Face Recognition Security Module using Deep Learning

R. Augustian Isaac
Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Tamil Nadu, India

Abhinav Agarwal
Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Tamil Nadu, India

Priyanshi Singh
Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Tamil Nadu, India

Abstract – This project designs face recognition system for smart home/office security applications. The design is implemented using webcam and programmed using dlib and openCV. The connection between cam and computer can be both wired and wireless. We are going to use a very simple approach to face recognition using deep learning. Here the deep neural network will generate an array of numbers that will represent face (known as face encodings). When two different images of the same person are passed, the network should return similar encodings (i.e. similar arrays) for both images, whereas when images of two different people are passed, the network should return very different encodings for the two images. This shows that the neural network requires to be trained to automatically recognise different features of faces and generate encodings based on that. The output of the neural network can be thought of as an identifier for a particular person's face, if you pass in different images of the same person, the output of the neural network will be very similar/close, whereas if you pass in images of a different person, the output will be very different. The additional feature here is that we are going to run this algorithm on database too, to make it more efficient than ever.

Index Terms – Face Recognition, Deep Learning, OpenCV, dlib

1. INTRODUCTION

With the advancement of technology, automation is being incorporated in all the fields. These days the need of security is emphasized and hence new methods are being created to serve the sole purpose of increased security. Biometric recognition security system was launched way back and by now hackers and hi-tech thieves have found a way to get past it.

So now facial recognition feature is launched in the field of security. It's a very effective way of ensuring that the person entering the premises is allowed to as his/her face is present in the system. The person doesn't have to do anything, just stand in front of the gate before entering and let the camera take image of your face and once the system verifies that you are allowed the gate is opened and the effort from your side will be zero.

Although, face recognition was introduced in this field long back it still has a drawback and that is that it requires multiple dataset of same person containing their face image in multiple different angles to recognize them. This takes extra space in database and if the algorithm is not able to find the specific angle image it can display false results. So in our system we are using dlib deep learning library along with OpenCV which requires only one specific image if the person. One another major drawback of existing systems is that sometimes the dataset images are not at all compatible with the algorithm and cause a lot of problem for the user while run time. To resolve that error, we have incorporated a web module to feed images to the database which will do a background check on the uploaded image at run time whether it will be compatible with our algorithm or not. The image will get added to database only and only if it is fully compatible to run with our algorithm.

It gives higher accuracy than any other face recognition machine learning algorithm. The maximum tolerance or difference in neural network of taken image and database image allowed in this system is 6%. If the neural network of both the images is more than the allowed limit then the system will deny the permission to enter.

2. RELATED WORK

- Design of Face Detection and Recognition System for Smart Home Security Application
- Effective Face Recognition using Deep Learning based Linear Discriminant Classification

- Forensic Face Photo-Sketch Recognition Using a Deep Learning-Based Architecture
- Face Recognition using modified Deep Learning Neural Network
- A Face-Recognition Approach using Deep Reinforcement Learning Approach for User Authentication
- Learning Face Recognition from Limited Training Data using Deep Neural Network

2.1 Design of Face Detection and Recognition System for Smart Home Security Application

In [1] basic idea is to to create a secure home environment by making the way to access a home by face detection instead of a key. It makes use of the traditional eigenfaces algorithm. When the person comes to the front gate the camera installed captures the image and feeds to the algorithm which then converts it into greyscale and then into a matrix and then compares the Euclidian distance between this image and dataset images and if that is less than threshold then person is authorized to access home else not.

2.2 Effective Face Recognition using Deep Learning based Linear Discriminant Classification

In [2] idea is to achieve effective face recognition using Deep Learning - Linear Discriminant Regression Classification (DL-LDRC) instead of traditional LRC. It is a recently developed method with enhanced accuracy as compared to LRC model. It is run on two, four and six train ORL and YALE datasets and gives better results than LRC.

2.3 Forensic Face Photo-Sketch Recognition Using a Deep Learning-Based Architecture

In [3] idea is to improve the efficiency of a crime investigation process by using a face recognition system to identify the person in the sketch made by artist as depicted by the eye-witness. The system is using deep convolutional network to learn the relationship between photos and sketches and make use of 3-D Morphable model to generate new images based on sketch. It was found that when multiple sketches are fed as dataset the performance is better. This system is one of the few that uses deep learning in hand drawn sketches.

2.4 Face Recognition using modified Deep Learning Neural Network

In [4] deep convolutional network has been implemented using MATLAB. Then two different synthetic image samples are generated from training set by applying noise. One with less distortion is termed Poisson Noise while the one with higher distortion is termed Gaussian noise. Then these synthetic images are added to previous training set to create augmented training data for CNN. Learning rate of CNN is kept to be 0.001. The major of drawback of this system is that in order to make it work, the batch size and momentum is prefixed. And also the system dropouts are frequent.

2.5 A Face-Recognition Approach using Deep Reinforcement Learning Approach for User Authentication

In [5] idea is to increase security while doing mobile payment by doing the user authentication through face recognition. The idea here is to increase the efficiency of the algorithm by making it work under varied light intensities as the user is on mobile so it can be in a very open space or a very confined space also. The system makes use of Deep Reinforcement Learning approach using python2.7 and Ubuntu on an Intel i7-6700 processor. The drawback still stays that the less the intensity of light, the less accurate the results will be.

2.6 Learning Face Recognition from Limited Training Data using Deep Neural Network

In [6] basic idea is to train the face recognition algorithm with minimal dataset using Deep Neural Network. The initial approach is to apply spatial transformer learning for face detection followed by designing a recognition network for efficient face recognition and finally displaying the details of this specially designed training procedure using limited dataset.

3. PORPOSED MODELLING

The core of the project is OpenCV library that helps process the images for the algorithm and dlib library which actually performs the face detection and recognition.

A. OpenCV

Open Source Computer Vision or OpenCV is a library comprising of programming functions which basically targets real-time computer vision. It was developed originally by Intel, and later was supported by Willow Garage and then by Itseez which was later, again, acquired by Intel. It is cross-platform library made free for use under the open-source BSD License. OpenCV supports various deep learning frameworks like TensorFlow, Torch/PyTorch and Caffe. OpenCV is written in C++ and same is its primary interface, although it still makes use of a less comprehensive but extensive older C interface. Also, there are bindings in Python, Java and MATLAB/OCTAVE.

The API for these interfaces is made available in the online documentation Wrappers in various languages such as C#, Perl, Ch, Haskell and Ruby have been developed to get attention by a wider audience. If the Intel's Integrated Performance Primitives are present on the system, the library will use these proprietary optimized routines to accelerate itself. OpenCV runs on various desktop operating systems as follows: Windows, Linux, macOS, FreeBSD, NetBSD, Open BSD. OpenCV runs on various mobile operating systems also which are as follows: Android, iOS, Maemo, BlackBerry

10. The user can get official releases from SourceForge or take the latest sources from GitHub. OpenCV uses CMake.

B. dlib library

Dlib is a general purpose cross-platform software library written in the programming language C++. Its design is heavily influenced by ideas from design by contract and component-based software engineering. Thus it is, first and foremost, a set of independent software components. It is open-source software released under a Boost Software License. Since development began in 2002, Dlib has grown to include a wide variety of tools. As of 2016, it contains software components for dealing with networking, threads, graphical user interfaces, data structures, linear algebra, machine learning, image processing, data mining, XML and text parsing, numerical optimization, Bayesian networks, and many other tasks. In recent years, much of the development has been focused on creating a broad set of statistical machine learning tools and in 2009 Dlib was published in the *Journal of Machine Learning Research*. Since then it has been used in a wide range of domains.

This system is divided in two parts. One is database creation part and other is live time face recognition. For database creation, we have created a simple web form that will be used to upload the image of the people allowed to have access to whatever the security system is for. This form has a background python code written using deep learning library to check whether the uploaded image is compatible with our algorithm or not. It also checks that one single image should not contain more than one person to avoid conflicts. It makes use of *get_frontal_face_detector()* function of dlib library to detect the faces in image and *shape_predictor()* of same library to access a *shape_predictor_68_face_landmarks.dat* a premade data file that contains all the possible human face shapes possible. This file is made available on the web by those whose expertise in the field.

Now comes the live time face recognition part. So when the person comes in front of camera, an image of the person is taken *cv2.OpenCapture()* of OpenCV library. First 30 frames are ignored so as to give time to the person to stay still for a better image. The image captured is made as an image file using *.read()* function and then temporarily stored in test database using *cv2.imwrite()*. Now the camera is released using *delete(camera)* to reduce battery consumption. Now both the images are stored in database, the people who are allowed to have access and the person trying to gain access currently.

The system performs checks as follows. First the allowed people database is accessed and then using for loop face encodings of all the images are created. To get face encodings we have created a function. This function first reads the image using *scipy.misc.imread()* and then shape of face in that image is taken which is recognised through our premade data file. Now the image is converted into an array using the function *np.array(face_recognition_model.compute_face_descriptor())* and these arrays of images are referred to as encodings. If, however, there are more than one or no face encodings in one particular image then the system produces an alert and stops. If not, then face encodings are generated for all images in the database. Same process happens for live time taken image stored in test database. We have created a function to compare face encodings of both the databases. This function returns *np.linalg.norm(known_faces - face, axis=1) <= TOLERANCE* that returns a binary value. If the value is 1 then it's a match and access is granted else denied. The TOLERANCE here is set by us. For this particular system we have set 0.6 tolerance which means that database image and live time image can be at max 6% different from each other. General System Architecture is displayed in Figure 1.

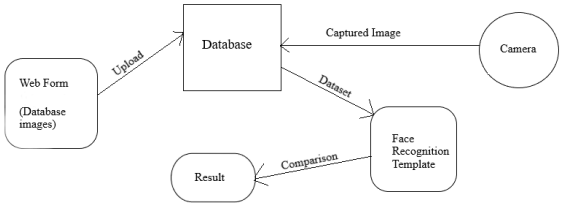

Figure 1. System Architecture

4. RESULTS AND DISCUSSIONS

For testing the mechanism, we are running our web form through xampp server and the database is also managed locally instead of cloud. The camera used is also the webcam installed in the laptop.

So first when we open our web form it will look like this. (Figure 2)

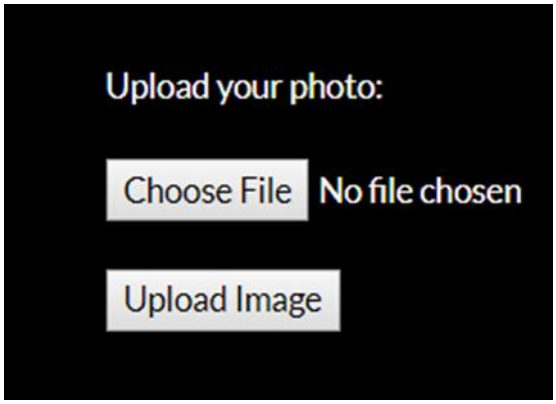

Figure 2. Web form

Now when the image is uploaded, if it is successfully accepted then following alert will be given. (Figure 3)

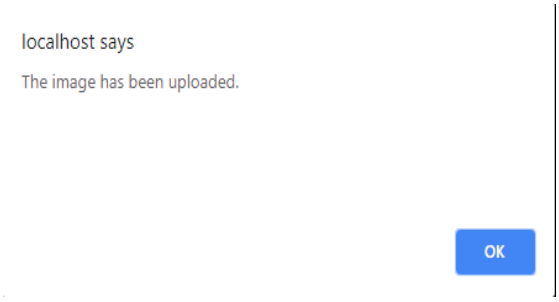

Figure 3. Success

If the image is not accepted, then the following or similar alerts will be shown. (Figure 4)

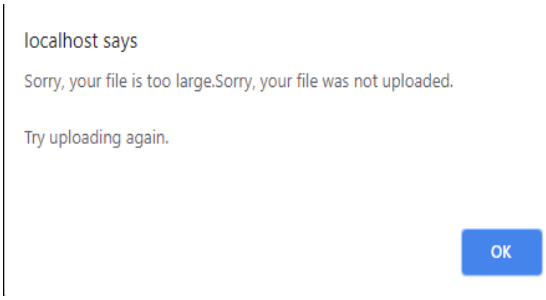

Figure 4. Fail

Once the database is created, we move onto the demonstration of face recognition part. Since we are doing it locally, we have simply tested the code on Python IDLE. While capturing the image it will display a text letting the person know that the image is being taken now. (Figure 5)

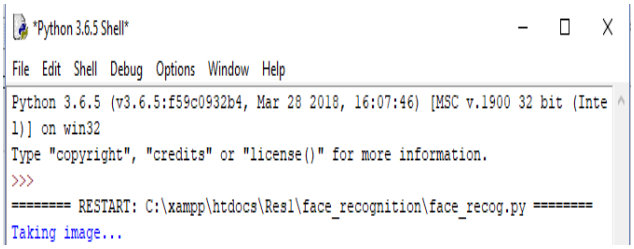

Figure 5. Image being taken

Now if the match is found then the access will be granted and following will appear. (Figure 6)

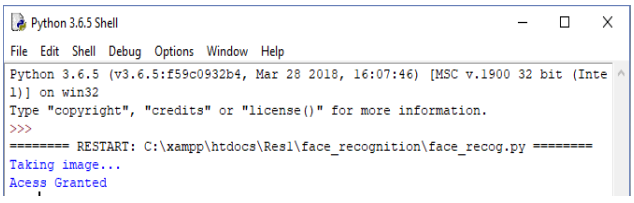

Figure 6. Access Grant

And if the match isn't found then the person is denied access. (Figure 7)

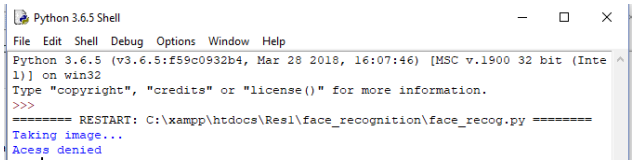

Figure 7. Access Denied

5. CONCLUSION

This system has designed a face recognition system for security purposes. The system performs accurately under normal conditions. In order for system to work properly, the database creation should be done with uttermost care. This is to make sure that all the authorized people are granted access and no outsider has the access.

Few modifications can be done in the system to make it more efficient. What we can do is while taking live time image if suppose there are two people present in the image then instead of showing an alert for that, it checks for both of them individually and grants or denies access accordingly. We can also apply neural regression machine algorithm in the system, then what will happen is that system will observe the pattern and will learn which person comes around which time and at that time instead of comparing with whole database, just compare the encodings with assumed person's encodings and if match is found then great and if isn't then it compares with whole database before displaying final result.

REFERENCES

[1] Dwi Ana Ratna Wati and Dika Abadianto, "Design of Face Detection and Recognition System for Smart Home Security Application" in 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering(ICITISEE).

[2] K. Shailaja and Dr. B. Anuradha, "Effective Face Recognition using Deep Learning based Linear Discriminant Classification" in 2016 IEEE International Conference on Computational Intelligence and Computing Research.

[3] Christian Galea and Rueben A. Farrugia, "Forensic Face Photo-Sketch Recognition Using a Deep Learning-Based Architecture" in November 2017 IEEE Signal Processing Letters, Vol. 24, No. 11.

[4] Umme Aiman and Virendra P. Vishwakarma, "Face Recognition using modified Deep Learning Neural Network" in 8th ICCNT 2017, IIT Delhi, India.

[5] Ping Wang, Wen-Hui Lin, Kuo-Ming Chao and Chi-Chun Lo, "A Face-Recognition Approach using Deep Reinforcement Learning Approach for User Authentication" in The Fourteenth IEEE International Conference on e-Business Engineering 2017.

[6] Xi Peng, Nalini Ratha and Sharathchandra Pankanti, "Learning Face Recognition from Limited Training Data using Deep Neural Network" in 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun Center, Cancun, Mexico.