

Realtime Face Recognition for Intelligent Door System

Huong-Giang Doan¹, Ngoc-Trung Nguyen²

¹Faculty of Control and Automation, Electric Power University, Viet Nam, giangdth@epu.edu.vn

²Department of Research Management and International Cooperation, Electric Power University, Viet Nam, trungnn@epu.edu.vn

ABSTRACT

This paper focus on in-deep analysis of the majority of both hand-craft based and CNN-based that proposed in states-of-the-art for face detection and face recognition method. This issue has been researched for recent decades as it enjoys many practical applications such as entertainments, security, tracking, human-machine interaction and so on. However, this problem has been faced to many challenges such as complex background, illumination, face pose, states of face, fake face. Which lead to high time cost, and low accuracy system. In recent times, thanks to the cutting-edge technique that give impressive performance on both hand craft-based and deep learning-based method. There have been improvement in detecting face exploiting feature vectors. Therefore, in this research, we comparing face detection model such as Haar Cascade[6], Dlib[19] and YOLO [27]. Then, we consider and compare affects of methods when extract feature vectors. Besides, we try to examine on various classifications to obtain the suitable system. Evaluation results confirm that the best accuracy rate achieves at 97.83%. The proposed method suggests a feasible solution addressing technical issues in using hand craft-based face recognition. Moreover, we deploy a realtime intelligent door system that utilize face recognition results to automatically control door.

Key words: Face detection, Face representation, Face Recognition, Convolution Neuron Network, Deep Learning, Machine Learning.

1. INTRODUCTION

Face recognition has been become attractive field in computer vision because of its wide applications such as identification[1], re-identification[2], control and automation [3], security([4], [30]), surveillance [5] and so on. Thus, there are many researches have concentrated this problem. Face recognition system consists some main steps: face detection([6], [26], [12]), face representation([7], [8], [9]) and recognition([22], [23], [24], [25]).

Firstly, face detection aims to detect face region in full images. In addition, for some specify applications, it only require one and/or some faces (not all). Many face detection methods proposed as Haar Cascade ([6], [7], [20]), Dlib [19], [21] or deep learning[27]. Author in ([10], [11]) try improve Dlib with skin color of face. These methods is independent development([13], [17]) for each system that has been not

contemporary considered. Therefore, the first step need evaluate and compare on not only time cost but also face detection accuracy in the same performance and dataset. This work is necessary in order to choose the best suitable face detection model.

Secondly, face representation could be divided into two groups: hand craft-based methods([8], [9]) and deep learning-based methods([14], [16], [18]).In the first trend, special points on face are characteristic, fixed and distinguish between humans. Therefore, many approaches have deployed to find face's main points. In [19], Dlib landmark exactly shows 68 points for frontal face. This tool have made stimulation for many face representation methods([8], [9]). Authors in [9]used coordinates of 7 points in center of face, a geometric invariance including area ratio and angle are utilized to present face which feature vector is too simple. This method only used inner points of face and ignore contour bounding around face. In [14], authors used CNN to deploy end-to-end face recognition system. In this paper, we propose face's feature vector from landmark points of Dlib [21]that combines both coordinate of landmark points and angles of some special points. In addition, we also transfer learning for pretrained CNN model with Resnet50[18] and MobileNet_V2[29].Then these CNN model is used to extract face feature. Moreover, we utilize traditional classifiers as Knn[26], SVM[22], Random Forest[23], Decision Tree[25] and Naïve Bayes [24] to examine the suitable classifier for the extracted features.

Finally, some applications are deployed using face recognition ([3], [4], [12], [15]) with special task such as recognition actors of a film in [12] or face modelling in [15]. In this paper, we would like present an intelligent door system that face recognition results is decoding role. This controlling system requires low time cost, avoid fake as well as high accuracy. The previous considerations aims to choice the best suitable face recognition system with trade of between high accuracy, low time cost and easy to deploy. In this work, we use an arduino card to transfer commands from computer to contactors that will control motors of door. A complete intelligent door is experienced by end-users.

The remaining of this paper is organized as follows: Section 2 describes our proposed approach. The experiments and results are analyzed in Section 3. Section 4 concludes this paper and proposes some future works.

2. PROPOSED METHOD

In this research, we propose a framework for face recognition as illustrated in Figure 1. It composes two main parts: (1) Training stream and (2) Testing phase. In both flows, it composes the following main cascade steps:

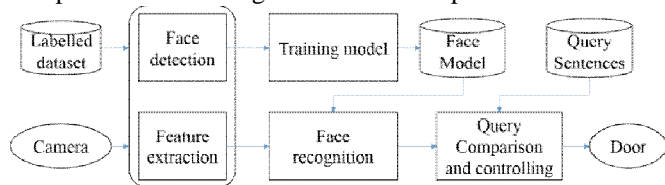


Figure 1: Propose hand posture framework using multi-modalities

- *Face detection*: This step consists of different sub-components such as capture data, detect face from image that remain face region. Then, cropped face images become input of face extractors. Three impressive detection method (HaarCascade and Dlib and Yolo) is exanimated.

- *Face extraction*: In this work, two streams is deployed. First, hand craft-based extractors are used (using Landmarks points). Second. Other stream is deep learning extractor with Resnet50 and Inception128.

- *Face classification*: This component classifies extracted features in the previous components to produce the label of face.

- *Query comparison*: in this task, many requirements sentences is stored in database. For beginning, three random commands are showed for end-users. This work aims to avoid falsification (using image and/or video).

- *Intelligent door using face recognition*: The face recognition results is utilized to control door system.

In the following, we will explain in more detail each component of the proposed framework.

2.1 Face detection

Face detection aims to identify locations of faces in digital image. This is importance step before recognize because it could be remove redundant information or noise cues. This work help reduce time cost as well as increase accuracy of entire system. For face detection, many state-of-the-art techniques have been proposed. In this paper, we will consider some main approaches such as Haar Like Cascade[6], Dlib[19] and YOLO [27]. Which is presented detail in following sections:

2.1.1 Haar Cascade for face detection

Haar Cascade is an object detection algorithm in machine learning. This algorithm was proposed by [6] that is composed four stages: Haar features selection, Integral image creating, Adaboost training and Cascade classifiers.

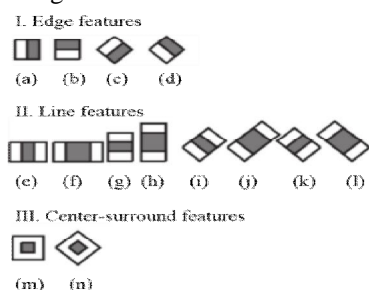


Figure 2: Haar-like features

- *Haar features* is kernel that used to detect objects in images (as showed in Figure 2). Given common features that it find on most common human faces: (1) a dark eye region compared to upper-cheeks; (2) a bright nose region compared to the eyes; (3) some specific location of eyes, mouth, nose... The feature value is simply computed by summing the pixels in the black area and subtracting the pixels in the white area.

- *Internal image creating*: kernels seem as a windows pass on an image. This window works on adjacent rectangular regions at a specific location in an image. It sums up the pixel intensities in each region and calculates the difference between these sums.

- *Adaboost training*: many calculated features are irrelevant. This stage aims to select the best features which includes both selects the best features and trains the classifiers.

- *Cascade Classifier*: This algorithm is a “strong” classifier that is constructed by “weak” classifiers. It utilizes a linear combination of weighted simple of “weak” classifiers.

In this paper, we used pre-train Haar cascade model that is trained through all above four steps and used a large face dataset. This xml file is published at [20]. Results of this approach are evaluated in detail at Section4.

2.1.2 Dlib tool

Dlib [21] is a hand craft-based modern that is written by C++ toolkit containing machine learning algorithms and tools. The library is originally written in C++ but it is easy to use Python bindings. This tool provide creating complex software to solve real world problems. This detector is based on histogram of oriented gradients (HOG) and linear SVM. Dlib is open source licensing. Thus, in this paper, we have focused on using Dlib [21] tool for faces detection and facial landmarks detection from images and video.

2.1.3 Yolo network

YOLO [27] (You Only Look Once) is one state-of-the-art network that is realtime and high accuracy for object detection. YOLO divides images into SxS grid with bounding box contain five elements: x, y, w, h and confidence score. While (x,y) is coordinate of top corner of bounding box, w and h are width and height of box. There have been five YOLO versions (from V1 to V5) at present. YOLO used Darknet framework and pretrained model on ImagesNet-1000 dataset. Differ from YOLO_V1 and YOLO_V2, YOLO_V3 uses Logistic Regression for bounding boxes classification, Logistic Classifier replaces for soft max layer of class predictions, Feature Pyramid Networks(FPN) and Darknet53. Thus, YOLO_V3 is more efficient than previous versions. In [28], YOLO_V3 model is transferred learning for face detection. In this paper, we applied this model to consider and compare with hand craft-based face detection methods.

2.1.4 Face detection implementation

In our context, (1) we used face images to control main door, therefore, face is opposite and frontal with camera; (2) only one person could be control door if they are nearest with camera. Therefore, given image I, pass I through Haar, Dlib or YOLO detector, faces is detected as the following (1) equation:

$$I_{faces} = f_{detector}(I) \text{ (detector = [Haar, Dlib, Yolo])} \quad (1)$$

Then, face is remained by the following (2) equation:

$$I_{face} = \text{Max area}(I_{faces})(2)$$

In this research, we utilize three different detectors to detect face region. We then examine both accuracy and time cost for this step. The best result will be used for next step as well as deploy our real application.

2.2 Face representation

2.2.1 Handcraft-based face representation

Given face region I_{face} , The 68 facial coordinate points ($P_i(x_i, y_i), i=[0 \div 67]$) is calculated as illustrated in Figure 3 following:

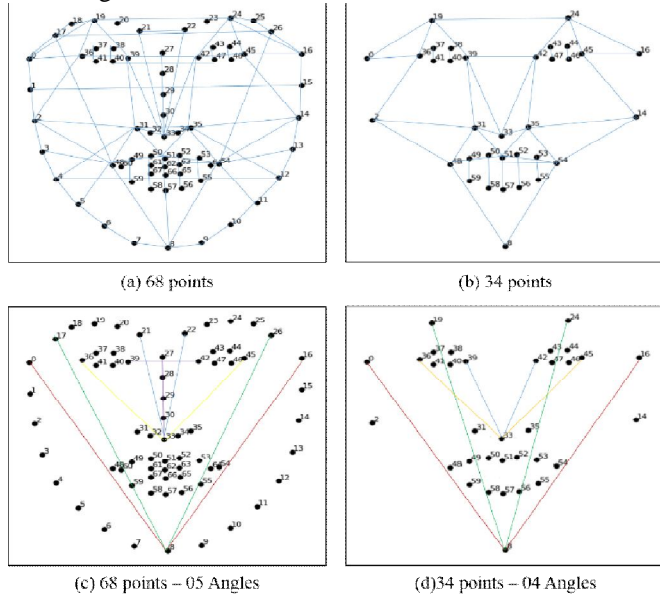


Figure 3:Landmark points of face.

While Jaw via: (P_0, P_{17}); Left eyebrows: (P_{17}, P_{21}); Right eyebrows: (P_{22}, P_{26}); Nose: (P_{27}, P_{35}); Left eyes: (P_{36}, P_{41}); Right eyes: (P_{42}, P_{45}) and Mouth is (P_{48}, P_{67}). Combination strategies of 68 points is important because it must present relation of main points in order to have the highest accuracy as well as obtain the lowest time cost. In this paper, we try to implement two methods: (1) remain 34 important points; (2) use 68 points. There are 04 models that is presented in detail at following sections:

- *The 34 points model:* In this model, we remain 34 points as Figure 3 (b). We firstly calculate the Euclid distances ($d_j; j=[0 \div 33]$). Distances is illustrated as blue lines connect pairwise between two points as the following (3) equation:

$$d_j = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}; (i = [0 \div N]) \quad (3)$$

Moreover, this representation does not contain absolutely positions between points. This distances are not efficient for various user's positions. For 34 points model, we divide all distances d_j with P_{39-42} (distance between points P_{39} and P_{42}) as the following (4) equation:

$$R_j = \frac{d_j}{d_{39-42}}; (j = [0 \div 39]) \quad (4)$$

All ratio of distances is append to create feature vector of the 34 points model as the following (5) equation:

$$F_{34points} = [R_0; R_1; R_2; \dots; R_{39}] \quad (5)$$

The 68 points model: corresponding to the 34 points model, although we use all 68 points but 80 Euclid distances ($d_j; j = [0 \div 79]$) are calculated as showed in Figure 3 (a). Then, we divide all distances d_j with P_{27-30} (distance between points P_{27} and P_{30}) as the following (6) equation:

$$R_j = \frac{d_j}{d_{27-30}}; (j = [0 \div 79]) \quad (6)$$

Finally, feature vector of the 68 points model is presented by the following (7) equation:

$$F_{68points}^{distance} = [R_0; R_1; R_2; \dots; R_{79}] \quad (7)$$

- *The 34 points-04 angles model and The 68 points-05 angles model:* Because of non-rigid face that some angles of face do not change. We compose angles for 34 points and 68 points as illustrated in Figure 3 (c,d). While couple lines are the same color that creates an angle θ_l (e.g. n_1^l connect points 21 and 33; n_2^l connects points 22 and 33). The angle θ_l is calculated as the following (8) equation:

$$\cos(\theta_l) = \frac{\overline{n_1^l} \cdot \overline{n_2^l}}{|\overline{n_1^l}| \cdot |\overline{n_2^l}|} \quad (8)$$

Feature vectors is presented in the following (9) equation:

$$F_{angle} = [\cos\theta_0, \cos\theta_1, \dots, \cos\theta_m] \quad (9)$$

While $m=[0 \div 3]$ for the 34 points model and $m=[0 \div 4]$ for the 68 points model. Finally, Euclid distance and angles features are normalized and concatenated as the following (10) equation:

$$F = \left[\|F_{angle}\|_2; \|F_{distance}\|_2 \right] \quad (10)$$

2.2.2 Deep learning-based face representation

In this paper, two CNN-based methods are used: Resnet50 [18] and MobileNet_V2 [29]. These models have obtained accuracy, computational complexity, model complexity, memory usage, and inference time on pattern recognition. In some cases, performance of MobileNet_V2 and Resnet50 are the same but time cost of MobileNet is lower than Resnet model. Both network use input image size (224x224 pixel) and pretrained model by ImageNet-1000.

In this paper, we must transfer learning model for face application of two networks on PC Pentium 3, NVIDIA 4GB, Then, the fine-tuned models are used to extract feature vectors at FC6 layers as illustrated in (11):

$$F_{Resnet50/MobileNet} = [X_1 \ X_2 \ \dots \ X_{2048}] \quad (11)$$

2.3 Face recognition

All feature vectors are utilized as input of classifiers. There are six types of feature vectors such as $F_{34points}$, $F_{68points}$, $F_{34points-angles}$, $F_{68points-angle}$, $F_{Resnet50}$ and $F_{MobileNet}$. In this paper, five classifiers are considered with Knn [26], SVM [22], Random Forest [23], Decision Tree [25] and Naïve

Bayes [24]. The output of classifiers will be one value among $\{1,2,\dots,K\}$ (K is faceID of participants) corresponding to the IDs of subjects in both two datasets. We will test all default kernels of classifiers to obtain the best efficient model.

2.4 Intelligent door system

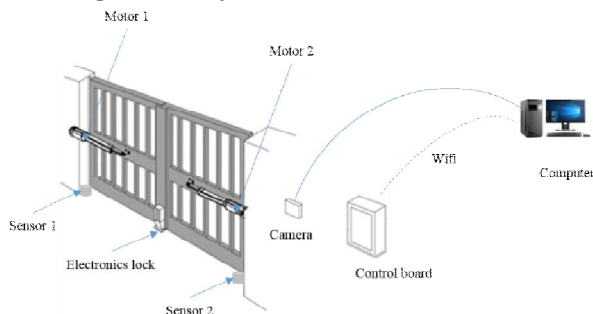


Figure 4:Hardware designing for intelligent door

In this section, hardware architecture is showed in Figure 4. Door system is composed by two motors to control open and close two haft parts of door; two sensors to detect states of the haft part doors; Electronics lock aims to latch. All components are connected to Arduino’s control board that is plug in LAN network with computer. In addition, computer is connected to camera through USB cable to capture images as showed in Figure 5 following.

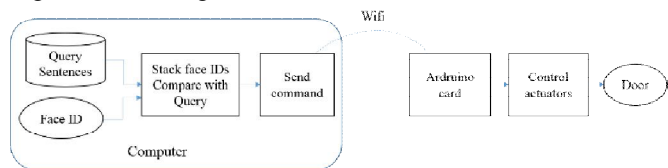


Figure 5:Query comparison and Door controlling

Given images sequence that are captured from camera. Computer processes and recognizes faceID. If faceID results returns true and closed states of two sensors are checked, lock is firstly opened. Then two motors are forwardly run until sensors is opened states. A timer is activated, then motors are inversely operated to the first position and lock is closed.

2.5 Dataset and Evaluation Protocol

2.5.1 Datasets

In this paper, our objective aims to deploy a real application. That our system must resolve private and security problems. Therefore, we try to capture two new datasets of Vietnamese: - *Dataset1*: is invited 6 males and 4 females. Each participant stands opposite with camera to capture a haft top of body. Image resolution is 480x640 pixels. Human rotates face and stand at three different positions (distances from user to camera: 0.5m; 0.7m and 0.9m). Each subject captures 300 to 400 images.

- *Dataset2*: is captured by 6 subjects (3 males and 3 females) and take a haft top of body too. Differ to Dataset1, this dataset includes two or three people in scene. Subject stands the nearest distance with camera who could be give permission task. Participants also stand at various distances to camera (as Dataset1). Each subject captures 550 to 600 images.

2.5.2 Protocol

In this paper, we use leave-one-out-cross-validation method to evaluate face recognition results. It is mean that the images

total of each subject is divided to 10. In an experimental, 9 parts are used to train model and remain images is utilized to evaluated. This method is illustrated detail in the following Figure 6.

While N is the images total of an user; ($m = N/10$) is the images number for testing (yellow color) and ($N-m$) images for training (blue color). This work is rolled from the first window to the end of images (from time 1 to time 10).

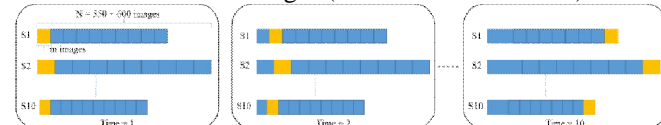


Figure 6: Protocol to evaluate face recognition system

4. EXPERIMENTAL

In this section, we will present the experimental results. Our objective is to evaluate impact of: (1) Face detection methods; (2) Hand craft-based face recognition; (3) Deep learning-based face; (4) Real application. These experiences are implemented on our dataset and protocol as presented detail at Section2.5. Evaluation is wrapped by Python language on Laptop Pentium core i5 3.1 GHz CPU;RAM 4GB.

4.1 Face detection

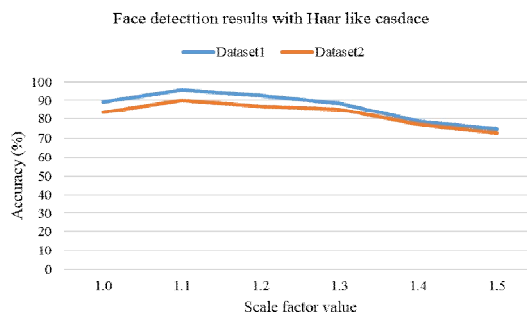


Figure 7: Face detection results withscale factors of Haar Cascade This part aims to evaluate the most efficiency detection face when adjust scale factor of Haar like cascade. The results are showed in the Figure 7 following. A glance at the figure, the highest detection accuracy obtains 94.6% and 90.1% when scale value is equal to 1.1 for both dataset1 and dataset2 respectively. The face detection accuracy is dramatically reduced when scale factor is increased (acc = 74.8% at scale = 1.5 for dataset1). The best efficient factor value (equals to 1.1) is used for the next evaluations.

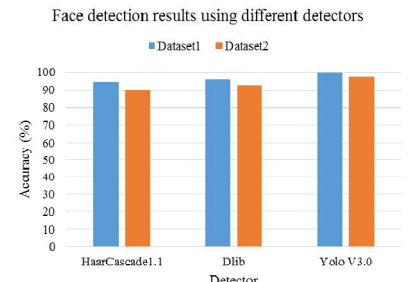


Figure 8: Face detection results with different methods

- Compare face detection accuracy between various methods: Given three detectors (the best Haar model; Dlib model and YOLO_V3 network) that are presented detail in Section 2. We evaluate all face detection accuracy of all subjects of Dataset1

and Dataset2 as presented in Section 2.5. It is apparent that, the face detection is highest with YOLO model (average accuracy is 97.1%). the slowly lower is Dlib with 96.3%. Although of using the best accuracy of Haar cascade model but it is the worst model with accuracy at 94.6%.

- Trade of between time cost and/or accuracy: Table 1 shows time and accuracy of various face detectors as: Haar like cascade, Dlib and YOLO_V3. It is clear that, deep learning technique (YOLO network version 3) accounts the highest value at 99.5%, but it spends the long time at 1.3731(s). In additionally, the best model of HaarCascade (scale factor is equals to 1.1) is worst results on both accuracy and computational time. To deploy a real application, the time cost is importance problem, we must trade of between time and accuracy. While Dlib detector requires the lowest time cost with 0.0656s and face detect accuracy obtains the sight lower at 96.3%. The face detection result of Dlib model will be utilized in the next evaluations.

Table 1: Face detection results of detectors on the Dataset1

| | HaarCascade1.1 | Dlib | YoloV3.0 |
|----------------|----------------|--------|----------|
| Avr. Acc (%) | 95.1 | 96.3 | 99.5 |
| Time cost (ms) | 0.0917 | 0.0656 | 0.3732 |

4.2 Hand craft-based face recognition

Table 2:Affect of SVM classifiers on various hand craft-based Feature representations

| | 39 factors | 68 factors | 39 factors + Angles | 68 factors+Angles |
|------------|--------------|--------------|---------------------|-------------------|
| Linear | 94.4 | 95.28 | 90.7 | 92.08 |
| Sigmoid | 13.05 | 3.98 | 14.08 | 15.7 |
| Rbf | 95.51 | 94.19 | 93.21 | 97.83 |
| Poly | 87.79 | 78.28 | 89.34 | 90.89 |

- Affects of SVM classifiers with the various face representations: In this evaluation, we would like to test on four face representations with various kernels of SVM classifier. A glance at the Table2, the Linear and Rbf kernel obtains the highest accuracy on all feature extractors. While Sigmoid kernel accounts the lowest results and Poly kernel is barely smaller in overall. Moreover, the 68 points - Angles model is more efficient than other models on all Kernels. Specially, the 68 points - Angles model uses the Rbf SVM classifier that reaches the best evaluation at 97.83%. Therefore, this classifier is also utilized on our nextexperiences.

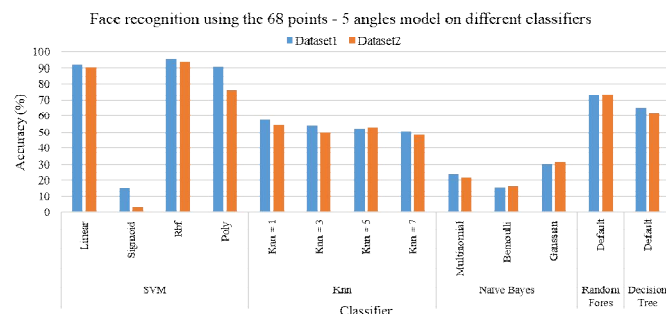


Figure 9: Face recognition results with different classifiers

- Hand craft-based face recognition results on different classifiers: This experience aims to evaluate efficient of classifiers such as: SVM, Knn, Naive Bayes, Random Forest and Decision Tree with the 68 points-Angles feature. The

Figure9 shows that, the Rbf SVM classifier also obtains the highest results on both dataset1 and dataset2 (with 95.2% and 93.6% respectively). Others accuracy of classifiers are worse than the RBF SVM classifier.

We then consider time cost of the Rbf SVM on self-design-based feature representations as illustrated in Table3.It reveals that spending times are similar for the hand craft-based face recognition systems. While accuracy of the 68 points-Angles model is the best value.

Table 3. Time cost of face recognition using hand craft-based face feature extractor

| | 39 factors | 68 factors | 39 factors +Angles | 68 factors+Angles |
|----------------|------------|------------|--------------------|-------------------|
| Time cost (ms) | 0.0598 | 0.0625 | 0.0615 | 0.0673 |

4.3 Deep learning face recognition

Figure 10 illustrates deep learning-based (Resnet50 and MobileNet network) face recognition results on two datasets. The Resnet architecture is more efficient than others method on both dataset1 and dataset2 with recognition accuracy at 100% and 96.9%, respectively. The MobileNet network is slightly lower with 95.6% and 94.1%. These results are comparative with hand craft-based method (the 68 points - Angle with Rbf SVM model) that only slowly smaller than deep learning-based. While time cost is dramatically higher than others (67.3ms vs 2.01s and 1.39s).

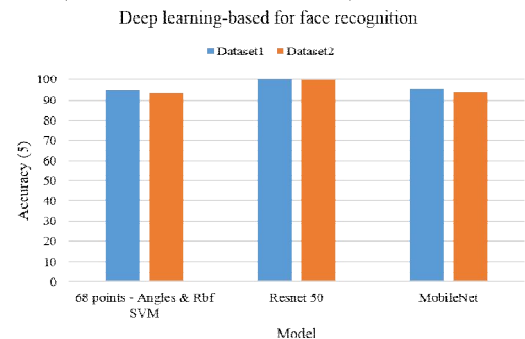


Figure 10: Face recognition results with CNN methods.

4.4 Time cost for door intelligent control system using face identification

In this section, we deployed a real application to automatic control door using face recognition as presented in Section 2.4. While system utilized random queries to require end-user (with three question such as: straight - slightly left- slightly up; slightly left- straight - slightly down;...). Therefore, time cost for one time is illustrated in Figure 11 following. We utilize Dlib model to detect face and show 68 landmark points. Then, the 68points-Angle model and the Rbf SVM classifier is deployed to recognize face. The result of face recognition is used to control door.

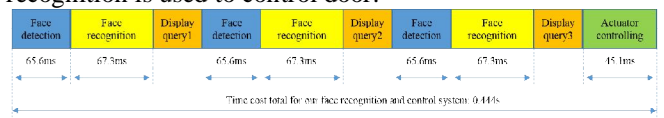


Figure 11: Time cost of our proposed system.

15 subjects are invited to test our system. We carefully measure both time cost as well as accuracy. A glance at the Figure 11, average consumption is 0.444s for each controlling while accuracy obtains 96.7% in overall.

5. CONCLUSION

In this paper, we have deeply investigated the results of face recognition and its application. Experimental is conducted on two datasets. The evaluations lead to some following conclusions: i) For face detection, Dlib method is more efficient than HaarCassade on both accuracy and time cost. Specially, Dlib spent smaller time cost than YOLO_V3; ii) Concerning approaches issue, the hand craft –based proposed method has obtained sight lower performance than deep learning – based method while computation cost is dramatically smaller on both datasets. These conclusions open some directions in our future works. Firstly, we will deploy evaluation of end-users. Secondly, we also test and compare with other state-of-the-art convolutional neuron network.

REFERENCES

1. N. R. Borkar and S. Kuwelkar, **Real-time implementation of face recognition system**, *International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 249-255, Erode, 2017.
2. K. Koide, E. Menegatti, M. Carraro, M. Munaro and J. Miura, **People tracking and re-identification by face recognition for RGB-D camera networks**, *European Conference on Mobile Robots (ECMR)*, pp. 1-7, 2017
3. X. Tian, **Face Recognition System and It's Application**, *First International Conference on Information Science and Engineering*, pp. 1244-1245, Nanjing, 2009.
4. D. A. Chowdhry, A. Hussain, M. Z. Ur Rehman, F. Ahmad, A. Ahmad and M. Pervaiz, **Smart security system for sensitive area using face recognition**, *CSUDET*, pp. 11-14, Selangor, 2013.
5. Z. Jian and S. Wan-juan, **Face detection for security surveillance system**, *5th International Conference on Computer Science & Education*, pp. 1735-1738, 2010.
6. P. A. Viola and M. J. Jones, **Rapid object detection using a boosted cascade of simple features**, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, I-I, 2001.
7. W. LU and M. YANG, **Face detection based on viola-jones algorithm applying composite features**, *International Conference on Robots Intelligent System (ICRIS)*, pp. 82–85, 2019.
8. A. Adouani, W. M. Ben Henia, and Z. Lachiri, **Comparison of haar-like, hog and lbp approaches for face detection in video sequences**, *16th International Multi-Conference on Systems, Signals Devices (SSD)*, pp. 266–271, 2019.
9. A. Juhong and C. Pintavirooj, **Face recognition based on facial landmark detection**, *10th Biomedical Engineering International Conference*, pp. 1–4, 2017.
10. X. Zhang, T. Gonnot, and J. Saniie, **Real-time face detection and recognition in complex background**, *Journal of Signal and Information Processing*, pp. 99–112, 2017.
11. A. S. M. Asif Anjum Akash and A. MAH, **Improvement of haar feature based face detection in opencv incorporating human skin color characteristic**, *Journal of Computer Science Applications and Information Technology*, 2016.
12. H. Xia and C. Li, **Face recognition and application of film and television actors based on dlib**, *12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, pp. 1–6, 2019.
13. N. Boyko, O. Basystiuk, and N. Shakhovska, **Performance evaluation and comparison of software for face recognition, based on dlib and opencv library**, *IEEE Second International Conference on Data Stream Mining Processing (DSMP)*, pp. 478–482, 2018.
14. S. Sharma, K. Shanmugasundaram, and S. K. Ramasamy, **Farec CNN based efficient face recognition technique using dlib**, *International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pp. 192–195, 2016.
15. X. Ren, J. Ding, J. Sun, et al., **Face modeling process based on dlib**, *Chinese Automation Congress (CAC)*, pp. 1969–1972, 2017.
16. B. Handaga, B. Murtiyasa, and J. Wantoro, **Attendance system based on deep learning face recognition without queue**, *Fourth International Conference on Informatics and Computing (ICIC)*, pp. 1–4, 2019.
17. Parikshit Pawde , Anjali Bhargava, Anuradha Dubey, Sheetal Hablani, Sweta Jain, **Real Time Attendance System using One Shot Learning for Face Recognition**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol.9, No. 2, pp. 1986 – 1990, 2020.
18. K. He, X. Zhang, S. Ren, et al., **Deep residual learning for image recognition**, *Computer Vision and Pattern Recognition (CVPR)*, 2016.
19. D. E. King, **Dlib-ml: A machine learning toolkit**, *J. Mach. Learn. Res.*, pp. 1755–1758, Oct, 2009.
20. <https://github.com/opencv/opencv/tree/master/data/haarcascades>, 2020
21. <http://dlib.net/>, 2020.
22. C. I. C. Burges, **A Tutorial on Support Vector Machines for Pattern Recognition**, vol. 43, pp. 1-43, 1997.
23. Ho, Tin Kam, **Random Decision Forests**, *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pp. 14–16, August 1995
24. McCallum, Andrew; Nigam, Kamal, **A comparison of event models for Naive Bayes text classification**, *AAAI-98 workshop on learning for text categorization*. 1998
25. Quinlan, J. R., **Induction of decision trees**, *Machine Learning*. pp. 81–106. 1986
26. Altman, Naomi S., **An introduction to kernel and nearest-neighbor nonparametric regression**. *The American Statistician*. Vol.46, No.3, pp. 175–185, 1992.
27. Joseph Redmon and Ali Farhadi, **YOLOv3: An Incremental Improvement**, *CoRR journal*, Vol. abs/1804.02767, 2018
28. <https://github.com/ssthanhng/yoloface>, 2020

29. M. Sandler, A. G. Howard, M. Zhu, et al., Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation, *CoRR journal*, Vol. abs/1801.04381, 2018.
30. Murugesan M., Santhosh M., Sasi Kumar T., Sasiwarman M., Valanarasu I., **Securing ATM Transactions using Face Recognition**, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 9, No. 2, pp. 1295 – 1299, 2020.