**Face Recognition System**

Major Project Report

Submitted For the Partial Fulfillment of the degree of

## Bachelor Of Technology

In

### DEPARTMENT OF COMPUTER SCIENCE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | By |  |  |  |
|  | **Sandeep Kumar Jaiswal** |  |  | **RAVI KUMAR** |
|  | **(Reg No. 17J970032)** |  |  | **(02051203117)** |
|  | Under Supervision of |  |  |  |
|  | **Ms. Sonia**  (HOD CSE department) |  |  |  |



COMPUTER SCIENCE DEPARTMENT AKIDO COLLEGE OF ENGINEERING (AFFILIATED TO MDU, ROHTAK)

Bahadurgarh, Haryana

### JUNE, 2021

# Acknowledgement

All three of us express our deep gratitude to Ms. Sonia , Head of Department, Information Technology for his valuable guidance and suggestions throughout our project work and research and development.

|  |  |  |
| --- | --- | --- |
|  | **Sandeep kumar Jaiswal** |  |
|  | **(Reg No. 17J970032)** |  |

# Abstract

Face recognition has rapidly evolved in recent times and popularly used in biometrics devices. In this chapter, author categorizes various face recognition techniques which are local, holistic, hybrid approaches. This categorization helps in understanding different methods and their relation to developed solutions. “Modern technology gives power not only to the police but empowers thieves too”. This statement leads to an ever-evolving technological world. Facial recognition techniques have some drawbacks, like they do not catch the non-real image. Image spoofing is the term used when someone uses a photo of a person in hardcopy or mobile phone to bypass the facial recognition test. To make system more secure, author perform liveness detection. It includes various approaches like optical flow algorithms, frequency analysis, variable focusing analysis, heuristic analysis (like eye blinking based on Conditional Random Fields), texture analysis and a combination of two or more approaches for better results. A review of latest works regarding face recognition and liveness test is presented. The main objective is to provide a simple path for future development of secured face detection methods.

# Certificate

This is to certify that Mr. Sandeep Kumar Jaiswal **(Reg No. 17J970032)** of CSE, 4th year has satisfactorily completed their work in the Major Project based on “Face Recognition System” under the guidance of Ms. Sonia. (Major Project Mentor) .

# Table of Contents

Cover Page 1

[Acknowledgement 2](#_TOC_250027)

[Abstract… 3](#_TOC_250026)

[Certificate 4](#_TOC_250025)

[Table of Contents 5](#_TOC_250024)

[List of Figures 7](#_TOC_250023)

1. Introduction 8
   1. [Objective… 12](#_TOC_250022)
   2. [Problems Discussed… 12](#_TOC_250021)
2. Literature Survey… 13
   1. [liveness detection 13](#_TOC_250020)
   2. [Different Approaches for liveness detection 16](#_TOC_250019)
      1. [Frequency and Text Based Analysis… 16](#_TOC_250018)
      2. [Variable Focused Based analysis… 17](#_TOC_250017)
      3. [Movement of the eye based analysis… 19](#_TOC_250016)
      4. [Optical Flow Based analysis: 20](#_TOC_250015)
      5. [Blinking Based analysis… 21](#_TOC_250014)
      6. [Component Dependent Descriptor based analysis… 22](#_TOC_250013)
      7. [3D Face Shape based analysis: 23](#_TOC_250012)
      8. [Binary Classification based analysis… 24](#_TOC_250011)
      9. [Scenic Clues based analysis… 25](#_TOC_250010)
      10. [Lip Movement based analysis: 26](#_TOC_250009)
      11. [Context Based analysis… 27](#_TOC_250008)
      12. [Combination of Standard Techniques based analysis: 27](#_TOC_250007)
   3. [Local Approach… 28](#_TOC_250006)
   4. [Holistic Approach… 31](#_TOC_250005)
   5. [Hybrid Approach… 33](#_TOC_250004)
   6. [Techniques used… 37](#_TOC_250003)
      1. [OpenCV… 37](#_TOC_250002)
      2. [Keras… 38](#_TOC_250001)
      3. [Deep Learning… 38](#_TOC_250000)
3. Proposed Solution… 40
4. Analysis… 49
5. Result… 53
6. Conclusion… 54
7. References… 55

# List of Figures

Fig 1: Image spoofing attacks… 9

Fig 2. Crucial elements of the typical face recognition system… 12

Fig 3. Face detection methodology… 13

Fig 4. Frequency and Text Based analysis using LBP… 17

Fig 5: Variable focused based analysis… 18

Fig 6: Movement of the eye based analysis working… 19

Fig 7: theDifference between images using Optical Flow based analysis… 21

Fig 8: Graphic structure of CRF-based blinking model… 21

Fig 9: Example of non real faces… 23

Fig 10: Scenic Clues Based analysis working… 25

Fig 11: Lip Movement based analysis algorithm… 26

Fig 12: Face maps creation flow chart… 30

Fig 13: Diagram showing the outputs of PCA and LDA algorithms… 32

Fig 14: Decomposing the training face images… 34

Fig 15: A example of Same Face in Shimon Face Database… 34

Fig 16: liveness detection Database… 34

Fig 17: deep learning architecture… 42

Fig 18: The process of training livenessnet… 44

Fig 19: Real Video… 46

Fig 20:Spoofed VIdeo… 46

Fig 21: Dataset of real images 47

Fig 22: Dataset of Spoofed image… 47

Fig 23: output of video file… 48

Fig 24: Graph of images of dimension(16\*16)… 49

Fig 25:model summary of images of dimension(16\*16). 49

Fig 26: Graph of images of dimension(32\*32)… 50

Fig 27:model summary of images of dimension(32\*32). 50

Fig 28: Graph of images of dimension(64\*64)… 51

Fig 29:model summary of images of dimension(64\*64). 51

# Chapter 1: Introduction

The general public has immense need for security measures against spoof attack. Biometrics is the fastest growing segment of such security industry. Some of the familiar techniques for identification are facial recognition, fingerprint recognition, handwriting verification, hand geometry, retinal and iris scanner. Among these techniques, the one which has developed rapidly in recent years is face recognition technology and it is more direct, user friendly and convenient compared to other methods. Therefore, it has been applied to various security systems.

But, in general, face recognition algorithms are not able to differentiate ‘live’ face from ‘not live’ face which is a major security issue. It is an easy way to spoof face recognition systems by facial pictures such as portrait photographs. In order to guard against such spoofing, a secure system needs liveness detection.

Biometrics is the technology of establishing the identity of an individual based on the physical or behavioural attributes of the person. The importance of biometrics in modern society has been strengthened by the need for large-scale identity management systems whose functionality depends on the accurate deduction of an individual’s identity on the framework of various applications. Some examples of these applications include sharing networked computer resources, granting access to nuclear facilities, performing remote financial transactions or boarding a commercial flight [15].

The main task of a security system is the verification of an individual’s identity. The primary reason for this is to prevent impostors from accessing protected resources. General techniques for security purposes are passwords or ID cards mechanisms, but these techniques of identity can easily be lost, hampered or may be stolen thereby undermine the intended security. With the help of physical and biological properties of human beings, a biometric system can offer more security for a security system.

Face recognition[4] has been a hot research area for its wide range of applications. In human identification scenarios, facial metrics are more naturally accessible than many other biometrics, such as iris, fingerprint, and palm print. Face recognition is also

highly valuable in human computer interaction, access control, video surveillance, and many other applications.

Liveness detection has been a very active research topic in fingerprint recognition and iris recognition communities in recent years. But in face recognition, approaches are very much limited to deal with this problem. Liveness is the act of differentiating the feature space into live and non-living. Imposters will try to introduce a large number of spoofed biometrics into system. With the help of liveness detection, the performance of a biometric system will improve. It is an important and challenging issue which determines the trustworthiness of biometric system security against spoofing.

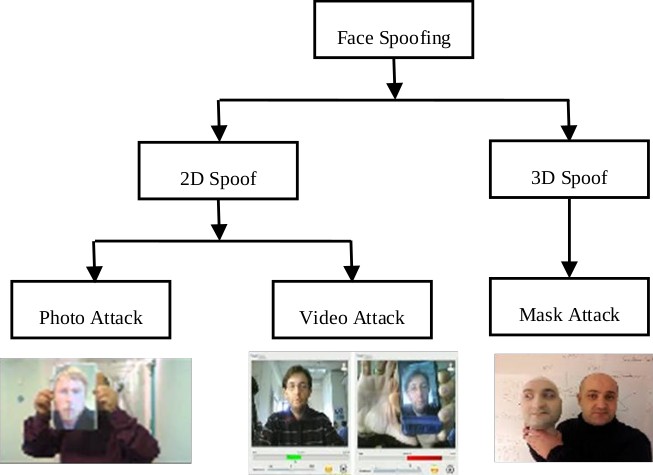


Fig 1: Image spoofing attacks

In face recognition, the usual attack methods may be classified into several categories. The classification is based on what verification proof is provided to the face verification system, such as a stolen photo, stolen face photos, recorded video, 3D face models with the abilities of blinking and lip moving, 3D face models with various expressions and so on. Anti-spoof problems should be well solved before face recognition systems could be widely applied in our daily life.

The objective[3] of developing biometric applications, such as facial recognition, has recently become important in smart cities. In addition, many scientists and engineers around the world have focused on establishing increasingly robust and accurate algorithms and methods for these types of system and their application in everyday life.

The objective of the project[3] is to provide user an algorithm which will detects the face of the user and unlock the applications or website’s according to that. It will first take the training data as an input from the camera of the device and then will train the model from the input and will detect the face according to that training of the model.

To work on the project, we will use Facial Recognition Technique in the project

In order to understand how Face Recognition works, let us first get an idea of the concept of a feature vector.

Every Machine Learning algorithm takes a dataset as input and learns from this data. The algorithm goes through the data and identifies patterns in the data. For instance, suppose we wish to identify whose face is present in a given image, there are multiple things we can look at as a pattern:

* Height/Width of face.
* Height and width may not be reliable since the image could be rescaled to a smaller face. However, even after rescaling, what remains unchanged are the ratios – the ratio of height of the face to the width of the face won’t change.
* Color of the face.
* Width of other parts of the face like lips, nose, etc.

Clearly, there is a pattern here – different faces have different dimensions like the ones above. Similar faces have similar dimensions. The challenging part is to convert a particular face into numbers – Machine Learning algorithms only understand numbers. This numerical representation of a “face” (or an element in the training set) is termed as a feature vector. A feature vector comprises various numbers in a specific order.

As a simple example, we can map a “face” into a feature vector which can comprise various features like:

* Height of face (cm)
* Width of face (cm)
* Average color of face (R, G, B)
* Width of lips (cm)
* Height of nose (cm)

So, our image is now a vector that could be represented as (23.1, 15.8, 255, 224, 189, 5.2, 4.4). Of course, there could be countless other features that could be derived from the image (for instance, hair color, facial hair, spectacles, etc). However, for example, let us consider just these 5 simple features.

Now, once we have encoded each image into a feature vector, the problem becomes much simpler. Clearly, when we have 2 faces (images) that represent the same person, the feature vectors derived will be quite similar. Put it the other way, the “distance” between the 2 feature vectors will be quite small.

Machine Learning can help us here with 2 things:

1. Deriving the feature vector: it is difficult to manually list down all of the features because there are just so many. A Machine Learning algorithm can intelligently label out many of such features. For instance, a complex feature could be: ratio of height of nose and width of forehead.
2. Matching algorithms: Once the feature vectors have been obtained, a Machine Learning algorithm needs to match a new image with the set of feature vectors present in the corpus.

We will use the OPENCV library in this project to detect the face of the person and to scan it. OpenCV is a collection of software algorithms put together in a library to be used by industry and academia for computer vision applications and research.

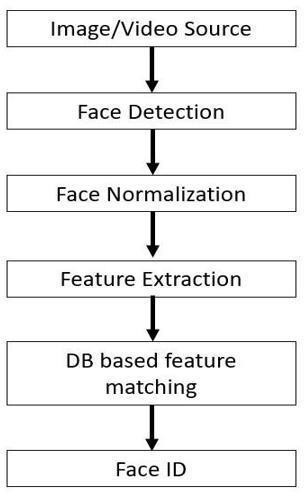


Fig 2. Crucial elements of the typical face recognition system.

## Objective

The main objective of this project is to improve our previous face recognition system by embedding a liveness detection algorithm to protect our project from image spoofing techniques.

## Problems Discussed

The main problem or the main weakness of our project was that it was not able to find out the difference between a photograph of a user and the live face of the user. So anybody can spoof our project by showing it a photograph of the user's face and unlock the device and therefore now we have worked on our project to remove this drawback and improve the accuracy.

# Chapter 2: Literature Survey

## Liveness Detection:

The general public has an immense need for security measures against spoof attacks. Biometrics is the fastest growing segment of such a security industry. Some of the familiar techniques for identification are facial recognition, fingerprint recognition, handwriting verification, hand geometry, retinal and iris scanner.



Fig 3: face detection methodology

Among these techniques, the one which has developed rapidly in recent years is face recognition technology and it is more direct, user friendly and convenient compared to other methods. Therefore, it has been applied to various security systems. But, in general, face recognition algorithms are not able to differentiate ‘live’ face from ‘not live’ face which is a major security issue. It is an easy way to spoof face recognition systems by facial pictures such as portrait photographs. In order to guard against such spoofing, a secure system needs liveness detection.

Liveness detection in biometrics is the ability of a system to detect if a fingerprint or face (or other biometrics) is real (from a live person present at the point of capture) or fake (from a spoof artifact or lifeless body part).

It comprises a set of technical features to counter biometric spoofing attacks where a replica imitating a person’s unique biometrics (like a fingerprint mold or 3D mask made of silicone) is presented to the biometric scanner to deceive or bypass the identification and authentication steps given by the system.

Liveness check uses algorithms that analyze data - after they are collected from biometric scanners and readers - to verify if the source is coming from a fake representation.

The need for unambiguous and secure identification and authentication has motivated a massive deployment of biometric systems worldwide.

[Increased public acceptance](https://www.thalesgroup.com/markets/digital-identity-and-security/government/inspired/where-facial-recognition-used), massive accuracy gains, a wide offer, and falling prices of sensors, IP cameras, and software have [accelerated these trends](https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/inspired/biometrics).

Here are some examples

* + - Today, over 1.2 Billion electronic passports are in circulation. They included a standardized ICAO-compliant holder’s picture and fingerprints in many countries.
    - The Indian biometric identification scheme consolidates the biometric and demographic data of over 1.26 Billion residents.
    - Many ID schemes integrate an electronic chip with a picture and fingerprints in addition to the biographical (name, date, and place of birth) data.

It’s a windfall for access, travel (self-service kiosks and automatic gates) but also civil [identification, eKYC](https://www.thalesgroup.com/en/markets/digital-identity-and-security/banking-payment/issuance/id-verification/know-your-customer) procedures, on-line customer registration and authentication, and more. Needless to say, biometric systems are also crucial for critical infrastructures such as border control, immigration and law enforcement, health and subsidies, population, and voter registration. Failure to prevent fingerprint spoofing attacks may have serious consequences.

This makes sense when you consider these other examples.

* + - [IDENT](https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/customer-cases/ident-automated-biometric-identification-system), the Automated Biometric Identification System, is a cornerstone of the United States’ border management and immigration. The central Department of Homeland Security system stores and processes over 200m identities, including biometric (ten fingers and a portrait) and associated biographic information.
    - The FBI automated fingerprint recognition system, named initially IAFIS (now NGI), is the world’s largest criminal history collection (more than 154m individuals) at the end of October 2020.
    - [The Eurodac](https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/customer-cases/eurodac) biometric system (European Dactyloscopy System) serves 32 countries in Europe.

The importance of spoof detection has been highlighted as early as 2013 by the European Commission’s TABULA RASA (Trusted Biometrics Under Spoofing Attacks) project.

Perhaps unsurprisingly, it has also been a topic of well-coordinated research in the United States since the launch of “Odin” in October 2017.

[The Odin program](https://www.iarpa.gov/index.php/research-programs/odin) has been initiated by the Intelligence Advanced Research Projects Activity (IARPA), an organization of the US Office of the Director of National Intelligence. Its goal is “to develop biometric presentation attack detection technologies to ensure biometric security systems can detect when someone attempts to disguise their biometric identity.”

Liveness detection is any technique used to detect a spoof attempt by determining whether the source of a biometric sample is a live human being or a fake representation. This is accomplished through algorithms that analyze data collected from biometric sensors to determine whether the source is live or reproduced.

There are two main categories of liveness detection:

* **Active:** Prompts the user to perform an action that cannot be easily replicated with a spoof. It might also incorporate multiple modalities, such as keystroke analysis or speaker recognition. The latter may analyze movement of a mouth to determine liveness.
* **Passive:** Uses algorithms to detect indicators of a non-live image without user interaction. Capture of high-quality biometric data during enrollment improves the performance of matching and liveness detection algorithms.

One or the other may be preferable in certain scenarios, but they generally work better together. Facial recognition is an ideal biometric modality for mobile authentication. It is intuitive and adaptable to most mobile devices, with widespread camera integration in commercial devices. It works with a familiar “selfie” pose. However, the widespread availability of digital facial images via social media makes facial biometrics more susceptible to spoofing. For this reason, it is critical to apply robust liveness detection for mobile biometric authentication solutions that use facial recognition.

A biometric liveness check refers to the verification that the features being presented to the biometric application are those of a living subject, and not a copy or imitation of those features.

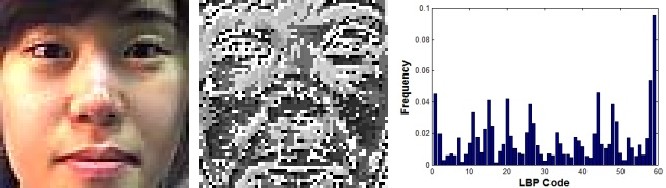
A liveness check allows a biometric application to discriminate in between the real biometric factor of a subject and artificial copies of those features making up the biometric factor. Liveness detection reduces the likelihood of spoofing attempts to succeed, and as such reduces the false acceptance rate. An example of a liveness check is a face recognition software requiring the subject to blink with his eyes, as such the software avoids spoofing through the use of photographs.

In facial recognition, liveness detection role is used distinguish between a live image and a 2D printed, 3D printed, or digital representation of a user’s face. Other spoof attempts may involve the use of a 3D mask. Spoof attempts can be detected through algorithms that recognize artifacts of a non-live sample, and may use “active” measures, such as a second modality (e.g. keystroke analysis or voice). Liveness detection methods significantly reduce the effectiveness of spoofing and other presentation attacks.

## Different Approaches for liveness detection:

### Frequency and Text Based Analysis:

This approach is used by Gahyun Kim et al [1]. The basic purpose is to differentiate between live face and fake face (2-D paper masks) in terms of shape and detailedness. The authors have proposed a single image-based fake face detection method based on frequency and texture analyses for differentiating live faces from 2-D paper masks. The authors have carried out a power spectrum based method for the frequency analysis, which exploits both the low frequency information and the information residing in the high frequency regions. Moreover, the description method based on Local Binary Pattern (LBP) has been implemented for analyzing the textures on the given facial images.





They tried to exploit frequency and texture information in differentiating the live face image from 2-D paper masks. The authors suggested that the frequency information is used because of two reasons. First one is that the difference in the existence of 3-D shapes, which leads to the difference in the low frequency regions which is related to the illumination component generated by overall shape of a face. Secondly, the difference in the detail information between the live faces and the masks triggers the discrepancy in the high frequency information. The texture information is taken as the images taken from the 2-D objects (especially, the illumination components) tend to suffer from the loss of texture information compared to the images taken from the 3-D objects. For feature extraction, frequency-based feature extraction, Texture-based feature extraction and Fusion-based feature extraction are being implemented.

### Variable Focused Based analysis:

The technique of face liveness detection using variable focusing was implemented by Sooyeon Kim et al. [3]. The key approach is to utilize the variation of pixel values by focusing between two images sequentially taken in different focuses which is one of the camera functions. Assuming that there is no big difference in movement, the authors have tried to find the difference in focus values between real and fake faces when two sequential images(in/out focus) are collected from each subject. In case of real faces, focused regions are clear and others are blurred due to depth information.

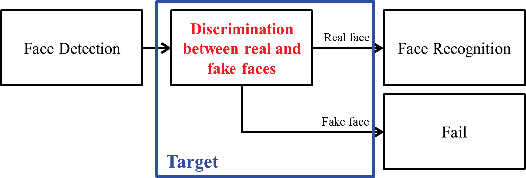


Fig 5: Variable focused based analysis

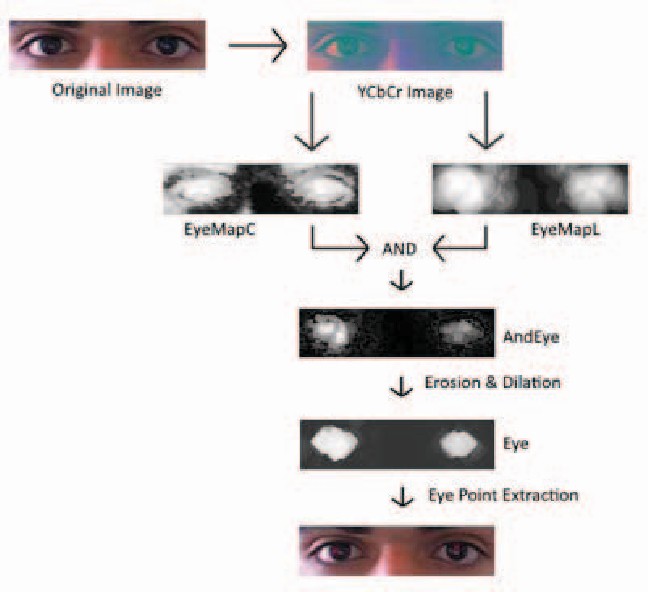
In contrast, there is little difference between images taken in different focuses from a printed copy of a face, because they are not solid. The basic constraint of this method is that it relies on the degree of Depth of Field (DoF) that determines the range of focus variations at pixels from the sequentially taken images. The DoF is the range between the nearest and farthest objects in a given focus. To increase the liveness detection performance, the authors have increased out focussing effect for which the DoF should be narrow. In this method, Sum Modified Laplacian(SML) is used for focus value measurement. The SML represents degrees of focusing in images and those values are represented as a transformed 2nd-order differential filter.

In the first step, two sequential pictures by focusing the camera on facial components are being. One is focused on a nose and the other is on ears. The nose is the closest to the camera lens, while the ears are the farthest. The depth gap between them is sufficient to express a 3D effect. In order to judge the degree of focusing, SMLs of both the pictures are being calculated. The third step is to get the difference of SMLs. For one-dimensional analysis, sum differences of SMLs (DoS) in each of columns are calculated. The authors found out that the sums of DoS of real faces show similar patterns consistently, whereas those of fake faces do not.

The differences in the patterns between real and fake faces are used as features to detect face liveness. For testing, the authors have considered False Acceptance Rate (FAR) and False Rejection Rate (FRR). FAR is a rate of the numbers of fake images misclassified as real and FRR is a rate of the numbers of real images misclassified as fake. The experimental results showed that when Depth of Field (DoF) is very small, FAR is 2.86% and FRR is 0.00% but when DoF is large, the average FAR and FRR is increased. Thus the results showed that this method is crucially dependent on DoF and for better results, it is very important to make DoF small.

### Movement of the eye based analysis:

The technique based on the analysis of movement of eyes was introduced by Hyung-Keun Jee et al. for embedded face recognition system [4]. The authors proposed a method for detecting eyes in sequential input images and then variation of each eye region is calculated and whether the input face is real or not is determined. The basic assumption is that because of blinking and uncontrolled movements of the pupils in human eyes, there should be big shape variations.



First center point of both eyes is detected in the input face image. Using both eyes, face regions are normalized and eye regions are extracted. After binarizing extracted eye regions, each binarized eye region is compared and variation is calculated. If the result is bigger than the threshold, the input image is recognized as a live face, if not, it is discriminated against in the photograph. For detection of the eye regions, the authors used the fact that the intensity of the eye region is lower than the rest of the face region if the image is considered as a 3D curve.

To find the eye region, first, Gaussian filtering to the face image is done, so that the smoothened 3D curve is obtained. In the curve, we extract all the local minimums using the method of the gradient descent. To reduce the invalid eye candidates, the eye classifier, which is trained by Viola's AdaBoost training methods, is used. After that, the face region is being normalized by about a size and rotation by using the center point of eyes because the input face can vary in size and orientation. To decrease the effect of illumination, Self Quotient Image (SQI) is applied. After Normalizing face region, eye regions are extracted based on the center of the eyes.

Then eye regions are binarized in order to have the pixel value of 0 and 1 by using a threshold. The threshold is obtained from the mean pixel value of each eye region. Eye regions from real faces have bigger variations in shape than regions obtained from fake faces. For calculating the liveness score of each eye region, the Hamming distance method is used. If two ordered lists of pixels are compared, the Hamming distance is the number of pixels that do not have the same value. If the average lifeness score is bigger than the threshold, the input image is recognized as a live face and in the case of the opposite it is discriminated against as a photograph.

### Optical Flow Based analysis:

The method based on the optical flow field was introduced by Bao et al. [5]. It analyzes the differences and properties of optical flow generated from 3D objects and 2D planes. The motion of the optical flow field is a combination of four basic movement types: Translation, rotation, moving and swing.

The authors found that the first three basic types are generating quite similar optical flow fields for both 2D and for 3D images. The fourth type creates the actual differences in the optical flow field. Their approach is basically based on the idea that the optical flow field for 2D objects can be represented as a projection transformation. The optical flow allows to deduce the reference field, thus allows to determine whether the test region is planar or not. For that, the difference among optical flow fields is calculated. To decide whether a face is a real face or not, this difference is being noted as a threshold.

The Experiment was conducted on three groups of sample data. The first group contained 100 printed face pictures that were translated and randomly rotated, the second group contains 100 pictures from group 1 that were folded and curled before the test, the third group consisted of faces of real people (10 people, each 10 times) doing gestures like swinging, shaking, etc. The authors conducted the experiment for 10 seconds. The camera had a sampling rate of 30 frames per second.

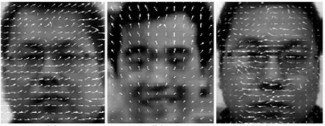


Fig 7: Showing the difference between images using Optical Flow based analysis

### Blinking Based analysis:

The blinking-based approach for liveness detection using Conditional Random Fields (CRFs) was introduced by Lin Sun et al.[7]. The authors have used CRFs to model blinking activities, for accommodating long-range dependencies on the observation sequence. Then they compared the CRF model with a discriminative model like AdaBoost and a generative model like HMM. Conditional random fields(CRFs) are probabilistic models for segmenting and labeling sequence data and mainly used in natural language processing for its accommodating long-range dependencies on the observation sequence.Blinking activity is an action represented by the image sequence which consists of images with close and non-close state. Graphic structure of CRF based blinking model. Here we show the model based on the context of observation of size 3. Label C and NC are closed and non close state analysis.

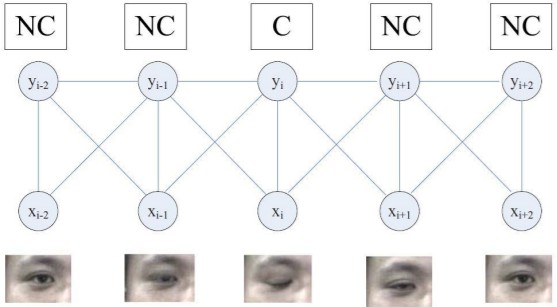


Fig 8: Blinking based analysis

The authors have applied a linear chain structure of CRFs. It has discrete eye state label data yt ∈ χ

= {1, 2, . . . , c}, t = 1. . .T, and observation xt. According to the authors, Half-open state is difficult to define commonly over the different individuals, since the eye size of half-open state depends on the person's eye appearance, for example, the open state of a small eye might look like the half-open state of a big eye. As for the blinking model, two state labels, C for close state and NC for non-close (include half-open and open), to label eye states that are being employed by the authors.

To test their approach, the authors have used video database including blinking video clips and imposter video clips, They used a total of 80 clips which is in blinking video database for 20 individuals, 4 clips for each individual: the first clip includes video without glasses in frontal view, the second clip is with thin rim glasses in frontal view, the third clip contains video with black frame glasses in frontal view, and the last clip is having video without glasses in upward view.

The video clips which are being used are of five seconds’ length with 30 fps and size of 320 × 240. The blinking number in each clip varies from 1 to 6 times. To test the ability against photo imposters, the authors have used 180 photo imposter video clips of 20 persons with various motions of photo, including rotating, folding and moving.

### Component Dependent Descriptor based analysis:

The technique of Component-based face coding approach for liveness detection was employed by Jianwei Yang et al. [9]. The authors have proposed a method which consists of four steps: (1) locating the components of face; (2) coding the low-level features respectively for all the components; (3) deriving the high-level face representation by pooling the codes with weights derived from Fisher criterion; (4) concatenating the histograms from all components into a classifier for identification.

The authors found out that significant operational differences between genuine faces and fake ones is that the former are captured by camera once, whereas the latter are obtained by re-capturing images of photos or screens. This will produce their appearance differences in three aspects: (1) Faces are blurred because of limited resolution of photos or screens and re-defocus of camera; (2) Faces appearance vary more or less for reflectance change caused by Gamma Correction of camera; (3) Face appearance also change for abnormal shading on surfaces of photos and screens.

At first, the authors have expanded the detected face to obtain the holistic-face (H-Face). Then the H-Face is divided into six components (parts) which includes contour region, facial region, left eye region, right eye region, mouth region and nose region.

Moreover, the contour region and facial region are further divided into 2 × 2 grids, respectively. For all the twelve components, dense low level features (e.g., LBP, LPQ, HOG, etc.) are extracted. Given the densely extracted local features, a component-based coding is performed based on an offline trained codebook to obtain local codes.

Then the codes are concatenated into a high-level descriptor with weights derived from Fisher criterion analysis. Fisher ratio is used to describe the difference of micro textures between genuine faces and fake faces. At last, the authors feed features into a support vector machine (SVM) classifier.

### 3D Face Shape based analysis:

The novel liveness detection method, based on the 3D structure of the face is proposed by Andrea Lagorio et al. [10]. The proposed approach allows a biometric system to differentiate a real face from a photo thus reducing the vulnerability. The authors suggested that the proposed approach can be implemented in different scenarios: either as an anti-spoofing tool, coupled with 2D face recognition systems; or can be integrated with a 3D face recognition system to perform an early detection of spoofing attacks.



Fig 9: From left to right columns, faces are made up of silica gel, rubber, photographs, and video relays respectively

The proposed algorithm computes the 3D features of the captured face data to determine if there is a live face presented in front of the camera or not. The authors show that the lack of surface variation in the scan is one of the key pieces of evidence that the acquisition comes from a 2D source. It has a very low surface curvature. Based on the computation of the mean curvature of the surface, a simple and fast method is implemented to compare the two 3D scans. An approximation of the actual curvature value at each point is computed from the principal components of the Cartesian coordinates within a given neighborhood.

The mean curvature of the 3D points lying on the face surface is then computed. The authors designed two experiments. In the first one, they used FS and GVS sets. The distribution of the mean curvature values for the two sets was separated, and the value of the False Rejection Rate (FRR), was computed as zero. In the second experiment they used the FS and the Bosforus sets. In order to determine the sensitivity of the algorithm, they perform various experiments with values ranging from 4 to 20. For different values of radius, the value of the False Rejection Rate (FRR) at rank 1 is always equal to zero.

### Binary Classification based analysis:

The technique of anti-spoof problems as a binary classification problem was introduced by Tan et al.[11]. The key approach which the authors have used is that a real human face is different from a face in a photo. A real face is a 3D object while a photo is 2D by itself. The surface roughness of a photo and a real face is different. The authors presented a real-time and non-intrusive method to address this based on individual images from a generic web camera.

The task is being formulated as a binary classification problem, in which, however, the distribution of positive and negative are largely overlapping in the input space, and a suitable representation space is found to be of great importance. Using the Lambertian model, they proposed two strategies to extract the essential information about different surface properties of a live human face or a photograph, in terms of latent samples. Based on these, two new extensions to the sparse logistic regression model were employed which allow quick and accurate spoof detection.

### Scenic Clues based analysis:

The technique of face liveness detection by exploring multiple scenic clues was introduced by Yan et al. [14]. The authors have proposed a method which includes three scenic clues: non-rigid motion, face background consistency and imaging banding effect for accurate and efficient face liveness detection. Non-rigid motion clue indicates the facial motions such as blinking, and a low rank matrix decomposition based image alignment approach is implemented to extract this nonrigid motion.

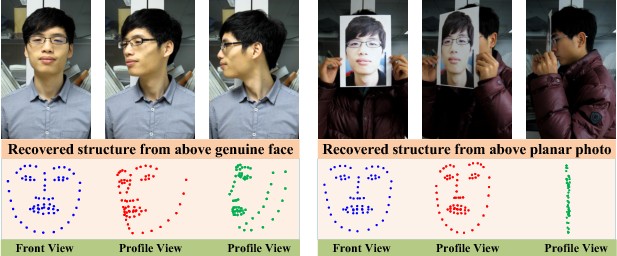


Fig 10: Scenic Clues Based analysis working

Face-background consistency clue assumes that the motion of face and background has high consistency for fake face images while low consistency for genuine faces, and this consistency can serve as an efficient liveness clue. The authors have implemented GMM based motion detection methods for face-background consistency. Image banding effect reflects the imaging quality defects introduced in the fake face reproduction, for which the authors have used wavelet decomposition for detection. The authors have fused these three clues for efficient liveness detection.

### Lip Movement based analysis:

The liveness detection approach using lip movement classification and face detection based on face landmarks was introduced by Kollreider et al. [16]. Their work is based on liveness detection using face landmarks. The proposed approach analyzes lip movements and lip reading for liveness detection. For classification of lip dynamics, SVM was used by the authors. They proposed an approach to locate the mouth regions and extract OFL in real time. They have used the XM2VTS database on various scenarios. Persons were recorded speaking digits from 0 to 9.

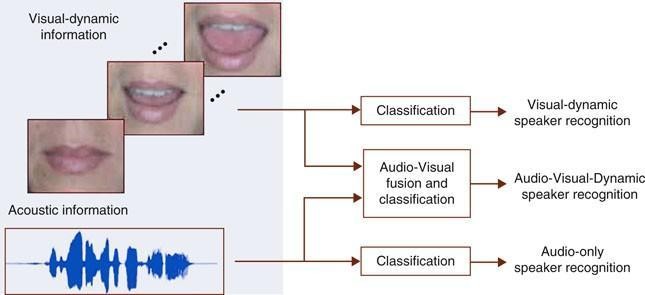


Fig 11: Lip Movement based analysis algorithm

The goal was to recognize the digits by lip-motion only. For a digit, they have used 100 short videos. For training (SVM classifier, cross validation), there were a total of 60 videos and for testing, a total of 40 videos were there. For each of digit videos, features vectors are extracted from mouth regions and given to a 10-class SVM. As a result, confusion matrices are obtained. Out of 100 individuals, recognition rate is 0.73 (73%). The authors proposed the method as an indication of liveness.

### Context Based analysis:

This novel technique of context based face anti-spoofing was introduced by Komulainen et al. [18]. The authors have followed the principle of attack-specific spoofing detection and engage in face spoofing scenarios in which scene information can be exploited. They are trying to detect whether someone is trying to spoof by presenting a fake face in front of the camera in the provided view. The basic idea was that the humans rely mainly on scene and context information during the detection of spoofing; the proposed algorithm tries to impersonate human behaviour and exploits scenic cues for determining whether a fake face is presented in front of the camera or not.

The proposed approach consists of a cascade of an upper-body (UB) and a spoofing medium (SM) detector which are based on histogram of oriented gradients (HOG) descriptors and linear support vector machines (SVM). The authors suggested that the method can operate either on a single video frame or video sequences.

### Combination of Standard Techniques based analysis:

The technique that combines standard techniques in 2D face biometrics was introduced by Kollreider et al. [19]. They have looked into the matter using real-time techniques and applied them to real life spoofing scenarios in an indoor environment. First of all, the algorithm searches for faces and if the face is detected, a timer is started to define the period for collecting evidence. Then evidence is collected for the liveness detection of the faces. For liveness detection, 3D properties or eye-blinking or mouth movements in non-interactive mode are being analyzed.

If no such response is found, responses are asked and checked at random. After the time period expires, verify the liveness of the face. For experimentation, a low cost web-cam that delivered 320x240 pixel frames at 25 fps was employed and computation was done on a standard laptop. The authors suggested that the performance of the proposed method is efficient for the task of public usage.

We studied around more than 30 research papers and concluded three approaches which we can use for the face recognition concept. These three approaches are:

1. Local Approach
2. Hybrid Approach
3. Holistic Approach

## Local Approach:

In the context of facial recognition, local approaches treat only certain facial features. They are more sensitive to facial expressions, remorse, and posture [1]. The main objective of these approaches is to search for specific features. Generally, we can divide these approaches into two categories:

* + 1. local appearance-based techniques: It is used to extract local features, while the whole face image is divided into small regions known as patches.
    2. Key-points-based techniques: It is used to detect the point of interest in the face image, after that the features present on this points are localized.

### Local Binary Patterns (LBP):

Local Binary Patterns is a very easy but an efficient texture operator which marks or labels the pixels of an image by thresholding the sides of each pixel and treating the result as a binary number[9].

### Artificial Neural Network:

The main aim of artificial neural network is basically to solve nonlinear problems. A radial basis function is integrated with a non-factor negative matrix factor to identify the neural network face. An artificial neural network approach has been proposed which has elements of both statistical and neural networks and refills the methods for detecting face images with partial distortion and occlusion.[10]

But the major drawback of this method is, like other statistical-based methods, is very much less accurate to model classes given only a small or single number of training patterns.

### Gabor wavelet‐based solutions:

Since Gabor features are considered as one the best method for face recognition because of its high accuracy therefore researchers all around the globe highly used Gabor wavelets for face recognition and face detection. Since, It is demonstrated to be discriminatory and robust to illumination and expression variations.[8] There are two types of strategies to capture information in Gabor texture:

1. Gabor phase‐based texture representation (GPTR)
2. Gabor magnitude‐based texture representation (GMTR)

One of the major limitation of the dimensionality of the Gabor feature space is quite high because images of the face are decorated with a bank of Gabor filters.[2]

Also, extraction of the Global features is computationally extensive therefore the use of features for real-time applications is currently baseless.

### Face descriptor-based method:

Face image description based on local features provides a global description. At first the local features of the face are extracted and then the extracted features are analysed and evaluated in the neighbouring pixels and united to form the final global description. [10]

One of the biggest advantages of this method is that it is compact, highly discriminative, and extraction of learning-based descriptor is easy. These methods are discriminative and robust to illumination and expression changes.

### 3D Based Face recognition:

Use of 3D capturing features is comparatively cheaper and faster than the other process and may researchers thought that the use of 3D face recognition process will give a lot better results than the use of 2D face recognition process because of its high accuracy.[3] One of the main advantages of using the 3D process is that the data sets do not depend upon the illumination and pose, and therefore, the representation of the object does not change with the change of any of these parameters which makes the whole system more robust. [5]

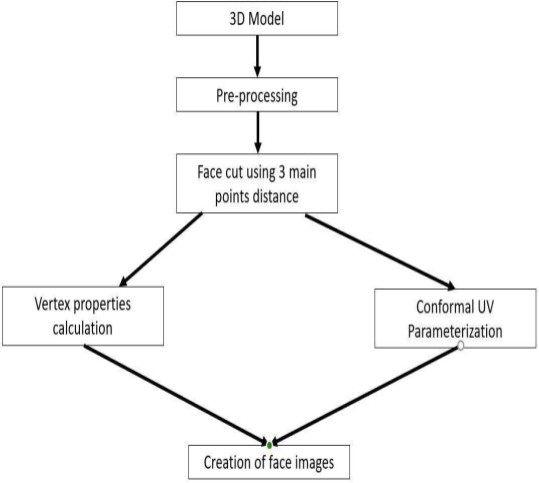


Fig 12. Face maps creation flow chart

### Video-based face recognition:

Analysis of video streams of facial images has received increasing attention in biometrics. An immediate advantage in using video information is the possibility of employing the redundancy present in the video sequence that improves still image systems.[4]

On the basis of how they leverage the multitude of information available in a video sequence, we can divide the VFR approaches into two broad categories:

1. Set based
2. Sequence based

The first stage of the video is face based face recognition (VFR) again to identify where a collection of videos is crossed to match all events of the person of interest.

## Holistic approach:

Holistic means the whole of something should be considered instead of just considering its parts. The word ‘holistic’ is derived from philosophy holism. In the holistic approach, the whole face is considered as a unit feature for recognition and detection [11]. It captures global information or data from faces to distinguish. Global data is basically a feature generated from the pixel data of the facial image. It compares the similarity between faces instead of features of face such as nose, eyes, ears, and face-cut etc. Global data helps us to distinguish the images and helps us to uniquely identify an individual or subject. The biggest disadvantage of this approach is that we must deal with small size images. Holistic approach is divided into two parts:

* + 1. Statistical methods
    2. Artificial intelligence methods

### Statistical methods

In this approach we calculate the face image density and then compared it with density values of images from database [12]. This calculation is costly and suffers from voids by path, generated from illumination and face orientation. Reduction in image size is suggested to reduce the cost. Some of the methods are:

* PCA
* LDA

### PCA (Principal Component Analysis)

It converts a group of correlated variables into the group of linear uncorrelated variables by using orthogonal transformation. It is also used for lowering the dimension of data while retaining most of the information [13]. The PCA algorithm was founded by Karl Pearson [14]. This technique is initially proposed by Thurstone [15] which analyzes the factors for reducing data, hence also called Factor Analysis Method.

PCA is also known as the Eigenvector or Eigenfaces approach. Eigenfaces are Principle Components that contain feature vectors of parts of face in the form of covariance matrix [16]. The illumination normalization for correlation of faces is very necessary. This [17] shows that even in the presence of varying lighting and scaling conditions, the

Eigenface approach is very efficient, fast, and reliable. Eigen-face based facial recognition algorithm which used a high level functioning principle [18].

KPCA (Kernel PCA) is the advanced version of PCA using kernel-based methods [19]. PCA can also be combined with other techniques to make it even more powerful like combining PCA with hexagonal image processing in [20].

### LDA (Linear Discriminant Analysis)

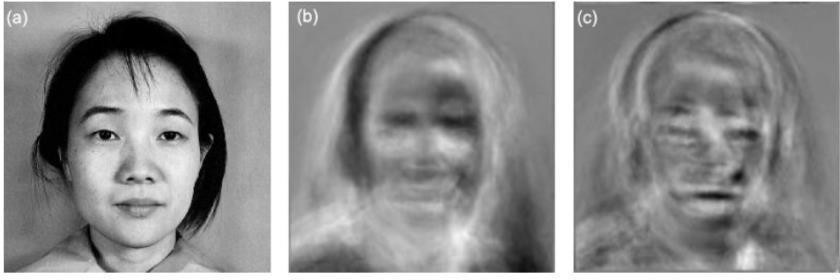
Dimensionality reduction and feature extraction are two main purposes of LDA [21]. It is also known as Fisher’s Linear Discriminate or Fisher-faces. Fisher’s original article was the earliest idea and did not contain the equal class covariance [22]. Zhao et al. presented the LDA [23]. Identity of individuals is determined by the data of individual which is found by increasing the grouping of individual faces and decreasing the grouping of different individual faces [24].

Fig 13. Diagram showing the outputs of PCA and LDA algorithms

### Artificial Intelligence Methods

Artificial means unnatural and intelligence means problem solving on own. AI is the field of computer science where we make computer machines able to solve problems and learn. We can also say that enabling machines to think and work like humans; and exhibit the characteristics of human Intelligence. Machine automatically recognizes the faces after the training of machines by various methods and. Some of the methods are:

1. Artificial Neural Network
2. Fuzzy Pattern Matching

### ANN (Artificial Neural Networks)

As the name suggests, ANNs are computer systems whose designs are inspired by the structure of biological neural networks or the brain, also synthesizing the information like the human brain. Structure of the information processing system is made by the collection of large numbers of connected nodes known as artificial neurons, and they work together to solve specific problems. ANNs are like humans, which learn by example, training and past experiences. ANNs are made for specific tasks such as data classification and pattern recognition.

Training a neural network for face detection is a tough task as non-facial images characterization is cumbersome. There are two sets of images, one with facial images and another with non-facial images. The number of subjects in the second set grows rapidly. Use of a big set of non-facial images is the solution to the above problem [25].

Neural network based face detection system [26]. It decides by dividing the image into small parts and whether each part has a face or not, by using multiple networks to increase its performance. This performs evaluation by calculating distance between a set of images and an input image by using three layers of weight [27].

### Fuzzy Pattern Matching

Fuzzy Matching is non-exact matching. It is also known as approximate matching. It helps us to find the nearby items related to target search. Many search engines framework work upon it. It helps us to find the result even if we have mistakes like typo or verbal errors. Two fuzzy models used to detect face in color images with characteristics like hair color and skin color [28]. With the help of fuzzy theory and builtin template, there is also a comparison between two models.

## Hybrid approach:

Hybrid approach is the combination of different algorithm based on local and holistic approach in order to get the benefits of the both algorithm it increase the chances of better results and efficiency for the face recognition system. Some of the method in hybrid approach.

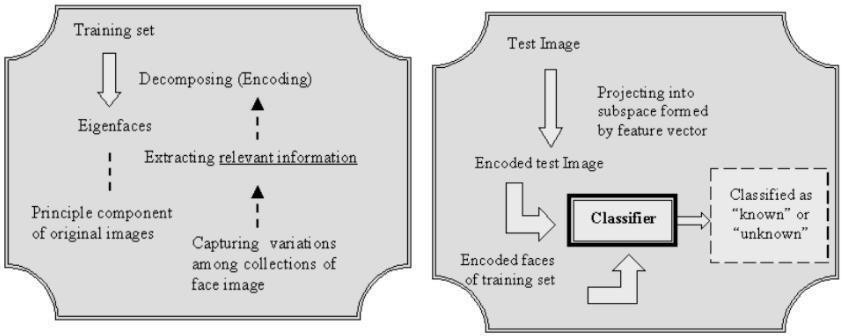


Fig 14. Decomposing the training face images; Classification of the input face image

Fig 15: A example of Same Face in Shimon Face Database

### Gabor Filters and Neural Network (GW-NN)

Er. Pooja Rani (R.I.E.T)2010 in this paper the author proposed to combine the Gabor filter and neural network. Author used a Gabor filter as a training algorithm and neural network use as a classifier. The accuracy of this method is 84.4% but it has some limitation as it identifies only upright frontal face and is restricted to rotation of the image and the side part of the image.

### Gabor Filters and Linear Discriminant Analysis (GW-LDA)

Samir F.Hafez1 proposed a 2D face recognition system in this method to combine the Gabor filters and linear discriminant analysis(LDA) . He only used an orthogonal Gabor filter instead of the whole Gabor filter. Then compress using LDA. It has an accuracy rate of 98.9%.They tested using CASSIA ORL and Cropped YaleB databases for the 2d image.

### DPCA and ANFIS classifier

Ketan Patel had proposed a hybrid approach by combining 2D-PCA (Principal Component Analysis) and Adaptive-neuro-fuzzy inference system. In this method PCA is used for feature extraction and then extracted feature vectors are applied by ANFIS classifier. It has the accuracy of 97.1%.

### DCT and PCA

Amal. E. Aswis 2015 journal he proposed a face recognition system based on

Discrete Cosine Transform(DCT) as a compressor and Principal Component Analysis as a classifier and they used a FACES94 and AT & T database consisting of different light and dimensions. This method attains a good result as it needs a low computation and low cost hardware implementation due to this the system gets a good training and better recognition time.

### ESPCA-KNN

Mrs. J. Savitha 2015 she proposed a method by combining Enhanced SPCA ((shift-pca) Scale Invariant Feature Extraction-Principal Element Analysis) as a compressor and KNN (K-Nearest Neighbour) as classifier from the experiment it is found that SPCA technique is comparatively superior than eigen faces or other relative technique. This journal results shows that positive results compare to different examine techniques.

### 2D-3D

Ajmal S. Mian 2007 He proposed a face recognition technique in which he combined a 2D-3D (Dimension) model. In which 3D-SFR (Spherical face representation) are used with SIFT (Scale-invariant feature extraction technique) to form a classifier. It has an accuracy of 99.74% and 98.3% verification rate at a 0.001 false acceptance rate.

### CNN-LSTM

Marco Bellentio 2016 he proposed a special face recognition technique in which he combined a convolution neural network and Long-short term memory in which he used a novel database (YouTube face (YTF),CMU motion of body (mobo), UNBC-Mcmaster shoulder plan etc ) which is consist of images and videos. In this technique he used cnn as a classifier and then fed the resulted vector image into lstm for prediction and the accuracy of this method is around a 92%.

### Convolution operations, LSTM recurrent units and ELM classifier

Jian su 2018 in this journal he proposed a technique by combining convolution operations, LSTM recurrent units and ELM classifier in this paper he used combined cnn to lstm to solve the image or video precision and due to this he gets a good precision rate and the efficiency but these method failed on real life application to solve this he used extreme learning method (ELM) into deep convolution network to improve real life application.

This method gets the efficiency rate of 90%.

### Multi-Subregion based Correlation Filter Bank (MS-CFB)

Yan yan1 2016 in this journal he proposed a technique in which he combined a multisubregion and correlation filter bank (MS-CFB) in this paper he combined a global feature with local features and also minimize Rayleigh quotient this is also optimize the overall feature extraction output this method enhance the local feature extraction efficiency and optimization and the overall accuracy of this method is 75%.

## Techniques used:

### OpenCV:

OpenCV (Open Source Computer Vision Library) is [a library of programming](https://en.wikipedia.org/wiki/Library_(computing)) [functions](https://en.wikipedia.org/wiki/Library_(computing)) mainly aimed at real-time [computer vision](https://en.wikipedia.org/wiki/Computer_vision).[[1]](https://en.wikipedia.org/wiki/OpenCV#cite_note-1) Originally developed by [Intel](https://en.wikipedia.org/wiki/Intel_Corporation), it was later supported by [Willow Garage](https://en.wikipedia.org/wiki/Willow_Garage) then Itseez (which was later acquired by Intel[[2]](https://en.wikipedia.org/wiki/OpenCV#cite_note-2)). The library is [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) and free for use under the [open-source](https://en.wikipedia.org/wiki/Open-source_software) [Apache 2 License](https://en.wikipedia.org/wiki/Apache_License). Starting with 2011, OpenCV features GPU acceleration for real-time operations.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people in the user community and an estimated number of downloads [exceeding 18 million](https://sourceforge.net/projects/opencvlibrary/files/stats/timeline?dates=2001-09-20%2Bto%2B2019-01-30). The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV’s deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, [Linux, Android](https://opencv.org/opencv/android/) and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured [CUDA](https://opencv.org/opencv/cuda/)and [OpenCL](https://opencv.org/opencv/opencl/) interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

### Keras:

[Keras is an open-source](https://en.wikipedia.org/wiki/Open-source_software) [software](https://en.wikipedia.org/wiki/AI_software) library that provides a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) interface for [artificial neural](https://en.wikipedia.org/wiki/Artificial_neural_network) [networks](https://en.wikipedia.org/wiki/Artificial_neural_network). Keras acts as an interface for the [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow) library.

Up until version 2.3, Keras supported multiple backends, [including TensorFlow](https://en.wikipedia.org/wiki/TensorFlow), [Microsoft](https://en.wikipedia.org/wiki/Microsoft_Cognitive_Toolkit) [Cognitive Toolkit](https://en.wikipedia.org/wiki/Microsoft_Cognitive_Toolkit), [Theano](https://en.wikipedia.org/wiki/Theano_(software)), and [PlaidML](https://en.wikipedia.org/wiki/PlaidML).[[2]](https://en.wikipedia.org/wiki/Keras#cite_note-2)[[3]](https://en.wikipedia.org/wiki/Keras#cite_note-why-keras-3)[[4]](https://en.wikipedia.org/wiki/Keras#cite_note-4) As of version 2.4, only [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow) is supported. Designed to enable fast experimentation with [deep neural networks](https://en.wikipedia.org/wiki/Deep_learning), it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System),[[5]](https://en.wikipedia.org/wiki/Keras#cite_note-5) and its primary author and maintainer is François Chollet, a [Google](https://en.wikipedia.org/wiki/Google) engineer. Chollet is also the author of the XCeption deep neural network model

Keras contains numerous implementations of commonly used neural-network building blocks such as [layers, objectives](https://en.wikipedia.org/wiki/Objective_function), [activation functions](https://en.wikipedia.org/wiki/Activation_function), [optimizers](https://en.wikipedia.org/wiki/Mathematical_optimization), and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on [GitHub](https://en.wikipedia.org/wiki/GitHub), and community support forums include the GitHub issues page, and a [Slack](https://en.wikipedia.org/wiki/Slack_(software)) channel.

In addition to standard neural networks, Keras has support [for convolutional](https://en.wikipedia.org/wiki/Convolutional_neural_networks) and [recurrent neural](https://en.wikipedia.org/wiki/Recurrent_neural_networks) [networks](https://en.wikipedia.org/wiki/Recurrent_neural_networks). It supports other common utility layers like [dropout](https://en.wikipedia.org/wiki/Dropout_(neural_networks)), [batch normalization](https://en.wikipedia.org/wiki/Batch_normalization), and [pooling](https://en.wikipedia.org/wiki/Pooling_(neural_networks)).[[7]](https://en.wikipedia.org/wiki/Keras#cite_note-7)

Keras allows users to productize deep models on smartphones [(iOS](https://en.wikipedia.org/wiki/IOS) and [Android](https://en.wikipedia.org/wiki/Android_(operating_system))), on the web, or on the [Java Virtual Machine](https://en.wikipedia.org/wiki/Java_Virtual_Machine).[[3]](https://en.wikipedia.org/wiki/Keras#cite_note-why-keras-3) It also allows use of distributed training of deep-learning models on clusters of [Graphics processing units (GPU)](https://en.wikipedia.org/wiki/Graphics_processing_unit) and [tensor processing units (TPU)](https://en.wikipedia.org/wiki/Tensor_processing_unit).

### Deep learning:

Deep learning (also known as deep structured learning) is part of a broader family [of machine](https://en.wikipedia.org/wiki/Machine_learning) [learning](https://en.wikipedia.org/wiki/Machine_learning) methods based on [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_networks) with [representation learning](https://en.wikipedia.org/wiki/Representation_learning). Learning can be [supervised](https://en.wikipedia.org/wiki/Supervised_learning), [semi-supervised](https://en.wikipedia.org/wiki/Semi-supervised_learning) or [unsupervised](https://en.wikipedia.org/wiki/Unsupervised_learning).[[1]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-BENGIO2012-1)[[2]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-SCHIDHUB-2)[[3]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-NatureBengio-3)

Deep-learning architectures such [as deep neural networks](https://en.wikipedia.org/wiki/Deep_learning#Deep_neural_networks), [deep belief networks](https://en.wikipedia.org/wiki/Deep_belief_network), [graph neural](https://en.wikipedia.org/w/index.php?title=Graph_neural_network&action=edit&redlink=1) [networks](https://en.wikipedia.org/w/index.php?title=Graph_neural_network&action=edit&redlink=1), [recurrent neural networks](https://en.wikipedia.org/wiki/Recurrent_neural_networks) and [convolutional neural networks](https://en.wikipedia.org/wiki/Convolutional_neural_networks) have been applied to fields including [computer vision](https://en.wikipedia.org/wiki/Computer_vision), [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition), [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), [machine](https://en.wikipedia.org/wiki/Machine_translation) [translation](https://en.wikipedia.org/wiki/Machine_translation), [bioinformatics](https://en.wikipedia.org/wiki/Bioinformatics), [drug design](https://en.wikipedia.org/wiki/Drug_design), [medical image analysis](https://en.wikipedia.org/wiki/Medical_image_analysis), material inspection and [board](https://en.wikipedia.org/wiki/Board_game)

[game](https://en.wikipedia.org/wiki/Board_game) programs, where they have produced results comparable to and in some cases surpassing human expert performance.[[4]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-4)[[5]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-%3A9-5)[[6]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-krizhevsky2012-6)[[7]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-7)

[Artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network) (ANNs) were inspired by information processing and distributed communication nodes in [biological systems](https://en.wikipedia.org/wiki/Biological_system). ANNs have various differences from biological [brains](https://en.wikipedia.org/wiki/Brain). Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analogue.[[8]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-8)[[9]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-9)[[10]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-10)

The adjective "deep" in deep learning refers to the use of multiple layers in the network. Early work showed that a [linear perceptron](https://en.wikipedia.org/wiki/Perceptron) cannot be a universal classifier, but that a network with a nonpolynomial activation function with one hidden layer of unbounded width can. Deep learning is a modern variation which is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation, while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed [connectionist](https://en.wikipedia.org/wiki/Connectionism) models, for the sake of efficiency, trainability and understandability, hence the "structured" part.

# Chapter 3: Proposed Solution

We have tried to create a project in which we have used several technologies like OpenCV, Keras, Deep Learning and CNN to implement the liveness or anti-spoofing algorithm into our model and make our project capable enough to find out the difference between a 2D photograph of the user and live face of a user .

## Project structure

├── dataset

│ ├── fake

│ └── real

├── face\_detector

│ ├── deploy.prototxt

└── res10\_300x300\_ssd\_iter\_140000.caffemodel

│

├── pretrained model

│ ├── init .py

│ └── livenessnet.py

├── videos

│ ├── fake.mp4

│ └── real.mp4

├── gather\_images.py

├── train.py

├── liveness\_demo.py

├── le.pickle

├── liveness.model

└── plot.png

## Generate Dataset

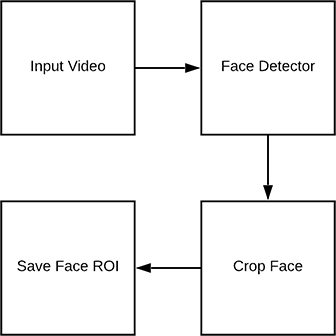


Fig 16.Detecting face ROIs in video for the purposes of building a liveness detection dataset.

**steps to create dataset**

gather information from the these two directories. 1.dataset/fake/: Contains face ROIs from the fake.mp4 file 2.dataset/real/: Holds face ROIs from the real.mov file.

### some key word that are used in our code for generating our image

* + - * --input : The path to our input video file.
      * --output : The path to the output directory where each of the cropped faces will be stored.
      * --detector : The path to the face detector. We’ll [be using OpenCV’s deep learning face detection](https://pyimagesearch.com/2018/02/26/face-detection-with-opencv-and-deep-learning/). The Caffe model.
      * --confidence : The minimum probability to filter weak face detections. By default, this value is 50%.
      * --skip : We don’t need to detect and store every image because adjacent frames will be similar. Instead, we’ll skip *N* frames between detections. You can alter the default of 16 using this argument.

## Implementing “LivenessNet” algorithm:

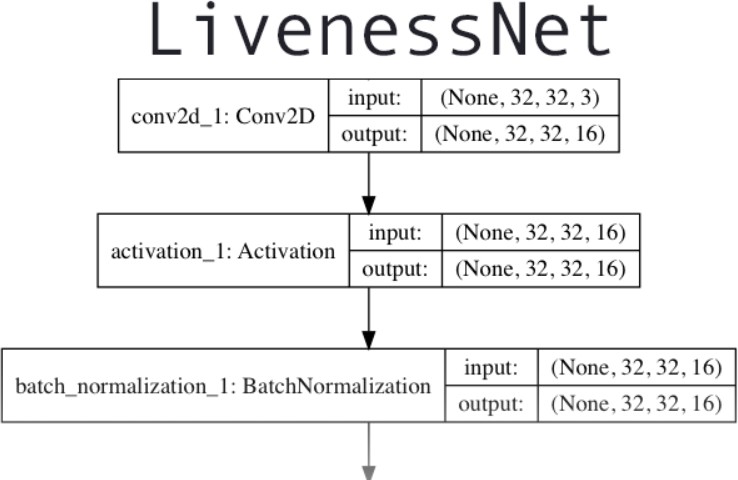


Fig: 17 Deep learning architecture for LivenessNet, a CNN designed to detect face liveness in images and videos.

The next step is to implement “LivenessNet”, our deep learning-based liveness detector.At the core, LivenessNet is actually just a simple Convolutional Neural Network.

We’ll be purposely keeping this network as shallow and with as few parameters as possible for two reasons:

1. To reduce the chances of overfitting on our small dataset.
2. To ensure our liveness detector is fast, capable of running in real-time (even on resource-constrained devices, such as the Raspberry Pi).

Let’s implement LivenessNet now — open up livenessnet.py and insert the following code:

All of our imports are from Keras. For an in-depth review of each of these layers and functions, be sure to refer to [*Deep Learning for Computer Vision with Python*](https://pyimagesearch.com/deep-learning-computer-vision-python-book/).

It consists of one static method, build. The build method accepts four parameters:

* + width : How wide the image/volume is.
  + height: How tall the image is.
  + depth : The number of channels for the image (in this case 3 since we’ll be working with RGB images).
  + classes: The number of classes. We have two total classes: “real” and “fake”.

Let’s begin adding layers to our CNN

Our CNN exhibits VGGNet-esque qualities. It is very shallow with only a few learned filters. Ideally, we won’t need a deep network to distinguish between real and spoofed faces.

The first CONV => RELU => CONV => RELU => POOL layer set is specified where batch normalization and dropout are also added.

Another CONV => RELU => CONV => RELU => POOL layer set is appended. Finally, we’ll add our FC => RELU layers:

### Creating the Liveness detector training script:

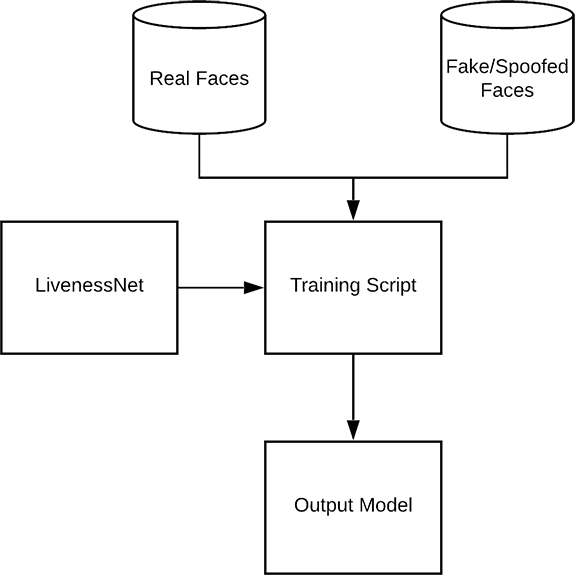


Fig 18: The process of training LivenessNet. Using both “real” and “spoofed/fake” images as our dataset, we can train a liveness detection model with OpenCV, Keras, and deep learning.

Given our dataset of real/spoofed images as well as our implementation of LivenessNet, we are now ready to train the network.

Our face liveness training script consists of a number of imports. Let’s review them now:

* + matplotlib : Used to generate a training plot. We specify the "Agg" backend so we can easily save our plot to disk.
  + LivenessNet : The liveness CNN that we defined in the previous section.
  + train\_test\_split : A function from scikit-learn which constructs splits of our data for training and testing.
  + classification\_report : Also from scikit-learn, this tool will generate a brief statistical report on our model’s performance.
  + ImageDataGenerator : Used for performing data augmentation, providing us with batches of randomly mutated images.
  + Adam : An optimizer that worked well for this model.
  + paths : From my imutils package, this module will help us to gather the paths to all of our image files on disk.
  + pyplot : Used to generate a nice training plot.
  + numpy : A numerical processing library for Python. It is an OpenCV requirement as well.
  + pickle : Used to serialize our label encoder to disk.
  + cv2 : Our OpenCV bindings.
  + os : This module can do quite a lot, but we’ll just be using it for it’s operating system path separator.

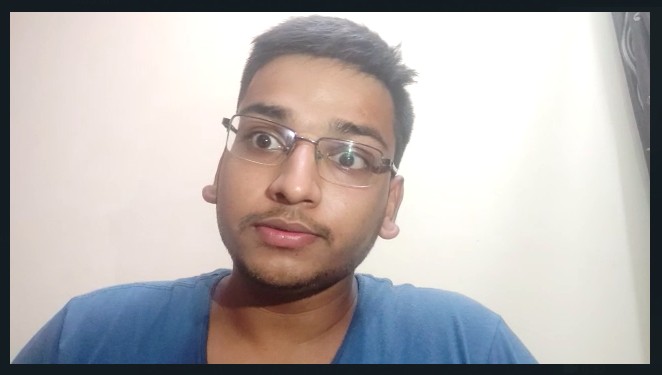


Fig 19: Real Video



Fig 20: Spoofed video

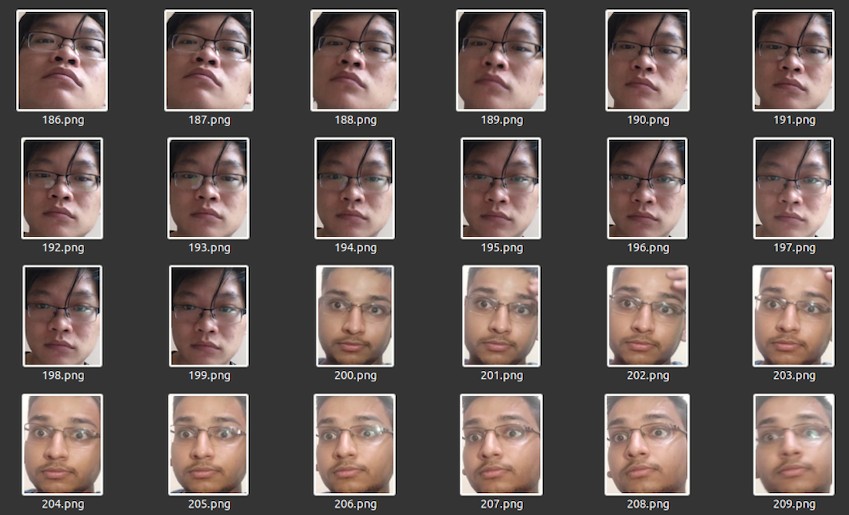


Fig 21: dataset of real images

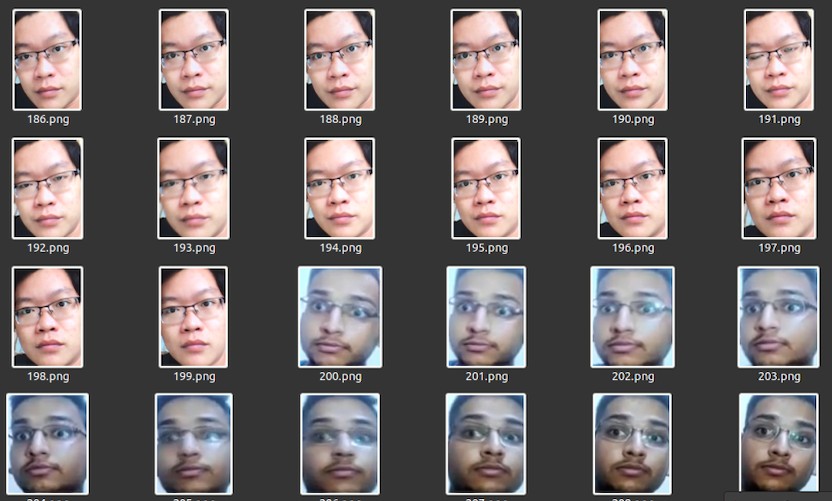


Fig 22: dataset of spoof images

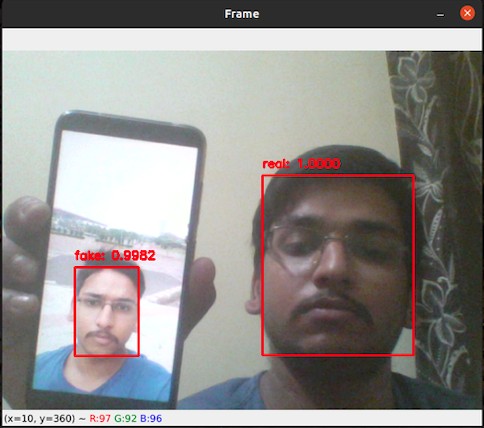


Fig 23: Output of demo file

# Chapter 4: Analysis

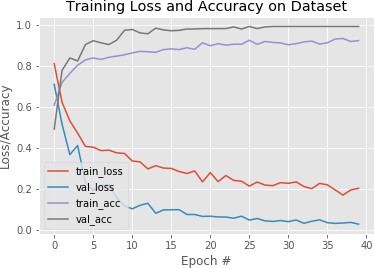


Fig 24: graph of (height: 16, width: 16)

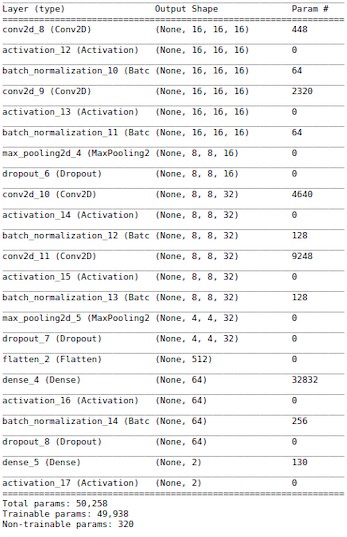


Fig25: model summary of (height: 16, width: 16)

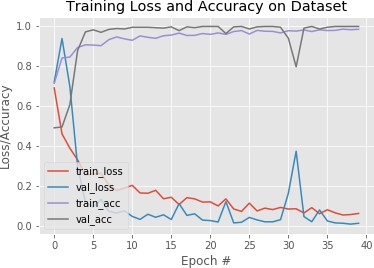


Fig 26: graph of (height: 32, width: 32)

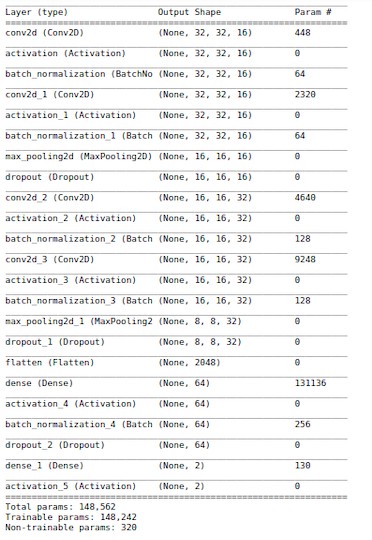


Fig 27:model summary of (height: 32, width: 32)

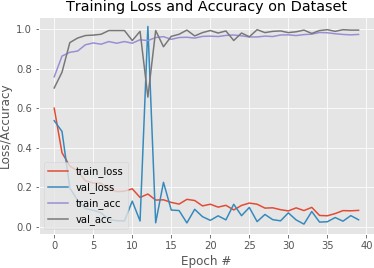


Fig 28: graph of (height: 64, width: 64)

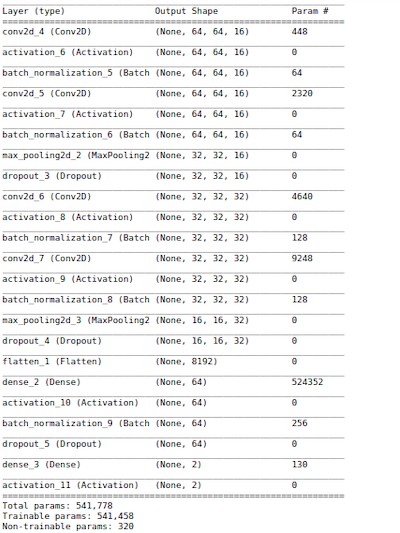


Fig 29: model summary of (height: 64, width: 64)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Trainable parameters | Non-trainable parameters | Total parameters |
| height :16, width: 16 | 49938 | 320 | 50258 |
| height :32, width: 32 | 148242 | 320 | 148562 |
| height :64, width: 64 | 541458 | 320 | 541778 |

Table: comparison between different height and width input with model's parameter

|  |  |
| --- | --- |
|  | Nth epoch |
| height :16, width: 16 | 13 |
| height :32, width: 32 | 5 |
| height :64, width: 64 | 7 |

Table: On which epoch , validation accuracy achieved the 100% accuracy

# Chapter 5: Result

We have successfully created our project to recognise the user face and unlock the device. In this report, we have also attached some snapshots of our running project.

# Chapter 6: Conclusion

All the above mentioned methods have better results compared to previous algorithms as they are used alone or as a single algorithm but when we combine the two different approaches the results are good under different conditions such as facial expression, light on the face ,picture quality, different dimension. But these algorithms have sum limitations as noise when we capture the image, light at the time of capturing the image, rotation of the image, translation of the images. This method can do better in the upcoming future as we know about more methods or when we combine the DL methods.

### Previous Work Summary

1. Studied different research papers on Face Recognition and work related to it.
2. Studied about the different possible ways of implementing Face Recognition and their advantages and disadvantages.
3. Researched on all the new works being done in this area and implemented our technique via different approaches using eyes, face structure, face cut as parameters
   1. **Future Plan**

### Include more parameters in the existing model

* + - * Apart from the current parameters that these projects have, more parameters can be included to make the system more reliable and full-proof.

### Improve the accuracy.

As of this moment the model has an accuracy of over 90%, we need to improve it and compare it to other models.

### Improve the scanning of the features.

We will improve the scanning of the features so that the model can figure out whether a use is himself standing in front of the camera or any other person is trying to open up the device just by using a user’s photograph. We will focus on the features that we have used in this project to recognise the face so that the project can figure out the difference between a 2D and a 3D image.

# Chapter 7: References

1. Liveness Detection with OpenCV by Adrian (Rosebrock) on March 11, 2019
2. An overview Of Face Liveness Detection Saptarshi Chakraborty1 and Dhrubajyoti Das2 1,2Dept. of Computer Science and Engineering, National Institute of Technology, Silchar, India.
3. How to Build a Machine Learning Model A Visual Guide to Learning Data Science
4. A Practical Introduction to Deep Learning with Caffe and Python by Adil Moujahid
5. <https://www.thalesgroup.com/en/markets/digital-identity-and> security/government/inspired/liveness-detection
6. Local vs global: Intelligent Local Face Recognition Daniel Riccio University of Naples, Federico II, Italy Andrea Casanova, Gianni Fenu University of Cagliari, Ital
7. A Face Recognition System Based on Local Feature Characterization, Paola CampadelliRaffaella Lanzarotti
8. 3D Face Recognition, By Naser Zaeri
9. A REVIEW OF FACE RECOGNITION METHODSM. PARISA BEHAM, S. MOHAMED MANSOOR ROOMI
10. Three-dimensional face recognition, From Wikipedia, the free encyclopedia
11. 3D face recognition: A survey. Song Zhou, Sheng Ziau
12. 3D Face RecognitionBerk G¨okberk, Albert Ali Salah, Ne¸se Aly¨uz, Lale Akarun
13. Face Recognition Methods & Applications Divyarajsinh N. Parmar1, Brijesh B. Mehta
14. Face Recognition: Understanding LBPH Algorithm, Kelvin Salton do Prado
15. Face Recognition: Issues, Methods and Alternative Application, By Waldemar Wójcik,

Konrad Gromaszek and Muhtar Junisbekov

1. Huang R.-J. J., "Detection strategies for face recognition using learning and evolution". (1998)
2. Jain A. K. and Dubes R. C., "Algorithms for clustering data", Prentice-Hall, Inc. (1988)
3. W. Zhao, A. Krishnaswamy, R. Chellappa, D.L. Swets,J. Weng, Discriminant analysis of principal components for face recognition, Face Recognition: From Theory to Applications, Springer, Berlin , 1988.
4. Pearson K., LIII. On lines and planes of closest fit to systems of points in space, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2(11), 559-572 (1901)
5. Thurstone L. L., Multiple factor analysis, (1947)
6. Sirovich, L. and M. Kirby, “Low-Dimensional procedure for the characterization of human faces”, J. Optical Soc. Am., 4: 519-524, 1987.
7. Pentland, B. Moghaddam and T. Starner, “View- Based and modular eigenspaces for face recognition”, Proceeding of IEEE CS Conference on Computer Vision and Pattern Recognition, pp: 84-91, 1994.
8. Dimitri, “Eigenface-based facial recognition”, available online at <http://openbio.sourceforge.net/resources/eigenfaces/eigenfaceshtml/> facesOptions.html, February 13, 2003.
9. Mika S., Schölkopf B., Smola A. J., Müller K.-R., Scholz M. and Rätsch G., Kernel PCA and De-Noising in Feature Spaces, NIPS, 536-542 (1998)
10. Sharif M., Khalid A., Raza M. and Mohsin S., Face Detection and Recognition Through Hexagonal Image Processing, Sindh Univ. Res. Jour. (Sci. Ser.), 44(4), 541- 548 (2012).