Personal Verification System Using ID Card and Face Photo

Adulwit Chinapas, Pattarawit Polpinit, Narong Intiruk, and Kanda Runapongsa Saikaew

***Abstract*—Generally, the process of verifying a person's identification in a bank is accomplished by an officer comparing a photo in an ID card with the actual face of the person. This process is prone to mistake as officers usually need to serve several people in a short time. This article proposes the personal verification system using an ID card and face photo by applying face detection and face comparison. A system based on several open source libraries for face recognition including Dlib, Facenet, and ArcFace is implemented. The experimental analysis shows that the system based on ArcFace yields the highest accuracy at 99.06% for face detection and 96.09% for face comparison. ArcFace outperforms other methods because it not only uses MTCNN but also adjusts face image to be in a straight direction as well as fixes the positions of eyebrows, eyes nose, and mouth so that all images have similar references.**

***Index Terms*—Face comparison, face detection.**

I. Introduction

Nowadays, ID cards are used as evidence for identification and are extremely important for several tasks, such as opening a bank account, applying for a credit card account, applying for a job, and other financial transactions. However, there is still counterfeiting of ID cards or use of other ID cards for one own benefit. For example, some foreigners forged ID cards to covert and evade police detentions. Some criminals smuggled ID cards to open bank accounts [1], [2].

Personal verification is commonly used by the ID card reader, which reads from the card chip contained in the ID card, with the card number, name, last name, date of birth, and expiration date. Traditionally, bank officers make a face comparison between the face on an ID card and the owner's face by using eyes. Such a method often incurs a mistake.

Therefore, this article has proposed and developed a system that automatically compares between a face on an ID card photo and a face photo to accelerate and increase the accuracy of the ID card verification procedure. Some commercial software, such as the Jumio program [2], can verify identity with face photo and ID card photo but the application is expensive, i.e., two US dollars per one verification.

The first step of personal verification using face comparison is face detection which has been studied for several years. There are a variety of approaches for face detection. The most popular methods are the Haar Classifier [3] and CNN [4] because Haar Classifier is easy to implement and CNN is quite accurate. Several widely used libraries that employ a Haar classifier include OpenCV and Dlib. On the other hand, Facenet and ArcFace apply MTCNN [5] which is a type of CNN. In this article, MTCNN was chosen because it had the best performance for face detection and several models of MTCNN were available.

After the face detection is completed, face comparison can proceed. The approaches which include Dlib, Facenet, and ArcFace are widely accepted. We chose ArcFace because it yielded 99.80% accuracy on LFW which was higher than the accuracy of Dlib (99.38%) and the accuracy of Facenet (99.68%).

The rest of the article is organized as follows: Section II describes the background of this study, Section III explains the methodology in face comparison using ID card and face photo, and the experimental results are presented and analyzed in Section IV. Finally, the conclusions are drawn in Section V.

II. Background

A. ID Card

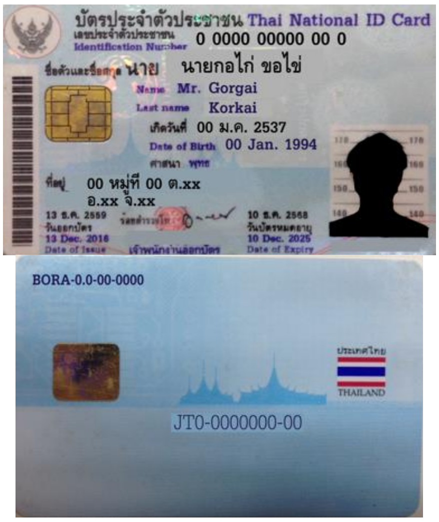


Fig. 1. Example for front and back of one ID card.

A Thai ID card [6] is an official document issued for Thai people who are in only house registration. It is required evidence to use for important documents such as driver's license and credit card. An ID card contains a face image of the cardholder without eyeglasses and a hat. The face image

Manuscript received August 25, 2018; revised April 1, 2019.

This research was supported and funded by T2P Co., Ltd., and the Faculty of Engineering, Khon Kaen University, Thailand.

Adulwit Chinapas, Pattarawit Polpinit, and Kanda Runapongsa Saikaew are with the Faculty of Engineering, Khon Kaen University, Thailand (Corresponding author; e-mail: krunapon@kku.ac.th).

doi: 10.18178/ijmlc.2019.9.4.818

background consists of lines with height. The card number has 13 digits is unique and have the card holder's first name, last name, birth date, address, issue and expiry dates as shown in Fig. 1.

B. Face Data Set

LFW (Labeled Faces in the Wild) [7] is a facial image database designed to study facial recognition problems. There are more than 13,000 facial images collected by the web. Each face has a personalized label with a total of 1,680 people. For one person, there are two or more distinct photos. This database is a challenge in face recognition competition.

FaceScrub [8] has faced datasets with over 100,000 images from 530 people. Large face datasets are essential for advanced face recognition research. The FaceSync database is built using this method, followed by self-validation and cleanup. It consists of a total of 141,130 images of 348 male and 347 female celebrities, with approximately 200 pictures per person, making it one of the most extensive public databases. These images were searched from the Internet and were under real-life conditions.

VGG [9] contains 2,622 faces. Each credential includes an associated text file that contains the related image and faces detection links.

MegaFace [10] having more than 4 million faces is considered to be desirable for face detection and facial recognition.

VGG2 [11] has more than 9000 identities spanning a wide range of different ethnicities, accents, professions and ages and more than 3.3 million faces that are captured by "In the wild," with pose and emotion variations and different lighting and occlusion conditions.

MS1M [12] contains 10 million images for 100k celebrities, and it is the largest publicly available image database one in the world.

LFW has been experimented in 113 articles that several algorithms achieved its accuracy of 100%. Therefore, this article chooses to employ open source libraries that are based on LFW with the accuracy of more than 99%. These libraries are Dlib, Facnet, and ArcFace (see Table I).

TABLE I: FACE DATASET USED FOR THIS PAPER

Data set name	# of images	# of people	age	year
LFW	13,000	1,680	-	2007
YTF	3,425 videos	-	-	2011
FaceScrub	141,130	695	-	2014
VGG	2.6 M	2,622	-	2015
MegaFace	4.7 M	690 K	Yes	2016
MS1M	10 M	100 K	Yes	2016
VGG2	3.3 M	9131	Yes	2017

C. Face Detection

Face detection is the process that detects the area of a person's face using a variety of techniques, such as Neural Network [13] which is a technique that produces high accuracy results but processes a long time on the other hand, the technique color model [14] is a popular technique but easily disturbed by light. Geometric Facial Template [15] is a fast processing technique, but the facial search is still limited.

Feature extraction [16] is used for the recognition process. The facial features of the facial techniques are Generic method, Feature template-based method, and Structured matching method.

The generic method is based on the basic features of the face for face extraction, including edge, lines, and curves of the face.

Feature template-based method is to apply the facial features template to the face feature in the image to detect important facial features such as eyes, nose, and mouth.

Structured matching method is considered a set of geometric shapes on the face, such as the width and length of the nose, the position of the mouth, and the shape of the chin.

Haar Classifier [17] is calculated by subtracting the value of the pixels white mask pixels from the value of the black mask. The characteristics of binary masks are in various forms. For example, face detection in the middle form is suitable for detecting the cheeks which are brighter than the eyes. Likewise, the mask in the right shape is suitable for detecting the nose, which is brighter than the eyes as shown in Fig. 2.

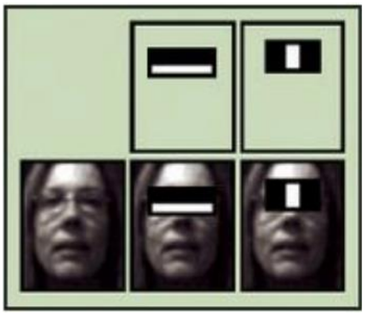


Fig. 2. Example of Haar Classifier for face detection.

MTCNN works in three stages and uses artificial neural networks for the network. The first part is a network of proposals that will serve potential positions and its bounding boxes in interest networks in R-CNN. The second detection and alignment use the image and the results of the first prediction. Reduce false positives and final box boundaries, refine forecasts and add face prediction.

D. Recognition

Template matching [18] is a method of calculating the similarity between the template and the image considered. Use the moving the template to another point in the image then find the correlation coefficient between the template and the initial image. The downside is that it takes a long time to compute for a large template. This method is ideal for finding features that are not larger than templates.

The statistical approach [19] is a method relying on the statistical data obtained from the feature extraction of the image. The image is taken into the recognition process to determine the characteristics of each image based on the decision of the statistical principle to assess whether the data is the required information. For more accuracy, the user can take the results with another recognition process.

E. Face Recognition

Face recognition compares the detected face with the human faces in a database.

Dlib [20] is the software developed based on machine learning in C++. Dlib model is composed of a ResNet-34 network with 29 conv layers trained on a dataset of about 3 million, such as the face scrub, and VGG dataset faces make accuracy 99.38% on LFW.

Facenet [10] is the state-of-the-art which Google researchers show face recognition on the LFW and YTF benchmarks. Facenet applies deep convolutional networks and yields the accuracy of 99.68% on LFW and 95.12% on YTF. OpenFace [11] is an implementation of the FaceNet for ease of use. It has 92.92% accuracy on LFW.

ArcFace [12] refines the most extensive publicly available training data, MS-Celeb-1M, in both automatic and manual way. Taking VGG2 as the training data, conduct extensive contrast experiments regarding the convolutional network settings and report the verification accuracy on LFW (and 99.80%). Propose a new loss function, additive angular margin (ArcFace), to learn highly discriminative features for robust face recognition. Achieves state-of-the-art results on the MegaFace Challenge, which is the most significant public face benchmark with one million faces for recognition.

F. Image pre-Processing

Image Processing [13] is the processing of images into the computer to identify pictures without the decision of the people. The calculation consists of a variety of methods, such as the calculation of the color of each point (Pixel) and the estimation of several areas (Area).

OpenCV [14] is a library for image processing and computer vision. It has several desirable features, such as being able to detect the edge of the picture, motion detection and image segmentation (Image segmentation). It is a non-programmatic set of instructions; thus, it requires a program to execute those commands. Commonly used languages are C, C ++, and Python. OpenCV consists of two parts: the data structure, which is used to store information, such as an image, a matrix, and coordinates. The ability of OpenCV to handle image pre-processing tasks including the perspective transformation which makes ID cards and faces straight for easier face detection handle.

Feature Matching [15] is a comparison of two images. The first is an original picture. The second picture is the one to find whether it is matched with the first one. It detects some identical picture frames from the original and crops only part in a frame by the perspective transformation [16].

III. Methodology

A. Data Gathering

We gathered face photos and ID card photos from 774 students at Khon Kaen University. Therefore, the number of face photos and ID card photos for data were 774 pairs. A sample pair of face photo and ID card is shown in Fig. 3.

B. System Design

In Fig. 4, the system flow starts with the ID card detection. Then face detection is executed before face comparison which is the final step.

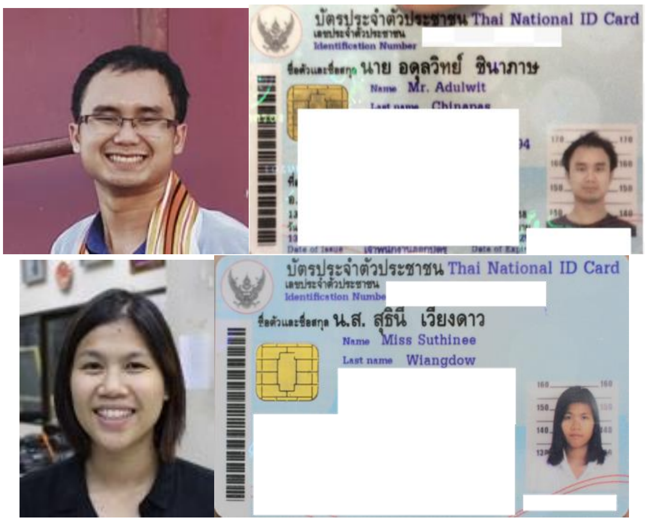


Fig. 3. Examples of matching between face photo and ID card for data gathering.

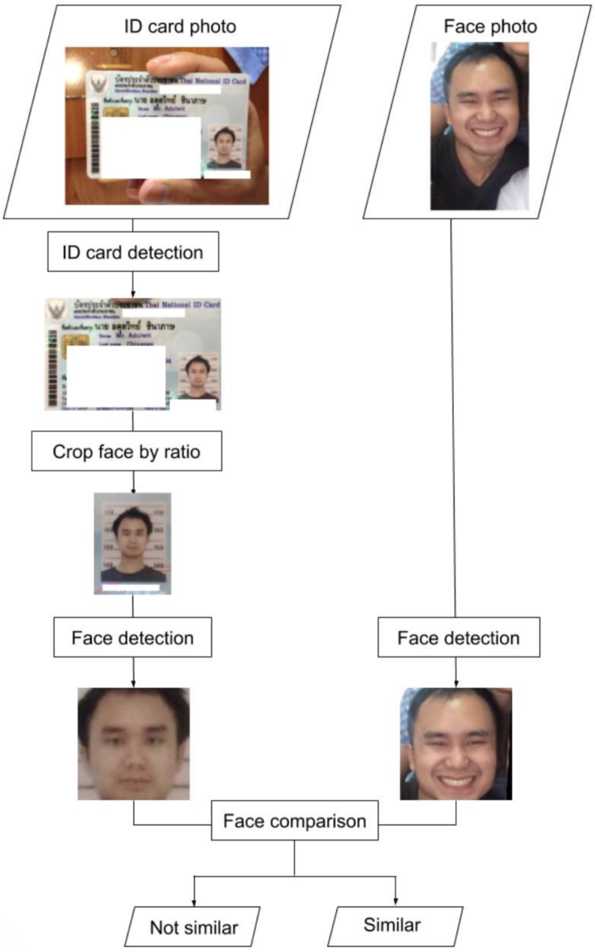


Fig. 4. Flowchart of personal verification system using ID card and face photo.

1) ID card detection: The ID card photo sizes are not equal. To start detecting face, every image is adjusted to the same scale. The tool to assist this is Feature Matching [21] in OpenCV [22] which is presented in Fig. 5. The green lines connect similar features from the first image to the second image. ID card is surrounded by a white frame and cropped by perspective transformation. Performance of ID card detection is 100% for data gathering.
2) Cropping face by the ratio: Since several ID card images have different resolutions, the numbers of pixels for the

cards with the same size are different. Therefore, we need to find the ratio of a cropped face and an ID card. Cropping face by the ratio needs to measure the width and the height of the ID card image and face image. For example, in Fig. 6, ID card image has width as (a) 1243px and height as (b) 780px. On the other hand, the face image has distance between the left edge of ID card image and that of the face image is (c) 920px. Likewise, the distance between the upper side of the ID card image and that of the face image is (d) 340px. Therefore, ratio left = 920 /1243= 0.74 and upper 340/780 = 0.43. Thus, for all ID card images, we use the ratio of the cropped face and an ID card image as 0.74 (left) and 0.43 (upper) in Fig. 7.

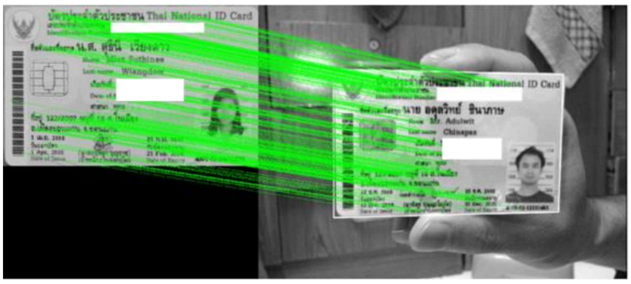


Fig. 5. Example for feature matching of one ID card to another.

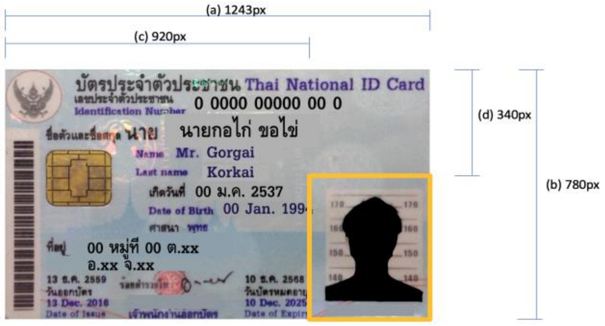


Fig. 6. Example finding ratio for cropping the face.

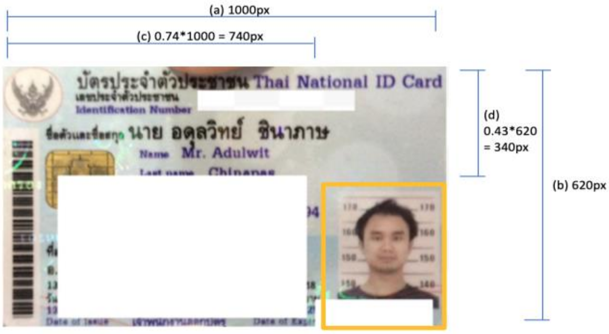


Fig. 7. Example result of cropping face by the ratio.

3) The Face detection: Arcface applies MTCNN detector for face detection and alignment model. The thresholds for the three stages used are 0.6, 0.7, and 0.8 (see Table II).

4) Face comparison: Arcface employs LResNet 100E-IR model that uses ResNet100 as a backend with modified input and output layers. We determine whether two faces are similar or not by using a threshold. If the distance value is less than a threshold, that means two faces are similar (see Table III). On the other hand, if the distance is greater than a threshold, that means two faces are different (see Table IV). Based on our experiments, one of the threshold values that result in an optimal performance is 1.4488.

TABLE II: EXAMPLES FOR FACE DETECTION BY ARCFACE

people	photo	Face detection
People 1		
People 2		

TABLE III: EXAMPLE FOR FACE COMPARISON OF THE SAME PEOPLE

Face	Face-id-card	distance value
		0.572750
		0.740060

TABLE IV: EXAMPLE FOR FACE COMPARISON OF DIFFERENT PEOPLE

Face	Face-id-card	distance value
		1.831967
		1.865852

IV. EXPERIMENTAL RESULTS

A. Performance of Face Detection

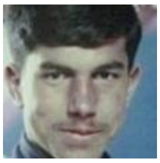
The accuracy for Dlib, Facenet, and ArcFace is 100% on faces dataset. On ID cards dataset, ArcFace and Facenet have the same accuracy of 99.06% which is higher than that of Dlib (93.66%) because Dlib uses Haar Classifier while ArcFace and Facenet employ MTCNN. The result of face detection on data gathering is presented in Table V.

TABLE V: RESULTS FOR FACE DETECTION BETWEEN DLIB, FACENET, AND ARCFACE

Method	Faces dataset	ID cards dataset
Dlib	100%	93.66%
Facenet	100%	99.06%
ArcFace	100%	99.06%

Although both ArcFace and Facenet apply MTCNN, ArcFace adjusts a face image by rotating an image to a straight face. As a result, the positions of eyebrows, eyes, nose, and mouth in different images are nearly the same which increases the effectiveness of face comparison as shown in Table VI.

TABLE VI: DIFFERENCE FOR FACE DETECTION BETWEEN DLIB, FACENET, AND ARCFACE

Original	Dlib	Facenet	ArcFace
			

For the scratched and reflected face-id-card images, ArcFace outperforms other methods. However, ArcFace cannot detect the pictures with the high intensity of light reflection as shown in Fig. 8.

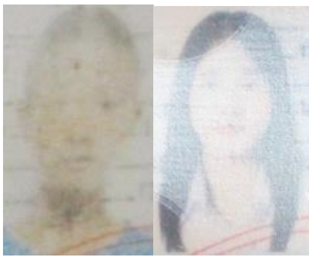


Fig. 8. Example of scratched and reflected face-id-card images.

B. *Performance of Face Comparison*

Table VII illustrates the result of face comparison on data gathering. ArcFace yields the highest accuracy because straightening face image minimizes errors.

Although ArcFace provides the most accurate face comparison, its processing time is much higher than other methods because it rotates an image to the straight face after detected but the other not. Table VIII compares the processing times of all approaches.

TABLE VII: ACCURACY FOR FACE COMPARISON BETWEEN DLIB, FACENET, AND ARCFACE

Method	Same people	Different people
Dlib	84.48%	84.86%
Facenet	85.90%	85.69%
ArcFace	96.09%	96.09%

TABLE VIII: PROCESSING TIME FOR FACE COMPARISON BETWEEN DLIB, FACENET, AND ARCFACE

Method	Time (sec)
Dlib	0.13
Facenet	0.41
ArcFace	1.29

One of the critical aspects of face comparison is the age difference between photos which results in the greater distance value as illustrated in Fig. 9. To adjust to the significant age difference, one may increase the threshold value. However, that would result in lower accuracy for two photos that are different. Thus, one of the remaining challenging problems is determining an appropriate threshold value that results in the optimal performance for both the case in which the faces are similar and the case in which the faces are different.

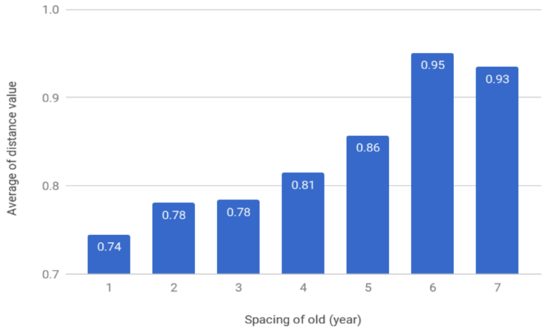


Fig. 9. Greater distance values with the increased age difference.

V. CONCLUSION

This article studies and implements the system with face detection and face comparison using three widely accepted methods which are Dlib, Facenet, and ArcFace. Our experimental analysis shows that ArcFace provides the most optimal solution with the highest accuracy of 96% because the straight face of ArcFace can compare prominent features on the face better than the other methods. Besides, ArcFace detects the face on the scratched and reflective image better than other methods. For future work, face comparison with the consideration of face aging is interesting as several ID cards are valid for a certain number of years. A period of more than one year is a sufficiently long time for someone's face to change, and hence the error of comparing faces can occur.

ACKNOWLEDGMENT

The authors would like to thank the Office of Management and Academic Development of Khon Kaen University for allowing us to collect data.

REFERENCES

[1] P. Banchanont. (2018). How to use information on front ID card. [Online]. Available: https://thematter.co/quick-bite/thai-id-card/49810

[2] Trusted ID and Identity Verification Solutions | Jumio. [Online]. Available: https://www.jumio.com/

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Comput. Vis. Pattern Recognit.*, vol. 1, pp. 511-518, 2001.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, issue 6, pp. 84-90, June 2017.

[5] K. Zhang, Z. Zhang, Z. Li *et al*., "Joint face detection and alignment using multi-task cascaded convolutional networks," *Spl*, no. 1, pp. 1–5, 2016.

[6] S. Sirasang, *The Exploitation of the Identity of Others by Fraud*, p. 2552.

[7] E. Learned-Miller, G. B. Huang, A. R. Chowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," *Advances in Face Detection and Facial Image Analysis*, Cham: Springer International Publishing, 2016, pp. 189-248.

[8] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *Proc. 2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 343-347.

[9] VGG Face Dataset. [Online]. Available: http://www.robots.ox.ac.uk/~vgg/data/vgg_face/

[10] FaceScrub. [Online]. Available: http://vintage.winklerbros.net/facescrub.html

[11] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, *VGGFace2: A Dataset for Recognising Faces Across Pose and Age*.

[12] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-Celeb-1M: A dataset and benchmark for large-scale face recognition," in *Proc. European Conference on Computer Vision*, Jul. 2016.

[13] C. N. R. Kumar and A. Bindu, "An efficient skin illumination compensation model for efficient face detection," in *Proc. IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, 2006, pp. 3444-3449.

[14] M. A. Berbar, H. M. Kelash, and A. A. Kandeel, "Faces and facial features detection in color images," in *Proc. Geometric Modeling and Imaging--New Trends (GMAI'06)*, pp. 209-214.

[15] V. Starovoitov and D. Samal, *A Geometric Approach to Face Recognition*, 1999.

[16] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399-458, 2003.

[17] I. But, L. B. Patterns, and B. Also, *Cascade Classifier Training*, pp. 1-9, 2017.

[18] OpenCV: Template Matching. [Online]. Available: https://docs.opencv.org/3.4/d4/dc6/tutorial_py_template_matching.html

[19] C. A. Rohde, "Introductory statistical inference with the likelihood function," *Introd. Stat. Inference with Likelihood Funct.*, pp. 1-332, 2014.

[20] dlib C++ Library. [Online]. Available: http://dlib.net/

[21] OpenCV: Feature Matching. [Online]. Available: https://docs.opencv.org/3.3.0/dc/dc3/tutorial_py_matcher.html

[22] OpenCV library. [Online]. Available: https://opencv.org/

Adulwit Chinapas was born in Ubon Ratchathani, Thailand, in 1994. He is with Khon Kaen University, Thailand. He received the B.S. degree in Computer Engineering from Khon Kaen University, Thailand, in 2017.

Pattarawit Polpinit received a bachelor degree in Mathematics from Cornell university, USA in 2001, a master degree in Computer science from Asian Institute of Technology, Thailand in 2003, and Ph.D. in Computer Science from University of Liverpool technology in 2009. He is currently an assistant professor at the Department of Computer Engineering, Khon Kaen University. His current research interests are the analysis of algorithm, image processing, and machine learning.

Narong Intiruk was received the B.Eng. degree in Computer Engineering from Khon Kaen University, Thailand, in 2014, and the M.Eng. degree in Computer Engineering from Chulalongkorn University, Thailand, in 2015. In 2015, he joined the National Electronics and Computer Technology Center of Thailand as a research assistant. In 2016, he joined T2P Co., Ltd. as the head of data scientists. He is currently working on deep learning, both research and commercial, to create a real-world solution using AI.

Kanda Runapongsa Saikaew received the B.S. degree in Electrical and Computer Engineering from Carnegie Mellon University, Pennsylvania, USA, in 1997, and the M.S. and Ph.D. degrees in Computer Science and Engineering from the University of Michigan at Ann Arbor, in 1999 and 2003, respectively. In 2003, she joined the Department of Computer Engineering, Khon Kaen University, as a Lecturer, and became an Associate Professor in 2015. Her current research interests include machine learning and social network analysis.