iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** II **Month of publication:** February 2023

DOI: https://doi.org/10.22214/ijraset.2023.49252

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue II Feb 2023- Available at www.ijraset.com

Smart Recognition System for High Security Zone

Adrita Banerjee[1], Asmita Bakshi[2], Ayonabha Chandra[3], Arindam Mandal[4], Dr. Ivy Majumdar[5], Dr. Surajit Mandal[6]
Department of Electronics and Communication Engineering, B.P. Poddar Institute of Management and Technology

***Abstract*: In today's world almost all the big institutions face significant security issues; thus, they need a few uniquely prepared staff to satisfy the necessary security.**
These work force, as being human, commit errors which might influence the degree of safety. A proposed answer for the previously mentioned problem could likewise be a Face Recognition Security System, which can distinguish gate crashers to confined or high security regions and help in limiting human mistake.
So far, we have studied the image comparison techniques and tested the same against our dataset of sample images. Our program can almost accurately separate two similar or dissimilar images. We have further tested our sample images against different backgrounds, angles, size etc. and observed the output. We have also done the facial recognition part, i.e., we are able to detect faces and match it with that face from our database and show the result. Finally, we have done iris recognition by comparing static images of iris which gives us an output by showing whether the iris matches or not. Also, in this report we have discussed about our future plan.
Our project will aim to create an application for following and distinguishing faces and iris in recordings and in cameras which can be utilized for multipurpose exercises.
Keywords: OpenCV, Face Encoding, Face Recognition, Iris Recognition, Brute Force Matcher, FLANN Matcher

I. INTRODUCTION

Real time face recognition is a part of biometrics. Biometrics is that the ability for a computer to acknowledge a person through a singular physical trait.
Face recognition provides the potential for the computer to acknowledge a person by facial characteristics. Today, biometrics is one among the fastest growing fields in advanced technology. Predictions indicate a biometrics explosion within subsequent century, to authenticate identities and avoid an unauthorized access to networks, database, and facilities. At the point when a potential intruder enters the secured zone, an array of captured images is taken by the camera and passed on to the software for being examined and matched to a current data set of pre-recognized individuals [1]. The structure, shape and proportions of the faces are compared during the face recognition steps. Face Detection is a task in image investigation which has many more applications, such as Facial Expression Analysis, Human Computer Interface (HCI), Security Systems, Face-Recognitions, Surveillance System, Personal Identity, Verifications, Man-Machine Interface, Content Base Image Retrieval (CBIR) etc. Face Detection is an interesting and challenging problem [2].
Automated iris recognition is yet one more alternative for non-invasive verification and identification of individuals. Iris is also unobtrusive; thus, fast enrolment or authentication is feasible [3]. Interestingly, the spatial patterns that are apparent within the human iris are highly distinctive to a personal [4], [5]. Like the face, the iris is an overt body that is available for remote (i.e., non-invasive) assessment.
Implementation of automated iris recognition can be subdivided into three parts. The first set of issues surrounds image acquisition. The second set cares with localizing the iris intrinsically from a captured image. The third part cares with matching an extracted iris pattern with candidate data base entries.

II. LITERATURE SURVEY

Michel Owayjan et.al. have proposed a solution for the major security issues by devising a Face Recognition Security System which can detect intruders to restricted or high security areas, and help in minimizing human error.
Smita Tripathi et.al., "Face Detection using Combined Skin Color Detector and Template Matching Method" discusses a new face detection method combining the Skin Colour Detector and the Template Matching Method.
Chai, T.-Y., et.al. Vote-based Iris Detection System presents an accurate vote-based method to detect and localize both irises from color images.

III. PROPOSED TECHNIQUE

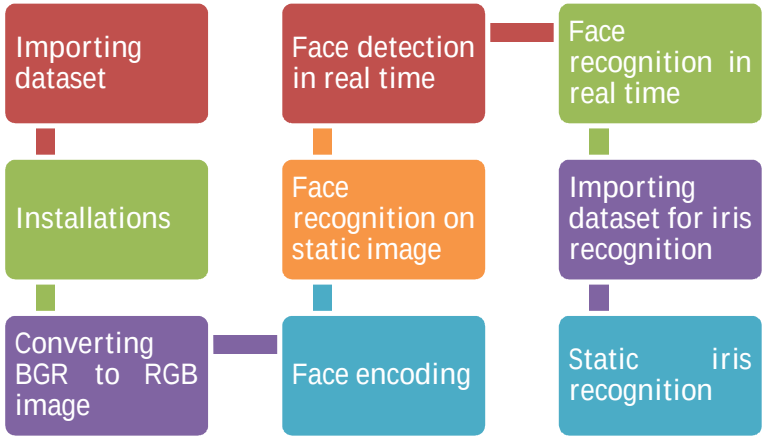


Fig.1: Flowchart of the Proposed Technique

IV. ANALYSIS AND DESIGN

At first, we have taken a partial dataset of about 150 images from github in order to verify our test cases against multitude of samples [6]. We have also added few pictures of our group members to the imported dataset to observe the real time face detection and face recognition.

A. Installations

1) At the beginning we installed the basic libraries.

The first library to install is opencv-python, as always run the command from the terminal.

- *pip install opencv-python* then proceed with face_recognition, this too installs with pip.
- *pip install face_recognition*

2) The first step is always to recall the libraries we have installed OpenCV and face_recognition in our project.

```
import cv2
import face_recognition
```

B. BGR to RGB Image Conversion and Face Encoding

1) With the usual OpenCV procedure, we extract the image, and convert it into RGB color format. Then we do the "face encoding" with the functions of the Face recognition library.

2) Same procedure for the second image and all other images that we have taken for comparison, we only change the name of the variables and used a for loop by giving the range of images we are comparing.

C. Face Recognition on Static Image

1) With a single line, we make a simple face comparison and print the result. If the images are the same, it will print True otherwise False.

2) *Face Detection in Real-Time* We will import cv2 and SimpleFacerec

3) This is a function of the file we have prepared, and it simply takes all the images contained in the images/ folder and encodes them.

With a simple OpenCV function, we take the webcam stream and loop it

```
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
```

4) Now we identify the face passing the frame of the webcam to this function detect_known_faces(frame). It will give us an array with the position at each moment of themovement.
5) Now it will show a rectangle around the face, hence detect the face and also will recognize the face by showing the registered name over the frame.

D. Importing Dataset for Iris Recognition

We have taken an authentic dataset [8] from github consisting of 108 different sources of iris images. Each directory consists of 5-8 samples of a single iris.

E. Static Iris Recognition

1) Iris recognition begins with finding an iris in an image, demarcating its inner and outer boundaries at the pupil and sclera, detecting upper and lower boundaries if they occlude, and detecting and excluding any superimposed eyelashes or reflections from the cornea or eyeglasses. Here, we try to detect the inner boundary of the iris image by finding the contours after doing the required processing.
2) Then, we find the circular region of the iris with the help of hough transform as implemented in opencv.
3) We find the polar the circular region.
4) We do the required processing using the CLAHE algorithm and later converting it to grayscale on the iris and the pupil.
5) Brute Force Matcher is used for matching the features of the first image with another image. It takes one descriptor of first image and matches to all the descriptors of the second image and then it goes to the second descriptor of first image and matches to all the descriptor of the second image and so on.
6) Using a FLANN based matcher we're going to introduce a more complex parameter drawing mechanism, that allows us to draw only the clear matches.

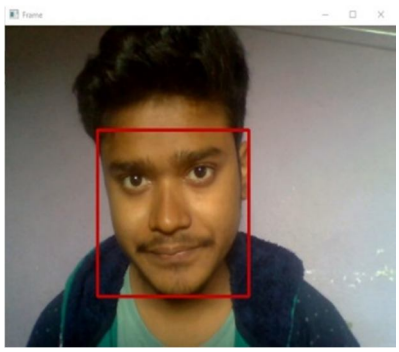
F. Performance Enhancers for Optimize Matcher Performance

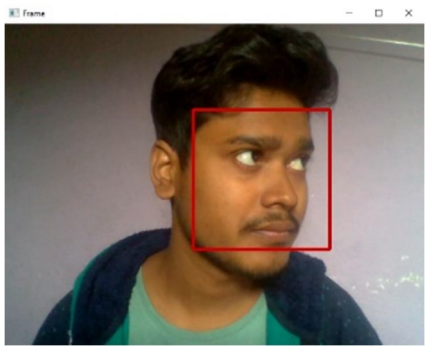
1) ORB (Oriented FAST and Rotated BRIEF) is basically a fusion of FAST keypoint detector and BRIEF descriptor with many modifications to enhance the performance.
2) BRISK (Binary Robust Invariant Scalable Keypoints) is a feature point detection and description algorithm with scale invariance and rotation invariance.
3) SIFT (Scale-Invariant Feature Transform) is a feature detection algorithm in computer vision to detect and describe local features in images.
4) *FAST (Features from Accelerated Segment Test)* is a corner detection method, which could be used to extract feature points and later used to track and map objects in many computers vision tasks.

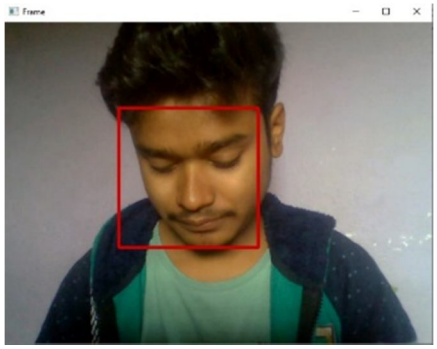
V. RESULT

We have tested a real time face detection as we have modified the mentioned dataset [6] according to our requirements and verified the result.

A. Images with different angle:

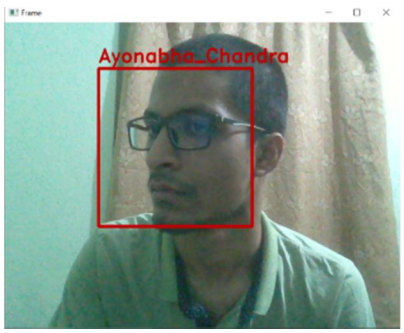

(Front View)

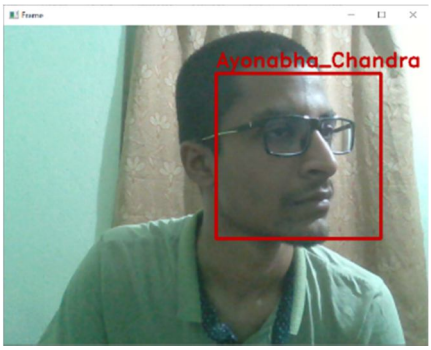

(Side View)


(Bottom View)

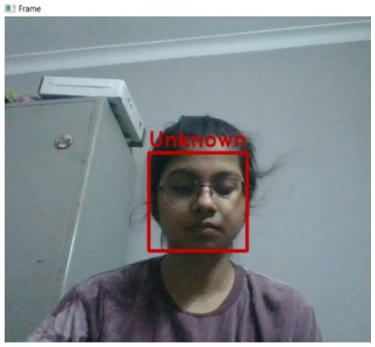
Our code is able to seamlessly recognize faces and correlate them with our database and successfully show us the result. The result of face recognition is shown below:

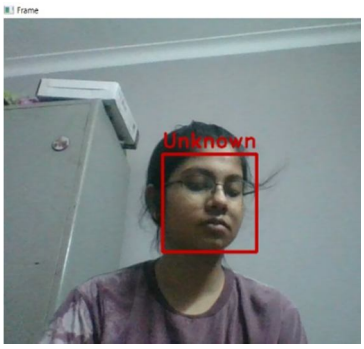
B. For the person Registered in Database

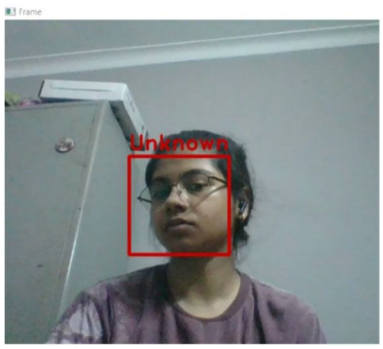




C. For the Person not Registered in Databas:

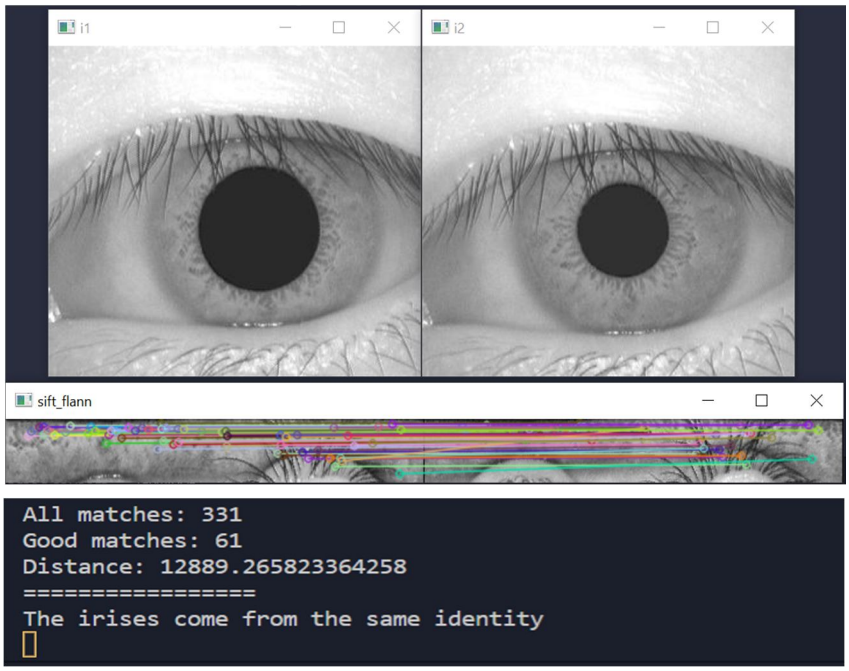




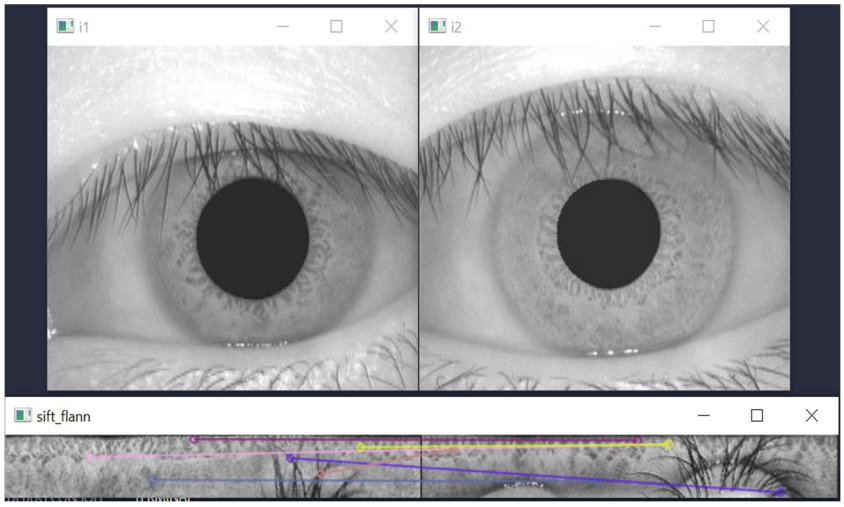


D. Iris Recognition

1) Comparing iris from Same Source



2) *Comparing iris from Different Sources*




```
All matches: 228
Good matches: 7
Distance: 1792.2489929199219
=================
The irises come from different identity
```

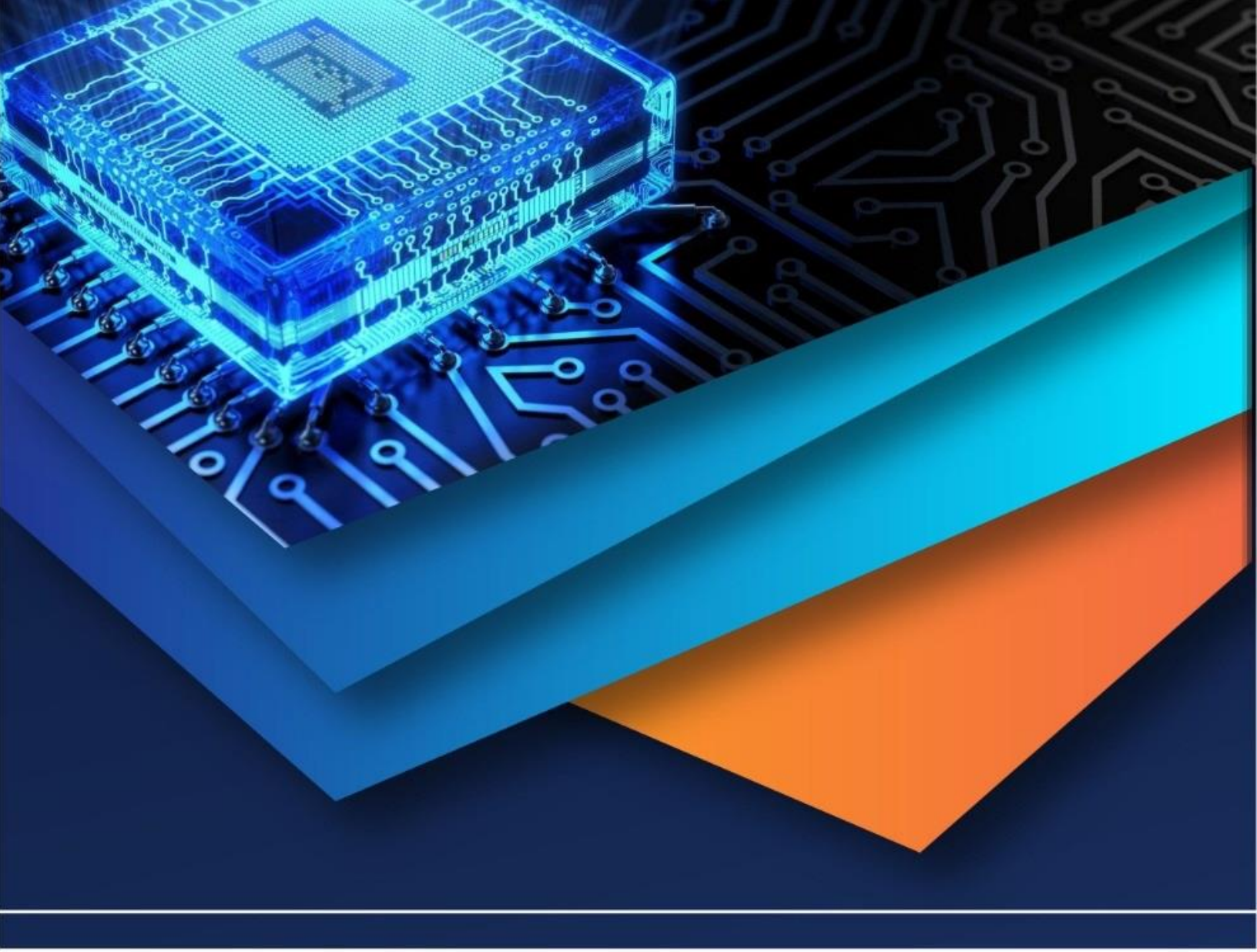
Due to the unavailability of iris scanner or high-resolution webcams, we were able to compare static iris images instead of real time dynamic comparison. Had we been able to avail the iris scanner, it would be possible to verify iris recognition in the real time.

VI. CONCLUSIONS

We have figured implementation of face recognition in real time. Static iris recognition has been implemented using Brute Force Matcher and FLANN Matcher and optimizers like ORB, BRISK, SIFT and FLANN algorithms enhanced the performance of the system. We can find accurate results in the cases of face recognition other than a few cases. The goal of making a robust security system can be improved further. Since we could do the iris recognition from a static approach, we still need to verify the real time iris recognition and integrate it in our system. We need to find a viable approach to synchronize the two functionalities of face recognition and iris recognition together. For that the project will follow through some extra cost. We are confident that given the environment, the prototype idea can be converted into reality.

REFERENCES

[1] Owayjan, Michel & Dergham, Amer & Haber, Gerges & Fakih, Nidal & Hamoush, Ahmad & Abdo, Elie. (2013). Face Recognition Security System.
[2] Smita Tripathi,Varsha Sharma," Face Detection using Combined Skin Color Detector and Template Matching Method". International Journal of Computer Applications, Volume 26– No.7, 2011
[3] Chai, T.-Y., Goi, B.-M., Tay, Y.-H., & Khoo, Y.-H. (2019). Vote-based Iris Detection System. Proceedings of the 2019 3rd International Conference on Digital Signal Processing - ICDSP 2019. doi:10.1145/3316551.3316558
[4] F. H. Adler, Physiology of the Eye. St. Louis, MO: Mosby, 1965.
[5] A. L. Kroeber, Anthropology. New York: Harcourt Brace Jovanovich, 1948
[6] https://drive.google.com/drive/folders/1QRt2us_EueWFilF-uJiwAbS4Jsb6IxAW
[7] https://github.com/serengil/deepface/tree/master/tests/dataset
[8] https://github.com/mvjq/IrisRecognition/tree/master/CASIA1



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY