IRIS RECOGNITION SYSTEM (IRS) USING
DEEP LEARNING TECHNIQUE

YOW SUE CHIN

UNIVERSITI SAINS MALAYSIA
2019
IRIS RECOGNITION SYSTEM (IRS) USING DEEP LEARNING TECHNIQUE

by

YOW SUE CHIN

Thesis submitted in fulfilment of the requirements for the degree of Bachelor of Engineering (Electronic Engineering)

June 2019
ACKNOWLEDGEMENT

The dissertation is dedicated to everyone in the field of image processing system who embarks the journey of expanding the collection of knowledge and transcendent passion for improving Iris Recognition System.

First and foremost, I wish to thank Universiti Sains Malaysia for giving me this opportunity to have my Final Year Project in fulfilling the Bachelor Degree of Electronic Engineering and also support me in terms of the facilities needed. I would like to convey my appreciation to all the lecturers who are delivering their knowledge to me throughout my studies conscientiously.

My uttermost gratitude goes to Dr. Ahmad Nazri Ali, my thesis advisor, and project supervisor, for his invaluable support and guidance that supported me throughout my project. With his valuable guidance and practical advice during the course of this research, various difficulties and challenges that I encountered were able to deal with. I am truly grateful for any form of help and cooperation offered to me throughout this duration.

My completion of this project could not be accomplished without the support of my beloved family and friends, thank you for their spiritual and mental support. It is the major motivation for me to keep up my pace in completing this project successfully.

Last, but by no means least, I would like to thank the staff of the School of Electrical & Electronic Engineering, Universiti Sains Malaysia (USM) for their kind cooperation. My heartfelt thanks.
# TABLE OF CONTENTS

ACKNOWLEDGEMENT ......................................................................................................................... i

TABLE OF CONTENTS ........................................................................................................................... ii

LIST OF TABLES .......................................................................................................................................... v

LIST OF FIGURES ......................................................................................................................................... vi

LIST OF ABBREVIATIONS ........................................................................................................................... viii

ABSTRAK .................................................................................................................................................. ix

ABSTRACT ................................................................................................................................................. x

CHAPTER 1 INTRODUCTION ..................................................................................................................... 1

1.1 Research Background ......................................................................................................................... 1

1.2 Problem Statement ............................................................................................................................. 3

1.3 Research Objectives .......................................................................................................................... 5

1.4 Project Scheme .................................................................................................................................... 5

1.5 Thesis Organization ............................................................................................................................ 5

CHAPTER 2 LITERATURE REVIEW .......................................................................................................... 6

2.1 Introduction .......................................................................................................................................... 6

2.2 Biometrics System ............................................................................................................................. 6

2.2.1 Human Iris Anatomy ..................................................................................................................... 7

2.2.2 Iris Recognition ............................................................................................................................. 8

2.3 Computer Vision (CV) ....................................................................................................................... 10

2.4 Artificial Intelligence (AI) ................................................................................................................ 11

2.4.1 Machine Learning (ML) ............................................................................................................... 12

2.4.2 Deep learning (DL) ....................................................................................................................... 13

2.4.3 Difference between ML and DL ................................................................................................... 15

2.5 Convolution Neural Network (ConvNets) ......................................................................................... 16

2.5.1 Convolution Layer / Filtering ....................................................................................................... 19

2.5.2 Nonlinearity .................................................................................................................................... 22

2.5.3 ImageNet Large Scale Visual Recognition Challenge, ILSVRC .............................................. 23

2.6 Transfer Learning ............................................................................................................................... 23
2.7 Previously Proposed Classical Machine Learning Method for Iris Recognition ................................................................. 25
2.8 Previously Proposed ConvNets for Iris Recognition ................................................................. 27
2.9 Summary ................................................................................................................................................. 29
CHAPTER 3 METHODOLOGY ................................................................................................................. 30
3.1 Introduction ........................................................................................................................................... 30
3.2 Project Workflow ............................................................................................................................... 30
3.2.1 Prerequisite requirement and preliminary setup......................................................................................... 31
  3.2.1.1 Hardware ........................................................................................................................................ 31
  3.2.1.2 Software ....................................................................................................................................... 31
  3.2.1.3 Iris datasets .................................................................................................................................. 32
3.2.2 Image Pre-processing ............................................................................................................................. 32
  3.2.2.1 Image Resize .................................................................................................................................. 32
  3.2.2.2 Image Dataset Allocation .............................................................................................................. 32
  3.2.2.3 Image Augmentation ...................................................................................................................... 33
3.2.3 Transfer Learning technique .................................................................................................................. 34
3.2.4 Pre-trained ConvNet Model .................................................................................................................. 36
  3.2.4.1 AlexNet ........................................................................................................................................ 36
  3.2.4.2 DenseNet201 ................................................................................................................................. 38
3.2.5 Bayesian Optimization .......................................................................................................................... 39
3.2.6 Performance Analysis .......................................................................................................................... 40
  3.2.6.1 Analysis of Classification Process ................................................................................................. 40
  3.2.6.2 Quantitative Analysis of Confidence Levels and Confidence Interval ............................................... 42
  3.2.6.3 Assessment of Algorithm Complexity .......................................................................................... 42
  3.2.6.4 Quantitative Analysis of the System ............................................................................................. 43
3.3 Summary .................................................................................................................................................. 43
CHAPTER 4 RESULTS AND DISCUSSIONS ...................................................................................... 44
4.1 Introduction ......................................................................................................................................... 44
4.2 Result and discussion for Pre-processing of Iris Image ........................................................................ 44
  4.2.1 Data Augmentation ......................................................................................................................... 44
4.3 Result and discussion of the transferability of pre-trained model ......................................................... 46
4.4 Result and discussion of algorithm performances regardless of Data Augmentation ................................................................. 48
4.5 Result and discussion for Bayesian Optimization .......................... 50
  4.5.1 Bayesian Optimization on AlexNet model .......................... 50
  4.5.2 Bayesian Optimization on DenseNet201 model ................. 52
4.6 Result and performance of layer freezing on training progress .......... 53
4.7 Result and performance of algorithm performances of the final algorithm 54
4.8 Summary .............................................................................. 56
CHAPTER 5 CONCLUSION ................................................................ 57
  5.1 Conclusion ........................................................................... 57
  5.2 Suggestion for future implementation .................................. 58
REFERENCES ............................................................................. 59
APPENDICES ............................................................................. 63
APPENDIX A MATLAB code for algorithm development
APPENDIX B SVM Classifier Training
APPENDIX C Feature Map extracted from AlexNet model
APPENDIX D Precision of proposed trained algorithm
LIST OF TABLES

Table 2.1 : Comparison of Various biometric techniques based on biometric traits [1] .................................................................................................................. 7
Table 2.2 : Performance Evaluation[1] ........................................................................ 8
Table 2.3 : Various iris dataset [16] ............................................................................ 9
Table 2.4 : Heterogeneity of iris dataset [16] .............................................................. 10
Table 3.1 : Parameters for data augmentation ........................................................... 33
Table 3.2 : Feature of AlexNet model ...................................................................... 36
Table 3.3 : Layer Specific feature for DenseNet family ............................................ 39
Table 3.4 : Range of hyperparameters variables ..................................................... 40
Table 3.5 : Standard confidence intervals ................................................................. 42
Table 4.1 : Transferability of pre-trained ConvNet model on CASIA-Iris-Interval V1 dataset ........................................................................................................ 47
Table 4.2 : Transferability of pre-trained DenseNet201 model on CASIA-Iris-Interval V1 dataset ........................................................................................................ 47
Table 4.3 : Performances analysis on AlexNet model regardless of data augmentation. A total of 324 training datasets. ....................................................... 49
Table 4.4 : Optimization result on Initial Learning Rate, Momentum and L2Regularization trained on AlexNet model with CASIA-Iris-Interval V1 dataset ........................................................................................................ 50
Table 4.5 : Optimization result on Initial Learning Rate, Momentum and L2Regularization trained on DenseNet201 model with CASIA-Iris-Interval V1 dataset ........................................................................................................ 52
Table 4.6 : Recognition Accuracy Rate versus Layer Freezing on DenseNet201 model with CASIA-Iris-Interval V1 dataset ...................................................... 54
Table 4.7 : Specification details and performances of re-trained model based on proposed technique ........................................................................................................ 55
Table D.1: Precision for AlexNet model on CASIA-Iris-Interval V1 dataset
Table D.2: Precision for DenseNet201 model on CASIA-Iris-Interval V1 dataset
LIST OF FIGURES

Figure 2.1 : Front view of a human eye with blue iris and a top view of the right eye [14] ................................................................. 8

Figure 2.2 : NIR Image [15] ........................................................................................................ 10

Figure 2.3 : Visible Wavelength Image[15] .............................................................................. 10

Figure 2.4 : Venn Diagram of AI, ML, RL, and DL [based on Good-fellow et al.(2016)] .............................................................................. 11

Figure 2.5 : Machine Learning .................................................................................................. 15

Figure 2.6 : Deep Learning ........................................................................................................ 12

Figure 2.7 : (a) Human neuron network (b) NN neuron network [24] ..................... 13

Figure 2.8 : (a) Feed-forward networks (b) and Feed-back networks [28] ........ 14

Figure 2.9 : Comparison between the Classical Method and Deep Learning Method ........................................................................................................ 15

Figure 2.10: Convolutional Neural Network Layer [26] ........................................... 17

Figure 2.11: Deep Learning Sketch ( Classifier training on feature extracted ) [17] 17

Figure 2.12: Deep Learning Sketch ( NN of feature extractor and classifier ) [17] .. 18

Figure 2.13: Deep Learning Sketch ( Layers form ) [17] .............................................. 18

Figure 2.14: 2D filter [19] ........................................................................................................ 19

Figure 2.15: Convolution with 2D filter (1 stride) [19] ..................................... 19

Figure 2.16: Convolution with 2D filter (2 stride) [19] .............................................. 20

Figure 2.17: Convolution with different dilation factor, d [19] ......................... 20

Figure 2.18: Max Pooling [19] .............................................................................................. 21

Figure 2.19: Entire Convolution Neural Network Architecture .......................... 22

Figure 2.20: Feature Map [17] .............................................................................................. 22

Figure 2.21: Type of activation function ................................................................................ 22

Figure 2.22: Recent ConvNets proposed in ILSVRC .................................................. 23

Figure 2.23: Transfer Function Block Diagram [16] ................................................. 24
Figure 2.24: Model Fit ................................................................. 24

Figure 2.25: (a) Original iris image, (b) Segmented iris image; The boundary of iris is segmented after applying Circular Hough Transform method [4] .... 26

Figure 2.26: (a) Polar iris normalization, (b) Polar noise normalization [4] .......... 26

Figure 3.1 : Project workflow ...................................................................... 31

Figure 3.2 : Modification layer on Deep Network Designer Tool....................... 35

Figure 3.3 : Architecture of AlexNet model ................................................. 37

Figure 3.4 : (a) Standard Neural Net (b) After applying dropout; Illustration of Dropout Neural Net Model [34] ....................................................... 37

Figure 3.5 : Architecture of DenseNet model .................................................. 38

Figure 3.6 : Example of confusion matrix ....................................................... 41

Figure 4.1 : Mini Batch of iris images without Data Augmentation .............. 45

Figure 4.2 : Mini Batch of iris images with Data Augmentation...................... 45

Figure 4.3 : Feature layer of the 1st convolutional layer in AlexNet model ......... 45

Figure 4.4 : Training plot of AlexNet model on non-augmentation CASIA-Iris-Interval V1 training dataset ........................................................................ 49

Figure 4.5 : Training plot of AlexNet model on augmentation CASIA-Iris-Interval V1 training dataset ........................................................................ 50

Figure 4.6 : Script result of Bayesian Optimization on AlexNet model with CASIA-Iris-Interval V1 dataset ................................................................. 51

Figure 4.7 : Script result of Bayesian Optimization on DenseNet201 model with CASIA-Iris-Interval V1 dataset ................................................................. 52

Figure 4.8 : Iris images of user ‘104’ in (a) training, (b) validation, (c) testing dataset in CASIA-Iris-Interval V1 dataset ......................................................... 56

Figure C.1 Feature Map extracted from the 2nd Convolutional Layer

Figure C.2 Feature Map extracted from the 3rd Convolutional Layer

Figure C.3 Feature Map extracted from the 4th Convolutional Layer

Figure C.4 Feature Map extracted from the 5th Convolutional Layer
# LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>CASIA</td>
<td>Institute of Automation, Chinese Academy of Sciences</td>
</tr>
<tr>
<td>CER</td>
<td>Crossover Rate</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>ConvNet</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>CV</td>
<td>Computer Vision</td>
</tr>
<tr>
<td>DL</td>
<td>Deep Learning</td>
</tr>
<tr>
<td>EER</td>
<td>Equal Error Rate</td>
</tr>
<tr>
<td>FAR</td>
<td>False Accept Rate</td>
</tr>
<tr>
<td>FRR</td>
<td>False Reject Rate</td>
</tr>
<tr>
<td>FTE</td>
<td>Equal Error Rate</td>
</tr>
<tr>
<td>HD</td>
<td>Hamming Distance</td>
</tr>
<tr>
<td>ILSVRC</td>
<td>ImageNet Large Scale Visual Recognition Challenge</td>
</tr>
<tr>
<td>IRS</td>
<td>Iris Recognition System</td>
</tr>
<tr>
<td>LEI</td>
<td>Lion’s Eye Institute</td>
</tr>
<tr>
<td>LSTM</td>
<td>Long Short-Term Memory</td>
</tr>
<tr>
<td>MATLAB</td>
<td>Matrix Laboratory</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>MLP</td>
<td>Multilayer Perceptrons</td>
</tr>
<tr>
<td>MNIST</td>
<td>Modified National Institute of Standards and Technology</td>
</tr>
<tr>
<td>NIR</td>
<td>Near-infrared</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Network</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal identification number</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent Neural Network</td>
</tr>
<tr>
<td>SPDNN</td>
<td>Semi-parallel Deep Neural Network</td>
</tr>
<tr>
<td>SRCNN</td>
<td>Super-Resolution Convolution Neural Network</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>UNHCR</td>
<td>United Nations High Commissioner for Refugees</td>
</tr>
<tr>
<td>VDCNN</td>
<td>Very Deep Convolutional Neural Network</td>
</tr>
</tbody>
</table>
SISTEM PENGIKTIRAFAN IRS MENGGUNAKAN
TEKNIK PEMBELAJARAN MENDALAM

ABSTRAK

Pengesahan biometrik semakin canggih dengan kemajuan teknologi, terutamanya pemprosesan imej progresif dan keupayaan pengkomputeran. Pengiktirafan melalui organ iris manusia adalah salah satu biometrik yang popular kerana ia menjanjikan pulangan yang lebih janji dalam penglihatan mesin, kebolehpercayaan dan mudah berbanding dengan ciri-ciri lain. Pembelajaran mesin bergantung kepada input sample data yang masuk, dan lakukan klasifikasi pada sampel data. Akhirnya, ramalkan output berdasarkan kebarangkalian. Untuk teknik sistem pengenalan klasik yang sebelumnya, segmentasi iris dengan tepat adalah peringkat penting dalam menjamin ketepatan yang tinggi untuk Sistem Pengiktirafan Iris (IRS).

Oleh itu, keperluan dataset imej mesti diperolehi dalam keadaan khusus, jika tidak, ia mengakibatkan kegagalan dalam menjamin ketepatan yang tinggi untuk Sistem Pengiktirafan Iris (IRS). Oleh itu, keperluan dataset imej mesti diperolehi dalam keadaan khusus, jika tidak, ia mengakibatkan kegagalan dalam menjamin ketepatan yang tinggi untuk Sistem Pengiktirafan Iris (IRS).

IRIS RECOGNITION SYSTEM (IRS) USING DEEP LEARNING TECHNIQUE

ABSTRACT

Biometric authentication becomes sophisticated with the advance of technology, especially progressive image processing, and computational capabilities. Iris recognition through human iris organ is one of the popular biometrics as it is promising higher accurate return in machine vision, reliability and simpler as compared to other traits. Machine Learning depends on the input fit in, do classification on the sample data. Finally, predict the output based on the probability. For the previous classical recognition system technique, accurate iris segmentation is the crucial part to guarantee high accuracy for Iris Recognition System (IRS). Hence, the requirement of the image dataset has to acquire under specific conditions, else it might be failed in iris segmentation as well as the hyperparameters set or algorithms applied unsuitable for it. Manual hyperparameter tuning on Machine Learning model may take time and failure if not fully understand the algorithms and feature of datasets work with. In this thesis, the Transfer Learning method is proposed to capitalize pre-trained Convolutional Neural Network (ConvNet) model introduced in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) on the IRS. Systematic analysis has been conducted to design an optimal deep network architecture to achieve high efficiency in feature extraction. AlexNet and DenseNet201 pre-trained model that poses different ConvNet architecture and layer depth were chosen and trained Support Vector Machine (SVM) for testing model transferability. CASIA-Iris-Interval V1 dataset is then re-trained on AlexNet and DenseNet201 model one by one. Finally, evaluation of the IRS performances after applying Data Augmentation and Bayesian Optimization. All the results recorded along the algorithm development process showed the success of proposed methodologies in gaining a higher performance algorithm. Undergo proposed methodology flow, AlexNet achieved an overall accuracy of 97.22% meanwhile DenseNet201 achieved an overall accuracy of 98.81%. Transferability of a pre-trained model on new target task is improved and meanwhile, the high recognition rate of the algorithm on small-size CASIA-Iris-Interval V1 iris image dataset is achieved.
CHAPTER 1
INTRODUCTION

1.1 Research Background

“Biometric” is originated from Greek words, which are ‘bios’ and ‘metrikos’, can translate into meaning “life measurement” directly. It is a methodological study that involves the application of statistical analysis to biological data, while biometric data are the computational data that is generated during a biometric acquisition process [1]. Based on a distinguishing dataset, which is unique and specific possessed by an individual, it enables a person to be identified and authenticated automatically through machines. These biometric systems can be developed through multiple biometric trait choices, such as facial features, iris, voice, fingerprints, and vein. Additionally, gait, typing, and signature are other types of biometric characteristics that are studied on individual behavioral. Moreover, it is a quickly developing technology, able to enhance user convenience and extensively deployed for security systems, especially in access control, government-based and commercial applications [1, 2].

Biometric is the evolution of the conventional method, like Personal identification number (PIN)s and keys and even more secure for authentication [1, 2]. A key of difference, however, is that the former has to incorporate probabilities in the authentication process to find the best match. The traditional method only compares the PINs or passwords supplied by someone making a transition is either correct or incorrect. Conversely, numerous advantages made a biometric step forward as it lowers the risk of being exposed or forgotten. These kinds of human metric more likely to be difficult imitated or faked to access the system as compared with the traditional method, which enables anyone knows the password access the system easily takes no notice of the actual owner. It also can be explained that the conventional authentication technique depends on something you have and something you know whereas biometric authentication is something you are.
The most common tool we are using today is mobile biometrics and the cost of performing biometric authentication is relatively low due to the enhancement of sensor technology. Because of the popularity and widespread use of mobile devices, the acceptance of both users and industries have increased and become the lifestyle trends in this generation era [1]. It is expected that the biometric market is becoming increasingly widespread and will reach 30 billion dollars by 2020 [2].

Among those biometric systems, the iris is one popular of biometric options that able to give a high recognition rate and difficult to be tricked, even using a lens or dead iris [2]. Iris recognition is a process that undergoes non-invasiveness, non-contact and hygienic nature. It possesses lesser false acceptance and false rejection rate than any other personal recognition system due to iris is protected behind the eyelid, the likelihood of damage is minimal and it is not subject to the effect of aging. According to the United Nations High Commissioner for Refugees (UNHCR) project [3] in Malawi Biometrics in a Humanitarian Context, fingerprint scored lower than iris in overall preference since they often face problems like fingerprint data have to be re-captured and scanning hardware needed cleaning four times a day for the guarantee getting good image quality.

However, there are two recognition subproblems that can be confusing, which are identification and verification. In identification, the algorithm reports the most match data from the database of known individuals. In verification or called authentication, the application includes control access, and it is either accepting or accept the claim with the algorithm that claimed the identity of the person. The most common biometric system implementation on low-cost embedded hardware involves the following procedure. Basically, the general biometric system constitutes of five modules which are Sensor, Feature Extraction, Matcher, Database and Decision Making [1].

- **Sensor module/Image acquisition:**
  In the form of video, audio, and an image or some other signal, the individual’s raw biometric data are captured with a sensor device.
• **Feature extraction module**
An automated process, which distinguishing biometric metric to generate template code that can be done using Computer Vision, Machine learning and Pattern Recognition techniques.

• **Database module**
Store various templates of a user and a record enrolment of biometric information.

• **Matching module**
Generate a match score between the requested templates and stored templates, which is computed one to find the similarity between two biometric samples.

• **Decision-making module**
According to the given threshold value, the decision either accepted or rejected based on the match score.

### 1.2 Problem Statement

The most common IRS is involving in five steps, which are image acquisition, iris segmentation, iris normalization, feature encoding, and matching. Those steps play an important role in training a machine to recognize image patterns based on image datasets provided. For iris recognition, informative features are being extracted from the raw image and those features are classified based on criteria. Correct prediction on the label indicates successful of iris recognition. In previous studies, there are still many limitations and problems looking for solutions. These included poor imagining conditions, flexibility of the algorithms, and small size datasets.

Libor Masek [4] made use of a handcrafted feature extraction method by doing classification only on the isolated iris region from eye images. In his paper, 83% of the Institute of Automation, Chinese Academy of Sciences (CASIA) database images segmented correctly while the Lion’s Eye Institute (LEI) database has only 62% of the images managed to segment successfully due to poor imaging conditions. LEI database that imaging under natural light is containing specular reflections and difficult to segment iris region as compared to CASIA database that uses near infra-red light for
illumination in image acquisition. And only those successful iris segmented template undergo normalization to transform in fix dimensions for further comparison. With high precision through experimental stimulation, Hough Transform introduced by Libor Masek [4] is widely implemented on other research papers. The experimental result in [5] showed a better result of 91.56% success rate of iris segmentation for updated iris database, CASIA V4 as compared to CASIA V1. However, further study on the Hough Transform also exposes the disadvantages of poor robustness and slow speed in the iris recognition process [6]. Extracted feature template needs to be encoded on a large scale and used for further pattern matching calculation.

A similar problem goes when the implementation of the IRS. Flexibility in terms of accepting the appropriate input image is seeking for improvement. The ready solution is through the effort made on both hardware and software, purposely to eliminate the unwanted features and highlighting the important details of the image captured. Most of the time, the system will not get any input if the most desired condition is not met [7, 8]. The selection of the software parameters based on the image quality enables a possibility of high accuracy identification. In this case, most of the hand-crafted algorithm unable to simply be adapted to new conditions without fine tuning on software parameters.

However, recent advancements in deep learning and computer vision indicate that feature extraction using ConvNet effectively extracting discriminative visual features iris images and eliminating the laborious feature-engineering task. The limitation of classical IRS to process iris images that captured in unconstraint environments is breakthrough. Despite the robustness of Deep Learning (DL) in training machine to solve various tasks, limited application of DL methods to the problem of iris recognition due to fact that huge amount of training data required in DL [9]. Some recent papers show ConvNet can be applied in either iris segmentation, feature extraction section, or end-to-end iris recognition [10-12]. Another problem encountered by researchers is found that the size of the presented network model is big and not easily implementable on a low power handheld hardware [12]. Utilize domain knowledge in selecting training smaller network model and choosing hyperparameters to improve the accuracy of iris recognition rate under unconstraint conditions, nevertheless, it may alter the performance [13].
1.3 Research Objectives

There are three objectives in this project, which include:
1. To develop an algorithm based on the Deep Learning technique for the IRS.
2. To improve the flexibility of the algorithms using Transfer Learning technique.
3. To overcome small size iris images dataset.

1.4 Project Scheme

The scheme of this project is manage to eliminate iris segmentation phase that are common in conventional IRS, since it is failure that able to affect the recognition rate indirect. By using Transfer Learning, transferability of pre-trained ConvNet model for IRS can be investigated and a high performance ConvNet algorithm that automated feature extraction can be developed. Data Augmentation and Bayesian Optimization are proposed to overcome small sample iris dataset problem in DL, prevent the trained algorithm from overfitting. Meanwhile, this is a fully software development project using software platform on hardware configuration declared in Section 3.2.1.

1.5 Thesis Organization

The thesis consists of five chapters, namely Introduction, Literature Review, Methodology, Results and Discussion, and Conclusion. Chapter One introduces about biometric technology and popularity of IRS in market, examines the problem statements that will be focused throughout the projects. There are a few objectives stated as well to be achieved at the end of the project. The project scope emphasizes on the Transfer Learning technique in selecting appropriate network model and tuning hyperparameters to optimize performances. The next chapter narrates the literature review of the project that recap the information of the biometric technology field and describe the available theory and method that had been used for this project. Chapter Three clarifies the methodology of the project. This part includes the methods of methodological approach, methods of data collection and analyses. Chapter Four then presents the outcomes of the project and evaluate the methodological choices on the result. The final Chapter summarizes the thesis throughout the whole project. Challenges and future improvements also stated in this chapter.
CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

In this chapter, the research and fundamentals regarding this project will be portrayed and explained in detail. At first, using the anatomy of eyes and IRS are discussed. It followed by explaining the difference between Artificial Intelligence (AI) and Convolutional Neural Network (CNN) in computer vision, and how they can relate with each other to realize the image recognition system. The last section of this chapter also touches and mention some of the latest previous work on this field. Finally, the literature review is summarized.

2.2 Biometrics System

People always struggle on which biometric trait is the best and secure to identify them from the entire population. The result of the research in [1] is illustrated in Table 2.1, it exhibits that the iris biometric method achieves consistently high performance as compared to other biometric trait and able to resist to all related hesitation issues that might be faced. According to [1], the iris pattern matches 1 in 10 billion people.

The factors that examine and analysis of the various biometric traits are

- Universality (Availability in an individual),
- Uniqueness (Distinct characteristics),
- Collectability (Ease of acquisition),
- Permanence (Permanency and Stability),
- Performance (Effectiveness of implementation),
- Acceptability (Quality of being tolerated),
- Circumvention (Ease use of a substitute).
Table 2.1: Comparison of Various biometric techniques based on biometric traits [1]

<table>
<thead>
<tr>
<th>Identifier / Criteria</th>
<th>Universality</th>
<th>Uniqueness</th>
<th>Collectability</th>
<th>Performance</th>
<th>Acceptability</th>
<th>Circumvention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fingerprint</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Face</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Iris</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Hand Geometry</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Retina</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>DNA</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Gait</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Odor</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Palm print</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Ear</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Hand Vein</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Signature</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Keystroke</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Voice</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Thermograms</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
</tbody>
</table>

However, consideration in different aspects such as the design and function of the application when making the decision is significant to ensure that the choice of the selected biometric trait is best fit to the system and guarantee good performance since each biometric has its own merits and demerits [1].

2.2.1 Human Iris Anatomy

A human eye is an organ of great complexity, each iris under eyelid has a highly unique characterized pattern of ridges and folds, which indicate a good biometric in term of the feature is illustrated in Figure 2.1. The iris is a circular pigmented diaphragm that lies the anterior chamber to the lens, mainly help to control the amount of light that reaches the retina. Instead of a distinctive feature, the biometric data are more stable as a human iris pattern does not change over lifetime and formation of the pattern is random and unrelated to any genetic factor [4, 5]. Even two eyes of an individual and identical twins possess uncorrelated iris pattern, able to reduce the error and the probability of any two people having the same characteristic inherent will be minimal. The characteristics of the iris also cannot be changed by eye surgery or the wearing of glasses and contact lenses [1].
2.2.2 Iris Recognition

Table 2.2 showed that False Accept Rate (FAR) and False Reject Rate (FRR) of iris biometric remained almost stable at around one percent, followed by low Crossover Rate (CER) and Equal Error Rate (FTE), confirming that this biometric trait is both very inclusive and secure.

Table 2.2 : Performance Evaluation [1]

<table>
<thead>
<tr>
<th>Identifier Criteria</th>
<th>Fingerprint</th>
<th>Face</th>
<th>Iris</th>
<th>Hand Geometry</th>
<th>Retina</th>
<th>DNA</th>
<th>Signature</th>
<th>Voice</th>
<th>Keystroke</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAR</td>
<td>2%</td>
<td>1%</td>
<td>0.94%</td>
<td>2%</td>
<td>0.91%</td>
<td>-</td>
<td>-</td>
<td>2%</td>
<td>7%</td>
</tr>
<tr>
<td>FRR</td>
<td>2%</td>
<td>10%</td>
<td>0.99%</td>
<td>2%</td>
<td>0.04%</td>
<td>-</td>
<td>-</td>
<td>10%</td>
<td>0.10%</td>
</tr>
<tr>
<td>CER</td>
<td>2%</td>
<td>-</td>
<td>0.01%</td>
<td>1%</td>
<td>0.80%</td>
<td>-</td>
<td>-</td>
<td>6%</td>
<td>1.80%</td>
</tr>
<tr>
<td>FTE</td>
<td>1%</td>
<td>-</td>
<td>0.50%</td>
<td>-</td>
<td>0.80%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

To get a high recognition rate, ideal image acquisition conditions are assumed in the most recognition systems. Most of the open source iris database in previous time are captured using Near-infrared (NIR) camera in close distance, along with the look and stare constraints condition. With the NIR camera, the pigmentation of the iris image is revealed and clear texture to obtain. It saves a bundle of engineers’ time in
filtering the noise. However, recent advances in iris recognition have proposed
different methods to process iris images captured in bad environments. These
unconstraint environments include a visible wavelength (VW) light source, on the
move and over the distance from the capturing device. Thus, the size of the iris will
vary according to the image capturing condition and environment. This may cause
pupil dilation, which affects the size of the iris portion and failure in iris segmentation
[15].

Table 2.3 and Table 2.4 summarizes the most used iris databases in research
and describes their imaging framework along with all characteristics of iris images in
each database [16, 17]. There are two categories of databases available. In the first
category, the iris dataset captured under NIR light source, compose as a noise-free iris
database (CASIA, BATH, MMU, ICE, and WVU). Continue with the second category,
the iris databases are noisy, since they have been captured under visible wavelength
light source (UPOL, UBIRISv.1, and UBIRISv.2) [16]. Table 2.3 below shows the
difference as the iris image captured in NIR light and visible light. A visible
wavelength lighting source causes reflection noise on the captured image, while NIR
lights do not, is demonstrated in Figure 2.2.

Table 2.3: Various iris dataset [16]

<table>
<thead>
<tr>
<th>Database Name</th>
<th>Database Size</th>
<th>Light Wave Length</th>
<th>Varying Distance</th>
<th>Camera</th>
<th>Sample Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASIA v1</td>
<td>756</td>
<td>NIR</td>
<td>No</td>
<td>CASIA camera</td>
<td></td>
</tr>
<tr>
<td>CASIA v2</td>
<td>2,355</td>
<td>NIR</td>
<td>No</td>
<td>CASIA camera</td>
<td></td>
</tr>
<tr>
<td>CASIA v3</td>
<td>22,051</td>
<td>NIR</td>
<td>No</td>
<td>OKI iris-pass h</td>
<td></td>
</tr>
<tr>
<td>CASIA v4</td>
<td>2,976</td>
<td>NIR</td>
<td>Yes</td>
<td>IKEMB-100 dual camera</td>
<td></td>
</tr>
<tr>
<td>Bath</td>
<td>16,000</td>
<td>NIR</td>
<td>No</td>
<td>ISG LW 1.3 S 1394</td>
<td></td>
</tr>
<tr>
<td>MMU 1</td>
<td>450</td>
<td>NIR</td>
<td>No</td>
<td>LG EOU 2200</td>
<td></td>
</tr>
<tr>
<td>MMU 2</td>
<td>995</td>
<td>NIR</td>
<td>No</td>
<td>Panasonic BM ET 100 US</td>
<td></td>
</tr>
<tr>
<td>ICE 1</td>
<td>2,900</td>
<td>NIR</td>
<td>No</td>
<td>LG EOU 2200</td>
<td></td>
</tr>
<tr>
<td>ICE 2</td>
<td>75,000</td>
<td>NIR</td>
<td>No</td>
<td>LG EOU 2200</td>
<td></td>
</tr>
<tr>
<td>WVU</td>
<td>3099</td>
<td>NIR</td>
<td>No</td>
<td>OKI iris-pass h</td>
<td></td>
</tr>
<tr>
<td>UPOL</td>
<td>384</td>
<td>Visible</td>
<td>No</td>
<td>Sony DSC 950P 3CCD with TOPCON TRC501A</td>
<td></td>
</tr>
<tr>
<td>UBIRIS v1</td>
<td>1877</td>
<td>Visible</td>
<td>No</td>
<td>NIKON E5700</td>
<td></td>
</tr>
<tr>
<td>UBIRIS v2</td>
<td>11,357</td>
<td>Visible</td>
<td>Yes</td>
<td>Canon EOS 5D</td>
<td></td>
</tr>
<tr>
<td>FRGC</td>
<td>50,000</td>
<td>Visible</td>
<td>Yes</td>
<td>Minolta Vivid 900/910</td>
<td></td>
</tr>
</tbody>
</table>
Table 2.4: Heterogeneity of iris dataset [16]

<table>
<thead>
<tr>
<th>Database</th>
<th>Noise</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
<th>Red Out</th>
<th>Green Out</th>
<th>Blue Out</th>
<th>Red End</th>
<th>Green End</th>
<th>Blue End</th>
<th>Out of Iris</th>
<th>68.8%</th>
<th>68.8%</th>
<th>68.8%</th>
<th>68.8%</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASIA 1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CASIA 2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CASIA 3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CASIA 4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>BATH</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MMU 1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MMU 2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>UCE 1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>UCE 2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>WVU</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>UPOL</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>UBIRIS-v1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>UBIRIS-v2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TRGC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

2.3 Computer Vision (CV)

The capabilities of human vision are concerned and challenged as we know that human may not have the most powerful visual system among all the species although for the act of the logical thinking and deciding, human visual system and its associated cognitive decision-making processes are one of the fastest in the planet. When facing this kind of limitation, scientists invent and develop tools such as telescopes, microscopes, and magnifiers to enable human being to see further and beyond the eyes [18]. Along with math researches, visual sensor and computing technology, CV techniques are being utilized today in a wide variety of real-world
applications, especially intelligent human-computer interaction, robotics, and multimedia [19]. Some examples of the hit case regarding this area are autonomous cars [20], syndrome analysis assistants on medical [21], and robots in manufacturing sites [22]. In a sense, a biometric system is a kind of CV that enables the camera to imitate eyes and train the computer how to see and understand the world.

2.4 Artificial Intelligence (AI)

The tale of human invention and technological developing is a narrative of the human species endlessly outgrowing its own capabilities and thence endlessly expanding its horizons and forward to the future[18]. Science evolution builds machines that able to think like a human and produce a number of powerful tools that are automatically solving problems that would normally require human intelligence. In this case, the human neural system is imitated and taking it as inspiration, a self-conscious machine can be self-learning and training by using large scale datasets. AI that incorporating human intelligence to machines and machines able to complete tasks based on a set of formulated rules. The machine-trained with high accuracy and precision specification enables them to take over numerous tasks that human being performs routinely and thus, drastically changing the ways we work and live [23]. Both Machine Learning (ML) and Deep Learning (DL) are the method to attain AI. From the picture below, we saw that DL is a subset of ML, which is also a subset of AI.

Figure 2.4 Venn Diagram of AI, ML, RL, and DL
[based on Good-fellow et al.(2016)]
2.4.1 Machine Learning (ML)

Figure 2.5 illustrate ML learning can be classified into three types, which are supervised learning, unsupervised learning, and reinforcement learning.

![Machine Learning Diagram]

Figure 2.5: Machine Learning

Supervised learning close to the process in which a human learns things. When the current output is wrong, the algorithm is modified and retrained until getting the correct output for given input respectively. Unsupervised learning is generally used for investigating the characteristics of the data and pre-processing the data. Hence, the training data of the unsupervised learning contains only inputs without correct outputs. And finally, Reinforcement learning employs sets of input, some output, and rewards as training data. It is commonly applied when optimal interaction is required, such as control and game plays.

For output, supervised learning can be divided into classification and regression, depending on the usage of the model. Classification determines which group or category the input data belongs to. Instead of output in classes, regression predicts values as the correct output in the training data. In contrast, unsupervised learning is clustering and categorized the same characteristics in the cluster without labelling.
2.4.2 Deep learning (DL)

In DL, the process of determining the model (NN) is called the learning rule. Figure 2.6 displays workflow of NN.

![Figure 2.6: Deep Learning](image)

NN like the gigantic network of neurons in the human brain, associates and forms specific information to the input provided. To simplify, node (circle in the Figure 2.7(b)) in NN is an analogy of neuron in brain and connection weight (Arrows pointed to circle in Figure 2.7(b)) in NN is an analogy of connection of neurons in the brain respectively [17]. The neuron just transmits signals from one neuron to the other and has no storage capability, which is shown in Figure 2.7(a). For weight, \( w_{ji} \) and bias, \( b_i \) of the corresponding signals, we can calculate the output, \( f(x) \) based on inputs, \( v_i \) given. The equation for the output, \( f(x) \) is listed in Equation 2.1.

![Figure 2.7](image)

(a) Human neuron network  (b) NN neuron network 2 [24]
\[ f(x) = w_{ji} v_i + b_i \]  \hspace{1cm} (2.1)

\( b_i \) is the bias, extra neuron to allows shift the activation function and give freedom to model and fit best to the given data activation function is one of the key factors that affect NN’s decision making and performance and choice of the suitable one is crucial to boost the NN performance [24, 25]. Types of activation function are Sigmoid, ReLU, Leaky ReLU, TanH and others, acting like a mathematical transfer function that converts the input to an output [26].

There are numerous of NN architectures being published and it can be either single layer or multi-layer neural networks with more than one hidden layer of the neurons [17]. Feed-forward networks and Feed-back networks in Figure 2.8 are the genetic categories under NN, based on the way of information is manipulated in the network. The examples for former architecture are Multilayer Perceptrons (MLP) and CNN, which there are no loops or cycles in the connection. For the latter’s instances, there are Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) which have loops in connections. The loops give network exhibit memorization ability and hence, the internally stored information is incorporated in processing and influence the final result [19].

![Feed-forward networks](image)

![Feed-back networks](image)

Figure 2.8 Feed-forward networks (a) and Feed-back networks (b) [28]
2.4.3 Difference between ML and DL

One of the earliest descriptions of was given by Samuel (1959). The author claims that ML empowering computer systems with the ability to “learn” by themselves using the provided dataset and make accurate predictions. This learning method takes the feature representation of instances for granted and all this depends on some prior knowledge on the data when doing feature selection [23]. While DL directly processes raw data (e.g. RGB images) and automatically identifies its pattern features. It typically used for complex tasks, such as image recognition, classification, and identification [26]. Compared with handcrafted features, learned representations require less human intervention and provide much better performance [17]. Meanwhile, DL offers simplicity by subtracting the manual feature extraction work, scalability to huge datasets, and flexibility of model on another related task. This can be illustrated clearly from Figure 2.9 that published in a journal that did research on DL to diagnose faultiness of equipment [27].

In [9], DL requires only one step to do the end-to-end mapping from the original signal (input) to the fault category (output) as compared to classical fault diagnosis. It combines both feature extraction and classification procedures that needed to develop a classical fault diagnosis algorithm by passing through a black box. However, the weight of the hidden layer can be transformed into features from the deep model which has been trained in DL.

![Figure 2.9: Comparison between the Classical Method and Deep Learning Method [27]](image-url)
DL is the evolution of ML technique [19] and the pros of it is Neural Network (NN) applied are more flexible and it is a good model and reliable in an approach of the task involving a large number of nonlinear input data and feature; for example images [26]. Instead of specifying how the image of every single category looks like one by one in code, DL require many examples of each image class with label. Then, the labelled images are fed into the computer and let the network learn about the visual appearance by self.

However, some claim that DL does not provide statistically better accuracy as compared to classical machine learning in their researches. This probability due to the network was not properly trained and features of datasets may not be too complex to predict, causing the problem of over-fitting and consequently, the dataset is overkilled [17]. NN are black boxes and we cannot know much each independent variable is influencing the dependent variable [26].

2.5 Convolution Neural Network (ConvNets)

The standard reference for ConvNets is from 1999 by LeCun et al., “Object Recognition with Gradient-Based Learning”. However, the first ConvNets idea that revealed, called neocognitron. It is proposed by Kunihiko Fukushima in the 1980s, which is a hierarchical multi-layered artificial neural capable of the robust visual pattern [8]. When coming to the world of DL, ConvNets are one of the most popular and high usage categories of NN, especially for high dimensional data, a field or matrix (e.g videos or images) [19]. ConvNets designed specifically for image recognition and classification problems [26]. Although there are similarities between ConvNets and standard NNs, there still some obvious differences among them. Both of them have multiple layers of NNs that extract information from input images and determine the class they fall into. For ConvNets, however, implementation of two-dimensions (or higher) matrix and arrange them in layers of neurons in three-dimensions [26], and it is being demonstrated in Figure 2.10. This is useful for image input since RGB is a three dimensional matrix and hence, ConvNets able to recognize visual patterns directly from the pixel images with minimal pre-processing whereas other regular neural nets unable scale well.
ConvNet is a sequence of layers of neural nets, wherein each hidden layer transforms one volume of activations to another through a differentiable function. The three type layers that build up ConvNet are the convolution layer, pooling layer and fully connected layer. As compared to the previous regular NN layer, the convolution layer does not employ connection weights and a weighted sum.

Figure 2.10 Convolutional Neural Network Layer [26]

The easiest exercises for beginners in training image classification using Transfer Learning method and ConvNets that available online are recognizing whether the animal a dog or a cat in the picture and Modified National Institute of Standards and Technology (MNIST) letter recognition. The features of an image can be extracted from a selected feature layer, and then, the features of images are used to train the classifier, is shown in Figure 2.11. After that, classifier is used to predict the outputs for other testing images.

Figure 2.11 Deep Learning Sketch (Classifier training on feature extracted) [17]
It also can be a ConvNet that trained from scratch look like the sketch in Figure 2.12. Architecture on different pre-trained ConvNet models will be different in terms of layer depth, model size, and hyperparameters. Adjusting on weights and biases parameters will also impact the output of the NN. Thus, in order to get rid of handcrafted features, ConvNet is introduced and the manual feature extraction design procedure is automated as it is incorporated in the NN. And the view in a deeper vision, the ConvNet consists of a NN that in-charge of feature extraction and followed by another NN that do classification based on the output of feature extraction network. Figure 2.12 shows the block diagram of Deep Learning and ConvNet acts as both feature extraction and classifier.

Figure 2.12 Deep Learning Sketch (NN of feature extractor and classifier) [17]

In the feature extraction network shown in Figure 2.13, it consists of piles of the convolutional layer and pooling layer, depends on the depth of the training network.

Figure 2.13 Deep Learning Sketch (Layers form) [17]
2.5.1 Convolution Layer / Filtering

The convolutional layer, which is one side of the feature extraction neural network, does the heavy math operations. To process an image for the feature extraction, the original image is convoluted with a convolution filter and the unique feature of it is transform becoming a feature map or activation map. Besides that, convolution operation also involves padding \((p)\), stride \((s)\), dilation factor \((d)\) instead of pre-defined filter size \((f)\). ConvNet’s convolution process like reading the book by a magnifying glass. It scans the book chunk by chunk, using filter along with the image from left to right, and then top to bottom, until it spans the entire image. In the convolution layer, the size of an image is decreased without losing the relationship between pixels due to they are only related to the adjacent and close pixels.

Figure 2.14 shows filter is a grid of discrete numbers and Figure 2.15 illustrates it is using a 2D image filter to convolute the input image.

\[
\begin{array}{cc}
2 & 0 \\
-1 & 3 \\
\end{array}
\]

Figure 2.14: 2D filter [19]

![Figure 2.15: Convolution with 2D filter (one stride) [19]](image)

For a bigger size input image, the output feature map can be reduced by setting a zero padding of one and a stride of two instead of one. By a high stride length, the convoluted process can be done quickly. The filter takes a step of two along with the horizontal or vertical position, as shown in Figure 2.16. While Figure 2.17 shows the effect of changing the dilation factor.
Mathematically, the output feature dimensions can be calculated using Equation 2.2 and Equation 2.3

\[
h' = \left\lfloor \frac{h - f + s}{s} \right\rfloor, \quad w' = \left\lfloor \frac{w - f + s}{s} \right\rfloor, \quad \ldots. (2.2)
\]

\[
h' = \frac{h - f - (d - 1)(f - 1) + s + 2p}{s}, \quad w' = \frac{w - f - (d - 1)(f - 1) + s + 2p}{s}, \quad \ldots. (2.3)
\]
Hyperparameters of the convolution layer, such as stride and padding, need to be set by the user based on cross-validation or experience prior to the filter learning. These hyperparameters can also be clarified as the design specification of a network architecture based on a given application.

A pooling layer acts like a math function that operates on blocks of the input feature map and combines the feature activations, reduces the size of the image. The math function can be maximum, average, or sum value inside these pixels. Similar to the previous convolution layer, the size of the pooled region and the stride need to be specified. Figure 2.18 shows the operation of the max-pooling layer when the size of the pooling region is $2 \times 2$ and the stride is one. The maximum activation is chosen from the selected block of values during max function operation and this is continued by sliding the window across the input feature maps with a step size defined by the stride until the end.

![Figure 2.18 Max Pooling](19)

Based on the entire CNN architecture illustrated in Figure 2.19, full-connected layers are usually placed toward the end of the architecture and identical to the weight layer, purposely to classify images. It is a network where each parameter is linked to one another to determine the true relation and effect of each parameter on the labels. The architecture of ConvNet on special operations will generate feature map and shown in Figure 2.20.
2.5.2 Nonlinearity

A nonlinear function after the weight layers in a ConvNet (e.g., convolutional and fully connected layers) is highly important since it allows a neural network to learn nonlinear mappings. A nonlinear function decides whether a neuron in the hidden layer will be fired or given all of its inputs and improve the performance of network indirect. The most common activation functions that are used in deep neural networks are illustrated in Figure 2.21.
2.5.3 ImageNet Large Scale Visual Recognition Challenge, ILSVRC

ImageNet project is a large visual database, designed for object recognition software research usage and has been running an annual competition in visual recognition, called ImageNet Large Scale Visual Recognition Challenge, ILSVRC from 2010 onwards. Some of the popular architecture of ConvNets that won in ILSVRC competition are AlexNet (ILSVRC 2012 winner), VGG (ILSVRC 2014 runner-up), GoogLeNet and Inception (ILSVRC 2014 winner), ResNet (ILSVRC 2015 winner), and DenseNet (2016). The evolution of ConvNets architecture is illustrated in Figure 2.22. The depth of the ‘winning’ ConvNets has gradually added from 8 layers in 2012 to 152 layers in 2015, while the recognition error rate has progressively decreased from 16.45% in 2012 to 3.57% in 2015 [9].

![Figure 2.22: Recent ConvNets proposed in ILSVRC](image)

2.6 Transfer Learning

The central concept of Transfer Learning is to employ the existing successful pre-trained DNN model to ‘transfer’ their learning to a more simplified problem. During this transfer training, some hidden layers can be frozen in order to decrease computation time. Traditional ML is isolated and single task learning, while Transfer Learning makes learning of a new task reliefs on the previous learned tasks. Hence, Transfer Learning process can be faster, more accurate and need less training data. Figure 2.23 illustrates the difference between traditional ML and Transfer Learning in the block diagram.
The pre-trained transfer model is simply chopped off at the last one or two layers since the shallower layers of a ConvNet are essentially learning how the features of the image look like. Only the last one or two layers of a ConvNet are performing the most complex tasks of summarizing and categorizing the vectored image data into their specific class that they are supposed in. The act of layer freezing also can avoid the problem like overfitting. Overfitting is a modelling error that occurs when the function is too closely fit to a limited set of data and not generalized. Figure 2.24 illustrates the model fit under three different conditions, which are under-fitting, appropriate-fitting and over-fitting.

In these steps, some hyperparameters may need to tune based on the result of training. However, the hyperparameters are the most varied and complex. It can the number of nodes, the number, types, and connectivity of the layers, selection of activation function, learning rate, momentum, number of epochs, and batch size. All these settings can lead to different results.
2.7 Previously Proposed Classical Machine Learning Method for Iris Recognition

Almost all Classical ML in IRS involves these elementary steps, including image acquisition, pre-processing, features encoding, and matching for user authentication or identification [8]. Iris segmentation is the most crucial part in the IRS since the feature of the iris has to be extracted correctly to enable achieve high accuracy in the matching stage.

- **Image acquisition:**
  Image acquisition is significant in the recognition system since poor imaging conditions negatively affect the recognition accuracy of the system. The majority of commercial iris acquisition cameras designed to capture iris images in the NIR band, which able to reveal structural patterns even for strongly pigmented irises, in contrast to images acquired under visible light.

- **Pre-processing**
  Pre-processing steps in IRS significant to generate a quality iris code template. The segmentation and normalization of the iris needed to detect the exact location and contour of an iris in an image.

  **Iris segmentation**
  Iris segmentation is a crucial part of classical iris recognition method. Only iris templates success segmented from eye images can proceed to iris normalization. Figure 2.25 demonstrates eye images before and after the iris segmentation process.

  1. **Candy edge detector** was used to remove any artifacts as it is capable of detecting a strong edge pupil and iris boundary.
  2. **Circular Hough transform (Wildes)** is adapted to the iris image for generating an edge map for the iris region to mark the inner boundary.
  3. **Daugman’s Integro differential operator (Daugman)** is used to locate outer boundary followed by eyelid and eyelashes detection using Hough transform three line model and adaptive threshold
technique respectively [15]. At the end of this stage, quality metrics always applied to the pre-processed image for image quality control.

Figure 2.25 Original iris image (a), and Segmented iris image (b),
The boundary of iris is segmented after applying
Circular HoughTransform method [4]

**Iris Normalization**
After testing for a range of radii with the image in datasets, the maximum value of the radius and centre coordinates for the boundary is stored. Then, Daugman’s rubber sheet model is used to transform the segmented iris to normalized polar coordinates [29]. Figure 2.26 displays both the normalized iris and noise templates.

Figure 2.26 (a) Polar iris normalization (b)Polar noise normalization [4]

The segmented iris region is normalized using Daugman’s rubber sheet to form fixed dimension template.

- **Feature Encoding**
  1. A binary feature vector is generated after applying 2D Gabor Filter (Daugman) and this method is suitable for texture representation and discrimination
  2. By applying the Gabor Wavelet Transform (Daugman), the data in the iris region resolutions is decomposed into components that appear at
different resolutions. The output of applying the wavelets is then encoded in order to provide a compact and discriminating representation of the iris pattern.

- **Matching for User authentication or identification**
  Typically, Hamming Distance (HD) as a measure of the dissimilarity between a pair of iris-codes. Fast comparisons between iris-codes are implemented by the simple Boolean exclusive-OR operator, XOR is applied to a pair of iriscode representations. A value of zero will represent perfect matching, however, a value near 0.5 will represent two independent irises [15].

### 2.8 Previously Proposed ConvNets for Iris Recognition

DL provides a flexible and scalable machine learning method that requires no more constraint on the feature extraction while training the algorithm, especially in iris recognition algorithm development. However, certain research results show that DL can underperform as compared to classical ML techniques sometimes. This result is difficult to predict and explain in mathematically as DL is a black box concept. Thus, this ignites the curiosity and spirit of researchers in investigating the root cause of failure and discovering for a new NN algorithm, tend to achieve high accuracy and precise rate for the specific systems. As DL becomes the technology trend, there are many research journals had been proposed and published, aim to study DL from a different perspective.

Instead of using a conventional algorithm for iris segmentation, researchers proposed a new algorithm and test on their effectiveness on both NIR and visible light database. Semi-parallel Deep Neural Network (SPDNN) [12] and IrisDenseNet [10] are developed and both of them perform well in the iris segmentation region, even with a low-resolution image.
Refer to paper in [9], investigation the performance of state-of-the-art ConvNets pre-trained on the ImageNet dataset for the iris recognition task given the superior performance of those ConvNets on the ILSVRC and a large number of other computer vision tasks. The ConvNets in use are AlexNet (ILSVRC 2012 winner), VGG (ILSVRC 2014 runner-up), GoogLeNet and Inception (ILSVRC 2014 winner), ResNet (ILSVRC 2015 winner), and DenseNet (2016). The normalized iris image is fed into those ConvNets feature extraction module and then do classification using Support Vector Machine (SVM) method. The iris datasets used here are ND-CrossSensor-2013 and CASIA-Iris-Thousand. The result of this research claims that off-the-shell pre-trained ConvNet features, even though originally trained for the problem of object recognition, can be appropriated to the iris recognition task.

In [30], the research does a study on the impact of the pre-processing input image to the recognition rate. VGG and ResNet-50 network were implemented and test on currently one of the most challenging NICE.II competition database, presenting noise and image resolution problems on the uncontrollable environment. This research aims to analyse the impact of not pre-processing of iris images. At first, its analysis on Data Augmentation technique on non-segmented iris images by rotating the original images at six specific angles and thus, expand the number of original training samples from 1,000 images for 171 classes to 7,000 images. The model trained with Data Augmentation reported a smaller standard deviation and Equal Error Rate (EER) values improved with a statistical difference. After that, it is tested again for the impact of the iris segmentation with Data Augmentation. ResNet-50 with non-segmented image and 4:2 aspect ratio of normalized with standard rubber sheet model leading the others with EER 17.11%. The final part in the research was the analysis effect of iris delineating on the dataset in both situations, non-segmented and non-normalized. The best result proved in this research is using ResNet-50 model with non-normalized, non-segmented and delineated iris images, achieved the lowest EER among others, which is 13.98%.
Some of the paper proposed and do research to request the answer for an interesting question, like “Is photo-realism important to iris recognition”. In [31, 32], Transfer Learning is implemented and these research aimed to explore if the nature of the image as well as if the pattern from the iris can affect the ConvNets Transfer Learning. Both of these use CASIA-Iris-Interval V3 as a target Image database and downscaled to the desired factor. The downscaled image then passes through Super-Resolution Convolution Neural Network (SRCNN) or Very Deep Convolutional Neural Network (VDCNN), the trained deep learning ConvNets, to reconstruct the final super-resolved image database. The result of these researches concluded that the texture database is more suitable to train the SRCNN for iris recognition. They trained the SCRNN by Transfer Learning, using pre-trained models that are learned on huge datasets. Hence, it is important to know which algorithm is more suitable to perform Transfer Learning and to be able to learn correct patterns that will be useful in the target database.

2.9 Summary

The studies of iris biometrics in authentication system have been outlined in this chapter. There are many reviews related computer aided imaging system regarding on the IRS. These include the performances of iris biometric in the authentication system, available public iris images datasets, and image processing techniques such as Transfer Learning that enable machine learn to recognize iris patterns. In artificial intelligence realm, both ML and DL techniques are being discussed and previous related work based on them also been listed in final part of this chapter.
CHAPTER 3

METHODOLOGY

3.1 Introduction

This project consists of software development. The outline of this chapter will be the software development methods and for the following sections, each method will be explained in detail. Brief on project flow is given in Section 3.2 and the sub sections include prerequisite requirement and preliminary setup to carry out the project in Section 3.2.1. Image pre-processing that involved Data Augmentation in the case to solve the small size training dataset problem in Section 3.2.2, Transfer Learning and the architecture of selected pre-trained ConvNets model that used for IRS algorithm development in Section 3.2.3 and Section 3.2.4 respectively. Section 3.2.5 declares Bayesian Optimization technique that is used to optimize the hyperparameters. After that, metrics for performance analysis is declared in Section 3.2.6.

3.2 Project Workflow

This is a fully software development project on training IRS algorithms. The proposed algorithm is implemented in fixed hardware and software platform mentioned in the next subsection. The algorithm is developed by training with the target dataset using deep learning technique and the proposed development flow shown in the flow chart in Figure 3.1. The training procedure has to start with load datasets into the workspace on Matrix Laboratory (MATLAB). Then, image pre-processing include image resize, image allocation on dataset image into three portions, and image augmentation. The raw training samples data are set to be augmented randomly. The proposed algorithm is revised and trained again with proposed techniques such as Bayesian Optimization and Layer Freezing until satisfied result obtained. The results are tabulated and performances are evaluated.
3.2.1 Prerequisite requirement and preliminary setup

3.2.1.1 Hardware

Intel Core i5 2.2GHz and 8GB installed RAM employing Window 8.1.

3.2.1.2 Software

MATLAB R2019a riches with ML libraries and experts in processing matrix calculation was used to develop the image processing algorithm in this project. In MATLAB, Deep Learning Toolbox with the pre-trained model, AlexNet and DenseNet201 support package were pre-installed. DL training framework is constructed and simulated on certain software platform until the proper design was achieved. The MATLAB code for algorithm development is in Appendix A.
3.2.1.3 Iris datasets

CASIA-Iris-Internal V1, an iris database that contains the complete eye part, was used as a target dataset along with this iris recognition project for algorithm simulation and development. It is a public iris dataset collected by the Institute of Automation at the Chinese Academy of Sciences. It is one of the oldest publicly available systems for evaluation of the iris biometric modality. This version contains 756 iris images from 108 human subjects. The images in CASIA-Iris-Interval V1 have been captured using a homemade iris camera, with eight circularly fitted NIR 850mn illuminators [15]. All the images are stored in the format of BPM at 280×320 resolutions.

3.2.2 Image Pre-processing

3.2.2.1 Image Resize

Training a network and making predictions on new data require images that match the input size of the network. In this section, all the images in the CASIA-Iris-Interval-V1 iris database were resized to fixed dimensions capable with the architecture of the model selected. For AlexNet and DenseNet201, the input images were resized to 227×227×3 and 224×224×3 respectively. Instead of image cropping to adjust image size, resizing method can prevent loss of informative features for further processing.

3.2.2.2 Image Dataset Allocation

In all algorithm training stage, all the datasets are equal distribute into three main sets with a ratio of 0.42:0.29:0.29, which are further labelled as training, validation and testing accordingly. Both training dataset and validation dataset are used during neural network training. The training dataset adjusts the weight of NN while the validation dataset prevents the NN from overfitting. Additionally, the testing dataset is only used to confirm the actual predictive power of the well-trained network. For CASIA-Iris-Interval V1, 108 users have a total of seven original iris image each and after allocation, it was divided into three sessions, contain training session (three images), validation session (two images) and testing session (two images). Hence, a total of 756 images is sorted to 324 training images data, followed by 216 validation images data and 216 testing images data.

32
3.2.2.3 Image Augmentation

Augmentation on training images helps prevent the trained network from overfitting to specific features by changing the apparent features in an image. It is one of the solutions for small size dataset in DL training by increasing the variety of training samples. Data Augmentation strategies can be in terms of flipping, scaling, cropping, shearing, rotation and translation. These operations on images can be performed separately or combined together [19]. For instance, rotation after cropping.

However, some of the augmentation methods were eliminated due to it is unconducive on increasing the iris recognition rate. For instance, random cropping may cause loss of some informative feature messages from the original images and eventually, cause the dropping of recognition accuracy rate. The practicable of image augmentation method in breaking the limitations on dataset size in IRS can be justified in [33].

In each iteration, training data in a mini batch is transformed based on a random combination of transformation strategies and the transformed images are not stored in memory. Hence, the actual number of training samples does not change via augmentation. The training data were perturbed in every epoch and the network trained using a slightly different training dataset copies in every epoch. Parameters for each strategy are tabulated in Table 3.1.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Definition</th>
<th>Range/ Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale</td>
<td>Range of uniform (isotropic) scaling</td>
<td>0.5 - 2</td>
</tr>
<tr>
<td>Rotation</td>
<td>Range of rotation, in degree</td>
<td>±10</td>
</tr>
<tr>
<td>X-axis translation</td>
<td>Range of horizontal translation, in pixels.</td>
<td>±30</td>
</tr>
<tr>
<td>Y-axis translation</td>
<td>Range of vertical translation, in pixels.</td>
<td>±30</td>
</tr>
</tbody>
</table>
3.2.3 Transfer Learning technique

Instead of training from scratch, Transfer Learning is a popular method that fine-tunes a pretrained network with the target dataset and allows to build an accurate model in a timesaving way. The steps of applying Transfer Learning are listed as following:

Step 1: Choose on pre-trained ConvNet model and Iris database.
Step 2: Optimize hyperparameters using Bayesian Optimization.
Step 3: Select the number of layers freeze. (optional)
Step 4: Retrain the ConvNet model with optimizing hyperparameters.
Step 5: Collect results in the form of data and figures.

Since the target dataset is a task different from the base training dataset, it needs to find a balance on the number of layers to train and freeze respectively. The model can be overfitting if trained too deep or the model unable to learn the features of the target dataset if it remains at the shallow end. In this case, this project was trained by freezing different neural layers without updating the weight on frozen layers and the last fully connected layer was replaced with the layer matching the number of classes in the target dataset, which is 108 for CASIA-Iris-Interval V1. The proposed algorithm also eliminates the step of iris segmentation and avoids error in the iris segmentation phase that may reduce the overall iris recognition rate. The performance of Transfer Learning by freezing different level of layers is recorded and further evaluated in Result and Discussion section.

In MATLAB software environment, a network can be modified via either programming code or through the Deep Network Designer app. Figure 3.2 shows the pre-trained model is imported into the Deep Network Designer app and the output size of fully connected layer is modified to an identical number match to the target database, which is 108 for CASIA-Iris-Interval V1. After done adjusting, the modified model is ready to be exported to the base workspace by clicking the ‘Export’ icon on the toolbar list.
Along the algorithm development process, the network training was stopped when one of the following criteria was met:

a) The validation accuracy remains unchanged, or
b) The training accuracy reaches a plateau and no longer improving, or
c) Overfitting occurs

Since the selection of the suitable pre-trained model is a major part in applying Deep Learning technique, the transferability of the pre-trained model on CASIA-Iris-Interval V1 dataset was pre-tested using trained SVM classifier. The MATLAB code for training on SVM classifier is displayed in Appendix B. The verification process was begun with identity which desire feature layer want to extract. The extracted feature of the training dataset were extracted using activation on desire feature layer on either AlexNet or DenseNet201 pre-trained model. The extracted features of training datasets with 324 samples were then used to train a multiclass SVM classifier. The test features were then extracted from the test sample sets, passing to the classifier and measure the accuracy of the trained classifier for its transferability. The classification accuracy results of SVM classifier on varies feature layer and model were verified and tabulated.
3.2.4 Pre-trained ConvNet Model

3.2.4.1 AlexNet

AlexNet, an 8 weight layer deep pre-trained ConvNet model that is trained on more than a million images from the ImageNet database [34], has learned rich feature representations for a wide range of images and able to classify images up to 1000 object categories. For example, objects such as keyboard, mouse, pencil and many animals. This architecture successful push ImageNet Classification accuracy by a significant stride in comparison to conventional classifier methodologies and become ILSVRC 2012 winner, outperformed other hand-crafted features and resulting in a top-5 error rate of 16.4%. It is composed of five convolutional layers followed by three fully connected layers and uses ReLu (Rectified Linear Unit) for the non-linear part. There is also dropout later after FC layer with probability 0.5. Meanwhile, it has an input size of 227×227×3. AlexNet model that available in MATLAB is build up with 25 layers and the list of the layer is listed in Table 3.2. Figure 3.3 illustrates the architecture and feature of the AlexNet model is based on Table 3.2. After that, Figure 3.4 shows the standard NN before and after applying dropout. Dropout randomly switches off the activation with probability and prevents the network from over-fitting [35].

<table>
<thead>
<tr>
<th>Layer</th>
<th>Filter/ Kernel Size</th>
<th>Stride; Padding</th>
<th>Activations</th>
<th>Weights; Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Image</td>
<td>227×227×3</td>
<td>-</td>
<td>227×227×3</td>
<td>-</td>
</tr>
<tr>
<td>conv1</td>
<td>96 - 11×11×3</td>
<td>[4 4]; [0 0 0 0]</td>
<td>55×55×96</td>
<td>11×11×3×96; 1×1×96</td>
</tr>
<tr>
<td>relu1 &amp; norm1</td>
<td>-</td>
<td>-</td>
<td>55×55×96</td>
<td>-</td>
</tr>
<tr>
<td>pool1</td>
<td>3×3</td>
<td>[2 2]; [0 0 0 0]</td>
<td>27×27×96</td>
<td>-</td>
</tr>
<tr>
<td>conv2</td>
<td>2 groups of 128 – 5×5×48</td>
<td>[2 2]; [2 2 2 2]</td>
<td>27×27×256</td>
<td>5×5×48×128×2; 1×1×128×2</td>
</tr>
<tr>
<td>relu2 &amp; Norm2</td>
<td>-</td>
<td>-</td>
<td>27×27×256</td>
<td>-</td>
</tr>
<tr>
<td>pool2</td>
<td>3×3</td>
<td>[2 2]; [0 0 0 0]</td>
<td>13×13×256</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.2 Feature of AlexNet model
Table 3.2 (Continue)

<table>
<thead>
<tr>
<th>Layer</th>
<th>Size</th>
<th>Filter Size</th>
<th>Stride</th>
<th>Output Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>conv3</strong></td>
<td>384 3×3×256</td>
<td>[1 1]; [2 2 2]</td>
<td>27×27×256</td>
<td>3×3×256×384; 1×1×384</td>
</tr>
<tr>
<td>relu3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>13×13×384</td>
</tr>
<tr>
<td><strong>conv4</strong></td>
<td>2 groups of 192 – 3×3×192</td>
<td>[1 1]; [1 1 1 1]</td>
<td>13×13×384</td>
<td>3×3×192×192×2; 1×1×192×2</td>
</tr>
<tr>
<td>relu4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>13×13×384</td>
</tr>
<tr>
<td><strong>conv5</strong></td>
<td>2 groups of 128 192 – 3×3×192</td>
<td>[1 1]; [1 1 1 1]</td>
<td>13×13×256</td>
<td>3×3×192×128×2; 1×1×128×2</td>
</tr>
<tr>
<td>relu5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>13×13×256</td>
</tr>
<tr>
<td>pool5</td>
<td>3×3</td>
<td>[2 2]; [0 0 0]</td>
<td>6×6×256</td>
<td>-</td>
</tr>
<tr>
<td><strong>fc6</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1×1×4096</td>
</tr>
<tr>
<td>relu6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1×1×4096</td>
</tr>
<tr>
<td>drop6</td>
<td>50%</td>
<td>-</td>
<td>-</td>
<td>1×1×4096</td>
</tr>
<tr>
<td><strong>fc7</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1×1×4096</td>
</tr>
<tr>
<td>relu7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1×1×4096</td>
</tr>
<tr>
<td>drop7</td>
<td>50%</td>
<td>-</td>
<td>-</td>
<td>1×1×4096</td>
</tr>
<tr>
<td><strong>Fc</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1×1×108</td>
</tr>
<tr>
<td>Softmax</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1×1×108</td>
</tr>
<tr>
<td>classoutput</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 3.3 Architecture of AlexNet model
3.2.4.2 DenseNet201

DenseNet201, a 201 weight layer deep ConvNet model that is trained on more than a million images from the ImageNet database [34], has learned rich feature representations for a wide range of images and able to classify images up to 1000 object categories. For example, objects such as keyboard, mouse, pencil and many animals. Each layer in the model passes its own informative feature map to all subsequent layers and obtains additional input from all the preceding layers at the same time. In MATLAB code, the DenseNet201 pre-trained model is build up with 709 layers and an input size of 224×224×3. The architecture of DenseNet is illustrated in Figure 3.5 and Table 3.3 tabulates the layer specific feature for the DenseNet family.
Bayesian Optimization

Bayesian Optimization is a method that able scientists to save time by tuning hyperparameters of machine learning automatically. It optimizes the hyperparameters set under constraints defined and tends to improve the performances of a model on a validation set. By taking posterior evaluations into account, it updates the surrogate function along with the feedback of the objective function when choosing for the hyperparameter to evaluate next. The objective function indicates how well a set of hyperparameters performs on the validation set [36]. There are some journals published about the implementation of Bayesian Optimization to improve the performances of Deep Convolution Network in object detection [37] and text prediction [38], but still have no journal that related to the application of Bayesian Optimization in IRS up to now.

Table 3.4 shows the range of a set of variables that looking for optimization. The three hyperparameters of the model that need to be optimized are learning rate, momentum and L2 regularization. The objective function for the Bayesian optimizer is created, along with training and validation data as inputs. Then, the classification error on the validation set was returned after the objective function was trained on ConvNet. Based on the validation error rate, the best model that chosen at the final has to be tested for generalization error.
Table 3.4 Range of hyperparameters variables

<table>
<thead>
<tr>
<th>Variables</th>
<th>Definitions</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Learning Rate</td>
<td>Control how quickly the weights are adjusted with respect the loss gradient</td>
<td>$[1 \times 10^{-2} \ 1]$</td>
</tr>
<tr>
<td>Stochastic gradient descent momentum</td>
<td>Accelerate SGD in the relevant direction and dampens oscillation</td>
<td>[0.8 0.98]</td>
</tr>
<tr>
<td>L2Regularization strength</td>
<td>Act of modifying a learning algorithm to avoid overfitting</td>
<td>$[1 \times 10^{-10} \ 1 \times 10^{-2}]$</td>
</tr>
</tbody>
</table>

3.2.6 Performance Analysis

This stage involved the performance measurement of the proposed algorithm training method. This included both qualitative and quantitative analyses on different stage in the training procedure. The performance analysis is observed in perspective of classification accuracy, complexity and computational time. Lastly, both confidence level and precision measurement also take into consideration when analysis on the performance of the selected algorithm.

3.2.6.1 Analysis of Classification Process

Both classification accuracy and error are reported to know how prone a model is to make mistakes. The accuracy of the iris recognition was analysed qualitatively by referring to the confusion matrix plotted. It showed the precision of prediction and overall accuracy in the GUI form. Figure 3.6 below showed the example of the confusion matrix. The confusion matrix is a $n \times m$ matrix where each row represents the true class of given data and each column represents the predicted class. A good model will have high values along the diagonal shape. The example in the confusion matrix below illustrates that in the class label ‘045’, there are two wrong prediction results on the class label ‘028’. All the classification results on training, validation, testing, and overall samples and were stored in a .mat file.
Figure 3.6 Example of confusion matrix

Quantitatively, the performance of an algorithm was observed in kind of data. In MATLAB, the accuracy rate is calculated based on the mean number of correct predictions. Mathematically, the equations that involved in calculating classification accuracy rate and classification error are showed in Equation 3.1 and Equation 3.2 respectively.

Classification accuracy rate (%) = 
\[
\frac{\sum_{i=1}^{N} \frac{Number \ of \ correct \ predictions}{Total \ number \ of \ predictions}}{N} \times 100 \% \quad \text{…….. (3.1)}
\]

Classification error = 
\[
\frac{Number \ of \ incorrect \ predictions}{Total \ number \ of \ predictions} \quad \text{…………………… (3.2)}
\]

The training accuracy is only focus on the recognition rate on training datasets; validation accuracy focus on the recognition rate on validation datasets; testing accuracy focus on the recognition rate on testing datasets; and then, overall accuracy focus on the recognition rate on both training, validation and testing datasets.
3.2.6.2 Quantitative Analysis of Confidence Levels and Confidence Interval

This session declares how to establish confidence in the measured performance of the induced classifiers by normal distribution approximation. Confidence levels and confidence intervals are a way of quantifying the uncertainty of an estimating the performance of a machine learning algorithm on unknown data. By using the central limit theorem, it can measure how much confidence can place in measurements from the samples. In this thesis, the Wilson score interval method is applied and the formula used for calculating the confidence interval is showed in Equation 3.3 below, where

\[ p = \hat{p} \pm z \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \]  

...(3.3)

where \( \hat{p} \), classification error;
\( z \), sigma value corresponding to level of confidence;
\( n \), total number of prediction samples

The common standard confidence intervals are displayed in Table 3.5 and 95% confidence level was selected to examine the performances of the classification accuracy.

<table>
<thead>
<tr>
<th>Confidence Level</th>
<th>( z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>90%</td>
<td>( \pm 1.65 )</td>
</tr>
<tr>
<td>95%</td>
<td>( \pm 1.96 )</td>
</tr>
<tr>
<td>99%</td>
<td>( \pm 2.58 )</td>
</tr>
</tbody>
</table>

3.2.6.3 Assessment of Algorithm Complexity

A fair comparison of the complexity of the proposed algorithm is examined by training and running the algorithms under identical software platform and system configuration. The computational time was measured using a laptop with specification Intel Core i5 2.2GHz, 8GB installed RAM employing Window 8.1. There is a MATLAB version R2019a implementation. By comparison, an algorithm with shorter computational time means that it is simpler in implementation than others.
3.2.6.4 Quantitative Analysis of the System

The overall performance of the system in all aspects was analysed. This included both the complexity of the algorithm training method and performances on the IRS.

3.3 Summary

In Chapter 3, procedures and methods in training on a pre-trained network are explained. Furthermore, the performances of the algorithm are measured in both qualitative and quantitative methods.
CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

Chapter 4 presents the outcome of the simulation and investigation on the proposed methodology. It is organized by first displaying the results for the image after pre-processing in Section 4.2. It is then discusses about the transferability of the pre-trained ConvNet model by doing the evaluation on classification accuracy in Section 4.3. Afterward, the performances of algorithms regardless of Data Augmentation are evaluated in Section 4.4. To optimize hyperparameters, Bayesian Optimization had been applied and training results is displayed in Section 4.5. Follow by this, Section 4.6 discusses about the impact of layer freezing techniques in solving the limitation on computational power. In the final, the comparative performances of each algorithm are evaluated and analysed in both quantitative and qualitative insight.

4.2 Result and discussion for Pre-processing of Iris Image

4.2.1 Data Augmentation

In MATLAB, training image datasets were automatically augmented with the constraint hyperparameter range in every training batch, it does not stored in memory. Mini batch of the augmented training image is reviewed and evaluated by naked eyes. Figure 4.1 and Figure 4.2 show mini batch of training images taken from CASIA-Iris-Interval V1 Dataset and undergo non-augmentation and augmentation respectively. It can be seen that the image data samples without Data Augmentation are same as the original images from the iris dataset. For Data Augmentation, the images are obviously being transformed and viewed in different perspective.
From Figure 4.1 and Figure 4.2, it can be verified that the training images from CASIA-Iris-Interval-V1 Iris Dataset are well augmented under constrained hyperparameters set in terms of scaling, rotation, and pixel translation.

Figure 4.1 Mini Batch of iris images without Data Augmentation

Figure 4.2 Mini Batch of iris images with Data Augmentation
4.3 Result and discussion of the transferability of pre-trained model

Although Transfer Learning allows a pre-trained model to reuse and perform other tasks in a prompt way, it is important to understand how to utilize it to achieve the best performances. Capable of ConvNet in feature extraction on images similar to the conventional feature encoding method. The conventional feature encoding method captures the specific patterns of the image and enables the machine to recognize it in digital form. The conventional feature encoding method usually transfers the images into code that represent specific patterns and features according to the algorithm designed by feature engineers. In contrast, ConvNets model generates an algorithm that enables a machine to differentiate important features of the target dataset and train on them without any human supervision. Figure 4.3 displays the feature maps extracted from the first convolutional layers of the AlexNet network. The channels in earlier layers learn simple features like edge and pattern of the iris. Instead of feature map from first convolution layers of AlexNet model, feature maps extracted from the rest of convolution layers also being illustrated in Appendix C.

![Figure 4.3 Feature layer of the 1st convolutional layer in AlexNet model](image)

However, the selection of suitable pre-trained source model can greatly reinforce the transferability of Transfer Learning. Only ConvNet models are being considered in this project as there are the models specially designed for image classification. Therefore, AlexNet and DenseNet201 pre-trained models were selected. Both Table 4.1 and Table 4.2 tabulate the accuracy rate of the SVM classifier after trained using the activation feature layer from different layers.
The layer’s name and position inside the table below are following the setting of the model in the Deep Learning Toolbox of MATLAB. From Table 4.1, AlexNet pre-trained model exhibits good result with 87.43% classification accuracy rate on the third convolution layer. Apart from that, Table 4.2 shows that classification accuracy rate had achieved the highest rate at 97.07% over 756 samples from CASIA-Iris-Interval-V1 when classifier trained on the 600$^{th}$ layer of DenseNet201 pre-trained model. Since both of the pre-trained networks demonstrated great results on the classification accuracy, it indicates that both pre-trained AlexNet model and DenseNet201 model able to adapt well on IRS and have potential to train with target database for better accuracy rate. In order to achieve a higher accuracy rate, the pre-trained model can be retrained on target dataset sources and update the weight of NN.

Table 4.1 Transferability of pre-trained AlexNet model on CASIA-Iris-Interval V1 dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>Layer’s name</th>
<th>Layer’s position</th>
<th>Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-trained AlexNet</td>
<td>‘conv3’</td>
<td>10$^{th}$</td>
<td>87.43%</td>
</tr>
<tr>
<td></td>
<td>‘conv4’</td>
<td>12$^{th}$</td>
<td>86.90%</td>
</tr>
<tr>
<td></td>
<td>‘conv5’</td>
<td>14$^{th}$</td>
<td>82.14%</td>
</tr>
</tbody>
</table>

Table 4.2 Transferability of pre-trained DenseNet201 model on CASIA-Iris-Interval V1 dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>Layer’s name</th>
<th>Layer’s position</th>
<th>Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-trained DenseNet201</td>
<td>‘conv4_block9_1_bn’</td>
<td>200$^{th}$</td>
<td>89.95%</td>
</tr>
<tr>
<td></td>
<td>‘conv4_block38_0_bn’</td>
<td>400$^{th}$</td>
<td>92.99%</td>
</tr>
<tr>
<td></td>
<td>‘conv5_block18_0_bn’</td>
<td>600$^{th}$</td>
<td>97.07%</td>
</tr>
</tbody>
</table>
4.4 Result and discussion of algorithm performances regardless of Data Augmentation

In this part, only AlexNet model is selected to test on the algorithm performances for the impact of Data Augmentation. From the view of qualitative, Table 4.3 shows that the accuracy rate is higher for the model that is trained with an augmented training dataset as compared to the non-augmented training dataset. According to the numerical result in Table 4.3, there are 5.92% higher in overall accuracy rate on the model that was applying the augmented training dataset method.

The training plot in Figure 4.4 reveals that overfitting was happening on the model with the non-augmented training dataset. The overfitting occurred when the validation accuracy is higher than training accuracy after an epoch of 10. Meanwhile, there are up to 13.43% high difference between the training accuracy rate and validation accuracy rate also indicates that the model is overfitting intensively. In Figure 4.5, there has no overfitting occur for the model training with the augmented training dataset. The training accuracy rate of the model increases gradually as the epoch increases.

One of the reasons causes overfitting is there were too few examples given to train on. When the model was training using a small size dataset, it results a poor generalization performance model and hence weak in iris recognition task. Meanwhile, augmentation on training data had made the training data infinite by enriching the training data and generating more dissimilar training sample examples via random transformation of existing one. It is an artificial way in expanding the training samples, enabling the algorithm to learn more details and features of the target dataset. This can be concluded that the accuracy obtained on a new dataset depends on both the quality and quantity of the training dataset.
Table 4.3: Performances analysis on AlexNet model regardless of Data Augmentation. A total of 324 training datasets.

<table>
<thead>
<tr>
<th>Model Performances</th>
<th>Non-augmented training dataset</th>
<th>Augmented training dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy(%)</td>
<td>Training</td>
<td>100.00</td>
</tr>
<tr>
<td></td>
<td>Validation</td>
<td>86.57</td>
</tr>
<tr>
<td></td>
<td>Testing</td>
<td>84.72</td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>91.80</td>
</tr>
<tr>
<td>Loss</td>
<td>Training</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Validation</td>
<td>0.1343</td>
</tr>
<tr>
<td></td>
<td>Testing</td>
<td>0.1528</td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>0.0820</td>
</tr>
</tbody>
</table>

Figure 4.4: Training plot of AlexNet model on non-augmented CASIA-Iris-Interval V1 training dataset
4.5 Result and discussion for Bayesian Optimization

In Table 4.4 and Table 4.5, Bayesian Optimization was used to find out the optimum values for the hyperparameters of the model. The three hyperparameters of the model that need to be optimized are learning rate, momentum, and L2regularization. The following sub session shows the results of Bayesian Optimization on the AlexNet model, as well as the DenseNet201 model that both trained with the target dataset.

4.5.1 Bayesian Optimization on AlexNet model

Both Table 4.4 and Figure 4.6 show the training results displayed on MATLAB command window after applying Bayesian Optimization on AlexNet model. The 8th iteration presents the least validation error of 0.20833 with 0.001 of initial learning rate value, 0.90305 of momentum value and 1.61×10⁻⁶ L2Regularization value.
<table>
<thead>
<tr>
<th>Iter</th>
<th>Eval. result</th>
<th>Objective runtime</th>
<th>BestSoFar (observed)</th>
<th>BestSoFar (estim.)</th>
<th>Initial LearnRate</th>
<th>Momentum</th>
<th>L2Regularization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Best</td>
<td>1102.1</td>
<td>1</td>
<td>1</td>
<td>0.0083643</td>
<td>0.92651</td>
<td>1.01×10⁻⁶</td>
</tr>
<tr>
<td>2</td>
<td>Best</td>
<td>1116.3</td>
<td>0.99074</td>
<td>0.99129</td>
<td>0.054036</td>
<td>0.82596</td>
<td>1.82×10⁻¹⁰</td>
</tr>
<tr>
<td>3</td>
<td>Accept</td>
<td>1135.1</td>
<td>0.99074</td>
<td>0.99109</td>
<td>0.065326</td>
<td>0.89065</td>
<td>0.0035962</td>
</tr>
<tr>
<td>4</td>
<td>Best</td>
<td>1167.7</td>
<td>0.31944</td>
<td>0.31951</td>
<td>0.0026205</td>
<td>0.80522</td>
<td>1.63×10⁻⁶</td>
</tr>
<tr>
<td>5</td>
<td>Best</td>
<td>1187.2</td>
<td>0.28241</td>
<td>0.2824</td>
<td>0.0017084</td>
<td>0.94551</td>
<td>9.14×10⁻⁷</td>
</tr>
<tr>
<td>6</td>
<td>Best</td>
<td>1242</td>
<td>0.24537</td>
<td>0.24549</td>
<td>0.0010002</td>
<td>0.8095</td>
<td>3.76×10⁻¹⁰</td>
</tr>
<tr>
<td>7</td>
<td>Accept</td>
<td>1443.1</td>
<td>0.24537</td>
<td>0.26538</td>
<td>0.0011496</td>
<td>0.90623</td>
<td>2.64×10⁻ⁱ⁰</td>
</tr>
<tr>
<td>8</td>
<td>Best</td>
<td>1201.8</td>
<td>0.20833</td>
<td>0.21361</td>
<td>0.001</td>
<td>0.90305</td>
<td>1.61×10⁻⁶</td>
</tr>
<tr>
<td>9</td>
<td>Accept</td>
<td>1186.1</td>
<td>0.20833</td>
<td>0.22347</td>
<td>0.0010028</td>
<td>0.87346</td>
<td>6.69×10⁻⁷</td>
</tr>
<tr>
<td>10</td>
<td>Accept</td>
<td>1202.2</td>
<td>0.20833</td>
<td>0.22505</td>
<td>0.0010042</td>
<td>0.8693</td>
<td>0.0092716</td>
</tr>
</tbody>
</table>

Optimization completed.
MaxObjectiveEvaluations of 10 reached.
Total function evaluations: 10
Total elapsed time: 1209.3883 seconds.
Total objective function evaluation time: 11983.5519

Best observed feasible point:
InitialLearnRate Momentum L2Regularization
0.001 0.90305 1.6113e-06

Observed objective function value = 0.20833
Estimated objective function value = 0.22505
Function evaluation time = 1201.8172

Best estimated feasible point (according to models):
InitialLearnRate Momentum L2Regularization
0.001 0.90305 1.6113e-06

Estimated objective function value = 0.22505
Estimated function evaluation time = 1195.15
4.5.2 Bayesian Optimization on DenseNet201 model

Table 4.5 and Figure 4.7 shows the training result after applying Bayesian Optimization on DenseNet201 model. The 4th iteration presents the least validation error of 0.037037 with 0.057675 of initial learning rate value, 0.8295 of momentum value and $9.11 \times 10^{-8}$ L2Regularization value.

Table 4.5 Optimization result on Initial Learning Rate, Momentum and L2Regularization trained on DenseNet201 model with CASIA-Iris-Interval V1 dataset

<table>
<thead>
<tr>
<th>Iter</th>
<th>Eval. result</th>
<th>Objective</th>
<th>Objective runtime</th>
<th>BestSoFar (observed)</th>
<th>BestSoFar (estim.)</th>
<th>Initial LearnRate</th>
<th>Momentum</th>
<th>L2-Regularization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Best</td>
<td>0.046296</td>
<td>8346.5</td>
<td>0.046296</td>
<td>0.046296</td>
<td>0.010902</td>
<td>0.91332</td>
<td>1.14×10^{-9}</td>
</tr>
<tr>
<td>2</td>
<td>Accept</td>
<td>0.9321</td>
<td>8367.5</td>
<td>0.046296</td>
<td>0.098622</td>
<td>0.0010587</td>
<td>0.81378</td>
<td>1.37×10^{-7}</td>
</tr>
<tr>
<td>3</td>
<td>Best</td>
<td>0.040123</td>
<td>8404</td>
<td>0.040123</td>
<td>0.040134</td>
<td>0.046683</td>
<td>0.94194</td>
<td>0.0002202</td>
</tr>
<tr>
<td>4</td>
<td>Best</td>
<td>0.037037</td>
<td>8203</td>
<td>0.037037</td>
<td>0.037331</td>
<td>0.057675</td>
<td>0.8295</td>
<td>9.11×10^{-8}</td>
</tr>
</tbody>
</table>

Optimization completed.
MaxTime of 28800 seconds reached.
Total function evaluations: 4
Total elapsed time: 33358.7022 seconds.
Total objective function evaluation time: 33320.9569

Best observed feasible point:
InitialLearnRate    Momentum    L2Regularization
___________    _________    ____________________
0.057675         0.8295        9.111e-08

Observed objective function value = 0.037037
Estimated objective function value = 0.037331
Function evaluation time = 8202.9941

Best estimated feasible point (according to models):
InitialLearnRate    Momentum    L2Regularization
___________    _________    ____________________
0.057675         0.8295        9.111e-08

Estimated objective function value = 0.037331
Estimated function evaluation time = 8329.8878

Figure 4.7 Script result of Bayesian Optimization on DenseNet201 model with CASIA-Iris-Interval V1 dataset
From the view of data, the DenseNet201 model performed lesser iteration than the AlexNet model in selecting hyperparameters as the maximum objective function evaluation time was achieved. Instead of less evaluation, the DenseNet201 model still has a lower estimated objective function value which is 0.037331 as compared to 0.22505 in the AlexNet model. Although the DenseNet model with a deeper layer has a higher classification accuracy rate in the validation set, the computational time estimated to train on the target dataset is massive, which is 8329.8878s. This is approximately six times higher than the AlexNet model, which needs 1195.15s.

For both models, different sets of hyperparameters show an obvious different impact on the performance of the model trained. Hence, search for possible hyperparameters by applying Bayesian Optimization made training progress effective and efficient instead of using trial and error method.

### 4.6 Result and performance of layer freezing on training progress

Along the project, there are few challenges encountered and one of them is the systems hanging when there are re-training process on the entire DenseNet201 model is implemented. Hence, layer freezing technique is utilized to reduce the burden of the system on computational works. Table 4.6 shows the impact of the layer freezing on both the computational time and the accuracy rate. As the layer freezing decreases, the computational time and accuracy rate of the DenseNet201 model increases. It indicates that when more layers are left for re-training, more computational power needed for weight updating and more accurate the result getting. From the Table 4.6, there are significant raising in computational time, which is from approximately one hour training period to fifteen hours training period when the layer froze decrease from 600th layer to 200th layer.
### Table 4.6 Recognition Accuracy Rate versus Layer Freezing on DenseNet201 model with CASIA-Iris-Interval V1 dataset

<table>
<thead>
<tr>
<th>Layer Freezing (out of 709 layer)</th>
<th>Computational Time (hh:mm:ss)</th>
<th>Performance</th>
<th>Accuracy Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>600&lt;sup&gt;th&lt;/sup&gt;</td>
<td>01:15:03</td>
<td>Validation</td>
<td>94.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test</td>
<td>92.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Overall</td>
<td>96.16</td>
</tr>
<tr>
<td>400&lt;sup&gt;th&lt;/sup&gt;</td>
<td>02:01:02</td>
<td>Validation</td>
<td>96.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test</td>
<td>96.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Overall</td>
<td>97.88</td>
</tr>
<tr>
<td>200&lt;sup&gt;th&lt;/sup&gt;</td>
<td>15:43:48</td>
<td>Validation</td>
<td>97.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test</td>
<td>98.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Overall</td>
<td>98.81</td>
</tr>
</tbody>
</table>

#### 4.7 Result and performance of algorithm performances of the final algorithm

Along with the algorithm training progress, both AlexNet and DenseNet201 algorithms that possess the highest accuracy rate in classifier CASIA-Iris-Interval V1 are being selected. Table 4.7 tabulates the specification details and performances of them independently. During the training process, re-train all layers on DenseNet201 made the computational hardware system hang and run out of memory. This is due to DenseNet201 is a big model and hardware need to have more computational power in order to process massive computational works. Under these circumstances, freezing the first 200 layers of the DenseNet was done in case to solve the problem of hardware computational power.

As compared to the AlexNet, the DenseNet201 model is a complex NN that possess 201 weight layer depth and thus, the computational time needed for training progress is much higher in the DenseNet201. 98.81% of the accuracy rate in the DenseNet201 model is slightly more than the accuracy rate of 97.22% in AlexNet. There is a trade-off between complexity and accuracy rate. AlexNet has much more difference between overall accuracy and test accuracy than DenseNet201, and this represents that AlexNet is much weaker in doing prediction on a new test set that unseen before. To evaluate the performance of accuracy from a statistical perspective,
AlexNet has a 95% probability that the test error rate will fall within a range of 0.0250 to 0.0861. Meanwhile, DenseNet201 has an interval range of 0.0005 to 0.0365.

Table 4.7 Specification details and performances of re-trained model based on proposed technique

<table>
<thead>
<tr>
<th>Model Specifications</th>
<th>Re-trained</th>
<th>AlexNet</th>
<th>DenseNet201</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity (Layer of depth)</td>
<td></td>
<td>8</td>
<td>201</td>
</tr>
<tr>
<td>Image Augmentation</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Bayesian Optimization</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Layer Freezing (Optional)</td>
<td>None</td>
<td></td>
<td>200th out of 709</td>
</tr>
<tr>
<td>Computational Time / hh:mm:ss</td>
<td>00:51:54</td>
<td>15:43:48</td>
<td></td>
</tr>
<tr>
<td>Processing Time /s</td>
<td>0.125955</td>
<td>1.467658</td>
<td></td>
</tr>
<tr>
<td>Overall Accuracy /%</td>
<td>97.22</td>
<td></td>
<td>98.81</td>
</tr>
<tr>
<td>Test Accuracy /%</td>
<td>94.44</td>
<td></td>
<td>98.15</td>
</tr>
<tr>
<td>95% Confidence Interval of Test Error</td>
<td>[0.0250 0.0861]</td>
<td></td>
<td>[0.0005 0.0365]</td>
</tr>
<tr>
<td>Lowest Precision Rate</td>
<td>42.86% on user no.'104'</td>
<td></td>
<td>57.14% on user no.'104'</td>
</tr>
</tbody>
</table>

Interestingly, both of these developed algorithms using different ConvNets model have a low precision rate on the registered user labeled ‘104’ in CASIA-Iris-Interval V1 datasets. Re-trained DenseNet201 model has four out of seven images (51.14%) predicted correctly in class ‘104’ while for retrained AlexNet model, it has only three out of total seven images (42.86%). The calculation part of precision for class is shown in Appendix D.

It can be explained that overall accuracy is mostly affected by low precision in the user class of ‘104’. After checking on the iris images respective user, it found that there is a bit difference between the training dataset (Figure 4.8 (a)) validation dataset (Figure 4.8 (b)) and test dataset (Figure 4.8 (c)). This can be due to training dataset has an iris image with smaller pupil’s diameter as compared to validation datasets and testing datasets. The weight parameter tuned based on the features extracted from the training dataset during the training stage affects the probability in prediction the class of the respective iris images. To solve this, shuffle images of dataset before training is suggested.
4.8 Summary

In this chapter, Transfer Learning is utilized in two different ways for various purposes. At first, SVM classifier trained using the features extracted from pre-trained model layers and test the transferability for both AlexNet and DenseNet201. Secondly, the pre-trained model is re-trained again after verifying the transferability of the pre-trained model for better performances. Along with model re-training process, others techniques like Data Augmentation on iris image datasets and Bayesian Optimization had been applied for better performances. For DenseNet201, Layer Freezing method was used to overcome the computational power problem on ready hardware. Finally, the overall performances of the re-trained model with the techniques proposed are tabulated and had been analysed in various circumstances.
CHAPTER 5

CONCLUSION

5.1 Conclusion

In this project, the IRS with the proposed technique workflow was successfully developed by using MATLAB code. The developed system able to recognize the 108 class of input image inserted from target dataset, CASIA-Iris-Interval V1 dataset. To overcome poor imaging condition problem, proposed training method with Transfer Learning able to take existing algorithms as a benchmark instead of scratch. Before the algorithm re-training process, the model is pre-test using an extracted feature on the SVM classifier and does classification on the target dataset. This method aid to identify the transferability of the selected model and successful rate of Transfer Learning after re-train on with new target sources. Based on the results in Section 4.3, both AlexNet and DenseNet201 model able to achieve more than 80% accuracy and hence, it indicates the model’s high transferability on the IRS.

For training dataset in this thesis, we only have three samples for each user class. Data Augmentation technique helps to transform original images of the target dataset and provide more different samples for the training procedure. This methods expands the size of the training dataset indirectly and can help prevent the algorithm from over-fitting. By implementing Data Augmentation work on training datasets, the small size iris datasets problem had been solved and overall accuracy rate of iris images on trained AlexNet model can be increased from 91.80% to 97.72%.

Moreover, optimum values for hyperparameters were successfully found through the Bayesian Optimization technique. With the automated estimation on optimum hyperparameters, works on the training process became effective and efficient as steps on tuning hyperparameters using trial and error technique can be eliminated. To overcome the hardware limitation, Layer Freezing techniques is applied to reduce the burden of the computational power of the model especially for large size model like DenseNet201 with 201 depth layers. When there are more layers had been frozen, less computational time is needed and less accurate the recognition rate.
Coupled with those proposed techniques, flexibility of the pre-trained model on new environment had improved in this project.

In conclusion, Transfer Learning is a powerful method to transfer knowledge for features extraction and classification on a new task. In this paper, both AlexNet and DenseNet201 model able to adapt on IRS with an high accuracy rate of 97.22% and 98.81% respectively. However, there is trade-off between the complexity of model with the computational time and algorithm processing time. To be justified, DenseNet201 is a computationally intensive network with its rich dense connections, allowing neurons to interact easily and achieving better recognition accuracy than AlexNet in the iris recognition task. Architecture of AlexNet model is small and may not able to fully capture the discriminative visual features in iris images as mentioned in [9]. The proposed methodology able to solve poor imaging condition in classical ML, flexibility of models, and limited dataset size as declared in problem statement section. It is important to understand the mechanism of Transfer Learning when trying to utilize it in a new environment and making smart choices on hyperparameters as well as a selection of the source dataset and model. All of the procedures are crucial in developing a well fit neural network that able to solve multiple tasks.

5.2 Suggestion for future implementation

Other than the selected criteria, there are still lots of other outstanding published networks such as GoogleNet and NaSNet-Large. This project can be extended to adapt on other available models on the various public iris datasets. To minimize the trade-off matter, the model reduction technique can be considered to eliminate redundant layers, thus decrease the size of the network. Hardware with high computational power needed to be taken into account when training on a huge ConvNet network. Besides, the size of the final trained model has to concern when it is designed to deploy a model to run locally on mobile. A portable IRS can be realized by integrating a powerful and memory-efficient network using a Transfer Learning technique onto a handy device.
REFERENCES


[23] Shai Shalev-Shwartz and S. Ben-David, UNDERSTANDING MACHINE LEARNING From Theory to Algorithms: Cambridge University Press, 2014.


APPENDICES

APPENDIX A - MATLAB code for algorithm development

%% load ConvNets network
alex = alexnet;
dense=densenet201;

%% modified AlexNet model
layers = alex.Layers;
layers(23) = fullyConnectedLayer(5);
layers(25) = classificationLayer;
layersTransfer = layers(1:end-3);
layers = [ layersTransfer;
            fullyConnectedLayer
            (numClasses,'WeightLearnRateFactor',20,...
            'BiasLearnRateFactor',20);
            softmaxLayer;
            classificationLayer];

%% modified Densnet201 model
layers = modifieddensenet.Layers;
lgraph = layerGraph(layers);
connections = modifieddensenet.Connections;
layers(1:700) = freezeWeights(layers(1:700));
lgraph = createLgraphUsingConnections(layers,connections);

%% load iris database
digitDatasetPath =
fullfile('C:\','Users','USER','Desktop','Iris Database');
imds = imageDatastore(digitDatasetPath, ...'
    'IncludeSubfolders',true,'LabelSource','foldernames');
% data allocation
[trainingImages, validationImages, testImages] = 
splitEachLabel(imds, 0.42, 0.29, 0.29);
inputSize = [227 227 3];
pixelRange = [-30 30];
Range = [-10 10];
scale = [0.5 2];
% data augmentation parameter
imageAugmenter = imageDataAugmenter( 
    'RandScale', scale, ...
    'RandRotation', Range, ...
    'RandXTranslation', pixelRange, ...
    'RandYTranslation', pixelRange);
% data augmentation
augimds = augmentedImageDatastore(inputSize(1:2), imds, ...
    'ColorPreprocessing', 'gray2rgb');
augimdsTrain = 
augmentedImageDatastore(inputSize(1:2), trainingImages, ...
    'ColorPreprocessing', 'gray2rgb', ...
    'DataAugmentation', imageAugmenter);
augimdsValidation= 
augmentedImageDatastore(inputSize(1:2), validationImages, .. ...
    'ColorPreprocessing', 'gray2rgb');
augimdsTest = 
augmentedImageDatastore(inputSize(1:2), testImages, ...
    'ColorPreprocessing', 'gray2rgb');

% display augmented train dataset
minibatch = preview(augimdsTrain);
[data, info] = read(augimdsTrain);
imshow(imtile(minibatch.input));
numClasses = numel(categories(trainingImages.Labels));
% options for training
opts = trainingOptions('sgdm', ...
    'ExecutionEnvironment','cpu', ...
    'InitialLearnRate',0.001,...
    'Momentum',0.8874,...
    'L2Regularization',3.197e-07,...
    'ValidationFrequency',10, ... 'ValidationData',augimdsValidation, ... 'MaxEpochs',120,... 'MiniBatchSize',224,... 'Shuffle','every-epoch',...
    'Plots','training-progress');

% AlexNet network train
myNet = trainNetwork(augimdsTrain, layers, opts);
% DenseNet201 network train
myNet = trainNetwork(augimdsTrain, lgraph, opts);

%% display Network layers in GUI form
analyzeNetwork(layers);

%% train
[YPredicted,scores] = classify(myNet,augimdsTrain);
YTrain = trainingImages.Labels;
accuracytrain = mean(YPredicted == YTrain);
Errortrain = 1 - mean(YPredicted == YTrain);

%% test
[YPredicted1,scores] = classify(myNet,augimdsTest);
YTest = testImages.Labels;
accuracytest = mean(YPredicted1 == YTest);
testError = 1 - mean(YPredicted1 == YTest);
%% validation
[YPredicted2,scores] = classify(myNet,augimdsValidation);
YVal = validationImages.Labels;
valError = 1 - mean(YPredicted2 == YVal);
accuracyval=mean(YPredicted2 == YVal);
%
%% Calculate confidence interval
NTest = numel(YTest);
testErrorSE = sqrt(testError*(1-testError)/NTest);
testError95CI = [testError - 1.96*testErrorSE, testError + 1.96*testErrorSE];
%
%% Confusion Matrix Sketch
NVal = numel(YVal);
figure
[cmat,classNames] = confusionmat(YVal,YPredicted2);
h = heatmap(classNames,classNames,cmat);
xlabel('Predicted Class');
ylabel('True Class');
title('Confusion Matrix');
%% SVM, columns
% Net= selected ConvNet model; AlexNet or DenseNet201
featureLayer = 'conv3'; % layer’s name
trainingFeatures = activations(Net, augimdsTrain, ...
featureLayer, 'MiniBatchSize', 108, 'OutputAs',
'columns');

% Get training labels from the trainingSet
trainingLabels = trainingImages.Labels;

% Train multiclass SVM classifier using a fast linear
solver, and set
% 'ObservationsIn' to 'columns' to match the arrangement
used for training
% features.
classifier = fitcecoc(trainingFeatures, ...
trainingLabels,'Learners', 'Linear', 'Coding', ...
'onevsall', 'ObservationsIn', 'columns');

%% Extract test features using the CNN
testFeatures = activations(alex, augimds, ...
featureLayer,'MiniBatchSize', 108, 'OutputAs', ...
'columns');

% Pass CNN image features to trained classifier
predictedLabels = predict(classifier, testFeatures, ...
'ObservationsIn', 'columns');

% Get the known labels
augimdsLabels = imds.Labels;

% Tabulate the results using a confusion matrix.
confMat = confusionmat(augimdsLabels, predictedLabels);

% Convert confusion matrix into percentage form
confMat = bsxfun(@rdivide, confMat, sum(confMat, 2))
ans=mean(diag(confMat))

%% Test accuracy
testImage = readimage(imds, 7);
testLabel = imds.Labels(7);
imageSize=[227 227 3];
ds = augmentedImageDatastore(inputSize(1:2), testImage);

% Extract image features using the CNN
imageFeatures = activations(myNet, ds, featureLayer, ...
    'OutputAs', 'columns');

% Make a prediction using the classifier
predictedLabel = predict(classifier, imageFeatures, ...
    'ObservationsIn', 'columns')
APPENDIX C - Feature Map extracted from AlexNet model

Figure C.1 Feature Map extracted from the 2\textsuperscript{nd} Convolutional Layer
Figure C.2 Feature Map extracted from the 3\textsuperscript{rd} Convolutional Layer
Figure C.3 Feature Map extracted from 4th Convolutional Layer
Figure C.4 Feature Map extracted from 5th Convolutional Layer
APPENDIX D- Precision of proposed trained algorithm

\[
\text{Precision} = \frac{TP}{TP+FP} \times 100\% \quad (7)
\]

where \(TP\), true positive,
Total number of testing sample that get the correct predicted class result;

\(FP\), false positive,
Total number of testing sample that get the wrong predicted class result

Table D.1: Precision for AlexNet model on CASIA-Iris-Interval V1 dataset

<table>
<thead>
<tr>
<th>AlexNet</th>
<th>Precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classes no. over total of 108</td>
<td>No. of (TP)</td>
</tr>
<tr>
<td>18</td>
<td>6</td>
</tr>
<tr>
<td>28</td>
<td>5</td>
</tr>
<tr>
<td>35</td>
<td>6</td>
</tr>
<tr>
<td>45</td>
<td>5</td>
</tr>
<tr>
<td>49</td>
<td>6</td>
</tr>
<tr>
<td>53</td>
<td>5</td>
</tr>
<tr>
<td>68</td>
<td>6</td>
</tr>
<tr>
<td>76</td>
<td>5</td>
</tr>
<tr>
<td>79</td>
<td>6</td>
</tr>
<tr>
<td>85</td>
<td>6</td>
</tr>
<tr>
<td>102</td>
<td>6</td>
</tr>
<tr>
<td>104</td>
<td>3</td>
</tr>
<tr>
<td>107</td>
<td>5</td>
</tr>
<tr>
<td>Others</td>
<td>7</td>
</tr>
</tbody>
</table>
### Table D.2: Precision for DenseNet201 model on CASIA-Iris-Interval V1 dataset

<table>
<thead>
<tr>
<th>Classes no. over total of 108</th>
<th>No. of (TP)</th>
<th>Precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>6</td>
<td>85.71</td>
</tr>
<tr>
<td>74</td>
<td>6</td>
<td>85.71</td>
</tr>
<tr>
<td>76</td>
<td>6</td>
<td>85.71</td>
</tr>
<tr>
<td>81</td>
<td>6</td>
<td>85.71</td>
</tr>
<tr>
<td>104</td>
<td>4</td>
<td>57.14</td>
</tr>
<tr>
<td>107</td>
<td>5</td>
<td>71.43</td>
</tr>
<tr>
<td>Others</td>
<td>7</td>
<td>100</td>
</tr>
</tbody>
</table>