American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-13, Issue-4, pp-188-206
www.ajer.org
Research Paper Open Access

Facial Recognition and Identification for Verifying Examination Candidates

ABSTRACT
This report contains the ways in which computer vision an important aspect of software engineering field can be used to detect and identify the face of an individual, using several python libraries in Python along with OpenCV. This report also contains a proposed source code which will help in detecting human face in real time. This implementation can be used at various platforms in machines, desktops, smartphones, websites and several software applications. Facial recognition is a modern technology capable of matching faces gotten from a video input media or image against a database of face. It is employed to authenticate users through ID verification service and works by comparing facial features from a given image with a database. In this project, an accuracy of recognition was achieved after increasing the size of the training data set to 20 images. This result was arrived at on a machine with system specification of Core i5, 16gb RAM, and 1080p camera.

Date of Submission: 09-04-2024 Date of acceptance: 21-04-2024

2.0 LITERATURE Of Related Works

This chapter describes a face detection, recognition and identification, it's benefits and related work done by others on face detection or face recognition technology.

2.1 COMPUTER VISION

Computer vision is a field of software engineering that deals with how computer programs or machines acquire and process images or video feeds. Some relevant techniques include text detection, object detection and tracking, facial recognition, pose recognition and much more.Technology giants like Facebook use some of these techniques to identify faces of its users from their pictures.

2.2 FACE DETECTION, ANALYSIS AND RECOGNITION

Face detection is a specialized form of object detection that locates human faces in an image. This can be combined with classification and facial geometry analysis techniques to infer details such as age and emotional state; and even recognize individuals based on their facial features.

2.3 OPEN CV

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

2.4 HARR CASCADE

Haar Cascade is a widely used algorithm for facial recognition using python and openCV. Haar Cascade works by looking for features in an image or video stream called Haar features. These features are built around facial features such as the eyes nose and mouth. These regions of a face have characteristic pixel properties in which some pixel areas are dark while some other areas are bright. The image is first converted to gray style before Haar Cascade algorithm can be used. Haar Cascade searches through the image or video stream from top to bottom until it finds the required facial features.

2.5 FACE CLASSIFIER

A face classifier is a file that has been trained using a huge set of data to detect faces. In order to use the Haar Cascade algorithm, a face classifier file needs to be attached to a known location in your computer, preferably the default storage location for your python programs. There are Many Haar Cascade face classifier available online for download.

2.6 ANALYSIS OF EXISTING RELATED WORK

I. **Suma S L et al 2018**, implemented a real time face recognition algorithm using Linear Binary Pattern Histogram (LBPH) and Viola Jones algorithm. This method consists of com fusion and recognition is done using Viola Jones algorithm. It is applied is for Face detection, feature extraction is done by LBPH technique and Euclidean Distance

Classifier is used for face recognition. This work had recognition rate of about "85%-95%". This work can be further amended to favor in all conditions such as brightness, In case of twins, beard and wearing goggles.

II. **Li Cuimei et al 2017,** implemented a human face detection algorithm using three weak classifiers including Haar cascade classifier. Skin hue histogram, Eye detection and Mouth detection are the three classifiers adopted by this method. This yields sufficiently high detection. The proposed method generates a position prediction value (PPV) to about 78.18% - 98.01%. This can be amended to detect human faces only of multiple

races and reduce the delay for detecting and recognizing various faces among different images of people with variation in light and background conditions.

III. **Souhail Guennouni et al 2017,** implement a face detection system by collating with

Haar cascade classifiers and edge orientation matching. Edge orientation matching algorithm and Haar-like feature selection combined cascade classifiers are the two techniques used in this system. This algorithm produces a better matching but the Detection speed is comparatively less.

IV. **Jiwen Lu using learning CBFD et al 2015,** proposed a face recognition system. The face representation and recognition is implemented via Compact Binary Face Descriptor (CBFD) feature learning method while coupled CBFD is executed for heterogeneous face matching by minimizing the modality gap of feature level. Collating with other Binary Codes Learning techniques, CBFD extracts compact and discriminative feature, hence produces a better recognition rate of about 93.80% is obtained. In this work, feature is learned only from one single layer. This system can achieve better performance by Learning Hierarchal features in deep networks.

V. **Venkatesh Bhutra et al. 2018,** With the world moving towards advanced technologies, security forms a crucial part of daily life. Among the many techniques used for this purpose, Face Recognition stands as an effective means of authentication and security. This paper deals with the use of principal components and security. PCA is a statistical approach used to simplify a data set. The minimum Euclidean distance found from the PCA technique is used to recognize the face. Raspberry Pi a low-cost ARM-based computer on a small circuit board, controls the servo motor and other sensors. The servo-motor is in turn attached to the doors of the home and opens up when the face is recognized.

VI. **Karan et al 2017,** Privacy and Security are two universal rights and to ensure that in our daily life we are secure, a lot of research is going on in the field of home security, and lOT is the turning point for the industry, where we connect everyday objects to share data for our betterment. Facial recognition is a well-established process in which the face is detected and identified out of the image. We aim to create a smart door, which secures the gateway on the basis of who we are. In our proof of concept of a smart door we have used a live HD camera on the front side of the setup attached to a display monitor connected with the camera to show who is standing in front of the door, also the whole system will be able to give voice outputs by processing text them on the Raspberry Pi ARM processor used and show the answers as output on the screen. We are using a set of electromagnets controlled by the microcontroller, which will act as a lock. So, a person can open the smart door with the help of facial recognition and at the Same time also be able to interact with it.

VII. **Saurabh Bajaj et al 2020,** Facial recognition used to be a futuristic idea. Now, however, we use face ID to access our smartphones and other smart devices every day. It's a secure and convenient form of access to our information. Now, face recognition is creating the same secure and convenient access to our physical spaces. Face recognition access control has quickly become the emerging technology in physical access due to a shift towards health-conscious solutions that also offer incredible security. Face recognition can now be used to provide access to buildings in a safe, touchless manner. Instead of a key card, systems provide a camera-based-reader, that authorizes entry using the person's face. All you have to do is walk up and look at the reader, it runs recognition, identifies the person, and sends a signal to unlock the doors.

VIII. **Singh, D et al 2020,** proposed a COVID-19 disease classification model to classify infected patients from chest CT images. a convolutional neural network (CNN) is used to classify COVID-19-infected patients as infected (+ve) or not (−ve). Additionally, the initial parameters of CNN are tuned using multi-objective differential evolution (MODE). The results show that the proposed CNN model outperforms competitive models, i.e., ANN, ANFIS, and CNN models in terms of accuracy, F-measure, sensitivity, specificity, and Kappa statistics by 1.9789%, 2.0928%, 1.8262%, 1.6827%, and 1.9276%, respectively.

IX. **Schiller, D et al 2020,** proposed a novel approach to transfer learning to automatic emotion recognition (AER) across various modalities. The proposed model used for facial expression recognition that utilizes saliency maps to transfer knowledge from anarbitrary source to a target network by mostly "hiding" non-relevant information. The proposed method is independent of the employed model since the experience is solely transferred via augmentation of the input data. The evaluation of the proposed model showed that the new model was able to adapt to the new domain faster when forced to focus on the parts of the input that were considered relevant sources.

X. **Prakash, R et al 2020,** proposed an automated face recognition method using Convolutional Neural Network (CNN) with a transfer learning approach. The CNN with weights learned from pre-trained model VGG-16. The extracted features are fed as input to the Fully connected layer and softmax activation for classification. Two publicly available databases of face images–Yale and AT&T are used to test the performance of the proposed method. Face recognition accuracy of 100% is achieved for AT&T database face images and 96.5% for Yale database face images. The results show that face recognition using CNN with transfer learning gives better classification accuracy in comparison with PCA method.

XI. Deng et al 2020, proposed additive angular margin loss (ArcFace) to accomplish face acknowledgment. The proposed ArcFace has an unmistakable geometric understanding as a result of the specific correspondence to geodesic separation on a hypersphere. They also introduced the broadest exploratory assessment against the FR method utilizing ten FR datasets. They indicated that ArcFace reliably beats the best in class and can be effectively actualized with irrelevant computational overhead. The verification performance of open-sourced FR models on LFW, CALFW, and CPLFW datasets reached 99.82%, 95.45%, and 92.08%, respectively.

XII. **Wang et al 2020,** proposed a large margin cosine loss (LMCL) by reformulating the SoftMax loss as a cosine loss by L2 normalizing the two highlights and weight vectors to evacuate outspread varieties and using the cosine edge term to expand the choice edge in precise space. They achieved the highest between-class difference and lowest intraclass fluctuation via cosine choice edge augmentation and normalization. They referred to their model, trained with LMCL, as CosFace. They based their experiment on the Labeled Face in the Wild (LFW), YouTube Faces (YTF), and MegaFace Challenge datasets. They confirmed the efficiency of their proposed approach, achieving 99.33%, 96.1%, 77.11%, and 89.88% accuracy on the LFW, YTF, MF1 Rank1, and MF1 Veri datasets,respectively.

XIII. **Tran et al 2020,** proposed a disentangled representation learning-generative adversarial network (DR-GAN) with three different developments. First, the encoder-decoder structure of the generator permits DR-GAN to gain proficiency with a discriminative and generative portrayal, including picture blending. Second, the portrayal is unraveled from other face varieties—for example, through the posture code given to the decoder and posture estimation in the discriminator. Third, DR-GAN can accept one or various pictures as information and produce one integrated portrayal alongside an arbitrary number of manufactured pictures. They tested their network using the Multi-PIE database. They contrasted their strategy and face acknowledgment techniques with Multi-PIE, CFP, and IJB-A and achieved average face confirmation exactness with greater than tenfold standard deviation. They accomplished equivalent execution on frontal-frontal confirmation with ~1.4% enhancement for frontal-profile verification.

XIV. Masi et al 2020, proposed to build prepared information sizes for face acknowledgment frameworks: domain explicit information development. They presented techniques to enhance realistic datasets with critical facial varieties by controlling the faces in the datasets while coordinating inquiry pictures presented by standard convolutional neural systems. They tested their framework against the LFW and IJB-A benchmarks and Janus CS2 on a large number of downloaded pictures. They reported the standard convention for unhindered, marked outside information and announced a mean grouping precision of 100% equal error rate.

XV. **Ding and Tao et al 2020,** proposed a far-reaching system based on convolutional neural networks (CNN) to overcome the difficulties faced in video-based face recognition (VFR). CNN learns obscure highlights by utilizing prepared information comprising misleadingly obscured information and still pictures. They proposed a trunk-branch ensemble CNN model (TBE-CNN) to improve CNN highlights to present varieties and impediments. TBE-CNN separates data from face pictures and zones picked around facial segments. TBE-CNN removes information by sharing the center and low-level convolutional layers between the branch and trunk systems. They proposed an improved triplet misfortune capacity to invigorate the influence of discriminative portrayals learned by TBE-CNN. TBE-CNN was tested on three video face databases: YouTube, COX Face, and PaSC Faces.

XVI. **Al-Waisy, et al 2020,** proposed a multimodal profound learning system that depends on nearby element presentation for k-based face acknowledgment. They consolidated the focal points of neighborhood handmade component descriptors with the DBN to report face acknowledgment in unconstrained circumstances. They proposed a multimodal nearby component extraction approach dependent on consolidating the upsides of fractal measurement with the curvelet change, and they called it the curvelet– fractal approach. The principal inspiration of this methodology is that the curvelet change can expertly present the fundamental facial structure, while the fractal measurement presents the surface descriptors of face pictures. They proposed a multimodal profound face acknowledgment (MDFR) approach, to include highlight presentation by preparing a DBN on nearby element portrayals. They compared the outcomes of the proposed MDFR approach with the curvelet–fractal approach on four face datasets: the LFW, CAS-PEAL-R1, FERET, and SDUMLA-HMT databases. The outcomes acquired from their proposed approaches outperformed different methodologies including WPCA, DBN, and LBP by accomplishing new outcomes on the four datasets.

XVII. Sivalingam et al 2020, proposed a proficient fractional face location strategy utilizing AlexNet CNN to detect emotions based on images of half-faces. They distinguished the key focal points and concentrated on textural highlights. They proposed an AlexNet CNN strategy to discriminatively coordinate the two removed nearby capabilities, and both the textural and geometrical data of neighborhood highlights were utilized for coordination. The comparability of two appearances was changed according to the separation between the adjusted capabilities. They tested their approach on four generally utilized face datasets and demonstrated the viability and constraints of their proposed method.

XVIII. **Jonnathann et al 2020,** presented a comparison between profound learning and conventional AI strategies (for example, artificial neural networks, extreme learningmachine, SVM, optimum-path forest, KNN) and deep learning. For facial biometric acknowledgment, they concentrated on CNNs. They used three datasets: AR Face,YALE,and SDUMLA-HMT.

XIX. Pinh Hsin Lee et al 2019, analyzed different face detection techniques which included; Template matching, Colour segmentation and morphological processing, Eigne Faces and SVM. Implementations of techniques such as FLD and SVM worked about as well as template matching.

XX. **Andreea Diana Dinca et al 2017,** worked on facial recognition using Microsoft Azure Face API. However, they discovered that there are a few restrictions when using
Microsoft's Face API: a free account can make 30.000 calls to the API per month and 20 per minute, whereas paid accounts can make 10 calls per second; only 64 face lists are allowed in one subscription; a face list cannot have more than 1000 faces; once a face has been added to a face list, the user receives the ID that was associated to that face, but there is no way to physically see what face/picture is represented by that ID anymore.

From the listed projects above, all local face recognition architecture are outdated. But with the advancement of OpenCV and Harr Cascade, it is possible to implement a facial recognition architecture with a 85% - 99% accuracy.

SUMMARY TABLE FOR ANALYSIS OF RELATED WORK

AUTHOR'S NAME	WORK DONE	FINDINGS (PARAMETER)	RECOMMENDATION
Suma S L et al 2018	Face Recognition Algorithm using Linear Binary Pattern Histogram (LBPH) and Viola Jones Algorithm	It had recognition rate of about 85-95%	This work can be further amended to favor in all conditions such as brightness, in case of twins, beard and wearing goggles.
Li Cuimei et al 2007	Face recognition algorithm using Skin hue histogram, Eye detection and Mouth detection	This resulted to a Position Prediction Value (PPV) of about 78.18-98.01%	This can be improved to detect human faces of multiple races and reduce the delay of detecting and recognizing faces among different images of people with varied lighting conditions
Souhail Guennouni et al 2017	Face detection system using edge orientation matching and harr cascade	Produced a better matching but really slow in detection	Can be improved in the detection speed
Jiwen Lu using learning CBFD et al 2015	Face recognition system via Compact Binary Face Descriptor (CBFD) feature learning method	Produced a recognition rate of about 93.80%	This system can achieve better performance by Learning Hierarchal features in deep networks
Venkatesh Bhutra et al. 2018	Face recognition with PCA technique	Was used to control sensors on a door using Raspberry Pi	Can be improved using better technologies and sensors
Karan et al 2017	Face detection using HD Camera	Used Raspberry Pi to process text output on screen	Can be improved using better technologies to program the Raspberry Pi
Saurabh Bajaj et al 2020	Face recognition for unlocking doors	Using embedded system	Can be improved using better technologies to program the embedded system

Singh, D et al 2020	Covid-19 disease classification using chest CT images using CNN	It had over 90% accuracy	Improves as the Neural Network improves
Schiller, D et al 2020	Automatic Emotion Recognition (AER)		Improve when data focuses on relevant sources

16

Parkash, R et al 2020	Face recognition using CNN	Had accuracy of about 96.5%	Using CNN with transfer learning gives better classification accuracy
Deng et al 2022	Additive Angular Margin Loss (ArcFace) for face acknowledgment	Verification performance reached 95.45%	can be effectively actualized with irrelevant computational overhead
Wang et al 2020	CosFace, a model trained with Large Margin Cosine Loss (LMCL)	Efficiency of their proposed approach ranged from 77.11 % and above	Can be optimized by improving the model
Tran Et al 2020	Face acknowledgement using Learning-generative adversarial network (DR-GAN)	They accomplished equivalent execution on frontal-frontal confirmation with ~1.4% enhancement for frontal-profile verification.	Can be improved by improving network
Masi et al 2020	Face acknowledgement system	Error rate was equal to precision	Using Harr-cascade works better

Ding and Tao et al 2020	Face recognition using CNN	TBE-CNN was tested on three video face databases: YouTube, COX Face, and PaSC Faces	Improves as the neural network improves
Al-Waisl et al	K-based face	The result of their	Using cascades yields better

2020	acknowledgement	approach, outperformed different methodologies	result
Sivalingam et al 2020	Emotion detector using CNN	approach was tested on on four generally utilized face datasets and showed the viability and limitations of their proposed method	Improves as the neural network improves
Jonathamm et al 2020	Facial biometrics using CNN	They used three datasets: AR Face, YALE, and SDUMLA-HMT	Improves as the neural network improves
Ping Hsin Lee, Vivek Srinivasan, and Arvind Sundararajan	Face detection, using different techniques	Implementations of techniques such as FLD and SVM worked about as well as template matching	using a highly heuristic and scale-specific method of clustering correlation peaks works better
Andreea Diana Dinca et al 2017	Face recognition with Microsoft Face API	a free account can make 30.000 calls to the API per month and 20 per minute, whereas paid accounts can make 10 calls per second; only 64 face lists are allowed in one subscription	A work around to this issue is to keep track of all the face lists by creating a folder for each new list

TABLE 2.6 Summary of literature review

III. DESIGN METHODOLOGY

The Face Recognition and Identification permission to access the camera and storage of the desktop computer. The application takes a photo from a live video from the inbuilt camera as input and outputs the analyzed photo subject matter. Facial features may also comprise of regions inside the face, varieties within the face structure, face cuts and points which have been organized and styled. Face extraction incorporates grabbing of the capabilities from camera. Face detection incorporates the elimination of the background and focusing on the foreground eliminating a few other elements separated from the face region, but the gadget nevertheless has a few drawbacks since it cannot come across the head be counted which can be a gift because of overlapping of faces or mixed-up recognition of faces having similar facial functions.

3.1.1 FACE DETECTION USING OPENCV

OpenCV (Open-Source Computer Vision) is a library of programming functions for real-time computer vision. The face detection part of the project was made using an OpenCV Library for Scala. The reason was that most Face APIs are restricted to doing detection on pictures only, whereas the project was required to have face detection done on a live video footage to speed up the process of checking student attendance and prevent queues before lectures.

The OpenCV library proved to be flexible enough for the project as it can accurately detect a face in real time and highlight it by drawing a rectangle around the faces of the students passing by. This all happens in a window separate from the face recognition so the lecturer can keep track of both students passing by while having their faces detected and the feedback from the recognition part of the system. While faces are being detected, the application takes a snapshot of the live footage every second and then sends it to the recognition system.

Fig 3.0: OpenCv

3.1.2 FACE RECOGNITION WITH HARR CASCADE

In this paper, Viola Jones algorithm is adapted for face detection. AdaBoost algorithm is united with Viola Jones algorithm to make a strong classifier. Haar like features are adapted by Viola -Jones for face detection

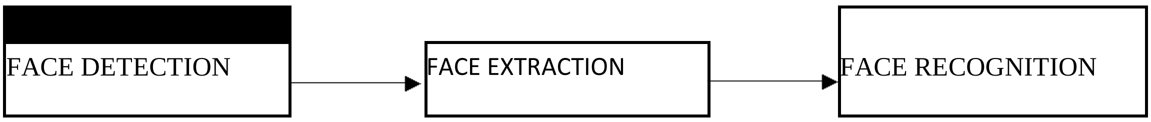

Fig. 3.1: Harr Cascade Block diagram

Examples for Haar-like features are displayed in Fig.2, which is to identify dissimilarities in dark and white regions of the images. From this representation, light region explain “to add” and dark region is “to subtract”.

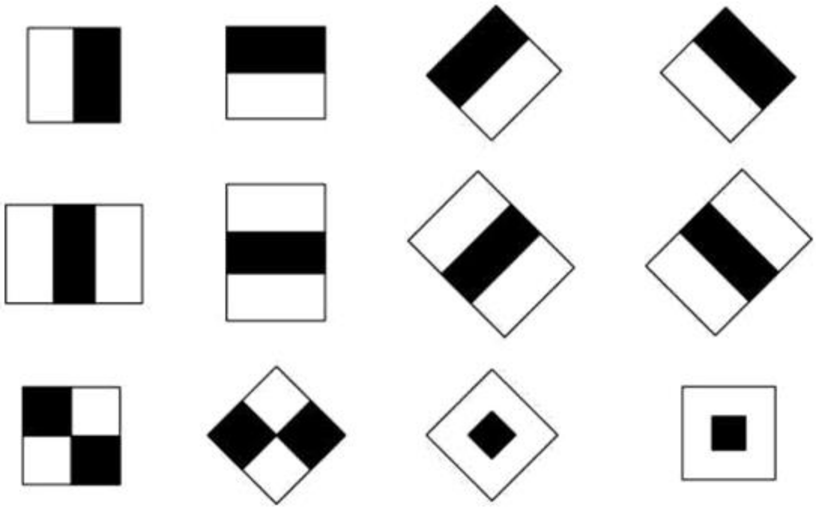

Fig. 3.2: Harr Like Features

Haar-like features is used by Haar cascade classifier for human face detection. There are three formations of Haar- like features. From the Fig.3, the first format is the edge feature, second type is the line feature and the last type is the four rectangle feature. Using the integral image, Haar like principle will provide fast computation. It’s called Haar-like features.

The Algorithm looks for specific haar feature of a face. This detection takes the image and converts it into 24X24 window and smears each Haar feature to that window pixel by pixel. Initially, the algorithm requires a lot of positive images (images of faces) and negative images (images without faces) to train the classifier.

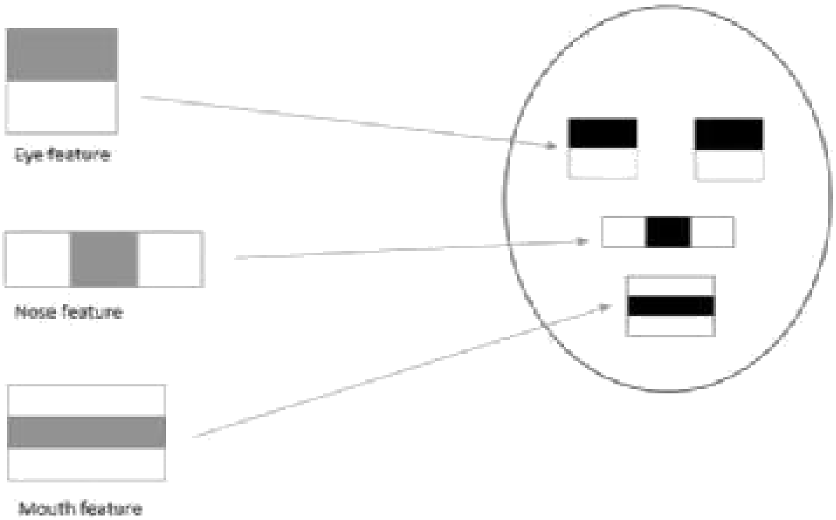

Fig. 3.3: Types of Haar- Like Features

Then, these features are extracted. Features are numerical values determined from images that are used to distinguish one image from another each feature is a single value acquired by subtracting the sum of the pixels beneath the white rectangle from the sum of the pixels beneath the black rectangle.

$$\text{Feature} = \Sigma\ (\text{pixels in black area}) - \Sigma\ (\text{pixels in white area}) \quad \text{(Eq. 1)}$$

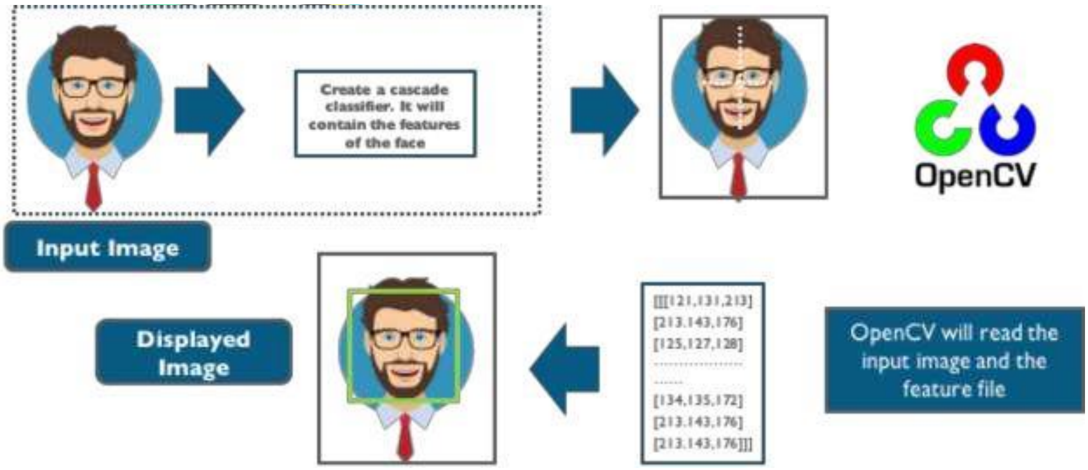

Fig. 3.4: Process of face detection and recognition

3.2 DESIGN TOOLS AND ANALYSIS

3.2.1 PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today

Fig. 3.2.1: Python

3.2.2 PYCHARM

PyCharm is an integrated development environment (IDE) used in computer programming, created for the Python programming language.

Fig 3.2.2 PyCharm

3.3 DESIGN ANALYSIS

This face recognition system can be analyzed using the following block diagram

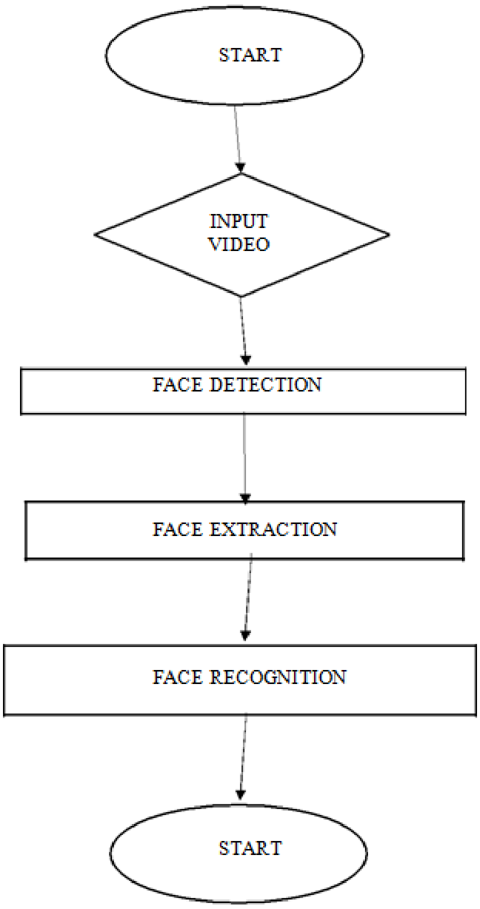

Fig 3.3: Flow chart showing process of the facial recognition system

IV. RESULTS (FIGURES & TABLES)

4.1 COMPARING RESULTS WITH A PREVIOUS PROJECT

PARAMETERS	AZURE FACE API	HARR CASCADE
Face Detection	Yes	Yes
Face Recognition (Image)	Yes	Yes
Face Recognition (Video)	No	Yes
Multi-Face Tracking	Yes	Yes
Customization (High/Low)	Low	High
Limitation	The free version has a max of 30 000 request per month, 20 per minute, only 64 face lists areallowed in one	Unlimited

	subscription; a face list cannot have more than 1000 faces	
License	Paid	Free
Internet Connection	Yes	Yes

Table 4.1: Comparing Microsoft Azure Face API with Harr Cascade

4.2 SWOT ANALYSIS

The following SWOT Analysis outlines the advantages and disadvantages a facial detection and recognition system can bring.

STRENGTH • It can be faster than other biometrics authentication methods. • There are a lot of models and methods that help build such a system. • It is easy to detect fake candidates. • Footage can be recorded to be manually checked. • It can be optimized to detect other face details such as the eye.	OPPORTUNITIES • It can speed up the process of verifying exam candidates. • Face recognition systems are becoming more reliable and accurate. • Such systems are easy to implement due to how advanced technology is today. • It offers protection against intruders as they can easily be detected.
WEAKNESS • It may not be as accurate as other biometrics authentication methods under bad environmental condition. • It depends on the distance and pose of the face as well as the quality, illumination of the image/video used.	THREATS • Other biometrics authentication methods can be more accurate. • Lightning can affect the result • It has to use a live video footage for recognition to be faster than other systems.

Table 4.2: SWOT Analysis

- Accessories such as glasses or hats may negatively impact the accuracy of the system.

4.3 FACIAL RECOGNITION vs BIOMETRIC AUTHENTICATION

PARAMETER	FACIAL RECOGNITION	BIOMETRIC AUTHENTICATION
SPEED	It is very fast	It could take time before biometric scanners detect the fingerprint
DURABILITY	Durability is dependent on the camera or video capture	Biometric scanners get weak with time because of the

	device	pressure being applied to it when used
ACCURACY	Accuracy is dependent on the lightning in the environment	It is very accurate
DEPENDENCIES	Performance ishighly dependent on the specification of camera	Performance is dependent on the technology used to build biometric sensor

Table 4.3: Comparing Facial Recognition and Biometric Authentication

The constantly changing state of biometrics technology makes it likely that fingerprint and facial recognition will both remain essential security tools in the near future. The global fingerprint sensor market is on track to achieve an annual compound growth rate of **14.5%** between **2020** and **2027**, while the global facial recognition market is positioned to grow at a comparable rate of **15.4%** between **2021** and **2028**. To keep up with biometrics trends and technology changes on the horizon, you need a security technology platform that stays up-to-date.

4.4 CONFIDENTIALITY, INTEGRITY AND AVAILABILITY (CIA) OF THE SOURCE CODE

Confidentiality has to do with keeping an organization's data private. This often means that only authorized users and processes should be able to access or modify data. In applying this source code to any platform or system, the images used to train the model will no longer be needed. Instead, the .XML file that contains labels of facial features will be used to recognize faces and will be inaccessible by anyone except the source code is being sold out.

Integrity means that data can be trusted. It should be maintained in a correct state, kept so that it may not be tampered with, and should be correct, authentic, and reliable. The integrity of the data used to train the model is dependent on the integrity of which ever organization that will use the source code. Therefore, if all that concerns the integrity of the organization is in check there will be no need to worry. Just as it is important that unauthorized users are kept out of an organization's data, data should be available to authorized users whenever they require it. This means keeping systems, networks, and devices up and running.

4.5 ACCURACY VS SIZE OF TRAINING DATA SET

In order to vet the accuracy of my source code, I tested the accuracy of the recognition after training my model with 3 images, 5 images, 10 images, 15 images and 20 images. And I observed that the larger the training dataset, the more accurate the recognition will be.

SIZE OF TRAINING DATA	ACCURACY (%)
3	80
5	90
10	95
15	98
20	100

Table 4.5: Showing size of training data and accuracy result

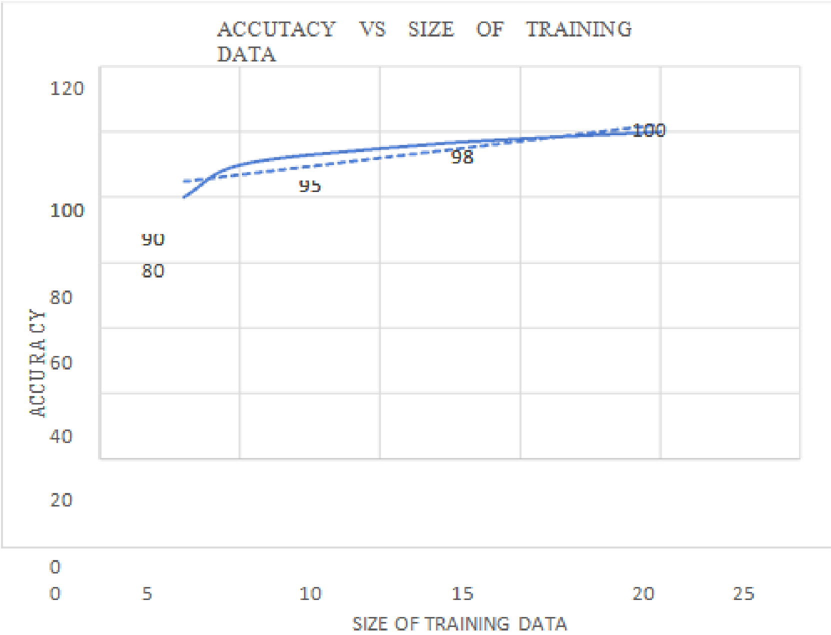

Fig 4.1: Graph showing Accuracy vs Training Data

V. CONCLUSION

A face detection and recognition system would certainly speed up the process of checking exam candidates in comparison to other biometrics authentication methods and in the right circumstances; it would be able to match their accuracy. Face recognition systems are currently associated with many top technological companies and industries making the work of face recognition easier. The use of python programming and OpenCV makes it an easier and handy tool or system which can be made by anyone according to their requirement. The proposed application discussed in this project will be helpful for many as it is user friendly and a cost efficient system. Nowadays there are a wide variety of software, whether it is a library like OpenCV or a Face API like Microsoft's, that makes face detection and recognition accessible and reliable and is constantly improving. Hence by the use of python and OpenCV the face recognition system can be designed for various purposes.

5.3 RECOMMENDATION

If this system would be implemented in a university, the following would be recommended:

- The operator of the desktop computer that would be used for the face recognition should stand at the entrance of the examination room at a distance of 2 feet to the subject. This would assure a better accuracy from the face recognition system.

- The environment should be properly lightened.

At least, 15 clear head shots of each exam candidates should be used as the training data.The pictures that are added to face lists should be clear and less than 3 megabytes in size.

APPENDIX

SCREENSHOTS OF CODE SNIPPET

```
import cv2
import os
import numpy as np
from PIL import Image
import pickle

BASE_DIR = os.path.dirname(os.path.abspath(__file__))
image_dir = os.path.join(BASE_DIR, 'images')

faceCascade = cv2.CascadeClassifier(r"cascades\data\haarcascade_frontalface_alt2.xml")
recognizer = cv2.face.LBPHFaceRecognizer_create()


current_id = 0
label_ids = {}
y_labels = []
x_train = []

for root, dirs, files in os.walk(image_dir):
    for file in files:
        if file.endswith('png') or file.endswith('jpg') or file.endswith('JPG'):
```

FIG 5.1a Faces-train.py Code to train face data set

```
            path = os.path.join(root, file)
            label = os.path.basename(os.path.dirname(path)).replace(' ', '-').lower()
            print(label, path)
            if label in label_ids:
                pass
            else:
                label_ids[label] = current_id
                current_id += 1

            id_ = label_ids[label]
            print(label_ids)
            pil_image = Image.open(path).convert('L')
            image_array = np.array(pil_image, 'uint8')
            print(image_array)
            faces = faceCascade.detectMultiScale(image_array, scaleFactor=1.05, minNeighbors=5, minSize=(30, 30))

            for (x, y, w, h) in faces:
                roi = image_array[y:y+h, x:x+w]
                x_train.append(roi)
                y_labels.append(id_)
```

FIG 5.1b Faces-train.py Code to train face data set

```python

            for (x, y, w, h) in faces:
                roi = image_array[y:y+h, x:x+w]
                x_train.append(roi)
                y_labels.append(id_)

print(y_labels)
print(x_train)

with open('labels.pickle','wb') as f:
    pickle.dump(label_ids, f)

recognizer.train(x_train, np.array(y_labels))
recognizer.save('trainner.yml')
```

FIG 5.1c Faces-train.py Code to train face data set

faces-train.py   faces.py

```python
import cv2
import pickle

# read face classifier file
faceCascade = cv2.CascadeClassifier(r"cascades\data\haarcascade_frontalface_alt2.xml")
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainner.yml')

labels = {}
with open('labels.pickle', 'rb') as f:
    og_labels = pickle.load(f)
    labels = {v:k for k, v in og_labels.items()}

video_capture = cv2.VideoCapture(0)

while True:

    # capture frame by frame
    ret, frame = video_capture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

```

FIG 5.2a Faces.py Code to detect faces in live video faces-train.py | faces.py

```python

        # detect faces
        faces = faceCascade.detectMultiScale(gray, scaleFactor=1.05, minNeighbors=5, minSize=(30, 30))

        # Draw a rectangle around the faces
        for (x, y, w, h) in faces:
            #print(x, y, w, h)
            roi_gray = gray[y:y+h, x:x+w]
            roi_color = frame[y:y+h, x:x+w]
            id_, conf = recognizer.predict(roi_gray)
            if conf>= 45 and conf <= 85:
                print(id_)
                print (labels[id_])
                font = cv2.FONT_HERSHEY_SIMPLEX
                name = labels[id_]
                color = (255, 0, 0)
                stroke = 1
                cv2.putText(frame, name, (x,y), font, 0.5, color, stroke, cv2.LINE_AA)

            img_item = 'test_image.png'
            img_item2 = 'test_image2.jpeg'
```

FIG 5.2b Faces.py Code to detect faces in live video

faces-train.py | faces.py

```python
            img_item = 'test_image.png'
            img_item2 = 'test_image2.jpeg'
            cv2.imwrite(img_item, roi_gray)
            cv2.imwrite(img_item2, roi_color)

            cv2.rectangle(frame, (x,y), (x+w,y+h), (255, 0, 0))



        # sDisplay resulting frame
        cv2.imshow('frame', frame)
        if cv2.waitKey(1) & 0xFF == ord("q"):
            break
    print("Face count is: ", len(faces))
    video_capture.release()
    cv2.destroyAllWindows()
```

FIG 5.3 Faces.py Code to recognize faces

REFERENCES

[1]. Wikipedia, “Python Programming Language”, [Online], Available: https://en.wikipedia.org/wiki/Python_(programming_language)

[2]. Coursera, “Python”, [Online], Available: https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python

[3]. Data Bricks, “PyCharm”, [Online], Available: https://databricks.com/glossary/what-is-pycharm

[4]. Visual Studio Code, “Visual Studio Code”, [Online], Available: https://code.visualstudio.com/docs

[5]. Andreea Diana Dinca, “Face Detection & Recognition”, IT Innovation Intern Report, October 2017 Department of Electrical Engineering, Polytechnic University, Brooklyn. Zhu Liu and Yao Wang

[6]. Face Recognition Homepage,[Online],Available: http://www.facerec.org/algorithms/

[7]. Harshada B. More, Anjali R. Bodkhe, “Survey Paper on Door Level Security using Face Recognition International Journal of Advanced Research in Computer and Communication Engineering ISO 3297-2007 Certified Vol. 6, Issue 3, March 2017.

[8]. Saurabh Batal, “Guide to Face Recognition Access Control" September 16, 2020

[9]. Keiki Prasade, Supriya Nalavade, Devaki Pathak, "FACE RECOGNITION BASED DOOR LOCKING SYSTEM" International Research Journal of Engineering and Technology (IRJET), 07 July 2018, Page 1253
[10]. Thulluri Krishna Vamsi, Kanchana Charan Sai, Vijayalakshmi M, Face recognition-based door unlocking system using Raspberry Pi, "International Journal of Advance Research Ideas and innovations in Technology.
[11]. https://www.academia.ed 43407917/Face Recognition Image Processing ed Door Lock_using OpenCV Python and Arduino
[12]. PLOS ONE, "Common recent works", [Online], Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0242269.
[13]. M.Hunke and A. Waibel, "Face Locating And Tracking for Human Computer Interaction",in Proc. Conf. Signals Systems and Computers, Nov 1994, vol2 pp 1277- 1281.
[14]. D. Cahi and K.N. Ngan, "Face segmentation using skin color map", IEEE Trans. On Circuit and Systems for Video Technology, vol 9, no 4, pp. 551-564, Jun. 1999.
[15]. Abbas El Gamal, "Introduction To Statistical Signal Processing", Lecture Notes.
[16]. Richard Duda, Peter Hart and David Stork, "Pattern Classification", 2nd ed., John Wiley and Sons.
[17]. Sirovich, L. and Kirby, M.: 1987, Low dimensional procedure for the characterization of human faces, J. Opt. Soc. Am. A 4, 519–524.
[18]. Merl, Gender classification with support vector machines, [Online], Available: www.merl.com/reports/docs/TR2000-01.pdf
[19]. Mohapatra, H. I. T. E. S. H. "HCR using neural network", PhD dissertation Biju Patnaik University of Technology, 2009.
[20]. Mohapatra, H., & Rath, A. K., "Detection and avoidance of water loss through municipality taps", in India by using smart taps and ICT. IET wireless sensor systems, page 447-457.
[21]. Mohapatra, H., & Rath, A. K. "Fault tolerance in WSN through PE-LEACH protocol", IET wireless sensor systems, 9(6), 358-365, 2019.
[22]. Mohapatra, H., Debnath, S., & Rath, A. K. "Energy management in wireless sensor network through EB-LEACH", International journal of research and analytical reviews (IJRAR), page 56-61, 2019.
[23]. Nirgude, V., Mahapatra, H., & Shivarkar, S. "Face recognition system using principal component analysis & linear discriminant analysis method simultaneously with 3d morphable model and neural network BPNN method", Global journal of advanced engineering technologies and sciences, 4(1), 1-6, 2017.
[24]. Panda, M., Pradhan, P., Mohapatra, H., & Barpanda, N. K. "Fault tolerant routing in heterogeneous environment. International journal of scientific & technology research, 8, 1009-1013, 2019.
[25]. Mohapatra, H., & Rath, A. K. "Fault-tolerant mechanism for wireless sensor network", IET wireless sensor systems, 10(1), 23-30, 2019.
[26]. Swain, D., Ramkrishna, G., Mahapatra, H., Patr, P., & Dhandrao, P. M. "A novel sorting technique to sort elements in ascending order. International journal of engineering and advanced technology", page 212-126, 2013.
[27]. Broumi, S., Dey, A., Talea, M., Bakali, A., Smarandache, F., Nagarajan, D & Kumar, R.
[28]. "Shortest path problem using Bellman algorithm under neutrosophic environment. Complex & intelligent systems", page 409-416, 2019.
[29]. Kumar, R., Edalatpanah, S. A., Jha, S., Broumi, S., Singh, R., & Dey, A. A multi objective programming approach to solve integer valued neutrosophic shortest path problems. Neutrosophic sets and systems, 24, 134-149, 2019.
[30]. Kumar, R., Dey, A., Broumi, S., & Smarandache, F. "A study of neutrosophic shortest path problem" In Neutrosophic graph theory and algorithms page148-179. IGI Global, 2020
[31]. Kumar, R., Edalatpanah, S. A., Jha, S., & Singh, R. "A novel approach to solve gaussian valued neutrosophic shortest path problems", International journal of engineering and advanced technology, 8(3), 347-353, 2019.
[32]. Kumar, R., Edalatpanah, S. A., Jha, S., Gayen, S., & Singh, R. "Shortest path problems using fuzzy weighted arc length". International journal of innovative technology and exploring engineering, 8(6), 724-731, 2019.
[33]. Kumar, R., Edaltpanah, S. A., Jha, S., & Broumi, S. "Neutrosophic shortest path problem. Neutrosophic sets and systems, 23(1), 2, 2018.
[34]. Kumar, R., Jha, S., & Singh, R. "A different approach for solving the shortest path problem under mixed fuzzy environment" International journal of fuzzy system applications (IJFSA), 9(2), 132-161, 2020.
[35]. Kumar, R., Jha, S., & Singh, R. Shortest path problem in network with type-2 triangular fuzzy arc length. Journal of applied research on industrial engineering, 4(1), 1-7, 2017.
[36]. Kumar, R., Edalatpanah, S. A., Jha, S., & Singh, R. "A Pythagorean fuzzy approach to the transportation problem", Complex & intelligent systems, 5(2), 255-263, 2019.
[37]. Pratihar, J., Kumar, R., Dey, A., & Broumi, S. "Transportation problem in neutrosophic
[38]. Environment" In Neutrosophic graph theory and algorithms (pp. 180-212). IGI Global, 2020
[39]. Mohapatra, H., Panda, S., Rath, A., Edalatpanah, S., & Kumar, R. :A tutorial on PowerShell pipeline and its loopholes. International journal of emerging trends in engineering research, 8(4), 975-982, 2020. Pratihar, J., Kumar, R., Edalatpanah, S. A., & Dey, A. "Modified Vogel's approximation method for transportation problem under uncertain environment" Complex & intelligent systems, 1-12, 2020.
[40]. Gayen, S., Smarandache, F., Jha, S., & Kumar, R. "Interval-valued neutrosophic subgroup based on interval-valued triple t-norm. In Neutrosophic sets in decision analysis and operations research (pp. 215-243). IGI Global, 2020.
[41]. Gayen, S., Smarandache, F., Jha, S., Singh, M. K., Broumi, S., & Kumar, R. "Introduction to plithogenic subgroup" In Neutrosophic graph theory and algorithms (pp. 213-259). IGI Global, 2020
[42]. Gayen, S., Jha, S., Singh, M., & Kumar, R. "On a generalized notion of anti-fuzzy subgroup and some characterizations" International journal of engineering and advanced technology, 8, 385-390, 2019.
[43]. Roy, S., Roy, S., & Bandyopadhyay, S. K. "A tutorial review on face detection" Intl. J. of engineering research & technology, 1(8), 10, 2012.
[44]. Singh, V., Shokeen, V., & Singh, B. "Face detection by Haar cascade classifier with simple and complex backgrounds images using opencv implementation" International journal of advanced technology in engineering and science, page 33-38, 2013.
[45]. Roy, S., & Podder, S. "Face detection and its applications" International journal of research in engineering & advanced technology, page1-10, 2013.
[46]. Parkhi, O. M., Vedaldi, A., & Zisserman, A. "Deep face recognition" [online] Retrieved from https://ora.ox.ac.uk/objects/uuid:a5f2e93f-2768-45bb-8508 74747f85cad1/download_file?file_format=pdf&safe_filename=parkhi15.pdf&type_of_work =Conference+item

[47]. Tolba, A. S., El-Baz, A. H., & El-Harby, A. A. “Face recognition: a literature
[48]. Review” International journal of signal processing, page 88-103, 2006.
[49]. Sun, X., Wu, P., & Hoi, S. C. “Face detection using deep learning: an improved faster RCNN approach” Neurocomputing, [online] https://doi.org/10.1016/j.neucom.2018.03.030, 2018.
[50]. Puri, R., Gupta, A., Sikri, M., Tiwari, M., Pathak, N., & Goel, S. “Emotion detection using image processing in python” In 5th international conference on “computing for sustainable global development (IndiaCom)” IEEE conference ID (Vol. 42835), 2018.
[51]. Klug, A., & De Rosier, D. J. “Optical filtering of electron micrographs: reconstruction of one-sided images” Nature, 212(5057), 29-32, 1966.
[52]. Billingsley, F. C. “Applications of digital image processing” Applied optics, page 289-299, 1970.
[53]. Andrews, H., & Patterson, C.” Singular value decompositions and digital image
[54]. Processing” IEEE transactions on acoustics, speech, and signal processing, 24(1), 26-53, 1976.
[55]. Goetcherian, V. “From binary to grey tone image processing using fuzzy logic concepts.
[56]. Pattern recognition” page 7-15, 1980.
[57]. Burt, P. J. “Fast filter transform for image processing” Computer graphics and image processing, 16(1), 20-51, 1981.
[58]. Sternberg, S. R. :Biomedical image processing” Computer, (1), 22-34, 1983.
[59]. Umbaugh, S. E. “Computer vision and image processing: a practical approach using cviptools with cdrom” Prentice Hall PTR, 1997.
[60]. Lehmann, T. M., Gonner, C., & Spitzer, K. “Survey: Interpolation methods in medical image processing. IEEE transactions on medical imaging, 18(11), 1049-1075, 1999.
[61]. Plaza, A., Benediktsson, J. A., Boardman, J. W., Brazile, J., Bruzzone, L., Camps-Valls, G., ... & Marconcini, M. “Recent advances in techniques for hyperspectral image processing” Remote sensing of environment, 113, S110-S122, 2009.
[62]. Viola, P., & Jones, M. “Rapid object detection using a boosted cascade of simple features” In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 (Vol. 1, pp. I-I). IEEE, December 2001.
[63]. Lienhart, R., & Maydt, J. An extended set of Haar-like features for rapid object detection. In Proceedings international conference on image processing (Vol. 1, pp. I-I). IEEE, September 2002.
[64]. Messom, C., & Barczak, A. “ Fast and efficient rotated Haar-like features using rotated integral images” In Australian conference on robotics and automation (pp. 1-6), December 2006
[65]. Haselhoff, A., & Kummert, A. “A vehicle detection system based on Haar and triangle features” In 2009 IEEE intelligent vehicles symposium (pp. 261-266). IEEE, June 2009.
[66]. Angriani, L., Dayat, A. R., & Amin, S. “Implementation method viola jones for detection many faces” International conference on computer systems (ICCS). Makassar, South Sulawesi, Indonesia., 2014.
[67]. AbdelRaouf, A., Higgins, C. A., Pridmore, T., & Khalil, M. I. “Arabic character recognition using a Haar cascade classifier approach (HCC)” Pattern analysis and applications, 19(2), 411-426, 2016.
[68]. Daliman, S., Abu-Bakar, S. A. R., & Azam, M. N. “Development of young oil palm tree recognition using Haar-based rectangular windows” IOP conference series: earth and environmental science (Vol. 37, No. 1, p. 012041), June 2016.
[69]. Meduri, P., & Telles, E. “A Haar-Cascade classifier based Smart Parking System” Proceedings of the international conference on image processing, computer vision, and pattern recognition (IPCV) (pp. 66-70). The Steering Committee of the World Congress in Computer Science, Computer, Engineering and Applied Computing (WorldComp), 2018. Rowley, H. A., Baluja, S., & Kanade, T. “Neural network-based face detection” IEEE transactions on pattern analysis and machine intelligence, 20(1), 23-38, 1998.
[70]. Chitra, S., & Balakrishnan, G. “Comparative study for two color spaces HSCbCr and YCbCr in skin color detection” Applied mathematical sciences, page 4229-4238, 2012. Bhat, V. S., & Pujari, D. J. “Face detection system using HSV color model and morphing operations” Proceedings of national conference on women in science & engineering (NCWSE’13) (pp. 200-204). SDMCET Dharwad, September 2013.Singhraghuvanshi, D., & Agrawal, D. “Human face detection by using skin color segmentation, face features and regions properties” International journal of computer applications (IJCA), 38(9), pp. 14-17, January 2012. Tripathi, S., Sharma, V., & Sharma, S. “Face detection using combined skin color detector and template matching method. International journal of computer applications, 26(7), 5-8, 2011. Amit, Y., Geman, D., & Wilder, K. “Joint induction of shape features and tree classifiers” IEEE transactions on pattern analysis and machine intelligence, 19(11), 1300-1305, 1997. Face-agency. (2020). Retrieved from https://www.face-agency.co.uk/123RF. (2020). [online] Retrieved from https://www.123rf.com
[71]. Dwivedi, D. (2018). Face detection for beginners. Retrieved from https://towardsdatascience.com