# Livenessnet with Hardware Interface for higher Security

Dr. Anil Kumar D
Associate Professor
*Electronics and Communications*
*BMS Institute of Technology and Management*
*Banglore, India*

Harshitha R
*Electronics and Communications*
*BMS Institute of Technology and Management*
*Banglore, India*

Vinutha P
*Electronics and Communications*
*BMS Institute of Technology and Management*
*Banglore, India*

Amulya C S
*Electronics and Communications*
*BMS Institute of Technology and Management*
*Banglore, India*

Sujay N
*Electronics and Communications*
*BMS Institute of Technology and Management*
*Banglore, India*

***Abstract*→ This paper presents a simple and easy face recognition system that includes liveness detection. The core programming for the face recognition and liveness detection is written in python programming which has inbuilt open end machine learning libraries such as TensorFlow and Keras which is high level neural network library. These help in training the data set. The hardware interfacing involves basic setup showing the working of the liveness system, which opens and closes the door when recognizable face is read.**

***Keywords— liveness, neural networks, Biometrics, training, Python, ESP controller.***

## I. INTRODUCTION

One of the most active study fields in the field of computer vision and machine learning is face recognition because of its potential use in practical applications like banking, forensic science, automatic attendance, etc. The need for security to offer access control is growing as digital technologies improve. It uses a variety of authentication techniques, such as a system centred on encrypted user names and passwords, smart cards, biometrics, etc., to keep all information secure. The new approach includes biometrics, which evaluate a person's bodily characteristics like iris, palm shape, fingerprint, speech, and face. Each one has unique qualities that allow it to be distinguished in a certain way depending on the situation. This product provides a face recognition method based on deep neural networks to analyse human face detection vividly in order to deliver more reliable results. Next, it is important to consider the face's identification as a unique object-class identification event. It searches for the location, size, and spacing of all the features that make up the class of "face features," such as the eyes, nose, and mouth. It examines the placement, size, and spacing of every feature that makes up a face, including the eyes, nose, and lips. Frontal face detection is the primary tenet of this class of object detection methodology [1].

## II. RELATED WORKS

Throughout the years there has been need and improvements in the face recognition systems. In [1] there has been review on all the traditional systems that were used which includes pattern recognition using the infrared rays, the oldest face recognition system used rays to define the features of the face and worked by recognizing those patterns. In [2], there's a discussion on the critical features necessary for accurate recognition, [3] also shows its findings in the dataset for the face recognition. [4] Discusses a neural network approach for the system. These systems depend immensely on the lighting and styling of the faces for recognition. This poses as a disadvantage as it brings a need for training the system for multiple inputs of the same space. These dependencies can be seen in [5, 7]. With the pandemic, the whole of workspace and education was made online. This brings a need for systems which can detect the liveness of the person. This is very necessary when few tests that are online require monitoring. Hence we are proposing the face recognition system with liveness detection.

## III. METHODOLOGY

To reduce such issues, live detection has been incorporated into the system. To ensure that only live face samples are saved for enrolment or authentication, live detection technology examines the face for physiological indicators of life. For our suggested system, we will approach the detection of brightness as a binary classification issue. We will train a convolutional neural network that can discriminate between authentic faces and fraudulent or faked faces using an input image. Figure 1 below shows the block diagram of the system[1].

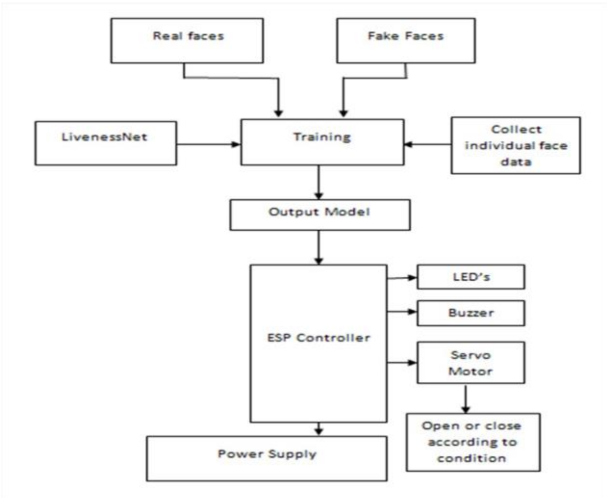
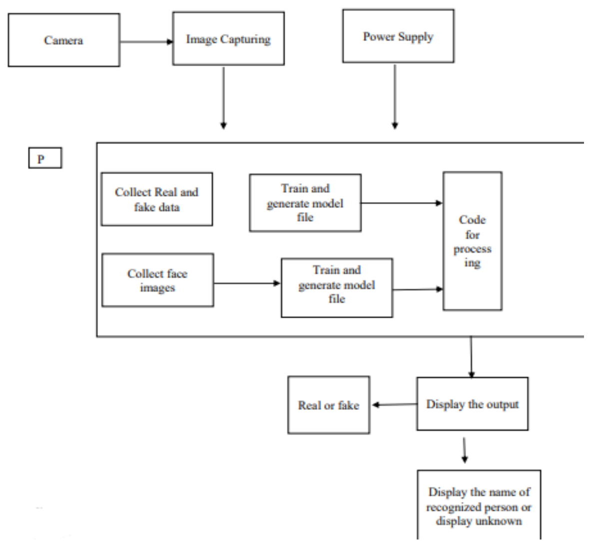



Figure 1. Block diagram of the system

### A. *Steps involved in training the system.*

- Creation/Collecting Of Dataset
- Face Embedding
- Training The Face Recognizer
- Recognize The Face
- Collection On Real Data Set
- Collect Of Fake Dataset
- Training Livnessnet (Fake Or Real Model File)
- Prediction Output

### B. *Architecture*




Architecture of system layers

### C. *Algorithm*

Convolution Neural Networks (CNN)

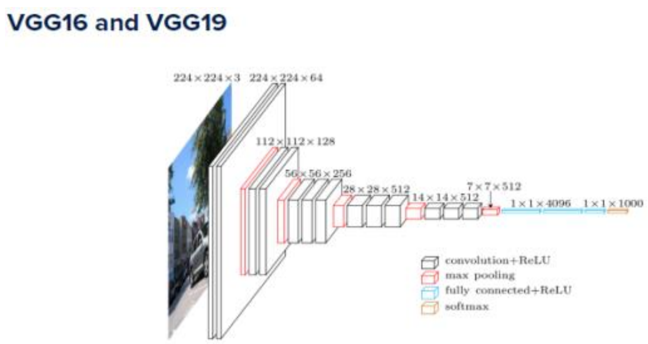



Figure 3: Visualization of the VGG architecture

Each layer in a CNN applies a different set of filters, typically hundreds or thousands of them, and combines the results, feeding the output into the next layer in the network. During training, a CNN automatically learns the values for these filters. In the context of image classification, our CNN may learn to:

- Detect edges from raw pixel data in the first layer.
- Use these edges to detect shapes (i.e., "blobs") in the second layer.
- Use these shapes to detect higher-level features such as facial structures, parts of a car, etc.in the highest layers of the network.

The last layer in a CNN uses these higher-level features to make predictions regarding the Contents of the image. In terms of deep learning, an (image) convolution is an element-wise multiplication of two matrices followed by a sum.

- Take two matrices (which both have the same dimensions).
- Multiply them, element-by-element (i.e., not the dot product, just a simple multiplication).
- Sum the elements together.

## IV. WORKING

### A. *Software requirement*

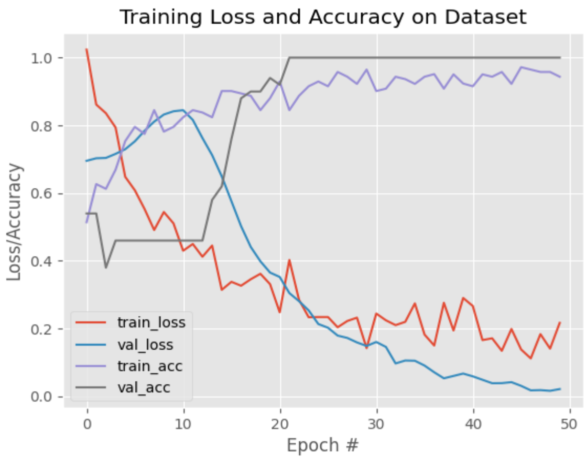
- Keras
- Tensorflow
- Tkinter
- Imutils
- Numpy
- OpenCV

The python code involves all the layers for activation till running. The code runs an epoch of 50 for the recognition. If necessary to improve the efficiency of the system, the epoch value can be increased in the source code. However this takes longer running time and more storage spaces.

The layer types involved are as follows

- Convolutional (CONV)
- Activation (ACT or RELU, where we use the same of the actual activation function)
- Pooling (POOL)
- Fully-connected (FC)
- Batch normalization (BN)
- Dropout (DO)

Once the code is executed, the application window opens and he epochs and efficiency graphs can be seen.



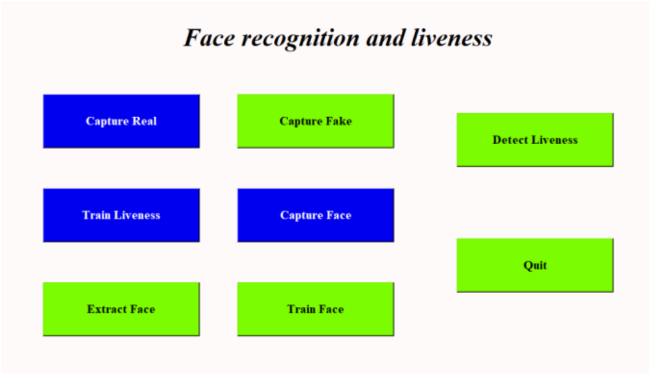
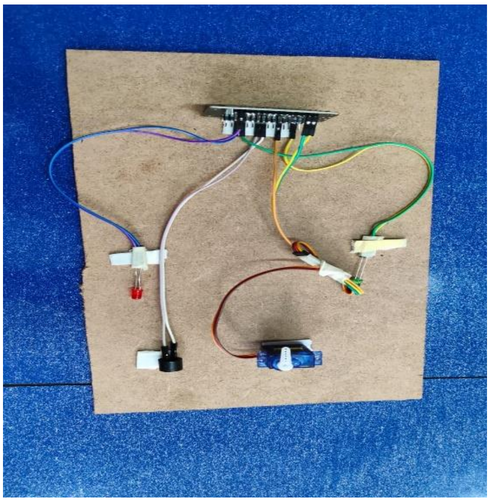



Figure 4. The application window and the training plot

The dataset of the faces must be trained and stored in dataset folder of the code. The dataset must involve both the real and fake dataset.

The hardware involves a basic demo of opening and closing of a door when liveness is detected. The face recognition is trained by the face embedding.



The hardware is interfaced through the aurdino. But the hardware can be implemented in any way as long the program is executed as the aurdino program defines the working of the board.

## V. Future Work

More and more training datasets in various settings and circumstances will be collected in the future. Images/frames showing faces of various skin tones, sizes, and shapes, for example. Another aspect to note is that the fictitious dataset we used to train our model was based solely on exposing photos from mobile phones, tablets, and other devices, rather than printed images or pictures. Hence, different types of data should be included in fake dataset. The work can be developed to an app that involve monitoring of students during tests and exams that are conducted online.

## VI. Conclusion

The proposed method for detecting people can recognize faces with excellent accuracy and resilience. It also features a liveness detector that uses the CNN-based "Liveness Net" to distinguish between phone and real faces. The initial stage in developing the suggested liveness detector facial recognition model is to gather data for training the Liveness Net model by holding a smartphone in front of the camera. The genuine dataset is derived from footage captured with the camera directly. The dataset is identical to the genuine dataset for recognition. The liveness of the frame is initially recognized using the Liveness Net model after the dataset has been prepared. The deep learning model built on the training dataset is then used to recognize the discovered real face[1].

## References


[1] Samana Jafri, Satish Chawan, Afifa Khan, “Face Recognition using Deep Neural Network with “LivenessNet”,” *Proceedings of the Fifth International Conference on Inventive Computation Technologies, 2020.*

[2] R.S.Ghiass, O. Arandjelović, H. Bendada, and X. Maldague, “Infrared face recognition: A comprehensive review of methodologies and databases,” *A Treatise on Electricity and Magnetism*, 3rd edition, vol. 2, pp.68–73,2014.

[3] N. Abudarham, L. Shkiller, and G. Yovel, “Critical features for face recognition*,”Cognition*, vol. 10, pp. 73-83, 2019.

[4] I. Masi, A. T. Tran, T. Hassner, G. Sahin, G. Medioni, “Face-specific data augmentation for unconstrained face recognition*,” International Journal of Computer Vision (IJCV*), vol. 127, pp. 642-660, 2019.

[5] C. Huang, Y. Li, C. C. Loy, “Deep imbalanced learning for face recognition and attribute prediction,” *IEEE transactions on pattern analysis and machine intelligence*, pp. 1-14, 2019.

[6] J. Deng, D. Zhang, Y. Deng, J. Guo, “Lightweight face recognition challenge," *Proceedings of the IEEE International Conference on Computer Vision Workshops*, vol. 10, pp. 1-10, 2019.

[7] S. L. Happy, A. Dasgupta, A. George and A. Routray, “A Video Database of Human Faces under Near Infra-Red Illumination for Human Computer Interaction Applications,” *IEEE Proceedings of 4th International Conference on Intelligent Human Computer Interaction*, Kharagpur, India, 2012.

[8] K. Shajahan, R. D. Rai, Ravishankara, “Face Recognition Using Livenessnet,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 7, no. 4, pp. 143-148, 2021.

[9] M. Sharma, J. Anuradha, H. K. Manne, G. S. C. Kashyap, “Facial detection using deep learning,” *IOP Conference Series Materials Science and Engineering*, vol. 10, pp. 1-10, 2017.

[10] Y. Singh, R. Toldo, L. Magri, S. Fantoni, A. Fusiello, “Method for 3D modelling based on structure from motion processing of sparse 2D images,” *United States Patent*, vol. 10, pp. 1-24, 2019.

[11] A. Bijoura, S. Banerjee, K. S. Pandey, Dr A. Arivoli, “Facial recognition using deep learning,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 07, no. 08, pp. 2719-2726, 2020