

# Detecting Impersonators in Examination Hall Using AI

Raj Shekhar Pandey  
*Computer Science & Engineering*  
MIET, MEERUT (of AKTU.)

Sanand Deep Singh  
*Computer Science & Engineering*  
MIET, MEERUT (of AKTU.)

Mr. Varun Kaushik  
*Computer Science & Engineering*  
MIET, MEERUT

**ABSTRACT\_** This system is made to make identity verification better and safer by finding and recognizing faces correctly. Institutions and security departments can use it to identify people and stop others from getting in without permission. The problem of identities and manual checks can be solved in a simple way without using a lot of computer power. New developments in Artificial Intelligence and Machine Learning have made face detection and recognition systems better and more reliable.

In this project we are making a face detection system that uses Artificial Intelligence. We are using OpenCV, Flask, face recognition, dlib and Python to build it. The system learns from pictures of authorized users faces that we store. It takes video from a camera finds faces in real time and checks them against the stored pictures. When it finds a match it shows the users identity on the screen. If someone unknown or unauthorized shows up the system tells the administrator so they can take action.

This solution makes things easier for people improves accuracy saves time and provides a security system. Administrators can check records look at activities and track unknown people when the system finds unusual faces. The face detection system is a way to keep people and places safe. It helps identity verification processes by detecting and recognizing faces accurately which is what face detection systems, like this one are designed to do.

## I. INTRODUCTION

The examination system is very important. We need to make sure it is safe and secure. Any problem that affects how the system works can be a threat. In schools and universities examinations are a part of how we measure what students know and how well they do. So we need to understand how the system

works and where the problems might be so we can stop them from happening.

One big problem with examinations is when people work together in a way that's not fair or when someone pretends to be someone else. Some people try to cheat by getting help during the exam so they can do better than others. Sometimes a person might ask someone to take the exam for them or help them during the test. This is not fair. It hurts the integrity of the exam.

Now we use things like special machines that check people faces and identities before the exam starts. We also have people watching to make sure everything is okay. These methods are not always enough during the exam. If the people watching are not paying attention or if someone is not honest then people might be able to cheat or pretend to be someone. So we need to find a way to check people identities all the time during the exam.

To solve this problem we made a system that uses artificial intelligence to detect faces. This system uses tools like OpenCV, Flask and Python. It takes a picture of the persons face during the exam. Checks it with the picture we have on file. It can also check the persons face again and again during the exam to make sure it is really them. If it finds a problem it will mark it so we can look at it later.

We keep all the records in a place so the people, in charge of the exam can look at them. They can then check again to make sure everything is okay. This system helps stop people from pretending to be someone it helps us watch the exams more closely it saves time and it makes sure the exams are fair and secure. The examination system is very important. This new system helps make it even better.

## LITERATURE SURVEY

Face detection is a part of computer vision. It helps find faces in pictures.

The goal is to see if a face is there how big it is and where it is in the image.

This helps get rid of things in the background like a person body, a chair or a car. Face detection is like finding an object in a picture. The object is a face.

Face detection is not the same as face localization.

Face detection is simply finding out if there is a face in a picture. Face localization is finding the spot and size of the face.

Some systems try to find one face while others try to find faces. This depends on what the system's used for. There are two ways to detect faces:

\*One way is to look for parts of the face like the eyes, nose and mouth.

\* The other way is to compare the picture to other pictures.

Artificial Intelligence and Machine Learning have made these methods better.

The AI Face Detection System uses tools like OpenCV, dlib and Python.

These tools help the system find faces accurately and, in time.

The system also uses Flask.

It provides face detection.

The proposed system gives results.

## BASE FEATURE APPROACH

The face detection and recognition systems use a method that looks at parts of the face. This method focuses on finding the eyes, nose, lips, eyebrows and jawline. These parts of the face help the system understand what a person face looks like. By looking at these parts the system can. Recognize faces in pictures or videos.

In the AI Face Detection System that we're talking about we use special tools to find the important parts of the face and get information from them. These tools help us find things, like the corners of the eyes the position of the nose the edges of the mouth and the edges of the face. We then compare this information with pictures of faces that we already have stored. This helps us make sure we are recognizing the person. It also makes the system work better.

We use OpenCV to take pictures look at videos find faces and get pictures ready to use. It helps us find faces in time from a webcam or camera. OpenCV also helps us make pictures look better by resizing them turning them into white and reducing noise. This makes it easier to recognize faces.

The face recognition library helps us make a set of numbers for

each face. We store these numbers in a database. When someone appears in front of the camera we take a picture of their face. Compare it to the numbers we already have. If we find a match we know who the person is. If not we do not know who they are.

We use Flask to make a website that people can use to interact with the system. It helps us manage user accounts, store pictures of faces and show results. Flask also gives us controls to manage the system. It makes it easy for people to use the system and see how well it is working.

We use Python to write all the code for the project. It helps us use all the tools and parts of the system together. Python is a language to use because it is easy to learn and it works well with artificial intelligence and computer vision.

By using special face recognition tools, Flask and Python the system can detect and recognize faces quickly and accurately. This method is good because it saves time and makes the system work better. It is a way to recognize faces and it makes it easier to verify people identities.

## PROPOSED SYSTEM

The proposed AI Face Detection System is designed to detect and recognize faces automatically in real time. It is developed using OpenCV, Python, Flask, dlib, and the face recognition library. The system collects face images of registered users and creates a dataset for training. Unique facial features are extracted and stored for identification. During execution, the camera captures live video and compares detected faces with stored data. It can be used in attendance systems, offices, educational institutions, and security applications with high accuracy.

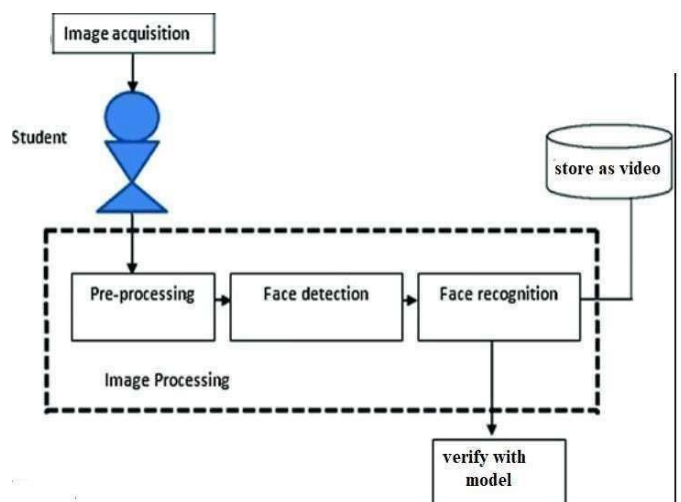


Fig 1:Architecture

## IMPLEMENTATION

### Image Acquisition

The first step in our system is to get images. We use a camera or sensor to take these images. Then we use them for other things we need to do. These images are like the material that our system uses. If we do not take these images we cannot do things like detect. Recognize things. The quality of the image we take is very important, for how our system works. When the images are clear it helps us find faces in the images accurately and it also helps us get better results when we try to recognize things. So we need to get images in a way for our system to work well and do things properly.

### Pre-processing

The first thing we do to improve image quality is pre-processing. This step is really important because it makes the image clearer and better before we even start looking at it. We get rid of the noise adjust the brightness and make the contrast better. This helps us deal with images that were taken in lighting. The main reason we do pre-processing is to get the image ready for finding and recognizing faces. When the image is quality we get more reliable results and fewer mistakes. This step makes the whole AI Face Detection System work better accurately and more efficiently.

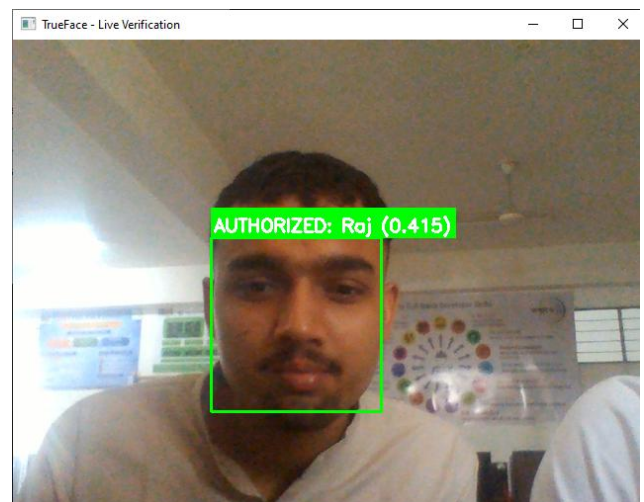
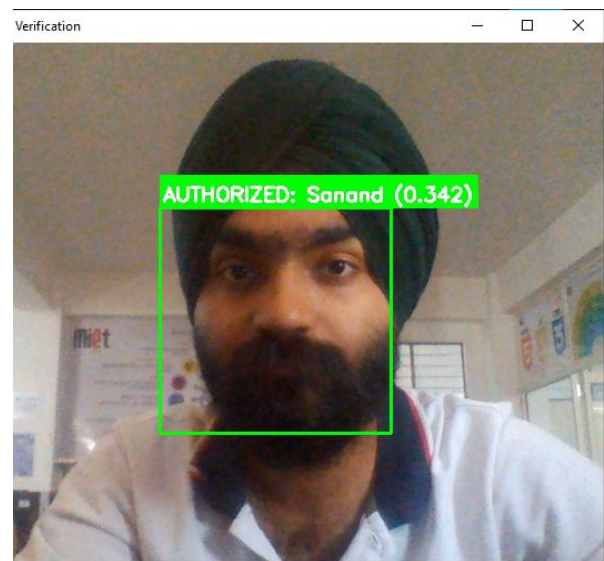
### Face Detection

We use face detection to find out if there is a face in a picture or a live video. The system we are using relies on OpenCV to find faces quickly and get it in real time. Once we find a face we look for parts like the eyes, nose, lips and jawline. Finding these parts helps us detect faces better and do things with the image. Face detection is a deal because we cannot recognize a face until we have found one.

### Face Recognition

Face recognition is when we figure out who someone is by looking at their features and comparing them to what we already know. In our system we turn the face we found into numbers using a face recognition library. We then compare these numbers to the people we already have in our system. If we find a match we know who the person is and we show it on the screen. If we do not find a match we say the face is unknown. This part of the system helps us identify people quickly safely and accurately which is really useful for things, like attendance, security and monitoring.

## 1. RESULTS AND DISCUSSION



## CONCLUSION

After research and implementation the proposed AI Face Detection system was successfully developed.

The proposed AI Face Detection system was developed using technologies such as OpenCV, dlib face recognition, Flask and Python.

It was designed to provide an automated solution for AI Face Detection. Recognition with improved speed, accuracy and reliability.

The results obtained during testing confirmed that the selected technologies and methods were suitable for achieving the objectives of the AI Face Detection project.

The proposed AI Face Detection system was able to detect faces accurately from live video streams and images.

It also successfully recognized registered users by comparing time facial encodings with stored datasets.

The integration of dlib landmarks and the face recognition library improved the precision of matching.

Compared to methods the proposed AI Face Detection system provides better real-time performance and higher recognition accuracy.

Although the AI Face Detection system performed effectively certain limitations were observed under challenging conditions.

These conditions include lighting, face rotation, partial occlusion and varying camera angles.

Recognition accuracy may decrease when the input image quality is low or when the face is not clearly visible.

These challenges can be reduced in work by adding advanced preprocessing techniques.

The developed AI Face Detection system can be widely used in attendance management, office security, surveillance systems, access control and identity verification applications.

It reduces effort saves time and increases operational security.

For large-scale environments, the AI Face Detection system can assist administrators and security personnel in identifying individuals efficiently.

In conclusion the proposed AI Face Detection System demonstrates that modern computer vision and artificial intelligence technologies can provide reliable biometric solutions.

With improvements the AI Face Detection system can become even more accurate, secure and adaptable, for real-world applications.

## REFERENCES

- [1] J. Deng et al., "RetinaFace: Single-shot multi-level face localisation in the wild," in Proc. CVPR, 2020.
- [2] K. Zhang et al., "Joint face detection and alignment using multitask cascaded convolutional networks," IEEE Signal Process. Lett., vol. 23, no. 10, pp. 1499–1503, 2016.
- [3] C. A. Trianti et al., "Integration of Flask and Python on the face recognition based attendance system," in Proc. ICITech, 2021.
- [4] P. Kaviya et al., "A web-based face recognition system using Flask and OpenCV for secure authentication," Int. J. Sci. Eng. Technol., 2026.
- [5] D. E. King, "High quality face recognition with deep metric learning," dlib C++ Library, Feb. 2017.

## AUTHOR PROFILES

**Mr. Sanand Deep Singh** is pursuing his B.Tech in Computer Science and Engineering from MIET Engineering College. He has a strong interest in the field of computing and technology. His areas of interest include C language, Data Science, and Python, Cyber Security and he is enthusiastic about exploring modern technologies and applying programming skills to solve real-world problems..

**Mr. Raj Shekhar Pandey** is currently studying B.Tech in Computer Science and Engineering from MIET Engineering college. He is passionate about the field of computer science and enjoys learning about emerging technologies. His key interests include java programming, Data Science, and Python, and he is keen on developing skills that help in solving practical and real-world problems through technology.