Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 157–163 DOI: 10.15439/2022R18 ISSN 2300-5963 ACSIS, Vol. 33

A Flexible Approach for Automatic Door Lock Using Face Recognition

Van-Dat Ly, Hong-Son Vu
Faculty of Electrical and Electronics Engineering, Hung Yen University of Technology and Education
Khoai Chau, Hung Yen, Vietnam
First author: lydat.ute@gmail.com
Corresponding author: hongson.ute@gmail.com

***Abstract*—The model of smart door lock using face recognition based on hardware is the Jetson TX2 embedded computer proposed in this paper. In order to recognize the faces, face detection is a very important step. This paper studies and evaluates two methods of face detection, namely Histograms of Oriented Gradients (HOG) method which represents the approach using facial features and Multi-task Cascaded Convolutional Neural Networks method (MTCNN) represents using of deep learning and neural networks. To evaluate these two methods, the experimental model is used to verify the hardware platform, which is the Jetson TX2 embedded computer. The face angle parameter is used to rate the detection level and accuracy for each method. In addition, the experimental model also evaluates the speed of face detection from the camera of these methods. Experimental results show that the average time for face detection by HOG and MTCNN method are respectively 0.16s and 0.58s. For face-to-face frames, both methods detect very well with an accuracy rate of 100%. However, with various face angles of 30o, 60o, 90o, the MTCNN method gives more accurate results, which is also consistent with published studies. The smart door lock model uses the MTCNN face detection method combined with the Facenet algorithm along with a data set of 200 images for 1 face with accuracy of 99%.**

***Index Terms*—Face detection, Face recognition, HOG method, MTCNN method, Jetson TX2 embedded computer, deep learning.**

I. Introduction

With the development of embedded computers with increasingly faster processing cores, face recognition problems are becoming easier. To detect a face on an image or a video frame, it is necessary to first detect the area of the image containing the human face. This can be considered as the most important step, deciding the accuracy of the identification problems. The face detection can base on algorithms, encoding features and relationships between face features, can also use face standard samples to describe faces or face features. The methods can be mentioned as Haarlike, Adaboost, HOG methods. Recently, methods using deep learning, machine learning, neural networks have been mentioned more and more. These method use models or patterns that will be learned from a set of training images that exhibit typical of the appearance of a human face in the image.

Some studies on the problems of face detection [1-4] have used the HOG method to detect people and recognize facial features. The effect of each calculation stage on the recognition efficiency was studied. Researches indicated that for good recognition results, small scale gradients, well oriented plots, high quality local contrast in overlapping descriptors were required.

There are some researches about face detection method using deep learning such as S3FD (Single Shot Scale-Invariant Face Detector) by S. Zhang et al. [5], the MTCNN method of Kaipeng Zhang et al. [6], the Faster R-CNN face detection method of S. Ren et al. [7]. Guangyong Zheng et al. [8] solved the difficult problem of training the deep neural networks by using ResNet to control the disappearance of gradient when face detection through video sequences. These methods were considered to be very efficient.

In this paper, the level of face detection at different angles and the same aspect ratio from the camera of two methods HOG and MTCNN based on the hardware platform Jetson TX2 are evaluated. Thereby helping to choose a face detection method for later recognition problems on embedded computers.

II. HOG method

The essence of the HOG method [1] is that the object shape is described by two matrices including the gradient magnitude matrix and the gradient direction matrix. To create these two matrices, the image is first divided into a grid of squares and on which many adjacent or overlapping local regions are defined. A local area consists of many local cells measuring 8x8 pixels. Then, a graph of gradient magnitude statistics is calculated on each local plot. The HOG descriptor is created by concatenating the histogram vectors corresponding to each cell into a composite vector. To improve accuracy, each value of the histogram vector over the local area will be normalized to a first-order or quadric order normal. This normalization is intended to provide better invariance to changes in illumination and shade.

The HOG method for object detection consists of 5 steps:

Preprocessing step: In this step, the frames will be changed to the same general size, to facilitate the equal division of the image into blocks, cells and feature calculation in the next steps.

$$I_x = I \cdot D_x; I_y = I \cdot D_y \quad (1)$$

$$D_x = \begin{bmatrix}-1 & 0 & 1\end{bmatrix}; D_y = \begin{bmatrix}1 & 0 & -1\end{bmatrix}^T \quad (2)$$

Where: T is the transpose matrix

Gradient magnitude:

$$G = \sqrt{I_x^2 + I_y^2} \quad (3)$$

Gradient calculation step: Performed by two convolutions of the original image with 2 vector dimensions respectively in two directions Ox and Oy.

The gradient direction:

$$\theta = \arctan\left(\frac{I_y}{I_x}\right) \quad (4)$$

Calculate feature vector for each cell step: Based on the slope, calculate the feature vectors in the direction on each cell. In which, the direction space is divided into p-dimensional feature vector of the cell. Then the tilt angle at each discrete pixel is transformed into the feature vector dimensions of the cell.

Block normalization step: By calculating an intensity threshold in a block and using that value to normalize all cells in the block, local histograms are contrast normalized for enhanced recognition form. The result after the normalization step will be a feature vector that is scaled to light, so it is more invariant to changes in lighting conditions.

Calculate HOG vector step: The feature vector of the block is calculated by concatenating the feature vector of each element in the cell together. The number of components of the feature vector at each block is calculated by the formula:

$$K = n_c \cdot K_1 \quad (5)$$

Where: K is the feature in a cell, n_c is the number of elements in a cell, and K_1 is the number of features in an element.

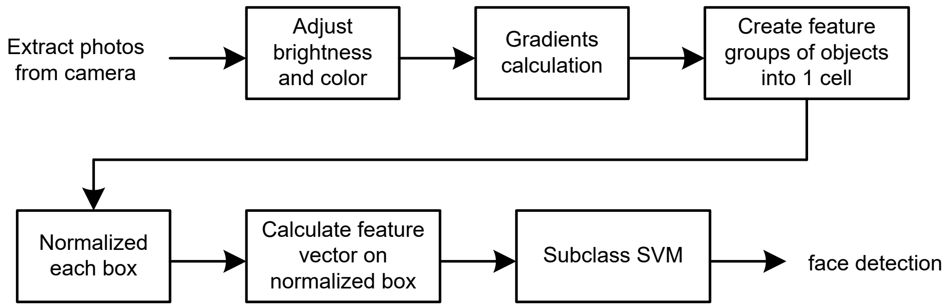

Fig. 1. Face detection procedure by HOG method

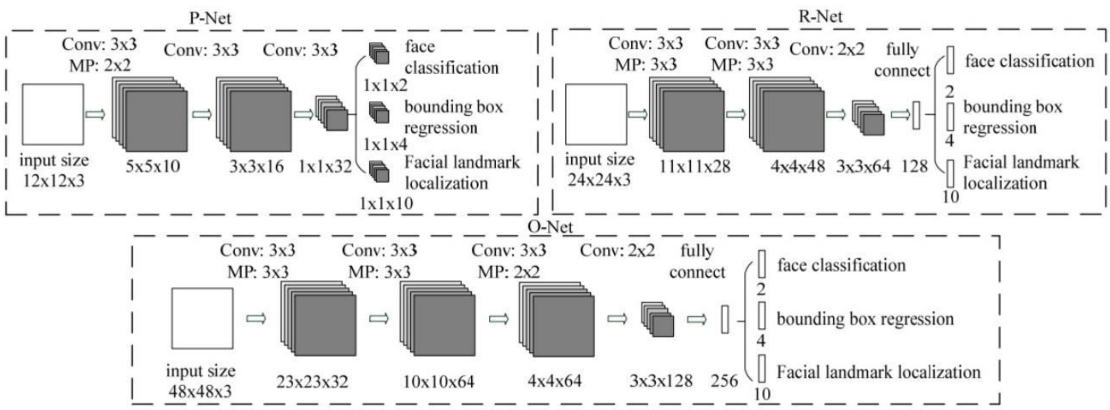

Fig. 2. MTCNN network architecture [6]

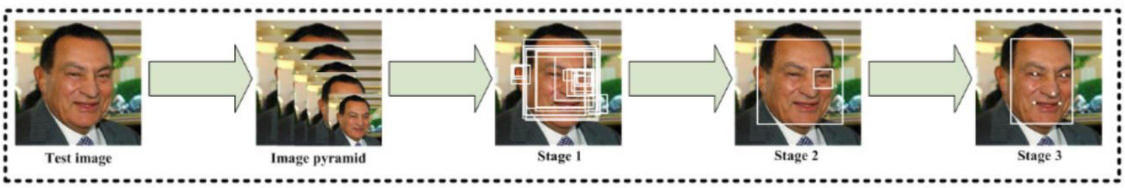

Fig. 3. Steps of image processing in MTCCN method [6]

III. MTCNN METHOD

MTCNN is a face detection method according to the cascade model [9, 10] using 3 neural networks namely P-Net, R-Net, O-Net as shown in Figure 2 to detect faces in 3 steps as follows:

Step 1: The input image will be resized, to create many copies from the original image with different sizes, from large to small, forming 1 Image Pyramid with each copy used 1 core 12x12 pixels and step of 2 to go through the entire image, detecting faces. Then, the above cores are transmitted over the P-Net (Proposal Network). The lattice results in a series of bounding boxes, each bounding box will contain four corner coordinates to determine the location in its containing core.

Step 2: The image continues through the R-Net (Refine Network) where the images will be inserted with zero-pixels in the missing parts of the bounding box if the bounding box exceeds the boundary of the image. All bounding boxes will now be resized to 24x24. The following result is also the new coordinates of the remaining boxes and is fed into the next network, the O-Net (Output Network).

Step 3: O-Net (Output Network), do the same as in R-Net, the images are resized to 48x48. But the output of the network will now return 3 values including: 4 coordinates of the bounding box (out [0]), coordinates of 5 landmarks on the face, including 2 eyes, 1 nose, 2 wings lips (out [1]) and confidence score of each box (out [2]). All will be saved into a library with the above 3 points. These steps are depicted in Figure 3.

IV. MODEL TO EVALUATE THE FACE RECOGNITION LEVEL OF HOG AND MTCNN METHOD

The hardware is an embedded computer Jetson TX2 of NVIDIA which is developed with powerful configuration with dual core NVIDIA Denver2 2GHz, 4 cores ARM Cortex-A57 2GHz, 8GB of memory and integrated 256-Core Pascal GPU, built-in camera as shown in Fig. 4. With this configuration, Jetson TX2 is widely used for computer vision problems, artificial intelligence, deep learning and machine learning problems.

Fig. 4. KIT Jetson TX2

To evaluate the above two methods, Python language and Open source computer vision library (OpenCV) tool are used. OpenCV is a repository of open source code used to process images, develop graphical applications in real time. OpenCV allows to improve the speed of the CPU when performing real-time operations.

The input image from the camera is processed by two methods, the detection time is calculated from the start of recognition until the face is detected for each method. The input images for recognition are front view, 30^o, 60^o and 90^o inclined view. When the face is identified, the HOG method will draw a red rectangle, the MTCNN method will draw a blue rectangle around the recognized face.

In addition, the accuracy of the methods was evaluated over 300 trials for each face sample. Accuracy will be calculated using the formula:

$$A(\%) = \frac{A_1}{300} \times 100 \tag{6}$$

Where: A is the accuracy, A_1 is the number of times the face was detected.

The algorithm for image processing is shown in Figure 5. In which, the time for face detection for each method is calculated from the start of calling the detection function until the face is detected. Because the recognition image is taken from the camera, the face detection using both methods above using a program will ensure objectivity when using the same input image source at the same time.

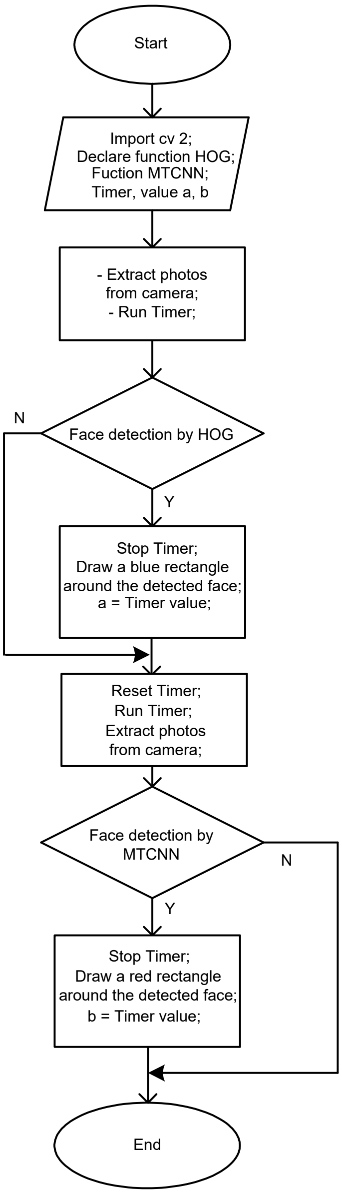

Fig. 5. Processing algorithm of face recognition

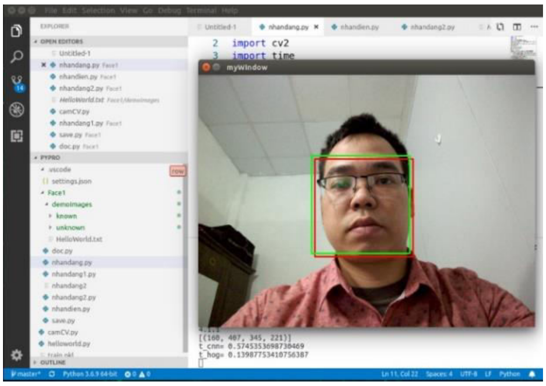

Fig. 6. Face detection in the front view

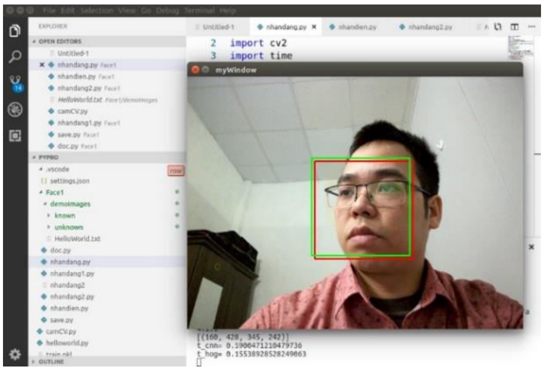

Fig. 7. Face detection with face anlge of 30^o

The results of face detection in the front view are presented as shown in Figure 6. The results show that both HOG and MTCNN methods can recognize the face well in front view.

Figure 7 presents face detection result with face angle of 30^o. The result show that HOG and MTCNN method have detected the face in this case.

Figure 8 and Figure 9 respectively describe the face detection results with various face angles of 60^o and 90^o. The results show that in this case, only the MTCNN method can detect the face, while the HOG method cannot detect it.

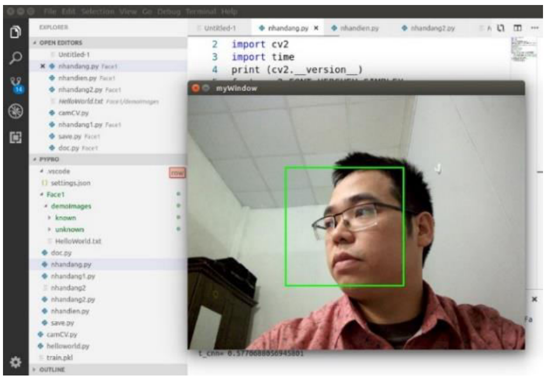

Fig. 8. Face detection with face anlge of 60^o

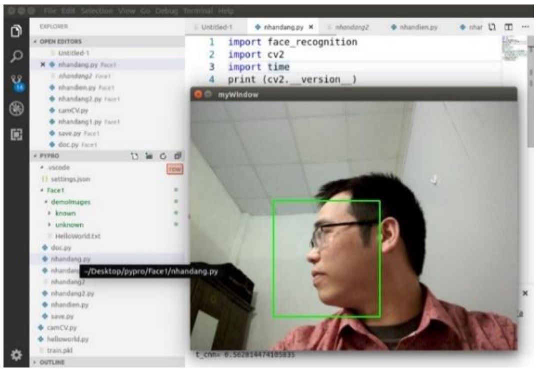

Fig. 9. Face detection with face anlge of 90^o

Synthesized results through 300 trials to identify faces by HOG method takes an average of 0.16s and MTCCN method takes about 0.58s. The accuracy over 300 trials are summarized in Table 1 and Figure 10.

The above results show that the processing time when running on the platform of Jetson TX2 for images with only 1 face of both methods gives very fast results. In which, the HOG method has a fast processing time of only 0.16s, but it is only accurate with direct angles and small tilt angles. The detecting time of the MTCNN method is slower than HOG method, but the average time is 0.58s, showing the processing power of the GPU core, this time also responds very well to the recognition problems. Besides, the MTCCN method gives very high accuracy at face tilt angles of 60^o and 90^o.

TABLE I. RESULT OF IMAGE DETECTION WITH DIFFERENT FACE ANGLE IN 300 TRIALS

Face angle	**Detection level**	
	HOG method	***MTCNN method***
Direct direction	100%	100%
30^o	91.3%	98.3%
60^o	22.6%	91.6%
90^o	0%	30.6%

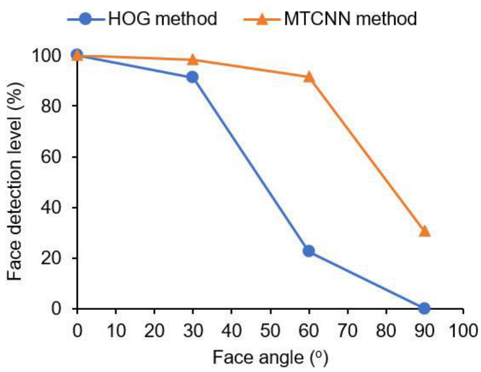

Fig. 10. Face detection with various face angle in 300 trials

V. Face Recognition and Application to Unlocking Problem

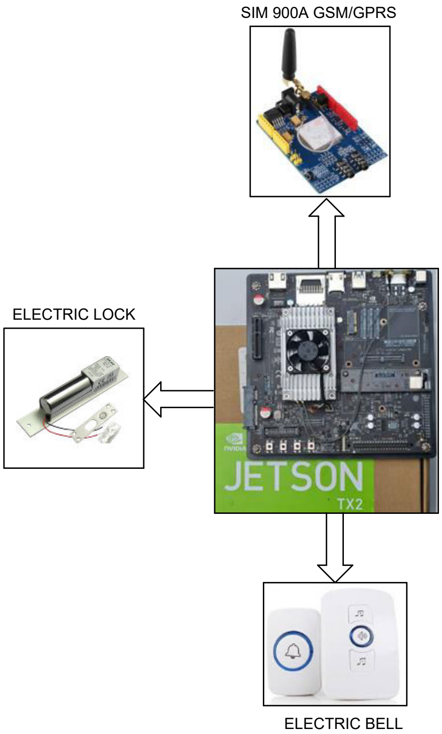

Fig. 11. Hardware structure of the system

From the evaluation results of the above two face detection methods, a model applies the face recognition algorithm to automatically unlock the door is built. The model uses the MTCCN method that combines the Facenet algorithm and the training image set of 200 images for one face. The SQLite application is used to create a dataset where the images of the dataset are taken directly from camera.

The system's hardware structure is designed with compact requirements that can be applied in real life. The face recognition application in opening and closing the door automatically based on the principle of identifying faces appearing in the camera frame. Thereby, the door is automatic unlocking if the face appearing in the camera frame matches with the database.

The main blocks of the system are depicted in Figure 11. The main equipment includes: Jetson TX2 embedded computer on the computer that has built-in camera. The actuator includes an electric lock, an electric bell and a 900A Sim module that sends messages to the owner's phone number. The camera has the function of collecting images, sending frames to an embedded computer for processing and executing image processing algorithms including face detecting and face recognition through the OpenCV library.

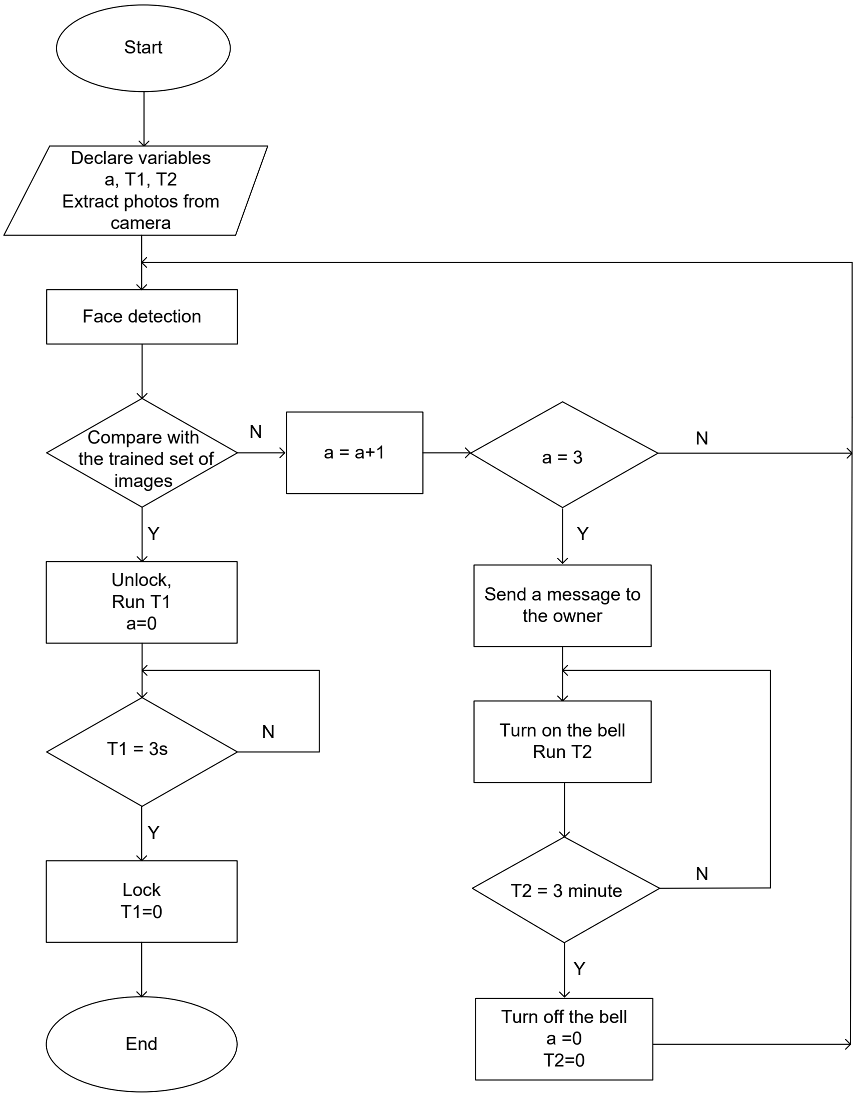

Fig. 12. Flowchart of face recognition algorithm for automatic door unlocking problem

The face detected from the camera frame is compared with the training set, if it matches, the person's name is given and the key is automatically unlocked. In case of 5 invalid face unlock times, the system will alarm with a buzzer for 3 minutes and send a message to the pre-set phone number via the 900A SIM module. The algorithm flowchart is shown in Figure 12. Through testing, it shows that with a frontal view, the average time to process the unlock is 0.7s and the accuracy reaches over 99%.

VI. CONCLUSIONS

In this paper, two models for face detection on the hardware platform Jetson TX2 have been deployed. Both methods showed good results for the front view. For the problem that needs to be identified at the inclined angle, the MTCNN method gave better efficiency. The MTCNN method and the Facenet algorithm were selected to apply to the automatic door unlocking problem. The system was compactly built and had high accuracy. In addition, the experimental research results can be applied to other problems such as detection, identification, warning or automatic time-keeping problems in corporate companies. Besides, the problem of reporting people doesn't wear facemasks during the complicated situation of the Covid-19 epidemic is also applied in this algorithm.

REFERENCES

[1] N. Dalal, B. Triggs, "Histograms of Oriented Gradients for Human Detection". IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005

[2] S K Eng* , H Ali, A Y Cheah, Y F Chong, Facial expression recognition in JAFFE and KDEF Datasets using histogram of oriented gradients and support vector machine, IOP Conf. Series: Materials Science and Engineering 705 (2019) 012031, doi:10.1088/1757-899X/705/1/012031

[3] S Yallamandaia, N Purnachand., "A novel face recognition technique using Convolutional Neural Network, HOG, and histogram of LBP features", IEEE, 2022.

[4] D.Lakshmi, R.Ponnusamy, "Facial emotion recognition using modified HOG and LBP features with deep stacked autoencoders", Microprocessors and Microsystems, vol. 82, 2021.

[5] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang and S. Z. Li, "Sˆ 3FD: Single Shot Scale-Invariant Face Detector," in Computer Vision (ICCV), 2017 IEEE International Conference on, 2017

[6] Kaipeng Zhang1 Zhanpeng Zhang2 Zhifeng Li1 Yu Qiao1, Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. IEEE Signal Processing Letters (SPL), vol. 23, no. 10, pp. 1499-1503, 2016

[7] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015

[8] Guangyong Zheng, Yuming Xu, “Efficient face detection and tracking in video sequences based on deep learning”, Information Sciences, vol. 568, pp. 265-285, 2021.

[9] P. Viola and M. J. Jones, "Robust real-time face detection," International journal of computer vision, vol. 57, pp. 137-154, 2004

[10] Chunming Wu, Ying Zhang, “MTCNN and FACENET Based Access Control System for Face Detection and Recognition”, Automatic Control and Computer Sciences, vol. 55, pp. 102-112, 2021