Deep Representation Learning in Computer Vision and Its Applications

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Fangyu Wu

November 2020

Dedication

This thesis is wholeheartedly dedicated to my beloved husband, Bowen Zhang, who have been my source of inspiration and gave me strength when I thought of giving up, who continually provide their spiritual, emotional, and financial support.

To my parents, friends, and colleagues who shared their words of advice and encouragement to finish the PhD study.

Abstract

Over the past decade, a branch of machine learning, called deep learning, has achieved remarkable successes in various computer vision tasks such as image classification, object detection, semantic segmentation, action recognition and image description generation. Deep learning aims at discovering multiple levels of distributed representations, which have been validated to be discriminatively powerful in many tasks. Distributed representation describes the same data features across multiple scalable and interdependent layers. Each layer defines the information with the same level of accuracy, but adjusted for the level of scale. The performance of deep learning methods depends heavily on the choice of data representation (or features) on which they are applied. Representation learning aims to learn representations of input data typically by transforming it or extracting features from it, which makes it easier to perform a task like classification or prediction.

Representation learning has been studied for many years in the field of conventional computer vision algorithms. The development and deployment of representation learning in deep learning algorithms are of vital importance since powerful deep models are proposed and also show an improving effect in many real-world applications. Focussing on deep learning, representation learning is the consequence of the function a model learns when the learning is captured in the parameters.

This thesis focus on representation learning in deep learning, starting from the recent progress in representation learning mechanism, followed by several contributions on representation learning targeting diverse applications in computer vision, including

vehicle re-identification (re-ID), traffic scene recognition, face recognition and few-shot classification.

For the traffic scene recognition, our contribution is twofold: firstly, we propose a novel traffic scene recognition methodology in the setting of granular computing, which involves the creation of information granulation by extracting the deep features upon local regions of the image for a compact feature representation, and design classifiers fusion method to further improve the performance of traffic scene recognition. Information granulation involves the process of data abstraction and derivation of knowledge from information or data. The second contribution is the creation of a new traffic scene dataset, named the "WZ-traffic". The WZ-traffic dataset consists of 6,035 labeled images which belong to 20 categories collected from both an image search engine as well as from personal photographs. The experiment results demonstrate that our method dramatically improves traffic scene recognition and brings potential benefits to many other real-world applications.

For the task of vehicle re-ID, most existing algorithms are developed in the fully-supervised setting, requiring access to a large number of labeled training data. To alleviate the large demand of training data and improve the performance of representation learning, we propose a semi-supervised deep learning scheme which makes learning rich feature representations from a limited number of labeled data possible. Secondly, we present a re-ranking algorithm for ranking optimization which is first introduced for the vehicle re-ID task. Since the sample label is not required, the process of the re-ranking algorithm can be performed in unsupervised learning. The experimental results show that the proposed networks achieved state-of-the-art performance on several benchmark datasets.

Although various face recognition methods in controlled environments have been proposed and achieved promising performance, there are still many challenges posed by uncontrolled environments.In this thesis, we design more powerful representation learning algorithms to address the challenges of various variations, including disguise accessories, illumination and pose. The experimental results show that the proposed networks achieved

state-of-the-art performance on several benchmark datasets.

For the few-shot classification, there are two main contributions in this thesis: firstly, we attempt to tackle the few-shot classification problem based on a novel representation learning model, named Capsule network, which combines the 3D convolution-based dynamic routing procedure to obtain a deep feature representation with semantic and spatial information. Secondly, we propose a novel attentive prototype concept to take account of all the instances in a given support class. Each instance is weighted by the reconstruction errors between the query and prototype candidates from the support set. The attentive prototype is robust to outliers by design and allows the performance to be improved by refraining from making predictions in the absence of sufficient confidence.

In conclusion, comprehensive research was carried out for the feature representation learning in computer vision and its applications, which include traffic scene recognition, vehicle re-ID, face recognition in uncontrolled environment and few-shot classification. Related research topics have also been discussed, for example, alleviating the large demand of training data by semi-supervised learning and domain adaptation. For the above computer vision applications, this thesis presents several contributions which proved to be effective in improving existing methods.

Acknowledgements

I would like to express my sincere gratitude to Prof. Bailing Zhang, my primary supervisor, who provided me an opportunity for a research study and constantly guides me in the area of machine learning and computer vision. My heartfelt thanks also go to Dr. Wenjin Lu and Prof. Jeremy S.Smith, my co-supervisors, for their valuable help and suggestions for my PhD study.

I want to express my thanks to my advisors, Dr.Wei Wang and Dr. Waleed AI- Nuaimy, who helped to evaluate my PhD studies and provide valuable suggestions for the research process.

I also want to thank all the co-authors of the published research, who offered advice, help, and comments to my research, which are of great help during the PhD study, especially Prof. Chaoyi Pang and Dr. Han Liu.

I want to thank many friends from the School of Advance Technology, especially Yuechun Wang, Qi Chen, Jing Qian, Yuxuan Zhao, Ziqiang Bi, Xianbin Hong and Xiaoxiao Wang. I also owe my sincere gratitude to my old friends who gave me their help and time in listening to me and helping me during the difficult time of these years.

Finally, I express my gratitude to my beloved parents who have always been helping me out of difficulties and supporting me without a word of complaint.

Contents

List of Figures

List of Tables

List of Abbreviations

CNNs Convolutional Neural Networks.

GANs Generative adversarial networks.

MLP Multi-layer Perceptron.

ReLU Rectified Linear Unit.

SPP Spatial Pyramid Pooling.

RoI Region of Interest.

RNNs Recurrent Neural Networks.

MTC Manifold Tangent Classifier.

PCA Principal Component Analysis.

BOF Bag of Features.

SIFT Scale Invariant Feature Transform.

HoG Histogram of Oriented Gradient.

BoVW Bag of Visual Words.

FV Fisher Vectors.

VLAD Vector of Locally Aggregated Descriptors.

SVM Support Vector Machine.

KNN k-Nearest Neighbours.

DT Decision Tree.

RF Random Forest.

GBT Gradient Boosted Trees Learner.

re-ID Re-identification.

LSRO Label Smoothing Regularization.

LSRO Label Smoothing Regularization for Outliers.

LBP Local Binary Patterns.

AP Average Precision

mAP Mean Average Precision.

CMC Cumulative Match Curve.

XQDA Cross-view Quadratic Discriminant Analysis.

FR Face Recognition.

DFR Disguised Face Recognition.

FC Fully Connected.

SGD Stochastic Gradient Descent.

NIR Near Infrared.

VLD Visible Light Domain.

MTCNN Multi-Task Cascaded Convolutional Network.

ILSVRC-2012 ImageNet Large Scale Visual Recognition Challenge-2012.

RL Representation Learning.

AI Artificial Intelligence.

ICA Independent Component Analysis.

RBM Restricted Boltzmann Machines.

LDA Linear Discriminant Analysis

Chapter 1

Introduction

1.1 Overview

Machine learning has powered many aspects of modern society: from conventional industry to current internet business such as web search engine, social networks, and content filtering. It is continuing to increase its impact on modern life. To name a few, the functionalities of machine learning include recognizing objects in images, translating one language to another, matching news items and recommending news based on user's interests.

Recently, one of the branches of machine learning family called deep learning has shown dominant performance in tasks mentioned previously and becomes increasingly important in machine learning and artificial intelligence. Conventional machine learning techniques were usually limited in their ability to process natural data in their raw form. For decades, constructing pattern recognition system required careful engineering and considerable domain expertise to design a feature extractor that transformed the raw data into a suitable internal representation, often a classifier or predictor, could classify or predict patterns in the input. These hand-crafted features, if not appropriately designed, could severely deteriorate the system performance. On the other hand, deep representation learning, is a set of learning methods that can be fed with only raw data and automatically discover the internal representation of the data during the process of learning.

Zeiler et al. [281] has given an empirical view of what the representation learning means, taking the example of one of the most popular models in deep learning called Convolutional Neural Network (CNN) [131]. In [281], the authors visualize each of the layers in the trained CNN to find what each layer represents. Interestingly, an image, for

example, comes in the form of an array of raw pixels, and the learned features in the first layer of representation usually represent the presence of edges at particular orientations and locations in the image. The second layer typically detects motifs by spotting specific arrangements of edges, regardless of small variations in the edge positions. The third layer may assemble motifs into larger combinations that correspond to parts of familiar objects, and subsequent layers would detect objects as combinations of these parts. The representation of the CNN becomes more abstract in the higher layers than the lower layers. The CNN is an example of representation learning, and show excellent performance in various machine learning tasks.

Despite their success, CNN suffer from inherent limitations: firstly, a deep CNN is trained with sufficient training samples, however, it's infeasible to annotate sufficient training samples in some tasks. Instead of only relying on the labeled data, an alternative scheme is with the help of a Generative Adversarial Network (GAN) [81] to learn good feature representation. GAN was first proposed to generate realistic images. GAN learns generative models without explicitly defining a loss function from the target distribution. Instead, GAN introduces a discriminator network which tries to differentiate real samples from generated samples. The whole network is trained using this adversarial training strategy. Recently, GAN have also been applied to image-to-image translation [299] which aims at learning a mapping function between two domains.

Secondly, although the pooling layer of CNN brings the advantages of reducing the computational complexity and translation invariance, it also loses the location information of relevant features. In addition to position, other instanced parameters, such as scale and rotation, which describe the characteristics of the object, cannot be effectively taken into account in the convolutional network. Hinton et al. introduced an efficient way to handle these instantiation parameters by a Capsule network [96]. A Capsule represents an object or a part of an object, whose activity vector encodes the instantiation parameters of that part.

More specifically, a Capsule network replaces the mechanisms of CNN's convolution kernel, which works independently of each other. If two convolution kernels are trained to activate two specific parts of an object, the same amount of activation will be generated regardless of the relative position of the object. Capsule network works by implementing a group of neurons to encode the spatial information and the probability of objects' existence. The length of the Capsule vector is the probability of the features in the image, and the orientation of the vector will represent its instantiation information. There are two different

capsule layers in the Capsule network proposed in [96]: a primary capsule layer groups convolutions to work together as a capsule unit, and a digit capsule layer obtained by calculating the agreement among different capsules through dynamic routing.

In this thesis, following the basic idea of the GAN and Capsule network, we employed, extended and improved the current representation learning methods in several computer vision tasks, including traffic scene recognition, vehicle re-ID, face recognition in uncontrolled environments and few-shot classification. In this thesis, the four important computer vision applications can be realized with the aid of the representation learning to improve the final performance in a challenging dataset. For traffic scene recognition, the compact representation associated with the traffic scene obtained from extracting the CNN features upon local regions of the image. For the vehicle re-ID, a semi-supervised deep learning scheme for vehicle re-ID task makes learning rich feature representations of vehicles from a limited number of labeled data. For the face recognition in uncontrolled environments with disguise accessories, illumination and pose, the representation learning is to achieve with the related research topics, including unsupervised domain adaptation, image-image translation and meta-learning, respectively. For the few-shot classification, feature representation was created from a Capsule network-based embedding module.

1.2 Motivations and Challenges

1.2.1 Motivations

As a paradigm shift in feature generation, representation learning techniques are considered as an important and inevitable part of state-of-the-art pattern recognition systems. These techniques attempt to extract and abstract essential information from raw input data. Representation learning-based methods of feature generation are in contrast to classical feature generation methods which are mainly based on the prior knowledge of the expert about the task. Moreover, deep representation learning has promoted the development of computer vision. Deep representation learning methods are necessary for AI-level applications that need to learn complicated functions that represent high-level abstractions. Existing deep representation learning approaches, exclusively rely on deep CNN to compute a holistic feature of each input image, which has several well-known problems: (1) A large number of annotated images are required by the CNN-based methods to obtain high performance, which is an obstacle for some computer vision tasks. (2) the pooling operation

discards the location information of an entity that the network may attempt to identify; (3) the spatial relationships between simple objects are missing. CNN inadequately learn to recognize the relationships between parts, wholes, and the importance of their instantiation makes it less competent to be an effective learner.

Inspired by above considerations, our main motivations for this thesis are to research the solutions of the CNN's limitations of representation learning in computer vision. Besides, we test the feasibility of the proposed methods for diverse and challenging real-world applications in computer vision, which include the traffic scene recognition, vehicle re-ID, face recognition in uncontrolled environments and few-shot classification.

1.2.2 Challenges

- Traffic scene recognition is a challenging task for still images. A traffic scene is generally composed of a collection of entities (e.g., objects) organized in a highly variable layout. Hence, a recognition model must fully consider the geometric invariance and transfer information about local elements from a still image. The challenge for this task is employ deep representation learning to discover the local information in an image.

- As discussed previously, the CNN-based method requires access to a large number of labeled training data. However, the data annotation is costly, the vehicle bounding box must be drawn and match a vehicle ID label, which results in the insufficiently labeled training data for vehicle re-ID. The challenge lies in applying a semi-supervised learning paradigm to exploit unlabelled data and improving the performance of the vehicle re-ID task.

- Many innovative methods have been put forward for face recognition and verification, and the accuracy of recognizing clear human faces in well-controlled environment is generally very high. However, the accuracy of current automated human face recognition systems degrades for uncontrolled conditions like pose, illumination, expression, and occlusion, etc. The challenge for this task is how to employ the representation learning to address the changes in lighting conditions, pose and disguise accessories. The reliability and robustness are important for these face recognition applications, particularly in security systems.

- The few-shot classification in computer vision is difficult since it tackles the problem

of classifying unseen data instances into new categories, given just a small number of labeled instances in each class. Besides, using the limited training samples to learn an embedding space with the representation learning is a challenging task, which tries to achieve a high-level intelligence since humans can rapidly learn novel visual concepts from only one or a few examples and then reliably recognize them later.

1.3 Thesis Contributions

- A comprehensive analysis of the representation learning in computer vision is presented in this thesis. Four applications of computer vision and deep learning, namely, the traffic scene recognition, vehicle re-ID, face recognition in uncontrolled environments and few-shot classification are discussed. Especially, in both of the four applications, representation learning methods are used to improve system performance.

- For the task of traffic scene recognition, an end-to-end representation learning network is proposed to extract feature maps from the local region to improve the discriminating capability of the model for the task. It is worthy to mention that this is one of the early attempts to implement local deep-learned feature extraction in a CNN model. (Chapter 3)

- For the task of vehicle re-ID, we adopted a GAN to generate unlabeled samples and enlarge the training set. A semi-supervised learning scheme with the CNN was proposed accordingly, which assigns a uniform label distribution to the unlabeled images to regularize the supervised model and improve the performance of the vehicle re-ID system. Besides, an improved re-ranking method based on Jaccard distance and k-reciprocal nearest neighbors is proposed to optimize the initial rank list. (Chapter 4)

- For the task of face recognition in uncontrolled environment, we propose representation learning methods for different variations, including disguise accessories, illumination and pose. Most existing disguised face recognition approaches follow a supervised learning framework. However, due to the domain shift problem, the CNN model trained on one dataset often fail to generalize well to another dataset. We proposed a novel Unsupervised Domain Adaptation Model (UDAM) to address the challenging

face recognition with domain bias. Following this idea, we further address the domain bias between the near infrared (NIR) image and the visual light (VIS) image via NIR-VIS image translation. The NIR-VIS image conversion model can transform near-infrared facial images into their corresponding VIS images while maintaining sufficient identity information to enable existing VIS facial recognition models to perform recognition. (Chapter 5)

- For the pose-robust face recognition, we propose a deep meta Capsule network-based Equivariant Embedding Model (DM-CEEM) with three distinct novelties. First, the proposed RB-Capsule network allows DM-CEEM to learn an equivariant embedding for pose variations and achieve the desired transformation for input face images. Second, we introduce a new version of a Capsule network called RB-Capsule network to extend Capsule network to perform a profile-to-frontal face transformation in deep feature space. Third, we train the DM-CEEM following the meta-learning strategy by treating a single overall classification target as multiple sub-tasks that satisfy specific unknown probabilities. In each sub-task, we sample the support and query sets randomly. (Chapter 5)

- We propose a new feature representation learning structure to encode relative spatial relationships between features by applying a Capsule network for few-shot classification. Besides, a new triplet loss designated to enhance the semantic feature embedding where similar samples are close to each other while dissimilar samples are farther apart; and an effective non-parametric classifier termed attentive prototypes in place of the simple prototypes in current few-shot classification. The proposed attentive prototype aggregates all of the instances in a support class, which are weighted by their importance, defined by the reconstruction error for a given query. Extensive experiments on three benchmark datasets demonstrate that our approach is effective for the few-shot classification task. (Chapter 6)

1.4 Thesis Structure

A diagram of the thesis structure is shown in Figure 1.1. A general introduction of the topic and background is provided in the introduction, followed by the preliminaries of deep learning and the representation learning applied in this thesis. Subsequently, four applications, namely, the traffic scene recognition, vehicle re-ID, face recognition in

uncontrolled environments and few-shot classification, are introduced with representation learning. Specifically, the topics which are all powered by the application of the proposed representation learning methods. Lastly, a conclusion of this thesis and future works are introduced in the last chapter.

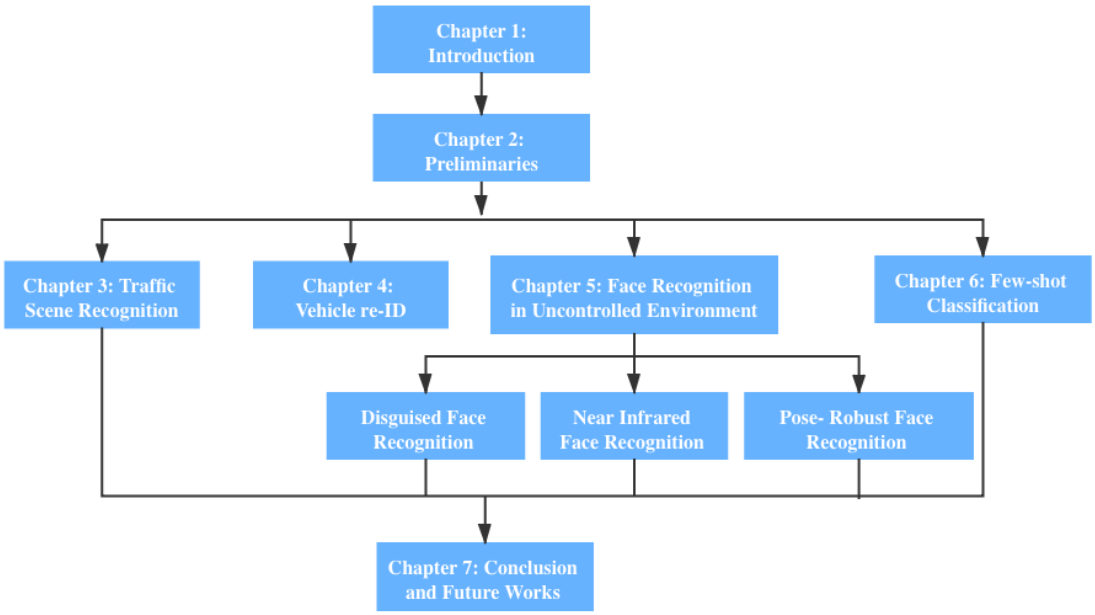


Figure 1.1: The structure of this thesis.

1.5 Publications

1.5.1 Periodical Papers

1. **Fangyu Wu**, Shiyang Yan, Jeremy S. Smith and Bailing Zhang. "Vehicle Re-identification in Still Images: Application of Semi-supervised Learning and Re-ranking, " Signal Processing-Image Communication, 2019.

2. **Fangyu Wu**, Shiyang Yan, Jeremy S. Smith, Wenjin Lu and Bailing Zhang. "Deep Multiple Classifiers Fusion for Traffic Scene Recognition, " Granular Computing, 2019.

3. **Fangyu Wu**, Jeremy S. Smith, Wenjin Lu, Chaoyi Pang, and Bailing Zhang. "Cap-

sule network based Embedding and an Attentive Prototypes for Few-Shot Learning," IEEE Transactions on Neural Networks and Learning Systems. (Under review)

1.5.2 Conference Papers

1. **Fangyu Wu**, Jeremy S.Smith, Wenjin Lu, Chaoyi Pang, and Bailing Zhang. "Attentive Prototype Few-shot Learning with Capsule Network-based Embedding," European Conference on Computer Vision (ECCV), 2020.

2. **Fangyu Wu**, Jeremy S. Smith, Wenjin Lu, and Bailing Zhang. "Pose-robust Face Recognition by Deep Meta Capsule network-based Equivariant Embedding," IEEE International Conference on Pattern Recognition (ICPR), 2020.

3. **Fangyu Wu**, Weihang You, Jeremy S. Smith, Wenjin Lu and Bailing Zhang. "Image-Image Translation to Enhance Near Infrared Face Recognition." IEEE International Conference on Image Processing (ICIP), 2019.

4. **Fangyu Wu**, Shiyang Yan, Jeremy S. Smith, Wenjin Lu and Bailing Zhang. "Unsupervised Domain Adaptation for Disguised Face Recognition." IEEE International Conference on Multimedia Expo (ICME), 2019.

5. **Fangyu Wu**, Shiyang Yan, Jeremy S. Smith and Bailing Zhang. "Joint Semi-supervised Learning and Re-ranking for Vehicle Re-identification." IEEE International Conference on Pattern Recognition (ICPR), 2018.

6. **Fangyu Wu**, Shiyang Yan, Jeremy S. Smith and Bailing Zhang. "Traffic Scene Recognition Based on Deep CNN and VLAD Spatial Pyramids." IEEE International Conference on Machine Learning and Cybernetics (ICMLC), 2017.

1.5.3 Patent Application

1. **Fangyu Wu**. "Real time defect detection for sewer pipeline inspection", 2019 (Submitted).

2. **Fangyu Wu**. "Few-shot defect detection for sewer pipeline inspection", 2019 (Submitted).

Chapter 2

Preliminaries of Deep Learning and Representation Learning

2.1 Preliminaries of Deep Learning

Deep learning algorithms are subsets of machine learning algorithms, that aim to discover multiple levels of distributed representations. Recently, various deep learning algorithms have been proposed to solve traditional machine learning problems. This chapter aims to introduce the preliminaries of deep learning algorithms which are related to the research topic of the thesis, followed by the introduction and review of the representation Learning mechanism.

2.1.1 Overview of Deep Learning

Deep learning originated from the study of Artificial Neural Networks (ANNs), which are computation models inspired by biological neural networks in human brains and have been extensively studied since the 1980s. An ANN consists of a collection of connected artificial neurons which simulate the neurons in a biological brain. It can be roughly characterised by the weights between layers of neurons whose output is computed based on some non-linear transformation function. At each layer, neurons compute a weighted sum of the inputs from the previous layer, using $Wx + b$ where W is a weight vector and b is a bias, and pass the result through a non-linear activation function σ, e.g., tanh, sigmoid and rectified linear unit (ReLU) [176].

One of the major reasons that ANNs with multiple fully connected layers have not gained popularity in many real-world applications for decades is their computation complexity. The idea of deep learning, also inspired by biological processes, powered by high-performance computing hardware, has made very deep models computationally feasible for real-world applications. For example, in a convolutional network, the connectivity between neurons resembles the organization of neurons in the animal visual cortex [130]. Each cortical neuron only responds to stimuli within a limited region of a visual field (also known as the receptive field); in a recurrent network, weights are shared among layers which not only reduces the number of parameters to be learned but also generalises better for input sequences of different lengths [130]. Training deep neural networks is notoriously expensive and would not be practical without the employment of high-performance computing hardware, e.g., Graphics Processing Units (GPUs). The highly parallel structure ensures efficient processing of large numbers of data blocks in parallel, making it suitable for training deep neural networks that have thousands of neurons performing the same computation at each layer. In recent years, deep learning methods have achieved results superior to other state-of-the-art machine learning methods and even human experts in many application areas.

2.1.2 Logistic Regression

Logistic Regression is a classical learning algorithm and a fundamental part of the neural network model [175]. Logistic Regression introduces the Logistic function into the Linear Regression model.

The distribution function of the Logistic distribution defines as:

$$P(x;\mu,s) = \frac{1}{e^{(-(x-\mu)/s)}} \tag{2.1}$$

The Logistic Regression model implements the following conditional probabilistic distribution:

$$\begin{aligned} P(y=1|x) &= \frac{e^{(w\cdot x+b)}}{1+e^{(w\cdot x+b))}} \\ P(y=0|x) &= \frac{1}{1+e^{(w\cdot x+b))}} \end{aligned} \tag{2.2}$$

where x is the input, and y is the output. w is the weight vector, b is the bias and $w \cdot x$ is

the dot product of w and x.

Equation 2.2 can get the conditional probabilities of the output to be 1 and 0 given the input samples.

The odds of an event is the ratio of the probability of happening of this event to the probability of not happening. If the probability of an event happening is P, then the odds of this event is $\frac{p}{1-p}$, also, the log odds of the event is $logit(P) = log\frac{p}{1-p}$.

For the Logistic Regression model, the log odds of the event is hence $log\frac{P(y=1|x)}{1-P(y=1|x)} = w \cdot x$, which indicates that the log odds of the Logistic Regression is a linear function of the input x.

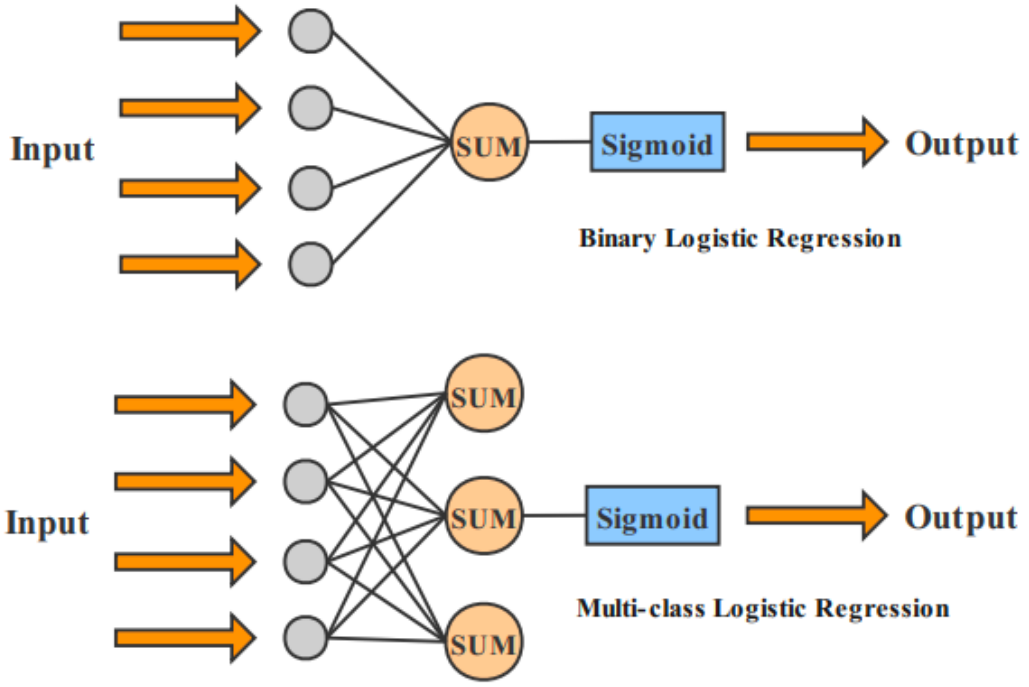


Figure 2.1: The neural network interpretation of Logistic Regression.

2.1.3 Basic Neural Network Model

From the viewpoint of a neural network, the Logistic Regression can be interpreted as one layer neural network. A Sigmoid (Logistic) activation function is the non-linear mapping function, which is shown in Figure 2.1.

If the multi-layer mapping is embedded in this system, it can form a neural network

learning model, or more specific, a feed-forward neural network [134] [237]. The feed-forward networks, or Multi-layer Perceptron (MLP), are vital in deep learning models. A feed-forward network aims to approximate some non-linear functions. The feed-forward networks are of extreme importance to machine learning practitioners.

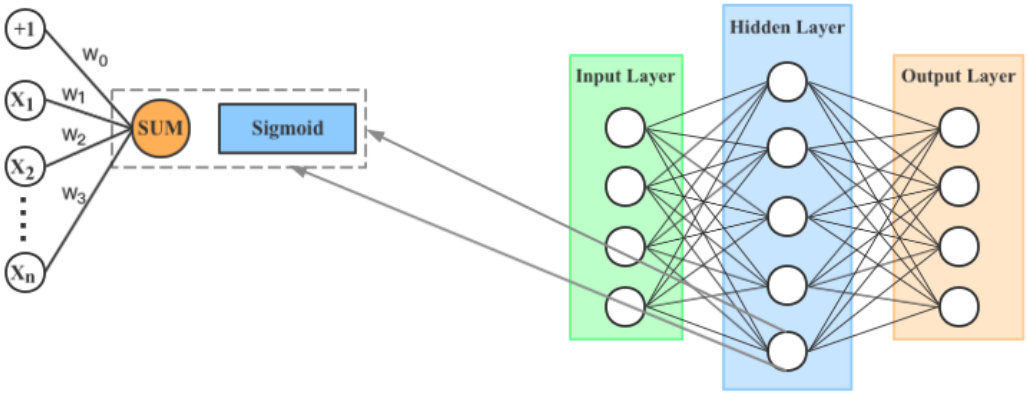


Figure 2.2: The structure of a feed-forward neural network.

They form the basis of many critical applications. For example, the convolutional network used for object recognition from images is a specific kind of feed-forward network. Feed-forward networks are a conceptual stepping stone on the path to recurrent networks, which power many natural language applications. Feed-forward neural networks are called networks because they are typically represented by composing together many different functions. The model is a directed acyclic graph that describes how the functions are composed together.

A general structure of the feed-forward network is shown in Figure 2.2. Each layer of the neural network performs matrix operation and nonlinear mapping. We can regard the neural network model as universal approximator [99], which approximates the measurable function to the required accuracy.

2.1.4 Convolutional Neural Network

CNN [131] is a particular type of neural network for processing data that has a known grid-like topology. Examples include natural language or speech data, which can be considered a 1D grid taking samples at regular intervals; and visual data, which can be considered a 3D grid of pixels. A typical CNN usually consists of a number of convolutional

layers, pooling layers, and fully connected layers as its hidden layers as illustrated in Figure 2.3. The convolutional layer aims to learn filters that represent features of the input (e.g., a particular shape) and generate a feature map. The pooling layer performs non-linear downsampling, which combines a cluster of neurons at one layer into a single neuron in the next based on non-linear functions such as max pooling and average pooling. Then, fully connected layers are added on the top of convolutional and pooling layers for final output.

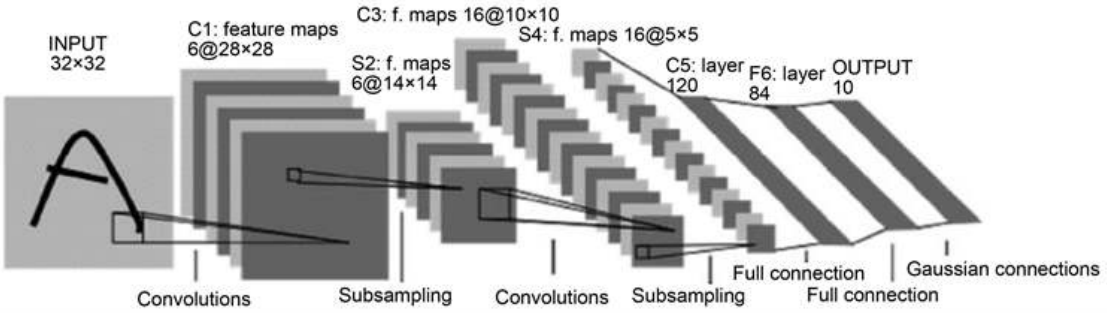


Figure 2.3: A typical CNN for hand-written digits recognition [133].

Convolution Operation

The convolution operation on a continuous function is defined in Equation 2.3. It can be interpreted as using a kernel function w(a) to calculate a weighted average of function $x(a)$ and $w(a)$.

$$s(t) = \int x(a)w(t-a)da \tag{2.3}$$

From the Latin 'convolvere', 'to convolve' means to roll together. For mathematical purposes, convolution is the integral measuring of how much two functions overlap as one passes over the other. Think of convolution as a way of mixing two functions by multiplying them. The convolutional operation can also defines as an asterisk, in Equation 2.4.

$$s(t) = (x \times w)(t) \tag{2.4}$$

In the case of CNN terminology, the function x and w represent the input and kernel, respectively.

In most cases, the data used are sampled not in every instance, but at a certain interval, in other words, these data are discredited. The time index t, consequently, then takes on

only integer values. The discrete convolution is defined in Equation 2.5.

$$s(t) = (x \times w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{2.5}$$

In machine learning applications, the input is usually a multi-dimensional array of data, and the kernel is usually a multi-dimensional array of parameters adopted by the learning algorithm. These multi-dimensional arrays will be referred as tensors.

In practice, the infinite summation can be implemented as a summation over a finite number of array elements. The tensors are considered zero everywhere except where the data is stored in the multi-dimensional arrays.

Also, convolutions can be used over more than one axis at a time. For instance, if a two-dimensional image I is taken as our input, a two-dimensional kernel K is utilized. Then, the two-dimensional convolution can be defined in Equation 2.6.

$$S(i,j) = (I \times K)(i,j) = \sum_{m} \sum_{n} I(m.n)K(i-m, j-n) \tag{2.6}$$

Since the convolution operation is commutative, alternatively, Equation 2.6 can also be written as:

$$S(i,j) = (I \times K)(i,j) = \sum_{m} \sum_{n} I(i-m, j-n)K(m,n) \tag{2.7}$$

A commonly used effective operation process of convolution in a CNN is described by Figure 2.4.

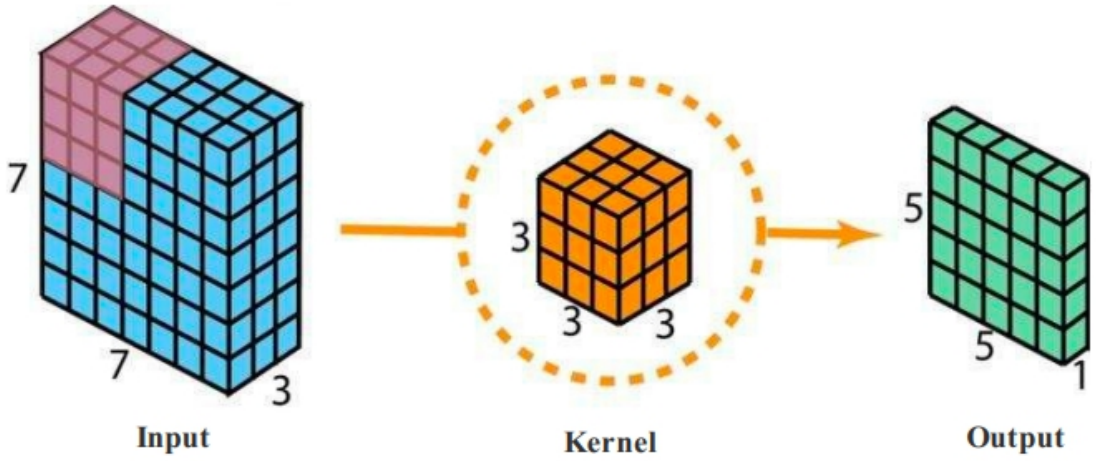


Figure 2.4: An illustration of the convolutional operation in CNN.

Pooling Operation

A typical layer of a CNN consists of three steps. In the first step, the layer performs several convolutions in parallel to produce a set of linear activations. In the second step, each linear activation is run through a non-linear activation function, such as the ReLU function [127]. This step is sometimes called the detector stage. In the third stage, a pooling function used to modify the output of the layer.

This section aims to give a general introduction to pooling. A pooling operation replaces the neural network output at a specific location with a summary statistic of the nearby outputs. The most commonly used pooling in CNN is max-pooling. Pooling helps to make the representation approximately invariant to small translations of the input. Invariance to the translation means that if the input is translated by a small amount, the values of most of the pooled outputs do not change, which increases the robustness of the neural recognition network. The application of pooling can be seen as adding an infinitely strong prior that the function that the layer learns must be invariant to small translations. When this assumption is correct, it can significantly improve the statistical efficiency of the network.

The max-pooling operation is shown in Figure 2.5. In each color-indicated grid, the max-pooling selects the maximum value to replace the data in the original grid, and form a new tensor as an output of the max-pooling layer.

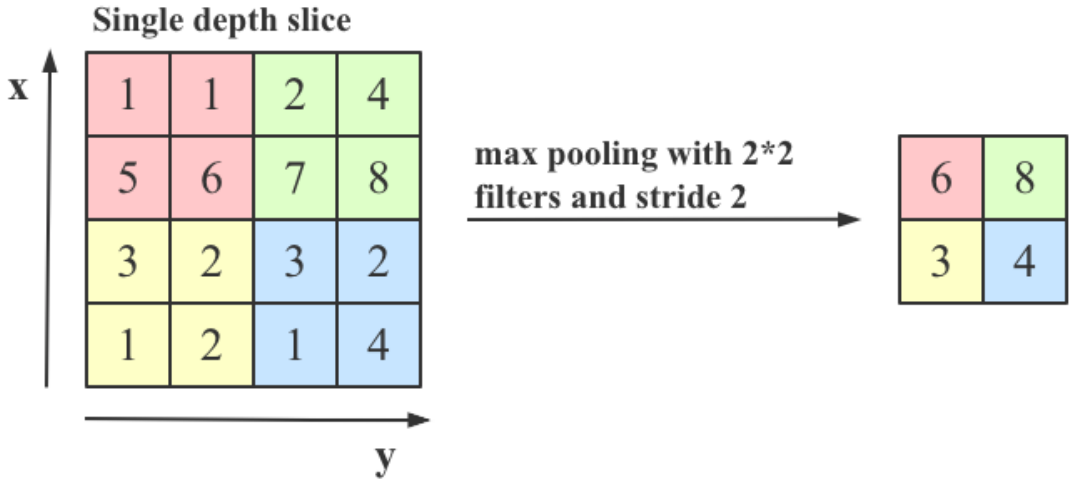


Figure 2.5: An illustration of the max pooling operation in a CNN.

Stochastic Pooling

A disadvantage of max-pooling is that it is sensitive to over-fitting on the training set, making it hard to generalize well during testing [279]. To solve this problem, Zeiler et al. [280] proposed a stochastic pooling approach which replaces the conventional deterministic pooling operations with a stochastic procedure, by randomly picking the activation within each pooling region according to a multinomial distribution. This stochastic nature is helpful in preventing the overfitting problem.

Spatial Pyramid Pooling (SPP) and Region-of-Interest Pooling (RoI pooling)

The CNN model requires a fixed-sized input image. This restriction may bring problems for images of arbitrary sizes, especially in the CNN-based object detection schemes. To eliminate this limitation, He et al. [88] replaced the last pooling layer with a SPP, for object recognition. The SPP can extract fixed-length features from arbitrary images (or region candidates), and can be applied in a CNN structure for arbitrary tasks, to improve the performance of the CNN model.

Subsequently, Girshick [77] proposed a simplified SPP layer for object recognition, called RoI pooling. This pooling layer is simpler and also enables the CNN model to handle arbitrary-sized input images. More importantly, the RoI Pooling layer enables the parameter sharing in the computation-intensive convolutional layers [77]. This research

is extremely important in object detection. Most subsequent research [200] [78] [87], for various tasks, employed the RoI pooling layer to deal with input images.

Spatial Transformers

Due to the typically small spatial support for max-pooling, the spatial invariance is only realised over a deep hierarchy of max-pooling and convolutions, and the intermediate features in a CNN model are not invariant to large transformations of the input data [32] [138]. To mitigate this issue, Jaderberg et al. [106] proposed an important model, the spatial transformer networks, which explicitly allows the spatial transformation of data in the network. The spatial transformers result in arbitrary CNN models that learn invariance to translation, scale, rotation and more generic warping. Also, the spatial transformer can be interpreted as an attention mechanism, but is more flexible and can be trained purely with back-propagation without reinforcement learning techniques.

Capsule Networks

Geoffrey Hinton pointed out many drawbacks of the max-pooling operation such as the side effect of 'coarse coding' [95]. To address this issue, Sabour et al. [208] proposed the 'Capsule Networks' in which a dynamic routing scheme is proposed between the capsules to replace the max-pooling. This type of 'routing-by-agreement' is more effective than the primitive form of routing in max-pooling, which allows neurons in one layer to ignore all but the most active feature detector in a local pool. This research is considered as a recent breakthrough in the deep learning area [265].

Activation Function

Activation in a neural network provides non-linear mappings that take the inputs and do some mathematical operations. Many such activation functions exist and are discussed as follows:

Sigmoid (Logistic)

This non-linearity takes an input a real-valued function and outputs value in the range of 0 and 1. It has been widely applied in neural networks for a long time. However, it suffers from saturating and vanishing gradient problem. The Equation 2.8 defines the Sigmoid

function.

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2.8}$$

Tanh

As shown in Equation 2.9, it is clear that Tanh can be considered as a scaled up version of a sigmoid, outputting values in the range of -1 and 1. The problem of saturating gradients also exists with this function. The Tanh function is widely applied in Recurrent Neural Networks (RNNs).

$$Tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = 2Sigmoid(2x) - 1 \tag{2.9}$$

ReLU

ReLU is a linear activation function which has a threshold at zero as shown in Equation 2.10. The convergence of gradient descent has been proved to be accelerated by applying ReLU [127].

$$ReLU(x) = max(0, x) \tag{2.10}$$

Training

In deep learning, each layer transforms the input data into a more abstract representation and the model learns to choose the best features that can improve performance. It can be used for both supervised learning and unsupervised learning tasks. In supervised learning, the objective is usually to learn a complex, non-linear function that maps the input to the output. This requires learning algorithms to generalize from the training data to unseen data in a "reasonable" way. An objective function is normally used to measure the error (or distance) between the predicted output and the desired output. A common objective function for classification tasks is Cross Entropy (CE), as shown in Equation 2.11, where x is an instance in the training set, $p(x)$ is the true probability distribution of the dependent variable while $q(x)$ is the predicted probability distribution.

$$CE = -\sum_{x} p(x) log q(x) \tag{2.11}$$

A common objective function for regression tasks is Mean Squared Error (MSE), as defined in Equation 2.12, where $|X|$ is the size of the training set, y is the ground truth, and $\hat{y}$ is the network output for regression problem.

$$MSE = \frac{1}{|X|}\sum_{x}(y(x) - \hat{y}(x))^2 \tag{2.12}$$

Some deep learning models can also be trained in unsupervised ways in which the output tries to recover the input and the objective is to minimise the reconstruction. For example, the work in [97] showed that deep belief networks can be trained in an unsupervised manner (pre-training), followed by a supervised fine-tuning, which resulted in superior performance.

After determining objective function, deep models are then trained to minimise the objective function with backpropagation and gradient descent techniques. Backpropagation algorithm distributes the error computed at the output layer backwards and the gradients of weights at different layers are calculated. At each layer, the gradient descent algorithm computes a gradient vector, and adjusts the weight vector along the opposite direction of the gradient vector to minimise the objective function. In practice, gradient-based learning algorithms, e.g., Stochastic Gradient Descent (SGD) [202], Adam [118] and RMSprop [246], have been widely adopted together with backpropagation for neural network training.

Regularisation

One of the common goals of machine learning algorithms is to generalise to unseen data. Overfitting happens when a model learns too well the details and the noise from training data while ignoring the general patterns, thus results in poor generalisation. Regularisation is a critical instrument in preventing overfitting. Some of the most common regularisation techniques for deep learning are: dataset augmentation, L1 and L2 regularisation, early stopping, and dropout, as suggested in [80].

Dataset augmentation

Overfitting can be a common problem when size of training data is too small compared with the number of model parameters to be learned. While an existing dataset may be limited, for some applications one may create synthetic data through a number of operations, e.g., rotate, scale and inject random unrelated images to enlarge a dataset. Besides creating synthetic data, multi-task learning and transfer learning techniques are also commonly used. In multi-task learning, related tasks using different datasets can be learned simultaneously. The work in [122], [230], [102] applied multi-task learning to

jointly learn people's movement and transportation mode patterns. Often when a training dataset may be too specific or small to learn a good model from scratch, transfer learning can be applied by pre-training a model with large available dataset and then fine-tuning the model with the data for specific tasks.

L1 and L2 regularisation

To prevent model from becoming too complex (e.g., large weights) and learning all the details and noise in the training dataset, a regularisation term can be added to the objective function. L1 regularisation (or Lasso regression) and L2 regularisation (or Ridge regression) are commonly used not only in deep learning but also many other machine learning algorithms. L1 regularisation is defined using absolute values of the weights and can perform some sort of feature selection, while L2 regularisation is defined using the squared values of the weights to penalise large model parameters.

Early stopping

In an ideal situation, as a model sees more data both training and test errors should constantly decrease. However, after certain number of epochs, the model may start to overfit and learn noise in the training set. In this case, the training error keeps going down while the test error starts to increase. Early stopping is used here to find the right moment to stop training to minimise the test error.

Dropout

It refers to a strategy that randomly drops out some units (hidden and visible) in a deep neural network to make nodes become more insensitive to the weights of the other nodes. It provides a way of approximately combining many different neural network architectures efficiently [231]. Subsequently, Warde-Farley [257] analysed the feasibility of the drop-outs and pointed out that drop-out is an effective ensemble learning method.

Pre-training and Fine-tuning

One of the purposes of pre-training for deep learning practitioners is preventing overfitting. It is associated with data augmentation and transfer-learning. Pre-training means initialising the CNN model with a set of pre-trained parameters rather than randomly-initialised

ones. Also, the deep neural networks are highly non-linear function. The backpropagation algorithm might lead the neural networks to local minima. Pre-training can provide a good start point for the initialisation of the parameters of the deep neural networks. It is a very popular practice in deep learning area, due to the advantages that it can accelerate the learning process and improve the generalisation capability. Erhan et al. [56] conducted an extensive research on why the pre-training steps help in raising the system performance. Deep learning researchers employ well-known CNN architecture pre-trained on ImageNet [42] dataset and fine-tune the model for the task at hand.

Common CNN Architectures

In this section, some of the commonly used CNN architectures in computer vision are presented.

LeNet

This CNN architecture was one of the pioneering research in CNNs by LeCun et al. [132]. In this research, the hand-written digits were recognised by a CNN. It finds application in reading zip codes, digits, and so on. The lack of high-level computing machines at that time restricted the large-scale application of CNNs.

AlexNet

This architecture developed by Alex Krizhevsky, Ilya Sutskever and Geoff Hinton [127] is credited as the first work in CNNs to popularise it in the field of computer vision. The network was similar to LeNet, but instead of alternating convolution layers and pooling layers, AlexNet had all the convolutional layers stacked together. Also, they proved the feasibility of ReLU function in training large-scale CNN. Moreover, compared to LeNet, this network is much bigger and deeper. AlexNet was able to win the ImageNet Large Scale Visual Recognition Challenge-2012 (ILSVRC-2012) [42]) competitions achieving top-1 and top-5 error rates on test dataset.

GoogleNet

This CNN architecture from Szegedy et al. [239] from Google won the ILSVR- C 2014 competition. They proposed a new architecture called Inception (v1) that gives more

utilisation of the computing resources in the network. GoogleNet is a particular incarnation that has twenty-two layers of Inception module but with less parameter compared to AlexNet. Later, many improvements had been made on Inception-v1, with the principle being the introduction of batch normalisation which led to Inception-v2 by Ioffe et al. [104]. More refinements were added to this version, and the architecture was referred to as Inception-v3 [240]. Also, the Inception network is continuing to be developed [238].

VGG-Net

A famous structure, developed by Karen Simonyan and Andrew Zisserman [224], called VGG-Net, has been adopted by many types of research for various computer vision tasks. The authors of [224] have done a through analysis of the depth factor in a CNN, keeping all other parameters fixed. This trial could have led to a vast number of parameters in the network, but it was efficiently controlled by using tiny 3x3 convolution filters in all layers. The VGG-Net was the runner-up in ILSVRC 2014 contest.

Residual-Net

A severe problem, preventing the CNN to be deeper, is the vanishing gradient problem [89]. He et al. developed a CNN framework by utilising a residual connection between layers, which can reduce the vanishing gradient effect on the training of a very deep network [89]. A primary drawback of this framework is that it is much expensive to evaluate due to the significant number of parameters. However, the number of parameters can be reduced to an extent by removing the first Fully-Connected layer (most of the parameters are in this layer in a CNN), without any effect on the final performance.

2.1.5 Recurrent Neural Networks (RNNs)

A RNN [55] contains links among neurons, and after unfolding it forms a directed graph along a sequence. This allows RNN to process data that can be modelled as temporal sequences of variable lengths, $x = (x_1, ..., x_T)$. At each time step t, the hidden state h_t of the RNN is updated using $h_t = f(h_{t-1}, x_t)$, where f is a non-linear function, which can be as simple as an sigmoid function and as complex as a long short-term memory unit. The Figure refrnn shows the unfolding of the computational graph of a RNN. A computational graph is a way to formalize the structure of a set of computations, such as those involved in taking inputs and parameters to outputs and final loss function.

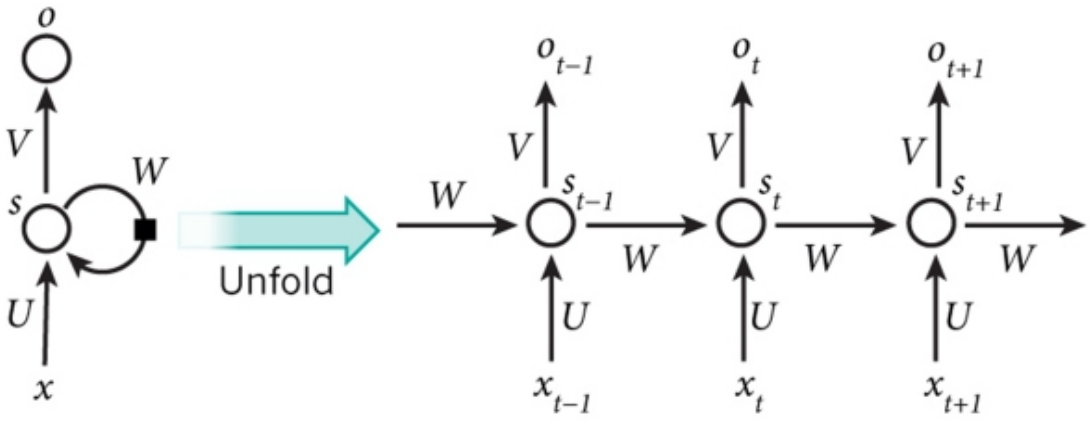


Figure 2.6: The unfolding of the computational graph of a RNN [130].

RNNs use the internal states to capture dependency among input data in a sequence, which makes them suitable to tasks such as natural language processing, speech recognition and smart city applications in which data demonstrates strong temporal correlations. As vanilla RNNs suffers from various limitations, they have not been used in any real-world applications. In what follows, we present two important RNN units: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), and two widely used architectures: Bi-directional RNN and RNN encoder-decoder.

Long Short-Term Memory

Vanilla RNNs have difficulties in modelling long sequences as the gradients in parameter updates tend to either explode or vanish during backpropagation. LSTM has been proposed to solve the vanishing and exploding gradient problem by introducing the idea of Constant Error Carousels (CEC) [98]. In the original LSTM, the activation function of the unit is replaced by the identity function in the CEC to enforce constant error flow. Later, various extended models have been proposed, e.g., by adding forget gate and peephole connection [83] in order to address the limitations of the original LSTM [98].

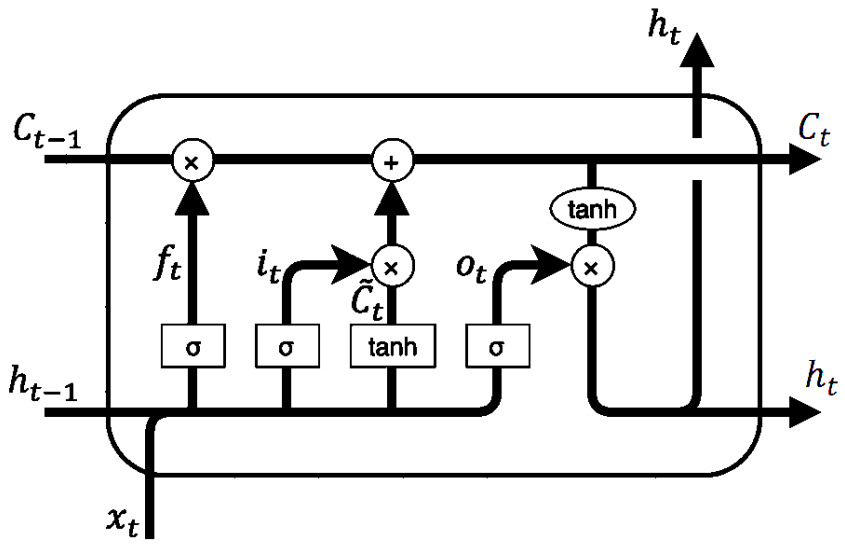


Figure 2.7: Long Short-Term Memory [180].

We briefly introduce the LSTM unit, following the notations used in [83], [180]. As shown in Figure 2.7, it contains a cell C, an input gate i, an output gate o and a forget gate f. The subscript t represents a particular time step. A standard LSTM updates the hidden state h by iterating the following steps shown in Equation 2.13, where all the W and U matrices are the learnable weights and the b vector represents the bias term (We ignore the subscripts for simplicity).

$$\begin{aligned} f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\ i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ C_t &= f_t * C_{t-1} + i_t * tanh(W_C x_t + U_C h_{t-1} + b_C) \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{2.13}$$

Gated Recurrent Unit

GRU, as shown in Figure 2.8, is another RNN unit introduced by Cho *et al.* [29]. It contains two gates: update gate z and reset gate r, where the update gate helps the model determine how much of the past information to remember and the reset gate is used to

decide how much of the past information to forget. The hidden state h is then updated iteratively using the following procedure shown in Equation 2.14, where all the W and U matrices are the learnable weights and the b vector represents the bias terms.

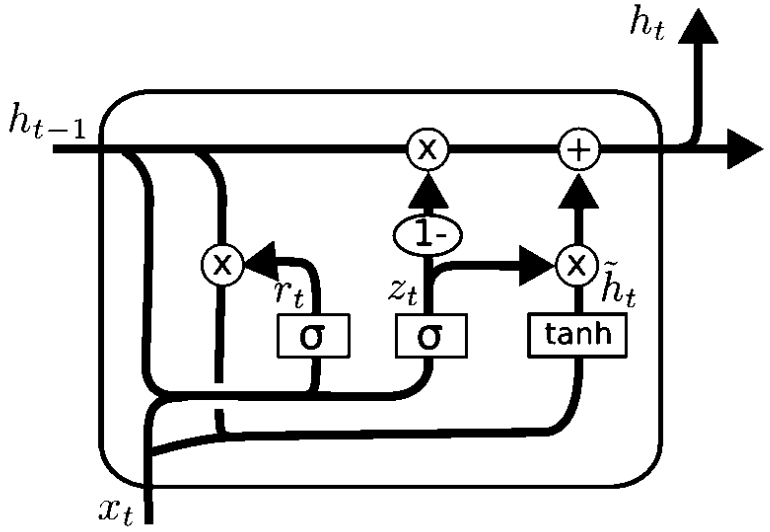


Figure 2.8: Gated Recurrent Unit (GRU) [180].

$$
\begin{aligned}
z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z) \\
r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) \\
h_t &= (1 - z_t) * h_{t-1} + z_t * \sigma_h(W_h x_t + \\
&\qquad U_h(r_t * h_{t-1}) + b_h)
\end{aligned}
\tag{2.14}
$$

Bi-directional RNN

A standard RNN can learn representations from data from previous time steps; however, representations from future time steps may help better understand the context and eliminate ambiguity. For example, in handwriting recognition, the performance can be significantly enhanced if the letters located before and after the current letter were known. Bi-directional RNNs [216] was proposed by stacking two LSTM RNNs, one processing the sequence from

left to right, the other one from the opposite direction, and finally concatenating the output of the two RNNs. With this structure, the output layer can integrate information from both past and future states.

Bi-directional RNNs have been commonly used in natural language processing [235] and speech recognition [83]. For sensor data processing, recent studies applied Bi-directional LSTM to recognise human activities [174], [52]. In this task, the input is a discrete sequence of equally spaced samples $\{x_1, x_2, ..., x_t\}$, where each data point x_t is a vector of individual samples observed by sensors at time t. The samples are segmented into windows of maximum time T and fed into the network with one direction from time 1 to T and another direction from time T to 1. The network can output the probabilities of different activity labels after a softmax layer. Both work [174], [52] reported the state-of-the-art performance compared to conventional techniques.

RNN Encoder-Decoder

In some applications, the input and output sequences have different lengths, e.g., in machine translation, the input sentence and the desired target sentence usually have different lengths. An important and effective technique for such application is the RNN based encoder-decoder architecture [29]. It contains two RNNs, one learns to encode an input sequence of certain length into a context vector representation (the encoder) and the other learns to decode the context vector representation back into an output sequence of different length (the decoder).

The architecture allows some smart city applications to produce a sequence of predictions for time series data. For example, in air quality and water quality prediction task, the work in [145] designed an encoder to find a suitable representation of the past observation data and used a decoder to generate a sequence of output, i.e., the air and water quality measurement in the next few minutes or even hours. In addition, the work applied spatial attention in the input layer and embed some external factors, e.g., time, weather and point of interests, in the context vector to further improve prediction results.

2.1.6 Generative Adversarial Networks (GANs)

The Theory of the GANs

GANs are an example of the generative model. GANs was first proposed in [81] in 2014. It is proposed initially to generate realistic images given a random signal. The fundamental

idea of GANs is to set up a game between two players. One of them is the generator, which creates samples that are intended to come from the same distribution as the training data. The other player is the discriminator which tries to differentiate the generated samples from real samples. The discriminator learns using traditional supervised learning algorithms, discriminating inputs into two categories (whether from generated or real samples). The generator is trained to deceive the discriminator. This is an adversarial game in which the generator tries to generate samples more like the real ones while the discriminator is trained to better discriminate between the generated and real samples. The generator must learn to make samples that are indistinguishable from the genuine samples to make the game successful, and hence, the generator network can learn to generate samples that are drawn from the same distribution as the training data.

In the original GANs, the adversarial framework applied when the models are both MLP [81]. In fact, CNNs can also be used in this framework [194], also RNNs [274]. To learn the generator's distribution p_g over data x, the GANs define a prior on input noise variables $p_z(Z)$, then represent a mapping to data space as $G(z;\theta_g)$, where G is the generator which is represented by a differentiable function such as neural networks. The discriminator is another neural network, $D(x;\theta_d)$ which outputs a single value, representing whether the samples are generated or real. Then the discriminator D is trained to maximise the probability that the correct labels are assigned to the training samples and generated samples. The generator, G, is trained simultaneously to minimize $log(1 - D(G(z)))$. In summary, the D and the G play a two-player minimax game as described in Equation 2.15. The structure of a typical GANs model is shown in Figure 2.9.

$$minmaxV(D,G) = E_{xP_{data}(x)}[logD(x)] + E_z pz(z)[log(1 - D(G(z)))] \tag{2.15}$$

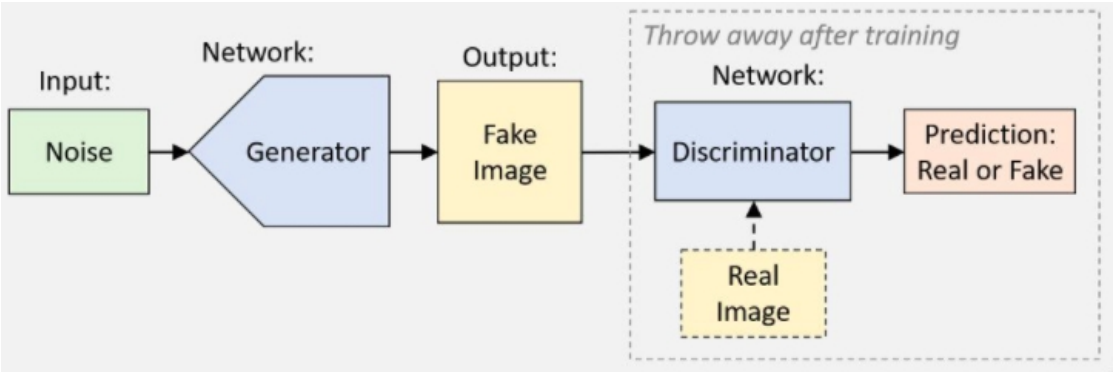


Figure 2.9: The structure of a typical GANs model [81].

The GANs framework is not restricted in image generation, in fact, it can be applied in many tasks. For instance, language generation is an essential task in natural language processing and also has significant practical value. Yu et al. [274] propose the SeqGAN for language generation. As explained in Equation 2.15, the generator and discriminator are trained simultaneously, which means that the gradient can be back propagated from the discriminator to the generator, since image generation is a continuous process. However, language generation is a discontinuous, often token by token. To directly apply GANs on the task of language generation is infeasible. To tackle this difficulty, Yu et al. [274] propose to use reinforcement technique in which the probability of the generated samples to be real is considered as a reward value for the generator. Hence, with the aid of reinforcement learning algorithms, the SeqGAN can be trained, with improving results over conventional supervised learning.

2.2 Representation Learning

2.2.1 Overview of Representation Learning

Representation Learning (RL) or feature learning is the task of finding a transformation of raw data in a way to improve the performance of machine learning tasks such as regression and classification. In fact, RL is essential for approaching real artificial intelligence. Moreover, RL is commonly considered as a potential candidate solution for numerous complex problems of data science. Furthermore, RL methods attempt to make some important concepts of real-world intelligence possible. As mentioned by Bengio and LeCun

[12], the most important reason that makes some methods of RL successful is their ability to utilize some general priors related to real-world intelligence. Some of these priors include smoothness, multiple explanatory factors, the sparsity of features, transfer learning, independence of features, natural clustering and distributed representation, semi-supervised learning, and hierarchical organization of features [14]. A typical RL method will be more powerful and valuable if it covers a larger set of the above mentioned general priors.

The performance of machine learning methods is heavily dependent on the choice of data representation (or features) on which they are applied. We hypothesize that this is because different representations can entangle and hide more or less the different explanatory factors of variation behind the data. Although specific domain knowledge can be used to help design representations, learning with generic priors can also be used, and the quest for AI is motivating the design of more powerful representation-learning algorithms implementing such priors.

As there are a variety of RL methods, different categorization of them is manageable. One possibility is to categorize RL methods into four main approaches, including sub-space based RL approaches which look for representations in the sub-spaces of the original feature space, manifold based RL approaches that represent raw data based on the embedded manifold hidden in the original space, shallow RL approaches, and deep RL approaches.

It is possible to consider RL methods in term of using or not using supervisory information for generating representations. Majority of RL methods such as principal component analysis (PCA), independent component analysis (ICA), restricted Boltzmann machines (RBM) perform unsupervised RL thus, they do not incorporate any class label or other supervisory information in the process of learning representations. In contrast to unsupervised RL methods, supervised RL methods like linear discriminant analysis (LDA) family, incorporate supervisory information in the process of learning representations. However, there are some RL methods that are naturally unsupervised but, they use additional information in the process of learning representations; hence, they are called soft supervised RL methods. Semi-supervised RL methods utilize both labeled and unlabeled data for generating representations. Worth to mention that the main focus of RL methods is on the unsupervised and semi- supervised methods of feature generation.

It is supposed that RL is the task of looking for a transformation (mapping) function $f : X^D \rightarrow Y^d$, which transforms (maps) data from the original feature space X, with the dimension D, to the representation space Y, with the dimension d. Dimensionality of the representation space is usually much smaller than the dimensionality of original

feature space. As an exception, in order to force generated representations to have a specific property, their dimension may be much greater than the dimension of data in the original space. Moreover, in some methods of RL like CNNs for classification, the output (results of mapping) is an encoding which is consistent with the final output of the pattern recognition system. In other words, the final output of the transformation is the predicted value of such tasks. Some RL methods have intermediate transformations and consequently representations organized into multiple layers. Such representation methods with multiple hierarchical layers are elaborated in the later sections.

2.2.2 Sub-space Based Representation Learning Approaches

Sub-space based approaches as almost early methods of RL attempt to look for a sub- space in the original feature space that better represents the original data. This representation is achieved by projecting data of the original feature space into new sub-space by applying the learned transformation function; the generated representation has some properties corresponding to the way base functions of the transformation are formed. In sub-space based RL methods, new features are commonly generated by a linear combination of original features through base functions; the base functions of transformation are learned by analyzing data in the original feature space. During the learning process of the base functions, independence, orthogonality, and sparsity as potential properties may be obtained. In the sections ahead, the most popular sub-space based RL methods, including PCA family, metric multi-dimensional scaling (MDS), ICA family, and LDA family, are considered.

Principal Component Analysis Family

PCA as a global method is one of the oldest techniques of unsupervised data representation which focus on the orthogonality of generated features [19]. The main purpose of PCA is to generate a low dimensional representation of the original observations and preserve maximum variance of the original data as well. The base functions of trans- formation are actually principal components hidden in the original data. A solution for finding transformation matrix is to use a portion of eigenvectors of the covariance matrix of the original data. The number of selected eigenvectors determines the dimension of the new representation. The eigenvalue corresponding to each eigenvector measures its importance in term of the amount of held variance.

PCA is suffering from the fact that the principal components are created by an explicit linear combination of all of the original observations. This phenomenon does not allow to interpret each principal component independently. In order not to use all of the original variables is to utilize Sparse PCA (SPCA) which reduce the dimensionality of the data by adding sparsity constraint to the original variables [301].

As it is the case in many real-world applications, if the generation mechanism of data is non-linear, the original PCA fails to recover true intrinsic dimensionality of the data. This is considered a shortcoming of PCA which is relieved by its kernelized version known as Kernel PCA (KPCA) [214].

It is also possible to derive PCA within a density estimation framework based on a probability density model of the observed data. In this case, the Gaussian latent- variable model is utilized to derive probabilistic formulation of PCA. Latent-variable formulation of obtaining principal axes leads naturally to an iterative and computationally efficient expectation-maximization solution for applying PCA commonly known as Probabilistic PCA [293].

Metric multi-dimensional Scaling

Metric multi-dimensional scaling is a linear technique for generating representations. In contrast to PCA which project data into a sub-space that preserves maximum variance, MDS project data into a sub-space which preserve pairwise squared distance. In other words, MDS attempts to preserve the dot product of samples in the new representation space [2]. The idea of distance preservation used in MDS has been used in one way or another in some manifold learning. As Eigen decomposition of the Gram matrix which holds pairwise dot product of samples is required for MDS, kernel PCA can be considered as a kernelized version of MDS, where the inner product in the input space is replaced by kernel operation in the Gram matrix.

Independent Component Analysis Family

ICA is another popular technique of sub-space based RL which is very similar to PCA. In contrast to PCA which uses variance as second-order statistical information, ICA uses higher order statistics for generating representations. Using higher order statistics force generated features to be mutually independent [34]. In a topological variation of ICA, independence assumption of generated features is removed and a degree of dependence

based on the distance between generated features is assigned. Mentioned distances lead to generate a topological map which is used by some applications of computer vision [103]. Kernel ICA is another variation of original ICA which uses calculated correlation in the reproducible kernel Hilbert space for generating non-linear representations [9].

Linear Discriminant Analysis Family

LDA is a global and supervised method of RL. In this method, the transformation matrix is obtained in a way to generate features that hold maximum variance an also bring maximum class separability by utilizing the within-class and the between-class amount of variances exist in the data. In other words, the transformation matrix is computed in a way that the amount of between-class variance relative to the amount of within class variance is maximized. Generating features that satisfy class separability property is desirable for many applications [64]. An incremental version of LDA is also proposed for those applications which demand generated representation space be updated at the arrival of new data sample [74]. To conclude sub-space based RL methods, many methods try to find sub-space in one way or another. This sub-space has some properties that are transferred to the generated features. The advantage of sub-space methods of representation generation is computational efficiency thanks to eigen decomposition technique. As sub-space methods are linear in nature, they cannot be successful when the original data are generated non-linearly. In the case of non-linearity, for better representation, other RL methods such as manifold family are potential candidates to be considered in the next section.

2.2.3 Manifold Based Representation Learning Approaches

Among the family of RL approaches, manifold based methods have attracted attention due to their non-linear nature, geometrical intuition, and computational feasibility. A strong assumption in most manifold learning methods is that the data appears in the original high dimensional feature space approximately belongs to a manifold with an intrinsic dimension less than the dimension of original space. In other words, the manifold is embedded in the original high dimensional feature space. The goal of manifold based RL methods is to find this low dimensional embedding and consequently generating a new representation of original observations based on the founded embedding. In contrast to sub-space based RL approaches which usually perform dimensionality reduction and consequently linear RL, manifold based approaches reduce the dimension in a non-linear

fashion by attempting to uncover intrinsic low-dimensional geometric structures hidden in the original high dimensional observation space.

Manifold based RL methods are categorized into three main groups of local, global, and hybrid; each method attempts to preserve different geometrical properties of the underlying manifold while attempting to reduce the dimension of original data.

Local Methods of Manifold Learning

Local manifold learning methods attempt to capture local interactions of samples in the original feature space and transfer captured interactions to the generated new low-dimensional representation space. The strategies followed by local methods of manifold learning lead to map nearby points of the original feature space to nearby points in the newly generated low-dimensional representation space. Computational efficiency and representation capacity are two characteristics of local methods. Computations of local methods are efficient because the matrix operands that exists in local methods are usually sparse.

Laplacian eigenmaps [11], local linear embedding (LLE) [203], and Hessian eigenmaps [50] are representative methods of local manifold learning family. Laplacian eigenmaps captures local interactions of data by utilizing Laplacian of the original data graph. Sensitivity to noise and outliers are considered as a shortcoming of Laplacian eigenmaps. Representations generated by LLE are invariant under rotation, translation, and scaling as geometrical transformations. Hessian eigenmaps is the only method of manifold learning capable of dealing with non-convex data. As all the methods based on Hessian operator needs to calculate second derivatives, they are sensitive to noises, especially in high dimensional data.

Global Methods of Manifold Learning

The fact that representations generated by global methods of manifold learning cause the nearby points to remain nearby and also faraway points remain faraway, tends these methods to give more faithful representation than local methods.

Isometric feature mapping or shortly ISOMAP is the most popular global method of manifold learning. ISOMAP uses the geodesic distance between all pairs of the data points to uncover the true structure of the manifold. Using geodesic distance instead of Euclidean distance leads faraway points in the original space to remains faraway in the representation

space. The reason for this desirable property is that some points that are close in term of Euclidean distance may be far in term of geodesic distance. In fact, geodesic distance allows learning the global structure of the data. ISOMAP is also considered as a variant of the MDS algorithm in which the Euclidean distances are changed to the Geodesic distances along the manifold [245].

Experimental result demonstrates ISOMAP cannot scale well for large datasets as it demands huge amounts of memory for storing distance matrices. In order to increase its scalability, landmark ISOMAP (L-ISOMAP) has been proposed by using a subset of data points known as landmark points [222].

Hybrid Methods of Manifold Learning

As mentioned previously, both local and global methods of manifold learning have their own advantages and disadvantages in terms of representation capability and computational efficiency. Hybrid methods of manifold learning usually attempt to globally align local manifolds and gain benefits of computational efficiency of local methods and quality representation generation of global methods. In other words, hybrid methods generate representations approximately as good as global methods by an efficient cost of local methods. Some of the well-known hybrid methods of manifold learning are conformal ISOMAP [222], manifold charting [18], and diffusion maps [33].

To conclude, manifold based methods of RL exist in different categories with different properties. Early local methods are sensitive to noises and outliers. Moreover, proper parameter tuning is mandatory for some methods. Experiments demonstrate global methods of manifold learning gives a better representation than local methods. However, this excellence comes with a higher cost of computation. As the computational cost of local methods is more reasonable, some hybrid methods attempt to follow the path of local methods for obtaining representations with the capability as close as global methods. Some manifold learning methods have a close relationship to sub- space based methods such as MDS and Kernel PCA. Despite many progress in manifold learning methods, the problem of manifold learning from noiseless and sufficiently dense data still remains a difficult challenge. Although manifold learning methods generate representations better than sub-space based approaches, still we need better methods for generating representations that meet the requirements of real-world intelligence.

2.2.4 Shallow Representation Learning Approaches

The focus of this section is the consideration of shallow RL approaches in term of representation capability and computational efficiency. As a matter of fact, sub-space and manifold based RL approaches are under the umbrella of shallow architectures. Also, some machine learning techniques such as multilayer perceptron with less than five layers and local kernel machines are considered as shallow architecture methods; these techniques generate a limited representation of input data in their mechanism prior to producing any prediction output.

In order to represent any function or learn behavior and underlying structure of any data by using shallow architectures, an exponential number of computational elements with respect to the input dimension is required. As a result, shallow methods are not compact enough. Compactness means fewer computational elements and consequently fewer free parameter tuning. Accordingly, non-compact nature of shallow methods of RL lead these methods to have poor generalization property.

As the majority of shallow architecture RL methods are indeed local estimators, they exhibit poor generalization while learning highly varying functions. The reason for lack of generalization, in this case, is that local estimators partition input space into regions whose number relates to the number of variations in the target function. Each partition needs its own parameters for learning the shape of that region. As a result, much more training examples are needed to support the training of variations in the target function. Kernel machines and many unsupervised RL methods such as ISO-MAP, LLE, and Kernel PCA are good examples of local estimators which are considered as shallow architecture RL techniques. In order to tackle limitations of kernel machines as local estimators, some techniques are needed to learn better feature space and consequently learning highly varying target functions in an efficient manner. Worth to mention, if the variations of target function are independent, no learning algorithm will perform better than local estimators [13]. Restricted Boltzmann machines (RBM) and autoencoders as shallow architecture methods of RL are introduced in the sections ahead.

Restricted Boltzmann Machines

Restricted Boltzmann machines (RBMs) are actually energy-based probabilistic graphical models which attempt to learn the distribution of input data. As Figure 2.10 depicts, a typical RBM has two layers of visible and hidden nodes. The visible layer nodes are

connected to the hidden layer nodes via weight matrix W. There are no visible-visible and hidden-hidden connections hence, these types of Boltzmann machines are so-called restricted. RBMs are able to compactly represent any distribution in case of providing enough hidden nodes. The scaler energy associated to each configuration of the nodes in a typical RBM is defined by Equation 2.16 as energy function and the probability distribution via mentioned energy function is described by Equations 2.17, 2.18, and 2.19. Here, b and c refer to the biases of visible and hidden nodes respectively [65].

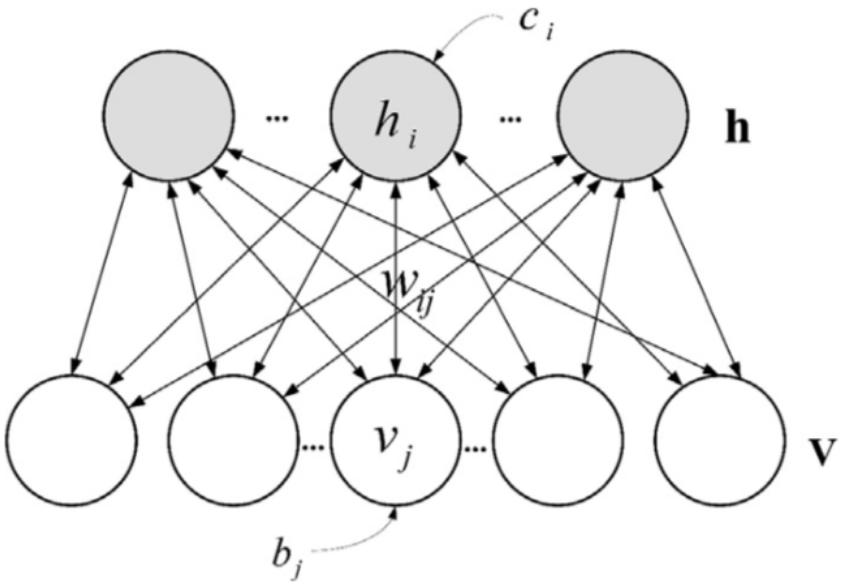


Figure 2.10: The architecture of a typical restricted Boltzmann machine [283].

$$E(v.h) = -bv - ch - hWv \tag{2.16}$$

$$p(x) = \frac{e^{-}F(x)}{Z} \tag{2.17}$$

$$Z = \sum_{x} e^{-F(x)} \tag{2.18}$$

$$F(x) = -log \sum_{h} e^{-E(x,h)} \tag{2.19}$$

In order to learn the desired configuration, the energy function should be modified through a stochastic gradient descend procedure on the empirical negative log-likelihood of the

data-set whose distribution needs to be learned. Equations 2.20 and 2.21 defines required log-likelihood and loss functions respectively. In these equations, h and D refers to the model parameters and training data respectively. The parameter set (h) which needs to be optimized include, weight matrix W, biases of visible nodes b, and biases of hidden nodes c. Gradient of negative log likelihood as described by Equation 2.22 has two terms refereed as positive and negative phases. Positive phase deals with the probability of the training data while, negative phase deals with probability of samples generated by the model itself. The negative phase allows to check what have been learned by the model up to current iteration. In order to make computation of the gradient tractable, the expectation of all possible configuration of visible nodes v under model distribution P is estimated via a fixed number of model samples known as negative particles. The negative particles N are sampled from P by running a Markov chain with Gibbs sampling as its transition operator. In order to efficiently optimize model parameters, contrastive divergence (CD) is utilized. CD-k initialize the Markov chain using one of the training examples and limits the transition just to k step. Experimental results demonstrate the value 1 for k is appropriate for learning data distribution [93]. For better performance, construction and training of RBMs need some proper settings, including the number of hidden units, the learning rate, the momentum, the initial values of weights, the weight-cost, and the size of mini batches of the gradient descent. To clarify the effect of these meta-parameters on each other, by having more hidden nodes, the representation capacity of RBMs increases with the cost of increasing training time. In addition, types of units to be used and decision on whether to update the states of each node stochastically or deterministically are important [94].

$$L(\theta.D) = \frac{1}{N} \sum_{x^{(i)} \in D} logp(x^{(i)}) \tag{2.20}$$

$$l(\theta.D) = -L(\theta.D) \tag{2.21}$$

$$-\frac{\delta logp(x)}{\delta \theta} \approx -\frac{\delta F(x)}{\delta \theta} - \frac{1}{|N|} \sum_{x \in N} \frac{\delta F(x)}{\delta \theta} \tag{2.22}$$

As the training of a typical RBM is converged, it is ready to generate a new representation in the hidden layer for any data presented to its visible layer. RBMs are also considered as multi-clustering methods which are a kind of distributed representation. Distributed representation as a requirement for real-world intelligence is the capability which leads each hidden node concerns one specific aspect of the data which have been

presented to its visible nodes. Distributed representation of RBMs enable generalization to a new combination of values of learned features beyond those have been seen during its training. RBMs are used in a variety of applications including analysis of complex computer tomography images [251].

Autoencoders

Autoencoders are actually unsupervised neural networks trained via back-propagation algorithm with the setting that target values are the input values [204]. A typical autoencoder is composed of an encoding unit that generates representations, decoding unit that reconstructs input from representation, and one hidden or representation layer which desired to captures main factors of variations hidden in the data. Early autoencoders attempt to learn a function which is an approximation to the identity function.

By applying some constraints on the autoencoder network and specifically its objective function, more interesting structures hidden in the data will be discovered. These constraints usually appear in different forms of regularization. Simplest regularization technique is the weight decay which forces the weights to be as small as possible. Going from linear hidden layer to non-linear one leads the autoencoders to capture multi-modal aspects of the input distribution [109]. Sparsity is a solution for preventing autoencoders from learning the identity function. In this setting, which is known as over-complete setting, the size of hidden layer is greater than the size of the input layer and many of the hidden nodes get zero or near zero values [196]. In order to force the hidden layer to learn more robust and generalized representation, denoising autoencoders that lead the network to learn representation from a corrupted or noisy version of the data are proposed. Representations generated from noisy data are more robust than their previous counterparts [253]. Variational autoencoders (VAEs) as a generative variation of autoencoder networks, attempt to generate new samples to exploring variations hidden in the data. In contrast to other methods of sample generation which are random, VAEs generate samples in the direction of existing data to fill the gaps in the latent space thanks to their continuous latent space [120].

2.2.5 Deep Representation Learning Approaches

Deep architectures are among potential solutions for tackling previously mentioned limitations of shallow RL approaches. As deep architectures of RL cover more general priors of

real-world intelligence, they are considered as the most promising paradigms for solving complex real-world problems of artificial intelligence up to know. In other words, multiple layers of representation in deep architectures facilitate the reorganization of feature space that causes machine learning methods to learn highly varying target functions. Deep RL methods are necessary for AI-level applications which need to learn complicated functions that represent high-level abstractions. Deep representations are obtained by utilizing deep architectures that are the composition of multiple stacked layers. These multiple processing layers attempt to automatically discover abstractions from lowest level observations to the highest level concepts. Abstractions in different layers allow building concept hierarchy as a necessity for real-world intelligence. In other words, higher layers attempt to amplify important aspects of raw data and suppress irrelevant variations [130].

Neural networks are considered as the most promising path for approaching deep RL. A typical deep neural network (DNN) is actually a network with multiple stacking layers of simple non-linear processing units. Because of the large number of layers and units per layer, training of such large networks demands a huge number of training data and computational power for better generalization.

Training of a typical DNN is commonly based on error gradient back propagation which relies on multiple passes over training data. As the number of parameters in DNNs is huge, too many training data and consequently long iterations are needed for proper optimization. In order to decrease the training time of DNNs as a large scale machine learning problem, stochastic gradient descend (SGD) has been proposed [17].

Training of deep neural networks is a difficult optimization problem because of vast parameter space with too many local optima and plateau which their computed gradient is zero. In order to train DNNs, layer-vise unsupervised pre-training, convolution, auto-associators, dropout, and other techniques are utilized. These techniques cause construction of special types of deep neural networks including, Deep Belief Networks, Convolutional Neural Networks, Deep Auto-encoding networks, and Dropout Networks respectively.

Chapter 3

Deep Multiple Classifier Fusion for Traffic Scene Recognition

Recent success in supervised learning can arguably be attributed to the paradigm shift from engineering representations to learning representations. Especially in the supervised setting, effective representations can be acquired directly from the labels. The goal of supervised learning is to learn a model $p(y|x)$. Here x could be real-valued high-dimensional vectors representing the raw contents of an image, an audio waveform, or sensory data in general, and y could be a low-dimensional vector representing a label in the case of classification. In this chapter, we have introduced a novel deep learning with multiple classifier fusion approach that emphasizes local-aware representation learning. With the local deep-learned features, the network can further import local region information from input, providing discriminative feature learning result for the representation learning component.

3.1 Introduction

Recognizing the traffic scene in front of a vehicle is an important task for autonomous driving [101]. Knowledge of the current traffic scene information can have several benefits: e.g., augmenting the driver's situational awareness, reducing driver workload, and automating all or part of the driving process. Despite the progresses in scene recognition [49], [84], [229], understanding the traffic scene in various environments remains largely unsolved. This is mainly due to the complexity of the traffic situations. First, many different traffic participants may be present and there are a variety of geometric layouts of roads

and crossroads. Furthermore, illumination conditions such as cast shadows caused by infrastructure or vegetation add extra complexities.

A traffic scene is generally composed of a collections of entities (e.g. objects) organized in a highly variable layout. This high variability in appearance has made reliable visual representation the primary choice in solving this problem. Among them, an image has been represented as bags of locally extracted visual features according to bag-of-features (BOF) methods, such as Scale Invariant Feature Transform (SIFT) [182] and Histogram of Oriented Gradient (HoG) [41]. For many high level vision tasks, these features can be pooled into an invariant image representation, e.g., Bag of Visual Words (BoVW) [38], Fisher Vectors (FV) [49], and Vector of Locally Aggregated Descriptors (VLAD) [110].

However, the rich variabilities hidden in the image cannot be reflected by the dominate patch encoding strategies, which are based on hand-crafted features. Recently, CNN have brought breakthroughs in image representations by emphasizing the significance of learning robust feature representations from raw data [127], [223]. CNN has the ability to detect complex features automatically by training multi-layer of convolutional filters in an end-to-end network, which is a prerequisite for many computer vision tasks, such as action recognition [268], vehicle recognition [262], [263] and object detection [77]. Despite these achievements, there are still some limitations in deep CNN, such as the lack of geometric invariance and the limitations in transferring information about local elements. Besides, a single classifier may have its own advantages and disadvantages in the classification task [298]. For the task of traffic scene recognition, a single classifier may be capable of learning some, but not all, specific characteristics of the traffic scene. So it is worth exploring multi-classifier fusion applied to traffic scene recognition to improve the classification performance.

To address the above issues, in this chapter, we propose a novel traffic scene recognition methodology in the setting of granular computing, which involves the creation of information granulation by extracting the CNN features upon local regions of the image for a compact representation, and design multiple levels of classifiers fusion method through fusing the outputs of the two ensemble classifiers (Random Forests and Gradient Boosted Trees) with the outputs of the selected single classifier. Second, we discuss how to improve the recognition rate by using the deep multi-classifier fusion method from the perspective of granular computing. Therefore, we are able to create information granulation and diverse classifiers to advance the performance.

To summarize, our main contributions are listed as follows:

- A deep multiple classifier fusion method based on granular computing has been proposed to create information granulation and multi-level of granularity, thus improve the performance for traffic scene recognition.
- A unified end-to-end deep network is built to integrate all algorithmic components, which makes the training process efficient and effective.
- We conduct extensive experiments and improve state-of-the-art traffic scene recognition performance on two benchmark datasets, WZ-traffic and FM2, and demonstrate the effectiveness of our proposal.

The rest of the chapter is organized as follows. In Section 3.2, we offer a brief overview of traffic scene recognition, multi-classifier fusion and granular computing. Section 3.3 provides a detailed description of the proposed methods. We also present how granular computing concepts are employed to design the framework for deep multi-classifier fusion. In Section 3.4, we describe the details of the new traffic scene dataset "WZ-traffic" which contains 20 traffic scenes classifications. For comparisons with other research, we conduct an analysis of the WZ-traffic and FM2 datasets, and discuss the results in terms of multiple comparison settings. In Section 3.5, we highlight the contributions of this work and suggest some future directions for research in this area.

3.2 Related Work

As an emerging research topic, traffic scene recognition has recently attracted significant interest [243], [169], [244]. In this section, we focus on three relevant research areas: traffic scene recognition, multi-classifier fusion and a review of granular computing concepts.

3.2.1 Traffic Scene Recognition

The automatic recognition of visual scenes is an important issue and plays a significant role in automatic transportation and traffic surveillance. A number of studies have been carried out under the daunting challenges of recognizing the traffic scene, mostly aimed at automatically analyzing the road environment, or detecting and classifying possible objects in the traffic scene, such as pedestrians and vehicles. For example, [57] proposed an urban scene understanding method by exploiting a pre-training classifier to label the segmentation regions. Also, a road classification scheme was introduced by [243], which

utilized the color, texture and edge features of the image sub-region. They then applied a convolutional network for the classification task.

Recently, based on the general data mining process, Taylor et al. [244] put forward a novel data mining methodology for driving-condition monitoring via CAN-bus data. In [162] a generalized Haar filter based deep network was applied for the object detection tasks in traffic scenes. A novel concept of the atomic scene has been proposed by [23], they established a framework for monocular traffic scene recognition by decomposing a traffic scene into atomic scenes.

3.2.2 Multi-classifier Fusion.

The effectiveness in solving classification tasks has been proven by many machine learning algorithms, such as the support vector machine (SVM) [36], k-nearest neighbours (KNN) [4], decision tree (DT) [79] and random forest (RF) [40]. A simple practice is to retain the best classifier and disregard the others after evaluating their performance. Alternatively, one could fuse the information provided by them, to achieve a better recognition rate. Recently, multi-classifier fusion has attracted attention in various computer vision tasks to achieve an improved performance. The final result of the classifiers fusion depends on the method of combining the decisions from different classifiers in accordance with the fusion rule.

In [128], six simple classifier fusion methods were theoretically studied, including minimum, maximum, median, average, oracle and majority votes. Due to the simplicity and good performance of these strategies, they may be the most obvious choice when building a multi-classifier system.

To determine the support $S_i(x)$ for class x_i, using the fusion rule R to perform a majority voting on the class-related probability predicted by each classifier, it can be defined as,

$$S_i(x) = R(P_{1,i}(x), , ..., P_{L,i}(x)), i = 1, 2, ..., m. \tag{3.1}$$

In the majority voting method, the class label of x predicted by each classifier should be computed firstly. Then, the support $S_i(x)$ can be robustly estimated as,

$$S_i(x) = \frac{v + 1}{L + m} \tag{3.2}$$

where v represents the number of votes received by the class x_i. Compared to frequency-based probability estimation, this probability usually does not affect the final result, while avoiding the problem of certain class labels that do not appear in the basic classifier output [53].

Fusion of feature sets and classifiers for facial expression recognition has been studied in [278]. Toufiq et al. [248] developed a dynamic decision selection method for face recognition that uses the least amount of facial information to take correct decision. In [177], a random subspace ensemble of SVM classifiers has been trained for scene recognition, and then the sum rules were used to combine the classifier results. In this work, we present a multi-classifier fusion approach by using various classifiers in the setting of ensemble learning which leads to an improvement in the recognition accuracy.

3.2.3 A Review of Granular Computing Concepts

From the aspect of philosophical perspectives, granular computing is a way of structured problem solving at the practical level [270]. There are two commonly concepts in granular computing: granules and granularity [188], [189]. In theory, a granule is defined as a collection of smaller units that can form a larger unit.

Various granules involves horizontal relationships and hierarchical relationship. If different granules involves horizontal relationships when if they are located in the same or different levels of granularity. Otherwise, these granules are in hierarchical relationships. For structural information processing, there are different levels of granularity for different sizes of granules. In ensemble learning, an ensemble of classifiers is viewed as a granule. Also, if the combination of classifiers involves different levels, each level represents a level of granularity.

In general, there are two main operations in granular computing including granulation and organization. The granulation operation aims at decomposing larger granules in a higher level of granularity into smaller granules at a lower level of granularity, while organization intends to integrate several parts into one. When designing the top-down and bottom-up approaches from a computer science perspective [269], the operations of granulation and organization are widely used, respectively [160].

In the content of set theory, a set of any formalism is regarded as a granule and each element in a set can be viewed as a particle. There are different formalisms of sets such as probabilistic sets [151], fuzzy sets [275], [137], interval-valued intuitionistic fuzzy sets

[8, 25] and rough sets [188]. They belong to information granulation which is one of the fundamentals of granular computing. In particular, a probability set can be considered a deterministic set when all elements belong to the set. Probabilistic sets provide a chance space to each set and view it as a granule. The chance space will be divided into subspaces which can be viewed as particles that are considered to be randomly selected to activate the occurrence of an event. Therefore, a whole chance space integrates all these particles.

The fuzzy sets view each set as a granule and gives each element a certain degree of membership in that set [27], [26]. In other words, each element belongs to a certain degree of a fuzzy set. In the setting of granular computing, a particle represents each part divided from the membership. In the context of rough set context, each set is viewed as a granule. As described in [151], rough sets use a boundary region to recover some elements with insufficient information.

Based on the above description, granular computing is effective in simplifying complex problems by breaking them down into several sub-problems. It can also be used to quantitatively measure qualitative properties in the context of information granulation. In practical applications, the theory of granular computing has been widely used to promote other research fields, such as computational intelligence [54], [116] and artificial intelligence [72], [165].

3.3 Overview of the Proposed Method

In this section, we describe the details of local deep-learning feature extraction and present the multi-classifier fusion framework. As illustration in Figure 3.1, the proposed method consists of four steps: 1) generating region proposals, 2) transfer learning, 3) reduction of feature dimensions, and 4) classification. The main components in our method will be described in detail. In addition, we will analyze the creativity of the method from the perspective of granular computing.

3.3.1 Region Proposal and Transfer Learning

In the setting of granular computing [149], a granule generally represents a large particle, which consists of smaller particles that can form a larger unit. Different from most existing methods which use global features extracted from whole images, we consider each image x as a granular and obtain a collection of local features from sub-granules: $x = \{x_1, x_2, ...x_n\}$.

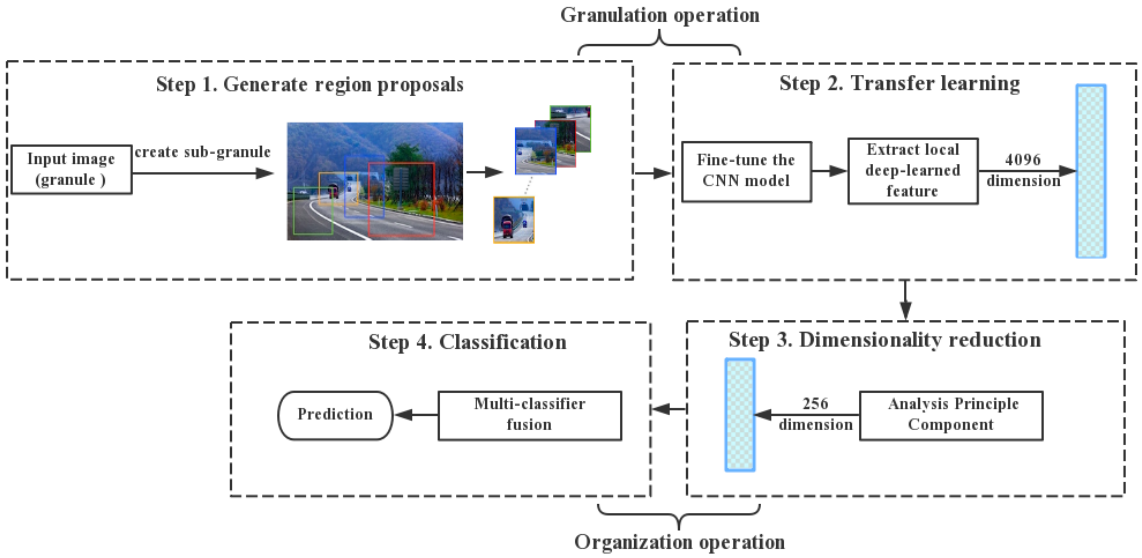


Figure 3.1: The workflow of our proposed traffic scene recognition system. The granulation operation includes generate region proposals (sub-granules) for each image (granule) and performs transfer learning to obtain local deep-learnt features. In the organization operation, we analyze the principle component to reorganize the local deep-learnt features and reduce their dimension of it. The type of traffic scenes can be recognized with multi-classifier fusion that also belongs to the organization operation.

So we capture contextual information from neighboring scenes and objects while preserving key local features. We start our work with a set of region proposals from images to pursue accuracy with affordable computing costs, each region proposal is viewed as a sub-granule of the original image. After observing the experimental results, we find that the top 1,000 ranked region proposals are sufficient for the representation of an image.

Once we have the 1,000 region proposals which were generated from the original images by the EdgeBoxes algorithm, we start the transfer learning in the second stage. We formalize transfer learning as follows: Given a source domain D_S and a target domain D_T, the learning task for D_S and D_T are T_S and T_T, respectively. We aim to use the knowledge from D_S and T_S to boost the learning ability of the target predictive function $f_T(\cdot)$ in T_T, where $D_S \neq D_T$, $T_S \neq T_T$. Transfer learning is particularly relevant when, given labeled source domain data D_S and target domain data D_T, we find that $|D_T| \ll |D_S|$.

In this chapter, we transfer knowledge from the ImageNet object recognition task P_1 to the target problem of traffic scene recognition P_2. In P_1, we have the task of object

classification with source domain data $D_1 = \{(x_{1_i}, y_{1_i})\}$ from ImageNet that consists of natural images $x_{1_i} \in X_1$ with labels. In P_2, we have a traffic scene prediction task with target domain data $D_2 = \{(x_{2_i}, y_{2_i})\}$ that consists of traffic scene images $x_{2_i} \in X_2$ and image labels. ImageNet is an object classification image dataset which consists of 14 million images belonging to 1,000 classes, major breakthroughs have been achieved with the help of sufficient data and CNN models in many computer vision tasks. CNN models trained on the ImageNet dataset are recognized as good generic feature extractors, with low-level and mid-level features such as edges and corners that are able to generalize to many new tasks. We achieve knowledge transfer using the parameters from VGG16 models trained on ImageNet. The VGG16 model has been fine-tuned on the traffic scene dataset using SGD with momentum.

We consider two ways of adapting the original VGG16 network. The first approach is to add a dropout layer before the final convolutional layer to reduce the risk of overfitting. Second, we modify the last fully-connected layer to have K neurons to predict the K-classes, where K is the number of the traffic scene types in the training set. We regard the traffic scene recognition as a multi-class classification problem, and apply the cross-entropy loss to transfer the model outputs to the value of probability for all classes. This corresponds to

$$l = -\sum_{k=1}^{K} log(\sigma p(k) q(k)) \tag{3.3}$$

where σ denotes the softmax activation function, $p(k) \in [0, 1]$ is the predictive probability of the input image belonging to class K and $q(k)$ denotes the ground truth distribution. Different with some methods which obtain features from the pooling layer, we extract the 4,096-dimensional feature vector from the first full connection layer (FC layer) for the region proposals generated from each image. However, it is time-consuming to extract the features of multiple regions (sub-granule) in the CNN.

To reduce the computational cost and run time, we implemented our algorithm on top of a fast R-CNN [77], in which the RoI projection scheme will complete the feature extraction of an image in only one feed forward process. Fast R-CNN is originally used for object detection and requires object category labels and annotations of bounding boxes. Usually, the annotations are done manually in general applications. In our work, the parts instances are viewed as objects and annotated automatically. We show the feature extraction process in Figure 3.2.

3.3.2 Dimensionality Reduction

As has being pointed out in [110], reducing the dimension of the original feature appropriately would further improve the recognition performance. Therefore, after extracting the CNN features from regions, we used principal component analysis [3] to reduce the feature dimension. However, it is not practical to perform conventional PCA training on all features due to the large number of features. We first randomly select some sample features for training and reduce the CNN features from 4,096 to 256 dimensions. Then we perform PCA on all the remaining features. In addition, we further investigate the effect of feature dimensions on overall recognition performance by comparing the performance of 512 dimensions.

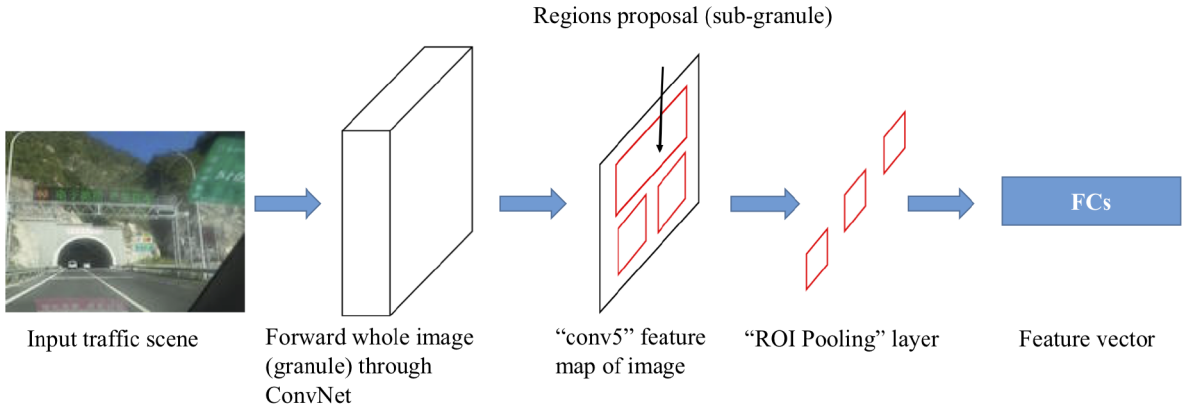


Figure 3.2: The process of deep feature extraction. Process the input traffic scene images (granule) contains a set of region proposals (sub-granule) through the CNN model, after generating the conv5 feature map of the image, the RoI pooling layer will extract features with one feed forward process.

3.3.3 Design of Multi-classifier Fusion Framework

There are two principles for multi-classifier fusion: a) each individual classifier has its own advantages; b) as indicated in [298], complementary advantages could to be achieved by encouraging diversity amongst the different classifiers.

Figure 3.3 shows the process of multi-classifier fusion. Firstly, we train several single classifiers including the popular SVM, KNN and MLP that have different learning strategies. To boost the recognition performance, in step 2, more diverse decision trees are obtained

by training two decision tree ensembles including Random Forests and Gradient Boosted Trees. To reduce the risk of over-fitting and to improve the level of generalization, we adopt the 10-fold cross validation to train and validate each classifier. Finally, we apply an algebraic rule to fuse the results of the two ensemble classifiers with the single classifiers to further improve the recognition performance. In particular, the proposed method involves the different levels of granularity. Each ensemble can be viewed as a granule, Random Forests and Gradient Boosted Trees are two independent granules. The final ensembles are organized to include the two ensembles and the single classifiers. Each of the levels of ensembles actually represents a level of granularity.

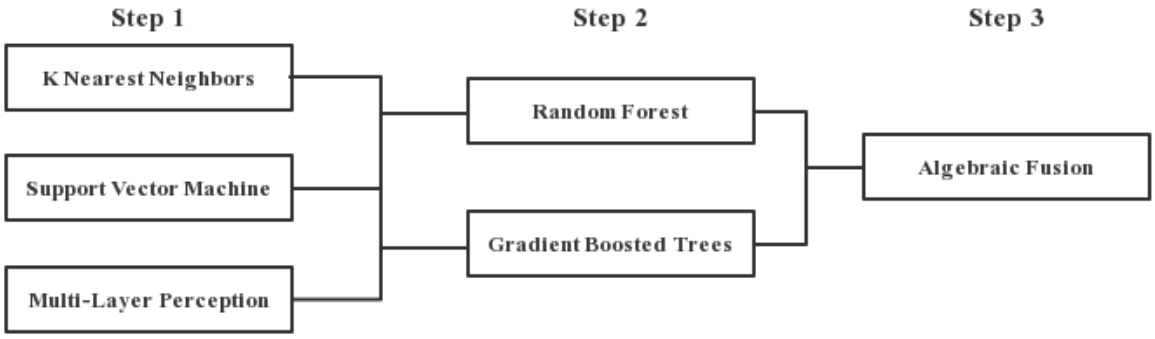


Figure 3.3: Step 1: train three single classifiers; Step 2: In order to increase the diversity of decision tree classifiers, two decision tree sets are trained by using RF and GBT respectively; Step 3: the trained single classifiers is combined with the decision tree sets through algebraic fusion.

Voting is the most popular method of classifier combination in the field of classifier fusion. In particular, voting-based set classification can be achieved by selecting the classes provided by most classifiers as their output, i.e. majority voting. In this way, voting-based ensemble classification is implemented.

Different from majority voting, weighted voting is another way of voting in which the class output is calculated with the weight of each single classifier. The class that obtains the highest weight will be derived for finally classifying an instance. The overall confidence (accuracy) of a classifier evaluated on a validation set will be used to estimate the weight of this classifier.

The precision or recall for a specific class are also used to measure the confidence in the

class level [150]. Also, due to the high degree of diversity between different instances, the confidence in classifying an instance cannot be represented by the confidence level measured for the classifier or each individual class. In our proposed framework, we use algebraic rules [298], which are based on the median/maximum/average of the hidden output (posterior probability of each class) to achieve the fusion of these classifiers trained by using different learning algorithms. Our traffic scene recognition algorithm is summarized in Algorithm 1.

Algorithm 1 Proposed traffic scene recognition pipeline

Input: Static traffic scene recognition dataset D including D_{train}, D_{val} and D_{test}.
Output: The prediction labels for D_{test}.
/*Granulation operation*/
1: Create region proposal (sub-granule) for traffic scene images (granule) in D.
2: Perform transfer learning using D_{train} and D_{val} (see Section 3.3.1).
3: Extract the local deep-learnt feature matrix H_{train}, H_{val} and H_{test} of the selected regions for each image in D_{train}, D_{val} and D_{test}.
/*Organization operation*/
4: Analyze the principal components in H_{train} to obtain the transformation matrix T.
5: **for** $i = 1$ to D_{train} **do**
6: Use the first i transformation vectors of T to compute. $H_{train_{transform}}$ by projecting H_{train} to the subspace of the principal components.
7: Evaluate the performance of $H_{train_{transform}}$ and save the result as $scores_i$.
8: **end for**
9: Obtain the i in which the $H_{train_{transform}}$ achieves the best scores. T_{select} is the first i transformation vectors of T.
10: Compute L_{train}, L_{val} and L_{test} by projecting H_{train}, H_{val} and H_{test} to the principal components subspace using W_{select}.
11: Train three basic classifiers KNN, SVM and MLP and two decision tree ensembles RF and GBT using L_{train}, L_{val}.
12: Obtain the posterior probability matrix P_{test} of the three basic classifiers and two decision tree ensembles on L_{test}.
13: Fuse the multiple P_{test} using algebraic rules.

.

3.3.4 Application of Granular Computing Concepts

We design the deep multi-classifier fusion method in the setting of granular computing, which is a paradigm of information processing. In the local deep-learned feature extraction

part, granulation is operated through decomposing the information of the original images into multiple region proposals which involves local information. Organization is operated through analyzing the principal components to reduce the feature dimensions. Different from general feature selection, we reorganize the various features into a low-dimension features with no information loss. A principal component is a feature that is regarded as a large information particle, which contains a plurality of features called small information particles. The whole process of dimensionality reduction belongs to information fusion, which utilizes the organizational operations in granular computing.

On the other hand, the framework of multi-classifiers fusion involves multiple levels of classifier fusion, and we view each of the levels as a specific level of granularity. In this setting, a primary ensemble containing three base classifiers is viewed as a granule at the basic level of granularity, whereas the final ensembles which may involves both base classifiers and lower level ensembles is viewed as a granule at the top level of granularity.

Multi-classifier Fusion vs. Deep multi-classifier fusion. Multi-classifier fusion and the proposed deep multi-classifier fusion have the same objective of outputting the prediction labels for the testing data. Multi-classifier fusion focuses on the classification task and leverages different classifiers to improve the performance. Deep multi-classifier fusion seamlessly integrates the two components including local deep-learned features extraction framework (step 1 to step 3 of Algorithm 1) and multi-classifier fusion into a unified system. In principle, the two components should collaborate with each other effectively: the former operation of granulation is essentially decomposition of the whole into multiple parts in a top-down information processing manner through extracting features from local patches through the FC layer of the CNN, whereas the latter organization operation is essentially the integration of multiple parts in a bottom-up information processing manner through achieving the complementary advantages of the different classifiers.

3.4 Experiments and Results

We will first describe the implementation details, and then briefly outline the experimental set up and performance comparison on the WZ-traffic and FM2 datasets.

3.4.1 Implementation Details

Deep Feature Extraction. Our experiments were conducted under the Linux operating system. The implementation of the deep feature extraction was undertaken on the Caffe deep learning framework [112]. We employed the VGG16, VGG-M-1024 and Cafenet models which were pre-trained on ImageNet, and then fine-tuned on specific datasets. We set the maximum number of training iterations and the learning rate to 10,000 and 0.0001, respectively. Other parameters are the same as the fast R-CNN [77].

Setting of Multi-classifier Fusion. The multi-classifier fusion experiment was built on the KNIME Analysis Platform, which has abundant nodes for applying machine learning algorithms. All experiments were conducted with 10-fold cross-validation. We divided each dataset into 10 parts including 7 parts for training and 1 parts for validation and the rest for testing. The performance of the three popular standard learning methods, SVM, KNN and MLP, were initially evaluated. We used the RBF kernel in the SVM learner and set the values of the sigma and overlap penalty to 13 and 1, respectively.

For the K nearest neighbors, we set the value of K equal to 7. In addition, we trained the MLP classifier through 150 iterations with 2 hidden layers and 10 units in each layer. Then we used the random forest learner (RF) and gradient boosted trees learner (GBT) to improve the performance of decision tree learning. As for random forest learner, the information gain ratio was used for the split criterion in the tree ensemble learner, we set the ensemble size, which means the number of decision trees that make up a random forest to 150. In addition, for the gradient boosted trees learner, the tree depth, number of models and learning rate were set as 10, 20 and 0.1, respectively. In the multi-classifier fusion stage, the mean, median and maximum rule of algebraic fusion were used to boost the prediction accuracy.

3.4.2 WZ-traffic Dataset

Although the task of traffic scene has already been studied for many years, there is still a lack of high quality traffic scene datasets. The existing dataset [221, 220] only collected images from the perspective of the driver, the position and orientation of the camera were changed slightly between videos. In addition, the categories of traffic scene in [221, 220] are also insufficient. Therefore, it's necessary to collect a traffic scene dataset with sufficient variations in traffic scene types, background and viewpoints.

To facilitate the research on traffic scene recognition and evaluate the proposed approach,

we created a new dataset of labeled traffic scenes, called the WZ-traffic dataset [261]. It contains 6,035 labeled images of 20 categories: highway, country road, gas station, indoor parking, outdoor parking, crossing, city stress, scenic gate, bridge, car wash, train station, autodrome, traffic circle, tunnel, tunnel entrance, bus station, booth, bus parking and traffic jams. The images were collected by us from both an image search engine as well as from personal photographs, and took into account sufficient variations in the background and viewpoints. Figure 3.4 presents sample examples from the corresponding traffic scene categories in this dataset.

Figure 3.4: Some examples of the WZ-traffic dataset.

We followed the step of deep feature extraction as previously explained, and applied multiple classifier fusion for the final prediction. To compare and evaluate the performance from different models, we selected the pre-trained CNN models VGG16, VGG-M-1024 and CaffeNet for following fine-tuning. We implemented the training process in the fast R-CNN framework [77]. After applying the region proposal algorithm, EdgeBoxes [300], to each image, we extracted the FC features from each region. Multi-classifier fusion was

Table 3.1: VGG16: Mean AP result on the WZ-traffic dataset [261] using different methods.

Method	Mean AP(%)
FC features (pre-trained model) [224]	83.12
FC features (fine-tuned model)	85.71
1,000 regions+FC features+PCA256	**87.43**
1,000 regions+FC features+PCA512	87.10
2,000 regions+FC features+PCA256	87.30
3,000 regions+FC features+PCA256	87.12

accomplished after PCA dimensionality reduction and feature clustering. More details about the experiment procedure are described as follows:

(1) Result from VGG16.

First, the fine-tuned CNN model and the pre-trained CNN model were applied to extract FC features. As shown in Table 3.1, with the same experimental setting, the fine-tuned model obtains about a 3% improvement (from 83.12% to 85.71%) in the recognition performance. This result indicates that the fine tuning of the CNN model can significantly boost the feature representation ability. Then, we provided recognition results for 2,000 and 3,000 boxes per image to verify that 1,000 regions per image are sufficient for deep feature representation. From Table 3.1, we can clearly observed that 1,000 boxes yields the best performance. To reduce the feature dimension, the PCA was used to reduce the CNN features from 4,096 dimensions to 256. We repeated the same experimental process and reduced the CNN features to 512 dimensions for comparison. It can be clearly seen, from Table 3.1, that the mAP results of 512 dimensions are slightly worse. Hence, the CNN features of 1,000 regions with 256 dimensions will be the focus for most of the experiments.

Table 3.2 shows the results for different single classifiers and multi-classifFier fusion. The outputs of the two ensemble classifiers (Random Forests and Gradient Boosted Trees) were fused with the outputs of the 3 single classifiers to further advance the performance further. The three multi-classifier fusion methods prove their capabilities of improving the recognition performance compared with the single classifiers. The highest recognition rates were obtained from the mean-based fusion method. Overall, the results show very supportive evidence for multi-classifier fusion towards advancing the overall classification performance.

Table 3.2: VGG16: Mean AP result on the WZ-traffic dataset [261] with individual and fusion classifiers.

Method (VGG16)	Mean AP(%)
MLP [71]	87.43
SVM [36]	88.26
KNN [4]	86.57
RF [40]	86.43
GBT [66]	88.06
Median-based fusion	89.90
Maxmium-based fusion	90.15
Mean-based fusion	**90.30**

(2) Results from VGG-M-1024 and CaffeNet.

To compare the performance with other CNN models, we select the middle scale CNN model VGG-M-1024 and small scale CNN model CaffeNet. The experiments were undertaken using the same conditions as VGG 16. Comparing the deep multi-classifier fusion methods result with VGG-M-1024 and CaffeNet which are 88.90% and 88.11%, respectively, in terms of the recognition rate, VGG16 performs better than VGG-M-1024 and CaffeNet models. Figure 3.5 shows the confusion matrix of our best recognition results on the WZ-traffic dataset. From the confusion matrix, we can observe that the proposed method performed well in recognizing tunnel, traffic circle and car wash. For the other types of traffic scene, our method also performed reasonably well.

3.4.3 FM2 Dataset

The FM2 dataset was introduced by Sikiric et al. [220] and contains 6,237 traffic scene images from the perspective of the driver. The images were extracted from videos of several drives on European roads, obtained using a camera installed in a vehicle. The traffic scene consists of dense traffic, highway, overpass, road, tunnel, exit, toll booth and settlement. Figure 3.6 provides some examples of the traffic dataset FM2.

There are no ground-truth regions provided in FM2 dataset, therefore, we fine-tuned the pre-trained VGG16 model which achieved the best performance on the WZ-traffic dataset compared with VGG-M-1024 and CaffeNet. When the training process of the CNN model was completed, we extracted the CNN features for the top 1,000 regions

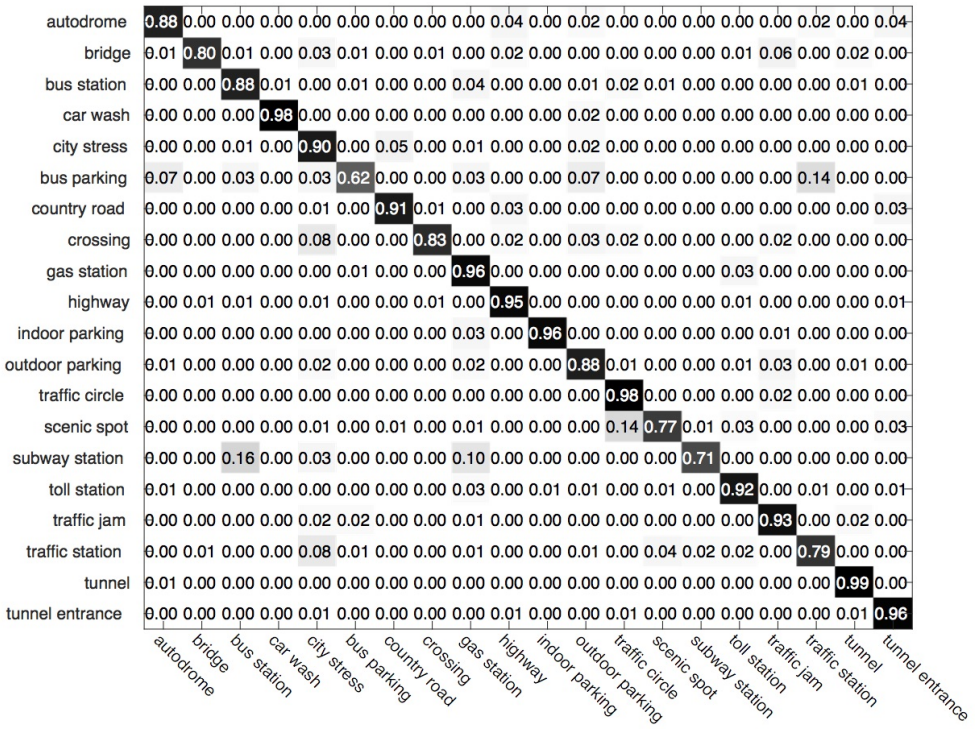


Figure 3.5: Confusion matrix of the best recognition results on the WZ-traffic dataset [261] (mean AP is 90.30%). The labels in the leftmost column and on the bottom represent the ground truth, the number in each row represents the corresponding prediction results.

identified by Edgeboxes. Multi-classifier fusion was accomplished after PCA dimensionality reduction. We can observe the following results from Table 3.3 and Table 3.4: On this dataset, satisfactory results are obtained when only the image-level CNN features are considered. Besides, the performance increased 0.87% (from 96.25% to 97.12%) when we implement the multi-classifier fusion on CNN feature. This improvement proves the complementarity of multi-classifier fusion and CNN features. Compared with the other methods shown in Table 3.5, we also obtained the most state-of-the-art results on the FM2 dataset.

Ablation Study

To verify the effectiveness of components in proposed method, we conducted ablation experiments on the FM2 for traffic scene recognition.

Table 3.3: VGG16: Mean AP result on the FM2 dataset [220] with different methods.

Method (VGG16)	Mean AP(%)
FC features(pre-trained model) [224]	93.41
FC features(fine-tuned model)	95.65
PCA256+FC features	**96.25**

Table 3.4: VGG16: Mean AP result on the FM2 dataset [220] with individual and fusion classifiers.

Method (VGG16)	Mean AP(%)
MLP [71]	96.25
SVM [36]	96.46
KNN [4]	95.87
RF [40]	96.13
GBT [66]	95.70
Median-based fusion	96.82
Maxmium-based fusion	96.95
Mean-based fusion	**97.12**

Figure 3.6: Some examples of the FM2 Dataset [220].

Impact of transfer learning: We directly extracted the CNN features from the first fully connected layers of the fine-tuned VGG16 model for each image without applying the region proposal algorithm to generate candidate objects. As shown in Table 3.3, the mAP accuracy is 95.65%. We evaluate the stand-alone performance of the fine-tuned VGG16 model by comparing the results of pre-trained VGG16 model in Table 3.3. The fine-tuned model produces a 2.24% improvement (from 93.41% to 95.65%) over the pre-trained model.

Impact of dimension reduction: In the test phase, we used the EdgeBoxes algorithm to generate 1,000 region proposal for each image, which are represented by 4,096-dimensional CNN features. To reduce the amount of feature computation and improve the performance, we perform dimension reduction on the CNN feature through PCA algorithm and reduce the CNN features to 256 dimensions. Table 3.3 shows that the mAP accuracy increases 0.6% (from 95.65% to 96.25%). In this setting, the multi-classifier fusion has not been taken into account.

Impact of Deep multi-classifier fusion: Finally, we fuse the hidden outputs (probability for each class) of the SVM, KNN, MLP, RF and GBT classifiers through the mean, median and maximum rule of algebraic fusion. Table 3.4 shows the detailed comparison results between our methods and five single classifiers baseline methods. Experimental results indicate that adding the multi-classifier fusion does improve the overall performance and the best performance in terms of mAP accuracy of mean-based fusion is 97.12%, Figure 3.7 presents the confusion matrices of the best recognition results on the FM2 database.

From the confusion matrix, we can see that the proposed method recognizes most of the traffic scene well, such as highway, tunnel and settlement. Figure 3.8 shows some correctly recognized examples in this dataset. For example, our method recognized Figure 3.8 (a) as booth with a 99.99% (0.9999) probability. We also compared our method with the state-of-the-art method in Table 3.5, and the comparisons indicate the competitiveness of the proposed method on the FM2 dataset.

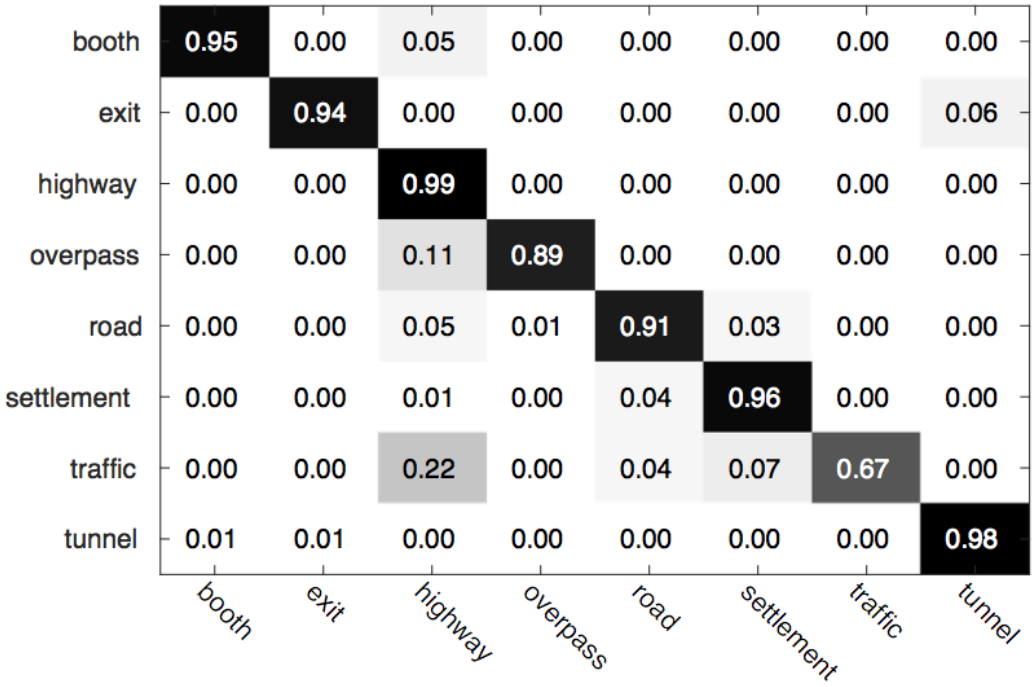


Figure 3.7: Confusion matrix of the best recognition results (mean AP is 97.12%) on the FM2 database [220]. The labels in the leftmost column and on the bottom represent the ground truth, the number in each row represents the corresponding prediction results.

3.5 Conclusion

In this chapter, we have proposed a novel deep multi-classifier fusion method in the setting of granular computing to improve the performance of traffic scene recognition. Different with the existing popular methods, we address the task of traffic scene recognition by creating information granulation and organization in a unified end-to-end deep network.

Figure 3.8: Some examples of correct recognition in the FM2 dataset [220], the predicted label and corresponding probability are provided for each image.

Table 3.5: Mean AP result on the traffic scene dataset FM2 compared previous results in [220].

Method	Mean AP(%)
BoW [39]	93.55
LLC [255]	92.68
SFV [125]	95.09
GIST[181]	93.30
Ours (Deep Multiple Classifier Fusion)	**97.12**

Specifically, the deep multi-classifier fusion method combines the advantages of local deep-learned features and multi-classifiers fusion. Through extracting CNN features of region proposal generated for each image, the information granulation will be created. In addition, organization is implemented by analyzing the principal components to reduce the feature dimensions. We also introduced a multi-classifier fusion method involves multiple levels of granularity to improve the performance. Thus, the deep multi-classifier fusion architecture makes it easy to handle the complex traffic scene. We conducted experiments on two different traffic scene datasets, including a public dataset and our own dataset. The experimental results show that the information of the local patches and the global background are significant to improve the performance of traffic scene recognition, while

the deep multi-classifiers fusion method brings performance improvement to traffic scene recognition. In the future, the deep multi-classifier fusion will be further improved to study the relationship between classes in a granular computing setup. Specifically, we will identify the relationships between information granules where each class is viewed as a granule. Besides, the proposed method in this chapter only take class labels of traffic scene into consideration. However, we find the attributes (e.g., weather conditions and road structures), containing detailed local descriptions, are beneficial in allowing the traffic scene recognition model to learn more discriminative feature representations. Therefore, we will propose an attribute-scene recognition network based on the complementarity of attribute labels and class labels in the future. We systematically investigate how the traffic scene and attribute recognition benefit each other.

Chapter 4

Vehicle Re-identification in Still Images: Application of Semi-supervised Learning and Re-ranking

The great success in applying deep neural networks to image classification, object detection and semantic segmentation has inspired us to explore their full ability in a wide variety of computer vision tasks. Unfortunately, training deep neural networks often requires a large amount of labeled data to learn adequate feature representations for visual understanding tasks. This greatly limits the applicability of deep neural networks when only a limited amount of labeled data is available for training the networks. Therefore, there has been an increasing interest in literature to learn deep feature representations in an unsupervised fashion to solve emerging visual tasks with insufficient labeled data. This chapter studies learning the representations of image in unsupervised and semi-supervised scenarios for vehicle re-ID task. Firstly, we obtain the unlabeled vehicle images by training the DCGAN which generated the distribution of photo-realistic images as a whole so that better feature representations can be derived from the trained generator. We then fine-tune the CNN model using the labeled dataset and unlabeled data to output feature representations with sufficient information. Finally, we perform the re-ranking step to improve the vehicle re-ID performance based on the feature representations of samples.

4.1 Introduction

With the explosive growth of video data captured by various surveillance cameras, there is an increasing demand for improved surveillance video analysis capabilities which require a large number of vehicle related tasks, such as vehicle detection, classification and verification. In this work, we focus on the task of vehicle re-ID in still images, which aims to quickly discover, locate and track the target vehicles across multiple cameras, thus automating the time consuming manual task. Vehicle re-ID has practical applications in surveillance systems and intelligent transportation [285]. In vehicle re-ID systems, a query image, also called a probe image, is compared with the gallery images that contain various vehicles captured by multiple cameras. Normally, a rank list is generated that has several matched images from the gallery set. Figure 4.1 further explains the vehicle re-ID task.

Traditionally, the combination of sensor data and multiple clues are used to solve the task of vehicle re-ID, such as the transit time [147] and the wireless magnetic sensors [129]. However, these methods are sensitive to the fickle environment (e.g., thunder and lightning) and require the extra cost of additional hardware. In addition, the license plate is an important clue which contains the unique ID of vehicle, thus the technologies related to license plate have been proposed in [201], [82]. Nevertheless, it is easy to occlude, remove, or even forge the license plate, especially in criminal circumstances. To alleviate these limitations, we focus on this task based on its visual appearance, which is essential for fully-fledged vehicle re-ID system.

To this end, the discriminative features should be extracted to distinguish different vehicles for robust vehicle re-ID [10]. Basically, there exists two challenges. (1) Different lighting and complex environments causes difficulties for appearance-based vehicle re-ID. Also, large variations in appearance will produce if capture vehicle using different cameras. How to take such large intra-class variance into account for feature representation is crucial. (2) Compared with the person re-ID, vehicle re-ID is more challenging as different vehicles can be visually very similar to each other, especially when they are from the same category. Figure 4.2 further explains the situations of intra-class variance and inter-class similarity.

The deep embedding method has shown generalization abilities and promising performance in the re-ID task, aiming to learning compact features embedded in some semantic spaces through a deep CNN. The objective of embedding is typically to express as pulling the features from similar images closer and pushing the features from dissimilar images further away. Among these methods, learning identity-sensitive and view-insensitive features

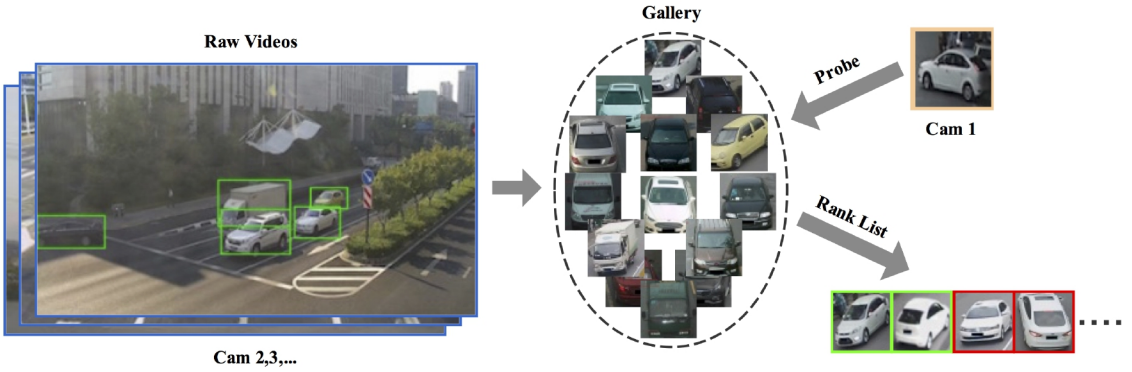


Figure 4.1: Explanation of the task of vehicle re-ID. Given a snapshot of a vehicle (the probe), a re-ID system retrieves from a database (the gallery) which contains a list of other snapshots of vehicles, usually taken from different cameras at different time, and ranks them by decreasing similarities to the probe.

are crucial to ensure the learning effectiveness of the CNN model. Hence rich labeled data from different camera views are required to learn a feature representation that is invariant to the appearance changes. However, relying on manually labelled data for each camera view results in poor scalability. This is due to two reasons: (1) It's a tedious and difficult task for humans to match an identity correctly among hundreds of data from each camera. (2) In real-world applications, there are a large number of cameras in a surveillance network (e.g., those in an airport or shopping mall), it's infeasible to annotate sufficient training samples from all the camera views. Therefore, these practical issues severely limit the applicability of the existing vehicle re-ID methods.

To alleviate the large demand of training data, the approaches of semi-supervised learning have been proposed recently which uses the unlabeled samples to boost the performance on a specific task. It is driven by the practical value in learning faster, cheaper, and better feature representations. Semi-supervised learning attempts to obtain a deep model that can more accurately predict unseen test data than a deep model learned only from labeled training data. Common semi-supervised learning methods include variants of generative models [119], co-training [287] and graph Laplacian based methods [51].

Above works in semi-supervised learning are based on the fact that sufficient unlabeled data is available. However, if the number of unlabeled sample is scarce or difficult to collect, traditional semi-supervised methods may become useless. In our work, instead of using unlabeled data from the real sample space, we propose a semi-supervised feature

(a) Intra-class variance **(b) Inter-class similarity**

Figure 4.2: Examples explaining the intra-class variance and inter-class similarity. (a) Due to the different viewing angles and illuminations levels of cameras, the images of each row on the left column from the same vehicle produce the significant intra-class variance. (b) The images of each row on the right column belonging to the different vehicles from the same class and produce inter-class similarities. It's challenging to distinguish the vehicles with similar appearance.

embedding method which directly uses a GAN to generate unlabeled samples. Goodfellow et al. [81] first proposed the GAN to obtain the optimal discriminator network between real samples and generated samples based on the min-max game between generator and discriminator. Besides, the performance of image generator network will be improved simultaneously. Rather than investigating how to enhance the quality of the generated samples [194], [7], our research will focus on how to use GAN to promote the performance of classifiers. Specifically, we incorporate the generated samples with original training images to train CNN models with semi-supervised learning.

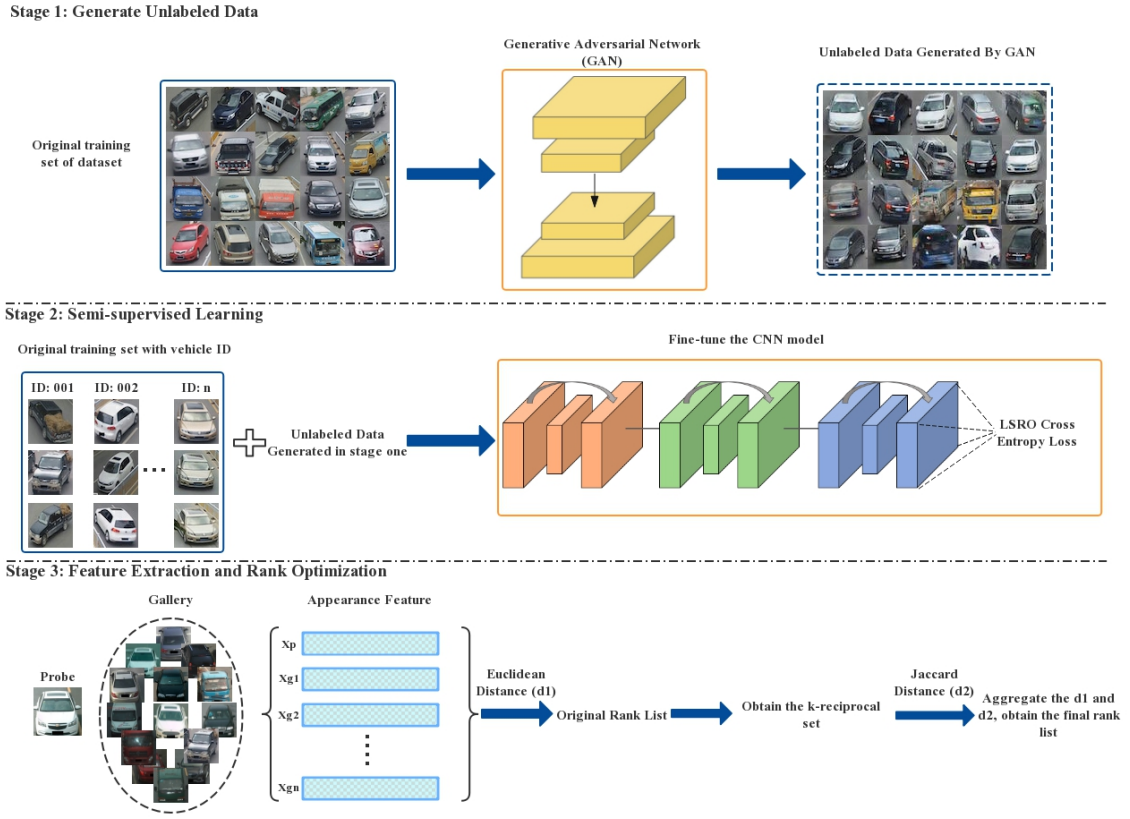


Figure 4.3: The workflow of the proposed method. There are three stages: (1) Generation of unlabeled data by using the original training set of vehicles to train the GAN [194]; (2) Semi-supervised learning by combining the labeled training set with vehicle ID and the unlabeled images data to fine-tune the CNN model with LSRO; (3) Feature extraction and rank optimization. We achieve an initial ranking based on the pairwise Euclidean distance of deep feature for the probe and each image in gallery set. To improve the initial ranking list, we finally add the re-ranking step.

As illustrated in Figure 4.3, there are three stages in the proposed algorithm. Initially, we obtain the generated vehicle images by using the original images in training set to train DCGAN [194]. In the second stage, we improve the discriminative power of the deep model for the re-ID task by using a larger training set which includes unlabeled images. More precisely, we use the initially labeled target dataset plus the unlabeled data generated in stage one to fine-tune the CNN model. In this manner, the improved ResNet-50 model [89]

is trained with all the data simultaneously. This stage is in the setting of semi-supervised learning, as the training dataset includes images with labels and images without labels.

Although significant progress has been achieved from previous researches of appearance based deep learning approaches for vehicle re-ID, their ranking accuracies are often unsatisfactory. To further improve the performance of vehicle re-ID, a technique is presented that uses a distance metric for rank optimization in the third stage. Specifically, we apply the trained CNN model from the second stage to extract the CNN features for probe image and each vehicle in gallery set. The initial ranking list can be achieved by calculating the pairwise Euclidean distances between the probe and the gallery. Then we compute the Euclidean distance and the Jaccard distance by comparing their k-reciprocal nearest neighbor set. We integrate the Euclidean distance and the Jaccard distance to obtain the proposed ranking list. We validate the performance of the proposed technique on three publically available vehicle re-ID datasets, VeRi-776 [158], VehicleID [152] and VehicleReID [277] dataset, all with promising results.

Our contributions can be summarized as follows:

- We propose a semi-supervised deep learning scheme for vehicle re-ID task which makes learning rich feature representations of vehicles from a limited number of labeled data possible.
- We present a re-ranking algorithm for ranking optimization which is firstly introduced for the vehicle re-ID task. Since the sample label is not required, the process of the re-ranking algorithm can be performed in unsupervised learning.
- We conduct extensive experiments and improve state-of-the-art vehicle re-ID performance on two benchmark datasets, VeRi-776 [158] and VehicleID [152] and demonstrate the effectiveness of our proposal. We apply the single shot setting on the VehicleReID [277] dataset for the first time and achieved promising results, providing baseline data for subsequent research.

4.2 Related work

As an emerging research topic, vehicle re-ID has recently attracted great significant interest [158], [152], [277], [157], [35]. In this section, we review the relevant works from three aspects: semi-supervised learning, re-ranking for person re-ID and vehicle re-ID.

4.2.1 Semi-supervised Learning

Semi-supervised learning exploits both the labeled data and unlabeled data to perform the learning task and bridges the gap between the fully-supervised learning and unsupervised learning. Some research exploits weak label annotations for each bounding box [185], or image [191] to enrich the training data. Compared with strong annotations, i.e., pixel-wise segmentations, weak annotations for bounding boxes and images cost less time. Therefore, they generally assume that there are a large number of weak annotations available for training, while the amount of training images with strong annotations are limited. In this setting, weakly annotated samples are used to update the supervised deep model by iteratively inferring and refining hypothetical segmentation labels.

A framework of semi-supervised feature selection has been introduced in [22], both labeled and unlabeled training data are exploited to analyse the feature space. The researches in [210], [179], [135] explore the idea of assigning virtual labels to the generated samples in the setting of semi-supervised learning. Salimans et al. [210] and Odena et al. [179] proposed an all-in-one method which simply take all the generated images as a new class. In practice, N defines the number of classes in the real training sets, then $N + 1$ is assigned to each generated sample. However, the generated samples tend to belong to the classes in N rather than the $N + 1$ class due to the fact that they are generated from distribution of the real samples. Without using an extra class, the method of assigning virtual label to generated samples has been proposed in [135], which exploits the maximum predicted probability generated for unlabeled image. After fedding an unlabeled sample into network, it will be fitted to a certain pre-defined class after several training epochs. A virtual label smoothing regularization for outliers (LSRO) was introduced by Zheng et al. [295] to address the over-fitting problem in [135]. LSRO assigns a uniform label distribution on generated samples to regularize the training process of deep network.

4.2.2 Re-ranking for Person re-ID

Recently, several re-ranking methods are proposed to improve the performance of person re-ID by optimizing the original ranking list [163], [166]. In [143], a re-ranking model is developed by analyzing the correlation of nearest neighbors of each pair images. Garcia et al. [69] introduced a re-ranking method for person re-ID, in which the content and content information are both considered to remove ambiguous samples. A bidirectional ranking method has been proposed in [139], which joins the contextual similarities with content

similarities to revise the initial ranking list.

Some researchers have exploited the nearest neighbors of the multiple baseline methods to the re-ranking task [271], [272]. In [271], the common nearest neighbors of local and global features are combined as new queries, then aggregate the global and local feature to optimize the initial ranking list. Ye et al. [272] calculated both the similarity and dissimilarity of the k-nearest neighbor set from different baseline methods to optimize the initial ranking list. These re-ranking methods have made contributions to discover the potential information from the k-nearest neighbors.

However, the overall performance from the above works may be restricted if the k-nearest neighbors are used to achieve the task of re-ranking directly, because false matches are often included. In the literature, the k-reciprocal nearest neighbor [111], [193] is effective to increase the amount of true matches on the top-k images. We regard the two images as k-reciprocal nearest neighbors [193] if they are both ranked between top-k in the ranking list when the other image is used as the probe. In this chapter, we propose an effective re-ranking method for vehicle re-ID and study the importance of the k-reciprocal neighbors.

4.2.3 Vehicle re-ID

In recent years, the researches on various computer vision tasks have achieved significant progresses, including object matching [241], [61], traffic scene recognition [260], action recognition [256], [268] and vehicle related works [35], [153]. Several researchers have proposed to apply the visual characteristics and the semantic attributes for vehicle retrieval. A vehicle retrieval and detection system was presented in [62], in which the task of attribute recognition and vehicle retrieval were both achieved. Liu et al. [158] exploited the real-world spatial-temporal environment to achieve a content assisted search for vehicle re-ID. There are some works focused on applying the LDA [291], [114] to optimize distance metrics in re-ID tasks. LDA learns a transformation matrix for feature space from high-dimensional to low-dimensional while preserving the class discrimination information as much as possible [232]. In [187], Local Fisher Discriminant Analysis was employed to learn a distance metic. Wu et al. [264] approximated the variations of intra-class and inter-class by training a hybrid deep architecture with an LDA criterion.

Additionally, hybrid features have been proposed to enhance the recognition of vehicle characteristics in some published works. For example, Cormier et al. [35] proposed a mixed

descriptor for low resolution vehicle re-ID, in which the local variance and local binary patterns (LBP) were combined. Liu et al. [153] presented a vehicle re-ID method that incorporated the feature of metric learning and vehicle model into one network. Despite these progresses on vehicle re-ID, how to exploit unlabeled samples and the re-ranking algorithm have not been well investigated in detail, which can significantly influence vehicle recognition performance. In this work, we propose to use GAN generated samples and re-ranking to boost the vehicle re-ID performance of off-the-shelf CNN.

4.3 Proposed Approach

4.3.1 Generative Adversarial Networks

A generator and a discriminator are two sub-networks in the GAN [81]. A generator produces a model distribution by transforming a random noise seed. A discriminator then tries to distinguish between samples between that model distribution and the target distribution. The training process of adversarial can be regarded as a minimax game: both the generator and discriminator oppose each other's objective and minimize its own cost, which leads a converged status that minimize the distance between the distribution of real samples and generated samples.

However, GANs have been known to be unstable to train, often resulting in generators that produce nonsensical outputs. There have been very limited published studies in trying to understand and visualize what GANs learn, and the intermediate representations of multilayer GANs. To solve these problems, DCGAN [194] improves the architectural topology of GANs, which makes them stable to train in most settings. Compared with GAN, DCGAN modifies the network details based on the original framework structure. Recently, many other variants of GAN have been proposed, such as conditional GAN [105] and stackedGAN [284]. While most of the previous researches are focus on studying the methods of generating more complex sample by training with high-quality images of objects. However, we do not focus on investigating more sophisticated sample generation methods, we aim to generate generate unlabeled samples from the low-quality surveillance image of vehicles. In consideration of all the above reasons, we choose to apply the DCGAN network [194] as the image generated model, thus helping improve the discriminative learning.

Five deconvolution functions are used to expand the tensor, which is defined as a data container with an N-dimensional array. The stride of the deconvolution filters 2 and their

size is 5×5. Following with a tanh activation function, we add one deconvolutional layer with a stride of 1 and kernel size 5×5 to fine-tune the result. An image can then be drawn from the generator net after training. We combine the original training set with the generated images and then fed them into the discriminator network. Five convolutional layers with a stride of 2 and kernel size 5×5 are used to identify whether the generated images are fake.

4.3.2 Label Smoothing Regularization for Outliers

Our model computes the probability of each class $n \in \{1, 2, ..., N\}$: $p(n|x) = \frac{exp(z_n)}{\sum_{n=1}^{N} exp(z_n)}$ for each training image x. Here, N is the number of pre-defined classes in the training set and z_n represents the logits or unnormalized log-probabilities. We normalize the ground-truth distribution over labels $q(n|x)$ for image x so that $\sum_n q(n|x)$=1. We define the he cross-entropy loss as Equation 4.1, which omits the dependence of p and q on example x.

$$l = -\sum_{n=1}^{N} log(p(n))q(n) \tag{4.1}$$

Minimizing the cross-entropy loss is equal to maximize the expected log-likelihood of a label, which is selected according to its ground-truth distribution $q(n)$. Cross-entropy loss is widely appied for gradient training of deep models. The gradient can formulated as $\frac{\partial l}{\partial z_n} = p(n) - q(n)$, the bounded range for it defined as $[-1, 1]$. Suppose there exsits a single ground truth label y, we can express the $q(n)$ as:

$$q(n) = \begin{cases} 0 & n \neq y \\ 1 & n = y \end{cases} \tag{4.2}$$

In this case, the objective of minimizing the cross-entropy loss is equal to maximize the predicted probability of the expected log-likelihood of the ground truth label. For a particular image x with ground truth y, the log-likelihood is maximized for $q(n)$, which equals to 1 for $n = y$. This maximum is not achievable for finite z_n but is approached if $z_y > z_n$ for all $n \neq y$, which means the logit of ground-truth label is larger than other logits. However, two problems can be caused. First, it may result in overfitting: the generalize can not be guaranteed if the model assigns full probability to the ground-truth label for each training example. Second, the model is overconfident about its predictions, resulting

in a larger difference between the maximum logit and all other logits.

To address the second problem, the label smoothing regularization (LSR) has been introduced in [240] to encourage the model to be less confident. While it not consistent with the goal of maximizing training tags, it does regularize the model and make it more adaptable. In [240], the label distribution $q_{LSR}(n)$ is written as:

$$q_{LSR}(n) = \begin{cases} \frac{\epsilon}{N} & n \neq y \\ 1 - \epsilon + \frac{\epsilon}{N} & n = y \end{cases} \tag{4.3}$$

where $\epsilon \in [0, 1]$ is a smoothing parameter. If set ϵ to zero, Equation 4.3 will reduce to Equation 4.2. On the contrary, the model may not be able to predict ground truth label if ϵ is too large. Therefore, the value of ϵ equals 0.1 in most cases. The cross-entropy loss evolves to Equation 4.4 by considering Equation 4.1 and Equation 4.3:

$$l_{LSR} = -(1 - \epsilon) log(p(y)) - \frac{\epsilon}{N} \sum_{n=1}^{N} log(p(n)) \tag{4.4}$$

In order to use the generated images in the process of deep feature learning, Zheng et al. [240] propose the label smoothing regularization for outliers (LSRO) method, which extends LSR [240] from the fully-supervised learning to the semi-supervised learning. It assumes the generated samples do not belong to any pre-defined class and sets the virtual label distribution to be uniform over all classes. Therefore, the maximum probability that is produced for the generated samples will be very low, which makes the network cannot make prediction for them. So the class label distribution for the unlabeled samples $q_{LSRO}(n)$ is defined as:

$$q_{LSRO}(n) = \frac{1}{N} \tag{4.5}$$

We combine Equation 4.1, Equation 4.2 and Equation 4.5 to rewrite the cross-entropy loss:

$$l_{new} = -(1 - Z) log(p(y)) - \frac{Z}{N} \sum_{n=1}^{N} log(p(n)) \tag{4.6}$$

For a real training image, $Z = 0$. For a generated training image, $Z = 1$. Therefore, the loss for the real images and generated images are different in the system. During the

training process, we define the loss of LSRO on a generated sample as follows:

$$l_{LSRO} = \frac{1}{N}\sum_{n=1}^{N} log(p(n)) \tag{4.7}$$

With the help of LSRO, we can regularize the model by processing more training images (outliers) that are located near the real training images in the sample space, which introduces more variances such as lighting and color. For example, if only one black-color vehicle exists in the training set, the discriminative power of the model will be limited because the model may be misled and regarded the black-color as discriminative feature. By adding generated images, such unlabeled black-color vehicle, the classifier will be punished if it misjudges the labeled black-color vehicle. In this manner, the network will be encouraged to look for more underlying causes and to be less prone to over-fitting.

4.3.3 Re-ranking Method

Problem Definition. Given a gallery set $G = \{g_i | i = 1, ..., T\}$ and a probe vehicle image b, where i defines the index of each image and T is the size of the gallery. After comparing the Euclidean distance between probe b and each image in gallery g_i, we reorder the indices of images in G so that $\{g_1, g_2, ..., g_T\}$ correspond to $L(b, G)$. The similarities between b and g_i satisfy $S(b, g_1) > S(b, g_2) > S(b, g_3) > \cdots > S(b, g_T)$. The objective of re-ranking method is to make more true matches rank top in the ranking list, thus improve the performance of the vehicle re-ID.

K-reciprocal Nearest Neighbors. Following [193], we define the k-nearest neighbors as the top-k samples of the ranking list of a probe b, it can be expressed as $R(b, k)$:

$$R(b, k) = \{g_1, g_2, ..., g_k\} \tag{4.8}$$

A potential assumption is that the returned image can be used for the subsequent re-ranking when it ranks within the k-nearest neighbors of the probe. However, some traditional methods which directly using the top-k images in the ranking list to perform re-ranking may introduce noise into the system and affect the final result. Therefore, we apply the k-reciprocal nearest neighbor $R^*(b, k)$ [111], [193] to solve this problem. It can be defined as:

$$R^*(b, k) = \{g_i | (g_i \in N(b, k)) \wedge (b \in N(g_i, k))\} \tag{4.9}$$

Algorithm 2 Rank Aggregation Algorithm

Input: A probe image b and a gallery set $G = \{g_i | i = 1, ..., T\}$
Output: A rank list for the probe image
Offline:
Compute the pairwise Euclidean distance between the probe vehicle b and images in gallery set.
Reorder the indices of images in G by sorting the pairwise Euclidean distance.
Correspond the set $\{g_1, ...g_T\}$ to the initial ranking list $L(b, G)$, and obtain the top-k galleries $R(b, k)$ from $L(b, G)$ of the probe image.
Query each image g_i in the gallery G.
Obtain the top-k galleries $R(g_i, k)$ of each image g_i.
Online:
for $i = 1$ to $|L(b, G)|$ **do**
    g_i is the i-th item in $L(b, G)$
    Get the k-reciprocal nearest neighbors of probe b by Equation 4.9
    Add more positive samples into $R_{new}(b, k)$ by Equation 4.10
end for
for $i = 1$ to $|R^*(b, k)|$ **do**
    g_i is the i-th item in $R(b, k)$
    Compute the new distance $d_j(b, g_i)$ between b and g_i by the Jaccard metric of their k-reciprocal sets as Equation 4.11
    Compute the final distance d_f between b and g_i as Equation 4.12
end for
Use the final distance to obtain the new rank list revised ranking list $L_{new}(b, G)$

.

Rank Aggregation. Compared with the k-nearest neighbors, the k-reciprocal nearest neighbors are more relevant to probe b. However, the true matches may not appear in the $R^*(b, k)$ due to the variations in occlusions, illuminations, poses and views. To solve this problem, for each sample q in $R^*(b, k)$, we add the half of the samples in its k-reciprocal nearest neighbors set into another set $R_{new}(b, k)$ as the following step:

$$R_{new}(b, k) \leftarrow R^*(b, k) \cup R^*(q, \frac{1}{2}k) \tag{4.10}$$

Therefore, $R_{new}(b, k)$ includes more images that are more relevant to the samples in $R^*(b, k)$. Then we consider the $R_{new}(b, k)$ as contextual knowledge and re-calculate the distance between the deep features of the probe and the images in gallery set. As described

in [272], the similarity of two images is higher if more duplicate samples in their k-reciprocal nearest neighbor sets. We calculate the new distance between the k-reciprocal sets of b and gallery g_i according to the Jaccard metric:

$$d_j(b, g_i) = 1 - \frac{|R_{new}(b, k) \cap R_{new}(g_i, k)|}{|R_{new}(b, k) \cup R_{new}(g_i, k)|} \tag{4.11}$$

Inspired by [297], the original distance and the Jaccard distance are aggregated to emphasize the importance of the original distance and improve the initial ranking list. We define the final distance d_f as:

$$d_f(b, g_i) = (1 - \lambda)d_j(b, g_i) + \lambda d(b, g_i) \tag{4.12}$$

where λ represents the weight of original distance in the final distance, and d represents the Euclidean distance. Finally, we obtain the new ranking list for probe b $L_{new}(b, G)$ by sorting the final distance d_f. We denote the size of $R_{new}(b, k)$ and $R(b, k)$ as $k1$ and $k2$, respectively. Our rank aggregation algorithm is summarized in Algorithm 1.

4.3.4 Complexity Analysis

In the proposed re-ranking method, calculating the pairwise distance of all image pairs requires a large amount of computational cost. We define the gallery size as t, $O(t^2)$ and $O(t^2 \, log \, t)$ represent the computation complexity of distance measure and the ranking process, respectively. Since the work of calculating the pairwise distance and obtaining the initial ranking list for the probe can be done in advance offline, the computation costs will be reduced in practical applications. Therefore, the computation costs include only $O(t)$ and $O(t \, log \, t)$, the former representing the calculation of pairwise distance between probe and gallery, the latter representing the complexity of ranking all final distances.

4.4 Experiments Results and Discussion

4.4.1 Datasets Introduction

Extensive experiments are conducted on three vehicle re-ID benchmark datasets: VeRi-776 [158], VehicleID [152] and VehicleReID dataset [277].

VeRi-776 [158] consists of 50,000 labeled images of 776 vehicles collected by 20 cameras in a road network in 24 hours. The specific information of vehicles are also provided, such as car model, camera locations and license plates. The dataset has been divided into two

parts, a training set and a testing set. The training set contains 37,778 images of 576 vehicles, and the testing set consists of 9,919 images belong to 200 vehicles. For the vehicle re-ID task, the 1,678 probe vehicle images in testing set are selected randomly to search the other images in testing set.

VehicleID [152] is currently the largest publicly available vehicle re-ID dataset. It contains 222,628 images belonging to 26,328 vehicles collected from the traffic surveillance system. There are two parts in the dataset: a training set and a testing set. The training set contains 113,346 images belong to 13,164 vehicles and the testing set contains 109,282 images captured from 13,164 vehicles. The testing data provides three subsets including small, medium and large scale for the vehicle re-ID task.

VehicleReID [277] contains 1,232 vehicle image pairs obtained from two surveillance cameras. The appearance of the same vehicle is changed by variations of viewpoints, illuminations and the locations of cameras.There are 553 vehicles from camera view A and 530 from camera view B, with 423 common vehicles in both views.

4.4.2 Implementation Details

CNN Baseline The ResNet-50 [89] model, which pre-trained on the ImageNet dataset is slightly improved and used in our experiments as the basic CNN network. We fine-tune the model using the training set to classify the training identities. ResNet-50 is a state-of-the-art architecture that exhibits top performance in several tasks in the field of computer vision, such as face identification, object classification and action recognition. It is composed of multiple basic blocks that are serially connected to each other and introduces shortcut connections summed after every few layers, so as to represent residual functions. In such way, it allows for a very deep architecture without hindering the learning process and at the same time shows less complexity in comparison to other networks of even smaller depth. Although there exists deeper versions of ResNet, we choose the 50-layer variant as the baseline model, as computation time is still crucial for this task.

We use the Matconvnet [252] package to implement the network training and resize all the images to 256×256. During training, random horizontal flipping is applied to crop the images to 224×224 randomly. A dropout layer has been inserted before the final convolutional layer to reduce the possibility of overfitting. Assume the original training set has K vehicle identities, we add K neurons in the last fully-connected layer to predict the K-classes. In most existing deep re-ID models, the final convolution layer will compute

the feature vector. Inspired by [156], which demonstrates that the useful information of mid-level identity-sensitive can be obtained before the last fully-connected layer in a DNN, as shown in Figure 4.4, we thus concat the $5a$, $5b$ and $5c$ convolutional layers of the ResNet-50 structures into a 2048-*dim* feature vector after the last fully-connected layer.

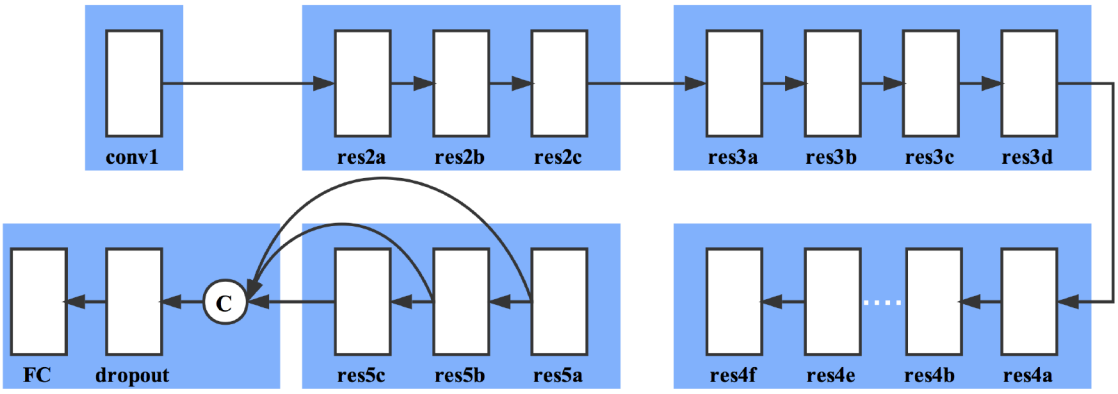


Figure 4.4: The structure of the improved ResNet-50 model. There are two modifications in the ResNet-50 model structure, 1) we add a dropout layer before FC layer to reduce the possibilities of overfitting; 2) we concat the $5a$, $5b$ and $5c$ convolutional layers to obtain mid-level identity-sensitive information.

The GAN model We used the Tensorflow [1] and DCGAN package to train the GAN model. Before training, we resize all the images in the training set to 128×128 and perform randomly flipped on them. The model is trained with mini-batch SGD with a mini-batch size of 64. We use a zero-centered Normal distribution to initialize the weights and set the standard deviation as 0.02. We apply the Adam stochastic optimization with parameters $\beta 1$ and $\beta 2$ which are used to define a memory for Adam, and average the gradient and squared gradient, respectively. Following the practice in [118], the good default settings for the tested machine learning problems are $\beta 1 = 0.9$, $\beta 2 = 0.99$. During testing, we fed a 50-*dim* random vector with Gaussian noise distribution into the GAN to generate vehicle images. Finally, all the generated samples are resized to 256 × 256 and are used in training the CNN with the LSRO. Figure 4.5 illustrates the generated and real samples on these three datasets. Although human can easily recognize the generated samples as fake, they are still effective in promoting the performance by adding the LSRO as virtual labels

in our experiment.

Figure 4.5: Examples of original images in training set and images generated by GAN from (a) VeRi-776 [158], (b) VehicleID [152], (c) VehicleReID [277].

Evaluation Metrics We use Mean Average Precision (mAP) and Cumulative Match Curve (CMC) to measure the re-ID performance.

Mean Average Precision: The mAP metric evaluates the overall performance of re-ID. For each probe image b, average precision is calculated as follows:

$$\rho = \frac{\sum_{k=1}^{n} P(k) \times rel(k)}{N_{gt}} \tag{4.13}$$

where k defines the rank in the list of retrieved vehicles, n denotes the number of retrieved vehicles, N_{gt} is the number of ground truth retrievals for the probe. $P(k)$ denotes the precision at cut-off k, $rel(k)$ indicates whether the k-th recall image is right match or not.

So we define the mAP as follows:

$$mAP = \frac{\sum_{b=1}^{Q} \rho}{Q} \tag{4.14}$$

where Q denotes the number of probe images.

Cumulative Match Characteristics: The CMC curve describes the expectation of positive samples within the first k ranks, we calculate the CMC value for top k ranks as follows:

$$CMC@k = \frac{\sum_{i=1}^{Q} f(b_i, k)}{Q} \tag{4.15}$$

where b_i is i-th probe vehicle, $f(b_i, k)$ is an indicator function which equals to 1 when the positive samples are within the top k ranks, otherwise, it equals to 0.

4.4.3 Semi-supervised Learning Results

Performance Comparisons on VeRi-776 Dataset The proposed method was evaluated on the VeRi-776 dataset [158] firstly which is the only existing vehicle re-ID dataset providing spatial and temporal annotations. We used the previously explained semi-supervised learning of the CNN model, and applied the re-ranking for the final identification. The CMC metric and mAP are adopted for the evaluation. We describe the details of experiment procedure and three comparative settings as follows:

(1) The CNN baseline.

Following the procedure of training and testing described in Section 4.4.2, the final results of the VeRi-776 dataset are reported in Table 4.1, Table 4.2 and Table 4.3. To evaluate the stand-alone performance of ResNet-50, we extracted the CNN feature from the first FC layer for each vehicle image and directly apply it for vehicle re-ID as a comparative baseline. As shown in Table 4.1, the CNN model with dropout layer gains about 5.46 points increase in mAP, from 48.90% to 54.36%. To select the best dropout rate, the extensive comparative experiments were further performed. As can be seen in Table 4.1, the best performance was achieved when the dropout rate is 0.9. Therefore, in our implementation, the final result of CNN baseline has a Rank-1 match rate of 85.88% and 54.59% mAP. We also compared the result of CNN baseline with other published vehicle re-ID results, from Table 4.2, the CNN baseline achieves better

Table 4.1: Match rate (CMC@Rank-R, %) and mAP (%) under different dropout rate on the VeRi-776 dataset [158]

Methods	Rank-1	Rank-5	Mean AP
CNN baseline (Without dropout layer)	82.54	90.52	48.90
CNN baseline (Dropout rate 0.5)	84.74	92.49	54.36
CNN baseline (Dropout rate 0.6)	86.23	92.37	53.95
CNN baseline (Dropout rate 0.7)	85.52	92.13	53.17
CNN baseline (Dropout rate 0.8)	85.76	92.67	54.47
CNN baseline (Dropout rate 0.9)	**85.88**	**92.85**	**54.59**

Table 4.2: Match rate (CMC@Rank-R, %) and mAP (%) for different methods on the VeRi-776 dataset [158]

Methods	Rank-1	Rank-5	Mean AP
FACT [157]	50.95	73.48	18.49
FACT+Plate-SNN+STR [158]	61.44	78.78	27.77
Siamese-CNN+Path-LSTM [219]	83.49	90.04	58.27
VGG+C+T+S [290]	86.59	92.85	57.40
CNN Baseline (Ours)	85.88	92.85	54.59
SSL (Ours)	88.57	93.56	61.07
SSL+re-ranking (Ours)	**89.69**	**95.41**	**69.90**

performance than previous works [158], [157]. There are no unlabeled samples in this scenario and the re-ranking methods have not been taken into account. We report the results of semi-supervised learning with different numbers of generated images in Table 4.3. The performance of vehicle re-ID has been improved when we fed different numbers of unlabeled data into the process of CNN training, which implies that CNN features alone are insufficient compared with semi-supervised learning.

(2) Semi-supervised learning with different numbers of generated images.

We trained DCGAN on the VeRi-776 training set, and combined the original training set with the generated images to fine-tune the CNN model. We evaluated the effect of the number of generated images on re-ID performance. Since unlabeled data is easy to obtain, we hope that as the number of unlabeled images increases, the model will obtain more general information. We compare the number of real training images (37,778) with the number of generated images fed into network, then two conclusions

Table 4.3: Match rate (CMC@Rank-R, %) and mAP (%) after using different numbers of generated images on the VeRi-776 dataset [158]

The number of generated images	Rank-1	Rank-5	Mean AP
0 (basel.)	85.88	92.85	54.59
2,000	86.12	92.96	55.43
5,000	86.78	93.21	57.68
8,000	88.31	93.35	59.34
10,000	**88.97**	**93.56**	**61.07**
30,000	88.19	93.54	59.00
50,000	87.90	92.90	59.10
70,000	87.34	92.61	58.87

are obtained after analyzing the results in Table 4.3. First, the baseline has been consistently improved by adding different numbers of generated images. Adding approximately 2 times generated images (70,000) that of the real training set still obtain +1.44 points improvement to rank-1 match rate.

Second, the peak performance is achieved when 0.3 times generated images (10,000) that of the real training set are added. From Table 4.3, when 10,000 generated images are added to the semi-supervised learning, the re-ID performance on VeRi-776 has been significantly improved. We observed the improvement of 3.09 points (from 85.88% to 88.97%), 0.71 points (from 92.85% to 93.56%) and 6.48 points (from 54.59% to 61.07%) in the Rank-1, Rank-5 match rates and mAP, respectively. Too many or too few images generated images incorporated into the semi-supervised learning will produce negative impacts on the model.

In semi-supervised learning with LSRO, generated images are used to learn more discriminative features and reduce the possible of over-fitting by assigning a uniform label distribution to the generated images to regularize the CNN model. When we incorporate too few GAN samples, the regularization ability of the LSRO is inadequate. In contrast, if we add too many GAN samples to fine-tune the network, the CNN model will tend to converge towards assigning a uniform label distribution to all the training images, which lead to overfitting and affect the discriminative learning from real images.Therefore, we recommend to make a trade-off of GAN samples to avoid poor regularization and overfitting.

(3) Ranking Optimization with different metrics.

Table 4.4: Match rate (CMC@Rank-R, %) and mAP (%) for the compared methods on the VeRi-776 dataset [158]

Methods	Rank-1	Rank-5	Mean AP
SSL+KISSME	86.84	92.37	60.12
SSL+KISSME+ re-ranking	**88.66**	**94.62**	**64.71**
SSL+XQDA	87.49	93.80	60.11
SSL+XQDA+ re-ranking	**88.72**	**94.92**	**67.48**

We set the parameter $k1$ =50, $k2$ = 10, and λ=0.3 which have the best performance in the test. After adding the step of re-ranking, the Rank-1, Rank-5 match rates and mAP are further improved to 89.69%, 95.41% and 69.90%. Table 4.2 compares the performance of our best approach and semi-supervised learning with re-ranking, against other state-of-the-art methods.

We compare our results with the methods in [158],[157], in which the hand crafted features were adopted for vehicle re-ID. It can be observed that our method achieves significant improvement over them, proving the advantage of deep feature. In [158], the license plate information (Plate-SNN) and spatio-temporal information were additionally used to improve the performance of vehicle re-ID. Compared with [158], our method based on vehicle appearance further yields an improvement of 28.25 points (from 61.44% to 89.69%) in Rank-1, 16.63 points (78.78% to 95.41%) in Rank-5 and 42.13 points (from 27.77% to 69.90%) in mAP. We also compare our method with the appearance-based deep learning approach [290], which improved triplet-wise training of CNN for vehicle re-ID. As shown in Table 4.2, the proposed method with both semi-supervised learning and re-ranking leads to significant improvements compared with the best method (VGG+C+T+S) in [290]. The CMC curves of the proposed methods are shown in Figure 4.6 (a).

Moreover, experiments conducted with two popular metric learning methods, KISSME [123] and Cross-view Quadratic Discriminant Analysis (XQDA) [146] verify the effectiveness of our ranking optimization method on different distance metrics as shown in Table 4.4. In [123], the Mahalanobis distance is learned by considering the log likelihood ratio test of two Gaussian distributions. Based on the idea of KISSME [123], the XQDA further learns a discriminant subspaces with more efficient metrics.

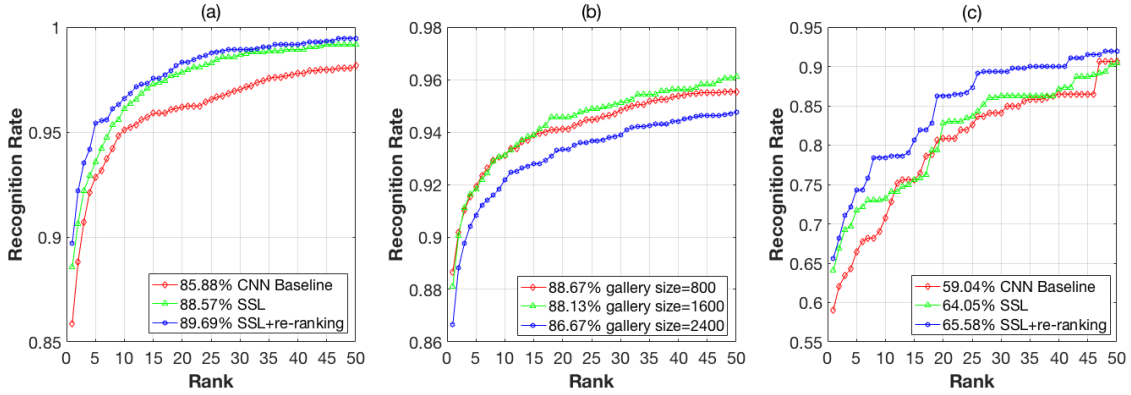


Figure 4.6: The CMC curves of the proposed methods on VeRi-776 (a), VehicleID (b), VehicleReID (c). The recognition rate shows the percentage of the probes that are correctly recognized within the top k matches in the gallery. The numbers in the legend of curves are the top 1 value of CMC.

Performance Comparisons on VehicleID Dataset We provide our results from the largest vehicle re-ID dataset [152] in Table 4.5 to further demonstrate the effectiveness of the proposed method. Following the dataset setting in [152], we randomly select one image from each vehicle and put it into gallery set, then the remaining images are all used as probe images. The details of the three testing subsets are listed in Table 4.6. We preform the testing process with different values of $k1$, $k2$ and λ, and obtain the best performance when $k1$=10, $k2$=6 and λ=0.3. The evaluation procedure was repeated for 10 times to evaluate model prediction accuracy and obtain the final CMC curve.

The detailed match rates from Rank-1 to Rank-50 of the proposed methods evaluated on the three scale test subset are presented in Figure 4.6 (b). For VehicleID dataset, we fine-tuned the improved ResNet-50 model by using the combination training set of original training set and 40,000 generated images. The vehicle re-ID results of our proposed method on three scale test subsets are shown in Table 4.5. Compared with the best state of the art method [290], the proposed method improves the Rank-1 and Rank-5 match rates for large subset by 2.44 points (from 84.23% to 86.67%) and 2.16 points (from 88.67% to 90.83%), respectively, which proves once again that our method has significant advantages. Four examples are shown in Figure 4.7. The proposed method, semi-supervised learning+re-ranking, effectively ranks more positive samples at the top of the ranking list which are not included in the ranking list of our baseline.

Table 4.5: Match rate (CMC@Rank-R, %) and mAP (%) of the comparison methods on the VehicleID dataset [152]

Methods		Small	Medium	Large
VGG+Triplet Loss [48]	Rank-1	40.40	35.40	31.90
VGG+CCL [152]		43.60	37.00	32.90
Mixed Diff+CCL [152]		49.00	42.80	38.20
VGG+C+T+S [290]		69.90	66.20	63.20
Baseline (Ours)		81.93	81.44	81.37
GAN+LSRO (Ours)		85.72	85.12	84.23
GAN+LSRO+ re-ranking (Ours)		**88.67**	**88.13**	**86.67**
VGG+Triplet Loss [48]	Rank-5	61.70	54.60	50.30
VGG+CCL [152]		64.20	57.10	53.30
Mixed Diff+CCL [152]		73.50	66.80	61.60
VGG+C+T+S [290]		87.30	82.30	79.40
CNN Baseline (Ours)		86.93	86.44	86.67
SSL (Ours)		89.12	88.12	88.67
SSL+re-ranking (Ours)		**91.92**	**91.81**	**90.83**
CNN Baseline (Ours)	mAP	70.13	66.67	65.47
SSL (Ours)		74.13	69.84	68.74
SSL+re-ranking (Ours)		**76.42**	**71.39**	**70.59**

Table 4.6: The three subset of testing set for the VehicleID Dataset [152]

Number of images	Small	Medium	Large
Gallery size	6,493	11,777	17,377
Probe size	800	1,600	2,400

Performance Comparisons on the VehicleReID Dataset Furthermore, we study the effectiveness of our method on the VehicleReID dataset by using the single shot setting. There are 423 vehicles from both camera view A and camera view B, for solving the vehicle re-ID task, we chosen this subset from the original sets. We randomly split the vehicles in both camera A and camera B into two almost equal subsets, where 211 vehicles for training and 212 vehicles for testing. Among the 212 vehicles for testing, we treat the images from camera A as the probe set and use the images from camera B as the gallery set. During the testing process, we search the 212 test vehicles in all vehicles from camera B.

We followed the semi-supervised learning method to fine-tune the CNN model as previously explained, and applied the ranking optimization algorithm for the final prediction.

Figure 4.7: Four examples of vehicle re-ID results (Rank-5) on the VehicleID dataset. For each probe, the ranking results produced by our baseline are presented in the first row, the second row corresponds to our proposed method (Semi-supervised learning+re-ranking) which improves the baseline ranking results. The green box indicates a true matches, the red box identifies the false matches.

Table 4.7: Match rate (CMC@Rank-R, %) and mAP (%) for the compared methods on the VehicleReID dataset [277]

Methods	Rank-1	Rank-5	Mean AP
CNN Baseline (Ours)	59.04	66.45	62.53
SSL (Ours)	64.05	72.56	66.64
SSL + re-ranking (Ours)	**65.58**	**74.29**	**70.12**

Specifically, the DCGAN was trained to generate unlabeled vehicle images, then we combined the generated images with original training set to fine-tune the improved ResNet-50 model. The ranking optimization was accomplished after the initial list generated by the Euclidean distance. We set the appropriate value to $k1$=6, $k2$=3 and λ=0.8. The testing phase is repeated for 10 times with the average results reported in Table 4.7. Our semi-supervised learning method gains 5.01 points improvement in Rank-1 match rate and significant 4.11 points improvement in mAP for CNN baseline. After applying the re-ranking algorithm, our method further gains an improvement of 1.53 points in Rank-1 match rate and 3.48 points in mAP. Experimental results demonstrate that our method is also effective on the re-ID problem of single-shot setting. Figure 4.6 shows the CMC curve on the VehicleReID dataset.

4.5 Further Evaluation

4.5.1 The Impact of the Scale of Random Vector Fed to the GAN.

The generator, G, used in GAN input a random noise vector z which passed through each layer in the network and generates a fake sample $G(z)$ from the final layer. We evaluate whether the scale of the random vector z fed to the GAN impacts the performance of vehicle re-ID. To investigate the effect, we tried three different ranges of the random vector, i.e., [-0.5,0.5], [-1,1], and [-1.5,1.5], with a normal distribution. The results of vehicle re-ID on the VeRi-776 dataset are presented in Table 4.8. We find that the [-0.5,0.5] yields higher re-ID performance than the other two ranges. The visual examples are shown in Figure 4.8. We find that visual examples of [-1.5,1.5] show obvious differences among the three ranges, with some strange shapes of vehicles. Typically, a larger range may contain some strange variations and affect the quality of generated images.

Figure 4.8: The GAN generated images with different scales of the random vector, i.e. [-0.5,0.5], [-1.0,1.0], [-1.5,1.5]. We hardly find any significant visual differences between them.

4.5.2 Analysis of the Parameters of Ranking Optimization Method

The parameters of ranking optimization method are evaluated in this sub-section. We observe the influence of $k1$, $k2$ and λ on the VeRi-776 dataset. Figure 4.9 (a)(b) show the impact of the size of k-reciprocal neighbors set on Rank-1 match rate and mAP. As $k1$ grows, the Rank-1 match rate first increases with fluctuations, and then starts a slow

Table 4.8: Match rate (CMC@Rank-R, %) and mAP (%) after using the GAN generated images with different scales of the random vector on the VeRi-776 dataset [158]

Random Range	Rank-1	Rank-5	Mean AP
[-0.5,0.5]	**89.65**	**95.41**	**68.97**
[-1,1]	89.46	95.12	68.46
[-1.5,1.5]	89.13	94.97	68.40

decrease after $k1$ passes the optimal point at around 50. Similarly, the mAP increases with the growth of $k1$, and it starts to slowly decline after $k1$ passes the optimal point. If $k1$ is too large, more false matches will be included in the k-reciprocal set and cause performance degradation.

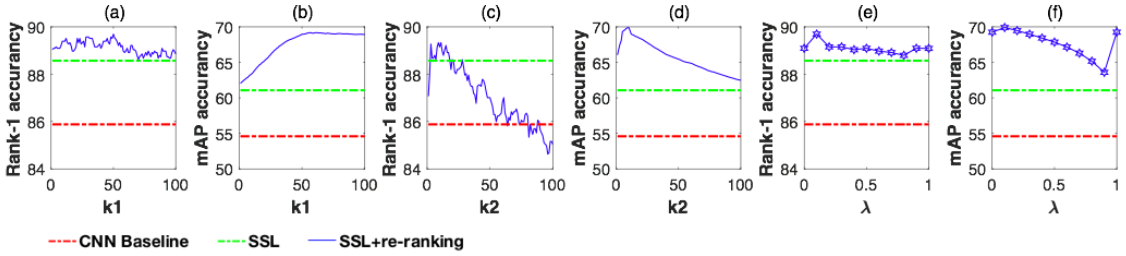


Figure 4.9: (a)(b): The impact of the parameter $k1$ on the performance of the VeRi-776 dataset. The $k2$ was fixed at 10 and λ set to 0.2; (c)(d): The impact of the parameter $k2$ on the performance of VeRi-776 dataset. The $k1$ was fixed at 50 and λ set to 0.2; (e)(f): The impact of the parameter λ on the performance of VeRi-776 dataset. The $k1$ was fixed at 50 and $k2$ at 10.

The impact of $k2$ is shown in Figure 4.9 (c)(d). Obviously, the performance will increase as $k2$ grows within a reasonable range (e.g, smaller than 10). However, the performance declines when the value of $k2$ is too large due to the set includes more false matches. In fact, it is very important to set an appropriate value to $k2$ and thus further enhance the performance.

Figure 4.9 (e)(f) show the impact of the parameter λ. The Jaccard distance is only considered when λ equals zero, in contrast, the Jaccard distance is left out when λ equals one, and the result is obtained using only the original distance. It can be observed that our method consistently outperforms the CNN baseline when the Jaccard distance is only

considered, which indicates that the proposed Jaccard distance is effective for re-ranking. Moreover, the performance is further improved when we consider the importance of original distance and set the value of λ arounds 0.2.

4.6 Conclusion

In this chapter, we proposed an effective semi-supervised learning approach augmented with ranking optimization for the vehicle re-ID problem. Specifically, a DCGAN model is exploited to generate the unlabelled images and effectively demonstrate their regularization ability when trained with an improved ResNet-50 baseline model. The unlabeled generated images are used to assist the labeled training images for simultaneous semi-supervised learning. We also addressed the re-ranking task by improving the k-reciprocal Nearest Neighbors method. The final distance based on the aggregation of the original distance and Jaccard distance produces effective improvement of the re-ID performance on VeRi-776, VehicleID and VehicleReID datasets. Our experimental results indicate that the proposed methods significantly outperforms state-of-the-arts methods on the VeRi-776 and VehicleID dataset. The proposed model has two sub-networks during the training process including a DCGAN and a CNN model with LSRO, which makes the end-to-end training impossible. In the future, we will extend the proposed method to the end-to-end network for vehicle re-ID. The advantage of end-to-end training is that we do not need supervision for individual sub-modules of the system.

Chapter 5

Face Recognition in Uncontrolled Environments

Recognizing people is one of the most important topics in computer vision and pattern recognition. Among various biometrics used for person recognition, the face is one of the most popular, since this ubiquitous biometric can be acquired in unconstrained environments while providing strong discriminative features for recognition. For this reason, face recognition became an extremely important tool that is used for video-surveillance and security systems, video-analytics software, and thousands of applications in our daily lives like entertainment, smart shopping, and automatic face tagging in photo collections.

For many applications, the performance of face recognition systems in controlled environments has now reached a satisfactory level; however, there are still many challenges posed by uncontrolled environments. Some of these challenges are posed by the problems caused by variations in disguise accessories, illumination, face pose, expression, and etc. To some extent, current state-of-the-art systems are able to cope with variability due to pose, illumination, expression, and size, which represent the challenges in unconstrained face recognition. In this chapter, we will address the face recognition problem under different variations, including disguise accessories, illumination and pose. For the disguised face recognition, we propose a framework based on the image-image translation method which collects a dataset of disguised faces that has the similar style to the target domain. We further improve the image-image translation method and apply it to near infrared face recognition to synthesize the visible light face images from near infrared inputs, thereby solving the illumination change problem. Compared with the variations of disguise

accessories and illumination, pose variations are more difficult to solve. Therefore, we propose a novel meta learning framework to learn face representations equivariant that is different from the disguised and near infrared face recognition methods.

5.1 Unsupervised Domain Adaptation for Disguised Face Recognition

5.1.1 Introduction

Within the past decades, face recognition (FR) has received a tremendous amount of attention owing to its wide range of potential applications, e.g., identity authentication, public security and surveillance. Many innovative and novel methods have been put forward for the tasks of visual face recognition and verification. Meanwhile, great challenges have been confronted by current FR systems, particularly when the accuracy significantly decreases while recognizing the same subjects with disguised appearances, such as wearing a wig or eyeglasses, changing hairstyle and so on [46].

Disguise usually involves intentional and unintentional changes on a face through which one can either impersonate or confuse someone's identity. Figure 5.1 clearly shows two examples of face obfuscation, in which the appearance of a subject can be varied by using different disguise accessories. To make automatic face recognition secure and usable, it is necessary to address the disguise problem. Current research in disguised face recognition (DFR) typically is based on a single-domain setting [45], [225]. Specifically, an algorithm first learns a CNN model from the training data, and then applies it to the test data. When the training data and testing data shares the same distribution, the learnt CNN model generally works well, since in this case the training error is an optimal estimate of the test error.

However, in real world applications, there is a need for transferring the learned knowledge from a source domain with abundant labeled data to a target domain where data is unlabeled or sparsely labeled. When CNN models trained on one domain and used on another domain with different distributions, the performance drops dramatically due to the domain bias [247]. To this end, we propose to solve the disguise face recognition task using domain adaptation [15], [250], which attempts to transfer the rich knowledge from the source domain, which is fully annotated, to another, different but related, domain to obtain a better CNN model.

Figure 5.1: Two samples of images with different disguise accessories.

Recently, attention transfer has been proposed and successfully adopted in several domain adaptation tasks [276], [142], which attempts to transfer attention knowledge from a powerful deeper network that is trained with sufficient training samples to a shallower network that can be trained with limited training data with the goal of improving the performance of the latter. However, it is still challenging to train such a high-quality cross-domain model for the DFR due to the large domain shift in the images. To deal with the large domain shift between source domain and target domain for the DFR, we can adopt the data in source domain to synthesize disguised face images as similar as the data in target domain by using GAN model, which has been proven to generate impressively realistic faces through a two-player game between a generator and a discriminator. For the GAN model, there are many promising image-to-image translation developments [155], [299], but they do not necessarily preserve the identity label of an image. Although the generated image may "look" like that it comes from the auxiliary domain, the underlying identity may be lost after image-image translation. Consequently, the desired model for our task is that it can generate disguised face images which should simultaneously preserve the identity label in source domain and transform helpful content information in target domain.

Inspired by the above discussions, we propose a novel Unsupervised Domain Adaptation Model (UDAM), which jointly transfer the rich knowledge from the source domain and discriminative representation end-to-end that mutually boost each other to achieve the disguised face recognition of target domain. In particular, UDAM includes a Domain Style Adaptation subNet (DSN) and a Attention Learning subNet (ALN) to learn the

representations. The DSN introduces unsupervised cross-domain adversarial training and a "learning to learn" strategy with the Siamese discriminator to achieve stronger generalizability and high-fidelity, underlying identity preserving face generation. In this setting, the model can satisfy the specific requirement of retaining identity information after image-image translation in the disguised face recognition, and we are able to create a dataset which has the similar style of the target domain in an unsupervised manner. ALN is a CNN for disguised face recognition with our proposed attention transfer strategy. The CNN model is trained by taking advantage of the sufficient labeled generated images, unlike previous approaches that distill knowledge through class probabilities [92], we propose to learn class-specific energy functions on spatial attention map, which is helpful to obtain an effective CNN model that less affected from the domain shift.

Our contributions can be summarized as follows:

- We present a deep architecture unifying image-image translation and disguised face recognition in a mutual boosting way, which inherits the merits of existing domain bias disguised face recognition methods. The proposed model achieves consistent improvement on both controlled and in-the-wild datasets.
- The local and global structural consistency of the style-translated disguised face images has been effectively enforced through pixel cycle-consistency and discriminative loss. Besides, the class-discriminative spatial attention maps from the CNN model trained by source domain are leveraged to boost the performance of disguised face recognition in target domain.

5.1.2 Proposed Method

Problem Definition

Suppose a labeled dataset A, is used to train a CNN model M_c of disguised face recognition. If the trained M_c is directly applied to a target unlabbeled dataset B collected from an entirely different domain with a different set of identities/classes, the model tends to have poor performance, due to the significant differences between A and B. Therefore, we attempt to learn an optimal CNN model for B using knowledge transferred from A.

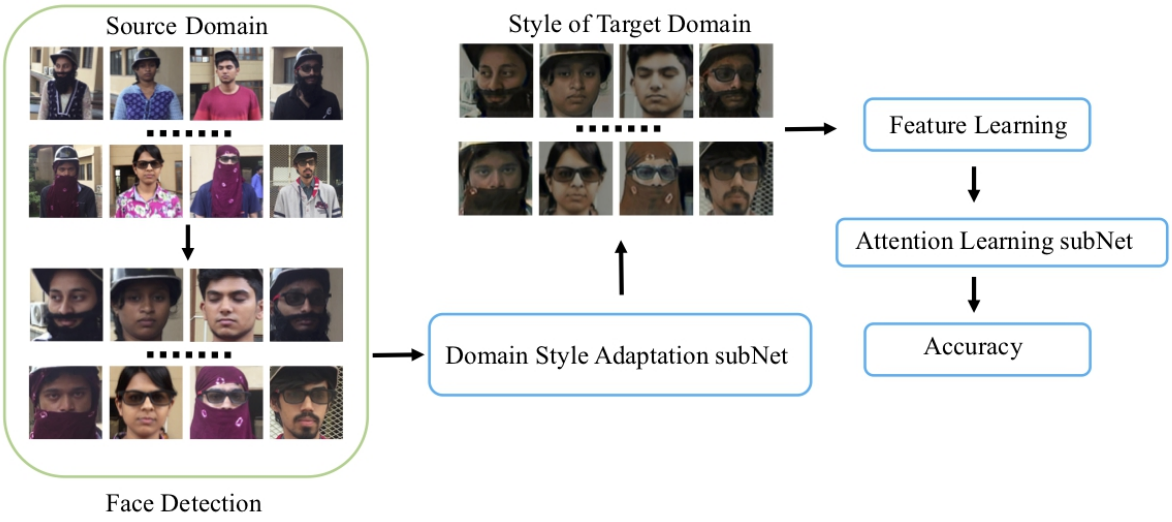


Figure 5.2: UDAM for disguised face recognition. First, we predict face and landmark location by MTCNN [286], then the Domain Style Adaptation subNet translates the style of the labeled images from a source dataset to the style of the target dataset. Finally, we train the CNN model with the translated images and use Attention Learning subNet to obtain the disguised face recognition.

UDAM

As shown in Figure 5.2, the proposed UDAM consists of a DSN and an ALN that jointly generate the domain-aware data and learn the disguised face representation end-to-end. We now present each component in detail.

DSN. We first introduce a mapping function G from source domain A to target domain B and train it to produce images that fool an adversarial discriminator D_B. Conversely, the adversarial discriminator attempts to classify the real target data from the source generated data. This corresponds to the loss function:

$$\begin{aligned}\mathcal{L}_{B_{adv}}(G, D_B, P_x, P_y) =& E_{y\sim p_y}[(D_B(y)-1)^2]+ \\ & E_{x\sim p_x}[(D_B(G(x))^2],\end{aligned} \tag{5.1}$$

where p_x and p_y denote the sample distributions in the source and target domain, respectively. However, with large enough capacity, a network can map the face images in the source domain to any random permutation of images in the target domain. As a result, it is undesirable in the DFR task, where we have to ensure the quality of the generated

faces. Thus, we introduce another mapping F from target to source and train it according to the same GAN loss, i.e.,

$$\begin{aligned}\mathcal{L}_{A_{adv}}(F, D_A, P_y, P_x) =& E_{x\sim p_x}[(D_A(x)-1)^2]+ \\ & E_{y\sim p_y}[(D_A(F(y))^2],\end{aligned} \tag{5.2}$$

We then introduce a cycle-consistency loss [299] to recover the original image after a cycle of translation and reverse translation, thereby enforcing cycle-consistency and preserving local structural information of the face images in source domain. The cycle-consistent loss can be expressed as:

$$\begin{aligned}\mathcal{L}_{cyc}(G, F) =& E_{x\sim p_x}[||F(G(x))-x||_1]+ \\ & E_{y\sim p_y}[||G(F(y))-y||_1],\end{aligned} \tag{5.3}$$

To encourage the domain style adaptation to preserve the identity information for each translated image, inspired by [43], we add the contrastive loss [86] in the cycle-consistency loss function to learn a latent space that constrains the learning of the mapping function. We use the contrastive loss [86] to train the Siamese network as follows:

$$\mathcal{L}_{con}(l, i_1, i_2) = (1-l)\left\{max(0, m-d)\right\}^2 + ld^2, \tag{5.4}$$

where i_1 and i_2 are a pair of input vectors, which are selected in an unsupervised manner. d denotes the Euclidean distance between normalized embeddings of two input vectors, and l represents the binary label of the pair. If i_1 and i_2 are positive image pair, l equals one. On the contrary, if i_1 and i_2 are negative image pair, l equals zero. $m \in [0, 2]$ represents the margin that defines the separability in the embedding space. The loss of the negative training pair is not back-propagated in the system when m equals zero. Both positive and negative sample pairs are considered if m is larger than zero. A larger m means that the loss of negative training samples has a higher weight in back propagation.

Based on the prior knowledge that the set of ID information is different in the source and target domains, there are two types of negative training pairs designed for generators G and F: 1) $G(i_A)$ and i_B, 2) $F(i_B)$ and i_A. Thus, a translated image should be of different ID information from any target image. Accordingly, the two dissimilar images are pushed away by the network. Taken together, the final Domain Style Adaptation subNet objective can be written as in Equation 5.5 by considering Equations 5.1, 5.2, 5.3, and 5.4:

$$\mathcal{L}_{sum} = \mathcal{L}_{B_{adv}} + \mathcal{L}_{A_{adv}} + \mathcal{L}_{cyc} + \mathcal{L}_{con} \tag{5.5}$$

ALN. Given that the style-translated dataset consisting of the translated images and their associated labels, the ResNet-50 [89] model is slightly improved and used in our experiments as the base network. It is pre-trained on the ImageNet [205] dataset, and fine-tuned on the translated images to classify the training identities. We discard the last 1000-dimensional classification layer and add two FC layers. Besides, to reduce the possibility of overfitting, a dropout layer [231] has been inserted before the final convolutional layer. The last FC layer is modified to have N neurons to predict the N-classes, where N is the number of the classes in the training set.

Once we obtain the CNN model for the style-translated dataset, we can further address the domain shift problem by using spatial attention map to exploit features from the convolutional layer. Class information and more general convolutional feature are incorporated through attention map, hence more transitions can be made across domains. Let $n \in (1, 2, ..., N)$ be the n-th pre-defined class of the real images in the target domain, where N is the number of classes. For a particular example x with single ground-truth label y, the last convolutional layer of the trained CNN model will produce K feature maps A^k. The image x is first forwardly propagated through the trained CNN model, then we adopt the Grad-CAM [217] to generate the spatial attention map $\mathcal{L}(x, y_n)$ by a weighted combination of the convolutional feature maps,

$$\mathcal{L}(x, y_n) = ReLU(\sum_x \alpha_k^{y_n} A^k) \tag{5.6}$$

The importance of the k-th feature map for the prediction class y_n will be captured by the weight $\alpha_k^{y_n}$ through calculating the back propagating gradients to the convolutional feature map A_k. For the spatial attention map of each image, an energy function has been defined as $\frac{E(\mathcal{L}(x,y_n)}{\sum_{n=1}^{N} E(\mathcal{L}(x,y_n))}$, which is the largest when $y = y_n$, and smaller otherwise. We define E based on a simple yet effective observation: Assuming that the CNN model has been pre-trained on the style-translated source domain to predict certain identity, given an image and its spatial attention map corresponding to an identity, if the facial attribute of the identity exists in the certain region, the attention map will generate the higher activations in the corresponding region. Therefore, a sliding window with size of 4×4 and step size of 1 will be applied over $\mathcal{L}(x, y_n)$. Then we calculate the sum of the value of $\mathcal{L}(x, y_n)$ within each sliding window as the local activation. We use the energy E to express the maximum of all local activations. For the target domain with N classes, we calculate the output score over each label as the mean energy across all local activations,

$$score(x, y_n) = \frac{1}{N} \sum_{C} E(\mathcal{L}(x, y_n)), \tag{5.7}$$

where C denotes the number of local activations. We infer the one with highest score as the predicted label,

$$y_p = \underset{y_n}{argmax} \;\; score(x, y_n) \tag{5.8}$$

5.1.3 Experiment

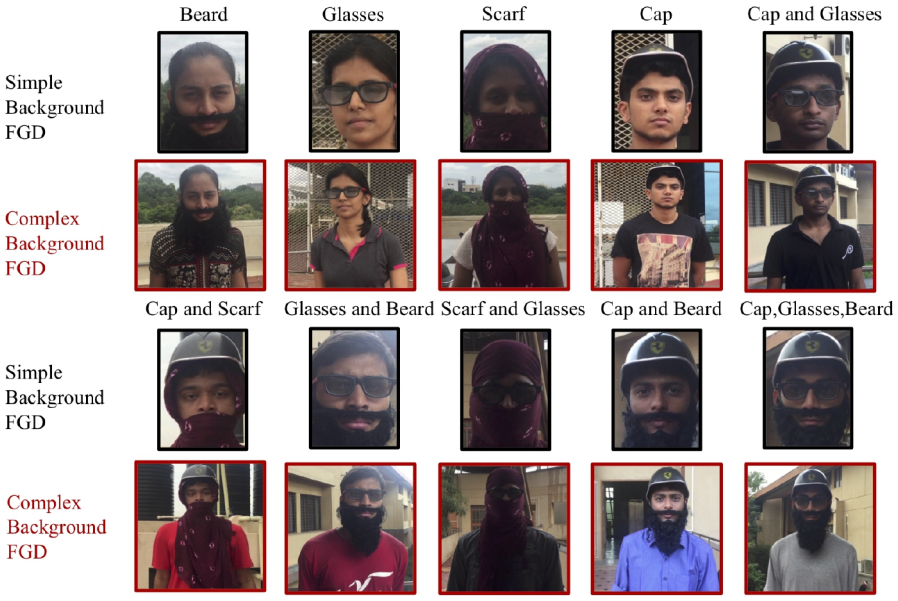


Figure 5.3: The illustration shows samples images with different disguises from both the Simple and Complex face disguise (FG) datasets.

Datasets

The Simple and Complex Face Disguise Dataset [225] contain 2000 images of 25 people with 10 different disguises varied each with (i) Simple and (ii) Complex backgrounds that contain people with 8 different background illuminations in the wild. The dataset is split into three fixed parts: 1000 training images, 500 validation images and 500 test images. The example images from each dataset are shown in Figure 5.4. We can observe that

Figure 5.4: Sample images from the IIIT-Delhi Disguise Version 1 Face Database (ID V1 Database).

the samples from the complex background dataset have a relatively complex background compared to the simple dataset.

The IIIT-Delhi Disguise Version 1 Face Database (ID V1 Database) [46] contains 681 visible spectrum images of 75 participants with disguise variations. The dataset is randomly divided into a training set with 35 subjects and a testing set with the remaining 40 subjects. All the face images are almost taken under constant illumination with neutral expression and frontal pose. The sample images from the database are shown in Figure **??**.

Implementation Details

Domain Style Adaptation model. We used Tensorflow [1] to train Domain Style Adaptation subNet using the training images of the dataset. Before the training process, we apply the MTCNN [286] to perform face detection for datasets and reduce the negative affect of the background. With an initial learning rate of 0.0002, and model stops training after 7 epochs. During the testing procedure, we employ the Generator G for Simple and Complex FGD $\rightarrow$ ID V1 Database translation and the Generative F for ID V1 Database $\rightarrow$ Simple and Complex FGD translation. The translated images are used to fine-tune the CNN model.

Feature learning. Specifically, ResNet-50 [89] pre-trained on ImageNet [205] is used for fine-tuning on the translated images. We modify the output of the last fully-connected layer to 25 and 35 for Simple and Complex FGD and ID V1 Database, respectively. A mini-batch SGD is used to train the CNN model on a GTX 1080 GPU. The initial learning

rate is set to 0.001, and decays to 0.0001 after 10 epochs. The trained CNN is then used to generate spatial attention maps for test images in target domain. We set the size of the attention map for ResNet-50 is 7×7.

Experiment results and Evaluation

To help analyze our model and show the benefit of each module, we design several unsupervised comparison methods as follows:
Setting-1: Source domain to target domain (S2T). This baseline uses the disguised face images in source domain to fine-tune the pre-trained CNN model and then tests it on target domain.
Setting-2: S2T_ DSN(without contrastive loss). We first train the DSN (without contrastive loss) using the source domain, and the generated disguised face images are used to train the CNN model.
Setting-3: S2T_ DSN. This baseline preserves the identity information for each translated image by adding contrastive loss to setting 2.
Setting-4: S2T_ UDAM(DSNALN). Proposed unsupervised domain adaptation method in this paper.

We first evaluated our method on the Simple and Complex Face Disguise Dataset, which is a disguised face dataset in the wild with varied disguises, covering different backgrounds and under varied illuminations. We translated the image style of ID V1 Database (source domain) to Simple and Complex Face Disguise Dataset (target domain) and then use the translated images to train the disguised face recognition model. Finally, we evaluated the methods on the test set of Simple and Complex Face Disguise Dataset.

Table 5.1 shows the detailed comparison results between our methods and three aforementioned baseline methods. The proposed method outperforms all the corresponding baselines with 8% to 12.6% improvement and 7.2% and 14.7% on the DFR accuracy for simple and complex version, respectively. We attribute this to the image generator and attention learning strategy in our method. Based on the results in Table 5.1, it is clear that S2T_ DSN(without contrastive loss) can achieve better performance with the S2T baseline, demonstrating its efficacy to transfer style across domains. With the help of contrastive loss, we preserve the identity information during the image translation process leading to 3% and 5.1% improvement over the Setting-2 for simple and complex version, respectively. Examples of translated images by DSN are shown in Figure 5.5.

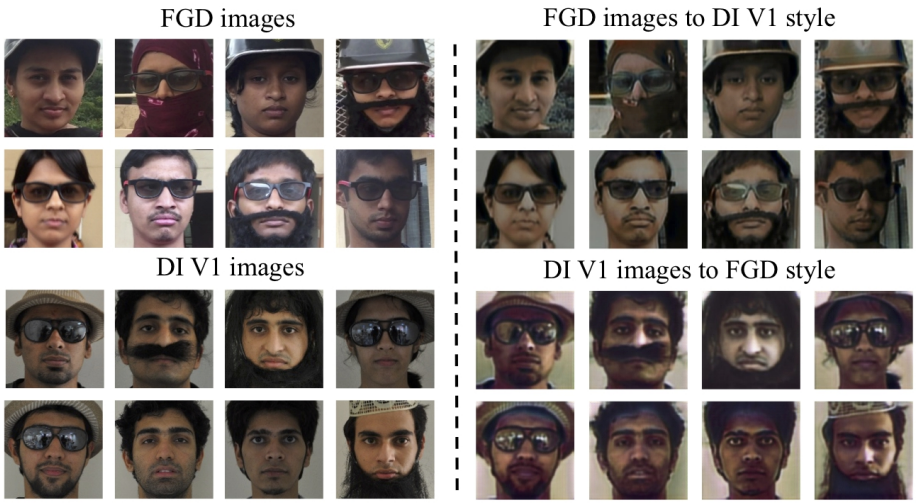


Figure 5.5: Upper right: FGD images which are translated to ID V1 style; Lower right: ID V1 images translated to FGD style.

Table 5.1: Face disguise classification accuracy (%) of our four unsupervised comparative settings on the Simple and Complex Face Disguise Dataset.

Method	Simple FGD	Complex FGD
S2T	54.6%	51.4%
S2T_ DSN (without $\mathcal{L}_{cyc}$)	56.2%	53.8%
S2T_ DSN	59.2%	58.9%
S2T_ UDAM (DSN&ALN)	**67.2**%	**66.1**%

Since all of the previous approaches are not unsupervised learning setting, we compared our method with the state-of-the-art supervised learning methods including DFI [225] and ITE [46] in Table 5.4. For complex FGD, we arrive at an accuracy = 66.1%, which is +3.5% higher than the best results in [225]. Compared with the second best method, ITE [46], our unsupervised domain adaptation method is +1.7% and +12.7% higher in accuracy for Simple and Complex FGD, respectively. The comparisons indicate the competitiveness of the proposed method on the simple and complex FG dataset.

To further test the effectiveness of our method, we treated the Simple and Complex Face Disguise Dataset and ID V1 Database as source domain and target domain, respectively. In Table 5.3, we show the face recognition performance comparison of our method with some

Table 5.2: Comparison with state-of-the-art methods on Simple and Complex Disguised Face Dataset.

Method	Simple FGD	Complex FGD
DFI [225]	**78.4**%	62.6%
ITE [46]	65.2%	53.4%
S2T_ UDAM (DSN&ALN)	67.5%	**66.1**%

Table 5.3: Face disguise classification accuracy (%) on the IIIT-Delhi Disguise Version 1 Face Database (ID V1 Database).

Method	ID VI Database
NoImage+ResNet	41.3%
S2T	29.7%
S2T_ DSN (without $\mathcal{L}_{cyc}$)	35.8%
S2T_ DSN	39.2%
S2T_ UDAM (DSN&ALN)	**45.2**%

baselines. There are several findings from the results. Firstly, the recognition accuracy shown in the last column of this table indicates that the proposed model drastically improve the performance, and the degree of improvement varies between 6% and 15.5%. This well verifies the proposed method is effective when the data in the target domain is limited and unlabeled, which is the general scenario for unsupervised domain adaptation problems. Moreover, the joint learning scheme of domain style adaptation and attention transfer learning also helps, since the two sub-nets leverage each other during end-to-end training to achieve a final win-win outcome.

We can not find existing methods that conduct experiments on this dataset under the same conditions with us. Thus we directly create baseline NoImage+ResNet, where we directly use the training set of ID V1 to fine-tune a ResNet-50 model. Table 5.3 shows our methods can achieve better recognition accuracy of 45.2%.

5.1.4 Conclusion

In this paper, we proposed a novel UDAM to address the challenging face recognition with domain bias. UDAM unifies a Domain Style Adaptation subNet (DSN) and a Attention Learning subNet (ALN) for disguised face recognition in an end-to-end deep

architecture. The DSN introduces unsupervised cross-domain adversarial training to provide style-translated images for effective attention transfer learning from ALN. Besides, the underlying (latent) ID information for the disguised face images also has been preserved after image-image translation. We conducted experiments on the Simple and Complex FGD and ID V1 Database, and shown the efficacy of the proposed method to adapt the domain shift problem, especially when the images in the target domain is unlabeled.

5.2 Image-Image Translation to Enhance Near Infrared Face Recognition

5.2.1 Introduction

Facial recognition has become one of the most active research areas in the field of computer vision due to its potential value for many applications such as security systems and surveillance. Despite significant progress in this domain, illumination has been regraded as one of the most significant impact factors in face recognition [107]. In this paper, we focus on NIR face recognition, which has the features of being insensitive to illumination changes and can perform well even in near darkness [198]. Many algorithms have been proposed to recognize faces in NIR images in recent years [282], [58]. A common drawback of all these methods is that they exploited hand-crafted features without applying a deep, global representation of the facial images, which has been shown to produce superior results for face recognition.

Our work is motivated by two recent developments. Firstly, the existing visible light domain (VLD) face recognition systems have achieved impressive performance [215], [258], owning to the development of deep networks and large face datasets [100], [85]. For example, in [215], 200 million images captured from 8 million subjects were used to train a deep network which achieve the best performance on a standard unconstrained face recognition benchmark called Labeled Faces in the Wild (LFW) [100]. With these significant advances, existing VLD-based face recognition systems should be extended to other research areas which are less well studied, such as near-infrared imaging (low-light). Unfortunately, due to the relatively small amount of training data available and the domain bias between near-infrared and visible light, the same success in VLD is not easily replicated in the near-infrared domain. This observation inspired us to resort to synthesize visible light face images from NIR inputs, which solves the illumination change problem and, at the same

time, is able to work with pre-existing face recognition systems.

Secondly, with the development of the GAN method, the community has made significant progress in solving image-image translation problems [154], [299]. Recently, SPGAN [44] introduced the Siamese network based on the CycleGAN framework in [299] to learn the image translation between two different domains and preserve the identity information of the person. However, these GAN based methods require sufficient input-output image pairs for training, which is not available for the near infrared domain with limited samples. To address this problem, it is important to introduce datasets that include sufficient near infrared and visible light image pairs.

Considering the above two issues, we makes two contributions. The first contribution is the creation of two new NIR datasets, named the "Outdoor NIR-VIS Face (ONVF) database" and "Indoor NIR Face (INF) database". The ONVF dataset contains 30,000 image pairs of 1,000 identities collected by a visible light camera and a near-infrared camera. The INF dataset consists of 470 near-infrared images which belong to 94 people captured by a near-infrared camera. The the details of the databases are described in Section 5.2.3.

As a second contribution, we propose a novel near infrared facial recognition method. To start, a Multi-Task Cascaded Convolutional Network (MTCNN) [286] is applied to achieve face detection and alignment, which is useful for handling background and occlusion variations in images. Then, the ONVF dataset is used to train a NIR-VIS image translation model that translates the near infrared face image to a visible light face image. After the translation, the generated VLD face is fed into an existing pre-trained VLD deep neural network face recognition model [215]. The intention is that better recognition results can be obtained without retraining or changing the VLD model. The framework of our proposed method is illustrated in Figure 5.6. The experimental results on the INF and CSIST [267] databases confirm that the proposed method achieves a favorable performance compared with published state-of-the-art methods [16], [148].

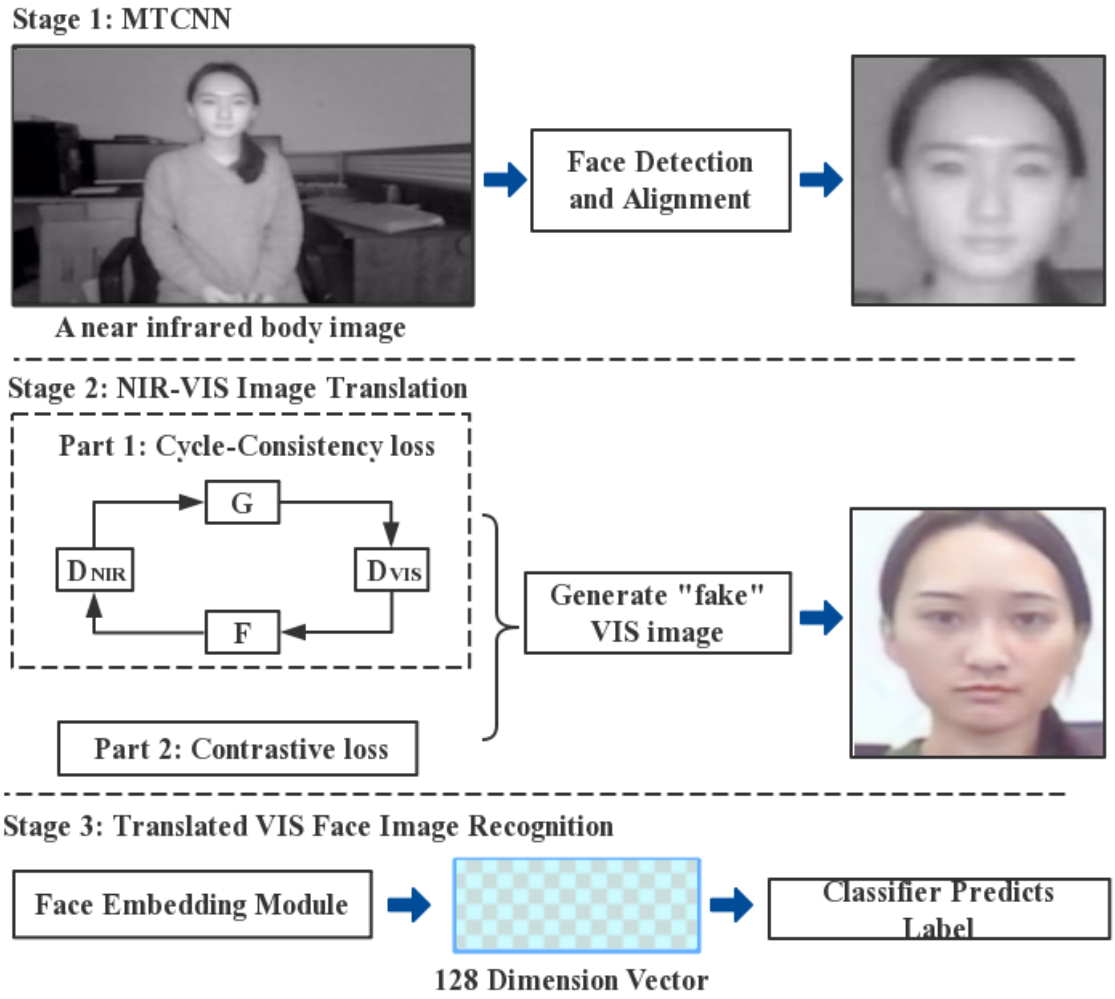


Figure 5.6: The workflow of the proposed approach. There are three stages: (1) Face detection and alignment using MTCNN [286]; (2) NIR-VIS Image translation by combining the cycle-consistency loss and contrastive loss; (3) Translated VIS face image recognition which obtains a 128 dimension feature vector for each translated VIS face image in the test set. Finally, the predict label will be output by the classifier.

5.2.2 Proposed Method

In this section, the proposed method is described, which consists of three steps: 1) Face detection and alignment, 2) NIR-VIS image translation, 3) Face embedding and

classification.

Face Detection and Alignment

Zhang et al. [286] proposed the MTCNN framework that exploits the inherent correlation between detection and alignment to improve their performance. In our work, we apply the MTCNN network to implement a face detection model on the ONVF dataset and CSIST. It essentially consists of three stages: (1) We exploit a list of candidate windows which are generated by the proposed network (P-Net) to classify the face and non-face and estimate the bounding box regression vector as the face position. (2) A large number of wrong candidates will be rejected by feeding all the candidates to a Refining Network (R-Net). (3) Another CNN, called O-Net, outputs the five facial landmarks.

NIR-VIS Image Translation

The goal of NIR-VIS image translation is to train a generator G that can transform the near infrared image X into its corresponding visible image Y, where the visible image Y contains sufficient identity information for the facial recognition task. To this end, we utilised image-image translation methods which aim at learning a mapping function between the two domains. Conditional GAN [105] is a representative method by using paired training data to produce impressive transition results. However, it is difficult to obtain sufficient paired training data in the real world. In [299], this framework has been extended to unsupervised image-to-image translation, meaning there is no requirement for image pairs. In our work, we applied the CycleGAN framework [299] to transform a NIR image to a VIS image. Two generator-discriminator pairs are introduced, G, D_{NIR} and F, D_{VIS} which map a sample from the NIR domain to the VIS domain and produce a sample that is indistinguishable from those in the VIS domain. For generator G and its associated discriminator D_{VIS}, we express the adversarial loss as

$$\begin{aligned}\mathcal{L}_{VIS_{adv}}(G, D_{VIS}, P_x, P_y) =& E_{y\sim p_y}[(D_{VIS}(y)-1)^2]+ \\ & E_{x\sim p_x}[(D_{VIS}(G(x))^2],\end{aligned} \tag{5.9}$$

where p_x and p_y denote the sample distributions in the NIR and VIS domains, respectively. For generator F and its associated discriminator D_{VIS} , the adversarial loss

is

$$\begin{aligned}\mathcal{L}_{NIS_{adv}}(F, D_A, P_y, P_x) =& E_{x\sim p_x}[(D_A(x)-1)^2]+ \\ & E_{y\sim p_y}[(D_A(F(y))^2],\end{aligned} \tag{5.10}$$

Due to the lack of paired training data, there exists multiple alternative mapping functions. To restore the original image after a cycle of translation and reverse translation, we then introduced a cycle-consistency loss [299] as:

$$\begin{aligned}\mathcal{L}_{cyc}(G, F) =& E_{x\sim p_x}[||F(G(x))-x||_1]+ \\ & E_{y\sim p_y}[||G(F(y))-y||_1],\end{aligned} \tag{5.11}$$

Similarity preservation is an important principle to exploit synthesised images generation from some GAN-based image-image translation schemes. In our work, additional constraints have been set on the mapping function to meet this special requirement for face image generation. Specifically, inspired by the success in [44], we add the contrastive loss [86] in the cycle-consistency loss function to learn a latent space that constrains the learning of the mapping function.

$$\mathcal{L}_{con}(l, i_1, i_2) = (1-l)\left\{max(0, m-d)\right\}^2 + ld^2, \tag{5.12}$$

where i_1 and i_2 are a pair of input vectors, which are selected in an unsupervised manner. d denotes the cosine distance between the normalized embedding of two input vectors and l represents the binary label of the pair. If i_1 and i_2 are a positive image pair, l equals one. On the contrary, if i_1 and i_2 are a negative image pair, l equals zero. Suppose two samples denoted as x_{NIR} and x_{VIS} come from the NIR domain and VIS domain, respectively. We define two positive pairs: 1) x_{NIR} and $G(x_{NIR})$, 2) x_{VIS} and $F(x_{VIS})$. The positive image pairs contain the same person, the only difference is that they have different styles (NIR or VIS). In the learning procedure, we encourage the whole network to pull these two images close. There are also two types of negative training pairs designed for generators G and F: 1) $G(x_{NIR})$ and x_{VIS}, 2) $F(x_{VIS})$ and x_{NIR}.

The separability in the embedding space has been represented as $m \in [0, 2]$. When m equals zero, there is no back-propagation for the negative training pair. If m is larger than zero, the system will consider the loss of both the positive and negative sample pairs. A larger m means that the loss of negative training samples have a higher weight for the back propagation. Taken together, the final NIR-VIS translation objective can be written

as in Equation 5.13 by considering Equations 5.9, 5.10, 5.11, and 5.12:

$$\mathcal{L}_{sum} = \mathcal{L}_{B_{adv}} + \mathcal{L}_{A_{adv}} + \mathcal{L}_{cyc} + \mathcal{L}_{con} \tag{5.13}$$

Face Embedding Module

A high-performance facial embedding module is critical to the entire facial recognition system. In this paper, our face embedding module is based on FaceNet [215] which is a deep metric learning network that uses two different CNN structures, [239] and [281]. We use the Inception-ResNet-v1 model, which achieves similar precision but with fewer parameters and lower computational complexity. First, the face-embedded module has been considered as a black box (Figure 5.7), and the Inception-ResNet-v1 model is the most important part of this end-to-end system. There is a batch input layer and deep CNN (Inception-ResNet-v1) in our network, which is then followed by L2 standardized for face embedding. The training network is then trained through the triples loss [239]. The basic idea is that the distance between the vectors of facial images from the same person is very small.

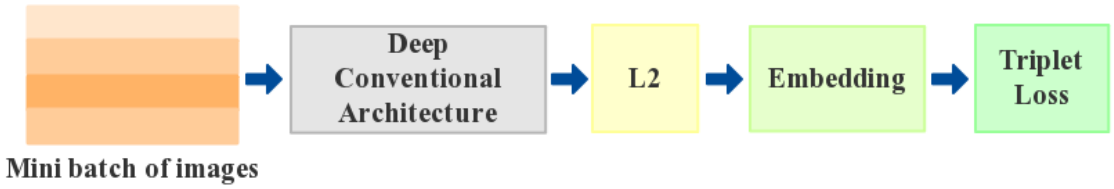


Figure 5.7: The Face Embedding Module.

5.2.3 Experiments

Datasets Introduction

ONVF database. The database was collected by our research team in the summer session of 2018 and the procedure lasted for several days. We captured the facial images from 1,000 subjects without constraints on the illumination and pose, each subject providing about 30 NIR and 30 VIS face images. There are different variations in the sample images such as pose, expression and focus, etc. During the process of collecting the database, we use JAI cameras with 1/2.7inch HM2131 image sensor which is sensitive to the NIR band.

The active light source was in the NIR spectrum between 780nm - 1,100nm and it was mounted on the camera. We applied the full ONVF database to train a NIR-VIS image translation model.

INF database. The INF database consists of 94 students from the University, including 57 male and 37 females. During the recording, a subject was asked to sit in front of the camera and their normal frontal face images were collected. The camera-face distance was between 80-120 cm, a convenient range for the user. There are 5 NIR face images per subject at a resolution of 640×480 pixels. In our experiments, we randomly selected 235 images for training and 235 images for testing.

CSIST database. There are two image sets in the CSIST database [267]: Lab1 and Lab2. Lab1 consists of 500 NIR images and 500 visible images captured from 50 subjects. In Lab2, 1000 NIR images and 1000 visible images are collected from 50 subjects under different illumination conditions. The image sizes for the Lab1 and Lab2 databases is 100×80 pixels. We randomly selected 50% of the database images for training and 50% for testing.

Implementation Details

Firstly, we implemented the facial detection and alignment using MTCNN [286], the facial images generated by the face pre-processing are 640×640 pixels. Then, we use the large ONVF database to train our NIR-VIS image translation model in Tensorflow [1]. During training, we set the initial learning rate and training epoch as 0.0002 and 7, respectively. During the testing procedure, we employed the trained NIR-VIS image translation model to translate the NIR face images in the INF database and CSIST database to visible facial images. Examples of images translated by NIR-VIS image translation are shown in Figure 5.8. Also, to carry out comparative studies, we also trained the CycleGAN using the same settings.

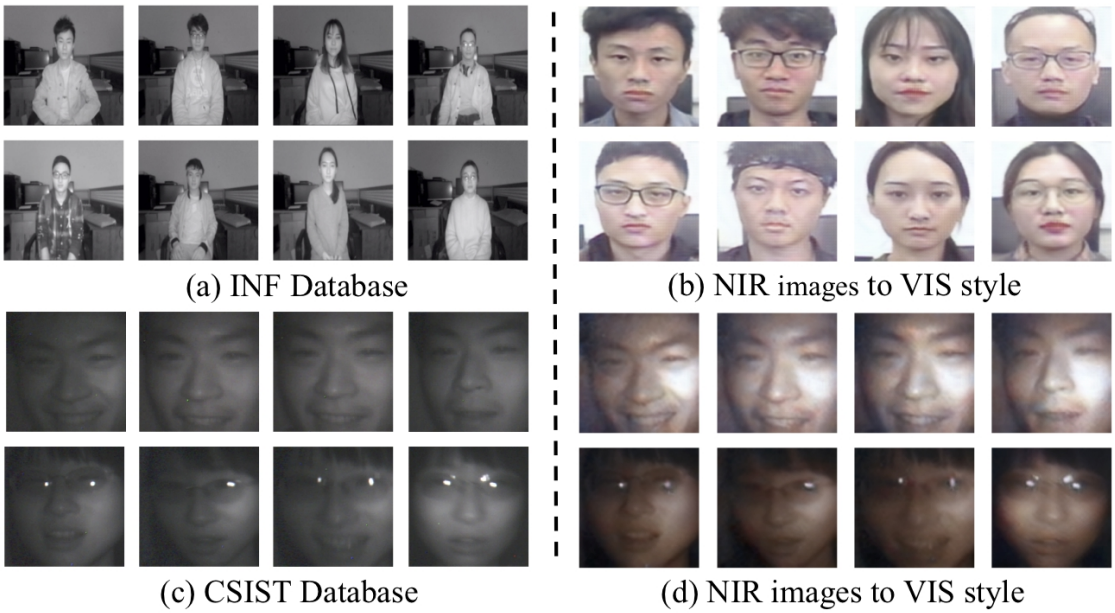


Figure 5.8: Example images translated from NIR to VIS.

For the Face Embedding Module, we extracted an image-level CNN based on the Inception-ResNet-v1 network [238] which was trained on a subset of the MSCeleb-1M database [85] and validated on the LFW database [100]. The model's architecture follows the Inception-ResNet-v1 network [238]. The input image size of the Inception-ResNet-v1 model is 160×160 pixels.

5.2.4 Results and Discussion

To help analyze our model and show the benefit of each module, we designed three baselines as follows:

Plain Near Infrared. No transformation was applied on this baseline. This baseline will indicate the effect of the domain gap between NIR and VIS on the face recognition models trained solely using VIS images.

CycleGAN. We first train the CycleGAN (without contrastive loss) using the ONSF Database, and the generated visual face images are fed into the face embedding model.

MTCNN+CycleGAN. Before training the CycleGAN, we added the face detection and alignment step by using MTCNN.

Table 5.4: NIR face image recognition accuracy (%) on the INF database

Method	INF database
Plain Near Infrared	79.8
CycleGAN	34.9
MTCNN+CycleGAN	97.5
Proposed method	**99.8**

Table 5.5: NIR face image recognition accuracy (%) on the CSIST database.

Method	Lab 1	Lab 2
SMRSN [16]	-	86.4
Score-level fusion [148]	-	88.63
Plain Near Infrared (ours)	65.2	77.2
CycleGAN (ours)	45.4	38.6
MTCNN+CycleGAN (ours)	94.2	86.3
Proposed method (ours)	**99.6**	**90.7**

The results on the INF dataset can be seen in Table 5.4. As can be seen from these results the proposed method outperforms all the baselines. Also, it is noteworthy to mention that the MTCNN leads to a 62.6% improvement which indicates that the face detection and alignment is able to further improve the system performance. The improvement brought by the contrastive loss is also validated on this dataset, with a 2.3% improvement over the second baseline.

We then recorded the recognition accuracy of our methods on the CSIST database as shown in Table 5.5. The MTCNN+CycleGAN method has an Average Precision (AP) value of 94.2% and 86.3% for Lab1 and Lab2, respectively, which is higher than the Plain Near Infrared and CycleGAN performance. This highlights the importance of reducing the domain bias that exists between the NIR and VIS images. With the help of contrastive loss, we preserve the identity information during the image translation process leading to a 5.4% and 4.4% improvement over the second baseline for Lab1 and Lab2 versions respectively.

The following observations can be made: (1) For the CSIST database, our method outperformed most of the previous methods which are only based on the near infrared domain. (2) The NIR-VIS image translation model is more effective at training a generator that can preserve the personal identity in the generated images. (3) The proposed model

shows good potential to achieve better results. Future work can be undertaken by fine-tuning the face embedding model on a specific dataset.

5.2.5 Conclusion

In this paper, we introduced the large ONSF database which includes various changes in pose, expressions and focus. We also created another INF database in the laboratory environment to test the performance of near infrared facial recognition. We propose a novel near infrared facial recognition method in an end-to-end deep architecture which includes face detection and alignment, NIR-VIS image translation and a face embedding module. This is the first time that it has been proposed to apply the image-image translation method to enhance the performance of near-infrared facial image recognition. This is achieved by synthesizing a virtual sample from an input near infrared face image. Using this approach, we reduce the intra-personal difference caused by the completely different illumination. Therefore, we can achieve much better recognition results by applying the existing pre-trained VLD deep neural network face recognition model. The proposed method was tested on the INF database and the CSIST dataset, with promising results.

5.3 Pose-robust Face Recognition by Deep Meta Capsule Network-based Equivariant Embedding

5.3.1 Introduction

Face recognition (FR) is one of the most actively researched topics in computer vision owing to its wide range of potential applications. Recently, many innovative methods based on deep learning have been put forward for face recognition and verification [234], [215], and the accuracy of recognizing clear human faces in well-controlled environments is generally very high. However, great challenges have been confronted by the efforts of developing pose-robust face recognition. A recent study [218] shows that the performance of most FR algorithms are reduced by more than 10% from frontal-frontal to frontal-profile face verification. This indicates that pose variation is still the most important challenge in real-world face recognition. The pose is defined as a combination of viewpoint and facial configuration. In this work, we aim to develop effective model for recognizing unconstrained faces with large pose variations.

The mainstream methods of Pose-Robust Face Recognition can be divided into two categories. First, some works employ face frontalization [209], [20], [249] to synthesize a frontal face from a given profile before recognition. Although it's an effective preprocessing method, achieving face frontalization with extreme profile faces is still challenging. Second, other studies design one joint model [234], [215], [115] or multiple pose-specific models [47], [167] to learn discriminative features directly from the non-frontal faces.

Recently, Cao et al. [21] proposed a deep residual equivariant mapping (DREAM) block to perform face frontalization in the deep feature space rather than in the image space. However, an important shortcoming of DREAM is the inaccurate hypothesis of the feature equivariance of CNNs. Despite their success, CNNs suffer from inherent limitations, most significantly the fact that CNNs lacks of transformation equivariant representations. This limitation has stimulated significant research in recent years [73]. Besides, the DREAM block trained using an existing face recognition loss (e.g., verification loss and identification loss) can hardly explain the generalization ability of metric from training set to testing set. Conversely, without the limit of singel objective on the overall training set, sampling sub-tasks from the original task may be useful for learning the potentially transferable information .

In this study, we aim to learn face representations equivariant to pose variations and propose a novel meta learning framework for Pose-Robust Face Recognition (PRFR). It is desirable that if the input image is transformed, e.g., by an out-of-plane rotation or pose variation, the learnt model should make predictions in a co-transformational way. Using this approach, we can actually map the features of the input image with arbitrary poses to the front space through the mapping function. With the help of meta learning, it helps the model to adapt to new tasks efficiently via extracting useful transferable knowledge from a set of auxiliary tasks.

The task of designing equivariant features has been extensively researched recently. For example, some work is devoted to extracting invariant local descriptors [161] on top of an equivariant detector [168]. Many papers have also studied the explicit inclusion of equivalence in representations [213], [228]. Recently, group equivariant CNNs [30], [31] have been proposed to ensure predictive responses to specific transformations of inputs. In particular, through constraining the CNN model family, the rotation of the input results in a corresponding rotation of its subsequent representation. However, these techniques are the most common design for the rotation and translation of inputs and fail to generalize to continuous transformation of deeper combinations. Recently, the Capsule network has been

introduced by Hinton et al. [96], [208] as an explicit step towards achieving equivariance. The motivation of the Capsule network is that the information processing should resemble a parse tree and the connections between layers should be determined by an iterative routing operation.

Based on the above discussion, we propose a deep meta Capsule network-based architecture Equivariant Embedding Model (DM-CEEM) for Pose-robust Face Recognition. In DM-CEEM, we formulate a pose equivariant embedding learning process following meta learning strategy and achieve the desired transformation for the input face image. Inspired by recent progress in equivariant learning, we introduce a new version of a capsule network called RB-Capsule network, which learns face representations equivariant to pose variations. In RB-Capsule network, we replace a single feature by a pose vector that represents the different internal properties, then generates residuals to convert a profile face to a frontal face. The residuals are generated according to the previous feature representation via some additional weight layers. In addition, the amount of residuals are adaptively controlled by a soft-gate warping-block. In this way, no residual is added to a frontal face while adding more residuals to extreme profile faces. We follow the practice of episodic training in [254] which is the most popular and effective meta learning methodology [226]. Specifically, we randomly sample a subset from the original training set as a sub-task in each episode and divide it into a support set and a query set, and optimize the model to match the query sample with positive support samples by distance. We make extensive comparisons with published state-of-the-art methods on the IJB-A and CFP datasets. Experiment results demonstrate our DM-CEEM can significantly improve the performance pose-robust face recognition. To summarize, our main contributions are listed as follows:

- A novel deep meta Capsule-based architecture Equivariant Embedding method (DM-CEEM) is proposed to learn feature equivariant to pose variations in the framework of meta-learning and improve the performance for pose-robust face recognition.
- We propose a new version of a Capsule network called RB-Capsule network, which is the first attempt to extend Capsule network to perform face transformations in the deep feature space
- We construct a unified end-to-end deep network to integrate the algorithmic components, thereby making the training process efficient and effective.
- We conduct extensive experiments and improve state-of-the-art pose-robust face

recognition performance on two benchmark datasets, IJB-A and CFP and demonstrate the effectiveness of our proposal.

5.3.2 Related Work

Our work is mostly related to the Capsule network, meta-learning and face recognition, and we start with a brief review of them.

Capsule network. The traditional CNN is not effective in capturing the hierarchical structure of the entities in the images [184], [259]. Hinton et al. [96] proposed the concept of "capsules" to learn part-whole relationships and preserve the spatial information. Capsule network was first introduced by Sabour et al. [208], which has attracted a lot of attention from researchers as a more effective image recognition algorithm. In [207], the matrix capsule learns the relationship between the observer (the pose) and the entity. Besides, a number of approaches have been introduced to implement and improve the capsule architecture [124], [294], [140]. Rajasegaran et al. [195] proposed a deep capsule network architecture called DeepCaps that increases its performance on more complex datasets.

Recently, Capsule network has also been introduced into many applications. In [108], the standard CNNs have been replaced by capsule networks as discriminators. Aryan et al. [172] utilizes a consistent dynamic routing mechanism to achieve the task of lung cancer screening. Turab et al. [288] successfully propose CNN-Capsule network for remote sensing image scene classification. Our work can be regarded as an improved version of capsules, which capsules to synthesize frontal faces with arbitrary poses in the deep feature space.

Meta Learning. The objective of meta-learning is to learn an embedding model so that the base learner can generalize well across tasks. For example, there are some meta learning methods [5], [28] interpret gradient update as a parametric and learnable function instead of a fixed ad-hoc routine. MAML [63] provides another promising direction in which learners' initial parameters can be learned for rapid adaptation. Several recent works [113], [173], [212] maintain the knowledge through memory-augmented models and access important and previously unseen information related to new tasks. Matching Networks [254] and its later developments [226], [70] is to learn a set of classifiers through prior tasks, and address the few shot learning problem by weighting these nearest neighbor classifiers. Different from the goal of Matching Networks and Prototypical Networks [226] that matching few-shot samples into positive classes using their neighbors in the support set, we focus on more general equivariant embedding learning for face recognition tasks,

rather than few-shot learning.

Face Recognition. Research on CNNs has significantly pushed forward the development of face recognition techniques. To alleviate the training time problem and learn a compact embedding for face representation, a light CNN framework has been introduced in [234]. Schroff et al. proposed a unified system for face verification, recognition and clustering [215]. Recently, pose variations have been taken into account in a number of recent works [218], [167], [296]. For examples, some researchers have proposed the methods for detecting and deforming 3D facial markers [242], face frontalization [249] and training deep models for learning pose-specific identity features [167]. Zhao et al. [292] propose a pose invariant model (PIM) which aggregates a discriminative learning sub-net (DLN) and a face frontalization sub-net (FFN). The generation of high fidelity frontalized face images in PIM makes it essentially a pixel-level alignment method. In contrast, the method in [90] explicitly considers feature-level alignments and use deformable convolutions with a spatial displacement field to extract deep feature for face recognition. Different with the existing research that require well-designed data augmentation or multi-task training, our approach is easy to implement and light-weight.

5.3.3 Background

Equivariance is a desirable property for a computer vision system. If an input image is transformed, e.g., by a rotation, then the system should make inferences in a predictable way via a co-transformed representation. We define $f : X \to Y$ as a function and P_ϕ and Q_ϕ are two sets of transformations parametrized by ϕ. f is equivariant to P and Q if

$$f(P_\phi(x)) = Q_\phi(f(x)) \tag{5.14}$$

where P_ϕ and Q_ϕ are a pair of transformations whose order with f can be exchanged when one is replaced with another. This means that first using f and then transforming the output with Q is same with transforming the input with P and then using f. We can assume that the neural network is equivariant, if network is able to against transformations automatically in P. This approach decreases the necessary of data augmentation as the implicit transformations hidden in data augmentation have already been learnt by the network. Conventional CNN architectures have the built-in translational equivariance, which means that the CNN does not have to learn the shifted versions of the same patterns.

5.3.4 Deep Meta Capsule Network-based Equivariant Embedding Model

Our goal is to learn face representations equivariant to pose variations and achieve transformation from profile face to frontal face in the deep embeddings space. As shown in Figure 5.9, the proposed DM-CEEM extends from an Capsule Network, and consists of a RB-Capsule network and a soft-gate warping-block that jointly learn discriminative and robust face representations disentangled from pose variance and perform face recognition end-to-end. In this section, we will introduce the problem formulation and describe the proposed method in detail.

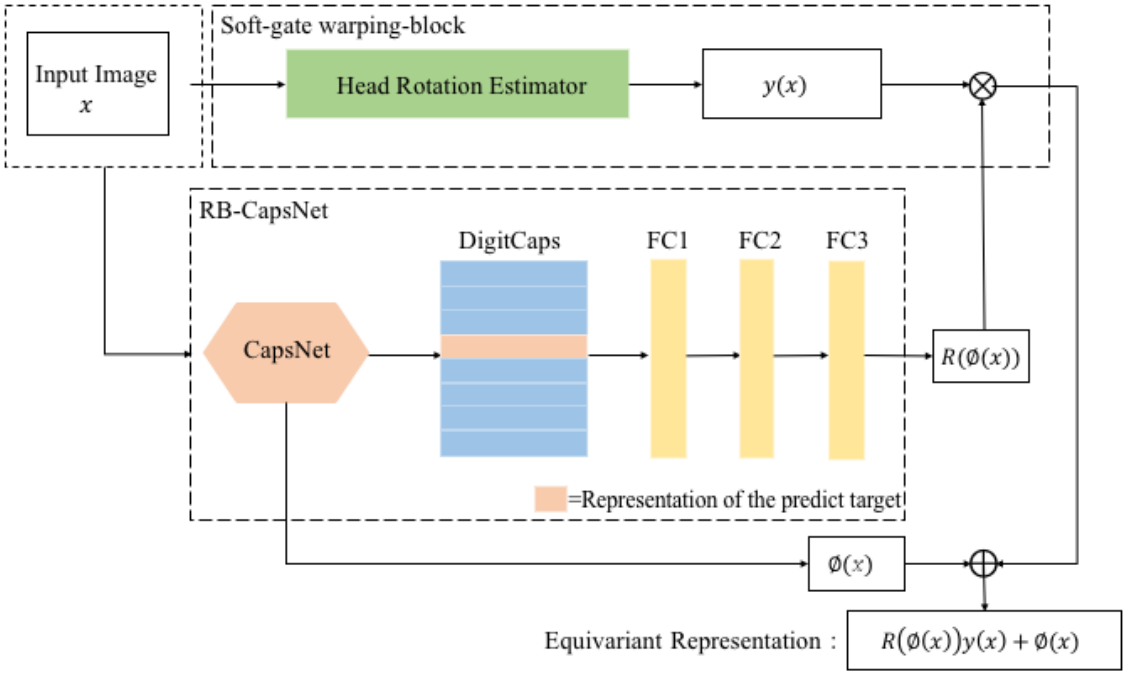


Figure 5.9: The framework of the Capsule network-based Equivariant Embedding Model (CEEM). The procedure consists of RB-Capsule network and Soft-gate warping-block.

5.3.5 Problem Formulation

We denote a Capsule network as a function ϕ that maps an image $x \in X$ to a representation vector $\phi(x) \in R^d$. If the transformation t of the input image can be transferred to the representation output, the representation ϕ is equivariant. Therefore, we can obtain the

equivariance with t when a map M_t exists, such that,

$$\forall x \in X : \phi \approx M_t \phi(x) \tag{5.15}$$

To facilitate the discussion, we suppose that frontal face image x_A and profile face image x_B are two types of face images. Inspired by Equation 5.15, we hope to use a mapping function M_t to obtain a transformed representation of a profile face image x_A, so that $M_t\phi(x_B) \approx \phi(x_A)$. To facilitate the incorporation of $M_t\phi(x_B)$ in a Capsule network, we define it as a sum of the original profile feature $\phi(x_B)$ with residuals provided by a residual function $R(\phi(x_B))$ weighted by a yaw coefficient $Y(x_B)$. It can be expressed as:

$$\begin{aligned} \phi(gx_B) &= M_t\phi(x_B) \\ &= \phi(x_B) + y(x_B)R(\phi(x_B)) \\ &\approx \phi(x_A) \end{aligned} \tag{5.16}$$

After this transformation, the fixed representation $\phi(x_B) + y(x_B)R(\phi(x_B))$ will be mapped to the frontal face space. We design the RB-Capsule network as a new version of Capsule network to obtain $\phi(x_B)$ and $R(\phi(x_B))$, and use the soft-gate warping-block to provide a higher magnitude $y(x_B)$ of residuals $R(\phi(x_B))$.

5.3.6 RB-Capsule Network

In pose-robust face recognition, the equivariant representation extracted from deep models is an essential function for feature learning. As explained in Section 5.3.1, we aim to propose a method to make the representation robust to pose variation. To fulfill this goal, we integrate a residual block with DeepCaps [195]. During training, DeepCaps learns an equivariant mapping between the input images and generated vector, and the residual block is to learns the latent space that bridges the discrepancy between profiles and frontal faces.

There are two main capsule types in the original Capsule network [208], namely the PrimaryCapsules and the DigitCaps. In Capsule network, 256 channels will be produced by an initial convolution layer and then rearranges another set of convolutions into 32×8D PrimaryCapsules. Using the mechanism of dynamic routing by agreement, the PrimaryCapsules are routed to the next DigitCaps layer. Therefore, the similar votes from PrimaryCapsules will contribute more strongly to the target DigitCaps. The dynamic

routing will update the contribution of votes. Based on the similarity between the output DigitCaps and the prediction vector, the dynamic routing will update the contribution of votes. Therefore, after the pervious previous layer of capsules (PrimaryCapsule layer) $\hat{u}_{j|i}$ providing the prediction vectors, where i denotes the index of a single capsule in the PrimaryCapsule layer and j represents the index of the DigitCaps capsule, the output vector (DigitCaps) is computed as,

$$s_j = \sum_i c_{ij} \hat{u}_{j|i} \tag{5.17}$$

where c_{ij} represents the coupling coefficients weighting the contributions of different prediction vectors,

$$c_{ij}^{(f_{out})} = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})} \tag{5.18}$$

where b_{ij} represents the log probability that the i_{th} PrimaryCapsule should be coupled to the j_{th} DigitCaps capsule. The sum of the weights of the contributions of the i_{th} PrimaryCapsule to each DigitCaps capsule in the next layer are normalized to one.

DeepCaps is a deep capsule network architecture proposed in [195] to improve the performance of the capsule networks for more complex image datasets. It extends the dynamic routing algorithm in [208] to stacked multiple layers, which essentially uses a 3D convolution to learn the spatial information between the capsules. The model consists of four main modules: skip connected CapsCells, 3D convolutional CapsCells, a fully-connected capsule layer and a decoder network. The skip-connected CapsCells have three ConvCaps layers, the first layer output is convolved and skip-connected to the last layer output. The motivation behind skipping connections is to borrow the idea from residual network to sustain a sound gradient flow in a deep model. The element-wise layer is used to combine the outputs of the two capsule layers after skipping the connection. DeepCaps has a unit with a ConvCaps3D layer, in which the number of route iterations is kept at 3. Then, before dynamic routing, the output of ConvCaps is flattened and connected with the output of the capsule, which is then followed by 3D routing (in CapsCell 3). Intuitively, this step helps to extend the model to a wide range of different datasets.

RB-Capsule network shares the upper part of Deepcaps, starting from the skip connected CapsCells, through the 3D convolutional CapsCells, a fully-connected capsule layer and and ending with a decoder network if needed. We set the number of routing iterations as three in all the experiments and the fully-connected capsule layer computes the 32-D feature

finally. The Residual block has three fully-connected layers with Parametric Rectified Linear Unit (PReLU) as the activation function. Then we fed the output of the digit capsule into a decoder consisting of 3 fully connected layers. Specifically, we train it by using SGD to minimizing the Euclidean distance between the frontal feature and the mapped profile feature.

$$\min_{\theta_R} E||\phi(x) + y(x)R(\phi(x)); -\phi(x_A)||_2^2 \tag{5.19}$$

where θ_R represents the parameters of $R(\cdot)$. The parameters of the $y(\cdot)$ branch are fixed. A dropout layer has been inserted before the last fully connected layer during the training process. In this work, we train the RB-Capsule network in the framework of meta-learning on multiple sub-tasks sampled from MS-Celeb-1M dataset.

Soft-gate warping-block

The soft-gate warping-block generates the soft yaw coefficient $y(x)$. Given an input face image, the head rotation estimator in the soft-gate warping-block will estimate the head rotation via the deepgaze algorithm presented in [186]. Then we non-linearly mapped the yaw angle to a positive value in the range of $[0, 1]$ as Equation 5.20.

$$y(x) = \frac{x - min}{max - min} \tag{5.20}$$

Based on this mapping, once the head rotation is greater than 60^o, the coefficient quickly reaches a value of 1 while more residuals are exerted to the extreme profile faces. To facilitate the incorporation of $M_t\phi(x_B)$ in a Capsule network, we define it as a sum of the residuals weighted by a yaw coefficient with original face feature.

Therefore, the fixed representation $\phi(x_B) + y(x_B)R(\phi(x_B))$ will be mapped to the deep space of frontal face via performing this transformation. With the help of yaw coefficient $y(x) \in [0, 1]$, Equation 5.16 is able to cope with input images of arbitrary pose. If the face deviates more from the frontal posture, then a higher range of magnitude will be provided. Intuitively, in the case where a face image that is frontal, $y(x) = 0$. During the process of decreasing the degree of head rotation from 90^o to 0^o, the value of $y(x)$ is gradually increased from 0 to 1.

The design of soft-gate warping-block is requisite and essential. If not using it, the model will blindly add residuals $R(\phi(x))$ to the input images of any poses, thus the performance

of face recognition will be affected. We view the design of combining a soft control gate as a correction mechanism that adopts the degree of head rotation to influence the process of feed-forward. The soft-gate warping-block determines the amount of the residuals to be fed into the next stage. It is worth emphasizing that the angles of pitch and roll are not considered in the model. The outputs of the residual block in the RB-Capsule and soft gate are multiplied and added to the initial representation $\phi(x)$. The final out of feature representation can be expresses as $\phi(x) + y(x)R(\phi(x))$.

Training algorithm

In our DM-CEEM method, the learning process and generalization ability of the model can be better explained by formulating learning process in a meta way instead of considering a single objective with the overall observation of training data. The single training goal is divided into multiple sub-tasks, and learn the meta metrics applicable to all sub-tasks. In our assumption, the test task and all sub-tasks are instances sampled from a task distribution $p(\mathcal{T})$.

We formulate the objective function of the proposed DM-CEEM method as:

$$\theta = \underset{\theta}{argmin}\, \mathbb{E}_{\mathcal{T}_k \sim p(\mathcal{T})}[\mathcal{L}_k(\theta; \mathcal{X}_k, \mathcal{Y}_k)] \tag{5.21}$$

where $\mathcal{L}_k(\theta; \mathcal{X}_k, \mathcal{Y}_k)$ represents the objective function of sampled sub-task $\mathcal{T}_k$. Specifically, assume there are N-class in training set, $M(M \leq N)$ classes will be randomly sampled from the original task to construct a new task. Similar to the form of meta learning, we randomly sample a support set $S = \{s_i^m | i = 1, ...n_s^m\}$ and a query set $Q = \left\{q_i^m | i = 1, ...n_q^m\right\}$ for the sub-task $\mathcal{Y}_k$), where $m = 1, ..., M$ is the different classes. For simplicity, we set the number of support samples and query samples for the different classes to be the same, i.e. $n_s^m = n_s$ and $n_q^m = n_q$. In each episode, we learn the model to correctly verify the query sample from Q with support samples in S. The overall formulation of our DM-CEEM method is:

$$\theta = \underset{\theta}{argmin}\, \mathbb{E}_{\mathcal{T}_k \sim p(\mathcal{T})}[\mathbb{E}_{S,Q \sim \mathcal{T}_k}[\mathcal{L}_k(\theta; \mathcal{Q}, \mathcal{S}]] \tag{5.22}$$

5.3.7 Experiments

Experimental Settings

Dataset. To demonstrate the performance of our approach, we conduct our experiment on two datasets: Celebrities in Frontal-Profile (CFP) dataset [218] and the IARPA Janus Benchmark A (IJB-A)[121]. CFP is a challenging dataset for examining the problem of pose-robust face recognition 'in the wild'. It contains 500 celebrities, each with 10 frontal and 4 profile face images. There are two evaluation protocols: frontal-frontal (FF) and frontal-profile (FP) face verification, each having 10 subset with 350 pairs of same-people and different-people. IJB-A is another large posture database and consists of 5, 396 images and 20, 412 video frames belonging to 500 subjects. The faces in the IJB- A dataset cover full pose variation (yaw angles between -90^o to $+90^o$), which is more challenging than the CFP dataset.

Implementation Details. We select a small subset of MS-Celeb-1M [85] as our training and testing sets. There are 16,104 images from 157 identities and 3,000 images from 36 identities in the training partition and testing partition, respectively. We resize all face images in the training and testing sets to 112×112. To evaluate the performance of the proposed method on the CFP dataset, we follow the standard 10-fold protocol in [218] and measure the cosine distance between the feature representations of the queries. For the IJB-A dataset, we follow the standard protocol in [249] and evaluate the performance of our method on both the identification (1:N) and face verification tasks (1:1). We implement the DM-CEEM using the PyTorch and employs the Adam optimization method as the gradient descent algorithm to perform the training. We set the class number of each sub-task and the number of support samples in each episode as 16 and 5 respectively. The initial learning rate is 0.0001 and the number of episode is 2,000,000 to quickly converge to an optimal solution quickly.

Evaluation on Celebrities in the Frontal-Profile (CFP) dataset

Our experiments on the CFP dataset required a shorter training time with the small size of the training data. Our system does not perform any supervised training using the target dataset; supervised training is only performed on the MS-Celeb-1M, which contrasts sharply with the previous works of [24] and [21], which train deep networks using the CASIA-WebFace dataset and the VGG-Face dataset respectively and require substantial

Table 5.6: Equal error rate (EER) for different methods on the CFP dataset [218] with the Frontal-Profile setting.

Method	Training Data	Equal error rate (%)
TPE [211]	CASIA-WebFace	8.85
FV-DCNN [24]	CASIA-WebFace	8.00
PIM [292]	MS-Celeb-1M	7.69
DREAM [21]	MS-Celeb-1M	6.43
p-CNN [273]	CASIA-Webface	5.94
ResNet-18	MS-Celeb-1M	9.23
ResNet-50	MS-Celeb-1M	8.13
DM-CEEM (Proposed method)	MS-Celeb-1M	**4.72**

training times and sufficiently labeled data.

We also explore the influence of using different component combinations of the proposed DM-CEEM, our baseline, two single CNN model (ResNet 18 and ResNet-50) trained on the subset of MS-Celeb-1M, respectively. As shown in Table 5.6, the DM-CEEM performs better than the CNN model and has a greater improvement on CFP. Meanwhile, the DM-CEEM achieves a faster convergence compared to the CNN model. The results reveal that the DM-CEEM has a better robustness on the face recognition dataset, in which the face images have complex internal pose variations. Figure 5.10 shows some sample image pairs from the CFP where our method is able to successfully verify the pairs whereas both ResNet-18 and ResNet-50 failed.

Figure 5.10: Examples of sample face pairs from the CFP dataset[218], the proposed method verifies them successfully.

Table 5.6 shows that our face verification performance comparisons with state-of-the-art methods on the CFP dataset. Our experiment result, which is obtained by DM-CEEM

trained with a subset of MS-Celeb-1M, outperforms the state-of-the-art results [273], [21].

Experimental results on IJB-A with Full Pose Variation

In order to further test the effectiveness of our proposed algorithm, we also conduct experiments on the IJB-A database. We presents the results of the proposed method in Table 5.7, also comparing to existing results for the Verification and Identification protocol in IJB-A. All the methods are tested with the same setting. Note that the training set of IJB-A is not used by any methods for comparison.

Table 5.7: The performance of face verification and face identification for different methods on the IJB-A benchmark [121]. Results reported are the 'average±standard deviation' (%) over the 10 folds specified in the IJB-A protocol. Symbol '-' defines that the metric is not available for that protocol. f.t. indicates fine tuning a deep network multiple times for each training split.

Method	Verification		Identification	
	TAR @ FAR=0.01	TAR @ FAR=0.001	Rank-1	Rank-5
DR-GAN[249]	77.4±2.7	53.9±4.3	85.5±1.5	94.7±1.1
PAMs [167]	82.6±1.8	65.2±3.7	84.0±1.2	92.5±0.8
TA (f.t.) [37]	93.9±1.3	-	92.8±1.0	-
PIM [292]	93.3±1.1	87.5±1.8	94.4±1.1	-
QAN (f.t.) [159]	94.2±1.5	89.3±3.9	-	-
DREAM [21]	94.4±0.9	86.8±1.5	92.6±1.1	96.8±1.0
HF-PIM [20]	95.2±0.7	89.7±1.4	96.1±0.5	97.9±0.2
DeepCaps with meta-learning	89.4±0.9	77.1±1.2	90.6±1.0	94.0±0.3
DM-CEEM (Proposed method)	**97.8±0.4**	**92.9±0.8**	**97.2±0.6**	**98.7±0.3**

It is noteworthy that the baseline alone achieved Rank-1 recognition accuracy of 90.6% and Rank-5 recognition accuracy of 94% on the identification task. As shown in Table II, by replacing the Capsule network with RB-Capsule network and adding the Soft-gate warping-block, our method significantly improves the performance on IJA-B dataset. We observed improvement of +6.6% (from 90.6% to 97.2%) and +4.7% (from 94% to 98.7%) on Rank-1 accuracy and Rank-5 accuracy on the identification task, respectively. Compared with the best state-of-the-art method [20], the proposed model improves the face verification accuracy (TAR@FAR=0.001) and Rank-1 match rates by 2.6% (from 95.2% to 97.8%) and 1.1% (from 96.1% to 97.2%), respectively, which proves once again that our method has significant advantages.

Further Discussions of Equivariance Embedding of Faces

Head rotation is the most visible property of an entity and one of the easiest operations to a face. To recognize the face image with random rotations, the proposed DM-CEEM learns the equivariant embedding taking rotation as the most significant instantiation parameter. Since the equivariant representations are nonlinear (after squashing), we expect to find the cosine similarity, or equivalently, inner product of vectors of these representations after normalization. An important observation is that,

$$\begin{aligned}&(DM - CEEM(rot(B, d)), DM - CEEM(A))\\ &\approx (DM - CEEM(rot(A, d)), DM - CEEM(B))\end{aligned} \tag{5.23}$$

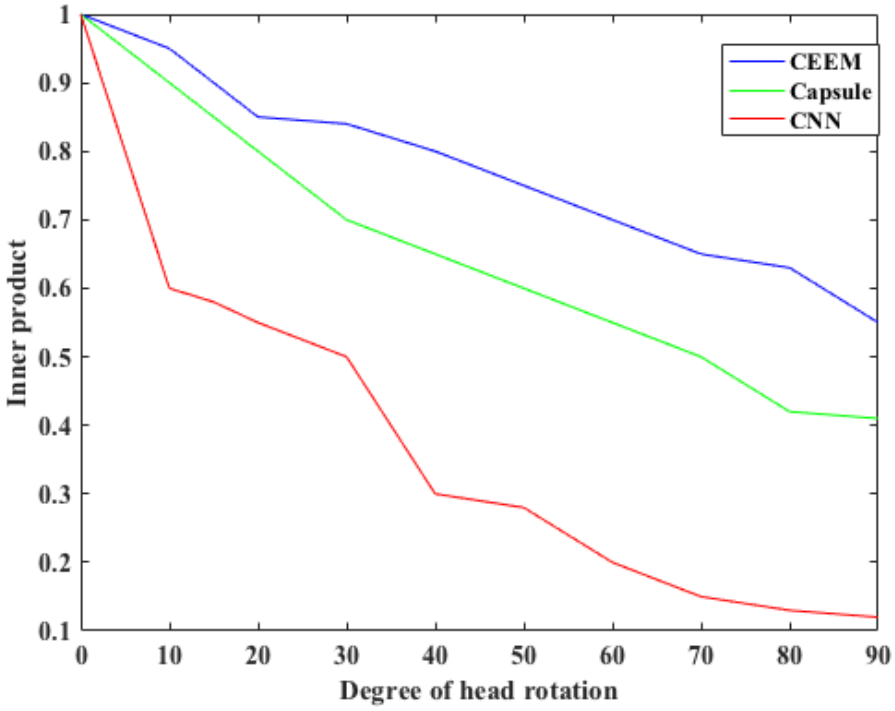


Figure 5.11: Face image with head rotation iteratively.

Where A, B are two face images, d defines the degree of head rotation. This suggests that rotation indeed dominates the embedding space. We further compare

$(DM-CEEM(rot(A,d)), DM-CEEM(A))$ with $(Capsule(rot(A,d)), Capsule(A))$ and $(CNN(rot(A,d)), CNN(A))$ when degree d varies. As illustrate in Figure 5.11, we find that for DM-CEEM, the inner product is still high when $d > 60$, for Capsule, the inner product decrease as d increases when $d < 90$. However, the curve is more complex with the change of d and the inner product is smaller than 0.5 when $d > 30$.

5.3.8 Conclusion

This paper has proposed a deep meta Capsule network-based Equivariant Embedding Model (DM-CEEM) to improve the performance of pose-robust face recognition. Different with the existing popular methods, we address the problem of learning an equivariant embedding for pose variations in a unified end-to-end deep network. In addition, we first propose to consider the target of single overall face recognition as multiple sub-tasks that satisfy a certain unknown probability, and randomly sample the support and query sets in each sub-task in one episode. Specifically, DM-CEEM combines the advantages of Capsule network and achieves the desired transformation in deep feature space in the framework of meta-learning. Through a new version of a capsule network called RB-Capsule network, a pose vector and corresponding residuals are generated to represent the different internal properties. We also introduced a soft-gate warping-block to adaptively control the amount of residuals. Thus, the DM-CEEM architecture makes it easy to handle profile faces and acquires the equivariance. Our experimental results indicate that the proposed methods significantly outperforms state-of-the-arts methods on the CFP and IJB-A datasets.

Chapter 6

Attentive Prototype Few-shot Classification with Capsule Network-based Embedding

Representation learninghas shown its effectiveness in many tasks of Chapter 2-5, including traffic scene recognition, vehicle re-ID and face recognition in uncontrolled environment. Previous works have proposed several techniques by using GANs for unsupervised representation learning and adopting the deep models (Capsule network and CNNs) for learning deep feature representation. However, most of these representation learning models need large amounts of data and many iterations to train their large number of parameters. When the training samples are scarce, it becomes challenging to learn to recognize new concepts due to the overfitting problem. In this chapter, we consider learning representation in the setting in which we have to recognize novel visual categories from very few labelled examples. The availability of only one or very few examples challenges the standard 'fine-tuning' practice in deep learning. Specifically, we focus on the setting where there exists a good common representation between source and target. If the learned representation is good enough, it is possible that a few samples are sufficient for learning the target the target task, which can be much smaller than the number of samples required to learn the target task from scratch. Besides, the proposed technique in this chapter is not limited to few-shot classification and is extensible to other tasks of Chapter 2-5 as well.

6.1 Introduction

Deep learning has been greatly advanced in recent years, with many successful applications in image processing, speech processing, natural language processing and other fields. However, the successes usually rely on the condition to access a large dataset for training. If the amount of training data is not large enough, the deep neural network would not be sufficiently trained. Consequently, it is significant to develop deep learning for image recognition in the case of a small number of samples, and enhance the adaptability of deep learning models in different problem domains.

Few-shot classification is one of the most promising research areas targeting deep learning models for various tasks with a very small amount of training dataset [183], [199], [206], [226], [236], [254], i.e., classifying unseen data instances (query examples) into a set of new categories, given just a small number of labeled instances in each class (support examples). The common scenario is a support set with only 1∼10 labeled examples per class. As a stark contrast, general classification problems with deep learning models [127], [239] often require thousands of examples per class. On the other hand, classes for training and testing sets are from two exclusive sets in few-shot classification, while in traditional classification problems they are the same. A key challenge, in few-shot classification, is to make best use of the limited data available in the support set in order to find the right generalizations as required by the task.

Few-shot classification is often elaborated as a meta-learning problem, with an emphasis on learning prior knowledge shared across a distribution of tasks [170], [226], [254]. There are two sub-tasks for meta-learning: an embedding that maps the input into a feature space and a base learner that maps the feature space to task variables. As a simple, efficient and the most popularly used few-shot classification algorithm, the prototypical network [226] tries to solve the problem by learning the metric space to perform classification. A query point (new point) is classified based on the distance between the created prototypical representation of each class and the query point. While the approach is extensively applied, there are a number of limitations that we'd like to address and seek better solutions.

Firstly, the prototypical representations [226], [254], generated by deep Convolutional Neural Networks, cannot account for the spatial relations between the parts of the image and are too sensitive to orientation. Secondly, a prototypical network [226] divides the output metric space into disjoint polygons where the nearest neighbor of any point inside a polygon is the pivot of the polygon. This is too rough to reflect various noise effects in

the data, thus compromising the discrimination and expressiveness of the prototype. It has been well-known that the performance of such a simple distance-based classification is severely influenced by the existing outliers, especially in the situations of small training sample size [68].

From the aforementioned discussion, we intend to improve the prototype network by proposing a capsule network [208] based embedding model and reconstruction-based prototypical learning within the framework of meta-learning. There are two main components in the proposed scheme: a capsule network-based embedding module which create feature representations, and an improved non-parametric classification scheme with an attentive prototype for each class in the support set, which is obtained by attentive aggregation over the representations of its support instances, where the weights are calculated using the reconstruction error for the query instance.

The training of the proposed network is based on the metric learning algorithm with an improved triplet-like loss, which generalizes the triplet network [215] to allow joint comparison with K negative prototypes in each mini-batch. This makes the feature embedding learning process more tally with the few-shot classification problem. We further propose a semi-hard mining technique to sample informative hard triplets, thus speeding up the convergence and stabilize the training procedure.

In summary, we proposed a new embedding approach for few-shot classification based on a capsule network, which features the capability to encode the part-whole relationships between various visual entities. An improved routing procedure using the DeepCaps mechanism [195] is designed to implement the embedding. With a class-specific output capsule, the proposed network can better preserve the semantic feature representation, and reduce the disturbances from irrelevant noisy information. The proposed attentive prototype scheme is query-dependent, rather than just averaging the feature points of a class for the prototype as in the vanilla prototype network, which means all of the feature points from the support set are attentively weighted in advance, and then the weighting values completely depend on the affinity relations between two feature points from the support set and the query set. By using reconstruction as an efficient expression of the affinity relation, the training points near the query feature point acquire more attention in the calculation of the weighting values.

The proposed approach has been experimentally evaluated on few-shot image classification tasks using three benchmark datasets, i.e. the *mini*ImageNet, *tiered*ImageNet and Fewshot-CIFAR100 datasets. The empirical results verify the superiority of our method

over the state-of-the-art approaches. The main contributions of our work are two-fold:

- We put forward a new few-shot classification approach with a capsule-based model, which combines a 3D convolution based on the dynamic routing procedure to obtain a semantic feature representation while preserving the spatial information between visual entities.
- We propose a novel attentive prototype concept to take account of all the instances in a given support class, with each instance being weighted by the reconstruction errors between the query and prototype candidates from the support set. The attentive prototype is robust to outliers by design and also allows the performance to be improved by refraining from making predictions in the absence of sufficient confidence.

6.2 Related work

6.2.1 Few-shot Classification

Few-shot classification aims to classify novel visual classes when very few labeled samples are available [59], [60]. Current methods usually tackle the challenge using meta-learning approaches or metric-learning approaches, with the representative works elaborated below.

Metric learning methods aim to learn a task-invariant metric, which provide an embedding space for learning from few-shot examples. Vinyals et al. [254] introduced the concept of episode training in few-shot classification, where metric learning-based approaches learn a distance metric between a test example and the training examples. Prototypical networks [226] learn a metric space in which classification can be performed by computing distances to prototype representations of each class. The learned embedding model maps the images of the same class closer to each other while different classes are spaced far away. The mean of the embedded support samples are utilized as the prototype to represent the class. The work in [141] goes beyond this by incorporating the context of the entire support set available by looking between the classes and identifying task-relevant features.

There are also interesting works that explore different metrics for the embedding space to provide more complex comparisons between support and query features. For example, the relation module proposed in [236] calculates the relation score between query images to identify unlabeled images. Kim et al. [117] proposed an edge-labeling Graph

Neural Network (EGNN) for few-shot classification. Metric-based task-specific feature representation learning has also been presented in many related works. Our work is a further exploration of the prototype based approaches [226], [236], aiming to enhance the performance of learning an embedding space by encoding the spatial relationship between features. Then the embedding space generates attentive prototype representations in a query-dependent scheme.

6.2.2 Capsule Networks

The capsule network [96] is a new type of neural network architecture proposed by Geoffrey Hinton, with the main motivation to address some of the shortcomings of CNNs. For example, the pooling layers of CNNs lose the location information of relevant features, one of the so-called instantiation parameters that characterize the object. Other instanced parameters include scale and rotation, which are also poorly represented in CNNs. Capsule network handles these instantiation parameters explicitly by representing an object or a part of an object. More specifically, a capsule network replaces the mechanisms of the convolution kernel in CNNs by implementing a group of neurons to encode the spatial information and the probability of the existence of objects. The length of the capsule vector is the probability of the features in the image, and the orientation of the vector will represent its instantiation information.

Sabour et al. [208] first proposed a dynamic routing algorithm for capsule networks in 2017 for the bottom-up feature integration, the essence of which is the realization of a clustering algorithm for the information transmission in the model. In [208], a Gaussian mixture model (GMM) was integrated into the feature integration process to adjust network parameters through EM routing. Since the seminal works [96], [208], a number of approaches have been proposed to implement and improve the capsule architecture [124], [140], [195], [294].

Many applications have been attempted by applying capsule networks, for example, intent detection [266], text classification [190] and computer vision [288], [289]. A sparse, unsupervised capsules network [197] was proposed showing that the network generalizes better than supervised masking, while potentially enabling deeper capsule networks. Rajasegaran et al. [195] proposed a deep capsule network architecture called DeepCaps that adapts the original routing algorithm for 3D convolutions and increases its performance on more complex datasets.

6.3 Method

6.3.1 Problem Definition: Few-shot Classification

Few-shot classification is to recognize novel categories with only one or few labeled examples by transferring visual patterns obtained from base categories to describe the novel categories. The problem is usually formulated with three datasets: a training set D_{train}, a support set $D_{support}$ and a query set D_{query}. The categories in D_{train} are defined as base categories C_{base}. The categories in $D_{support}$ and D_{test} are novel categories which are exclusive with the training set D_{train}. If the support set contains M categories and each category has K image examples, this few-shot classification problem is defined as M-way K-shot learning. We follow the practice of episodic training in [254] which is the most popular and effective meta learning methodology [226], [236].

6.3.2 Approach Details

In this section, we first revisit the DeepCaps network [195], which is designed for more complex image datasets. We then extend it to the scenario of few-shot classification and describe the proposed algorithm in detail.

DeepCaps Revisit

DeepCaps is a deep capsule network architecture proposed in [195] to improve the performance of the capsule networks for more complex image datasets. It extends the dynamic routing algorithm in [208] to stacked multiple layers, which essentially uses a 3D convolution to learn the spatial information between the capsules. The model consists of four main modules: skip connected CapsCells, 3D convolutional CapsCells, a fully-connected capsule layer and a decoder network. The skip-connected CapsCells have three ConvCaps layers, the first layer output is convolved and skip-connected to the last layer output. The motivation behind skipping connections is to borrow the idea from residual networks to sustain a sound gradient flow in a deep model. The element-wise layer is used to combine the outputs of the two capsule layers after skipping the connection.

DeepCaps has a unit with a ConvCaps3D layer, in which the number of route iterations is kept at 3. Then, before dynamic routing, the output of ConvCaps is flattened and connected with the output of the capsule, which is then followed by 3D routing (in CapsCell 3). Intuitively, this step helps to extend the model to a wide range of different datasets.

For example, for a dataset composed of images with less rich information, such as MNIST, the low-level capsule from cell 1 or cell 2 is sufficient, while for a more complex dataset, we need the deeper 3D ConvCaps to capture rich information content. Once all capsules are collected and connected, they are routed to the class capsule through the fully-connected capsule layer.

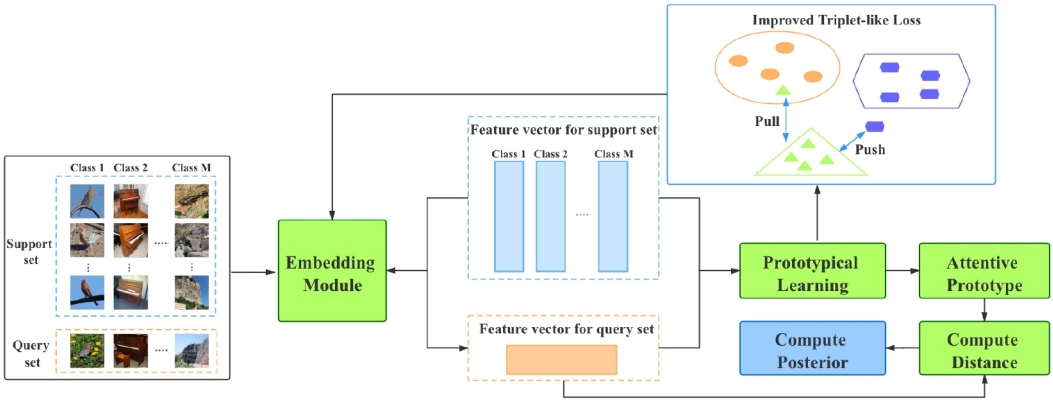


Figure 6.1: Framework of the proposed method for few-shot classification. We perform joint end-to-end training of the Embedding Module (modified DeepCaps) together with the Prototypical Learning via an improved triplet-like loss from the training dataset. The well-learned embedding features are used to compute the distances among the query images and the attentive prototype generated from the support set. The final classification is performed by calculating the posterior probability for the query instance.

Network Architecture

As explained in the Introduction, our proposed model has two parts: (1) a modified DeepCaps network with improved triplet-like loss that learns the deep embedding space, and (2) a non-parameter classification scheme that produces a prototype vector for each class candidate, which is derived from the attentive aggregation over the representations of its support instances, where the weights are calculated using the reconstruction errors for the query instance from respective support instances in the embedding space. The final classification is performed by calculating the posterior probability for the query instance based on the distances between the embedding vectors of the query and the attentive prototype. Figure 6.1 schematically illustrates an overview of our approach to few-shot

image classification. Each of the parts is described in detail below.

Embedding module. We follow the practice of episodic training in [254] which is the most popular and effective meta learning methodology [226], [236]. We construct support set S and query set Q from D_{train} in each episode to train the model.

$$\begin{aligned} S &= \{s_1, s_2, .., s_K\}, \\ Q &= \{q_1, q_2, ..., q_N\}, \end{aligned} \tag{6.1}$$

where K and N represent the number of samples in the support set and query set for each class, respectively. As shown in Figure 6.2, we first feed the samples S and Q into the convolution layer and CapsCells, then the collected capsules are routed to the class capsules after the Flat Caps layer. Here, the decision making happens via L_2 and the input image is encoded into the final capsule vector. The length of the capsule's output vector represents the probability that the object represented by the capsule exists in the current input. We assume the class capsules as $P \in Y^{b \times d}$ which consists of the activity vectors for all classes, where b and d represents the number of classes in the final class capsule and capsule dimension, respectively. Then, we only feed the activity vector of predicted class $P_m \in Y^{1 \times d}$ into the final embedding space in our setting, where $m = argmax_i(||P_i||_2^2)$. The embedding space acts as a better regularizer for the capsule networks, since it is forced to learn the activity vectors jointly within a constrained Y^d space. The function of margin loss used in DeepCaps enhances the class probability of the true class, while suppressing the class probabilities of the other classes. In this paper, we propose the improved triplet-like loss based on an attentive prototype to train the embedding module and learn more discriminative features.

Attentive prototype. The prototypical network in [226] computes a D dimensional feature representation $p_i \in \mathbb{R}^D$, or prototype, of each class through an embedding function $f_\phi : \mathbb{R}^D \to \mathbb{R}^M$ with learnable parameters ϕ. Each prototype is the mean vector of the embedded support points belonging to its class:

$$p_i = \frac{1}{|s_i|} \sum_{(x_i, y_i) \in s_i} f_\phi(x_i) \tag{6.2}$$

where each $x_i \in s_i$ is the D-dimensional feature vector of an example from class i. Given a distance function $d : \mathbb{R}^D \times \mathbb{R}^D \to [0, +\infty)$, prototypical networks produce a distribution over classes for a query point x based on a softmax over distances to the prototypes in the

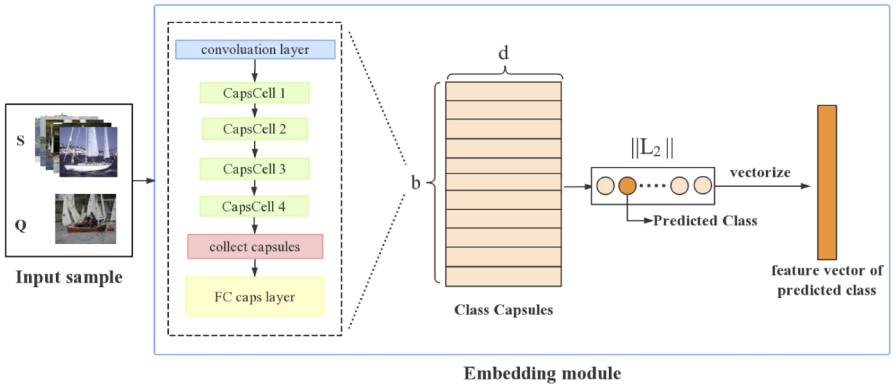


Figure 6.2: The architecture of the embedding module in which obtains only the activity vectors of the predicted class.

embedding space:

$$p_\phi(y = t|x) = \frac{exp(-d(f_\phi(x), p_t))}{\sum_{t'} exp(-d(f_\phi(x), p_{t'}))} \tag{6.3}$$

Learning proceeds by minimizing the negative log-probability $J(\phi) = -logp_\phi(y = t|x)$ of the true class t via SGD. Most prototypical networks for few-shot classification use some simple non-parametric classifiers, such as kNN. It is well known that non-parametric classifiers are usually affected by existing outliers [67], which is particularly serious when the number of samples is small, the scenario addressed by few-shot classification. A practical and reliable classifier should be robust to outliers. Motivated by this observation, we propose an improved algorithm based on the local mean classifier [171]. Given all prototype instances of a class, we calculate their reconstruction errors for the query instance, which are then used for the weighted average of prototype instances. The new prototype aggregates attentive contributions from all of the instances. The reconstruction error between the new prototype and the query instance not only provides a discrimination criteria for the classes, but also serves as a reference for the reliability of the classification.

More specifically, with K support samples $\{x_{i1}, x_{i2}, ..., x_{iK}\}$ selected for class i, a membership γ_{ij} can be defined for a query instance q by employing normalized Gaussian

functions with the samples in support sets, e.g.,

$$\gamma_{ij} = \frac{exp(\frac{||q-x_{ij}||^2}{2\sigma_i^2})}{\sum_{l=1}^{K} exp(\frac{||q-x_{il}||^2}{2\sigma_i^2})}, j = 1, ..., K, i = 1, ..., M \tag{6.4}$$

where x_{ij} are the j-th samples in class i, and σ_i is the width of the Gaussian defined for class i, and we set the value σ_i relatively small (e.g, σ_i=0.1).

Then, for each class i, an attentive prototype pattern $\hat{q}_i$ can be defined for a query sample q

$$\hat{q}_i = \frac{\sum_{j=1}^{K} \gamma_{ij} x_{ij}}{\sum_{l=1}^{K} \gamma_{ij}}, i = 1, ..., M \tag{6.5}$$

Where γ_{ij} is defined in Equation 6.4 and $\hat{q}_i$ can be considered as the generalized support samples from class i for the query instance q. Here we want to ensure that an image q^a (anchor) of a specific class in the query set is closer to the attentive prototype of the positive class $\hat{q}^p$ (positive) than it is to multiple $\hat{q}^n$ (negative) attentive prototypes.

$$||q^a - \hat{q}^p||_2^2 + \alpha < ||q^a - \hat{q}^n||_2^2, \forall q^a \in Q. \tag{6.6}$$

f where α is a margin that is enforced between positive and negative pairs, Q is the query set cardinality MN. The loss that is being minimized is then:

$$\sum_{m=1}^{MN} \left[||f(q_m^a) - f(\hat{q}_m^p))||_2^2 - ||f(q_m^a) - f(\hat{q}_m^n)||_2^2 + \alpha\right]_+ \tag{6.7}$$

For image classification, a query image can be classified based on the comparison of the errors between the reconstructed vectors and the presented image. That is, a query image q is assigned to class m^* if

$$m^* = \underset{m}{argmin}\, err_m \tag{6.8}$$

where $err_m = ||q - \hat{q}_m||, m = 1, ..., M$.

Improved Triplet-like loss. In order to ensure fast convergence it is crucial to select triplets that violate the triplet constraint in Equation 6.7. The traditional triplet loss interacts with only one negative sample (and equivalently one negative class) for each update in the network, while we actually need to compare the query image with multiple

different classes in few-shot classification. Hence, the triplet loss may not be effective for the feature embedding learning, particularly when we have several classes to handle in the few-shot classification setting. Inspired by [6], [227], we generalize the traditional triplet loss with E-negatives prototypes to allow simultaneous comparisons jointly with the E negative prototypes instead of just one negative prototype, in one mini-batch. This extension makes the feature comparison more effective and faithful to the few-shot classification procedure, since in each update, the network can compare a sample with multiple negative classes.

In particular, we randomly choose the E negative prototypes $\hat{q}^{n_e}, e = \{1, 2, ..., E\}$ to form into a triplet. Accordingly, the optimization objective evolves to:

$$\begin{aligned}\mathcal{L}(q_m^a, \hat{q}_m^p, \hat{x}_m^n) = \sum_{m=1}^{MN} \frac{1}{E} \sum_{e=1}^{E} \big[||f(q_m^a) - f(\hat{q}_m^p))||_2^2 \\ - ||f(q_m^a) - f(\hat{q}_m^{n_e})||_2^2 + \alpha \big]_+\end{aligned} \tag{6.9}$$

For the sample q_m^a in the query set, the optimization shall maximize the distance to the negative prototype q_m^n to be larger than the distance to the positive prototypes q_m^p in the feature space. For each anchor sample q_m^a, we then learn the positive prototype q_m^p from the support set of the same class as q_m^a and further randomly select E other negative prototypes whose classes are different from q_m^a. Compared with the traditional triplet loss, each forward update in our improved Triplet-like loss includes more inter-class variations, thus making the learnt feature embedding more discriminative for samples from different classes.

Mining hard triplets is an important part of metric learning with the triplet loss, as otherwise training will soon stagnate [91]. This is because when the model begins to converge, the embedding space learns how to correctly map the triples relatively quickly. Thus most triples satisfying the margin will not contribute to the gradient in the learning process. To speed up the convergence and stabilize the training procedure, we propose a new hard-triplet mining strategy to sample more informative hard triplets in each episode. Specifically, triplets will be randomly selected in each episode as described above, we then check whether the sampled triplets satisfy the margin. The triplets that have already met the margin will be removed and the network training will proceed with the remaining triplets.

6.4 Experiments

Extensive experiments have been conducted to evaluate and compare the proposed method for few-shot classification using on three challenging few-shot classification benchmarks datasets, *mini*ImageNet [254], *tiered*ImageNet [199] and Fewshot-CIFAR100 (FC100) [183]. All the experiments are implemented based on PyTorch and run with NVIDIA 2080ti GPUs.

6.4.1 Datasets

*mini***ImageNet** is the most popular few-shot classification benchmark proposed by [254] and derived from the original ILSVRC-12 dataset [205]. It contains 100 randomly sampled different categories, each with 600 images of size 84×84 pixels. The *tiered***ImageNet** [199] is a larger subset of ILSVRC-12 [205] with 608 classes and 779,165 images in total. The classes in *tiered*ImageNet are grouped into 34 categories corresponding to higher-level nodes in the ImageNet hierarchy curated by humans [42]. Each hierarchical category contains 10 to 20 classes, which are divided into 20 training (351 classes), 6 validation (97 classes) and 8 test (160 classes) categories. **Fewshot-CIFAR100 (FC100)** is based on the popular object classification dataset CIFAR100 [126]. Oreshkin et al. [183] offer a more challenging class split of CIFAR100 for few-shot classification. The FC100 further groups the 100 classes into 20 superclasses. Thus the training set has 60 classes belonging to 12 superclasses, the validation and test data consist of 20 classes each belonging to 5 superclasses each.

6.4.2 Implementation Details

Following the general few-shot classification experiment settings [226], [236], we conducted 5-way 5-shot and 5-way 1-shot classifications. The Adam optimizer is exploited with an initial learning rate of 0.001. The total training episodes on *mini*ImageNet, *tiered*ImageNet and FC100 are 600,000, 1,000,000 and 1,000,000, respectively. The learning rate is dropped by 10% every 100,000 episodes or when the loss enters a plateau. The weight decay is set to 0.0003. We report the mean accuracy (%) over 600 randomly generated episodes from the test set.

Few-shot classification method	5-Way 1-Shot	5-Way 5-Shot
Matching Networks [254]	43.56 ± 0.84	55.31±0.73
MAML [63]	48.70±1.84	63.11±0.92
Relation Net [236]	50.44±0.82	65.32±0.70
REPTILE [178]	49.97±0.32	65.99±0.58
Prototypical Net [226]	49.42±0.78	68.20±0.66
Predict Params [192]	59.60±0.41	73.74 ± 0.19
LwoF [75]	60.06±0.14	76.39 ± 0.11
TADAM [183]	58.50±0.30	76.70±0.30
EGNN [117]	–	66.85
EGNN+Transduction [117]	–	76.37
CTM [141]	62.05±0.55	78.63±0.06
wDAE-GNN [76]	62.96±0.15	78.85±0.10
MetaOptNet-SVM-trainval [136]	64.09±0.62	80.00±0.45
CTM, data augment [141]	64.12±0.82	80.51±0.13
Baseline	59.71±0.35	75.21±0.43
Ours	63.23±0.26	80.17±0.33
Ours, data augment	**66.43±0.26**	**82.13±0.21**

Table 6.1: Few-shot classification accuracies (%) on *mini*ImageNet.

6.4.3 Results Evaluation

Comparison with the baseline model. Using the training/testing data split and the procedure described in Section 6.3, the baseline in Table 6.1, Table 6.2 and Table 6.3 evaluate a model with modified DeepCaps, without the attentive prototype. The accuracy is 75.21±0.43%, 78.41±0.34% and 59.8±1.0% and in the 5-way 5-shot setting on *mini*ImageNet, *tiered*ImageNet and FC100 respectively. Our baseline results are on a par with those reported in [236], [226]. As shown in Table 6.1, Table 6.2 and Table 6.3, using the attentive prototype strategy in the model training with improved triplet-like loss, our method significantly improves the accuracy on all three datasets. There are obvious improvements of approximately +4.96% (from 75.21% to 80.17%), +4.83% (from 78.41% to 83.24%), +2.5% (from 57.3% to 59.8%) under the 5-way 5-shot setting for *mini*ImageNet, *tiered*ImageNet and FC100, respectively. These results indicate that the proposed approach is tolerant to large intra- and inter-class variations and produces marked improvements over the baseline.

Comparison with the state-of-the-art methods. We also compare our method with some state-of-the-art methods on *mini*ImageNet, *tiered*ImageNet in Table 6.1 and

Few-shot classification method	5-Way 1-Shot	5-Way 5-Shot
MAML [63]	51.67±1.81	70.30±0.08
Meta-SGD [144], reported by [206]	62.95±0.03	79.34±0.06
LEO [206]	66.33±0.05	81.44±0.09
Relation Net [236]	54.48±0.93	71.32±0.78
Prototypical Net [226]	53.31±0.89	72.69±0.74
EGNN [117]	–	70.98
EGNN+Transduction [117]	–	80.15
CTM [141]	64.78±0.11	81.05±0.52
MetaOptNet-SVM-trainval [136]	65.81±0.74	81.75±0.53
CTM, data augmention [141]	68.41±0.39	84.28±1.73
Baseline	63.25±0.31	78.41±0.34
Ours	65.53±0.21	83.24±0.18
Ours, data augmention	**69.87±0.32**	**86.35±0.41**

Table 6.2: Few-shot classification accuracies (%) on *tiered*ImageNet.

Table 6.2, respectively. On *mini*ImageNet, we achieve a **5-way 1-shot accuracy =63.23±0.26, 5-way 5-shot accuracy =80.17 ± 0.33%** when using the proposed method, which has a highly competitive performance compared with the state-of-the-art. On *tiered*ImageNet, we arrive at **5-way 1-shot accuracy = 65.53±0.21, 5-way 5-shot accuracy =83.24 ± 0.18%** which is also very competitive. The previous best result was produced by introducing a Category Traversal Module [141] and data augmention that can be inserted as a plug-and-play module into most metric-learning based few-shot learners. We further investigate whether the data augmention could work on our model. By training a version of our model with basic data augmentation, we obtain the improved results **5-way 5-shot accuracy = 82.13±0.21%** on *mini*ImageNet. On *tiered*ImageNet, we also observe a performance **5-way 5-shot accuracy = 86.35±0.41%**.

For the FC100 dataset, our proposed method is superior to all the other methods [63], [183], [233] in accuracy. The comparisons consistently confirm the competitiveness of the proposed method on few-shot image classification. In terms of size and computational cost, for the models trained on mini-ImageNet, the proposed model has only 7.22 million parameters, while the ResNet-18 used in the existing SOTA approach has 33.16 million parameters. We also tested both models' inference time, ResNet-18 takes 3.65 ms for a 64 × 64 ×3 image, while our model takes only 1.67 ms for a 64 × 64 ×3 image. In summary, our proposed attentive prototype learning scheme improve over the previous methods, mainly

Few-shot classification method	5-Way 1-Shot	5-Way 5-Shot	5-Way 10-Shot
MAML [63]	38.1±1.7	50.4±1.0	56.2±0.8
TADAM [183]	40.1±0.4	56.1±0.4	61.6±0.5
MTL [233]	45.1±1.8	57.6±0.9	63.4±0.8
Baseline	44.2±1.3	57.3±0.8	62.8±0.6
Ours	**47.5±0.9**	**59.8±1.0**	**65.4±0.5**

Table 6.3: Few-shot classification accuracies (%) on the FC100 dataset.

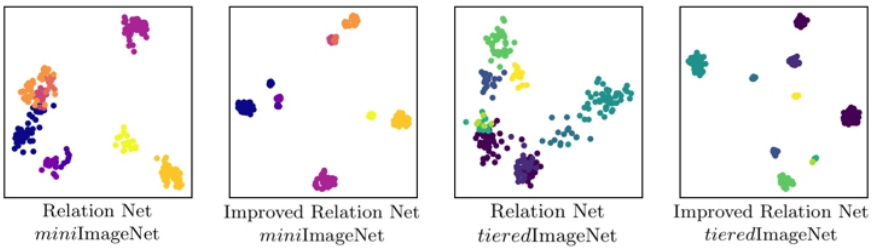
Few-shot classification method	*mini*ImageNet		*tiered*ImageNet	
	5-Way 5 shot	10-Way 5 shot	5-Way 5-shot	10-Way 5-shot
Prototypical Net [226]	68.20	-	72.69	-
Ours (average mechanism)	76.32	58.41	80.31	62.17
Ours (attentive prototype)	80.17	63.12	83.24	66.33
Relation Net [236]	65.32	–	71.32	–
Relation Net [236] (our implementation)	80.91	64.34	83.98	67.86

Table 6.4: Ablation study on the attentive prototype and embedding module.

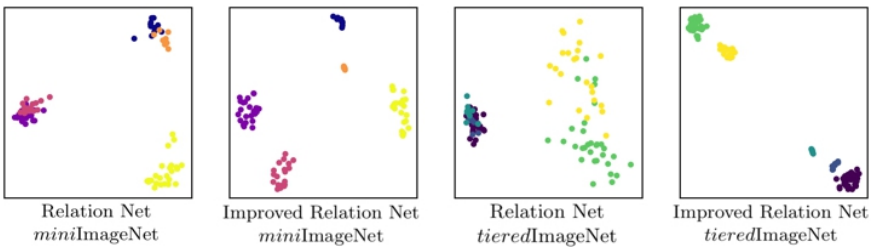
due to the better embedding space provided by the capsule network and the attentive prototyping scheme. The importance value is used as the weighting value for the support set instances, which is completely dependent on the affinity relationship between the two feature points from the support set and the query. The importance weighting values vary exponentially, with larger value reflecting nearby pairs of feature points and a smaller value for the distant pair. This conforms that the feature points from the support set that are nearer to the query feature point should be given more attention.

Ablation study: To verify the effectiveness of components in the proposed method, we conducted ablation experiments on the *mini*ImageNet and *tiered*ImageNet datasets. First, to investigate the contribution of the designed attentive prototype method, we compare the performance of the proposed method with vanilla prototypical networks [226]. Then, we verify the effectiveness of our proposed feature embedding module by embedding it into the metric-based algorithm Relation Net [236].Table 6.4 summarizes the performance of the different variants of our method.

1)*Attentive prototype:* In vanilla prototypical networks [226], the prototypes are defined as the averages the embed features of each class in the support set. Such a simple class-wise



(a) 5-way 5-shot setting



(b) 10-way 5-shot setting

Figure 6.3: The t-SNE visualization [164] of the improved feature embeddings learnt by our proposed approach..

feature takes all instances into consideration equally. Our attentive prototype scheme is a better replacement. A variant of DeepCaps is applied with improved triplet-like loss to learn the feature embedding instead of a shallow CNN network. To further verify the effectiveness of our attentive prototype, we also compared the average-based prototypes created from our embedding framework. The experimental results on *mini*ImageNet and *tiered*ImageNet are summarized in Table 6.4. It can be observed that the attentive prototype gains an approximately 3%-4% increase after replacing the average mechanism. This shows that the attentive prototypes can be more 'typical' when compared to the original average vectors by giving different weights for different instances.

2)*Embedding module:* The embedding is switched from four convolutional blocks in Relation Net [236] to the modified DeepCaps model and the supervision loss is changed to the improved triplet-like loss. Table 6.4 shows the results obtained by the improvements over the Relation Net. We find that the improved Relation Net exceeds the original model by approximately +10%. This shows the ability of the proposed capsule network-based embedding network to improve the performance of the metric based method. Figure 6.3 visualizes the feature distribution using t-SNE [164] for the features computed in 5-way

Few-shot classification method	5-Way 1-Shot	5-Way 5-Shot
Setting-1	59.71±0.35	75.21±0.43
Setting-2	61.76±0.12	78.45±0.23
Setting-3	63.23±0.26	80.17±0.33

Table 6.5: Few-shot classification accuracies (%) on *mini*ImageNet.

5-shot setting and 10-way 5-shot setting. As can be clearly observed, the improved Relation Net model has more compact and separable clusters, indicating that features are more discriminative for the task. This is caused by the design of the embedding module.

3)*Improved Triplet-like loss:* To help analyze our model and show the benefit of improved Triplet-like loss, we design several comparison methods as follows: Setting-1: Baseline model (modified DeepCaps); Setting-2: Using the attentive prototype strategy in the model training; Setting-3: Based on the Setting 2, we add the improved triplet-like loss to make the feature comparison more effective. With the help of improved triplet-like loss, we observed an improvement of +1.5% as shown in Table 6.5. Thus making the learnt feature embedding more discriminative for samples from different classes.

6.5 Conclusion

In this paper, we proposed a new few-shot classification scheme aiming to improve the metric learning-based prototypical network. Our proposed scheme has the following novel characteristics: (1) a new embedding space created by a capsule network, which is unique in its capability to encode the relative spatial relationship between features. The network is trained with a novel triple-loss designed to learn the embedding space; (2) an effective and robust non-parameter classification scheme, named attentive prototypes, to replace the simple feature average for prototypes. The instances from the support set are taken into account to generate prototypes, with their importance being calculated by the reconstruction error for a given query. Experimental results showed that the proposed method outperforms the other few-shot classification algorithms on all of the miniImageNet, tieredImageNet and FC100 datasets. However, the proposed method in this chapter merely consider instant pairwise query-support relationships but fail to explore support-support relationships among the labelled support samples, let alone that among the unlabeled ones. In the future, we will explicitly explore the relationships between

each two samples and propose to propagate information from relevant samples for feature embedding enhancement.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

A summary of the conclusions of this thesis is given below:

- In Chapter 1, we first give an overview of the research topics, followed by the motivations and challenges in the research topics and then present the general architecture of the thesis and summarise the main contributions of the thesis.
- The recent development of machine learning and its applications are introduced, followed by a description of relevant deep learning theories and the recent development of the representation learning. In Chapter 2, We also presents a comprehensive review of the recent literatures of the related deep learning models.
- In Chapter 3, a deep multiple classifier fusion method based on granular computing is proposed to create information granulation and multi-level of granularity. Especially, the unified end-to-end deep network is built to integrate all algorithmic components, which makes the training process efficient and effective. We collect a new traffic scene dataset, named the 'WZ-traffic'. It consists of 6,035 labeled images which belong to 20 categories collected from both an image search engine as well as from personal photographs. This method achieved the state-of-the-art results on two benchmark datasets: WZ-traffic and FM2 dataset.
- In Chapter 4, we propose a semi-supervised learning system based on the Convolutional Neural Network (CNN) and re-ranking strategy for Vehicle re-ID. Specifically,

we adopt the structure of Generative Adversarial Network (GAN) to obtain more vehicle images and enrich the training set, then a uniform label distribution will be assigned to the unlabeled samples according to the Label Smoothing Regularization for Outliers (LSRO), which regularizes the supervised learning model and improves the performance of re-ID. To optimize the re-ID results, an improved re-ranking method is exploited to optimize the initial rank list. Experimental results on publically available datasets, VeRi-776, VehicleID and VehicleReID, demonstrate that the method significantly outperforms the state- of-the-art.

- In Chapter 4, a Generative Adversarial Network (GAN) was adopted to generate unlabeled samples and enlarge the training set. A semi-supervised learning scheme with the Convolutional Neural Networks (CNN) was proposed accordingly, which assigns a uniform label distribution to the unlabeled images to regularize the supervised model and improve the performance of the vehicle re-ID system. Besides, an improved re-ranking method based on Jaccard distance and k-reciprocal nearest neighbors is proposed to optimize the initial rank list. Extensive experiments over the benchmark datasets VeRi-776, VehicleID and VehicleReID have demonstrated that the proposed method outperforms the state-of-the-art approaches for vehicle re-ID.

- In Chapter 5, we address the face recognition problem under different variations, including disguise accessories, illumination and pose. For the disguised face recognition, we propose a novel Unsupervised Domain Adaptation Model (UDAM), which jointly transfer the rich knowledge from the source domain and discriminative representation end-to-end that mutually boost each other to achieve the disguised face recognition of target domain. For the near infrared face recognition, we first propose to apply the image-image translation method to enhance the performance of near-infrared facial image recognition. Using this approach, we reduce the intra-personal difference caused by the completely different illumination. Therefore, we can achieve much better recognition results by applying the existing pre-trained VLD deep neural network face recognition model. For the pose-robust face recognition (PRFR), we aim to learn face representations equivariant to pose variations and propose a novel meta learning framework. It is desirable that if the input image is transformed, e.g., by an out-of-plane rotation or pose variation, the learnt model should make predictions in a co-transformational way. Using this approach, we can actually map

the features of the input image with arbitrary poses to the front space through the mapping function. With the help of meta learning, it helps the model to adapt to new tasks efficiently via extracting useful transferable knowledge from a set of auxiliary tasks.

- In Chapter 6, we further work on the well-known few-shot learning method known as prototypical networks for better performance. Our contributions include (1) a new embedding structure to encode relative spatial relationships between features by applying capsule network; (2) a new triplet loss designated to enhance the semantic feature embedding where similar samples are close to each other while dissimilar samples are farther apart; and (3) an effective nonparametric classifier termed attentive prototypes in place of the simple prototypes in current few-shot learning. The proposed attentive prototype aggregates all of the instances in a support class which are weighted by their importance defined by the reconstruction error for a given query. The reconstruction error allows the classification posterior probability to be estimated, which corresponds to the classification confidence score. Extensive experiments on three benchmark datasets demonstrate that our approach is effective for the few-shot classification task.

- This thesis comprehensively studies the recent development of representation learning in computer vision and deep learning. In four application cases: the traffic scene recognition, vehicle re-identification, face recognition under uncontrolled environments and few-shot learning, the representation learning methods have shown powerful capability of extracting useful information. Also, several related research topics have been discussed, including the granular computing, semi-supervised learning, domain adaptation and meta-learning.

7.2 Future work

- **Video analysis.** The applications of computer vision in this thesis center on images, with less focused on sequences of images (i.e. video frames). We will garner more attention on video-based tasks in the future. Video allows for deeper situational understanding, because sequences of images provide new information about action. Future works include the following tasks: 1) Abnormal Event Detection. Pedestrian abnormal event detection is an active research area to improve traffic safety for

intelligent transportation systems (ITS). We will propose an efficient method to automatically detect and track far-away pedestrians in traffic video to determine the abnormal behavior events. 2) Behavior Prediction. We will propose a representation learning method to track an obstacle through a sequence of images and understand its behavior to predict the next move.

- **Self-supervised representation learning.** The future research will study the self-supervised learning method that focuses on beneficial properties of representation and their abilities in generalizing to real-world tasks. Self-supervised representation learning is a promising subclass of unsupervised learning, which provides an opportunity for better utilizing unlabeled data by setting the learning objectives to learn from the internal cues. The feature representation obtained by self-supervision can be used in downstream tasks such as classification, object detection, segmentation, and anomaly detection.

- **Representation learning with 3D Data.** In this thesis, we solved many 2D computer vision tasks and achieved promising performance. The increasing abundance of 3D data encouraged us to exploit this richer content for addressing several computer vision problems related to understanding 3D scenes in the future. Indeed, the possibility of using the additionally provided attributes of depth and full 3D geometry represents an important advantage that can significantly boost the performance of several applications.

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] Hervé Abdi. Metric multidimensional scaling (mds): analyzing distance matrices. *Encyclopedia of measurement and statistics*, pages 1–13, 2007.

[3] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[4] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *American Statistician*, 46(3):175–185, 1992.

[5] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016.

[6] Sercan O Arik and Tomas Pfister. Attention-based prototypical learning towards interpretable, confident and robust deep neural networks. *arXiv preprint arXiv:1902.06292*, 2019.

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[8] Krassimir T Atanassov. Interval valued intuitionistic fuzzy sets. In *Intuitionistic Fuzzy Sets*, pages 139–177. Springer, 1999.

[9] Francis R Bach and Michael I Jordan. Kernel independent component analysis. *Journal of machine learning research*, 3(Jul):1–48, 2002.

[10] Yan Bai, Yihang Lou, Feng Gao, Shiqi Wang, Yuwei Wu, and Lingyu Duan. Group sensitive triplet embedding for vehicle re-identification. *IEEE Transactions on Multimedia*, 2018.

[11] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[12] Y Bengio, Y LeCun, et al. Scaling learning algorithms towards ai larg. *Scale Kernel Mach*, 34(5), 2007.

[13] Yoshua Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[14] 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.

[15] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. *Emnlp*, pages 120–128, 2006.

[16] Mohamed Anouar Borgi, Demetrio Labate, Maher El Arbi, and Chokri Ben Amar. Sparse multi-regularized shearlet-network using convex relaxation for face recognition. In *International Conference on Pattern Recognition*, 2014.

[17] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[18] Matthew Brand. Charting a manifold. In *Advances in neural information processing systems*, pages 985–992, 2003.

[19] Jorge Cadima and Ian T Jolliffe. Loading and correlations in the interpretation of principle compenents. *Journal of applied Statistics*, 22(2):203–214, 1995.

[20] Jie Cao, Yibo Hu, Hongwen Zhang, Ran He, and Zhenan Sun. Learning a high fidelity pose invariant model for high-resolution face frontalization. In *Advances in Neural Information Processing Systems*, pages 2867–2877, 2018.

[21] Kaidi Cao, Yu Rong, Cheng Li, Xiaoou Tang, and Chen Change Loy. Pose-robust face recognition via deep residual equivariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5187–5196, 2018.

[22] Xiaojun Chang and Yi Yang. Semisupervised feature analysis by mining correlations among multiple tasks. *IEEE transactions on neural networks and learning systems*, 28(10):2294–2305, 2017.

[23] Chao-Yeh Chen, Wongun Choi, and Manmohan Chandraker. Atomic scenes for scalable traffic scene recognition in monocular videos. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.

[24] Jun-Cheng Chen, Jingxiao Zheng, Vishal M Patel, and Rama Chellappa. Fisher vector encoded deep convolutional features for unconstrained face verification. In *IEEE International Conference on Image Processing*, pages 2981–2985. IEEE, 2016.

[25] Shyi-Ming Chen, Li-Wei Lee, Hsiang-Chuan Liu, and Szu-Wei Yang. Multiattribute decision making based on interval-valued intuitionistic fuzzy values. *Expert Systems with Applications*, 39(12):10343–10351, 2012.

[26] Shyi-Ming Chen and Kurniawan Tanuwijaya. Fuzzy forecasting based on high-order fuzzy logical relationships and automatic clustering techniques. *Expert Systems with Applications*, 38(12):15425–15437, 2011.

[27] Shyi-Ming Chen and Jeng-Yih Wang. Document retrieval using knowledge-based fuzzy information retrieval techniques. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(5):793–803, 1995.

[28] Yutian Chen, Matthew W Hoffman, Sergio Gómez Colmenarejo, Misha Denil, Timothy P Lillicrap, Matt Botvinick, and Nando De Freitas. Learning to learn without gradient descent by gradient descent. In *IEEE International Conference on Machine Learning*, pages 748–756. JMLR. org, 2017.

[29] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014.

[30] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *IEEE International Conference on Machine Learning*, pages 2990–2999, 2016.

[31] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

[32] Taco S Cohen and Max Welling. Transformation properties of learned visual representations. *arXiv preprint arXiv:1412.7659*, 2014.

[33] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

[34] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[35] Mickael Cormier, Lars Wilko Sommer, and Michael Teutsch. Low resolution vehicle re-identification based on appearance features for wide area motion imagery. In *Applications of Computer Vision Workshops (WACVW), 2016 IEEE Winter*, pages 1–7. IEEE, 2016.

[36] Corinna Cortes and Vladimir Vapnik. *Support-Vector Networks*. Kluwer Academic Publishers, 1995.

[37] Nate Crosswhite, Jeffrey Byrne, Chris Stauffer, Omkar Parkhi, Qiong Cao, and Andrew Zisserman. Template adaptation for face verification and identification. *Image and Vision Computing*, 79:35–48, 2018.

[38] Gabriela Csurka and Florent Perronnin. Fisher vectors: Beyond bag-of-visual-words image representations. In *International Conference on Computer Vision, Imaging and Computer Graphics*, pages 28–42. Springer, 2010.

[39] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, pages 1–2. Prague, 2004.

[40] Adele Cutler, D. Richard Cutler, and John R. Stevens. Random forests. *Machine Learning*, 45(1):157–176, 2004.

[41] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[42] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[43] Weijian Deng, Liang Zheng, Guoliang Kang, Yi Yang, Qixiang Ye, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person reidentification. In *CVPR*, page 6, 2018.

[44] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person reidentification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 6, 2018.

[45] T. I Dhamecha, A Nigam, R Singh, and M Vatsa. Disguise detection and face recognition in visible and thermal spectrums. In *International Conference on Biometrics*, pages 1–8, 2013.

[46] Tejas Indulal Dhamecha, Richa Singh, Mayank Vatsa, and Ajay Kumar. Recognizing disguised faces: Human and machine evaluation. *Plos One*, 9(7):e99212, 2014.

[47] Changxing Ding and Dacheng Tao. Robust face recognition via multimodal deep face representation. *IEEE Transactions on Multimedia*, 17(11):2049–2058, 2015.

[48] Shengyong Ding, Liang Lin, Guangrun Wang, and Hongyang Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 48(10):2993–3003, 2015.

[49] Mandar Dixit, Si Chen, Dashan Gao, Nikhil Rasiwasia, and Nuno Vasconcelos. Scene classification with semantic fisher vectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2974–2983, 2015.

[50] David L Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.

[51] Gauthier Doquire and Michel Verleysen. A graph laplacian based approach to semi-supervised feature selection for regression problems. *Neurocomputing*, 121:5–13, 2013.

[52] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.

[53] Robert PW Duin and David MJ Tax. Classifier conditional posterior probabilities. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 611–619. Springer, 1998.

[54] Paul Augustine Ejegwa. Distance and similarity measures for pythagorean fuzzy sets. *Granular Computing*, pages 1–14, 2018.

[55] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[56] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of machine learning research*, 11(2), 2010.

[57] Andreas Ess, Tobias Müller, Helmut Grabner, and Luc J Van Gool. Segmentation-based urban traffic scene understanding. In *BMVC*, volume 1, page 2, 2009.

[58] S Farokhi, UU Sheikh, J Flusser, SM Shamsuddin, and H Hashemi. Evaluating feature extractors and dimension reduction methods for near infrared face recognition systems. *Jurnal Teknologi*, 70:23–33, 2014.

[59] Li Fe-Fei et al. A bayesian approach to unsupervised one-shot learning of object categories. In *IEEE InternationalConference on Computer Vision (ICCV)*, pages 1134–1141, 2003.

[60] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 28(4):594–611, 2006.

[61] Marin Ferecatu and Hichem Sahbi. Multi-view object matching and tracking using canonical correlation analysis. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 2109–2112. IEEE, 2009.

[62] 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.

[63] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1126–1135, 2017.

[64] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[65] Yoav Freund and David Haussler. Unsupervised learning of distributions on binary vectors using two layer networks. In *Advances in neural information processing systems*, pages 912–919, 1992.

[66] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics and data analysis*, 38(4):367–378, 2002.

[67] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013.

[68] Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[69] Jorge Garcia, Niki Martinel, Christian Micheloni, and Alfredo Gardel. Person re-identification ranking optimisation by discriminant context information analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1305–1313, 2015.

[70] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

[71] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron) a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

[72] Harish Garg and Gagandeep Kaur. Novel distance measures for cubic intuitionistic fuzzy sets and their applications to pattern recognitions and medical diagnosis. *Granular Computing*, pages 1–16, 2018.

[73] Mohsen Ghafoorian, Nico Karssemeijer, Tom Heskes, Inge WM van Uden, Clara I Sanchez, Geert Litjens, Frank-Erik de Leeuw, Bram van Ginneken, Elena Marchiori, and Bram Platel. Location sensitive deep convolutional neural networks for segmentation of white matter hyperintensities. *Scientific Reports*, 7(1):5110, 2017.

[74] Youness Aliyari Ghassabeh, Frank Rudzicz, and Hamid Abrishami Moghaddam. Fast incremental lda feature extraction. *Pattern Recognition*, 48(6):1999–2012, 2015.

[75] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4367–4375, 2018.

[76] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[77] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[78] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Contextual action recognition with r* cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1080–1088, 2015.

[79] Leroy A Gondy, C. Rindflesch B Thomas, and Nave Bayes. Programs for machine learning. *Advances in Neural Information Processing Systems*, 79(2):937–944, 1993.

[80] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[81] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[82] Chao Gou, Kunfeng Wang, Yanjie Yao, and Zhengxi Li. Vehicle license plate recognition based on extremal regions and restricted boltzmann machines. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1096–1107, 2016.

[83] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[84] Michelle R Greene, Abraham P Botros, Diane M Beck, and Li Fei-Fei. What you see is what you expect: rapid scene understanding benefits from prior experience. *Attention, Perception, & Psychophysics*, 77(4):1239–1251, 2015.

[85] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: Challenge of recognizing one million celebrities in the real world. *Electronic Imaging*, 2016(11):1–6, 2016.

[86] R. Hadsell, S. Chopra, and Y. Lecun. Dimensionality reduction by learning an invariant mapping. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, pages 1735–1742, 2006.

[87] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[88] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[89] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[90] Mingjie He, Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. Deformable face net for pose invariant face recognition. *Pattern Recognition*, 100:107113, 2020.

[91] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

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

[93] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[94] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer, 2012.

[95] Geoffrey E Hinton et al. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986.

[96] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011.

[97] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[98] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[99] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[100] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[101] Timothy Huang, Daphne Koller, Jitendra Malik, G Ogasawara, B Rao, Stuart J Russell, and Joseph Weber. Automatic symbolic traffic scene analysis using belief networks. In *AAAI*, volume 94, pages 966–972, 1994.

[102] Wenhao Huang, Guojie Song, Haikun Hong, and Kunqing Xie. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2191–2201, 2014.

[103] Aapo Hyvärinen, Patrik O Hoyer, and Mika Inki. Topographic independent component analysis. *Neural computation*, 13(7):1527–1558, 2001.

[104] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[105] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[106] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[107] Anil K Jain and Stan Z Li. *Handbook of face recognition*. Springer, 2011.

[108] Ayush Jaiswal, Wael AbdAlmageed, Yue Wu, and Premkumar Natarajan. Capsulegan: Generative adversarial capsule network. In *ECCV*, pages 0–0, 2018.

[109] Nathalie Japkowicz, Stephen Jose Hanson, and Mark A Gluck. Nonlinear autoassociation is not equivalent to pca. *Neural computation*, 12(3):531–545, 2000.

[110] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[111] Herve Jegou, Hedi Harzallah, and Cordelia Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[112] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[113] Łukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. In *International Conference on Learning Representations*, 2018.

[114] Meina Kan, Shiguang Shan, Haihong Zhang, Shihong Lao, and Xilin Chen. Multi-view discriminant analysis. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):188–194, 2016.

[115] Bong-Nam Kang, Yonghyun Kim, Bongjin Jun, and Daijin Kim. Attentional feature-pair relation networks for accurate face recognition. In *IEEE International Conference on Computer Vision*, pages 5472–5481, 2019.

[116] Muhammad Sajjad Ali Khan, Saleem Abdullah, Asad Ali, Fazli Amin, and Khaista Rahman. Hybrid aggregation operators based on pythagorean hesitant fuzzy sets and their application to group decision making. *Granular Computing*, pages 1–14, 2018.

[117] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–20, 2019.

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

[119] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[120] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[121] 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, 2015.

[122] Arief Koesdwiady, Ridha Soua, and Fakhreddine Karray. Improving traffic flow prediction with weather information in connected cars: a deep learning approach. *IEEE Transactions on Vehicular Technology*, 65(12):9508–9517, 2016.

[123] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2288–2295. IEEE, 2012.

[124] Adam R Kosiorek, Sara Sabour, Yee Whye Teh, and Geoffrey E Hinton. Stacked capsule autoencoders. *arXiv preprint arXiv:1906.06818*, 2019.

[125] Josip Krapac, Jakob Verbeek, and Frédéric Jurie. Modeling spatial layout with fisher vectors for image categorization. In *2011 International Conference on Computer Vision*, pages 1487–1494. IEEE, 2011.

[126] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[127] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[128] Ludmila I Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transactions on pattern analysis and machine intelligence*, 24(2):281–286, 2002.

[129] Karric Kwong, Robert Kavaler, Ram Rajagopal, and Pravin Varaiya. Arterial travel time estimation based on vehicle re-identification using wireless magnetic sensors. *Transportation Research Part C: Emerging Technologies*, 17(6):586–606, 2009.

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

[131] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[132] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[133] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[134] Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988.

[135] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013.

[136] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[137] Li-Wei Lee and Shyi-Ming Chen. Fuzzy risk analysis based on fuzzy numbers with different shapes and different deviations. *Expert Systems with Applications*, 34(4):2763–2771, 2008.

[138] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 991–999, 2015.

[139] Qingming Leng, Ruimin Hu, Chao Liang, Yimin Wang, and Jun Chen. Person re-identification with content and context re-ranking. *Multimedia Tools and Applications*, 74(17):6989–7014, 2015.

[140] Jan Eric Lenssen, Matthias Fey, and Pascal Libuschewski. Group equivariant capsule networks. In *Advances in neural information processing systems (NIPS)*, pages 8844–8853, 2018.

[141] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10, 2019.

[142] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Attention transfer from web images for video recognition. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1–9. ACM, 2017.

[143] Wei Li, Yang Wu, Masayuki Mukunoki, and Michihiko Minoh. Common-near-neighbor analysis for person re-identification. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 1621–1624. IEEE, 2012.

[144] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

[145] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. GeoMAN: multi-level attention networks for geo-sensory time series prediction. In *IJCAI*, 2018.

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

[147] Wei-Hua Lin and Daoqin Tong. Vehicle re-identification with dynamic time windows for vehicle passage time estimation. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1057–1063, 2011.

[148] Guodong Liu, Shuai Zhang, and Zhihua Xie. A novel infrared and visible face fusion recognition method based on non-subsampled contourlet transform. In *CISP-BMEI*, pages 1–6. IEEE, 2017.

[149] Han Liu and Mihaela Cocea. Granular computing-based approach of rule learning for binary classification. *Granular Computing*, pages 1–9, 2018.

[150] Han Liu and Alexander Gegov. *Collaborative Decision Making by Ensemble Rule Based Classification Systems*. Springer International Publishing, 2015.

[151] Han Liu, Alexander Gegov, and Mihaela Cocea. Rule-based systems: a granular computing perspective. *Granular Computing*, 1(4):259–274, 2016.

[152] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[153] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pages 2167–2175, 2016.

[154] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.

[155] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 469–477. Curran Associates, Inc., 2016.

[156] Xihui Liu, Haiyu Zhao, Maoqing Tian, Lu Sheng, Jing Shao, Shuai Yi, Junjie Yan, and Xiaogang Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. *arXiv preprint arXiv:1709.09930*, 2017.

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

[158] 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. Springer, 2016.

[159] Yu Liu, Junjie Yan, and Wanli Ouyang. Quality aware network for set to set recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5790–5799, 2017.

[160] Yuanyuan Liu, Jiabei Zeng, Shiguang Shan, and Zhuo Zheng. Multi-channel pose-aware convolution neural networks for multi-view facial expression recognition. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 458–465. IEEE, 2018.

[161] David G Lowe et al. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, pages 1150–1157, 1999.

[162] Keyu Lu, Jian Li, Xiangjing An, and Hangen He. Generalized haar filter based deep networks for real-time object detection in traffic scene. *arXiv preprint arXiv:1610.09609*, 2016.

[163] Andy Jinhua Ma and Ping Li. Query based adaptive re-ranking for person re-identification. In *Asian Conference on Computer Vision*, pages 397–412. Springer, 2014.

[164] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[165] Prasenjit Mandal and AS Ranadive. Hesitant bipolar-valued fuzzy sets and bipolar-valued hesitant fuzzy sets and their applications in multi-attribute group decision making. *Granular Computing*, pages 1–25, 2018.

[166] Niki Martinel, Abir Das, Christian Micheloni, and Amit K Roy-Chowdhury. Temporal model adaptation for person re-identification. In *European Conference on Computer Vision*, pages 858–877. Springer, 2016.

[167] Iacopo Masi, Stephen Rawls, Gérard Medioni, and Prem Natarajan. Pose-aware face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4838–4846, 2016.

[168] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[169] Luc Mioulet, Toby P Breckon, Andre Mouton, Haichao Liang, and Takashi Morie. Gabor features for real-time road environment classification. In *Industrial Technology (ICIT), 2013 IEEE International Conference on*, pages 1117–1121. IEEE, 2013.

[170] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

[171] Yoshihiro Mitani and Yoshihiko Hamamoto. A local mean-based nonparametric classifier. *Pattern Recognition Letters*, 27(10):1151–1159, 2006.

[172] Aryan Mobiny and Hien Van Nguyen. Fast capsnet for lung cancer screening. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 741–749. Springer, 2018.

[173] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *IEEE International Conference on Machine Learning*, pages 2554–2563. JMLR. org, 2017.

[174] Abdulmajid Murad and Jae-Young Pyun. Deep recurrent neural networks for human activity recognition. *Sensors*, 17(11):2556, 2017.

[175] Raymond H Myers and Raymond H Myers. *Classical and modern regression with applications*, volume 2. Duxbury press Belmont, CA, 1990.

[176] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[177] Loris Nanni and Alessandra Lumini. Heterogeneous bag-of-features for object/scene recognition. *Applied Soft Computing*, 13(4):2171–2178, 2013.

[178] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[179] Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

[180] Christopher Olah. Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/. accessed: 2018-09-14.

[181] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[182] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[183] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in neural information processing systems (NIPS)*, pages 721–731, 2018.

[184] Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2865–2873, 2015.

[185] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015.

[186] Massimiliano Patacchiola and Angelo Cangelosi. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, 71:132–143, 2017.

[187] Sateesh Pedagadi, James Orwell, Sergio Velastin, and Boghos Boghossian. Local fisher discriminant analysis for pedestrian re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3318–3325, 2013.

[188] W Pedrycz. Information granules and their use in schemes of knowledge management. *Scientia Iranica*, 18(3):602–610, 2011.

[189] Witold Pedrycz and Shyi-Ming Chen. *Granular computing and decision-making: interactive and iterative approaches*, volume 10. Springer, 2015.

[190] Hao Peng, Jianxin Li, Qiran Gong, Senzhang Wang, Lifang He, Bo Li, Lihong Wang, and Philip S Yu. Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification. *arXiv preprint arXiv:1906.04898*, 2019.

[191] Pedro O Pinheiro and Ronan Collobert. Weakly supervised semantic segmentation with convolutional networks. In *CVPR*, page 6. Citeseer, 2015.

[192] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7229–7238, 2018.

[193] Danfeng Qin, Stephan Gammeter, Lukas Bossard, Till Quack, and Luc Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 777–784. IEEE, 2011.

[194] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[195] Jathushan Rajasegaran, Vinoj Jayasundara, Sandaru Jayasekara, Hirunima Jayasekara, Suranga Seneviratne, and Ranga Rodrigo. Deepcaps: Going deeper with capsule networks. In *CVPR*, pages 10725–10733, 2019.

[196] Marc'Aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann L Cun. Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144, 2007.

[197] David Rawlinson, Abdelrahman Ahmed, and Gideon Kowadlo. Sparse unsupervised capsules generalize better. *arXiv preprint arXiv:1804.06094*, 2018.

[198] Christopher Reale, Nasser M Nasrabadi, Heesung Kwon, and Rama Chellappa. Seeing the forest from the trees: A holistic approach to near-infrared heterogeneous face recognition. In *CVPRW*, pages 54–62, 2016.

[199] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations (ICLR)*, 2018.

[200] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[201] Slobodan Ribaric, Aladdin Ariyaeeinia, and Nikola Pavesic. De-identification for privacy protection in multimedia content: A survey. *Signal Processing: Image Communication*, 47:131–151, 2016.

[202] Herbert Robbins and Sutton Monro. A stochastic approximation method. In *Herbert Robbins Selected Papers*, pages 102–109. Springer, 1985.

[203] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[204] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[205] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet

large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[206] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations (ICLR)*, 2018.

[207] Sara Sabour, Nicholas Frosst, and G Hinton. Matrix capsules with em routing. In *International Conference on Learning Representations*, 2018.

[208] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

[209] Christos Sagonas, Yannis Panagakis, Stefanos Zafeiriou, and Maja Pantic. Robust statistical face frontalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3871–3879, 2015.

[210] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[211] Swami Sankaranarayanan, Azadeh Alavi, Carlos D Castillo, and Rama Chellappa. Triplet probabilistic embedding for face verification and clustering. In *IEEE International Conference on Biometrics Theory, Applications and Systems*, pages 1–8. IEEE, 2016.

[212] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *IEEE International Conference on Machine Learning*, pages 1842–1850, 2016.

[213] Uwe Schmidt and Stefan Roth. Learning rotation-aware features: From invariant priors to equivariant descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2050–2057. IEEE, 2012.

[214] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[215] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015.

[216] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[217] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017.

[218] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1–9. IEEE, 2016.

[219] 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. *arXiv preprint arXiv:1708.03918*, 2017.

[220] Ivan Sikiric, Karla Brkic, Josip Krapac, and Sinisa Segvic. Image representations on a budget: Traffic scene classification in a restricted bandwidth scenario. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 845–852. IEEE, 2014.

[221] Ivan Sikirić, Karla Brkić, and Siniša Šegvić. Classifying traffic scenes using the gist image descriptor. *arXiv preprint arXiv:1310.0316*, 2013.

[222] Vin D Silva and Joshua B Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems*, pages 721–728, 2003.

[223] Marcel Simon, Erik Rodner, and Joachim Denzler. Part detector discovery in deep convolutional neural networks. In *Asian Conference on Computer Vision*, pages 162–177. Springer, 2014.

[224] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[225] Amarjot Singh, Devendra Patil, G Meghana Reddy, and Sn Omkar. Disguised face identification (dfi) with facial keypoints using spatial fusion convolutional network.

In *IEEE International Conference on Computer Vision Workshop*, pages 1648–1655, 2017.

[226] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems (NIPS)*, pages 4077–4087, 2017.

[227] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems (NIPS)*, pages 1857–1865, 2016.

[228] Kihyuk Sohn and Honglak Lee. Learning invariant representations with local transformations. *arXiv preprint arXiv:1206.6418*, 2012.

[229] Xinhang Song, Shuqiang Jiang, and Luis Herranz. Joint multi-feature spatial context for scene recognition on the semantic manifold. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1312–1320, 2015.

[230] Xuan Song, Hiroshi Kanasugi, and Ryosuke Shibasaki. DeepTransport: prediction and simulation of human mobility and transportation mode at a citywide level. In *IJCAI*, pages 2618–2624, 2016.

[231] Nitish Srivastava, 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.

[232] Wanchen Sui, Xinxiao Wu, Yang Feng, and Yunde Jia. Heterogeneous discriminant analysis for cross-view action recognition. *Neurocomputing*, 191:286–295, 2016.

[233] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 403–412, 2019.

[234] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014.

[235] Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. Translation modeling with bidirectional recurrent neural networks. In *Proceedings of the 2014*

Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 14–25, 2014.

[236] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1199–1208, 2018.

[237] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62, 1997.

[238] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

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

[240] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[241] Syed Fahad Tahir and Andrea Cavallaro. Low-cost multi-camera object matching. In *IEEE International Conference on Acoustics*, 2014.

[242] 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, 2014.

[243] Isabelle Tang and Toby P Breckon. Automatic road environment classification. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):476–484, 2011.

[244] Phillip Taylor, Nathan Griffiths, Abhir Bhalerao, Sarabjot Anand, Thomas Popham, Zhou Xu, and Adam Gelencser. Data mining for vehicle telemetry. *Applied Artificial Intelligence*, 30(3):233–256, 2016.

[245] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[246] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[247] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition*, pages 1521–1528, 2011.

[248] Rizoan Toufiq and Md. Rabiul Isalm. Face recognition system using soft-output classifier fusion method. In *International Conference on Electrical, Computer and Telecommunication Engineering*, pages 1–4, 2017.

[249] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1415–1424, 2017.

[250] Diego Uribe. Domain adaptation in sentiment classification. In *Ninth International Conference on Machine Learning and Applications*, pages 857–860, 2011.

[251] Gijs van Tulder and Marleen de Bruijne. Combining generative and discriminative representation learning for lung ct analysis with convolutional restricted boltzmann machines. *IEEE transactions on medical imaging*, 35(5):1262–1272, 2016.

[252] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015.

[253] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[254] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems (NIPS)*, pages 3630–3638, 2016.

[255] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3360–3367. Citeseer, 2010.

[256] Ling Wang and Hichem Sahbi. Nonlinear cross-view sample enrichment for action recognition. In *European Conference on Computer Vision*, pages 47–62. Springer, 2014.

[257] David Warde-Farley, Ian J Goodfellow, Aaron Courville, and Yoshua Bengio. An empirical analysis of dropout in piecewise linear networks. *arXiv preprint arXiv:1312.6197*, 2013.

[258] 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. Springer, 2016.

[259] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5028–5037, 2017.

[260] Fang-Yu Wu, Shi-Yang Yan, Jeremy S Smith, and Bai-Ling Zhang. Traffic scene recognition based on deep cnn and vlad spatial pyramids. In *Machine Learning and Cybernetics (ICMLC), 2017 International Conference on*, volume 1, pages 156–161. IEEE, 2017.

[261] Fangyu Wu. WZ-traffic dataset. `https://github.com/Fangyu0505/traffic-scene-recognition`, 2019. [Online; accessed 10-March-2019].

[262] Fangyu Wu, Shiyang Yan, Jeremy S Smith, and Bailing Zhang. Vehicle re-identification in still images: Application of semi-supervised learning and re-ranking. pages 278–283, 2018.

[263] Fangyu Wu, Shiyang Yan, Jeremy S Smith, and Bailing Zhang. Vehicle re-identification in still images: Application of semi-supervised learning and re-ranking. *Signal Processing: Image Communication*, 2019.

[264] Lin Wu, Chunhua Shen, and Anton van den Hengel. Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification. *Pattern Recognition*, 65:238–250, 2017.

[265] Edgar Xi, Selina Bing, and Yang Jin. Capsule network performance on complex data. *arXiv preprint arXiv:1712.03480*, 2017.

[266] Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S Yu. Zero-shot user intent detection via capsule neural networks. *arXiv preprint arXiv:1809.00385*, 2018.

[267] Yong Xu, Aini Zhong, Jian Yang, and David Zhang. Bimodal biometrics based on a representation and recognition approach. *Optical Engineering*, 50(3):183–183, 2011.

[268] Shiyang Yan, Jeremy S. Smith, and Bailing Zhang. Action recognition from still images based on deep vlad spatial pyramids. *Signal Processing Image Communication*, 54:118–129, 2017.

[269] JingTao Yao. Information granulation and granular relationships. In *2005 IEEE International Conference on Granular Computing*, volume 1, pages 326–329. IEEE, 2005.

[270] Yiyu Yao. Perspectives of granular computing. In *2005 IEEE international conference on granular computing*, volume 1, pages 85–90. IEEE, 2005.

[271] Mang Ye, Jun Chen, Qingming Leng, Chao Liang, Zheng Wang, and Kaimin Sun. Coupled-view based ranking optimization for person re-identification. In *International Conference on Multimedia Modeling*, pages 105–117. Springer, 2015.

[272] Mang Ye, Chao Liang, Yi Yu, Zheng Wang, Qingming Leng, Chunxia Xiao, Jun Chen, and Ruimin Hu. Person reidentification via ranking aggregation of similarity pulling and dissimilarity pushing. *IEEE Transactions on Multimedia*, 18(12):2553–2566, 2016.

[273] Xi Yin and Xiaoming Liu. Multi-task convolutional neural network for pose-invariant face recognition. *IEEE Transactions on Image Processing*, 27(2):964–975, 2017.

[274] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[275] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[276] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[277] Dominik Zapletal and Adam Herout. Vehicle re-identification for automatic video traffic surveillance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–31, 2016.

[278] Thiago H. H. Zavaschi, Alceu S. Britto Jr, Luiz E. S. Oliveira, and Alessandro L. Koerich. Fusion of feature sets and classifiers for facial expression recognition. *Expert Systems with Applications*, 40(2):646–655, 2013.

[279] Matthew D Zeiler. *Hierarchical convolutional deep learning in computer vision*. PhD thesis, New York University, 2013.

[280] Matthew D Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013.

[281] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014.

[282] Baochang Zhang, Lei Zhang, David Zhang, and Linlin Shen. Directional binary code with application to polyu near-infrared face database. *Pattern Recognition Letters*, 31(14):2337–2344, 2010.

[283] Chun-Yang Zhang, CL Philip Chen, Dewang Chen, and Kin Tek Ng. Mapreduce based distributed learning algorithm for restricted boltzmann machine. *Neurocomputing*, 198:4–11, 2016.

[284] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE Int. Conf. Comput. Vision (ICCV)*, pages 5907–5915, 2017.

[285] Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, and Cheng Chen. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1624–1639, 2011.

[286] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[287] Mi Zhang, Jie Tang, Xuchen Zhang, and Xiangyang Xue. Addressing cold start in recommender systems: A semi-supervised co-training algorithm. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 73–82. ACM, 2014.

[288] Wei Zhang, Ping Tang, and Lijun Zhao. Remote sensing image scene classification using cnn-capsnet. *Remote Sensing*, 11(5):494, 2019.

[289] XiaoQing Zhang and Shu-Guang Zhao. Cervical image classification based on image segmentation preprocessing and a capsnet network model. *International Journal of Imaging Systems and Technology*, 29(1):19–28, 2019.

[290] Yiheng Zhang, Dong Liu, and Zheng-Jun Zha. Improving triplet-wise training of convolutional neural network for vehicle re-identification. In *Multimedia and Expo (ICME), 2017 IEEE International Conference on*, pages 1386–1391. IEEE, 2017.

[291] Cairong Zhao, Xuekuan Wang, Duoqian Miao, Hanli Wang, Weishi Zheng, Yong Xu, and David Zhang. Maximal granularity structure and generalized multi-view discriminant analysis for person re-identification. *Pattern Recognition*, 79:79–96, 2018.

[292] Jian Zhao, Yu Cheng, Yan Xu, Lin Xiong, Jianshu Li, Fang Zhao, Karlekar Jayashree, Sugiri Pranata, Shengmei Shen, Junliang Xing, et al. Towards pose invariant face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2207–2216, 2018.

[293] Jianhua Zhao, LH Philip, and James T Kwok. Bilinear probabilistic principal component analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 23(3):492–503, 2012.

[294] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1009–1018, 2019.

[295] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. *arXiv preprint arXiv:1701.07717*, 2017.

[296] Yuanyi Zhong, Jiansheng Chen, and Bo Huang. Toward end-to-end face recognition through alignment learning. *IEEE signal processing letters*, 24(8):1213–1217, 2017.

[297] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. *arXiv preprint arXiv:1701.08398*, 2017.

[298] Zhi Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. Taylor and Francis, 2012.

[299] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.

[300] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.

[301] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.