DEPARTMENT OF COMPUTER SCIENCE

Re-identification in Home Environments using Facial and Appearance Features

Yue Sun
Supervisor: Prof. Majid Mirmehdi

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Bachelor of Science in the Faculty of Engineering

---

Monday 7^{th} May, 2018

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of Bachelor of Science in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Yue Sun, April 2018

Acknowledgements

I would like to thank Professor Majid Mirmehdi for his excellent support and help, and give me the right direction to guide me. I would like to also thank Dr Victor Ponce Lopez for his patience and excellent help.

Abstract

Re-identification is a vital task in video surveillance, and it is often viewed as an image retrieval task (i.e. given one image to search relevant images from all other images captured by different cameras.). The majority of the state-of-art methodologies focus on data which is obtained in an outdoor environment and are based on spatial and temporal features [1]. However, there is a need for methodologies that can handle data from the home environment due to the increasing demand for health monitoring. The LIMA dataset [2] , where the images are captured by a low-cost camera in the home environment, provided by the SPHERE project, includes realistic challenges. As an example, people in the scene may change clothes, and an unknown identity may appear in the scene. Since these are challenging situations, facial features can help to discriminate better amongst different identities that appear in the scene. Therefore, We have provided some re-identification results for LIMA dataset and methodologies focus on both facial and appearance features. Besides, we also label each image in the LIMA dataset to provide a LIMA face dataset (binary label, face or not containing face) to support the face detection performance measurement. Both traditional and modern approaches are employed to compare the performance in face detection and recognition. Furthermore, CNN with only appearance features is trained to see the drawbacks of the appearance features. Then, we propose a two-stream convolutional neural network to combine facial and appearance features together to predict the ID of the person, and it can be shown that the facial features indeed improve the performance through a simple experiment.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The UK and other countries are suffering long-term health conditions, health conditions generally require continuous monitoring for years. The continuous management for obesity, depression, diabetes, stroke and falls is usually required outside hospitals, and these should be monitored inside a home environment. Meanwhile, This project is related to the SPHERE (a Sensor Platform for HEalthcare in a Residential environment) work package 2 which incorporates action recognition and multi-user tracking system for the house environment to estimate the activity of users in their daily life. Presently, work package 2's vision team employs a tracking system without understanding which person is being tracked. In this sense, the re-identification by employing facial and appearance features can be used to get the ID of the participant.

1.2 Contributions

Presently, the majority of re-identifications methods use temporal and spatial features in some outdoor environments, these features may not be robust enough in home environments since the cloths, or other features can be changed more frequently. Therefore, some methodologies can handle data from the home environment is required due to the increasing demand for health monitoring in the home environment, and LIMA dataset can be an appropriate dataset for this purpose since the images are captured by a low-cost camera in the home environment and include realistic challenges. As an example, people in the scene may change clothes and an unknown identity may appear in the scene and more details of LIMA dataset will be introduced in section 3. Since these are challenging situations, facial features can help to discriminate better amongst different identities that appear in the scene. Thus, we provide some re-identification results on the LIMA dataset by using facial and appearance features.
Regarding facial features, a face is required to be detected from an image so that features can be extracted from the face and we label each image in the LIMA dataset to provide a LIMA face dataset to support face detection performance measurements. Concerning face detection, HoG-Based and CNN-Based detection approaches are both employed to compare the performance. In terms of face recognition, eigenface and CNN-Based approaches are both utilised to see the drawbacks of eigenface.
Regarding appearance features, a CNN-Based approach is used to extract appearance features and predict the identity of a person. Finally, we propose a methodology, namely 2 Stream Convolutional Neural Network to combine facial features and appearance features, to handle data well from the home environment. Also, an experiment is designed to see the effectiveness of facial features.

1.3 Structure of Thesis

This thesis contains seven parts:

Chapter 2 includes the investigation of some traditional and state-of-art approaches to do detection and recognition and discuss some previous works about the low-resolution recognition and re-identification.

Chapter 3 generally introduces the three subsets of LIMA dataset, and the advantages and disadvantages of each subset will be reviewed.

Chapter 4 discusses the procedure of how to label images to give a binary label (either face or non-face) and investigate existing software for annotation tool. Furthermore, a simple annotation tool is designed and implemented. The difficulties and performance of labelling face is discussed in this chapter as well.

Chapter 5 includes the results of face detection and recognition on LIMA datasets by using two detection (HOG with SVM and Faster-RCNN) and two recognition (eigenface and CNN) approaches. The drawback of eigenface feature and a comparison between learnt CNN features and eigenface are discussed.

Chapter 6 includes the performance of CNN with appearance features on LIMA dataset and a discussion of the reliability of appearance features.

Chapter 7 proposes a 2 stream convolutional neural network by utilising the face and appearance features and discuss the strategies of training this 2 stream CNN. Furthermore, a simple experiment is done to show the influence of the facial features. Moreover, the limitation of this network is investigated.

Chapter 9 discuss the future works and provides some potential ways to improve the 2 stream network and some possible training strategies are proposed.

2 Background

The potential approaches and previous work for re-identifications will be discussed in this section.

2.1 Possible Approaches

Some traditional and modern approaches for doing re-identifications will be reviewed. Since we focus on face and appearance features only, possible methods will be focused on the face detection and feature extractions. In terms of detection, Viola-Jones, HoG-Based and CNN-Based approaches will be reviewed. Regarding feature extractions, face and appearance features can be extracted in the same way, so eigenface, CNN-Based methods will be discussed in this section.

2.1.1 Detection

Viola–Jones Approach [3] was first introduced by Paul Viola and Michael Jones, which mainly provided a framework to solve the object detection but focusing on face detection task. Viola-Jones framework consists several stages, namely sliding window, extracting haar-like features, training a classifier via Adaboost [4] and cascading classifiers.

1. Sliding Window:
   The sliding window approach is a brute force method to move the window over the whole image with a range of window size. For each window, it is fed into the next stage and determines whether this window is containing an object or not. The method certainly has some limitations and advantages. One main advantage of this is that the sliding window is very simple to implement. However, it is a challenge to define the window size for different object detection tasks and it is computationally infeasible. Generally, the searching space remains huge. Meanwhile, if the ratio of positive examples and negative examples in a image is low, it actually waste a lot of computation resources to compute negative examples.

2. Haar-like Features:
   Haar-like Features are contrast-encoding box features defined by sums or differences over the average pixel value of a set of adjacent rectangular image regions. They encode fundamental contrast alignments such as edges, lines and point features. The Integral image is defined as the double integral of the 2D input image, it can be calculated by integration along each spatial dimensions.

   $$Integral(x, y) = \sum_0^y \sum_0^x I(x, y)$$

   and it can be computed in the linear time by using the formula:

   $$\text{II}(-1, y) = 0;\ \text{II}(x, y) = \text{II}(x - 1, y) + A(x, y);$$
   $$A(x, -1) = 0;\ A(x, y) = A(x, y - 1) + I(x, y);$$

After computing the integral image, Haar-like features can be computed by looking at four entries, we can get the sum of a particular region. There are some advantages and drawbacks. Haar-like features are fast to compute, but Haar-Like features can only represent simple shapes, and it is not rotation, symmetry invariant and is sensitive to illumination variations.

3. Adaboost Learning
   The AdaBoost approach is a popular boosting technique which selects a set of different weak classifiers and combines them into a single strong classifier. A collection of positive and negative examples are provided as the input and assign equal weight to each example initially. The weight represents the difficulty of a sample. It tests weak classifiers and sees how well it is doing based on each single haar-like features. Furthermore, it selects the best classifiers with lowest error rate. Then, it updates weights so that it gives the false classified example more weights. Large weights mean more difficult, so it will force to focus on difficult examples. After training, we can get a list of weak classifiers and corresponding weights.

4. Cascading classifiers
   It orders classifiers regarding its weights (complexity), the first level uses only two classifiers, if the window does not contain a face determined by these two strong classifiers, then, the window is discarded and will not feed into next level. By using the cascading technique, it reduces a lot of computation and still provides nearly 0 false negative rate.

HOG-based (Histogram of oriented gradients) [5] detection uses the similar idea of the Viola-Jones, it first uses sliding windows to generate a set of small regions and extracts HOG features and feeds HOG features into SVM and SVM will decide whether it contains a face or not.

1. It computes the gradients of an image.

2. It divides an image into smaller cells and creates histograms of the gradients with 9 bins for each cell based on the magnitude of the gradient and the direction of the gradient.

3. Group the adjacent cells into a block.

4. To account for the change in illuminations and contrast, it normalises the gradient within a block. After normalising the gradient within a block, it can form a feature vector by combining all normalised histogram of oriented gradients for a particular region. Then, the feature vector is fed as input to SVM. By providing positive and negative examples for SVM, we can determine whether the region containing a face or not via SVM.

CNN-based detection becomes increasingly popular since the last few years with the help of a large number of available computation resources

and a significant amount of labelled data. Convolutional Neural Network has shown its powerful capacity in the image classification. In 2012, Krizhevsky et al [6] shows the significant higher image classification results on the ImageNet large Scale Visual Recognition (ILSVRC) [7] by using CNN which is trained on 1.2 million labelled images.

1. R-CNN.
   Motivated by Krizhevsky et al's result, Ross Girshick first proposed a method (i.e. R-CNN)[8] to bring CNN to object detection task and shows a very high performance compared with HOG-features in 2014. The approach he proposed improves mean average precision (mAP) by approximately 30% by comparing with the best result on VOC 2012. Generally, R-CNN mainly contains two stages, namely region proposal and CNN feature extraction.

   (a) Region Proposal.
   In the region proposal stage, a list of potential regions is required to be generated and feed them into a CNN and let CNN decide whether a small region contains a desirable object or not. The region is required to have certain property (e.g. category-independent) in order to adapt various detection tasks. In the RCNN paper, the algorithm used to generate a set of regions which have a high probability to contain a class-independent object is selective search [9].

      i. Selective Search
      Selective search is a technique which can be used to generate a list of potential object locations. The exhaustive search (e.g. sliding window) is a reasonable way to consider all locations and all scales in an image. However, the search space is usually considerable large and the computation is usually expensive. Therefore, the selective search makes some assumptions about the objects: the nature of images and the different uses of an object category are hierarchical and there may not exist a universal approach to solve this problem by using a single strategy. Thus, it first uses the algorithm called Efficient Graph-Based Image Segmentation[10] to generate a set of possible regions, then, it makes use of hierarchical grouping and a lot of different strategies by using multiple different colour spaces and similarity measures. Then, the number of possible object locations would be much less than the exhaustive search while the quality of object locations remains high.

   (b) Convolutional Neural Network Feature Extraction.
   The Convolutional Neural Network commonly consists of three different layers, namely convolutional layer, pooling layer and fully connected layer.

      - Convolutional layer:
      In computer vision, convolution operations were generally

utilised for edge detection, for example, sobel operator, prewitt operator and roberts cross operator. The parameters of convolution function used in sobel, prewitt and roberts are fixed. In a neural network, convolutional layer makes use of convolution operation but with learnable parameters. The parameters in the convolution function can be updated by using back-propagation, so convolution function can automatically find the better function to extract features. Moreover, the convolutional layer extracts low-level features (e.g. edges) in the early layers of the CNN and it combines low-level features into high-level features in later layers, which make use of the hierarchical information of an object.

- Pooling Layer:
  Pooling layer is a form of non-linear down-sampling function, it gradually reduces the input size to decline the number of parameters and computation but remains important features. Pooling layer has several variants, max pooling and average pooling.
- Fully Connected Layer
  A fully connected layer is the linear combinations of an input features from a last layer. Using an non-linear activation function, it can learn a non-linear function such that it maps the input to the output based on the loss function since the parameters will be updated through back-propagation to minimise the loss function.

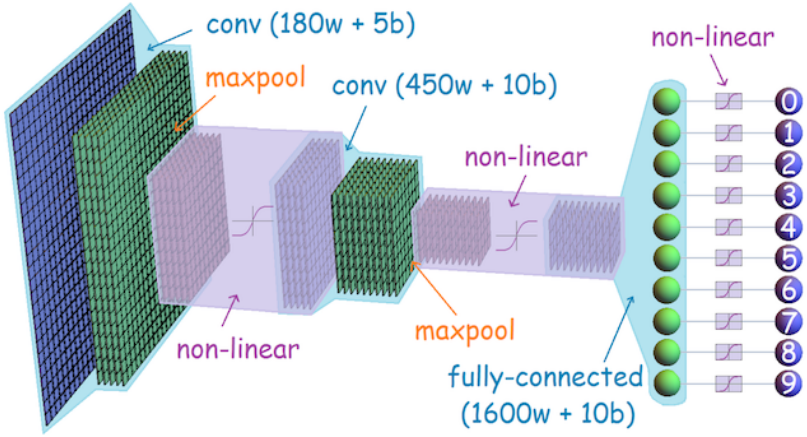


Figure 2.1: An example of CNN

By utilizing the selective search, the number of regions fed into CNN would be much less, and each region is fed to the CNN to classify whether this region containing a face or not.

2. Faster R-CNN
   Faster R-CNN[11] is an improved version of Fast R-CNN [12] and R-CNN [8], the limitation of the Fast R-CNN and R-CNN is the region

proposal part, which takes up the majority of the whole computation to generate a set of potential object locations. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun use a new network (Region proposal region (RPN)) to generate a set of potential object regions, and use bounding box regression to adjust proposed regions. Comparing with R-CNN, an image only needs to pass forward propagation once instead of feeding each potential region into CNN in R-CNN model as many potential region in the R-CNN are overlapped, hence, the majority of computation in R-CNN is duplicated and can be avoided. The results of Faster R-CNN showed that it achieves about 17 FPS detection rate on a GPU and achieves state-of-the-art object detection accuracy on PASCAL VOC [13] 2007, 2012, and MS COCO datasets[11].

2.1.2 Recognition

The **Eigenface** approach is introduced in 1987 by Sirovich and Kirby [14], and eigenface feature is a set of feature vector which can be computed through the eigenvectors of a covariance matrix. It can be thought as a dimension reduction method which reduces the high dimension data to low data dimension and it is a straightforward approach to archive the recognition task with high accuracy. It contains several stages:

(i) Reshape the image to column or row vector and subtract the mean.

(ii) Compute the eigenvalues and eigenvectors of the covariance matrix.

(iii) Chooses the **k** components whose total variance is greater than a threshold (which is usually set to be 95%)

(iv) Project the face image to PCA basis represented by k principal components, the feature vector is the factors of linear combination of k principal components.

A support vector machine (SVM) [15] can be trained by eigenface feature vectors for the positive and negative examples to find the best hyper-planes separating for each class, then, SVM now can predict the label given an new image.

The **CNN** can be either used as a feature extractor to extract features and train a support vector machine to classify images or to use softmax layer to directly output the probability of a class. Regarding face recognition task, it often meets the one-shot learning problem. One-shot learning problem is that it allows the computer to recognise the new person by providing only one or a few examples of that person. Thus, a simple solution to solve it is that it trains a network as a feature extractor to find a similarity function such that the images representing the same person will output low value and the images representing different people will output very high value. There are many objective functions could achieve it, as an example, triplet loss function[16] is defined as follows:
Given 3 input images:

$$\mathcal{L}(A, P, N) = max(||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + \alpha, 0)$$

where A, P, N and α represents anchor, positive, negative examples, margin α is enforced between positive and negatives example to prevent both $||f(A) - f(P)||^2$ and $||f(A) - f(N)||^2$ achieve 0.
The triplet loss can learn a similarity function such that it pushes data in the same class closer and pushes data in different classes further. In the end, the network can now output a feature vector encoding the face, a simple SVM can be trained to take the feature vector as input and output the class label.

2.2 Previous Work

In this section, mainly two type of previous works will be reviewed, namely person re-identification, low-resolution face recognition problem.

2.2.1 Person Re-identification

This project can be more related to re-identification task, methodologies for re-identification task has been developed for several years starting from 1997, a Bayesian method [17] is proposed to predict the appearance of an object in one camera based on other cameras. After that, re-identification focus on image-based and video-based approaches. Regarding image-based approaches, it can be further divided into hand-designed system and deep learning based system. The hand-designed system mainly uses some colour features, e.g. using 8 color channels and 21 texture filters [18] and [19] [20] uses similar ideas to [18], or using RGB, YUV and HSV colour spaces and LBP texture histograms [21]. Since the mentioned methodologies for image-based re-ID all use some low-level features, some middle-level features can be employed to perform re-ID task as well. As an example, SDALF [22] combines some low-level features together to middle-level features by some weighting functions. Concerning deep learning based approaches, it mainly focuses on CNN-based classification [6] [23] and CNN-based detection [11] [8].

In terms of video-based re-ID, it can be divided into two parts as well, namely hand designed and deep learning based system. In terms of hand designed features, [24] uses colour based features and a segmentation is utilised to detect a person in the scene. Furthermore, [25] trains a boosting model based on some covariance features to perform re-ID. Regarding deep learning based systems, [1] uses both temporal and spatial features to predict the identity of a person.
This reviewed history of re-identifications are based on the paper [26], and two approaches among mentioned methods will be discussed in details.

In **A Two Stream Siamese Convolutional Neural Network For Person Re-Identification** paper [1], they firstly make an assumption about the re-identification task: The spatial and temporal features' representation does not need to be the same and both spatial and temporal features

can be useful for processing re-identification task. The paper proposes a two stream Siamese convolutional neural network to capture temporal and spatial features separately. Meanwhile, a weighted CNN objective function is designed to combine the Siamese cost of the spatial features and temporal features with the objective function of predicting a person's label together to train the whole network at once.
The proposed network is shown in Figure 2.2, each CNN in either spatial network or in temporal network shares weights. However, CNNs in different networks does not share weights (i.e. the weights of CNN in spatial net is different than weights in temporal net), this could allow the network learn different parameters to capture different features. The input of spatial network is sequences of RGB images, and the input of the temporal network is sequences of optical flow images. Publicly PRID2011 [27] and iLIDS-VID [28] datasets are evaluated by the proposed approach, they show that the matching accuracy is approximately 4% higher than the accuracy achieved by the cross-view quadratic discriminant analysis used in combination with the hierarchical Gaussian descriptor [29] (GOG+XQDA), and about 5% higher than a method based on RNN [30].

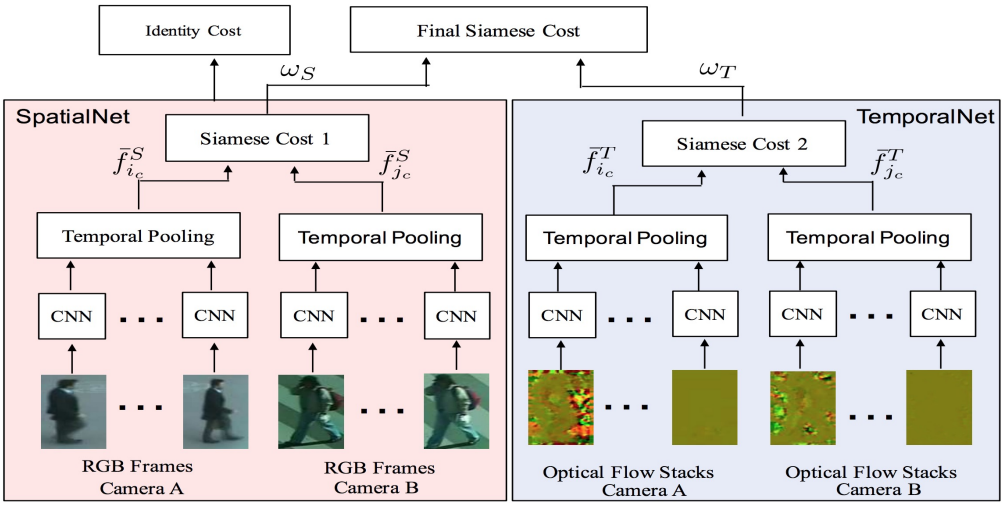


Figure 2.2: The proposed two stream network in paper [1].

The **Improving Person Re-identification by Attribute and Identity Learning** paper proposed an idea that it combines attribute recognition and person identification to improve the re-identification performance since the re-identification and attribute recognition share a common target [23]. A simple convolutional neural network is designed to learn a re-ID feature vector and predict the person's attribute simultaneously. The proposed network is demonstrated in Figure 2.3. During training, it both predicts person's ID and its attribute, and the total loss is the sum of weighed each loss. The performance shows that this network significantly improves the re-ID baseline on two datasets, namely Market1501 dataset [31], and DukeMTMC-reID [32] dataset.

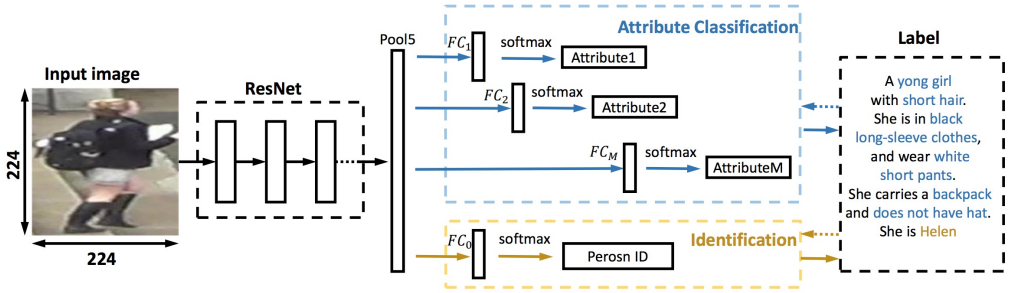


Figure 2.3: The proposed network in the paper [23].

2.2.2 Low Resolution Face Recognition

This project can relate to the very low resolution face recognition problem as well because of low-valued camera and images obtained by these cameras in the dataset are mainly suffering the low-resolution issue. Since we focus on the re-identification problem, the low resolution face recognition will only be reviewed very briefly.

Very Low Resolution Face Recognition Problem paper [33] describes a method to handle the very low resolution (VLR) recognition problem which is commonly seen in many surveillance applications. It first determines the relationship between the high resolution(HR) and the low resolution(LR) space by providing the HR and LR pairs in the training set, and propose a new super resolution (SR) algorithm to tune the mapping **R** between HR and LR space by minimising the error function between the ground truth and the reconstruction through the mapping **R**. Furthermore, several experiments have done to test the performance of their algorithm, they have shown that it has a significant improvement on the real surveillance video for recognition task.

Studying Very Low Resolution Recognition Using Deep Networks paper[34] describes a deep learning based method to incorporate super resolution, domain adaptation and robust regression to handle very low resolution recognition (VLRR) problem. Firstly, HR and corresponding LR image are grouped as a pair in the training set. Meanwhile, an assumption is made that LR and HR image features do not fully overlap even after complicated transforms. Thus, a model is formulated which called partially coupled SR networks (PCSRN), shown in Figure 2.4. Some shared and unshared filters are introduced so that unshared filters can learn some domain-specific features to support shared filters to handle mismatches. The experiments on face identification, digit recognition and font recognition show that it has a significant improvement.

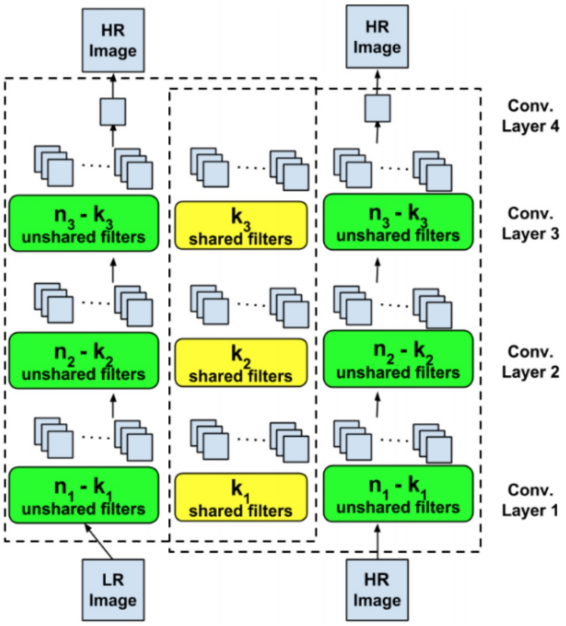


Figure 2.4: A model is described in the paper[34].

3 Introduction to LIMA Dataset

The dataset is used for this project is the LIMA dataset which is provided by the SPHERE project. The images are captured by using low-valued camera (for example, Microsoft Kinect, Asus Xtion, Prime Sense and Genius WideCam 1050 [35]) in home environments and is pre-processed by using a tracking system so that each image in the dataset is cropped by the bounding-box provided by the tracking system. Therefore, each image will usually contain one person only. Three subsets of LIMA dataset are provided and the features of each subset will be discussed in the next few subsections.

3.1 Subset 1

The subset 1 contains 106,770 JPG files in total from the three camera views, each image is formate as "timestamp.jpg" and the subset 1 includes 3 ground truth label files and 1 dataset description file. Further, the ground truth label files are formatted as "labels_view{1-3}.txt". In the ground truth file, it has two columns which are in the form of timestamp and participant ID separated by comma. Meanwhile, the ground truth file only contains the person ID and does not contain whether an image contains face or not. In this subset, it has 8 classes, namely -1, 0, 1, 2, 3, 5, 6, 8. The label -1 means there is an unknown person in the scene, and label 0 represents there is no person in the scene. Furthermore, each remaining class represents different people in the scene. The structure of files:

56693851,0
56693854,0
56693856,0
56693859,0
56693861,0
56693876,0

The total number of rows in the ground truth files is more than the actual images. For example, "56763132.jpg" in the ground truth files has two labels shown in the figure 3.1, namely -1 and 3. Based on the description file, there are 2 main reasons causing this issue:

- Some images could not be retrieved from the database.
- The files are created for all participants ID (i.e. including unknown ID).

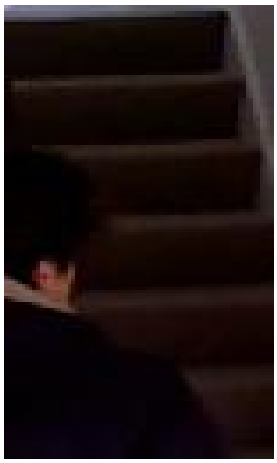


Figure 3.1: The multiple labels (-1, 3) for one image.

3.2 Subset 2

The subset 2 contains 188,427 JPG files in total from the three camera views, each image is named as "<timestamp>_<class-id>.jpg". There are 7 classes in this subset, the classes -1 and 0 have been grouped into one single class 0 for simplicity. The structure of files is shown below:

56693851_0.jpg
56693854_0.jpg
56693856_0.jpg
56693859_0.jpg
56693861_0.jpg
56693876_0.jpg

Meanwhile, this subset fixed the issue of the subset 1, each image now has only one label, for instance, The timestamps 56763132 now has two images, namely 0 and 3. It is illustrated in the Figure 3.2.

(a) 56763132_0.jpg (b) 56763132_3.jpg

Figure 3.2: The subset 3 fixed the issue of multiple labels on one image .

3.3 Subset 3

The subset 3 contains 188,424 JPG files in total from the three camera views as well. However, the images are categorized into sessions, each image in a particular session will be very similar since the images come from only one or more video files so the majority of images are continuous frames, and the person's movement is very small between each frame. Nevertheless, the images will be much different when images are in different sessions because the images come from different sessions are usually captured in the different locations or different time. This subset will allow us to perform the leave-one-out validation easier by keeping one session out each time we run our learning method. However, not all session contains all labels, e.g. session 9 only has 0, 1, 4, 5, 6 and does not have 3. The structure of this subset is shown in Figure 3.3.

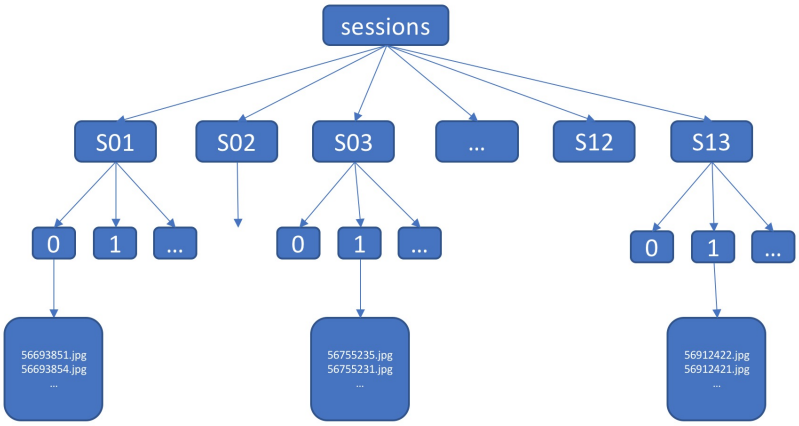


Figure 3.3: The structure of subset 3.

3.4 Disadvantages of the LIMA Dataset

There are some drawbacks of this datasets. Firstly, the label 0 contains a larger number of very similar images. This may not generalize the model well when training a model with this dataset, as an example, two images frequently appear in session 8 with label 0 shown in Figure 3.4.

Figure 3.4: The dataset contains a lot of similar images.

Furthermore, the temporal information is broken as even in one session the images may come from different video files, as illustrated in the Figure 3.5.

It is shown that the middle image suddenly appears in continuous frames. This may forbid us to utilizes the temporal features while training model.

(a) 56726602.jpg (b) 56726603.jpg (c) 56726604.jpg

Figure 3.5: Three images with continuous filenames.

The Optical flow could be a good way to calculate the temporal features for videos frames. Nevertheless, each image may have different sizes in the LIMA dataset, this results in that the temporal features may not be computed through the optical flow approach since optical flow require two images to be the same size. Meanwhile, this cannot be fixed by resizing as the image ratio will be destroyed when resizing the image to a fixed size.

4 Ground Truth for Face

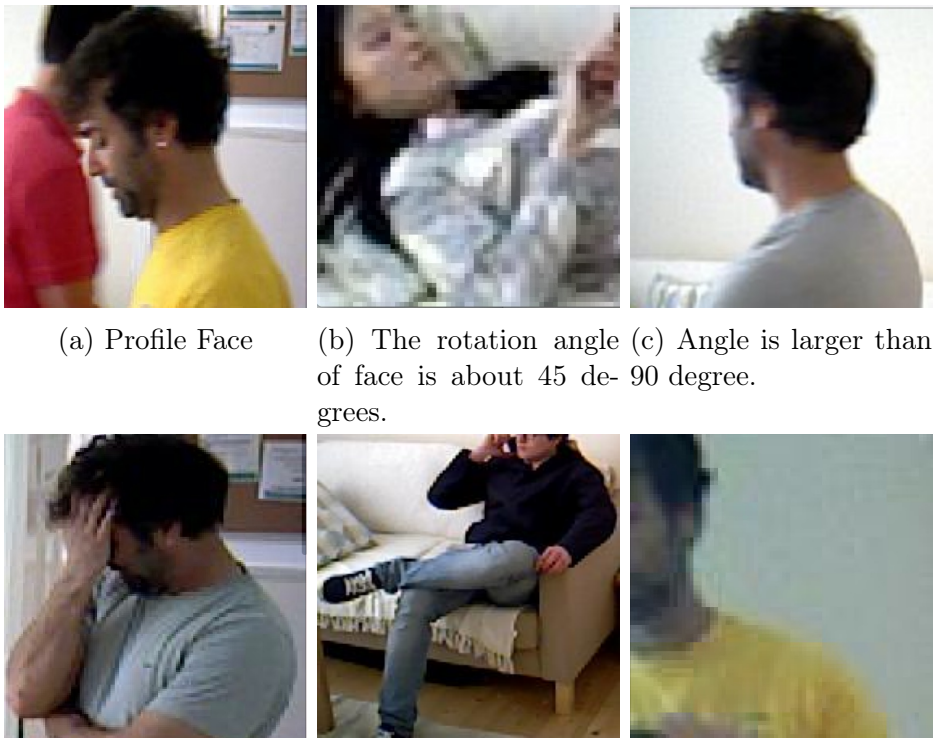
We only have person's identity for each image in three subsets, but it only can be useful when we perform the recognition task and hence it is reasonable to provide a binary label to represent whether an image contains face or not in order to measure the performance of the face detection. Thus, We provide a LIMA face dataset to support the face detection performance measurements by using subset 3.

4.1 Label Criteria

The criteria will be kept as simple as possible since complex criteria can be challenging to apply.

- Firstly, it needs to define some angles, i.e. the angle of rotation of a face, we define the frontal face has 0 degrees and the face can freely rotate around X, Y, Z axis, and the profile face will have 90 degrees in Y-axis. Frontal face, profile face, different poses will be labelled as face. As an instance, the Figure 4.1a shows that a profile face will be labelled as face and 4.1b illustrates that a face with approximately 45 degrees of rotation angle around Z axis will be labelled as face as well. Furthermore, every face whose angle is greater than the 90 degrees will not be labelled as face. As an instance, the Figure 4.1c shows that the angle of rotation of the face is larger than 90 degrees and the majority of his face is invisible.

- If the facial region is not obvious or too small, or the majority of the facial region is blocked, this kind of images will not be labelled as face. e.g. the Figure 4.1d shows an example that the angle of the rotation is around 90 degree, but the most of facial region is not visible, and thus it will not be labelled as face. Meanwhile, it will depend on the face area inside the image if the face is partially outside the image. The Figure 4.1e shows that the image is labelled as face since the image contains enough face region and the nose, mouth of the person provide sufficient information to give the identity. The Figure 4.1f shows that the majority of a facial region is outside an image and the remaining information cannot give discriminative features to predict the identity of the person. Hence, it will not be labelled as face.



(a) Profile Face

(b) The rotation angle of face is about 45 degrees.

(c) Angle is larger than 90 degree.

(d) Profile face is blocked by his hand.

(e) The face is partially outside the image.

(f) The face is partially outside the image.

Figure 4.1: Different situations.

4.2 Software for Ground Truth Labelling

Considering this amount of images, we only have a tiny amount of time to label data. For this reason, it requires a tool to fulfil a few metrics:

- The tool has to be sufficiently convenient and straightforward.
- The tool has to provide efficient ways to label data.
- It does not need sophisticated functionalities, for example, the cloud-based platform is not helpful when dealing with very large dataset.

4.2.1 Existing Software

There are some existing annotation tools for ground truth labelling, some tools will be reviewed and discuss its benefits and drawbacks in this section.

- Labelbox [36]: It is a data labelling platform and provides complex annotation tools (e.g. draw bounding box for image classification, draw shapes for image segmentation, text segmentation). It could output some useful statistic, as an example, it is able to show the average of time for labelling each image and the overall time for all images and it illustrates the daily progress by drawing graph. Furthermore, it supports multiple output formats (e.g. JSON, and CSV). Nonetheless, one of main drawbacks is that the tool requires the data to be uploaded to their cloud and there are approximate

188,424 images in the LIMA dataset, it would require enormous time to transfer data. In fact, the Labelbox website gets stuck when all images are selected to be uploaded.

- Images Annotation Programme [37]: It is a full web application, it does not require the data to be uploaded to their cloud, and it provide bounding box annotation tools. However, One of main drawbacks is that it does not support any other annotation methods (e.g. segmentation, binary label (without bounding box)). Furthermore, it only supports xml output format.

- Dataturks [38]: It a web-based annotation application containing all general annotation tools (e.g. bounding box, image classification, image segmentation, text classification and so forth). It supports collaborative environment and can invite people to label via email. However, it requires data to be uploaded and the total max files size is 100 MB for free plans and it demands mouse to click buttons.

4.2.2 Personal Designed Tool

All above present annotation tools offer comparable and beneficial functionalities. Nonetheless, all of those software are struggling some unique troubles. In order to suit to our propose, a simple ground truth label interface has been designed and implemented, but more details about designed and package choices and implementation details can be found in Appendix A.1.

4.3 Difficulty Analysis

Throughout the labelling process, some images are generally tricky to decide whether an image contains a face or not.

- Light Conditions: some images are suffering terrible light conditions as shown in Figure 4.2a, it is far even tough for a human to decide whether this kind of images contains a face or not, especially when a human has recognized quite a few images and suffering from fatigue.

- It is tough to decide when the angle of rotation is around the 90 degrees and the angle is measured by sight and can be very inaccurate. For example, the angle of the image is slightly larger than 90 degrees shown in Figure 5.1b, it could both say that it has a face or it does not have a face.

- Low Resolution: Some images have relatively low resolution shown in Figure 5.1c, although, the angle of rotation is not larger than 90 degrees, the resolution is too low to decide whether it contains a face or not.

(a) Bad light condition. (b) Around 90 degree of rotation angle (c) Low resolution.

Figure 4.2: Different tough cases.

4.4 Labelling Speed and Performance

The subset 3 dataset still contains some unavoidable issues that make huge influences on the speed of labelling. As we discussed the drawbacks of this LIMA dataset, the images may come from different videos files even in one session but with continuous file names. Spatial-temporal information is really useful for a human to do identification for video files. However, we have to slow down as some images suddenly appear in continuous frames shown in Figure 3.5. In fact, this situation occurs quite frequent among all sessions, especially it occurs much more frequent in session 10 with label 1, and there is no simple solution to solve it. Thus, the speed of labelling is approximate 100 images per minute, and it takes about roughly 31 hours to complete.

4.5 The Result of Face Ground Truth

	face	non-face	total
number of images	60939	127485	188424
%	0.32	0.68	

Table 4.1: The face ground truth.

5 Re-Identification via Face Features

5.1 Face Detection

As the input image is already a cropped image and usually contains one person only. Therefore, a simple assumption can be made for face detection, namely we only detect one face per image and select the most confident one or the largest one if multiple faces are detected for simplicity. Nevertheless, the drawback of selecting the most confident one is that a frontal face may generally get higher probability than a profile face, and it is not fair for a person who stands far from a camera, resulting in a smaller bounding box. In this session, we are mainly going to use one traditional method and one deep learning based method to perform face detection, namely HOG with SVM and Faster RCNN approaches.

5.1.1 HOG Based Face Detection

The framework of the HOG traditional method is generally described in the background section. The pre-trained HOG model from dlib is used to perform face detection, the dlib is a c++ library containing machine learning and deep learning algorithms to solve the real problems, but it contains python API that can be used to work with dlib tools and python. According to the dlib official website, the HOG pre-trained model is trained by approximate 3000 images from labelled faces in the wild dataset and the total training time takes only about 3 minutes.
The labelled face in the wild (LFW) [39] dataset is a collection of images collected from the web and there are approximately more than 13,000 images in the dataset. However, the majority of the face is frontal face and the majority of classes contain only one, or two images and about 1680 of the people have two or more distinct images. It can be seen in Figure 5.1 that the image has very high resolution and is very different than our LIMA dataset.

(a) Abdel Nasser Assidi (b) Alejandro Atchugarry (c) Alison Krauss 0001

Figure 5.1: The example of LFW dataset.

As the training set for the HOG pre-trained model is the LFW dataset, the pre-trained model is, in fact, a frontal face detector. Nonetheless, our dataset comes from video files and the majority of faces are not be frontal

face. In this case, the HOG pre-trained model should not perform well in our LIMA dataset and this issue is reflected in the error rate in Table 5.1. It is evident that the 0.08 recall shows HOG method can only detect a small amount of face among all faces, which means only approximately 0.08 faces are frontal faces. The precision 0.99 shows that the majority of detected faces are actually faces. The below in Figure 5.2 shows some examples of mis-classified images, the electric induction hob with three circles indeed looks like a face.

(a) Electric Induction Hob

(b) Clothes.

Figure 5.2: The mis-classified images by HOG with SVM.

labels	precision	recall	f1-score	support
non-face	0.69	1.00	0.82	127485
face	0.99	0.08	0.14	60939
avg/total	0.79	0.70	0.60	188424

Table 5.1: HOG with up-sampling rate 1

The up-sampling rate is a parameter to control how many times an image is rescaled, (i.e. 2 will rescale the image twice.) The pre-trained face detector looks for faces which are about 80 by 80 pixels or larger. However, the face is approximately 40 by 40 pixels in our dataset, so the up-sampling the image can allow it to detect smaller faces. The dlib package uses laplacian pyramid to up-sample the image, this will make the image blurred and bigger. In Table 5.2 and 5.3, it is apparent to see that the number of detected face increases along with up-sampling rate from 1 to 3. Nevertheless, it takes longer time to process the whole dataset, roughly 1 hour, 4 hours and 14 hours for HOG with up-sampling rate 1, 2 and 3 respectively on Intel (R) Core (TM) i7-8700 CPU @ 3.20GHz and GeForce GTX 1080 TI with 11264 MB memory. An even larger up-sampling rate is not recommended as the face region may be too blurred to detect it, and it does not improve significantly from rate 2 to 3 and the time taken to process entire dataset can be considerably longer (roughly 42 hours for rate 4).

labels	precision	recall	f1-score	support
non-face	0.71	1.00	0.83	127485
face	0.99	0.13	0.23	60939
avg/total	0.80	0.72	0.64	188424

Table 5.2: HOG with up-sampling rate 2

labels	precision	recall	f1-score	support
non-face	0.71	1.00	0.83	127485
face	0.98	0.13	0.23	60939
avg/total	0.79	0.72	0.64	188424

Table 5.3: HOG with up-sampling rate 3

5.1.2 Faster RCNN Detection

We stick with dlib to do face detection as dlib package supports Faster RCNN pre-trained model as well. The pre-trained model is trained by approximate 6975 faces dataset which is a collection of face images selected from many publicly available datasets (excluding the FDDB [40] dataset), there images come from ImageNet [7], AFLW [41], Pascal VOC [13], VGG dataset [42], WIDER [43], and face scrub [44]. The advantage of this is that the face has a wide range of poses rather than containing of frontal faces. This dataset is created by dlib and the whole dataset is shown in Figure 5.3 which contains all faces in the dataset.

Figure 5.3: All images from the dataset created by dlib.

The performance of Faster RCNN on LIMA dataset is shown in Figure 5.4, the number of detected face is increased significantly from up-sampling rate 1 to rate 2 and 3, recall represents how many faces are correctly

detected over all faces, the recall increments dramatically from 0.13 for HOG approach to 0.62 for Faster RCNN. However, the precision slightly drops since some profile or different poses faces are difficult to determine whether it contains a face or not in ground truth as discussed in section 4 difficulty analysis.

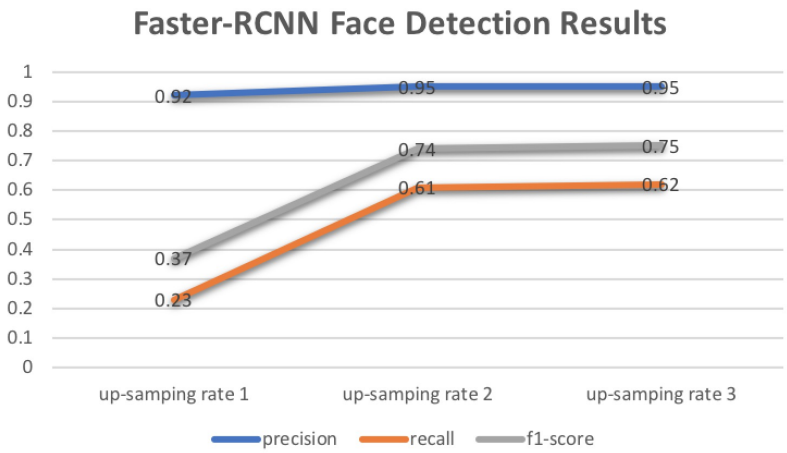


Figure 5.4: The Faster-RCNN results on LIMA dataset.

5.2 Face Recognition

In this section, the face dataset is provided by the previous section, both HOG with up-sampling rate 1 and Faster RCNN with up-sampling rate 2 results are selected to support face recognition and face has the same label as the image. However, one of main drawbacks of directly using image label is that the labels of some detected faces should not correspond with the image's label. As an example, the below image in Figure 5.5 has label 5 and it is the label of the woman wearing with white cloth but the detected face should not be the same with the image's label. Furthermore, faces in the label 0 are generally mis-labelled, the majority of detected faces are known people. However, these face images are still kept in the face dataset for simplicity.

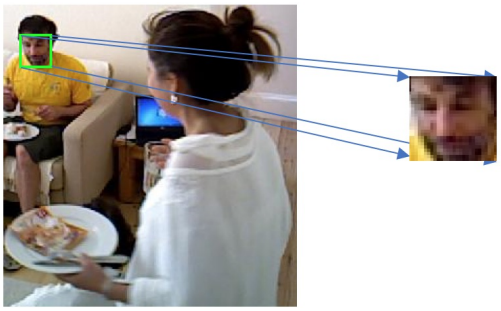


Figure 5.5: The label of image is 5 (it is tracking the white cloth woman), but it detects the face whose label should be 4.

5.2.1 Eigenface

The procedure of computing eigenface features require a fixed size over all images, 32x32 size is chosen to perform the computation by resizing all images to this size since a face region is generally a square. The dataset (total number of images: 4972) provided by HOG method with up-sampling rate 1 is randomly split into 75% training set and 25% test set, and support vector machine is trained with grid search and radial basis function kernel to find the best parameters. Using kernel method in SVM allows the data mapped to high dimension space and allow SVM to find a linear hyperplane in that high dimension space. The formula of radial basis function:

$$K(x, x') = exp(-\frac{||x - x'||^2}{2\sigma^2}) \tag{1}$$

where $\gamma = -\frac{1}{2\sigma^2}$, and C is the penalty term in support vector machine. The γ is to used to control similarity between two points, if the γ is very low, which means it has very high variance, two data points can be considered to be similar even they are far from each other. However, if γ is very high, the data point has to be closer enough to be considered as similar points. The potential parameters for radial basis function is shown in Table 5.4:

parameters						
C	1e3	5e3	1e4	5e4	1e5	
γ	0.0001	0.0005	0.001	0.005	0.01	0.1

Table 5.4: Potential parameters for SVM with kernel RBF

Nonetheless, tuning parameters is time-consuming by grid search, it takes about 4 minutes, 6-7 hours to train SVM for HOG's and Faster-RCNN's results respectively. The results of face recognition by eigenface on LIMA

labels	precision	recall	f1-score	num of images
0	0.47	0.31	0.37	52
1	0.92	0.96	0.94	275
2	0.95	0.94	0.95	336
3	1	0.86	0.92	35
4	0.88	0.94	0.91	192
5	0.95	0.97	0.96	0.96
6	0.98	0.92	0.95	53
avg/total	0.92	0.92	0.92	1243

Table 5.5: EigenFace results on HOG with rate 1.

test set is shown in Table 5.5 and some example eigenfaces are shown in Figure 5.6. The performance on label 0 only has 0.47 precision and 0.31 recall and 0.37 f1-score, it is expected as the number of face in the training set is generally fewer than others, and some faces in the label 0 are mislabelled as mentioned in section 5.2.

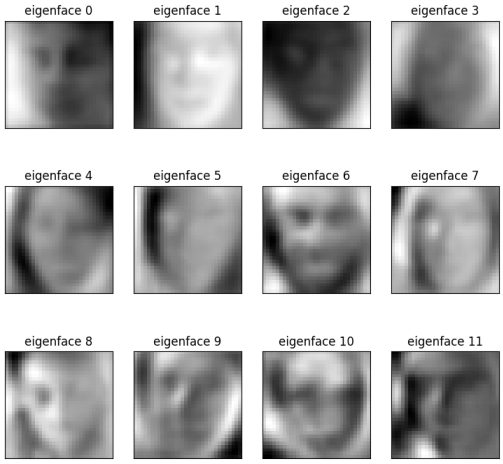


Figure 5.6: The example of eigenface in LIMA dataset.

By using the face training set provided by Faster RCNN with up-sampling rate 2, it is evident to see that the performance is, in fact, increased slightly in Table 5.6. Although the eigenface feature is not robust, the support vector machine still can be trained to a good classifier since we have provided enormous training data (approximately 29130 faces).

labels	precision	recall	f1-score	num of images
0	0.81	0.76	0.78	715
1	0.92	0.96	0.94	2780
2	0.97	0.97	0.97	1564
3	0.86	0.87	0.87	449
4	0.97	0.96	0.96	1999
5	0.94	0.90	0.92	993
6	0.97	0.97	0.97	1210
avg/total	0.94	0.94	0.94	9710

Table 5.6: EigenFace Results on LIMA test sets (random selected).

Nevertheless, the eigenface feature is sensitive to many factors and is not robust enough to recognize the face with different poses and different illumination. To explain the reason why the eigenface is not robust, we can first recap how eigenface works. Consider the below 2-dimension dataset in Figure 5.7, it could say that the green point X represents a profile face of a particular person, and others are frontal faces of that person as the data of profile face and frontal face can be very different. In order to compute the eigenface feature, we first need to extract the mean, and compute the eigenvectors and eigenvalues, the two red arrows represent two eigenvectors (i.e. two principal component axis) of the frontal face, then the data of frontal face can be represented as the linear combination of the components.

$$Y = aV_0 + bV_1 + ... + nV_n \tag{2}$$

where Y is a data point, [a ... n] is factors (a feature vector), [V_0 ... V_n] is eigenvectors. When the profile face data is projected into PCA basis, the projected components correspond to factors [a ... n] in a linear combination of a profile face can be very different than the projected components of a frontal face. Therefore, we can get very different feature vectors even for a single person. Furthermore, the above training set and test set are randomly selected from the whole dataset and a lot of faces in the dataset are very comparable, hence, the distribution of the training set and the test set can be very similar. The support vector machine is still able to find a good hyper-plane to put the face with different poses of a single person into a class and separate that class with other classes since we have enough data, so the performance increment slightly.

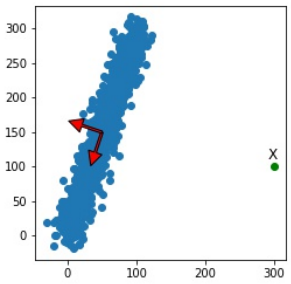


Figure 5.7: A simple dataset used to illustrate the drawback of eigenface feature.

In order to test the performance of eigenface when the distribution of training and test set is divergent, session S09 is selected to be a test set, and remaining sessions are used as a training set. The session S09 (it only contains images for label 0, 1, 4, 5, 6) is a challenging session as people lie on the sofa in many frames or it has many images containing face with large rotations as shown in Figure 5.8. The following table 5.7 shows that the performance of eigenface with support vector machine decreased dramatically, the precision, recall and F1-score is dropped from 0.94, 0.94 and 0.94 to 0.82, 0.52 and 0.55 respectively as the eigenface with SVM does not generalize well in this case.

(a) Lie on the sofa and has large rotation.

(b) Profile face and has large rotation.

Figure 5.8: Tough cases in session 9.

labels	precision	recall	f1-score	num of images
0	0.00	0.00	0.00	14
1	0.50	0.80	0.61	608
2	0.00	0.00	0.00	0
3	0.00	0.00	0.00	0
4	0.97	0.79	0.87	1529
5	0.43	0.87	0.58	338
6	0.88	0.17	0.29	2042
avg/total	0.82	0.52	0.52	4531

Table 5.7: EigenFace Results on LIMA session S09.

5.2.2 CNN Based Recognition

A slightly modified version of the LeNet-5 [45] architecture of CNN is used to classify images, the original LeNet-5 architecture is invented in the 1990 year which is one of the very first convolutional neural networks. The architecture of LeNet-5 is shown below in Figure 5.10.

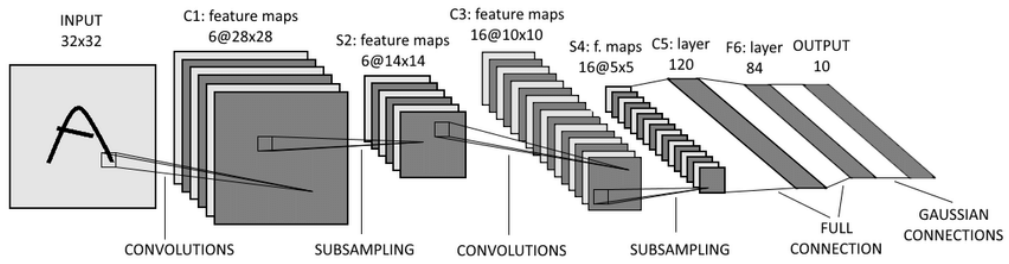


Figure 5.9: The architecture of LeNet-5.

The LeNet-5 utilized a simple idea, it uses convolution with learnable parameters to extract spatial features, and a pooling layer is introduced after a convolution layer to sub-sample the image in order to reduce the number of parameters, non-linearity activation function (tanh, Relu) is used to better approximate non-linear function and finally a fully connected network (a traditional multi-layer perception) with softmax activation function is introduced to classify images. We slightly modified the LeNet-5 architecture, the modified LeNet-5 is shown in Figure 5.10, L2 regularize and drop-out layer is added to reduce the overfitting and batch normalization layer [46] is introduced to improve the training speed and add a slight regularization effect.

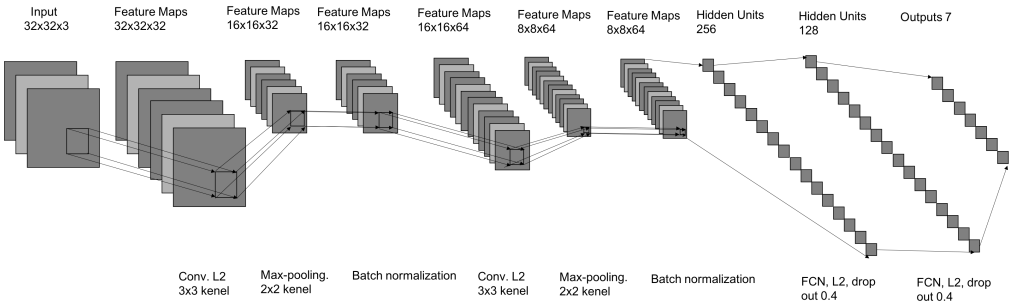
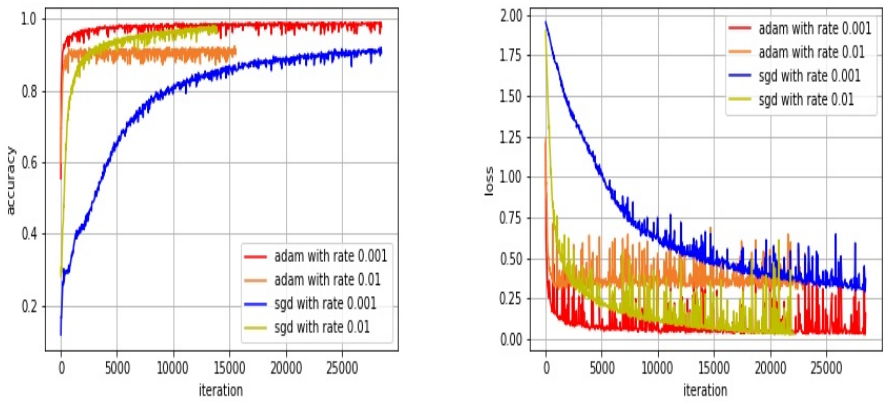


Figure 5.10: The architecture of modified LeNet-5.

Firstly, randomly selecting (75%) training and (25%) test set is performed to see whether the performance of CNN can be comparable with eigenface. During training, two optimization methods are employed to compare the performance, namely stochastic gradient descent (SGD) [47] and Adam [48]. The performance with different learning rate is shown in Figure 5.11, we can see that Adam algorithm generally allows the model to converge quickly, however, it can get stuck if the learning rate is too large, SGD approach can always get a comparable performance in accuracy but the speed of convergence highly depends on learning rate. Tuning appropriate learning rate for SGD is a time-consuming task to get a comparable convergence speed. Hence, the Adam optimization method is employed throughout the remaining tasks.



(a) The training set accuracy by different optimization algorithms and learning rates.

(b) The training set loss by different optimization algorithms and learning rates.

Figure 5.11: The adam and sgd optimization algorithms on the LIMA set.

From the Table 5.8, the performance of CNN illustrate that the CNN can easily beat eigenface, the precision, recall and f1-score are increased by 2%. However, this experiment is still not representative since the distributions of training and test set are too similar as discussed above.

labels	precision	recall	f1-score	support
0	0.87	0.77	0.82	715
1	0.96	0.98	0.97	2780
2	0.98	0.99	0.99	1564
3	0.88	0.92	0.90	449
4	0.97	0.99	0.98	1999
5	0.98	0.95	0.96	993
6	0.98	0.99	0.99	1210
avg/total	0.96	0.96	0.96	9710

Table 5.8: The CNN method with Adam learning rate 0.001 on LIMA face dataset detected by Faster RCNN with up-sampling rate 2, and test set is random selected.

Another experiment is done to see the effect of different distributions of training and test set, and session S09 is still used as a test set, the left chart in Figure 5.12 illustrates the performance decline significantly, the average of recall and f1-score decreases from 0.96, 0.96 to 0.74 and 0.80 respectively. Nevertheless, the performance of CNN (0.79 average recall) is still much better than eigenface with SVM (0.52 average recall). However, the accuracy on the training set is approximately 0.95 and it is much better than test set, the model is over-fitted to the training set and does not generalize well. To improve the generalization, some strategies below can be employed:

- collect more data.
- choose simpler model.
- add regularization in neural network.

Presently, the model is already simple enough with only a few layers, regularization has already been performed in the network, and we only detect these number of faces and cannot collect more face from LIMA. However, data augmentation can be performed to improve the generalization, real-time data augmentation approach is utilized instead of conventional augmentation methods, traditional augmentation methods require the data to be generated before training, it dramatically expands the size of dataset and increases the training time. Nonetheless, through real-time augmentation it can improve generalization while at the same time reducing the need for a larger dataset.

The real-time data augmentation is done by CPU while the model is busy training on GPU, this can make the data augmentation much convenient and faster. Some images will be randomly flipped left to right or right to left, and some images will be randomly rotated from -90 degrees to 90 degrees. The rotation method greatly improves the generalization since there are many images containing faces which have rotation angels between -90 to 90 degrees in our test set as mentioned in the eigenface section. The result is shown on the right-hand side of Figure 5.12.

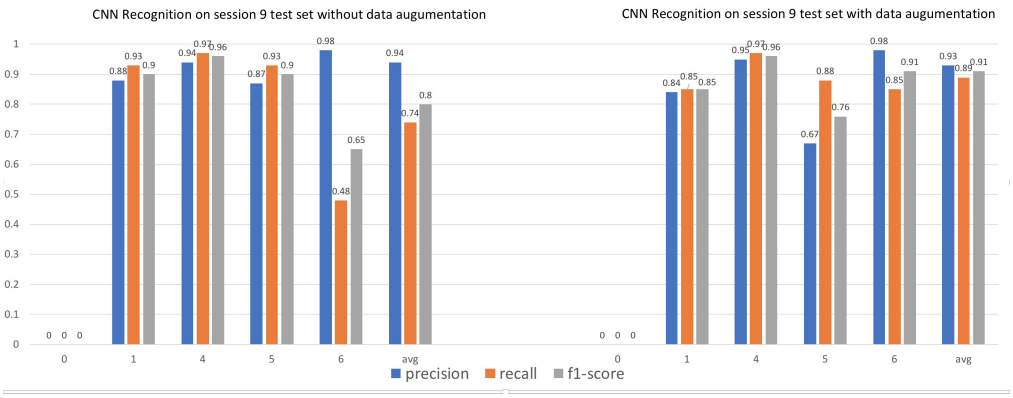


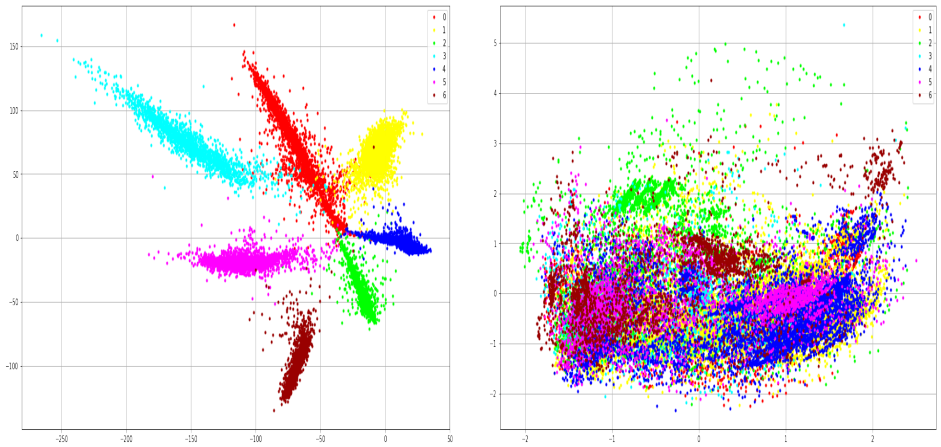
Figure 5.12: The CNN performance on session 09 test set with or without data augmentation.

The recall and f1-score for label 6 increases from 0.48, 0.65 to 0.85 and 0.91 as the faces in images with label 6 have large rotation angles and the data augmentation indeed improves the generalisation. However, the performance for some labels drops slightly, the main reason is that the accuracy on the training set is still approximately 0.92. The model slightly under-fits the training set, with more complex models and more training iterations could help the model solve the issues.

5.3 Comparison between Features Learnt by CNN and Eigenface Features

Robustness of two features (eigenface and features learnt by CNN) can be argued. The 2 dimension features are used instead of other high dimensions since 2 dimension features are straightforward to visualise.

The eigenface 2 dimension features are shown in Figure 5.13b, and the Figure 5.13a shows the learnt 2 dimension features by CNN. It is evident that eigenface features for the majority of classes are indistinguishable and when the face of a single person is at different poses, the eigenface outputs very different feature vectors as shown in Figure 5.13b. The kernel function in SVM could map the eigenface features to higher dimensions so that SVM could still find a good hyperplane in that space. Nevertheless, the features learnt by CNN can be much easier to separate and when the face of a single person is at different poses, the CNN still outputs very similar feature vectors. Although the 2 dimension for eigenface is not sufficient, it is still evident that the features learnt by CNN are much robust than eigenface features.



(a) The learnt 2-d features by CNN with LIMA face dataset provided by the Faster-RCNN with up-sampling rate 2.

(b) The eigenface 2-d features.

Figure 5.13: The features learnt by two different approaches.

6 Re-Identification via Appearance Features

Appearance features (including the colour of the garment, face, hair and so forth) can be more useful than facial features in some cases. From the ground truth for face, we can see that only 32% of images contains a face. Considering the image in Figure 6.1, a facial feature is no longer available but appearance feature can be used to recognize the person.

Figure 6.1: The face is not visible.

Firstly, this method also experimented on randomly selected training set and test set to show that what CNN learns from images and whether the learnt features is representative or not. The network architecture is the same as face recognition in the last section, but instead of 32x32 image size, a slightly larger size is used as 32x32 is too small for appearance.

There have some potential choices for size, e.g. 64x64, 128x128, 256x256, 224 x 224 and 299x299. 224x224 and 299x299 are commonly used in VGG net [49] and inception-v3 net [50]. Nevertheless, the total amount of CPU memory required for the whole LIMA dataset is approximately 8.6 GB, 34.5 GB, 138 GB, 105 GB, 188 GB for size 64, 128, 256, 224 and 299 respectively. The available of memory we have is about 16 GB memory and some additional swap spaces, 64x64 size is chosen as the other sizes cannot be fully fitted into the memory and some images have to stored in the swap spaces and accessing swap spaces is much expensive than accessing a memory. The Table 6.1 shows the comparison of training speed and total loading time to the memory for 64x64 and 128x128 and it shows both training speed and total loading time for size 128x128 is much slower and longer than 64x64. Although 64x64 is still too small for appearance, larger images require larger memory and we can see that the performance of 64x64 size still achieves good performance. Hence, 64x64 is still a reasonable choice.

image size	64x64	128x128
training speed per epoch (s)	24.8	1560
Total load time (s)	335	988

Table 6.1: The training speed and total loading time to memory for two different sizes.

labels	precision	recall	f1-score	support
0	1	0.99	1	12428
1	1	1	1	10341
2	1	1	1	5242
3	0.99	1	0.99	4932
4	1	0.99	1	5164
5	0.99	0.99	0.99	3750
6	0.99	0.99	0.99	5250
avg/total	1	1	1	47107

Table 6.2: CNN with appearance features' results on LIMA test sets (random selected)

In Table 6.2, the performance achieves nearly 100 percent. The reason why it performs so well is same as before, the distribution of training set and test set are too similar. The below showed some mis-classified images in Figure 6.2, the Figure 6.2a and Figure 6.2b should be labelled as 1, but they are predicted as 1 and 5 respectively. When the white cloth's region increases, the model, in fact, predicts it as 5 instead of 1 shown in Figure 6.2b, the only difference between two images is the white cloth dominates the second image. Furthermore, the white garment is usually worn by label 3, 5, 6. Therefore, the CNN start to think that the 5 is more possible, even it has a clear face in it and the learnt features for CNN is not representative and can be easily defeated by only some small modifications.

(a) Label is 0 but predict 1

(b) Label is 0 but predict 5

Figure 6.2: The mis-classified images.

The performance of appearance features is shown below when the distribution of training and test set are different. Meanwhile, one technique is chosen to use is the early stopping. Early stopping is a regularization technique that can be used to prevent the overfitting. This can be achieved by calculating the accuracy or other metrics every epoch (1 epoch = iterate over all training sets), and the model is saved to the disk every epoch. Therefore, a relatively good model which has higher accuracy on both training and test set can be selected among all models. The following model can achieve relative higher results on both training set and test set than others. On the test set, it achieves about 0.79 accuracy.

labels	precision	recall	f1-score	support
0	0.78	0.79	0.79	1288
1	0.97	0.91	0.94	2563
2	0.00	0.00	0.00	0
3	0.00	0.00	0.00	0
4	0.97	0.72	0.83	4025
5	0.98	0.88	0.93	1512
6	0.91	0.76	0.83	5106
avg/total	0.94	0.79	0.86	14494

Table 6.3: The performance of CNN results on LIMA with appearance features on LIMA test sets (session 09).

(a) Label is 5 but predict 1

(b) Label is 6 but predict 2

(c) The training set contains a lot of this images with label 2.

Figure 6.3: The mis-classified images on LIMA test set (session 09).

The Figure 6.3 shows some mis-classified images in test set, it is obvious that sometimes the appearance features are not reliable, even there is a face in front of the camera in the image, but the model still predicts 1 instead of 5 shown in Figure 6.3a as the majority of region in the image is black and label 1 usually wears black cloth. Furthermore, the Figure 6.3b shows that the image has label 6 and predicts it as 2, because the model is over-fitted to the training set. The 6.3c shows that there are many images which has the label 2 with white cloth women lying on the sofa in the training set, but not much images which has label 6 with white cloth women lying on the sofa are in the training set. Therefore, this model, in fact, remembers this situation that a person wearing a white cloth and lying on the sofa is label 2. In terms of appearance features, 6.3b and 6.3c indeed look very similar, the appearance feature is not really reliable.

7 Combining Appearance and Facial Features

Only appearance and only facial features all have some limitations, combing facial and appearance features can be a good way to improve the performance. It is motivated by that facial features of a person do not change significantly while the style or colour of garment, hair, shoes may be changed in videos, especially the videos are captured in the home environment. Furthermore, the appearance actually includes the face. However, the area of a face region is much smaller than the area of an image. Hence, the face region will not contribute much to appearance features, but a facial feature can be very discriminative if an image contains a face. There are many ways to combine facial and appearance features together, two potential ways will be discussed in this section.

7.1 Early Fusion

Regarding early fusion, we can extract a face from an image and combine the face image and the original image into one image (e.g. 6 channels, an image in 1,2,3 channels and a face image in 4, 5, 6 channels), then the image is fed into a CNN to train. However, the size of an image is generally larger than a face image, the face image can either be resized to the image size or add zero padding. If the image is resized, the face may be too blurred and the CNN may not extract useful features from it. If the zero padding is used and the image has a very large size (e.g. 224x224) and face has a small size (e.g. 32x32), then the majority of values in 4, 5 and 6 channels will be 0. Since we want to increase the weight of facial features, zero padding could not expand the influence of facial features. However, it cannot be concluded whether this method works or not without doing an experiment. We mainly focus on late fusion method for simplicity in this section.

7.2 Late Fusion

We propose a 2-stream convolutional neural network to combine both face and appearance features. The proposed 2-stream convolutional neural network is shown in Figure 7.1. The face feature is computed by first extracting the face out of an image and compute the 128 dimension feature vector to represent the face through a convolutional neural network, this is used to expand the influence of the facial feature as the facial feature is only tiny part of appearance but can be very robust. If there is no face in the image, we directly output a feature vector of 0 of length 128. Then, a second network is used to extract the appearance features, the two CNNs does not share weights between each other since we may want two CNNs to extract different representations for different things. After outputting 2 feature vectors, the facial and appearance features may have the same weights as the length of two features are the same. Therefore, the late fusion has probability to expand the influence of facial features and the late fusion could be a more nature way to combine features. Then, two 128-d

feature vectors are concatenated together into a big 256 feature vector and followed by a few fully connected network. The fully connected network after the concatenation is used to find a better function to decide when the facial feature is more important or when the appearance feature is more important than face.

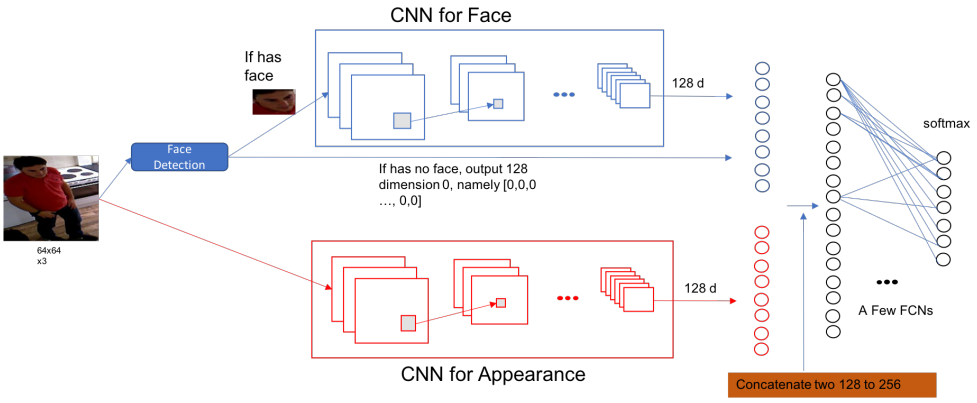


Figure 7.1: The proposed of general 2 stream CNN to combine both face and appearance features.

7.3 Training Strategies for the Proposed 2 Stream CNN

This 2 stream CNN is quite difficult to train since some images do not contain a face, but there are many ways to train this 2 stream convolutional neural network. For simplicity, two simple training strategies are described in this section, and other training strategies will be discussed in the future work section.

- The first stage is to train the two CNN separately. After the training of two CNNs, a few fully connected networks are trained in the second stage by using the output of two CNNs. In this second stage, the two CNNs are not trained and the parameters in two CNNs should be fixed. After the second stage, the whole network has all be trained and ready to be used.

- The face CNN is trained first and the appearance CNN is trained by the image and the output of the face network. There are many existing face pre-trained model across the web, therefore, the face network does not has to be trained from scratch, it can be relatively easier and reduce the amount of training time.

7.4 The Performance of 2 Stream CNN.

The second training strategy is chosen to train the network for simplicity. The OpenFace [51] pre-trained CNN is used as the face network to generate facial features. The facial features are then used as an input to the network, the network architecture for appearance feature is the same as CNN for face recognition. After combing two features, a few fully connected layers are introduced. These layers are used to learn a non-linear

function such that automatically find a weight to balance the face features and appearance features. The performance is shown below, we can see that the recall is increased from 0.79 to 0.90, it is approximately 10% improvement. However, there are two potential reasons why the performance

labels	precision	recall	f1-score	support
0	0.85	0.80	82	1288
1	0.85	0.97	0.91	2563
2	0.00	0.00	0.00	0
3	0.00	0.00	0.00	0
4	0.97	0.93	0.95	4025
5	0.96	0.95	0.96	1512
6	0.96	0.85	0.90	5106
avg/total	0.93	0.90	0.91	14494

Table 7.1: 2SCNN Results on LIMA session 09

increases dramatically on the test set. Firstly, the network may suddenly find a good local minimum so that the learnt parameters generalise the model well but the facial feature does not make any improvement. Secondly, the facial feature indeed improves the network to perform well. A simple experiment can be designed to test whether the facial features do improve the performance or not:

1. Store mis-classified images by the model with appearance features only to folder A.

2. Store mis-classified images by the 2 stream CNN model to folder B.

3. Compare these two folders, if the image appears in the folder A, but it disappears in the folder B and stores these images in folder C. Then, these images are corrected by the 2 stream CNN model.

4. Furthermore, the face detection can be performed on these images in folder C. If the majority of these images contain faces, then, it could say that the facial features do help the model to classify these images correctly.

5. Face detection can also be done on images in folder B, this can further provide evidence that the facial features do improve the network if the majority of images do not contain face in folder B as these images are mis-classified by the 2 stream CNN model.

All mis-classified images with label 6 are used to do the above experiment, the results are shown in table 7.2. It is obvious that about 960 images are in the folder C which is corrected by 2 stream model. Furthermore, about 67% images in the folder C contain a face, and only about 33% images in the folder B contain a face. It is evident that the majority of corrected images by 2-Stream model contain a face and the majority of mis-classified images by 2-Stream model does not contain a face. Therefore, it can be concluded that the facial features indeed improve the performance.

	folder B	folder C
the number of images containing a face	252	634
the number of images not containing a face	511	326
the percent of images containing a face	33%	67%
total	763	960

Table 7.2: Face detection results for folder B and folder C.

The image in Figure 7.2a is very similar to the image in Figure 7.2b. Nevertheless, the 2 stream model is able to classify the image in Figure 7.2b correctly but it cannot correctly classify the image in Figure 7.2a. The face detection is performed on both images, a result shows that Faster-RCNN cannot detect a face from an image in Figure 7.2a even with up-sampling rate 4, but Faster-RCNN can detect a face from an image in Figure 7.2b. It provides a further evidence that the facial feature can be a very discriminative feature to predict the person's identity in the 2 stream model. Meanwhile, if we can improve the performance of face detections, 2 stream model would enhance the recognition performance as well.

(a) The image is mis-classified by the 2 stream model.

(b) This image is mis-classified by model with appearance features only, but it is correctly classified by 2 stream model.

Figure 7.2: Two images are very similar, but 2 stream model can only recognise the right image correctly.

Some mis-classified images by 2 stream CNN model are shown in Figure 7.3 but these images are not mis-classified by the model with only appearance features. Clearly, the weight of facial features has a larger weight than appearance features in this case, the facial features make the performance worse for some images which do not contain a face.

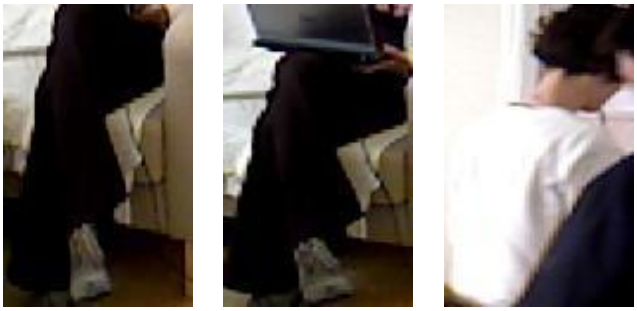


Figure 7.3: The mis-classified images by 2-stream model.

8 Conclusion

In section 2, we investigated some potential approaches to do re-identification based on facial and appearance features.

In section 3, we made some introductions about three subsets from LIMA dataset and discussed the advantages and disadvantages for each subset.

In section 4, we discussed the procedure of how to label the images and investigated some potential software for annotation. Furthermore, we designed a simple software to satisfy our purpose. We then discussed some challenges about labelling an image.

In section 5, we provided detection results on LIMA dataset by two approaches and the results of two recognition approaches on LIMA dataset were provided as well. Meanwhile, we did some experiments when the distribution of training and test set are similar and when they are different in order to see whether the trained model is generalized well or not. We then investigated some data augmentation techniques to improve the model's generalisation. Then, we compared the learnt features by CNN and eigenface, and we concluded that the features learnt by CNN are much robust than eigenface.

In section 6, we provided the re-identification results on LIMA dataset by using appearance features, we then discussed the drawbacks of the appearance features.

In section 7, we discussed some potential way to combine the facial and appearance features together. We then proposed a 2 Stream Convolutional Neural Network to do re-identification on LIMA dataset, and provided some potential ways to train the 2 Stream CNN. Furthermore, we designed a simple experiment to show the effectiveness of facial features. Based on the experiment' results, we concluded that the facial features indeed help the model to improve the performance.

9 Future Work

9.1 Potential Way to Improve 2 Stream CNN

The strategy of training 2-stream convolutional neural network is annoying as it all contains two stages of training. The face network cannot be trained if there is no face detected, this prevents us to train the network at once. It could be fixed by using the images which all contain a face. However, this may not generalise the appearance feature well as there is no images containing the human back and other poses. Meanwhile, concatenation may not be a good way to combine features, more clever way could be used, for example, max or average of two features.
Currently, the size for an input image is 64x64, 64x64 is actually too small for appearance features, the performance of the network may improve if a larger input size is used.

9.2 Early Fusion

In section 7, we have described some general drawbacks of the early fusion method. However, we need some evidence to see whether early fusion method works well or not.

9.3 Complex Network Architectures and Fine-Tune

Presently, the 2 stream network uses a simple and shallow network, some more complex network architectures could be a good way to improve the performance, e.g. VGG net [49], Alex net [6], ResNet [52], inception-v3 [50], inception-v4 [53]. Meanwhile, these architecture has some pre-trained model for different tasks, fine-tune can be performed for these network without training from scratch since different tasks may share similar low-features.

A Appendix

A.1 Personal Designed Tool

In this section, we will discuss some design choices for the personal annotation tool for labelling face.

A.1.1 Design Choice

- Mouse or Keyboards:
  The majority of existing tools require mouse to click the button, although, using mouse to click the button or moving the mouse to click another button is inefficient as it takes time to move mouse among buttons. For this massive number of images, the small amount of mouse moving time would have huge impacts. However, misclassified rate would be slightly increased considering the use of keyboards might be too efficient and it requires human brains to react in a tiny amount of time, but this can be easily solved by either slowing down a bit or doing a second review. Therefore, keyboards would still be an reasonable desire for this motive.

- Small, Median Or Even Larger Images:
  Images in the dataset have large variations in terms of the image size, a few images are smaller than 64x64 and some images are larger than 224 x 224. Larger images will be relatively simpler for human to recognize. Consequently, the image is resized to 2 times bigger while it is displayed on the screen in order to improve the accuracy. Three or more times is not always helpful as resizing will blur images and introduce some artefacts.

A.1.2 Implementation

The subset 3 of LIMA dataset has been chosen to perform face labelling as subset 3 has fixed some issue of the subset 1 and it is exactly the same as subset 2 and offers additional features.

- Main Idea:
  Firstly, it reads a list of image files' names and sorts the list with increasing order so that it can follow the temporal order since human can use the temporal features between two frames. Based on the current image, it can be straightforward to determine whether the next image contains a face or not. After that, we can display the image and click the key either "a" or "d", and move the image to corresponding folders (either face or non-face).

- Packages Choice:
  There are many GUI packages can be used to visualize the image, e.g. scikit-image [54], OpenCV [55] and so forth. Both OpenCV and Scikit-image are packages containing high quality image processing algorithms and both are convenient to resize and visualize

image. Nevertheless, Scikit-image is relatively difficult than OpenCV to capture keyboard input. Then, OpenCV is chosen to support the annotation tool.

References

[1] D. Chung, K. Tahboub, and E. J. Delp. "A Two Stream Siamese Convolutional Neural Network for Person Re-identification". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 1992–2000. DOI: 10.1109/ICCV.2017.218.

[2] R. Layne et al. "A Dataset for Persistent Multi-target Multi-camera Tracking in RGB-D". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 1462–1470. DOI: 10.1109/CVPRW.2017.189.

[3] Michael Jones Paul Viola. "Rapid Object Detection using a Boosted Cascade of Simple Features". In: *Computer Vision and Pattern Recognition*. 2001.

[4] Yoav Freund and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In: *Computational Learning Theory: Eurocolt '95*. Springer-Verlag, 1995, pp. 23–37.

[5] Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. 2005, pp. 886–893. DOI: 10.1109/CVPR.2005.177. URL: https://doi.org/10.1109/CVPR.2005.177.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[7] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[8] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Computer Vision and Pattern Recognition*. 2014.

[9] J. R. R. Uijlings et al. "Selective Search for Object Recognition". In: *International Journal of Computer Vision* 104.2 (2013), pp. 154–171. URL: https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013.

[10] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. "Efficient Graph-Based Image Segmentation". In: *International Journal of Computer Vision* 59.2 (2004), pp. 167–181. ISSN: 1573-1405. DOI: 10.1023/B:VISI.0000022288.19776.77. URL: https://doi.org/10.1023/B:VISI.0000022288.19776.77.

[11] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *CoRR* abs/1506.01497 (2015). arXiv: 1506.01497. URL: http://arxiv.org/abs/1506.01497.

[12] Ross B. Girshick. “Fast R-CNN”. In: *CoRR* abs/1504.08083 (2015). arXiv: 1504.08083. URL: http://arxiv.org/abs/1504.08083.

[13] M. Everingham et al. “The Pascal Visual Object Classes Challenge: A Retrospective”. In: *International Journal of Computer Vision* 111.1 (Jan. 2015), pp. 98–136.

[14] L. Sirovich and M. Kirby. “Low-Dimensional Procedure for the Characterization of Human Faces”. In: *Journal of the Optical Society of America A* 4.3 (1987), pp. 519–524.

[15] Corinna Cortes and Vladimir Vapnik. “Support-Vector Networks”. In: *Mach. Learn.* 20.3 (Sept. 1995), pp. 273–297. ISSN: 0885-6125. DOI: 10.1023/A:1022627411411. URL: https://doi.org/10.1023/A:1022627411411.

[16] Florian Schroff, Dmitry Kalenichenko, and James Philbin. “FaceNet: A Unified Embedding for Face Recognition and Clustering”. In: *CoRR* abs/1503.03832 (2015). arXiv: 1503.03832. URL: http://arxiv.org/abs/1503.03832.

[17] Timothy Huang and Stuart Russell. “Object Identification in a Bayesian Context”. In: *Proceedings of the Fifteenth International Joint Conference on Artifical Intelligence - Volume 2*. IJCAI’97. Nagoya, Japan: Morgan Kaufmann Publishers Inc., 1997, pp. 1276–1282. ISBN: 1-555860-480-4. URL: http://dl.acm.org/citation.cfm?id=1622270.1622340.

[18] Douglas Gray and Hai Tao. “Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features”. In: *Computer Vision – ECCV 2008*. Ed. by David Forsyth, Philip Torr, and Andrew Zisserman. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 262–275. ISBN: 978-3-540-88682-2.

[19] Bryan Prosser et al. “Person Re-Identification by Support Vector Ranking”. In: *Proceedings of the British Machine Vision Conference*. doi:10.5244/C.24.21. BMVA Press, 2010, pp. 21.1–21.11. ISBN: 1-901725-40-5.

[20] W. S. Zheng, S. Gong, and T. Xiang. “Reidentification by Relative Distance Comparison”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.3 (2013), pp. 653–668. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2012.138.

[21] Alexis Mignon and Frederic Jurie. *PCCA: A new approach for distance learning from sparse pairwise constraints*. June 2012.

[22] M. Farenzena et al. “Person re-identification by symmetry-driven accumulation of local features”. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010, pp. 2360–2367. DOI: 10.1109/CVPR.2010.5539926.

[23] Yutian Lin et al. “Improving Person Re-identification by Attribute and Identity Learning”. In: *CoRR* abs/1703.07220 (2017). arXiv: 1703.07220. URL: http://arxiv.org/abs/1703.07220.

[24] L. Bazzani et al. “Multiple-Shot Person Re-identification by HPE Signature”. In: *2010 20th International Conference on Pattern Recognition*. 2010, pp. 1413–1416. DOI: 10.1109/ICPR.2010.349.

[25] Martin Hirzer et al. “Person Re-identification by Descriptive and Discriminative Classification”. In: *Image Analysis*. Ed. by Anders Heyden and Fredrik Kahl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 91–102. ISBN: 978-3-642-21227-7.

[26] Liang Zheng, Yi Yang, and Alexander G. Hauptmann. “Person Re-identification: Past, Present and Future”. In: *CoRR* abs/1610.02984 (2016). arXiv: 1610.02984. URL: http://arxiv.org/abs/1610.02984.

[27] Martin Hirzer et al. “Person Re-Identification by Descriptive and Discriminative Classification”. In: *Proc. Scandinavian Conference on Image Analysis (SCIA)*. 2011.

[28] Xiaolong Ma et al. “Person Re-Identification by Unsupervised Video Matching”. In: *CoRR* abs/1611.08512 (2016). arXiv: 1611.08512. URL: http://arxiv.org/abs/1611.08512.

[29] Tetsu Matsukawa et al. “Hierarchical Gaussian Descriptors with Application to Person Re-Identification”. In: *CoRR* abs/1706.04318 (2017). arXiv: 1706.04318. URL: http://arxiv.org/abs/1706.04318.

[30] N McLaughlin, J Martinez del Rincon, and P Miller. “Recurrent Convolutional Network for Video-based Person Re-Identification”. In: *CVPR*. 2016.

[31] Liang Zheng et al. “Scalable Person Re-identification: A Benchmark”. In: *Computer Vision, IEEE International Conference on*. 2015.

[32] Ergys Ristani et al. “Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking”. In: *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*. 2016.

[33] W. W. W. Zou and P. C. Yuen. “Very Low Resolution Face Recognition Problem”. In: *IEEE Transactions on Image Processing* 21.1 (2012), pp. 327–340. ISSN: 1057-7149. DOI: 10.1109/TIP.2011.2162423.

[34] Zhangyang Wang et al. “Studying Very Low Resolution Recognition Using Deep Networks”. In: *CoRR* abs/1601.04153 (2016). arXiv: 1601.04153. URL: http://arxiv.org/abs/1601.04153.

[35] Professor Majid Mirmehdi. *Work Package 2: Video Monitoring*. 2015. URL: http://www.irc-sphere.ac.uk/work-package-2 (visited on 04/09/2018).

[36] Labelbox. *Labelbox*. https://github.com/Labelbox/Labelbox. 2018.

[37] Images Annotation Programme. *Images Annotation Programme*. https://github.com/frederictost/images_annotation_programme. 2018.

[38] dataturks. *dataturks*. https://dataturks.com/. 2018.

[39] Gary B. Huang et al. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. rep. 07-49. University of Massachusetts, Amherst, 2007.

[40] Vidit Jain and Erik Learned-Miller. *FDDB: A Benchmark for Face Detection in Unconstrained Settings*. Tech. rep. UM-CS-2010-009. University of Massachusetts, Amherst, 2010.

[41] Peter M. Roth Martin Koestinger Paul Wohlhart and Horst Bischof. "Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization". In: *Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*. 2011.

[42] O. M. Parkhi, A. Vedaldi, and A. Zisserman. "Deep Face Recognition". In: *British Machine Vision Conference*. 2015.

[43] Shuo Yang et al. "WIDER FACE: A Face Detection Benchmark". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[44] H. W. Ng and S. Winkler. "A data-driven approach to cleaning large face datasets". In: *2014 IEEE International Conference on Image Processing (ICIP)*. 2014, pp. 343–347. DOI: 10.1109/ICIP.2014.7025068.

[45] Y. LeCun et al. "Gradient-Based Learning Applied to Document Recognition". In: *Intelligent Signal Processing*. IEEE Press, 2001, pp. 306–351.

[46] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *CoRR* abs/1502.03167 (2015). arXiv: 1502.03167. URL: http://arxiv.org/abs/1502.03167.

[47] J. Kiefer and J. Wolfowitz. "Stochastic Estimation of the Maximum of a Regression Function". In: *Ann. Math. Statist.* 23.3 (Sept. 1952), pp. 462–466. DOI: 10.1214/aoms/1177729392. URL: https://doi.org/10.1214/aoms/1177729392.

[48] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014). arXiv: 1412.6980. URL: http://arxiv.org/abs/1412.6980.

[49] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014). arXiv: 1409.1556. URL: http://arxiv.org/abs/1409.1556.

[50] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *CoRR* abs/1512.00567 (2015). arXiv: 1512.00567. URL: http://arxiv.org/abs/1512.00567.

[51] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. *OpenFace: A general-purpose face recognition library with mobile applications*. Tech. rep. CMU-CS-16-118, CMU School of Computer Science, 2016.

[52] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[53] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *CoRR* abs/1602.07261 (2016). arXiv: 1602.07261. URL: http://arxiv.org/abs/1602.07261.

[54] Stéfan van der Walt et al. "scikit-image: image processing in Python". In: *PeerJ* 2 (June 2014), e453. ISSN: 2167-8359. DOI: 10.7717/peerj.453. URL: http://dx.doi.org/10.7717/peerj.453.

[55] Itseez. *Open Source Computer Vision Library*. https://github.com/itseez/opencv. 2015.