\boldsymbol{W}$ to produce probabilities that maximise the likelihood of the observed data $\mathbf{X}$, given labels $\mathbf{Y}$ This is referred as the *maximum likelihood* estimation. Model f provides a probabilistic output $\hat{\mathbf{y}} = P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{W})$, we can compute the maximum likelihood

Let $\mathbf{X}$ represent all our inputs and $\mathbf{Y}$ all our observed targets, where the example indexed by i consists of a feature vector $\mathbf{x}^{(i)}$ and a one-hot label vector $\mathbf{y}^{(i)}$. We also assume the training samples are independent and identically distributed. Joint probability of dataset can be computed as a product over the individual conditional probabilities and the maximum likelihood of the entire dataset can be written as:

$$P(\mathbf{Y} \mid \mathbf{X}) = \prod_{i=1}^{n} P(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}). \tag{A.11}$$

We maximise $P(\mathbf{Y} \mid \mathbf{X})$, equivalent to minimising the negative log-likelihood:

$$-\log P(\mathbf{Y} \mid \mathbf{X}) = \sum_{i=1}^{n} -\log P(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}) = \sum_{i=1}^{n} \mathcal{L}_{CE}(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)}), \tag{A.12}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss function.

The negative log likelihood $-\log P(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)})$ or equivalently the cross-entropy $\mathcal{L}_{CE}$ between training label $\mathbf{y}$ and model's prediction $\hat{\mathbf{y}}$ over k classes is computed by:

$$\mathcal{L}_{CE} = -\sum_{j=1}^{k} y_j \log \hat{y}_j \tag{A.13}$$

Since $\mathbf{y} = (y_1, \ldots, y_C)$ is a one-hot vector that encodes ground-truth label $y \in \{1, \ldots, C\}$ by assigning the y-th term as 1 and others as 0, *i.e.*,

$$y_j = \begin{cases} 1 & : j = y \\ 0 & : j \neq y \end{cases}, \tag{A.14}$$

the sum over all its terms j is multiplied by 0 for all but one term following the definition of $\mathcal{L}_{CE}$.

Minibatch Stochastic Gradient Descent

The objective function is usually the average of the loss functions for each example in the training dataset. We assume that $f_i(\mathbf{x})$ is the loss function of the training dataset with n examples, an index of i, and parameter vector of $\mathbf{x}$, then we have the objective function

$$f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}).$$

The gradient of the objective function at $\mathbf{x}$ is computed as

$$\nabla f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\mathbf{x}).$$

in vanilla gradient descent training dataset is large, the cost of gradient descent for each iteration will be very high. There is a trade-off between statistical efficiency arising from SGD and computational efficiency arising from processing large batches of data at a time. Minibatch stochastic gradient descent offers the best of both worlds: computational and statistical efficiency. Stochastic gradient descent (SGD) reduces computational cost at each iteration. At each iteration of stochastic gradient descent, we uniformly sample an index $i \in \{1, \ldots, n\}$ for data examples at random, and compute the gradient $\nabla f_i(\mathbf{x})$ to update $\mathbf{x}$:

$$\mathbf{x} \leftarrow \mathbf{x} - \eta \nabla f_i(\mathbf{x}).$$

Here, η is the learning rate. whenever we perform

$\mathbf{w} \leftarrow \mathbf{w} - \eta_t \mathbf{g}_t$ where

$$\mathbf{g}_t = \partial_{\mathbf{w}} f(\mathbf{x}_t, \mathbf{w})$$

We can increase the computational efficiency of this operation by applying it to a minibatch of observations at a time. That is, we replace the gradient $\mathbf{g}_t$ over a single observation by one over a small batch

$$\mathbf{g}_t = \partial_{\mathbf{w}} \frac{1}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} f(\mathbf{x}_i, \mathbf{w})$$

Minibatch SGD had the nice side-effect that averaging gradients reduced the amount of variance. The minibatch SGD can be calculated by:

$$\mathbf{g}_{t,t-1} = \partial_{\mathbf{w}} \frac{1}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} f(\mathbf{x}_i, \mathbf{w}_{t-1}) = \frac{1}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} \mathbf{h}_{i,t-1}$$

To keep the notation simple, here we used

$\mathbf{h}_{i,t-1} = \partial_{\mathbf{w}} f(\mathbf{x}_i, \mathbf{w}_{t-1})$

as the SGD for sample i using the weights updated at time $t-1$.

$$\mathbf{v}_t = \beta \mathbf{v}_{t-1} + \mathbf{g}_{t,t-1}$$

for some $\beta \in (0,1)$. This effectively replaces the instantaneous gradient by one that's been averaged over multiple *past* gradients. $\mathbf{v}$ is called *momentum*. Momentum prevents stalling of the optimisation process that is much more likely to occur for stochastic gradient descent.