Proceedings of the Fifth International Conference on Inventive Computation Technologies (ICICT-2020)
IEEE Xplore Part Number:CFP20F70-ART; ISBN:978-1-7281-4685-0

Face Recognition using Deep Neural Network with "LivenessNet"

Samana Jafri, Satish Chawan, Afifa Khan
Department of Electronics and Telecommunication
University of Mumbai
samana84@gmail.com, satyachavan@gmail.com, khanaara5454@gmail.com

***Abstract* - There is a continuous increase in the amount of population over the globe, and this, in turn, increases the number of complex datasets over a period. This necessitates improving artificial intelligence algorithms for better and accurate categorization of data. The most defining characteristic of the human body of the face. Every person's face is unique, although have the same structure such as noise, eyes, lips, etc. but it can vary strikingly. It's within this variance which lies the distinguishing characteristics that can be used to identify one person from another. Face recognition is a popular concept which is commonly used in surveillance cameras at public places for security purposes. The "Face Recognition using DNN with LivenessNet" presents a face recognition method based on deep neural networks for liveness. Any algorithm is considered to be efficient only if it is robust and accurate. It provides accurate results with face spoofing quickly and efficiently. The main advantage of using this technique is identifying the uniqueness in the datasets by capturing the real-time face data through different modes & jitter. Also providing accurate face recognition model which can be used for safety and security purpose.**

Keywords – Face recognition, Artificial intelligence, Deep Neural Networks, Computer Vision, OpenCV, and LivenessNet.

I. INTRODUCTION

Nowadays face recognition is one of the most active research areas of computer vision and machine learning field due to its potential value for practical application such as in the banking system, forensic science, automatic attendance, etc. With the advancement in digital technologies, there's an increasing demand for security to give access control. It uses different authentication methods for keeping all the information secure like method based on encrypted user name & password, smart card, biometrics, etc. Among all the latest approach is biometrics which measures the physical characteristics of people like iris, patterns on palm, fingerprints, voice and face. Each one is having different characteristics that are distinguished in its own way based on the requirement. Various face recognition algorithms have been proposed in the last few years, but due to high variabilities like pose, illumination, expression change, image resolution and other factors; face recognition has become too complex. Therefore, this field is still active for research purpose.

This paper describes a face recognition approach based on deep neural networks to evaluate human face detection with liveness to provide more robust results. First is the face detection which can be considered as a particular case of object-class detection. It searches for the position and dimensions of all the attributes such as eyes, nose, lips, the distance between them, etc. belonging to a class called a face. The frontal face detection is the main concept of this class object detection model.

DNN stands for Deep Neural Network. DNN is a complex neural network with more than 3 number of layers. It uses mathematical modeling to process data in hard ways. The structure & function of neural network in deep learning is similar to that of the human brain such as recognizing patterns of an image along with the way of input through different layers of interconnected neurons. Many scientists define deep neural networks as networks that have an input layer, an output layer and at least one hidden layer in between. Each layer performs specific types of sorting and ordering in a process that some refer to as "feature hierarchy." It is mainly used for unstructured or unlabeled data. The name "Deep Learning" itself represents the deep networks used to represent a specific form of ML algorithms to classify and order information.

Systems using face recognition models can be fooled easily by unreal faces and spoofing like exposing the photos of persons face through mobile phones or printed pictures. Therefore there's a need to handle such scenarios for making such systems more secure & efficient.

In this face-recognition approach, it first detects all the expected human facial features such as nose, lips, ears, forehead, etc. and determine the position with the distance between them in the given grey-level image. The uniformity and symmetry of the face are determined after a series of iterations and various facial features are tested to confirm their presence. Also, represent a face recognition as FPAF (Face Priority AutoFocus), which is a function that distinguishes all the human faces in the given frame so that the focal point will be set along with suitable exposure. For implementing this face recognition approach using deep neural networks, OpenCV is used. The main advantage of this approach is that it is highly accurate and it can be executed in real-time.

II. RELATED WORKS

Face recognition is a great area of research and development in modern science. In the past few decades, many theories and practical experiments were performed for face recognition but were found to provide unsatisfactory results. Different classification algorithms based on supervised learning have been developed and implemented at software level for labelling and classifying data over the past few years. Many kinds of research from a vast pool of specializations

have put a lot of effort and produced work related to this approach. Face recognition is now considered to be one of the best of all approaches for identifying a person as it doesn't require any human interference in the process of recognizing faces.

The architecture of neural networks can be derived in the form of a simple portion of the neuron in a human brain. These are necessarily weighted unions and integration of inputs that proceed through numerous non-sequential or non-linear functions. These neural networks are based on an iterative learning method which is popularly called as back-propagation and optimizer (such as stochastic gradient descent (SGD)). Deep Neural Networks (DNN) is based on simple neural network architectures and these consists of various hidden layers. These networks are very commonly used for categorization. Convolutional Neural Networks (CNN) present altogether a contrasting kind of structural proposal to the study of neural networks. The primary objective behind CNN is to make use of feed-forward networks coupled with convolutional layers which consist of local and global pooling layers. A. Krizhevsky made use of CNN, however, it made use of two-dimensional convolutional layers coupled with two dimensional feature space of the image [1]. One dimensional convolutional layers are used for text and sequences with word embedding as the input feature space. Recurrent Neural Networks (RNN) is the last type of deep learning architecture where outputs from the neurons are given back into the network as inputs for the next step. It has been effectively performed for Natural Language Processing (NLP) [8].

Viola and Jones were the first ones who proposed a work which was based on applying rectangular boxes for a face. However, it had many limitations as its feature size was big. The overall number of Haar_like features was 0.16 million in a 24 × 24 image and moreover, it was not handled for wild faces and frontal faces. After analysing and identifying the above issue, people from various specializations put a lot of efforts to initiate with additional complex features (HOG, SIFT, SURF, and ACF). Dlib is another well-known method for face detection that utilized support vector machine as a classifier. Also, combining the multiple detections that should be trained separately in different views is one of the simplest methods [4].

Multiple deformable models were applied by Zhu et al. to capture faces in contrasting views. Also, Shenet et al gave a retrieval based model integrated with different learning. These models demanded training and testing where it required more time as well as it had less efficiency. Garcia et al. developed a neural network in 2002 to locate the semi - frontal human faces in the complicated images.

The approach described in this paper takes all these ideas and theories into consideration and the proposed approach provides a much better and accurate results [4].

III. MAIN OBJECTIVE

Object recognition is considered one of the most popular computer technologies, which integrates image processing and computer vision and it interconnects with detecting examples of an entity such as human faces, buildings, trees, cars, etc. The key objective of face recognition algorithms is to decide and analyse whether there is any face existing in an image or not. In the past few decades, a lot of study and work has been done in the field of Face Recognition and Face Detection to make it more advanced and accurate, but a revolution took place in this field when Viola-Jones came up with a Real-Time Face Detector, which was capable of detecting faces in real-time with high accuracy. The initial and foremost step for face recognition is face detection and it is used to detect faces in the images. It is an integral part of object recognition and can be used in several important areas such as security, biometrics, law, statistical analysis, entertainment, safety, etc. It is used to recognize faces in real time for monitoring and tracking the location of a person or other entities. It is most commonly used in mobile cameras and DSLRs for identifying multiple appearances in the frame. The best example which can be derived from a popular day to day application is Facebook which also uses face recognition algorithms to detect faces in the images and recognize them [7].

The main objective of the proposed approach is to recognize face data of different people and identify them with accurate results and efficiently.

IV. FLOWCHART

Figure 1 shows the flowchart for training LivenessNet using both dataset ie. real & fake images:

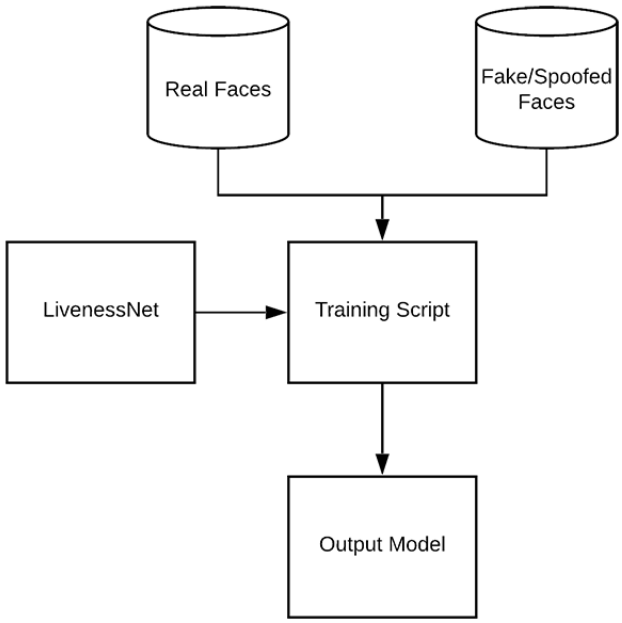

Figure 1: Flowchart for training model.

V. METHODOLOGY

- **Creation of dataset**
  The first step in building our face recognizer is to collect examples of each face to identify. There are various methods which can be used to create the datasets. The most practical way is to collect video/images of people by asking their physical presence to some special area/place defined for the purpose. This method can be repeated for a period of time in different conditions, moods, situations, etc. for adding variability & uniqueness in the dataset. The other way is to collect videos & images of people from online platforms in case of public figures or famous

personalities. At last, way to create a custom dataset is to manually search and download images to create face dataset from them. Among all the methods mentioned above, this is the most hectic way to create or collect dataset which will consume more time and energy. Hence, the first one is preferred in cases such as schools, colleges, organizations, etc where it will be easy to set up [6]. But in some cases, the later ways would be used.

- **Training the face recognizer**
  First, a python script is created for training the data and place it in the same folder as the datasets as shown in figure 2 below and need to import all the required libraries such as OpenCV, OS, NumPy, etc. before starting the script. Further, a function is created that will load the data to be trained from the dataset folder as shown in figure 3 below. This function will also capture the faces from the datasets. After completing the python script and run it and this will train the entire dataset for facial recognition as shown in figure 4.

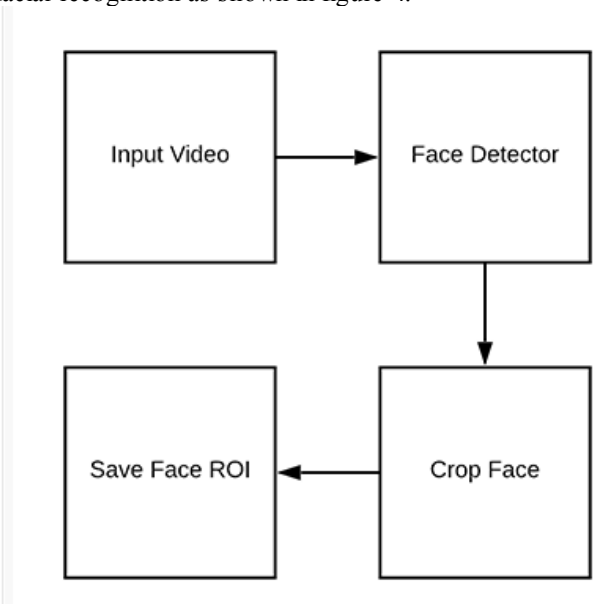

Figure 2: Block Diagram for preparing dataset for "livenessNet".

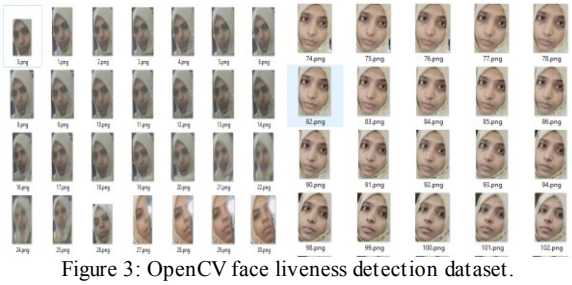

Figure 3: OpenCV face liveness detection dataset.

- **Camera Integration**
  Camera is one of the most important components in the facial recognition process. The camera will be used to capture images of a person in real time so that they can be compared with the trained dataset. The camera will be integrated with the script that is generated for facial recognition in order to capture images in real time.

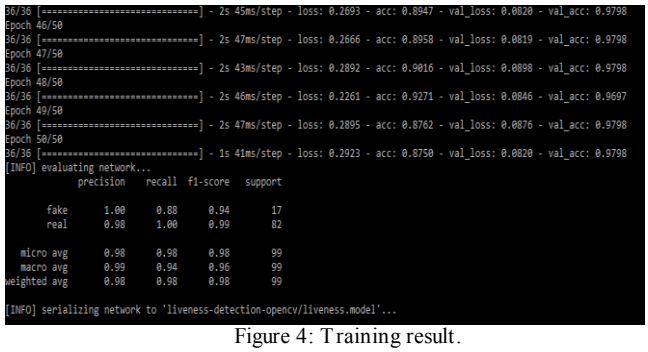

Figure 4: Training result.

- **Implementation of face recognizer**
  After creating a productive dataset, training the dataset and integrating the camera module, and the implementation of our face recognizer. As soon as a person arrives in front of the camera, it will capture the image of the person in real time and compare its face with the trained dataset. The captured image will then be compared with the trained dataset and obtain a result with an appropriate accuracy value. The accuracy value may vary depending on the productivity of the dataset and also on the parameters provided in the training script. In order to obtain accurate results, must make sure that all the parameters are provided correctly and the dataset is trained properly. If a face that entered the view of the camera is unknown and could not be identified when compared with the trained dataset, the intruder is marked as "Unknown".

- **Liveness detection to prevent spoofing**
  The numbers of approaches for liveness detection are as follows:

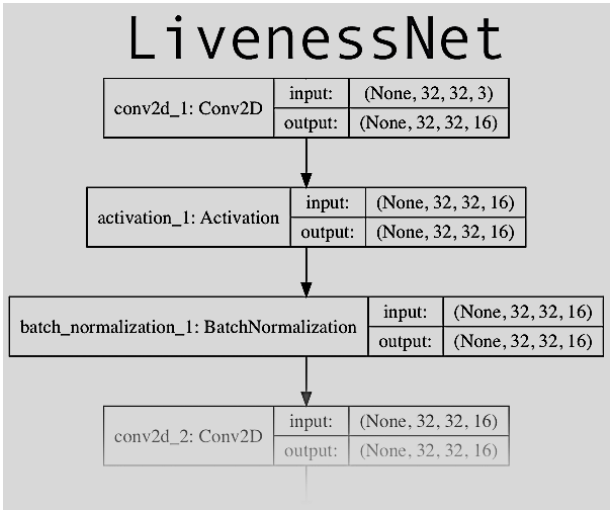

Figure 5: Livenessnet architecture to detect real and fake faces [6].

The different techniques used to detect liveness of the image/frame are as follows:

1. **In Variable focusing analysis,** the changes in pixel values between two consecutive frames/images are examined.
2. **Frequency analysis** - In this technique, it uses Fourier transformation to examine faces in the image.
3. **Texture analysis** - This technique includes computing LBPs i.e. Local Binary Patterns in the face

regions of the image along with using an SVM to classify fake or real faces.

4. **In Heuristic-based algorithms,** the movement of eyes, lips and direction are tracked to detect the real & fake images.
5. **Optical Flow algorithms -** It examines the optical flow properties and differences in the frames/images.
6. The technique used in Apple's iPhone is similar to **3D face shape** which is used to help systems to detect the real & fake images**.**
7. **Combinations of the above** techniques are used to enable systems engineers to build a correct model for liveness detection in face recognition systems.

The created dataset is then trained and the same procedure is followed as above. After training dataset, the python script is implemented to detect the real and fake images [6] using LivenessNet architecture as shown in figure 5.

VI. RESULTS AND DISCUSSION

The overall accuracy achieved is **98.5%** approximately for our model to recognize face along with liveness on the validation dataset as shown in figure 6. Created a Python + OpenCV script for demonstrating the full face recognition with liveness detection pipeline in action to load the model for liveness detection with face recognition and applied it to the frames/images recorded in real-time [6].

The main fallback of our proposed method is the limited dataset and that too of an only single person which is around **393 images** out of which 336 belongs to the real class & 57 belongs to the fake class).

The future work would be to gather more & more training dataset in different conditions and situations. Like images/frames containing faces with different skin tone and size/shapes. Another point is that the fake dataset used to train our model was only based on exposing photos from mobile phones, tablets, etc and not on printed images or pictures. Hence, different types of data should be included in fake dataset.

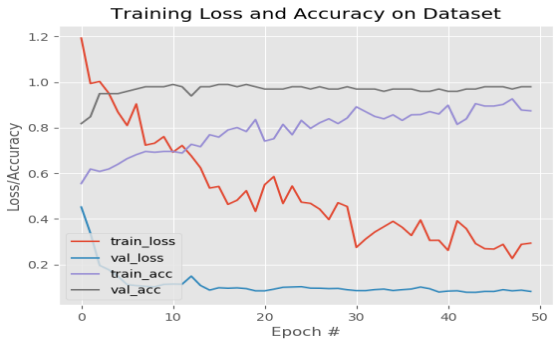

Figure 6: Output curves of training and loss history.

The proposed network is purposely not so deep to ensure that:

1. The chances of overfitting are reduced on our small dataset and the model should run in real-time. [3].

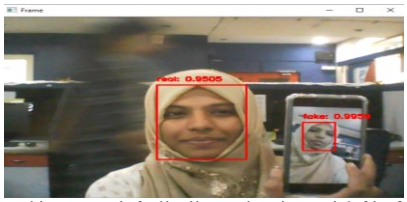

Figure 7: Processed image result for liveliness showing real & fake face.

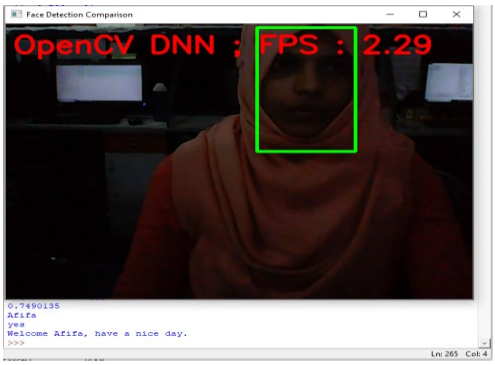

Figure 8: Processed image results for face recognition with 75 % probability.

VII. CONCLUSION

Hence by using the proposed approach, can recognize faces with high robustness and accuracy. It also includes the liveness detector for detecting fake & real faces by using the "LivenessNet" which is based on the CNN model.

The first step in building the proposed face recognition model with liveness detector is to collect dataset for training the LivenessNet model by holding a mobile phone in front of the camera. The real dataset is extracted from the video recorded directly with the camera. For recognition, the dataset is the same as the real dataset. After preparing the dataset, first, the liveness of the frame is detected using LivenessNet model. Then the detected real face is recognized using the deep learning model build on the training dataset.

As shown in figure 7 & figure 8, the proposed method is able to recognize faces with real & fake images.

VIII. REFERENCES

[1] Kamran Kowsari, Donald E. Brown, Laura E. Barnes, (2018). "RMDL: Random Multimodal Deep Learning for Classification", ICISDM '18, April 9–11, 2018, Lakeland, FL, US

[2] Mengjia Yan Horizon, Mengao Zhao Horizon, Zining Xu Horizon (2019). "VarGFaceNet: An Efficient Variable Group Convolutional Neural Network for Lightweight Face Recognition".

[3] C.Anuradha, R.Kavitha, A.V.Allin Geo, S.R.Srividhya (2019). "Liveness Detection with Opencv", International Journal of Innovative Technology and Exploring Engineering (IJITEE).

[4] Manik Sharma, J Anuradha, H KManne and G S Kashyap (2017). "Facial detection using deep learning", School of Computing Science and Engineering, VIT University, Vellore - 632014, India (DOI: 10.1088/1757-899X/263/4/042092)

[5] Lei Zhang,Ji Liu, Bob Zhang, David Zhang,Ce Zhu (2019). "Deep Cascade Model based Face Recognition: When Deep-layered Learning Meets Small Data", IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. X, NO. X, AUG 2019.

[6] Adrian Rosebrock (2018). "How to build a custom face recognition dataset", Pyimagesearch.

[7] Divyansh Dwivedi (2018). "Face Detection For Beginners", Towards Data Science (https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9).

[8] Jason Brownlee (2017). "How to Use Word Embedding Layers for Deep Learning with Keras", Machine Learning Mastery (https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/).