iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IV **Month of publication:** April 2022

DOI: https://doi.org/10.22214/ijraset.2022.41456

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 10 Issue IV Apr 2022- Available at www.ijraset.com

Face and Eye Detection for Interpreting Malpractices in Examination Hall

Vaibhavi Patil[1], Sakshi Patil[2], Krishna Ganjegi[3], Pallavi Chandratre[4]
[1]*Department of Computer Engineering, Shivajirao S. Jondhale College of Engineering, Dombivli, India*

Abstract: ***One of the most difficult problems in computer vision is detecting faces and eyes. The purpose of this work is to give a review of the available literature on face and eye detection, as well as assessment of gaze. With the growing popularity of systems based on face and eye detection in a range of disciplines in recent years, academia and industry have paid close attention to this topic. Face and eye identification has been the subject of numerous investigations. Face and eye detection systems have made significant process despite numerous challenges such as varying illumination conditions, wearing glasses, having facial hair or moustache on the face, and varying orientation poses or occlusion of the face. We categorize face detection models and look at basic face detection methods in this paper.***

We categorize face detection models and look at basic face detection methos in this paper. Then we'll go through eye detection and estimation techniques.

Keywords: Image Processing, Face Detection, Eye Detection, Gaze Estimation

I. INTRODUCTION

In this world of automation, computers and technology has become man's best friend. Everyone is dependent on technologies and during COVID–19 technology has played a huge role in humans' life. As we know this ongoing pandemic has affected everyone's life. Specially students have been affected a lot during this time. Exams were also conducted in online mode as students were meant to stay in home. In this circumstances mass cheating in online mode increased. Malpractices began to increased and eligible impersonators were detected. During the recent survey 56 eligible impersonators were caught in exam. So online exam has become a threat to students' life. Maintaining secrecy, authenticity, and other factors qualities that are required. To solve this problem, we need an effective method with fewer people. Advances in machine learning and AI technology have made it easier to solve this problem. Colleges started taking classes and tests online, for courses in all fields. We know that maintaining academic discipline in the exams is imperative. One cannot expect the same level of seriousness and focus from a graduate level student and a school student. Each student would have their learning, understanding, and information retaining capabilities. In this situation, malpractice during academic work would be on the rise, be it in the form of plagiarism or cheating during tests. We believe the implementation of an Artificial Intelligence Based Proctoring System is the need of the hour. The quality of one`s online certificate is dependent directly on the quality of the testing process that one undergoes to obtain it. In the same way that exams would be monitored when taken in schools and colleges, they need to be proctored when being conducted online. An AIPS is needed to keep a check on all students, as when giving exams online, there are more ways and opportunities for a student to cheat. The exact ratio of teachers to students that used to be in place for physical exams to monitor them would not be practical in this scenario.

II. PROPOSED SYSTEM

In the proposed system the main motive is to detect the movements of eyes. With the help of eyeball movement, we can test whether the student is looking right or left and up and down. Eye detection is done by using DLIB's facial key point detector. Open CV is used for detecting images for further processing. The next step will be a need of facial key point detector so that it will capture the real movement of eyes. First, step in this processing will be finding eyes before finding the faces and to find eyes we need to detect faces. Blue box will be detected around the face which will simply be the co-ordinates of the face. To detect the face, we will be using DLIB frontal face detector. After, detecting the face, eye movements can be captured. Mouth detection is similar to eye detection. With the help of facial landmark detection mouth is detected and lip movement is captured. DLIB's facial key points are again used and the distance between the lips key points is noted for 100 frames and averaged. If user opens mouth the distances between the lip points increases. If the increase in distance is more than a certain value for at least three outer pairs and two inner pairs then infringement is reported.

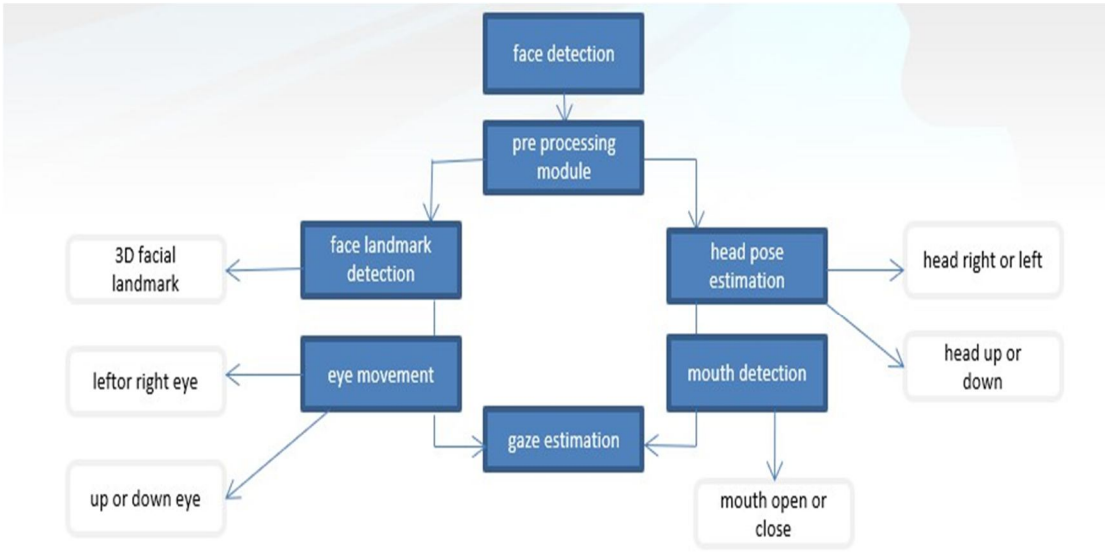


Fig.1 Architecture

III. PROPOSED FEATURES

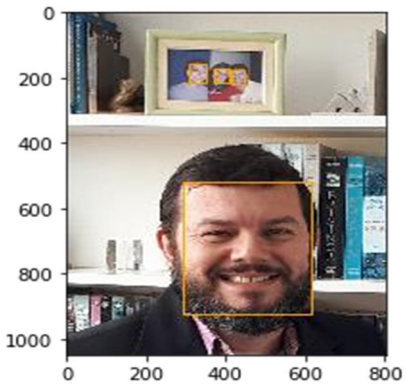
1) *Face Detection:* Face detection will be the first step for detecting impersonators using pythons DLIB facial key point detector. OpenCV will be used for further processing.
2) *Gaze Tracking:* Gaze tracking means tracking the movement of eye which will be done with the help of facial landmark detector. Movement of eyes will be captured of the test taker whether he is trying to look up or down and right or left.
3) *Real-time Head Pose Estimation:* This feature allows the user to create a head pose estimator that can calculate the angle in degrees at which the head is angled. Because of the degrees, the student's head position will change as result of their actions. It is perfect for capturing movements of head like the test taker is moving his head up or down and right or left. The goal is to extract the faces and apply facial landmarks to them so that they can be trained and stored as record files.
4) *Mouth detection* – This is very similar to eye detection. DLIB's facial key points are again used for this task and distance between the lip key points.

IV. IMPLEMENTATION

1) *MTCNN:* MTCNN is a Python library (pip). The most of it is self-explanatory, but it essentially outputs coordinates, or pixel values, for a rectangle where the MTCNN algorithm spotted faces. The above "box" value returns the entire face's location, followed by a "confidence" rating. You'll also have access to other face landmarks, known as "key points," if you want to do more complicated extractions or algorithms. The MTCNN model, in particular, located the eyes, mouth, and nose!

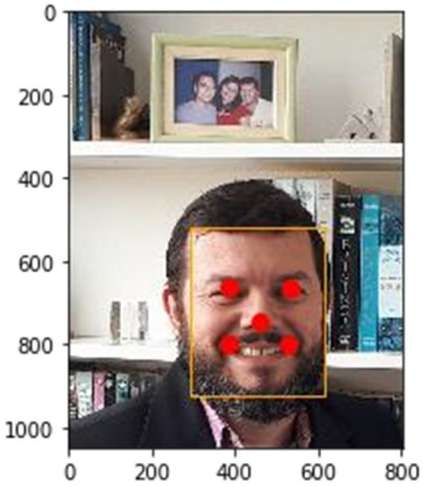
a) Drawing a Box Around Faces

To demonstrate this even better let us draw a box around the face using matplotlib:

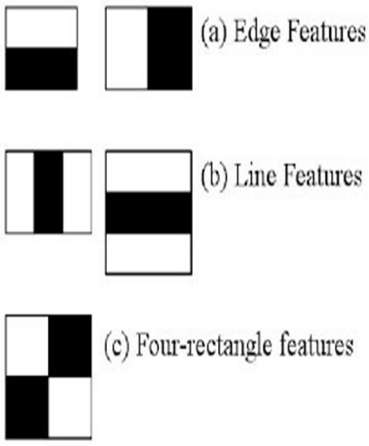


b) Displaying eyes, Mouth and Nose Around Faces

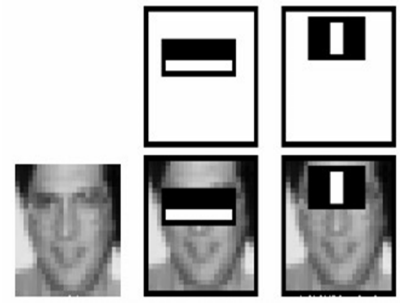
Now let us Take a look at the aforementioned "key points" that the MTCNN model returned.
We will now use these to graph the nose, mouth and eyes as well.



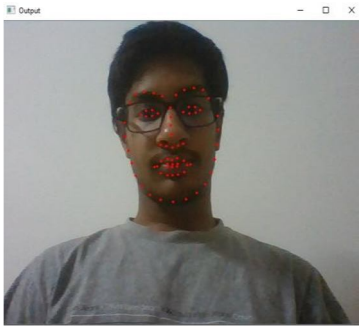
2) HAAR: Paul Viola and Michael Jones provided an excellent approach for object detection using HAAR feature-based cascade classifiers in their 2001 work, "Rapid Object Detection with a Boosted Cascade of Simple Features". It's a machine-learning-based strategy that includes using a large number of positive and negative photographs to train a cascade function. After then, it's utilised to find items in other photos. We'll be working on facial detection here. To train the classifier, the method requires a large number of positive images (images of faces) and negative images (images without faces). Then we have to extract features from it.. HAAR characteristics, as illustrated in the graphic below, are employed for this. They're really similar to our convolutional kernel**.** Each feature is a single value calculated by subtracting the sum of pixels beneath the white rectangle from the total of pixels beneath the black rectangle**.**



To calculate a large number of features, all feasible sizes and positions of each kernel are now used. We must find the total of the pixels under the white and black rectangles for each feature computation. They used integral pictures to tackle the problem. It reduces the sum of the pixels, regardless of how many there are, to a four-pixel operation. However, the majority of traits we calculated are irrelevant. Take a look at the illustration below. Two good aspects are shown in the first row. The first attribute chosen appears to be the fact that the area around the eyes is frequently darker than the area around the nose and cheekbones. The second feature chosen is based on the fact that the eyes are darker than the nasal bridge. However, the application of the same windows to the cheeks or any other location is irrelevant. So, how do we pick the best features from a list of 160000+ options?



3) *DLIB:* Before we can move on to image processing, we must first find eyes, and in order to find eyes, we must first find a face. The DLIB module's facial key point detector accepts as input a rectangle object that is merely the coordinates of a face. We may utilise DLIB's integrated frontal face detector to find faces. For this challenge, you can use any classifier. If accuracy is important to you and time is not a concern, I recommend using a CNN because it will provide much better results, especially for non-frontal facing faces and partially occluded faces.



4) *OpenCV:* OpenCV (Open-Source Computer Vision) is a programming library geared mostly at real-time computer vision. Willow Garage later funded the project, which was originally developed by Intel. The library is open-source and cross-platform.

a) Image/video I/O, processing, display
b) Object/feature detection
c) Geometry-based monocular or stereo computer vision,
d) Computational photography
e) Machine learning & clustering
f) CUDA acceleration

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IV Apr 2022- Available at www.ijraset.com

V. SOFTWARE REQUIREMENTS

1) *OpenCV* - It is a tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection.
2) *DLIB* - It is a tool used for landmark's facial detector.
3) *Python* - It plays a vital role in AI coding language by providing it with good frameworks.
4) *Tensor flow* – It is an open-source AI library which allows developer to create large scale neural networks with many layers.

VI. CONCLUSIONS

Detecting malpractice using AI system is developed and employed for detecting impersonators in exam hall .Candidates will be examined in online mode where their face will be detected and eye movements will be captured .While they are appearing for their exams their movements of head will also be detected i.e. if the student is trying to look up or down and right or left .Further movement of lips will be captured which will be very useful to detect the malpractice. This process will be useful for examining students for the invigilators.

VII. ACKNOWLEDGEMENT

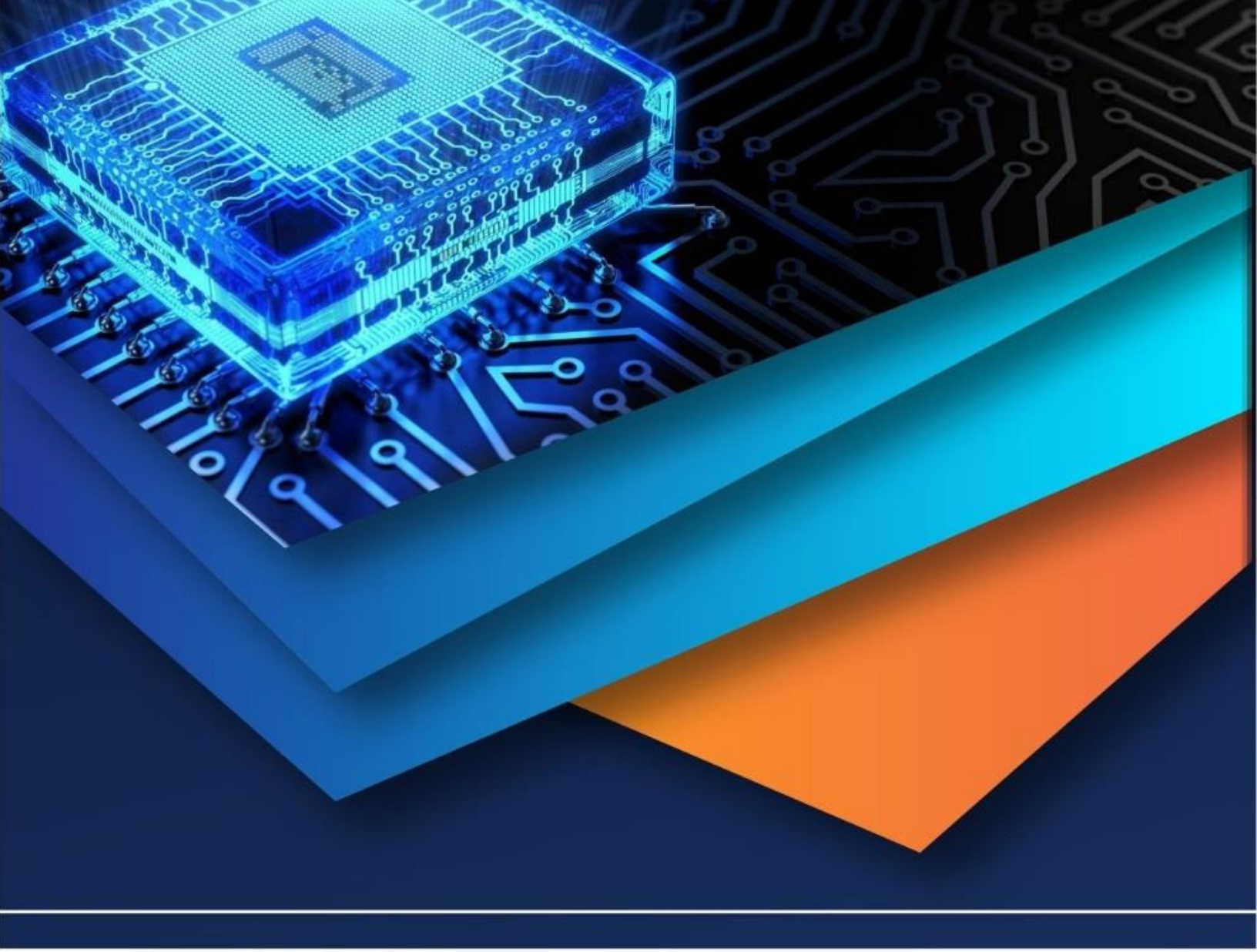
We sincerely wish to thank the project guide Prof. Pallavi Chandratre for her encouraging and inspiring guidance that helped us to make our project a success. Our project guide made sure we were on track at all times with her expert guidance, kind advice, and timely motivation which helped us determine our project. We would like to thank our project coordinator Prof. Reena Deshmukh for all the support she provided concerning our project.

We also express our deepest thanks to our HOD Dr. Uttara Gogate whose benevolence helped us by making the computer facilities available to us for our project in our laboratory and making it a true success. Without his kind and keen co-operation, our project would have been stifled to a standstill.

Lastly, we would like to thank our college principal Dr. Pramod Rodge for providing lab facilities and permitting us to go on with our project. We would also like to thank our colleagues who helped us directly or indirectly during our project.

REFERENCES

[1] Viola and M. J. Jones, "Robust real-time face detection," Int. J. Comput. Vision, vol. 57, no. 2, pp. 137–154, May 2004.
[2] An automated fraud detection of hall tickets in an offline examination system using ANN classifier (Shridevi Soma and Vidyashree S.A. Article: An Automated Fraud Detection of Hall Ticket in an Offline Examination System using ANN Classifier. International Journal of Computer Applications 126(8):7-12, September 2015. Published by Foundation of Computer Science)
[3] A Fraud detection based Online Test and Behaviour Identification Implementing Visualization Techniques (Harish Babu, Kalidasu, B. Prasanna Kumar, Haripriya Priyadarshini.
[4] X. Xu, S. Shah, and I. A. Kakadiaris. Face alignment via an ensembleof random ferns. In Proc. IEEE International Conference on Identity, Security and Behaviour Analysis, Sendai, Japan, Feb. 29 - Mar. 2 2016.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY