An Efficient Security Access System for Computers using Image Processing Techniques

V.B.C.Ayyappan, P.Gouda, C.A Origanti, S.Sreedhar
Visiting Students, Amritha University, India

***Abstract*—This paper describes the method for an efficient face recognition system for access control to a computing device while utilizing image processing techniques. This unique method can be used for Windows, Linux, Unix operating systems. The proposed system detects the user's face with a 98% probability and detects the intruder's face with less than 1% probability. The detection time for the face is less than 2 seconds and unlocks the system within 2 seconds because of triggering all the tasks from windows, but still making it efficient from the previous works made. Additionally, the system application alerts the user when an intruder tries to unlock the system and also sends the image of the intruder's face to the user, thus assuring maximum security to the system.**

I. INTRODUCTION

Recently, there has been significant interest in the development of facial recognition due to its use in biometric technology which provides good security to any access. This has been very reliable since this method uses distinguished facial features to identify a person hence offering a quick, automatic, seamless verification experience.

Graphics Card	Training and Testing Time	RAM	Detection + Unlocking Time
2 GB**	40 sec	4 GB**	9+9
6 GB*	20 sec	8 GB*	4+4
12 GB*	10 sec	16 GB*	2+2

Table 1. Different System Requirements and their results
Legend: '*' are experimentally verified, '**'are theoretically analyzed

In this paper, a deep neural network model is used to train and test the data. This system is created after studying various deep-learning models and other techniques for image segmentation and enhancement and analyzing their results against our model. This article purely describes only one efficient model and their results have been discussed. Also, this is integrated into the windows and Ubuntu operating system for ubiquity. The proposed system works well for Windows OS 8 and above, Ubuntu version 16 and above.

II. PROCESS

A. Input

A large number of face samples are captured using a haar cascade frontal face algorithm. In this case, 300 images are captured to cover all the possible facial features.

B. Image Segmentation

Firstly, it creates a boundary around the face to be detected using an object detection method. Instance segmentation is done to identify a single instance of the image, in our case a single face. Threshold segmentation sets a threshold value for the pixel values in the picture above which will be discarded for identifying the profile of a face. Mask R- CNN method is applied only for multiple objects so that it masks the background faces and captures only the frontal face [1].

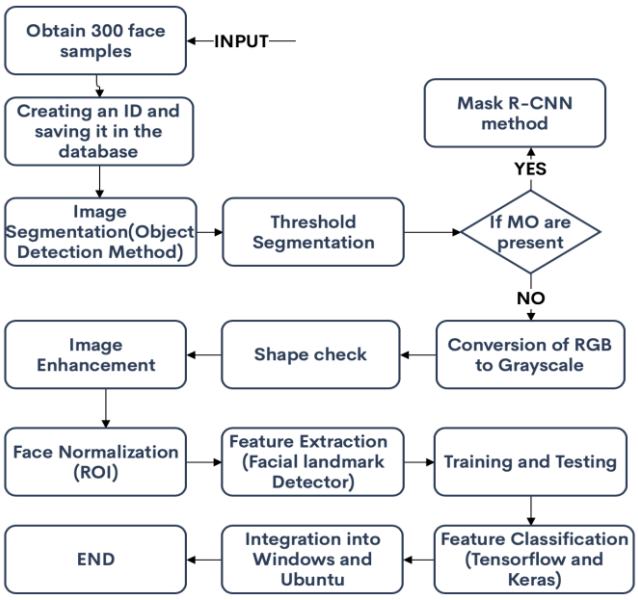

Fig1. Process flow of the system

C. Image Enhancement

The first stage in enhancing the image is to convert the RGB image to grayscale for conducting operations on it easily. Due to this conversion step, the background and the face may be blended or distorted, so it is necessary to do a face shape check. After this noise is removed from the image, sharpened and brightened.

D. Face Normalization

Face detection is a process of localizing and extracting the face region from the background. Those features are independent of face features and will affect the recognition rate significantly. The system of feature detection has detected three features such as left eyeball center, right eyeball center & mouth center. Each feature is detected using the Region of Interests (ROI) [2]. Then, Geometric normalization is applied to align the eyeball center and mouth center alignment. Lastly, brightness normalization is done by mean centering the pixel values and linearly smoothening the edges and normalizing the contrast with histogram equalization.

E. Feature Extraction

Feature extraction is the process by which certain features of interest within an image are detected and represented for further processing. Facial landmarks are used to localize and represent salient regions of the face detecting facial landmarks is a *subset* of the *shape prediction* problem.

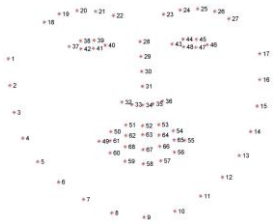

Fig 2. 68 Point facial landmark vector points

The pre-trained facial landmark detector inside the dlib library is used to estimate the location 68(x,y) coordinates that map to the facial structures on the face [3]. Finally, 9/10th of the data is sent for training, 1/10th of the data is used for testing the data.

F. Feature Classification

Feature mapping is a pattern recognition technique used to categorize a huge number of data into a number of different classes. Firstly, the data is sent into the tensor flow and then into Keras which acts as a top layer to the system [4].

II. RESULTS AND DISCUSSION

After the data is passed into the KERAS, we have got the value accuracy and value loss for the overall neural network and also loss and accuracy for the training and testing data in each epoch. It is clear if epochs are increased and the rate is decreased, there may be an increased accuracy, but dropouts in neurons will be high, overfitting the system, so the system will not be efficient. Thereby increasing the probability of detecting the intruder's face. The input shape for the MLP is 128 and activation functions used are relu and softmax and the Stochastic Gradient Descent(SGD) optimizer is used.

No.of Epochs	Batch Size	Learning Rate	Loss	Accuracy	MLP Error	Probability
300	50	0.01	0.16	0.936	4.95%	98.2%
500	30	0.1	0.01	0.998	14.5%	64.8%
150	50	0.01	0.38	0.62	8.23%	88.4%

Table 2. Results by changing different parameters.

III. CONCLUSIONS

An efficient system for face recognition is created after studying various methods and analyzing them to increase the robustness and accuracy of the system. It is also integrated into the windows operating system without compromising the OS-user agreement by triggering tasks in the Windows task manager. Therefore, the system is very reliable and accurate with the results shown above. In the future, by increasing the number of facial points one can increase the accuracy of the system as more vector coordinates give many efficient results.

IV. ACKNOWLEDGMENTS

The authors would like to thank Dr. Tahernezhadi and Dr. Azad from the department of electrical engineering for their technical and moral support throughout this project.

VI. REFERENCES

[1]Y. Li and Z. Guo, "An effective method of 3D facial features segmentation," 2014 7th International Congress on Image and Signal Processing, Dalian, 2014, pp. 375-380.

[2]S. R. Benedict and J. S. Kumar, "Geometric shaped facial feature extraction for face recognition," 2016 IEEE International Conference on Advances in Computer Applications (ICACA), Coimbatore, 2016, pp. 275-278.

[3] J. C. T. Kwong, F. C. C. Garcia, P. A. R. Abu and R. S. J. Reyes, "Emotion Recognition via Facial Expression: Utilization of Numerous Feature Descriptors in Different Machine Learning Algorithms," TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Korea (South), 2018.

[4] D. Demirović, E. Skejić, A. Šerifović–Trbalić "Performance of Some Image Processing Algorithms in Tensorflow," 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP), Maribor, 2018.