

# Face Recognition Using Livenessnet

Shajahan K<sup>1</sup>, Rathish Rai D<sup>2</sup>, Ravishankara<sup>3</sup>

<sup>1,2</sup>UG Scholar, Department of Computer Science and Engineering, Srinivas Institute of Technology, Mangalore, Karnataka, India

<sup>3</sup>Associate Professor, Department of Computer Science and Engineering, Srinivas Institute of Technology, Mangalore, Karnataka, India

## ABSTRACT

### Article Info

Volume 7, Issue 4

Page Number: 143-148

### Publication Issue :

July-August-2021

### Article History

Accepted : 06 July 2021

Published : 13 July 2021

Every person's face is unique, although have the same structure such as nose, eyes, lips, etc. but it can vary strikingly. It's within this variance which lies in 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. With the advancement of digital technologies, the demand for security to provide access control is increasing. It uses various methods of authentication to keep all details secure, such as a system focused on encrypted user name & password, smart card, biometrics, etc. 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. It provides accurate face recognition model which can be used for safety and security purpose.

Keywords: LivenessNet, Face Recognition, DNN, CNN, VCG Architecture

## I. INTRODUCTION

Face recognition is one of the most active research areas in the field of computer vision and machine learning. With the advancement of digital technologies, the demand for security to provide access control is increasing. It uses various methods of authentication to keep all details secure, such as a system focused on encrypted user name & password, smart card, biometrics, etc. Due to its potential value for practical applications such as banking, forensic

science, automatic attendance, etc., The new solution incorporates biometrics that assess people's physical features, such as iris, palm shapes, fingerprints, speech 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.

In order to provide more robust results, this product describes a face recognition approach based on deep neural networks to evaluate human face detection with vividness. Next, the identification of the face should be viewed as a special object-class identification event. It looks for the position and dimensions of all the features belonging to a class called a face, such as eyes, nose, mouth, the distance between them, etc.

## II. LITERATURE SURVEY

In the research done by Lior Shkiler and Galit Yovel, Face recognition is a computationally challenging task that humans perform effortlessly. This remarkable ability was better for familiar faces than unfamiliar faces. To account for humans superior ability to recognize familiar faces, current theories suggest that different features are used for the representation of familiar and unfamiliar faces. In the current study, they applied a reverse engineering approach to reveal which facial features are critical for familiar face recognition. In contrast to current views, they have discovered that the same subset of features that are used for matching unfamiliar faces, are also used for matching as well as recognition of familiar faces. Also show that these features are also used by a deep neural network face recognition algorithm. So they proposed a new framework that assumes similar perceptual representation for all faces and integrates cognition and perception to account for humans' superior recognition of familiar faces.

In the research done by Deng and Jiankang, face representation using Deep Convolutional Neural Network (DCNN) embedding is the method of choice for face recognition. Current state-of-the-art face recognition systems can achieve high accuracy on

existing in-the-wild datasets. However, most of these datasets employ quite limited comparisons during the evaluation, which does not simulate a real-world scenario, where extensive comparisons are encountered by a face recognition system. They proposed two large-scale datasets and define an extensive comparison metric for an unbiased evaluation of deep face recognition models. To ensure fair comparison during the competition. They define light-model track and large-model track, respectively. Each track has strict constraints on computational complexity and model size. To the best of our knowledge, this was the most comprehensive and unbiased benchmarks for deep face recognition.

In the research done by Krishnapriya, paper report on a methodical investigation into differences in face recognition accuracy between African-American and Caucasian image cohorts of the MORPH dataset. They found that, for all four matchers considered, the impostor and the genuine distributions are statistically significantly different between cohorts. For a fixed decision threshold, the African-American image cohort has a higher false match rate and a lower false non-match rate. ROC curves compare verification rates at the same false match rate, but the different cohorts achieve the same false match rate at different thresholds. This means that ROC comparisons are not relevant to operational scenarios that use a fixed decision threshold. They showed that, for the ResNet matcher, the two cohorts have approximately equal separation of impostor and genuine distributions. Using ICAO compliance as a standard of image quality, They found that the initial image cohorts have unequal rates of good quality images. The ICAO-compliant subsets of the original image cohorts show improved accuracy, with the main effect being to reducing the low-similarity tail of the genuine distributions.

In the research done by Manik Sharma and J Anuradha, they have observed that Facebook has developed an uncanny ability to recognize people in photographs. They had to tag people in photos by clicking on them and typing their name. Now as soon as we upload a photo, Facebook tags everyone on its own. Facebook can recognize faces with 98% accuracy which is pretty much as good as humans can do. This technology is called Face Detection. Face detection was a popular topic in biometrics. They have surveillance cameras in public places for video capture as well as security purposes. The main advantages of this algorithm over other are uniqueness and approval. They need speed and accuracy to identify. But face detection is really a series of several related problems: First, look at a picture and find all the faces in it. Second, focus on each face and understand that even if a face is turned in a weird direction or in bad lighting, it was still the same person. Third select features which can be used to identify each face uniquely like size of the eyes, face etc. Finally, compare these features to data they have to find the person name. As a human, your brain is wired to do all of this automatically and instantly. In fact, humans are too good at recognizing faces. Computers are not capable of this kind of high-level generalization this process separately. The growth of face detection is largely driven by growing applications such as credit card verification, surveillance video images, authentication for banking and security system access.

In the research done by Huang and Chen Data for face analysis often exhibit highly-skewed class distribution, i.e., most data belong to a few majority classes, while the minority classes only contain a scarce amount of instances. To mitigate this issue, contemporary deep learning methods typically follow classic strategies such as class re-sampling or cost-sensitive training. In this paper, we conduct extensive and systematic experiments to validate the

effectiveness of these classic schemes for representation learning on class-imbalanced data. We further demonstrate that more discriminative deep representation can be learned by enforcing a deep network to maintain inter-cluster margins both within and between classes. This tight constraint effectively reduces the class imbalance inherent in the local data neighborhood, thus carving much more balanced class boundaries locally. We show that it is easy to deploy angular margins between the cluster distributions on a hypersphere manifold. Such learned Cluster-based Large Margin Local Embedding (CLMLE), when combined with a simple k-nearest cluster algorithm, shows significant improvements in accuracy over existing methods on both face recognition and face attribute prediction tasks that exhibit imbalanced class distribution.

In the search done by Masi and Lacopo We identify two issues as key to developing effective face recognition systems: maximizing the appearance variations of training images and minimizing appearance variations in test images. The former is required to train the system for whatever appearance variations it will ultimately encounter and is often addressed by collecting massive training sets with millions of face images. The latter involves various forms of appearance normalization for removing distracting nuisance factors at test time and making test faces easier to compare. We describe novel, efficient face-specific data augmentation techniques and show them to be ideally suited for both purposes. By using knowledge of faces, their 3D shapes, and appearances, we show the following: (a) We can artificially enrich training data for face recognition with face-specific appearance variations. (b) This synthetic training data can be efficiently produced online, thereby reducing the massive storage requirements of large-scale training sets and simplifying training for many appearance variations Together, with additional technical novelties, we

describe a highly effective face recognition pipeline which, at the time of submission, obtains state-of-the-art results across multiple benchmarks.

### III. OBJECTIVE

The aim of our project is to discuss liveness detection, including what it is and why we need it to improve our face recognition systems. From there we will review the dataset we will be using to perform liveness detection, including How to build to a dataset for liveness detection and our example real versus fake face images. We will also review our project structure for the liveness detector project as well. In order to create the liveness detector, we will be training a deep neural network capable of distinguishing between real versus fake faces.

### IV. METHODOLOGY

#### CREATION/COLLECTING OF DATASET

Collecting images of each face to identify is the first stage in creating our face recognizer. The datasets can be created using a variety of techniques. The most feasible method is to collect video/images of people by requesting their physical presence in a designated area/place. This method can be used to add diversity and uniqueness to a dataset by repeating it over time in varied conditions, moods, situations, and so on. In the case of public figures or famous persons, another option is to collect videos and photographs of them from web sites. Finally, manually searching and downloading photographs to construct a face dataset from them is a way to create a custom dataset.

#### FACE EMBEDDINGS

By creating face embeddings you are converting a face image into numerical data. That data is then represented as a vector in a latent semantic space. The closer the embeddings are to each other in the latent space, the more likely they are of the same

person Having ROI for only faces present in the captured image dataset

#### TRAINING THE FACE RECOGNIZER

Before starting the script, you must first build a python script for training the data and save it in the same folder as the datasets. You must also import all of the essential libraries such as OpenCV, OS, NumPy, and others.

In addition, a function is written to load the data for training from the dataset folder. Faces from the datasets will also be captured by this function. After completing the python script and running it, the full dataset will be trained for facial recognition.

#### RECOGNIZE THE FACE

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".

#### COLLECTION ON REAL DATA SET

Collecting images of each face to identify for a real image, i.e. having a person standing in front of the camera and collecting of data and save it in real data set folder

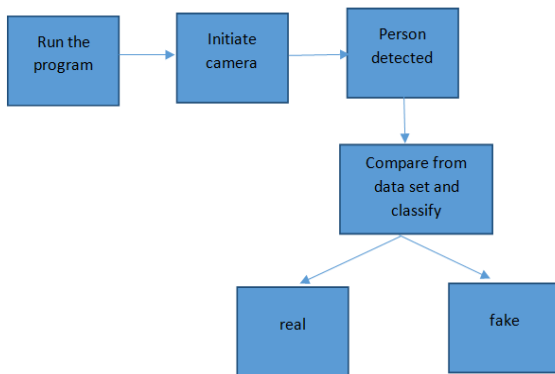
#### COLLECT OF FAKE DATASET

Collecting images of each face to identify for a fake image, i.e. having a mobile taken picture of a person and showing to the camera and collecting of data and save all the pictures in the fake dataset folder

**TRAINING LIVENESS DATA**

**PREDICTION OUTPUT**

**DECTION OF REAL AND FAKE IMAGES**



**ALGORITHM USED: CNN**

**Convolutional Neural Networks (CNN)**

**VGG16 and VGG19**

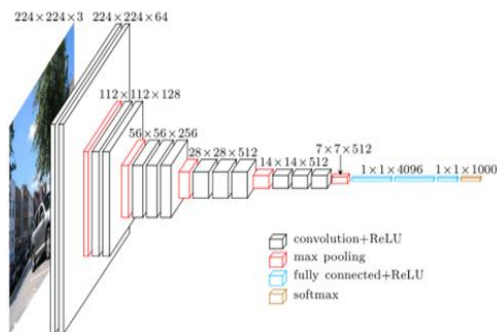


Fig 1 : A Visualization of VGG Architecture

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.

**Layer types:**

There are many types of layers used to build Convolutional Neural Networks, but the ones you are most likely to encounter include:

- 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)

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.

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

**V. ANALYSIS**

In this 100 augmentations of single image will be captured in 100 different angles. So that it will be more accurate during detection of images.



Fig 2 : training loss and accuracy on dataset.

Person name	No. of Testing augmentations	Result accuracy for fake	Result Accuracy for real
Person1	100	1	0.95
Person2	100	1	0.95
Person3	100	1	0.96
Person4	100	0.97	0.95
Person5	100	0.98	0.95
Person6	100	1	0.96
Person7	100	0.98	0.96
Person8	100	0.98	0.95
Person9	100	1	0.95
Person10	100	1	0.96

Table 1.1 : Result Analysis Table

Average overall accuracy for real=0.9580=95.80%

Average overall accuracy for fake=0.9910=99.10%

## 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 "LivenessNet" 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 LivenessNet 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 LivenessNet 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.

## VII. REFERENCES

- [1]. Abudarham, Naphtali, LiorShkiller, and GalitYovel. "Critical features for face recognition." *Cognition* 182 (2019): 73-83
- [2]. Deng, Jiankang, et al. "Lightweight face recognition challenge." *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2019.
- [3]. Krishnapriya, K. S., et al. "Characterizing the variability in face recognition accuracy relative to race." preprint arXiv:1904.07325 (2019).
- [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]. Huang, Chen, et al."Deep imbalanced learning for face attribute prediction." *IEEE transactions on pattern analysis and machine intelligence* (2019).
- [6]. Masi, Iacopo, et al."Face-specific data augmentation for unconstrained face recognition." *International Journal of Computer Vision* 127.6-7 (2019): 642-667

### Cite this article as :

Shajahan K, Rathish Rai D, Ravishankara, "Face Recognition Using Livenessnet", *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, ISSN : 2456-3307, Volume 7 Issue 4, pp. 143-148, July-August 2021. Available at

doi : <https://doi.org/10.32628/CSEIT217433>

Journal URL : <https://ijsrcseit.com/CSEIT217433>