การตรวจจับและรู้จำภาพหลายใบหน้าด้วยระบบคอมพิวเตอร์วิทัศน์ฝังตัว

นายสะหวาด ชัยประดิษฐ์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2560
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Multiple Face Detection and Recognition on Embedded Computer Vision System

Mr. Savath Saypadith

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2017
Copyright of Chulalongkorn University

Thesis Title	Multiple Face Detection and Recognition on Embedded Computer Vision System
By	Mr. Savath Saypadith
Field of Study	Electrical Engineering
Thesis Advisor	Associate Professor Supavadee Aramvith, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

.....Dean of the Faculty of Engineering
(Associate Professor Supot Teachavorasinskun, D.Eng.)

THESIS COMMITTEE

.....Chairman
(Assistant Professor Suree Pumrin, Ph.D.)

.....Thesis Advisor
(Associate Professor Supavadee Aramvith, Ph.D.)

.....External Examiner
(Nattachai Watcharapinchai, Ph.D.)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สะหวาด ชัยประดิษฐ์ : การตรวจจับและรู้จำภาพหลายใบหน้าด้วยระบบคอมพิวเตอร์วิทัศน์ฝังตัว (Multiple Face Detection and Recognition on Embedded Computer Vision System) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. สุภาวดี อร่ามวิทย์, 50 หน้า.

การรู้จำใบหน้าถูกใช้อย่างกว้างขวางในแอพลิเคชั่นต่างๆมากมาย ดังเช่น การพิสูจน์ตัวจริงทางชีวมิติ ระบบการตรวจตรา การระบุตัวตนผู้ใช้งาน และเทคโนโลยีการระบุตัวตนในมือถือต่างๆ นอกจากนี้ ขั้นตอนวิธีทันสมัยที่อยู่ภายใต้ระบบโครงข่ายใยประสาทเชิงสังวัฒนาการ (Convolutional Neural Network, CNN) นั้นมีความแม่นยำในการรู้จำถึง 99% อย่างไรก็ตาม การใช้ CNN ในระบบการฝังตัว (embedded system) สำหรับการรู้จำภาพหลายใบหน้าพร้อมกันแบบเวลาจริงยังคงมีข้อจำกัด เนื่องจากใช้ทรัพยากรในการคำนวณที่สูง ในงานวิจัยนี้ เรานำเสนอกรอบการทำงานสำหรับการรู้จำใบหน้าหลายหน้าพร้อมกันซึ่งประกอบไปด้วย ขั้นตอนวิธีการตรวจจับใบหน้า (face detection) การรู้จำใบหน้า (face recognition) และการติดตามใบหน้า (face tracking) โดยขั้นตอนวิธีการรู้จำใบหน้าที่นำเสนออยู่ภายใต้ CNN แบบเชิงลึก ซึ่งใช้พารามิเตอร์ในการคำนวณที่ต่ำ โดยมีการติดตามใบหน้าที่มีประสิทธิภาพในขั้นตอนการตรวจจับใบหน้า และใช้เวลาประมวลผลในการรู้จำที่ลดลง โดยเราทดสอบกรอบการทำงานที่นำเสนอผ่านบอร์ด NVIDIA Jetson TX2 ซึ่งผลการทดสอบแสดงให้เห็นว่ากรอบการทำงานที่นำเสนอสามารถรู้จำใบหลายใบหน้า ได้พร้อมกันถึง 8 ใบหน้าแบบเวลาจริงได้ โดยมีเวลาการประมวลผล 5-10 เฟรมต่อวินาที และมีอัตราการรู้จำถึง 90%

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาควิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2560

5970375021 : MAJOR ELECTRICAL ENGINEERING

KEYWORDS: FACE RECOGNITION / COMPUTER VISION / CNN

SAVATH SAYPADITH: Multiple Face Detection and Recognition on Embedded Computer Vision System. ADVISOR: ASSOC. PROF. SUPAVADEE ARAMVITH, Ph.D., 50 pp.

Face recognition is widely used in many applications such as biometric for authentication, surveillance system, user-identification, and personalized technology. The state-of-the-art algorithm based on Convolutional Neural Network (CNN) can achieve up to 99% of recognition accuracy. However, there is a limitation to implement the CNN based technique into embedded system to recognize multiple face in real-time as it requires extensive computation. In this thesis, we propose a framework for multiple face recognition which consists of face detection algorithm, face recognition, and tracking. Our face recognition algorithm based on state of the art deep CNN with small computational parameters. The tracking is very effective to track the face within the scene. These lead to the reduction of the recognition processing time. We implemented the proposed framework into the NVIDIA Jetson TX2 board. The experimental results show that our proposed framework can recognize multiple faces up to 8 faces at the same time in real time with 5-10 fps of processing time and the recognition rate up to 90%.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Department: Electrical Engineering Student's Signature .....

Field of Study: Electrical Engineering Advisor's Signature .....

Academic Year: 2017

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to my thesis advisor Assoc. Prof. Dr. Supavadee Aramvith for her tireless efforts, encouragement, support and guidance throughout the entire course of my study. This achievement also would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance.

I also would like to thank the members of my thesis committee Dr. Nattachai Watcharapinchai and the chairperson, Asst. Prof. Dr. Suree Pumrin for giving critical reviews of this work and for their advice on my thesis.

Additionally, I appreciate my senior of Video Technology Research Group (VTRG), Watchara Ruangsang, Ei Ei Tun, Sovann Chen and my friend, Jitin Khemwong, who have spent their valuable time to give suggestion, help, encourage and share good throughs to me. I also thankful to all the staffs of International School of Engineering (ISE) and AUN/SEED-Net, for their support.

Finally, I must express my gratitude to my parents, everyone in my family for their love, understanding, providing me with unfailing support and continuous encouragement throughout my years of study. Without their support and encouragement, this accomplishment would not have been possible. This research has been supported in part by the Collaborative Research Project entitled Video Processing and Transmission, JICA Project for AUN/SEED-Net, Japan.

CONTENTS

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Page

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

LIST OF TABLES

Page

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

LIST OF FIGURES

Page

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Chapter 1
Introduction

1.1 Motivation and Significance of the Research Problem

Video surveillance system is widely used in the places that need a security such as company, shop, restaurant, private house, private event, road, train and subway stations. Most of them aim to improve quality and performance of security by using either digital or information and communication technologies. The surveillance system has many limitations. Noises also become a challenge to accurately analyze video. Most of the needed video information is lost because of the noise, low ambient lighting, and the low-resolution itself. It is difficult to detect the face of the object that far from the camera, almost of the video camera has no built-in face recognition function. Many systems are installed in the building to improve the security such as keycard, Personal identification number (PIN) and biometric system. Face recognition has been widely used in security systems and human-machine interaction systems. Cameras can also use this technology to track human faces and count the number of people in a certain location or even coming in through an entrance. An image can be varied in brightness, background, camera position and distance, and many other criteria which directly or indirectly affected to the analysis process. In addition, we need an algorithm to process the face image to make it accurate in recognition system. However, many face recognition algorithms based on deep learning are implemented to recognize face under variations in pose, lighting, and expression.

The general face recognition process based on deep learning is illustrated in Figure 1.1. Usually, the first applied technique is to locate the face in the image and next, handling the recognition system that given a label name over that face. The system divided into two phases: Training phase and Recognition phase. Training phase is a process to learn the face image from database and constructs a face model. Finally, the model can be an input for recognition module in the recognition phase.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

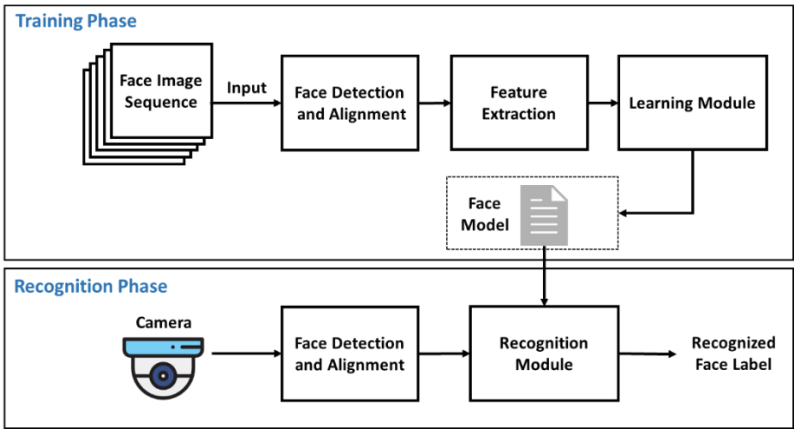


Figure 1.1. Overview of general face detection and recognition processes.

Deep learning models exploits multiple layer of information processing, feature extraction and transformation which is applies in pattern analysis and classification task, have been shown to overcome these challenges. Convolutional neural networks (CNN) was proposed by Lecun et al. [1] for deep learning model and have become the best architecture for most of the detection, classification and recognition task.

Convolutional Neural Networks is a kind of Artificial Neural Network with more than three layers. It is an efficient method developed in recent year and has solved the problems in the field of Artificial Intelligence. CNN has achieved a significantly result on vision community, improving the classification problems such as action, object, text, and voice classification. A typical architecture of CNN for face recognition is given in Figure 1.2. Each unit of a layer connects to a set of units locates in a previous layer with the ability of extracting the properties from the raw input image.

There are many types of layers used to build Convolutional Neural Networks. Typically, the network architecture consists of Convolution layer (CONV), Pooling layer (POOL), Rectified Linear Unit (ReLU) layer and Fully Connected layer (FC).

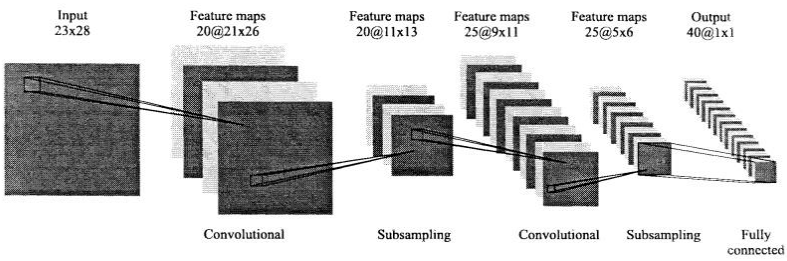


Figure 1.2. A typical convolutional network [1].

In the context of image classification, CNN may learn to detect edge from raw pixel data in the first layer, then using these edges to detect shape in the second layer. Finally, using these shapes to detect higher-level features in the highest layers of the network.

In this research we designed a framework for multiple face recognition system by adopted the algorithm for face detection, alignment, feature extraction and classification. The model we use for learning the face feature is able to run in the embedded computer vision which is suitable for real-time application.

1.2 Research Contribution

The main contribution in this research is to propose multiple face recognition framework which can be run in the embedded system. The framework divided into two phases. The first phase is a training phase that put the training set of the people face images into the recognition system. The second phase is the recognition phase which can be the detection and recognition algorithms. Face detection algorithm automatically detects the face region of the input videos and using the detected region to recognize and track the people face.

The challenging of face recognition system is face detection due to the variation, illumination, occlusion of the face image. For face detection, Convolutional Neural Networks (CNN) is trained to predict the face to ensure that the face is correctly localized in the image scene. Finally, face recognition based on CNN is applied to the region of detected face. Moreover, to reduce the processing time of overall computation in each step, we applied the correlation filter tracking technique to track a face in each frame of recognition.

1.3 Objectives

1. To develop an algorithm for face recognition system with single face and multiple faces in unconstrained environment due to illuminations, various poses and expressions.
2. To apply an algorithm for reducing the processing time using face detection and tracking technique.
3. To develop face recognition framework for applying on the embedded computer vision system.

1.4 Scopes

1. Design a face recognition framework which uses indoor CCTV camera with the resolution of 640x360 pixels as an input and a frontal view of face image without occlusion and apply to NVidia Jetson TX2 board.
2. The minimum size of face image that can be detected and recognized is 86x86 pixels.
3. Apply face tracking using correlation filter method to reduce the processing time.
4. Apply Deep Convolutional Neural Network to learn and recognize the face feature.
5. Evaluate the performance of the system by using Precision, Recall, F-Measure, recognition rate and computation time.

1.5 Expected Outputs

1. Understand Deep Convolutional Neural Network (DNN).
2. Show multiple face recognition with bounding box and person name.
3. Be able to run in real-time within embedded computer vision.
4. The real-time multiple face recognition can run at 10 fps as a frame rate. The expected processing time should not be greater than 0.10 second per frame.

1.6 Research Procedures

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1. Review literatures related to face detection, face tracking, hardware implementation, and face recognition.
2. Collection the personalized dataset and group these into different names.
3. Train the model of face recognition using the small architecture to reduce the usage of GPU memory.
4. Train a face classifier of the personalized dataset.
5. Select the most suitable method for face detection and face recognition which applied into embedded computer vision.
6. Implement face detection, tracking and recognition method with the personalized dataset.
7. Evaluation the performance of the proposed face recognition framework.

8. Summarize the results, analysis, and conclude the performance of the proposed framework.
9. Submit the paper
10. Take the proposal examination.
11. Write the thesis
12. Take the thesis defense examination.

1.7 Thesis Organization

This thesis is organized into five chapters which including this chapter. The rest of this thesis are organized and provided with descriptions as follows:

Chapter 2: describes some background and literature reviews that related to the thesis such as: understanding the fundamental of Neural Network, review of face detection and tracking algorithm, face recognition and hardware platform.

Chapter 3: the proposed framework has been presented which describes about the overall of framework including learning algorithm, face detection, face tracking and face recognition.

Chapter 4: explains the experimental and the results as well as the experimental discussion.

Chapter 5: conclusions and future works of the thesis.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Chapter 2
Literature Reviews and Background

This chapter we provided a review and background of the algorithm which is used in this research. The topics in this chapter is including fundamental of neural network, face detection and alignment, face recognition methods and face tracking algorithm.

2.1 Fundamental of Neural Networks

As a modern computer has been developed in the last decade, scientists continue to improve the performance and learn to use machines effectively for tasks that are relatively to human's brain. As we are a human, we learn easily to recognize the name of people or distinguish a name of one person to another. More experience allows us to understand and improve our performance. The development of artificial neural networks began long time ago, the motivation is to be trying to understand the human brain. The interest of neural network to researchers is different in many areas. For computer vision, neural networks are a powerful tool for machine to learn the feature of the image for detection, classification and recognition. We give a fundamental of biological of the neural network as well as the technique that use in computer vision as the following.

2.1.1 Biological Background

Neural networks are inspired by structure and behaviour of a human's brain biological neural network and its ability to learn. The basic computational unit of the brain is a neuron which consist of approximately 86 billion neurons. A neural network consists of interconnected nodes referred to as units. The basic structure of the brain structure illustrated in Figure 2.1. The dendrites receive signal from other neuron which provide input signal, then produces output signals and connect to other neurons.

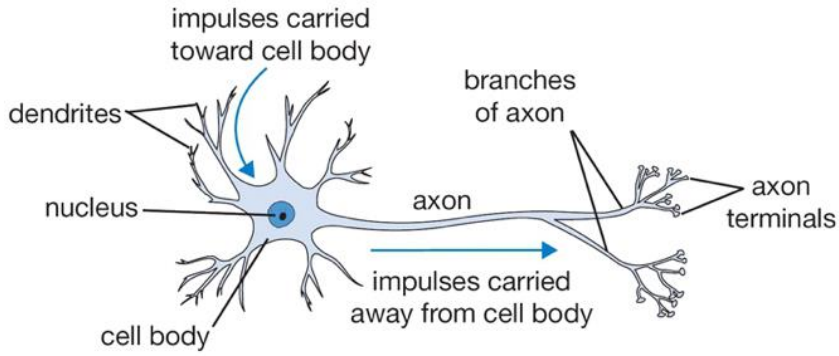


Figure 2.1. biological neuron of human brain.
Source: http://cs231n.github.io/neural-networks-1/

Figure 2.2 illustrated the model of a neural network node. The signal from node is transmit from the previous layers (x), multiply by weights (w) and summed up with the bias (b). The summed signal is mapped through an activation function to produce the node output (y).

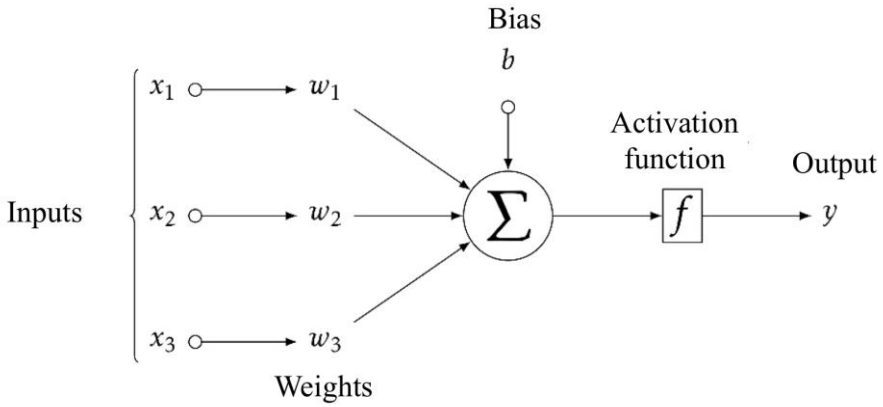


Figure 2.2. Model of a neural network node.

2.1.2 The Multi-Layer Perceptron

The form of neural network is the single layer perceptron which consists of one layer of output unit. The network can only learn with simple pattern with linear function. To learn more complex patterns, more layers need to be add in order to learn with more feature and dimensional of the data. This lead to increase the depth of the network. More layer of the network can refer to as “deep learning”. Generally, the first layer of the network is called the input layer and the last layer called the output layer which transmit the signal between the hidden layers. The hidden layers can be a convolutional layer or pooling layer as shown in Figure 2.3.

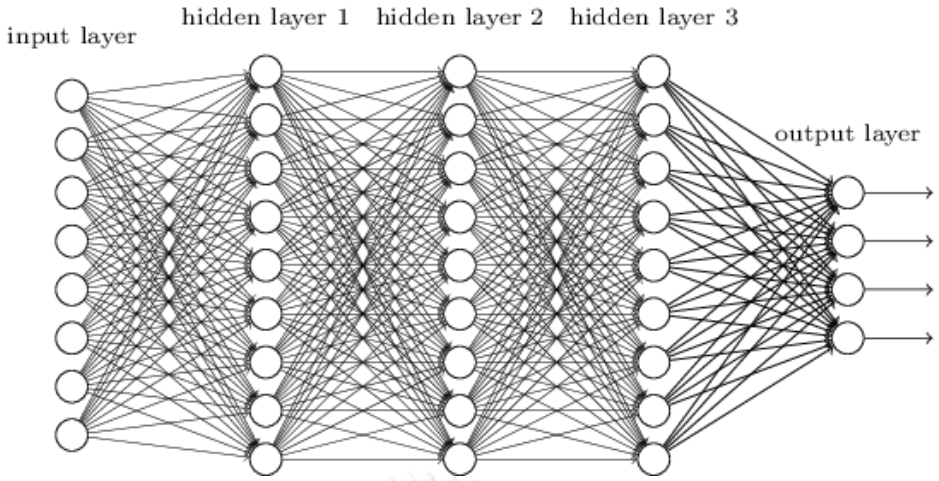
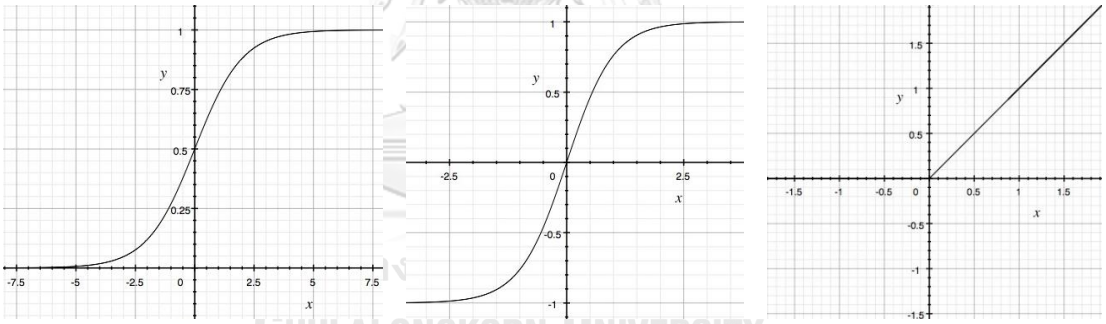


Figure 2.3. General network architecture which consists of input, hidden and output layer.

2.1.3 Activation Function

The activation function is an important for the neural network. It is used to define the output of a unit given a set of inputs. The commonly used of activation function are indicated in Figure 2.4.



(a) (b) (c)

Figure 2.4. the common used activation function in neural network: Sigmoid function(a), Tanh function(b), and Rectified Linear Unit - ReLu(c)

Sigmoid activation function is defined as $\sigma(x) = 1/(1 + e^{-x})$. The function takes a number as input, and outputs a number with a continuous range from 0 to 1, which is commonly used for the output node of a binary classification problem. The first drawback is that the outputs are not centered around zero, which can cause undesirable behavior during learning. The other drawback is the units output mostly 0 and 1 instead of anything in between, and thus the gradient at these regions is very low. If the gradient falls to zero, the network will stop learning.

Tanh activation function is very similar to the sigmoid function. It is actually the scaled version of the sigmoid function. Tanh activation function can be defined as $\tanh(x) = 2\,\sigma(2x) - 1$ or can be directly written as $\tanh(x) = \frac{2}{1+e^{-2x}} - 1$. Tanh works similar like sigmoid function but is symmetric over the origin. it ranges from -1 to 1. Tanh activation function is basically solved the problem of the values all being of the same sign. The function is non-linear, so we can easily backpropagate the errors.

Rectified Linear Unit (ReLu) is the most widely used as activation function in neural network. ReLu can be defined as $f(x) = \max(0, x)$. In other word, the ReLu function is simply threshold at zero. ReLu function is greatly accelerate the convergence of stochastic gradient descent compared to the sigmoid and tanh functions.

Another activation function that also used in neural network is softmax function. Softmax is a variant of the sigmoid function. Instead of binary classification, softmax used for the output units of a multiclass classification problem, where the goal is to classify instances into one of more than two classes. The mathematical equation for the softmax function is $f(x) = e^{x} / \sum_{i=0}^{j} e^{x_i}$. The output is the probability of each target class over the all possible target classes. A result of this is the sum of the probabilities of all classes will be equal to one. The softmax function is not used in hidden units, but only as the activation function as the output units.

2.1.4 Loss Function

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

A model of neural network has to trained with enough training samples to enable the network model to achieve high accuracy and good prediction to unseen data. The training sample must contain bias and variance to be a good representation of general case. For example, if we train a face model, several faces are not a good representation for a general feature of human face. During the training, the dataset should split into training set, validation set and test set. The training used for training the network, validate and tuning the hyperparameter with the validation set and test the network accuracy with the test set. The goal of the loss function is to measure how far of a solution from the optimal solution. The aim of the training process is thus to find a function with a smallest possible loss. There are several functions that used in neural network for classification tasks where we want to minimize the number of misclassified

training pairs. Cross-entropy and triplet loss are widely used for classification task as we will describe as the follows.

Cross-entropy measures the performance of a classification model with the output of a probability value from 0 to 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. In other word, the lowest predicted probability, the highest loss value and the actual observation label would be bad. A perfect model would have a loss of 0. Cross-entropy can calculate as:

$$Cross-entropy = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{2.1}$$

where M is the number of classes, y is a binary indicator (0 or 1) if class label c is the correct classification for observation o, and p is the predicted probability observation o is of class c.

Triplet loss was introduced by [2] which used in networks that map the data to a point in a d-dimensional space. The triplet loss based on measuring the distance between these points. The idea of triplet loss in face network training is to ensure that the reference face (anchor point) with the same face (positive point) have their embeddings close together in the embedding space, where two examples with different face (negative point) have their embeddings far away. In other word, the triplet loss tries to minimize the distance between two faces of the same person and maximize the distance of different face which visualize in Figure 2.5.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

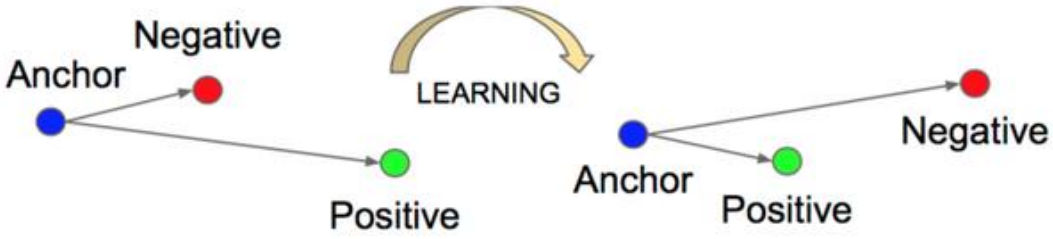


Figure 2.5. Triplet loss in FaceNet[2].

2.1.5 Training a Network

Training aims to minimize the loss function by tuning the network parameters. The network parameter often referred to the weights of neural network that can be learned by training a set of training data in a process called supervised or unsupervised

learning, the weights are initialized to small and randomly generated. Typically, in classification problem, input could be an image and the output could denote to label of the object that represent an image.

2.1.6 Backpropagation and Optimization Function

Figure 2.6 illustrated the schematic of gradient descent. Error $J(w)$ is a function of internal parameters of model such as weight and bias. The weight is initialized, the optimization function calculates the gradient and move the weight with the learning rate in the opposite direction of the calculated gradient and this process is repeated until we reach the minima of loss function.

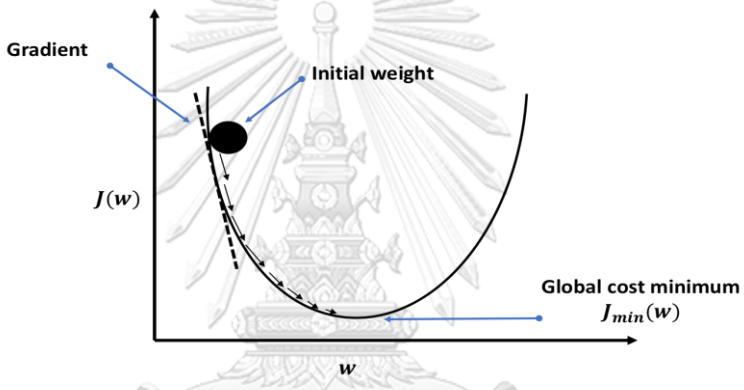


Figure 2.6. Schematic of gradient descent.

Optimization function is used to update the weights and biases such as the parameter to reduce the error. There is another optimization algorithm such as Adagrad, Adadelta, RMSprop, Adam, provide an alternative to classical gradient descent.

2.2 Convolutional Neural Networks

Convolutional neural networks (CNN) are very similar to ordinary neural networks. The neurons of CNN are transmitting the signal together which have learnable weights and biases. Each neuron receives the inputs, performs a dot product and optionally follows it with a non-linearity. In the task of image classification, the inputs are images that is encode properties into the architecture. These then applied the forward function to reduce the amount of the parameters in the network. In particular, the layer of CNN has a neuron of three dimensions: width, height and color channel (this referred to the depth in neural network).

Typically, the layers used to build the CNN consists of three types architectures: convolutional layer, pooling layer, and fully-connected layer. The detail of these layers describe in the sub-topic as follows.

2.2.1 Convolutional Layer

Convolutional layers apply a convolution operation of filters to the input, passing the result to the next layer. The aim of the convolutional layer is to learn the object features in the image. Convolutional layer parameters consist of a set of the learnable filters. The filter that convolutes to the input can be the edge filter, line filter, sharpen, blur, etc. The filter also called as a kernel in the image processing. Given a two-dimensional image *I*, and a filter *K*, we compute the convolved image *I*K* by overlapping the filter on the top of the image from left top to right bottom and sum of elementwise products between the image and the filter as shown in Figure 2.7.

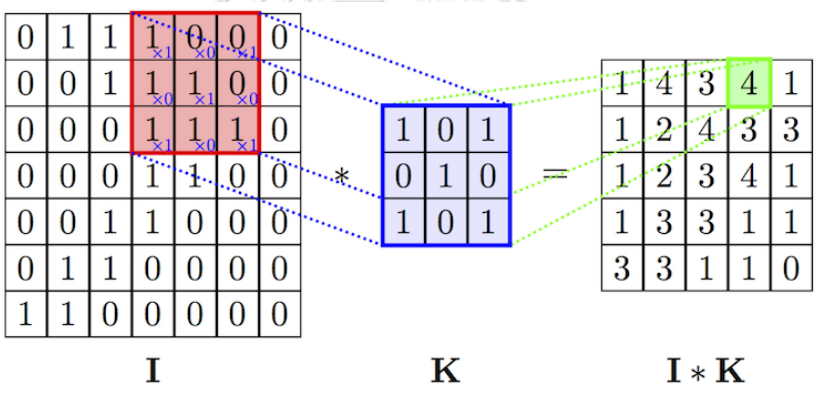


Figure 2.7. A convolution of an input image and the filter.
Source: http://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/figures/convolve.png

2.2.2 Pooling layer

Pooling layer is common used in convolutional neural network, its function is to reduce the spatial size of the representation to reduce a network parameters and computation in the network. The pooling layer also control overfitting during training the network model. The operation of computing the pooling layer is a max function where the maximum is the value of the pooling output. The example of the pooling layer show as follows.

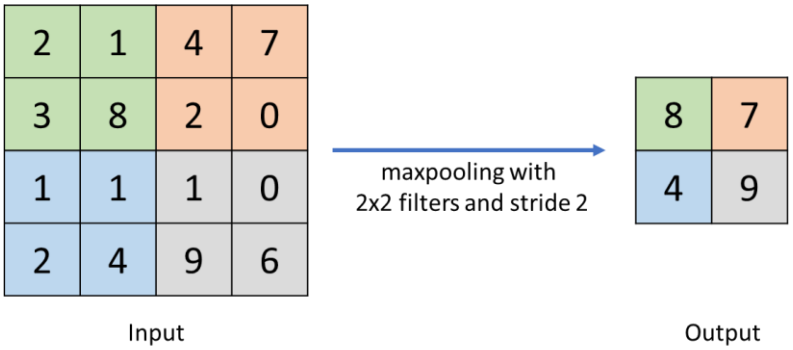


Figure 2.8. A pooling of the input image witha 2x2 filters and stride of 2.

2.2.3 Fully-connected layer

Fully-connected layer connects every neuron in one layer to every neuron in another layer. The last fully-connected layer uses a softmax activation function to classify the features of the input image into various classes based on the training dataset. For example, in the task of face recognition algorithm, fully-connected layer output the probability of each class labels.

2.2.4 Hyperparameter

Hyperparameters are variables initialize before training the neural network model. Neural network can have many hyperparameters. This includes the optimizer's hyperparameters: learning rate, decay rates, step size, and batch size as well as model's hyperparameters: number of layers, number of units at each layer, dropout rate and activation function (ReLu, Sigmoid, Tanh).

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.3 Face Detection

In face recognition system, an input image consists of many faces and complex background. The high variation of the human face appearance causes the face detection. The most important step in face recognition system is face detection which needs to be located the face and computes the region of interest. Face detection is a first step to all facial analysis algorithms including face alignment, face recognition, face parsing and face verification. To determine a certain image of a human face, we need to define the general structure of a face such as eyes, nose, forehead, chin and mouth, then determine the number of face, the exact location and the size of all the faces. Therefore, many researches have been proposed method to detect face with the different accuracy and false detection rates.

Face detection algorithm proposed by Viola and Jones [3], the algorithm is scan a sub-window capable of detecting faces across a given input image. The Viola-Jones face detector contains three main ideas that can run in real time: the image integral, classifier learning with AdaBoost, and the attentional cascade structure. The approach would be rescaled the image to different size and then run the fixed size detector through these images. However, it may consume much time due to the calculation of the different size images.

Histogram of Oriented Gradients (HOG) feature descriptor [4] has been proposed to detect objects in computer vision and image processing. The basic idea is that local shape information described by the distribution of intensity gradients or edge direction even without precise information about the location of the edges themselves. The algorithm divided image into small sub-images call “cells”, then accumulating a histogram of edge orientations within that cell. Finally, combines entries histogram and used as the feature vector describing the object. The algorithm also counts occurrences of gradient orientation in localized portion of an image. In addition, HOG descriptor is used Support Vector Machine (SVM) as a classifier. The algorithm is achieved a significant result for face detection.

Some researches proved that neural network can improve the performance of face detection, but it increases the execution time. Farfade et al. [5] proposed a method called Deep Dense Face Detector (DDFD) which is able to detect faces in a wide range of orientations using a single model based on deep convolutional neural networks. The method dose not required segmentation, bounding-box regression, or SVM classifiers. The study found that the method is able to detect faces from different angles and there seem to be correlation between distribution of positive examples in the training set and score of the proposed face detector.

Due to the features of Viola and Jones algorithm have a limited representation capacity which containing a large feature pool size. Yang, et al. [6] proposed an algorithm by adopting a novel variant of channel features called aggregate channel features. The algorithm extract features directly as pixel values without computing rectangular sums at various location and scales using integral images which has only serval thousands of the features pool size. This leads to a faster feature computation for boosting learning. The experiment proved that LUV channel and gradient magnitude

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

and 6-bin histograms computed on RGB color space are the best choice for face detection. Larger detection window size achieves a good performance but may cause miss small faces and lead to inefficient in the detection. Thus, 80 x 80 and a subsampling factor of 4 are optimal values and most reasonable according to the experiment. As a result, the face method using aggregate channel features shows competitive performance against with previous works while faster computation on VGA images.

Zhang et al. [7] proposed a framework to detect face and alignment using unified cascaded Convolutional Neural Networks (CNN) by multi-task learning which called MTCNN. The framework consists of three stages: first, to obtain the bounding box regression, they utilized a convolutional network called Proposal Network (P-Net). After getting the estimated bounding box regression candidates, they employ non-maximum suppression (NMS) to ignore bounding boxes that significantly overlap each other. All candidates are fed to Refine Network (R-Net), R-Net aims to reject a large number of false candidates, performs calibration with bounding box regression, and NMS candidate merge. Finally, the Output Network (O-Net) describe the details of the face and output five facial landmarks' position. It takes 16fps for face detection and alignment on 2.60 GHz CPU and 99fps on Nvidia Titan Black GPU. The approach achieved 95% in average of accuracy in each stage across several challenging benchmarks while keeping real-time performance.

จุฬาลงกรณ์มหาวิทยาลัย

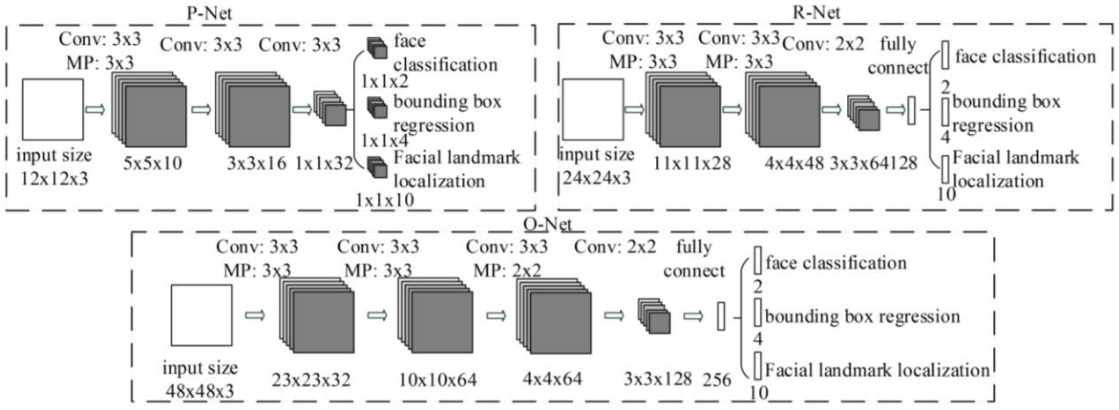


Figure 2.9. The architectures of P-Net, R-Net, and O-Net. (MP: max pooling and Conv: convolution)

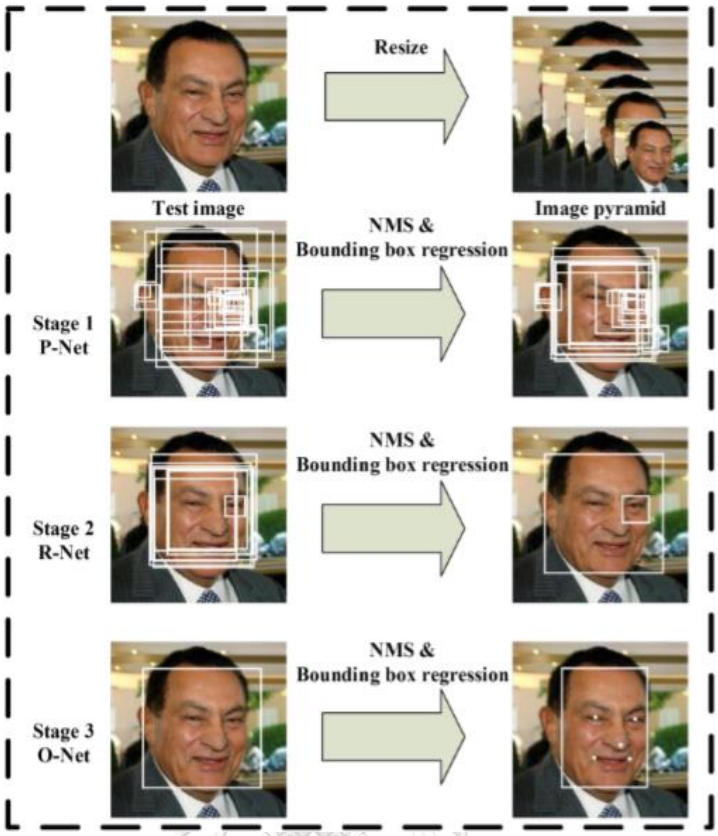


Figure 2.10. Pipeline of [7] framework to detect and align the face with three stages.

2.4 Face Alignment

The task of accurately localizing the set of landmark points that define the shape of the face is called face alignment. Face alignment is a key module in the pipeline after face detection and before feature extraction and classification of most facial analysis algorithm. In addition, face images may have required to pass through pre-processing steps to rotate and align the face due to the non-planarity and non-rigid expression. It is common for a face to not be perfectly center aligned with the image. To deal with this issue, we apply an image transformation to center a face based on the location of landmark points.

Kazemi et al. [8] proposed algorithm to precisely estimate the face's landmark position by utilizing an ensemble of regression tree. The approach can directly estimate from a sparse subset of pixel intensities in real-time with high quality predictions. Kazemi et al. proposed shrinkage and averaging the predictions of multiple regression trees as regularization method in shape regression, in order to reduce and to produce better regularization. The method is trained with iBUG 300-W (68 landmarks) [9] and

HELEN (194 landmarks) [10] database with the large variation in pose, expression, illumination, background, occlusion, and image quality. The visualizing of 68 facial landmark coordinates are show in Figure 2.11. The aim of the approach is to determine 68 or 194 landmark points on every face. As show in Figure 2.12, these landmark points consist of the chin, the edge of the eyes, eyebrow, mouth and nose. Finally, using these 68 points to rotate, scale and translate face to align with frontal face image.

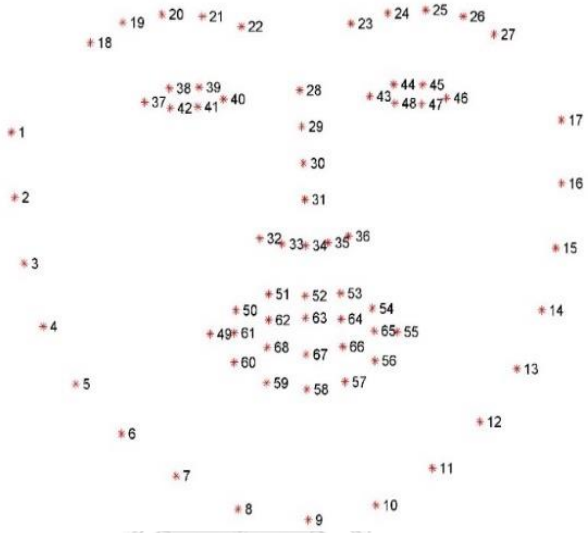


Figure 2.11. Visualizing the 68 facial landmark coordinates from the iBUG 300-W dataset.

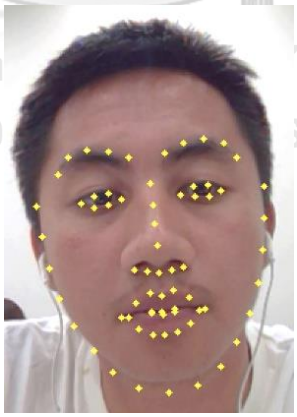


Figure 2.12. 68 Landmark points detections.

Facial landmark point approach can be estimated using the cascade of regressor as shown in Eq. (2.2). The facial feature points are typically represented as a feature vector $S = (x_1y_1, x_2y_2, \dots, x_py_p) \in \mathbb{R}^{2p}$. Where each pair (x_iy_i) is the coordinate of i-th facial landmark in a face image I and p is the number of landmarks, in this example is

68 as shown in Figure 2. Let $\hat{S}^{(t)}$ denote the current estimate of facial vector S. Each regressor $r_t(.,.)$, in the cascade predicts an update vector from the image and $\hat{S}^{(t)}$, which is added to the current shape estimate $\hat{S}^{(t)}$.

$$S^{(t+1)} = S^{(t)} + r_t(I, S^{(t)}) \tag{2.2}$$

The regressor r_t makes its prediction based on features, such as pixel intensity values, computed from image I and indexed relative to the current shape estimate $\hat{S}^{(t)}$.

Lee et al. [11] proposed a face alignment method using cascade Gaussian Process Regression Tree (cGPRT) that utilized Difference of Gaussian (DoG) features. cGPRT gives good generalization in the same prediction time when compared with Cascade Regression Tree (CRT) but high computational complexity. However, the method performed best with large amount of error reduction compared with state-of-the-art method.

Many of research in face alignment focused on estimates 2D landmarks. Jourabloo et al. [12] proposed a novel regression-based approach for pose-invariant 3D face alignment. The aim of this method is to estimate both 2D and 3D location of face landmarks. The method is extended from cascaded regressor for 2D landmark estimation. By learning two regressors of each cascade layer, one for predicting the update for the camera projection matrix, and the other is to predicts the update for the 3D shape parameter. As a result, the approach is able to estimate the locations of both 2D and 3D landmarks, as well as their 2D visibilities for a 2D image.

The challenges of face alignment task in real world application come from the large variation, illumination and occlusion. Zhang et al. [13] proposed a framework to overcome these problem by using unified cascaded CNNs by multi-task learning. The proposed consists of three stages. First, quickly produces candidate windows through a shallow CNN. Second, rejecting the number of non-face in the window. Finally, using CNN to refine the result and output facial landmarks positions.

2.5 Face Tracking

Both face detection and recognition processes can consume significant amount of the processing time. The tracking algorithm is the essential part to reduce the processing time. Danelljan et al. [14] proposed approach to learn discriminative

correlation filters based on a scale of pyramid representation. The method is improved the performance compared to an exhaustive scale search with significant scale variation. A closely related approach, proposed by Bolme et al. [15], is based on finding an adaptive correlation filter by minimizing the output sum of squared error (MOSSE). Danelljan et al. method traines a classifier on a scale pyramid. This allows to independently estimate the target scale after the optimal translation is found.

Basically, the tracker learns a discriminative correlation filter used to localize the target in a new frame. Grayscale image patches $f_1, \dots, f_t$ of the target appearance as training samples and labeling with the correlation outputs $g_1, \dots, g_t$ from the filter. By minimizing the sum of squared errors with the optima correlation filter h_t at the time step t is obtained by:

$$\varepsilon = \sum_{j=1}^{t} \left\| h_t * f_j - g_j \right\|^2 = \frac{1}{MN} \sum_{j=1}^{t} \left\| \overline{H}_t F_j - G_j \right\|^2 \tag{2.3}$$

The $\star$ denoted circular correlation. The function f_j, g_j and h_t are all of size $M \times N$. The discrete Fourier transforms (DFTs) is denoted by capital letters as the corresponding function. In this case g_j is generated from ground truth 2D Gaussian shaped peak centered on the target in training image f_j. Training is conducted in the Fourier domain to compute the image in different domain to take the advantage of space-wise of element between input and the output. $\overline{H}_t$ can calculate as following.

Chulalongkorn University

$$\overline{H}_t = \frac{G_j}{F_j} \tag{2.4}$$

The bar $\overline{H}_t$ represents complex conjugation and the product $\overline{H}_t F_j$ is point-wise. Eq. (2.4) is minimized by choosing:

$$H_t = \frac{\sum_{j=1}^{t} G_j F_j}{\sum_{j=1}^{t} \overline{F}_j F_j} \tag{2.5}$$

Then minimizing the sum of squared error between the actual output and the desired output of the convolution.

$$\min_{\bar{H}} \sum_{i} \left| F_j \star \bar{H} - G_j \right|^2 \tag{2.6}$$

Given an image patch z of size $M \times N$ in a new frame. The new target location is estimated to be at the maximum correlation score by y that can calculate by

$$y = \mathcal{F}^{-1}\{\bar{H}_t Z\} \tag{2.7}$$

Where $\mathcal{F}^{-1}$denotes the inverse DFT operator.

To estimate the target in multi-dimensional features for a variety of applications, the discriminative correlation filter above has been extended. Let f be a rectangular patch of the target, extracted from a *d*-dimensional feature map representation of an image. By denoting feature dimension number $l \in \{1, 2, ..., d\}$ of f by f^l. Then minimizing the cost function to find an optima correlation filter h, which consists of filter h^l in each feature dimensions.

$$\varepsilon = \left\| \sum_{l=1}^{d} h_t * f_j - g_j \right\| + \lambda \sum_{l=1}^{d} \left\| h^l \right\|^2 \tag{2.8}$$

Where g is the desired correlation output associated with the training example *f*. the parameter $\lambda \geq 0$ controls the impact of the regularization term and to avoid overfitting. And the correlation filter H^l can compute as follows.

$$H^l = \frac{\overline{G} F^l}{\sum_{k=1}^{d} \overline{F^k} F^k + \lambda} \tag{2.9}$$

To obtain a robust approximation, an optimal filter can be obtained by minimizing the output error over all training patch to mitigate the zero-frequency components in the spectrum of f in the regularization parameter, then we update the numerator A_t^l and denominator B_l of the correlation filter H_t^l.

$$A_t^l = (1-\eta) A_{t-1}^l + \eta \overline{G_t} F_t^l \tag{2.10}$$

$$B_t = (1-\eta)B_{t-1} + \eta\sum_{k=1}^{d}\overline{F_l^k}F_t^k \tag{2.11}$$

η denotes a learning rate parameter and the correlation score y at the rectangular region z of a feature map are computed using Eq. (2.12). the new target state is then found by maximizing the score y.

$$y = \Im^{-1}\left\{\frac{\sum_{i=1}^{n}\overline{A^l}\overline{Z^l}}{B+\lambda}\right\} \tag{2.12}$$

2.6 Network Architectures

When applying deep learning to the application such as object detection, classification and verification. Since there are different targets in real situation that we want to detect, classify and recognize, the pre-trained model has been train using network architecture with different database based on the target object. There are many network architectures has been proposed. The review of several architectures method is described as follows.

With convolutional network becoming competitive in the computer vision fields, a researcher attempts to create and improve a network architecture to achieve better accuracy. Krizhevsky et al. [16] presents a deep convolutional neural network architecture for image classification task, the network known as AlexNet which is won the challenge of 2012 ImageNet LSVRC-2012 competition by reducing the top-5 error from 26.2% to 15.3%. AlexNet contains 5 convolutional layers and 3 fully connected layers. AlexNet claim that using Rectified Linear Units (ReLUs) instead of Tanh to add non-linearity accelerate the speed of training the data by several times at the same accuracy. AlexNet utilize dropout instead of regularization as well as data augmentation to deal with overfitting, dropout is applied before the first two fully-connected layers with the rate of 0.5. However, using dropout is doubled the number of iterations of the training to converge.

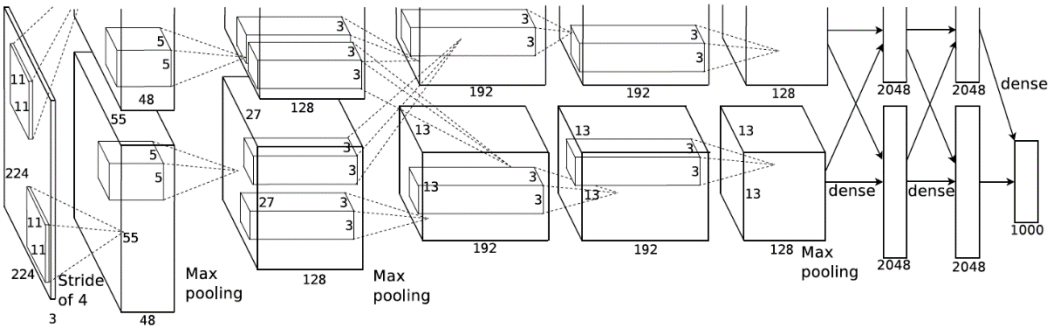


Figure 2.13. AlexNet architecture [16].

Simonyan et al. [17] proposed a network architecture called VGGNet by investigate the effect of the convolutional network depth on its accuracy based on [16] work. VGGNet consists of 16 convolutional layers with performing only 3x3 convolutional filters in all layers instead of convolving with large kernel-sized filter with the input at every pixel. This lead to decrease the number of parameters as well as the number of features. The stride of convolution is fixed to 1 pixel for 3x3 convolution layer which effected to increasing the computation cost and come up with applying 1x1 convolution layers.

Szegedy et al. [18] proposed a deep convolutional neural network architecture codenamed Inception, the architecture aim to optimize the quality of the context of classification and detection. This architecture also called GoogLeNet which consists of 22 layers deep network. The main idea of the inception architecture is to find the optimize local sparse structure in a convolutional network. The typical use of dimension reduction as well as reduce the larger number of filters of the last stage to the next layer, inception module utilized 1x1 convolutions before convolving over them with the expensive 3x3 and 5x5 convolutions. The Inception module with dimension reduction is shown in Figure 2.14. Moreover, GoogLeNet is to replace the fully-connected layers at the end with a simple global average pooling which averages out the channel values across the 2D feature map, after the last convolutional layer. This drastically reduces the total number of parameters without affecting to the accuracy.

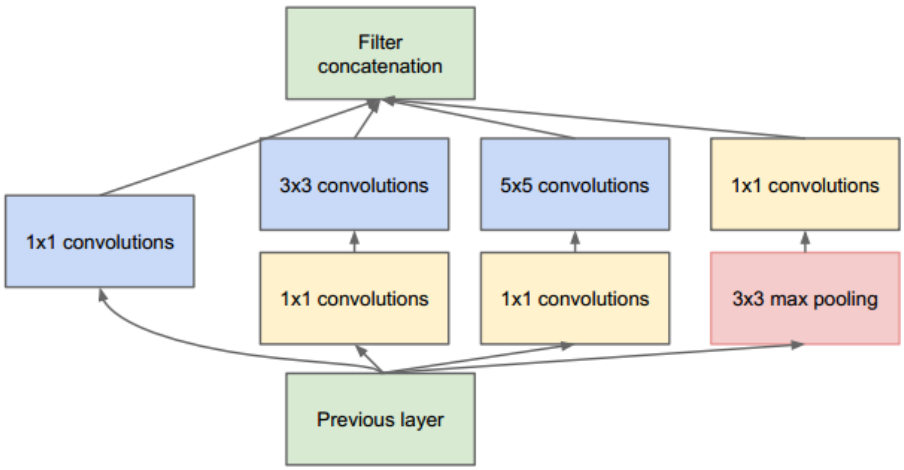


Figure 2.14. Inception module with dimensional reduction[18].

Since Inception networks tend to be very deep architectures and the residual connections are important in training of such networks. The idea is to replace the Inception in the filter concatenation stage with residual connections in order to improve the training speed.

Forrest et al. [19] proposed an algorithm to train the network by reducing the parameter compared to the original network of AlexNet. A Fire module has been defined which contains of Squeeze module and Expand Module. The architecture of Fire module and SqueezNet architecture are shown in Figure 2.15 and Figure 2.16. Squeeze module applies a 1x1 convolution that reduce the number of channels or dimensionality. Whereas, expand module applies 1x1 and 3x3 convolution, both applied to the output of the Squeeze module. Their results are concatenated. Using above algorithm may lead to drop the accuracy. To improve the accuracy, they use late max pooling rather than early while not needing more parameters. They also use pruning from Deep Compression to reduce parameter further. Pruning simply collects the 50% of all parameters of a layer that have the lowest values and sets them to zero. As a result, the method can reduce 50x parameter of AlexNet from 1.2M parameters before pruning to 0.4M after pruning. 510x file size reduction of AlexNet when combined with Deep Compression.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

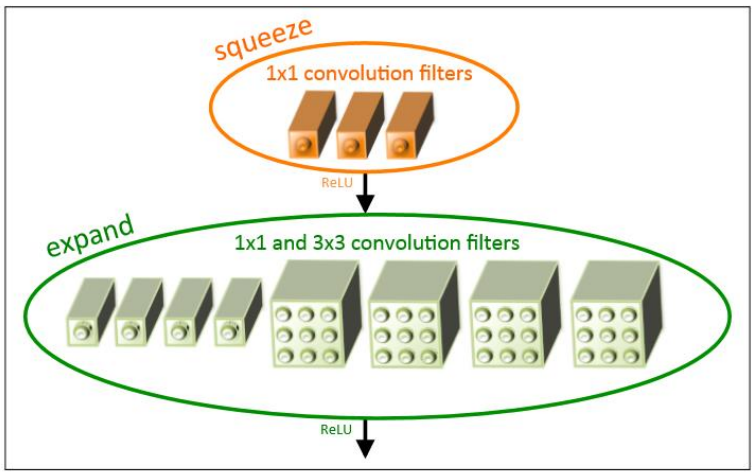


Figure 2.15. Organization of convolution filters in the Fire module.

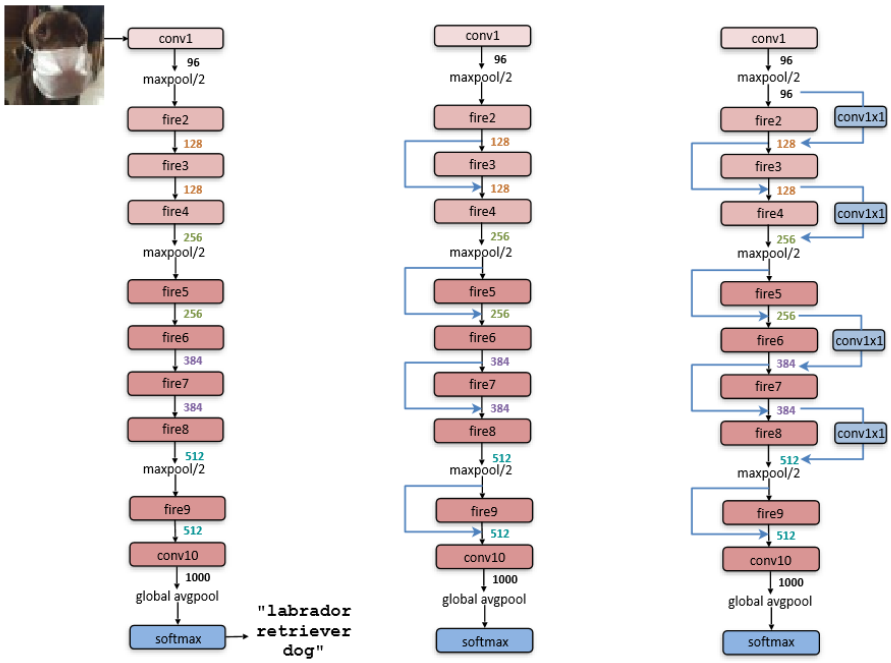


Figure 2.16. Macroarchitectural view of SqueezeNet architecture.

2.7 Face Feature Learning Algorithm Based on Convolutional Neural Network Architecture.

The most important part of face recognition system is to learn the unique feature of the face to tell the distinction from other people. By focusing on a face and be able to understand even if a face is changed by illumination or bad condition, the system should recognize the face still the same person. However, many researches found that

the most accurate approach is to let the computer learn the measurement by itself. Deep learning was proposed for face recognition and archived a good performance.

Taigman et al. [20] proposed an algorithm for face recognition called DeepFace which is combined 3D alignment and similarity metric with deep neural network. The network architecture is learned from the raw pixel RGB values as shown in Figure 2.17. It employs a nine-layer neural network with over 120 million parameters. The approach was trained on Social Face Classification (SFC) dataset which is a large collection of photo from Facebook. The dataset contains 4.4 million face images. 67 fiducial points are extracted by a Support Vector Regressor (SVR) trained to predict point configurations from an image descriptor based on Local Binary Pattern (LBP) histogram. The DeepFace runs at 0.33 seconds per image using core intel 2.2 GHz CPU, accounting for image decoding, face detection and alignment, the feedforward network, and the final classification output. In order to normalize the input image to make the face robust to different view angles, DeepFace models a face in 3D and aligns it to appear as a frontal face. Then, the normalized input is fed to a single convolution-pooling-convolution filter. Next, 3 locally connected layers and 2 fully connected layers are used to make final predictions. As a result, the system achieved 97.35% in accuracy and reducing the error more than 27%.

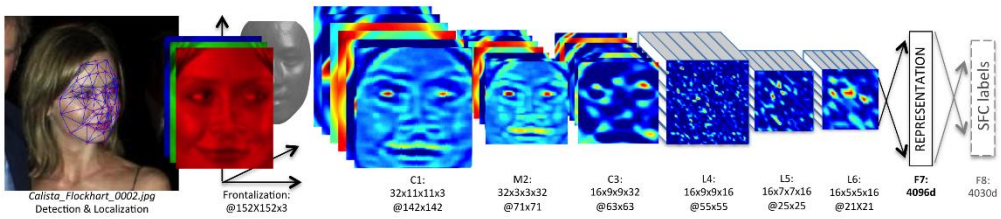


Figure 2.17. Outline of the DeepFace architecture [20].

Sun et al. [21] proposed the method called DeepID3. The method proposed two deep network architectures based on stacked convolution and inception layers to make them suitable to face recognition. By improving the method from DeepID2+ [22], they utilized Joint identification-verification supervisory signals to the last fully connected layer as well as a few fully connected layers branches out from pooling layer. This method is also used in DeepID3 to make the architecture better supervise early feature extraction processes. In addition, the DeepID3 network is significant deeper compared

to DeepID2+ by using ten to fifteen non-linear feature extraction layers. As a result, DeepID3 archived 99.53% for face verification accuracy and 96.0% for face identification accuracy.

Schroff et al. [2] proposed a face recognition algorithm called FaceNet based on Deep convolutional network which is trained to directly optimize the embedding itself. The approach is directly learning from the pixels of the face. FaceNet were used the triplet loss to minimizes the distance between an anchor and positive, it is considered the face of the same person have small distances, where the faces of distinct people have large distance. The network is trained such that the squared L2- normalization distances in the embedding space directly correspond to face similarity. They trained the CNN using Stochastic Gradient Descent (SGD) with standard back propagation and AdaGrad. The approach achieved the accuracy of 99.63%. In addition, the system used 128 bytes per face which is suitable for running in the embedded computer vision.

2.8 Hardware Platform

The hardware for face detection and recognition based on deep learning is required high computational resource to learn a feature in many layers in CNN. The hardware should support parallel programming which can compute multiple processes at the same time.

Recently, the embedded system company has designed and developed the GPU to the smaller board with consuming low power consumption, real-time and light weight. NVIDIA has developed the board that use GPU to process and support CUDA core which has ability to do parallel processing. Rungsuptaweekoon et al. [23] evaluated the performance of the embedded GPU system by using YOLO image recognition algorithm. The experiment is evaluated by NVIDIA Jetson TX1, Jetson TX2 and NVIDIA Tesla P40. In the experiment, the energy usage is divided into different profile modes by boosting the GPU and CPU frequency called Max-Q, Max-P and Max-N. The result indicated that Jetson TX2 has the advantages in image recognition which is used lower power consumption but getting higher FPS compared to Jetson TX1.

Chapter 3
Proposed Framework

This chapter is described overall of the proposed framework. The chapter is divided into four parts including overview of the framework, face detection and alignment, face recognition, face tracking algorithm and summary of the framework.

3.1 Overview of the Proposed Framework

An overview of proposed framework is shown in Figure 3.1, we first applied face detection and alignment, then using the face bounding box from detection as input to recognizes the face. Finally, the adaptive face tracking is applied to track the face in a video.

The main consideration for real-time face recognition application is a processing time as well as the accuracy. In the experiment, face detection consumed a lot of computational time to locate the face in the image. Therefore, we applied adaptive face tracking algorithm to reduce the processing time in each step.

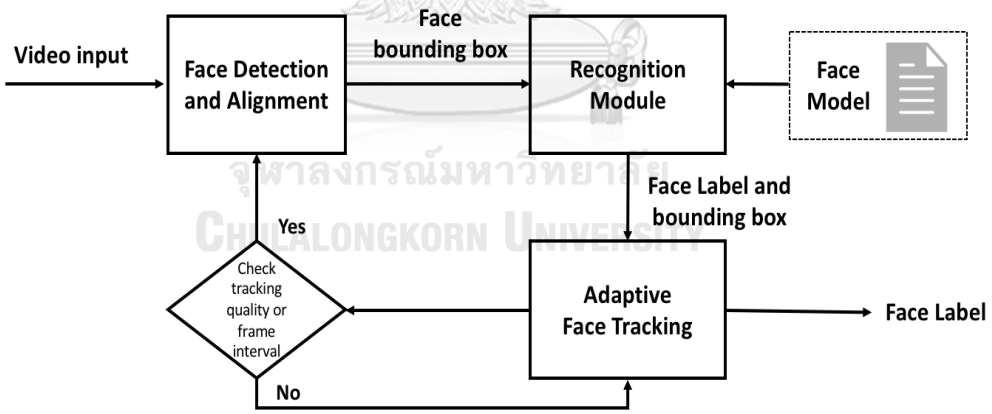


Figure 3.1. Overview of proposed framework.

The proposed framework architecture shows in Figure 3.2. The camera we use CCTV as an input video with the resolution of 640x360 pixels, 25 frames per second (fps) as frame rate and the framework is implemented in the NVIDIA Jetson TX2 board which has ability to process parallelize to recognize multiple faces in real-time.

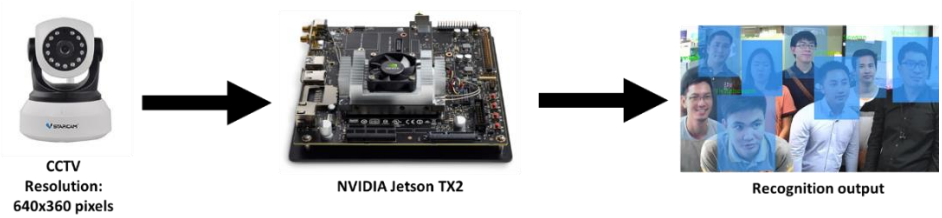


Figure 3.2. Framework architecture.

The detail in each step of the framework are describe in the topic as follows.

3.2 Face Detection and Alignment

As mentioned earlier, face detection is the essential part in the framework as an input to recognition module, we utilized the face detection algorithm based on MTCNN algorithm[7], the algorithm based on CNN which can detect the face in many variation and illumination condition. Moreover, the output of the detection is the bounding box and facial landmark for alignment which increase the recognition accurately.

Given the input image of 640x360 pixels, the algorithm resizes the image into different scales which we called the image pyramid. The scale factor is 0.709 which gives highest precision and recall. Decreasing scale factor allows the detector to scan the image smaller as well as increases the computational time. These different images size is fed to the three stages CNN framework for face detection and alignment. The first stage classification (P-Net) classifies the image whether face or non-face, the bounding box regression from R-Net and 5 facial landmarks localization are extracted from O-Net which is used to align the face. All of the face image from detector is resized to 160x160 pixels before recognition step.

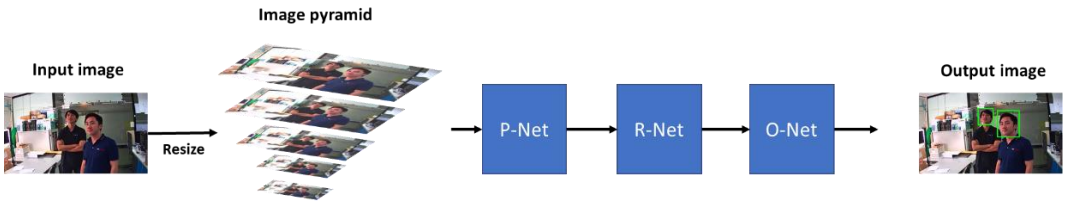


Figure 3.3. face detection and alignment diagram.

3.3 Face Recognition

Most of the face recognition based on CNN is designed for high-performance computing due to the computational cost in the network layers. In the case of

recognizing the face, we need to get high accuracy to tell whether the face or non-face, as well as telling who this face in the image is. The learning algorithm should extract in the deep detail of face components such as eye, nose, forehead, chin and mouth.

In this research, we applied face recognition method based on FaceNet [2], all of the input image size is 160x160 pixels. FaceNet algorithm is based on Inception architecture [18]. The goal is to minimize the embedding of the input image in the feature space of the same person while maximizing the face of different person to be far away. To decrease the parameter of the network as well as decreasing the size of the model in order to save the memory usage of the GPU, we trained the model with SqueezeNet [19] architecture instead. This lead to faster training and face prediction but may have slightly drops in accuracy. Moreover, the model size also decreased.

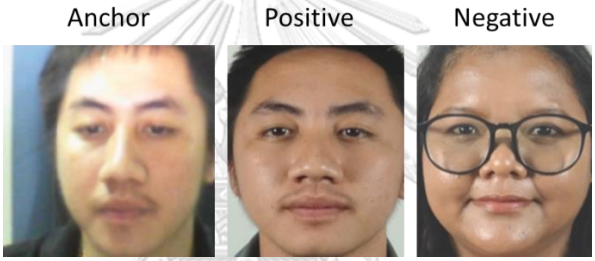


Figure 3.4. Example of anchor, positive and negative face image for training.

In the training, let *N* be the number of face images we have. At iteration *i*, pick an anchor face x_i^a of a specific person which is close to all other positive face images x_i^p of the same person, and a negative example x_i^n, belonging to a different person as shown in Figure 3.4. The embedding of x_i^a, x_i^p and x_i^n are represented as $f(x_i^a)$, $f(x_i^p)$ and $f(x_i^n)$. By using embedding, the feature of face can map and represented as vector in Euclidean space. Base on [2], the distance of these three images can calculate by Eq. (3.1) to ensure that the same person has small distance while the different person has large distance.

$$\left\| f(x_i^a) - f(x_i^p) \right\|_2^2 + \alpha < \left\| f(x_i^a) - f(x_i^n) \right\|_2^2 \tag{3.1}$$

Then minimizing the triplet loss.

$$L_{triplet} = \sum_{i=1}^{N} [\left\| f(x_i^a) - f(x_i^p) \right\|_2^2 - \left\| f(x_i^a) - f(x_i^n) \right\|_2^2 + \alpha] \tag{3.2}$$

where α represents a margin that is enforced between positive and negative pairs.

Support Vector Machine (SVM) classifier is used to classify the person in the feature space. The output of the prediction is the probability distribution of the person in the database which can calculate by (3.3) based on [24]. The maximum prediction is considered as a label of the person if the value greater than the threshold value, otherwise, we considered as unknown person.

$$P(label / input) = \frac{1}{(1+\exp(A* f(input) + B)} \tag{3.3}$$

where *P(label/input)* is the probability that input belongs to label and *f(input)* is a signed distance of the embedding in the feature space, *A* and *B* are the parameters optimizing and getting from the training classifier.

The maximum of the probability is varied by the amount of person in the database. In other word, when increasing the number of person in the database, the maximum probability is decrease. This lead to varying the threshold when the database is increased.

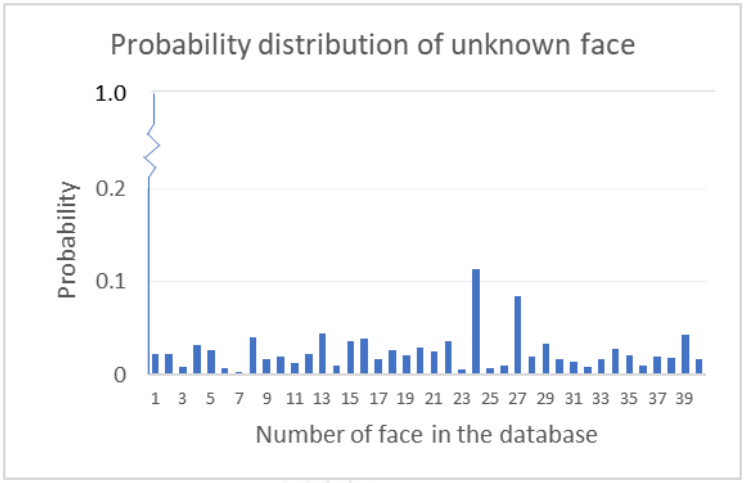
we used Eq. (3.4) to calculate the threshold of the predicting , δ , by the assumption that the probability of the prediction has the ratio difference between the maximum and the others.

$$\delta = C \times \frac{\sum_{1}^{N} x_i - x_{\max}}{N-1} \tag{3.4}$$

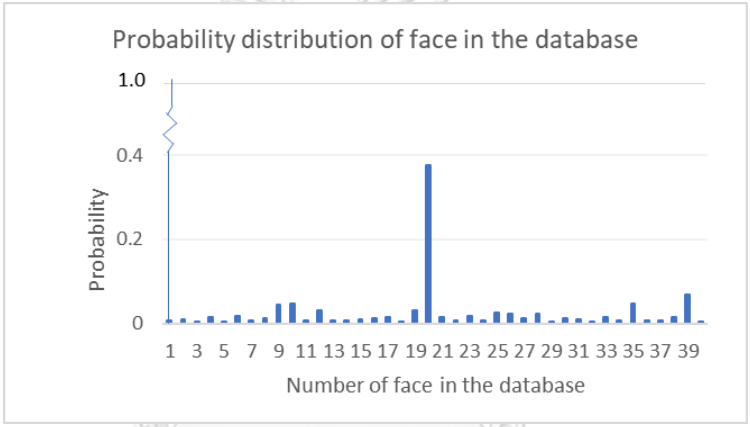
จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

where *C* is the ratio value between maximum probability and the others (*C*=1,2, …, n), x_i denotes the probability of *i* person, x_{max} is the maximum probability of the prediction, and *N* is the total number of person in the database.

Recognition rate is depended on the value of parameter *C* in Eq. (3.4). As show in Figure 3.5, when recognizing the face in database, the maximum probability is totally higher than others compared with recognizing the unknown face. The optimal *C* value, which archive highest recognition rate, is equal to 10.



(a)



(b)

Figure 3.5. (a) Probability distribution of predicting unknown face. (b) Probability distribution of predicting face in the database.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.4 Adaptive Face Tracking

The processing time is a main problem in face recognition system for real-time application due to the processes of each step such as face detection, face alignment and face recognition itself. In the experiment, when we applied face detection algorithm in the whole image of every frame in video, face detection consumes high processing time in each frame. Therefore, we applied algorithm to track the face using correlation filter based on [15]. The algorithm tracked the face from the bounding box that getting from detected face. Then, using frame interval and the quality of tracking to decide to reset the detection.

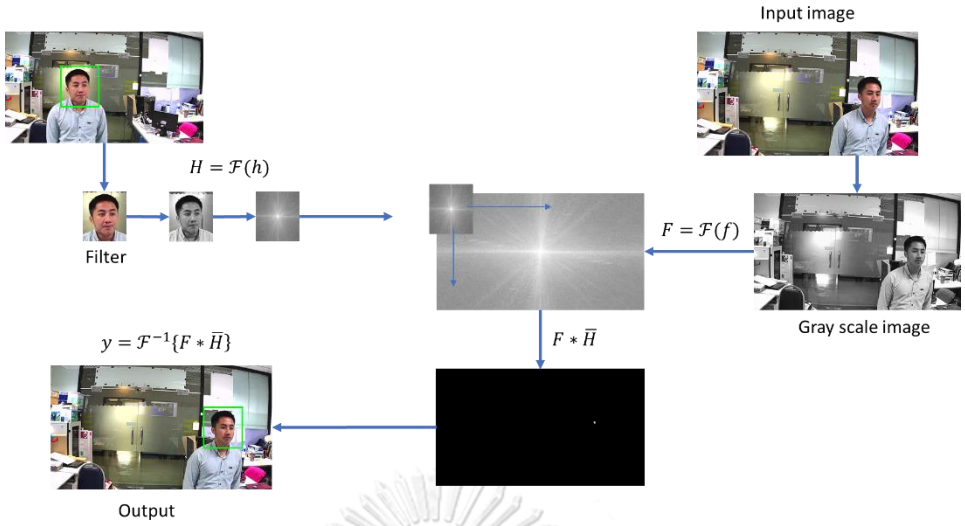


Figure 3.6. Pipeline of face tracking algorithm.

Given an input image *f* in a new frame and filter *h* (the filter is a face image region from the detection), the 2D Fourier transform of the input image $F = \mathcal{F}(f)$ and the filter $H = \mathcal{F}(h)$ are computed. The new target location is estimated to be at the maximum correlation score of y which can compute as follows:

$$y = \mathcal{F}^{-1}\{F * \bar{H}\} \tag{3.5}$$

3.5 Summary of proposed framework

The proposed framework can be summarized as follows.

1. We applied face detection algorithm based on CNN [7] to detect and localize the face in the image.
2. The face is learned and recognized using FaceNet [2]. To reduce the processing time of recognition, we train the model with SqueezeNet [19] architecture which reduce the parameter for feature learning.
3. Applying adaptive tracking algorithm [14] to reduce the processing time of detection and recognition.
4. Implementing the proposed framework into Nvidia Jetson TX2 board. The board can parallelize recognize multiple face and can apply deep CNN within the board.

Chapter 4
Experiment Result and Discussion

In this chapter, we examined the performance of the proposed framework. The detail of dataset we used in the evaluation is described in this chapter as well as the experimental system setup and follow by the experimental results and discussion.

4.1 Experimental System Setup

According to the purpose of this study, we will implement the framework into Nvidia Jetson TX2 board to recognize the face in the surveillance system. Therefore, we will use Closed-Circuit Television (CCTV) camera as an input. The video retrieved from CCTV camera using IP address as a protocol and default resolution is 1280x720. In the case of recognition in this framework we use a resolution of 640x360 which is suitable for real-time recognition. The Nvidia Jetson TX2 board technical specifications are shown in Table 4.1.

Table 4.1. NVidia Jetson TX2 Specifications.

	Details
CPU	ARM Cortex-A57 (quad-core) @ 2GHz + NVIDIA Denver2 (dual-core) @ 2GHz
GPU	256-core Pascal @ 1300MHz
Memory	8GB 128-bit LPDDR4 @ 1866Mhz and 59.7 GB/s of memory bandwidth
Storage	32GB eMMC 5.1
Encoder	4Kp60, 4Kp30, 1080p30
Decoder	4Kp60
Camera	12 lanes MIPI CSI-2 | 2.5 Gb/sec per lane | 1400 megapixels/sec ISP
Display	HDMI 2.0 / DP 1.2 / eDP 1.2 | 2x MIPI DSI
Wireless	802.11a/b/g/n/ac 2×2 867Mbps and Bluetooth 4.1 support

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Ethernet	10/100/1000 BASE-T Ethernet
USB	USB 3.0 + USB 2.0
Misc I/O	UART, SPI, I2C, I2S, GPIOs
Socket	400-pin Samtec board-to-board connector, 50x87mm
Power	7.5W

We developed and implemented the algorithm to recognize the face which can be run in NVidia Jetson TX2 board using Python version 2.7 with OpenCV, sckit-learn, dlib, numpy and TensorFlow libraries.

To extract the face feature in detail, the system need a large computational resource. TensorFlow is an open source system applied for large scale of machine learning processes for deep insight. TensorFlow library is used for numerical computations with the help of data flow graphs which reduces the processing time for object recognition based on deep convolutional neural network.

4.2 Datasets

In this research, we used different datasets for training, validation and testing where each dataset consists of the faces with different variations, illuminations, poses and occlusion. The detail of datasets is described as the following.

4.2.1 Training

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Face recognition need a model for prediction. There are many pre-trained models that provided in public. The model is trained by different datasets and parameters such as learning rate, batch size, epoch size, image size, optimizer method, etc. in order to achieved highest accuracy and reduce the weight of CNN network. If dataset has less data to train, it comes up with underfitting. In contrast, if the dataset contains lots of data and training the data too well, the training model become overfitting. Therefore, the training required a good optimal value of parameter as well as the model architecture.

VGGFace2 dataset [25] has been used for training the network. The dataset contains 3.31 million images of 9,131 subjects with the resolution of 224x224. Images are downloaded from Google Image Search and have large variations in pose, age, illumination, ethnicity and profession as shown in Figure 4.1.

Figure 4.1. Example of VGGFace2 dataset.

4.2.2 Validation

The validation set is used for minimizing overfitting as well as tuning the parameters such as weights and biases to increase the accuracy over the dataset that has not been shown in the training before. LFW dataset [26] has been used for validate the model in this work. The dataset contains 13,233 face images with 5,749 different subjects. The resolution of the images is 250x250 pixels.

จุฬาลงกรณ์มหาวิทยาลัย

Figure 4.2. Example of LFW dataset.

4.2.3 Testing

Test set is the face set that used to measure the performance of face recognition algorithm in term of recognition rate and time to recognize the face. YoutubeFace dataset [27] used for measuring the performance of tracking. SCface dataset [28] and FEI face dataset [29] are used for measuring the recognition rate.

YoutubeFace database is a database of face videos, containing of 3,425 videos of 1,595 different people. We used YoutubeFace database to measure performance of tracking due the dataset contains a people movement within the scene and unconstrained of the background. Whereas, FEI face database used for measuring the recognition rate of the algorithm. The database contains 2,800 images of 200 people. 14 images for one people. All the images are RGB image and rotate up to about 180 degrees in frontal position. The original size of each image is 640x480 pixel. The frontal face image and the label of each person was used for training a classifier in the experiment.

Figure 4.3. Example of FEI dataset.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

SCface is the dataset for testing face recognition algorithm in real-world conditions. The camera setup in different directions and distances. The images of the subject were taken by camera 1-8 with the distance of 4.20, 2.60 and 1.00 meters. The resolutions are 100x75 for distance of 1.00m, 144x108 for distance of 2.60m and 224x168 for distance of 4.20m. These lead to difference of image qualities in each camera view.

Figure 4.4. Example of SCface dataset. The images were taken from three distance that results three resolutions of the image.

To measure the processing time and recognition rate of multiple face recognition, we generated the CUFace dataset. The dataset consists of two parts: training and testing. The training dataset were captured by a camera with the resolution of 4528x3400 containing 255 images of 45 individuals (5 images per person). The testing set were captured by a CCTV camera with a people standing in front of the camera. The resolution is 640x360 pixels. The distance of the people from the camera is between 1.00-1.20 meters. The example of CUFace dataset shown in Figure 4.5.

(a)

(b)

Figure 4.5. (a) Training set and (b) Testing set examples of CUFace dataset.

The datasets we used in the experiment are summarize as follows.

Table 4.2. Summary of the dataset using in the experiment.

Datasets	Classes	Images	Resolutions
VGGFace 2	9,131	3.31 million	224x224
LFW	13,233	5,749	250x250
YouTubeFace	1,595	3,425 (videos)	Multiple resolutions
FEI	200	2,800	640x480
SCFace	130	4,160	Multiple resolutions
CUFace	45	255	4528x3400

4.3 Model and Classifier Training

The model was trained on SqueezeNet architecture with VGGFace2 dataset, the input image size is 224x224 pixels. The parameters for training are initialized as follows.

Table 4.3. Initial parameter using in the model training.

Parameters	Values
Architecture	SqueezeNet
Learning rate	0.1
Training epochs	500
Optimizer	RMSPROP

The model accuracy stable at 98.2% after training about 250 epochs.

To train a classifier, we cropped and aligned the image and resized to 160x160, then utilizing image augmentation technique to generate 40 faces (as show in Figure 4.6) from original face images with many variations such as adding noise, illumination, rotate, shear the image, etc. These 40 images are used as an input to train a classifier. The accuracy of the recognition is varied by the training sample as we will discuss in the results.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

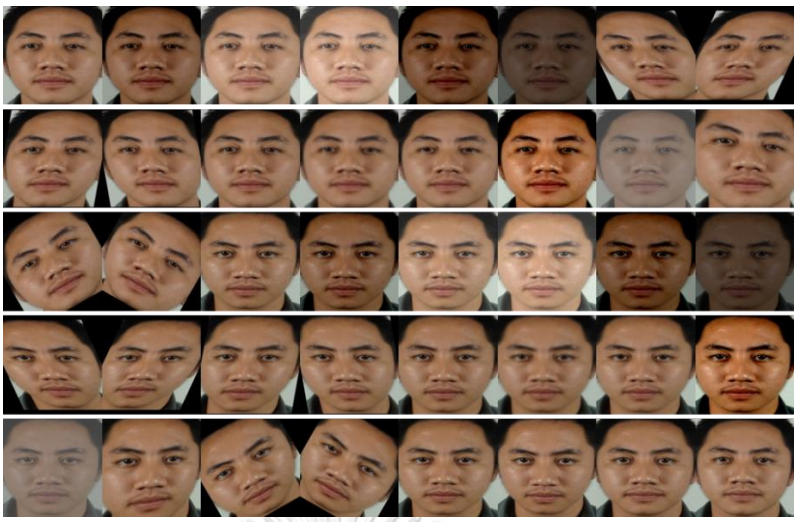


Figure 4.6. Example of 40 faces generated from image augmentation algorithm.

4.4 Evaluation Metrics

We evaluated the performance using Precision, Recall, F-Measure, Recognition rate and computational time as a metrics. Since the dataset provided the ground truth bounding box of the face. We denoted GT as a ground truth and SUT as a system under test or the predicted result. The confusion matrix has been used for considering True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) with the following condition.

- TP: when object in GT presented and SUT correctly predicted.
- FP: when object in GT not presented, but SUT predicted.
- TN: when object in both GT and SUT not presented.
- FN: when object in GT presented, but SUT not predicted.

In the detection algorithm, we consider TP, FP, FN and TN by calculate Intersection over Union (IOU). IOU can calculate by Eq. (4.1), the overlapping of the area in SUT and GT.

$$\text{IOU} = \frac{\text{Area of overlap}}{\text{Area of union}} \tag{4.1}$$

An intersection over union score greater than 0.5 is normally considered as a good detection.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 4.7. Example of IOU in face detection. Green bounding box is a GT and red bounding box is SUT.

4.4.1 Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations, precision can calculate by Eq. (4.2) as follows.

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP} \tag{4.2}$$

4.4.2 Recall

Recall is the proportion of real positive cases that are correctly predicted positive.

$$\text{Recall} = \frac{\#TP}{\#TP + \#FN} \tag{4.3}$$

4.4.3 F-Measure

The average of the precision and recall may not enough to judge the performance of the algorithm. F-Measure is a metric to measure the overall performance of the algorithm which can calculated by Eq. (4.4).

$$\text{F-Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

4.4.4 Recognition Rate (RR)

Recognition rate is the percentage of the correct recognize face image with the total of testing image.

$$\text{Recognition rate} = \frac{\text{the number of recognized images}}{\text{the number of testing images}} \times 100 \tag{4.5}$$

4.4.5 Computational Time

Computational time of the detection and recognition in a framework presented in seconds.

$$\text{Computation time} = \frac{\text{run time for all testing images}}{\text{the number of testing images}} \tag{4.6}$$

4.5 Experimental Results

The experimental results in this work can be divided into three parts: performance of detection and tracking algorithm, recognition rate, and processing time.

4.5.1 Detection and tracking algorithm

To show the tracking algorithm performance, we used YoutubeFace database. The database contains the video clip of people movement. It is difficult to detect the face in very complex background as well as the variation of the face. The tracking algorithm performance is comparable to normal detection while the processing time is faster 61.7% in averaged. The results are shown in Table 4.4.

Table 4.4. Face detection performance in FEI database.

	Precision	**Recall**	**F-Measure**	**Processing time(s)**
Detection	0.97	0.95	0.96	0.094
Detection with tracking	0.90	0.99	0.94	0.036

จุฬาลงกรณ์มหาวิทยาลัย

Figure 4.8. Output of the detection. Green bounding box is GT and red bounding box is SUT.

4.5.2 Recognition algorithm performance

We used the model to train classifier as well as recognizing the face in the experiment. The recognition rate depends on the number of trained face. We test the

algorithm with FEI database. The test set consists of 200 people face images in different direction which is difficult to recognize when the features of face does not appear. The result in Figure 4.9 indicates that 40 trained face images achieves the highest recognition rate. In this experiment, our framework can recognize 200 people with the recognition rate of 90%.

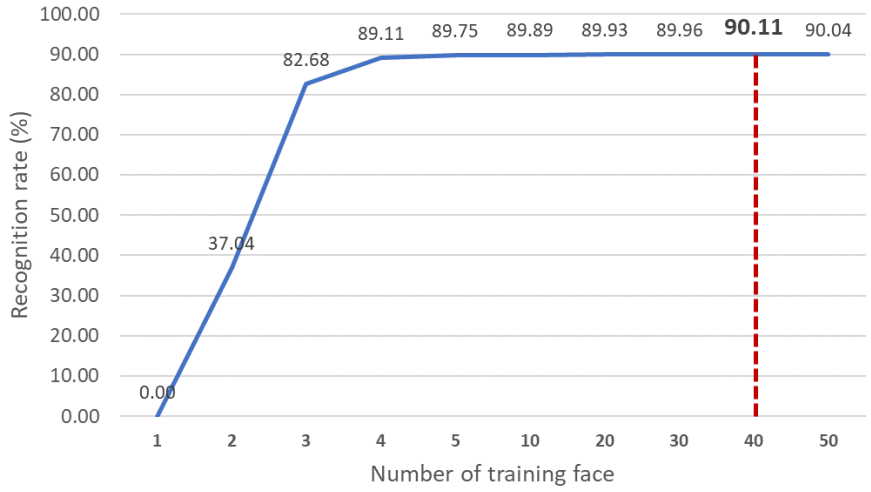


Figure 4.9. A trade-off between recognition rate and number of training face on FEI database.

We tested a framework with SCFace database to measure the recognition rate in real-world condition such as illumination, variation and low-quality images. The database contains day time and night time views in different camera distances. The result in Table 4.5 indicates that the recognition algorithm cannot recognize in low quality of face image as well as in the night time views.

Table 4.5. Recognition rate on SCFace dataset including day time and night time.

Dataset	Correct	Incorrect	Total testing image	RR (%)
SCFace	1,344	1,192	2860	46.99

Therefore, we separated test the algorithm with day time and night time camera. Then comparing the performance to see the trade-off of recognition rate and the distance of object from camera. The recognition algorithm can recognize well on the distance around 1-2 meters and be able to distinct over hundred people with the recognition rate about 80%. The result show as follows.

Table 4.6. Recognition rate on SCFace database on day time camera.

Day time camera	**Recognition rate (%)**		
	Distance 1	Distance 2	Distance 3
Camera 1	31.54	75.38	82.31
Camera 2	17.69	65.38	82.31
Camera 3	20.77	68.46	89.23
Camera 4	14.62	62.31	89.23
Camera 5	10.77	51.54	70.77
Average	19.08	64.62	**82.77**

Table 4.7. Recognition rate on SCFace database on night time camera.

Night time camera	**Recognition rate (%)**		
	Distance 1	Distance 2	Distance 3
Camera 6	1.54	10.77	38.46
Camera 7	1.54	12.31	40.77
Average	1.54	11.54	39.62

Figure 4.10. The output of the recognition algorithm in Day time camera in SCFace database.

Figure 4.11. The output of the recognition algorithm in Night time camera in SCFace database.

To measure recognition rate and the processing time of recognition algorithm of multiple face, we tested the algorithm with CUFace database. The result compared normal recognition and recognition using tracking algorithm which speed up the processing by 56.10% in averaged. The frame rate varied by the number of face in recognition, our algorithm can recognize multiple face around 5-10 fps with recognition rate around 90%.

Table 4.8. Recognition rate on CUFace database.

Number of face	**Processing time (s)**		**Speed up (%)**	**RR(%) of recognition with tracking**
	Recognition	Recognition with tracking		
1	0.19	0.05	75.79	96.00
2	0.23	0.07	69.78	98.67
3	0.25	0.09	63.39	95.11
4	0.28	0.12	57.30	94.50
5	0.32	0.16	50.31	84.73
6	0.35	0.20	44.13	84.11
7	0.39	0.22	43.78	83.67
8	0.41	0.23	44.31	85.50
Average	**0.30**	**0.14**	**56.10**	**90.29**

The outputs of the recognition are show in the Figure 4.12 and Figure 4.13. The bounding box is located the face with the label (name) of the person. Our framework can recognize the face in some variation such as wearing the hat and face expression. We can conclude that the recognition algorithm can recognize the face when a face detector can detect face components such as eyes, eyebrows, nose and mouth.

Figure 4.12. The output of recognition algorithm on CUFace database.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 4.13. The output of recognition algorithm with occlusion on CUFace database.

Chapter 5
Conclusion and Future works

To detect and recognize multiple face using convolutional neural network on embedded computer vision is very challenging due to the computational time of each steps. Conventional face detection and recognition algorithms still have a limitation when detect and recognize face with a many variation, illumination and face expression.

To address this problem, we proposed a framework for multiple face detection and recognition which implement to the embedded computer vision, i,e., Nvidia Jetson TX2 board. The framework consists of face detection algorithm based on CNN, recognition algorithm with the small weight of model which is reducing the parameter of learning to reduce the processing time. Finally, the tracking algorithm is applied to track a face in the video. These lead to reduce the processing time of the framework.

The experimental results demonstrated that the detection algorithm can detect the face in many variation, illumination and face expression. The framework can recognize multiple faces with the frame rate of 5-10 fps and the recognition rate up to 90%.

For future works, the occlusion and the resolution of the face image are a problem for face recognition. Image super-resolution can be applied to small face image to improve the recognition rate when the face is far from the camera. In addition, the training face images can also include more variations to recognize face in the night time view as well as recognizing the face in outdoor camera.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[2] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815-823.

[3] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137-154, 2004.

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

[5] S. S. Farfade, M. J. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015, pp. 643-650: ACM.

[6] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," in *Biometrics (IJCB), 2014 IEEE International Joint Conference on*, 2014, pp. 1-8: IEEE.

[7] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499-1503, 2016.

[8] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867-1874.

[9] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397-403.

[10] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *European Conference on Computer Vision*, 2012, pp. 679-692: Springer.

[11] D. Lee, H. Park, and C. D. Yoo, "Face alignment using cascade gaussian process regression trees," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4204-4212.

[12] A. Jourabloo and X. Liu, "Pose-invariant 3D face alignment," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3694-3702.

[13] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *European Conference on Computer Vision*, 2014, pp. 94-108: Springer.

[14] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference, Nottingham, September 1-5, 2014*, 2014: BMVA Press.

[15] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 2544-2550: IEEE.

CHULALONGKORN UNIVERSITY

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.
[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[18] C. Szegedy *et al.*, "Going deeper with convolutions," 2015: Cvpr.
[19] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size," *arXiv preprint arXiv:1602.07360*, 2016.
[20] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701-1708.
[21] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *arXiv preprint arXiv:1502.00873*, 2015.
[22] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2892-2900.
[23] K. Rungsuptaweekoon, V. Visoottiviseth, and R. Takano, "Evaluating the power efficiency of deep learning inference on embedded GPU systems," in *Information Technology (INCIT), 2017 2nd International Conference on*, 2017, pp. 1-5: IEEE.
[24] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61-74, 1999.
[25] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," *arXiv preprint arXiv:1710.08092*, 2017.
[26] G. B. Huang and E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," *Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep*, pp. 14-003, 2014.
[27] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 529-534: IEEE.
[28] M. Grgic, K. Delac, and S. Grgic, "SCface–surveillance cameras face database," *Multimedia tools and applications*, vol. 51, no. 3, pp. 863-879, 2011.
[29] C. E. Thomaz and G. A. Giraldi, "A new ranking method for principal components analysis and its application to face image analysis," *Image and Vision Computing*, vol. 28, no. 6, pp. 902-913, 2010.

APPENDIX

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

VITA

Savath Saypadith was born in Laos, in 1994. He received his B.Eng. degree in Information Technology from National University of Laos, Vientiane, Laos, in 2015. He was supported by AUN/SEED-Net, JICA scholarship from 2016-2018 to pursue his Master's degree in Electrical Engineering at the Digital Signal Processing Research Laboratory, Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University, Thailand. His research interests include computer vision and machine learning.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY