

**ANALYSIS OF ALGORITHM**

**PROJECT ON**

**DETECTING IMPERSONATORS IN EXAMINATION HALL USING AI**

**COURSE NAME**-ANALYSIS OF ALGORITHM

**PROFESSOR NAME**-HALEH KHOJASTEH

**SUBMITTED BY**

SRI VAISHNAVI KUMBAR

SUVARNA KABIRAJ SAHU

SAI NARENDRA MUGATHA

**ABSTRACT**

* In order to detect the impersonators in examination halls, it is important to manage a better way of examination handling system which can help in reducing malpractices happening in examination centers.
* According to various news reports day-by-day there are cases in which the potential impersonators are detected by national testing agencies.
* To solve this problem, an efficient methodology is required which is automatic and does not require manual efforts.
* With the advancement of machine learning and AI technology, it is easy to solve this problem.
* In this project we are trying to build an AI System in which images of students can be saved which can be mapped to a different model through transfer learning process to get proper results.
* After detection if the student is validated to enter the examination hall, then the system shows the hall ticket number and the name of the student, otherwise it appears as unknown

**OBJECTIVES**

* To build a quick and precise student verification process with minimal effort.
* To reduce security breaches.
* To reduce impersonators issue with live verification.
* To take less time for prediction and processing as the prediction is done automatic using trained model.

**INTRODUCTION**

* Examinations play a vital role in the process of learning, no matter how much we hate exams or get annoyed with it, exams are important.
* Exams build confidence among students which helps them to work hard It helps students in various factors like discipline, time management, work management, brain exercise and expand their knowledge base and research skills.
* The purpose of our project is to find a way to ease the examination process and to bring more advancement and security in this process.
* We are trying to build an AI based detector to detect the impersonators who are trying to impersonate the other person in examination halls.
* As there are various manual testing systems already in place, in which manually the student’s name is mapped to its examination admit card or connect card, but it’s a manual and time taking process.
* Also, there can be a scenario, where the student can change the picture on the examination hall ticket to get an admit in the examination center.
* In our proposed system initially images of each student are collected, and each dataset consists of 5-10 images of each student. These images are trained using KD tree algorithm using image processing technique and model is saved in system.
* This model can be used for automatic prediction of student in exam halls from live video or images.
* A threat is any factor that poses any form of harm to the proper functioning and satisfying the objective of the system in a secure way ensuring confidentiality, authenticity, and other necessary features.
* An Examination is a critical asset in the fast-growing learning environment. In order to assess the threats, we should understand the nature of the system and must analyze the environment where the system is deployed.
* Collusion is identified as the highest rated threat in an examination. Intent of most threats backtracks to the motive of cheating by the candidates for obtaining assistance during examination to enhance their chances in the examination amongst the competitors.
* Collusion is when the candidate invites any third party for impersonation or for aid/help the candidate in the examination. Currently authentication of the user before taking up the examination is carried out using biometric authentication and manual verification of credentials by the invigilator. This process ensures authentication at entry level.
* Due to lack of integrity of the invigilator may increase the chance of collusion in an examination.
* The only way to address this issue is to conduct a runtime dynamic authentication during the examination. This paper proposes an approach to overcome this disadvantage.
* Candidate registration photograph is used for comparison with the picture recorded while the candidate enters the examination hall.
* Then this recently recorded image is compared with the images captured during random intervals for identification of violations.
* Violation here refers to mismatch of features above a certain degree which drives to the decision that the person currently taking up the examination is not the corresponding candidate. This entire process is secured and conducted in those corresponding systems to avoid any intrusion and modification of crucial data.
* Violations identified by the algorithm is logged in a centralized repository under control of the examination authority. Manual verification by the invigilator is triggered to confirm the violation.

**LITERATURE SURVEY**

* Face detection is a computer technology that determines the location and size of human face in arbitrary (digital) image.
* The facial features are detected and any other objects like trees, buildings, and bodies etc are ignored from the digital image. It can be regarded as a specific case of object-class detection, where the task is finding the location and sizes of all objects in an image that belong to a given class.
* Face detection can be regarded as a more general case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one). Basically, there are two types of approaches to detect facial part in the given image i.e., feature base and image base approach.
* Feature base approach tries to extract features of the image and match it against the knowledge of the face features. While image base approach tries to get best match between training and testing images.

**SOFTWARE REQUIREMENT SPECIFICATION**

* SOFTWARE HARDWARE REQUIREMENT: HARDWARE REQUIREMENTS: System: Pentium IV 2.4 GHz. Hard Disk: 100 GB. Monitor: 15 VGA Color. Mouse: Logitech. RAM: 1 GB.
* SOFTWARE REQUIREMENTS: Operating system: Windows XP/7/10 Coding Language: python Development Kit anaconda Library: keras, OpenCV Dataset: any student’s dataset

**LIBRARIES USED IN PYTHON**

1. **Requests** - The most famous http library written by Kenneth Reitz. It’s a must have for every python developer.
2. **Dlib** is used to estimate the location of coordinates that map the facial points on a person’s face.
3. **Imutils** is a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges.
4. **NumPy** is used to perform a wide variety of mathematical operations on arrays.
5. **OpenCV** is a Python library that allows you to perform image processing and computer vision tasks
6. The **sklearn** library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction.
7. **Facial recognition** is a way of identifying or confirming an individual's identity using their face. Facial recognition systems can be used to identify people in photos, videos, or in real-time.

**SYSTEM DESIGN**

**UML DIAGRAMS**

**CLASS DIAGRAM –**

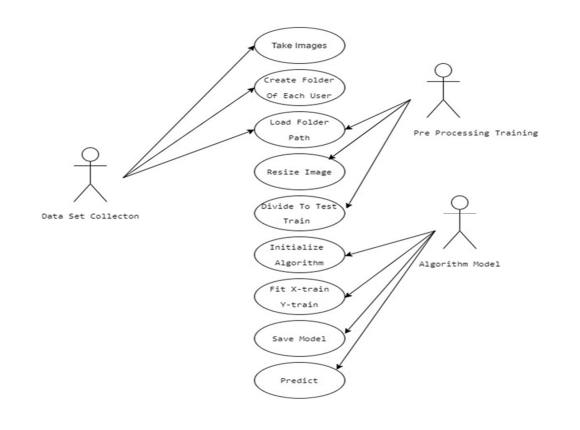
* Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Diagram, schematic

Description automatically generated

**USECASE DIAGRAM –**

* Use case diagrams are considered for high level requirement analysis of a system. So, when the requirements of a system are analyzed, the functionalities are captured in use cases. So, we can say that uses cases are nothing, but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.



**SEQUENCE DIAGRAM –**

* A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows, as parallel vertical lines ("lifelines"), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

Diagram

Description automatically generated

**ACTIVITY DIAGRAM**

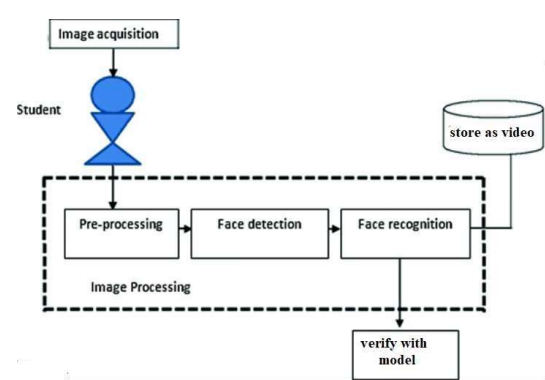
* Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Diagram

Description automatically generated

**ARCHITECTURE MODEL**

* Step 1: The student’s image will be captured
* Step 2: Processing of the image captured
* Step 3: Face Detection as it is unique for all
* Step 4: Face Recognition based on view or feature
* Step 5: Video recording while the detection takes place which will be mapped to the data set stored



**Image Acquisition**: Digital image Processing. In image processing, it is defined as the action of retrieving an image from some source, usually a hardware-based source for processing. It is the first step in the workflow sequence because, without an image, no processing is possible.

**Pre-processing**: Pre-processing is a common name for operations with images at the lowest level of abstraction — both input and output are intensity images. These iconic images are of the same kind as the original data captured by the sensor, with an intensity image usually represented by a matrix of image function values (brightness’s). The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing, although geometric transformations of images (e.g., rotation, scaling, translation) are classified among pre-processing methods here since similar techniques are used.

**Face Detection**: The face is one of the easiest ways to distinguish the individual identity of each other. Face recognition is a personal identification system that uses personal characteristics of a person to identify the person's identity. Human face recognition procedure basically consists of two phases, namely face detection, where this process takes place very rapidly in humans, except under conditions where the object is located at a short distance away, the next is the introduction, which recognize a face as individuals. Stage is then replicated and developed as a model for facial image recognition (face recognition) is one of the much-studied biometrics technology and developed by experts. There are two kinds of methods that are currently popular in developed face recognition pattern namely, Eigenface method and Fisherface method. Facial image recognition Eigenface method is based on the reduction of facedimensional space using Principal Component Analysis (PCA) for facial features. The main purpose of the use of PCA on face recognition using Eigen faces was formed (face space) by finding the eigenvector corresponding to the largest eigenvalue of the face image. The area of this project face detection system with face recognition is Image processing. The software requirements for this project is MATLAB software. Keywords: face detection, Eigen face, PCA, matlab Extension: There are vast number of applications from this face detection project, this project can be extended that the various parts in the face can be detect which are in various directions and shapes.

**Face Recognition**: There are two predominant approaches to the face recognition problem: Geometric (feature based) and photometric (view based). As researcher interest in face recognition continued, many different algorithms were developed, three of which have been well studied in face recognition literature. Recognition algorithms can be divided into two main approaches, geometric and photometric stereo

**CODE**

**  **

**STEPS TO RUN THE PROGRAM**

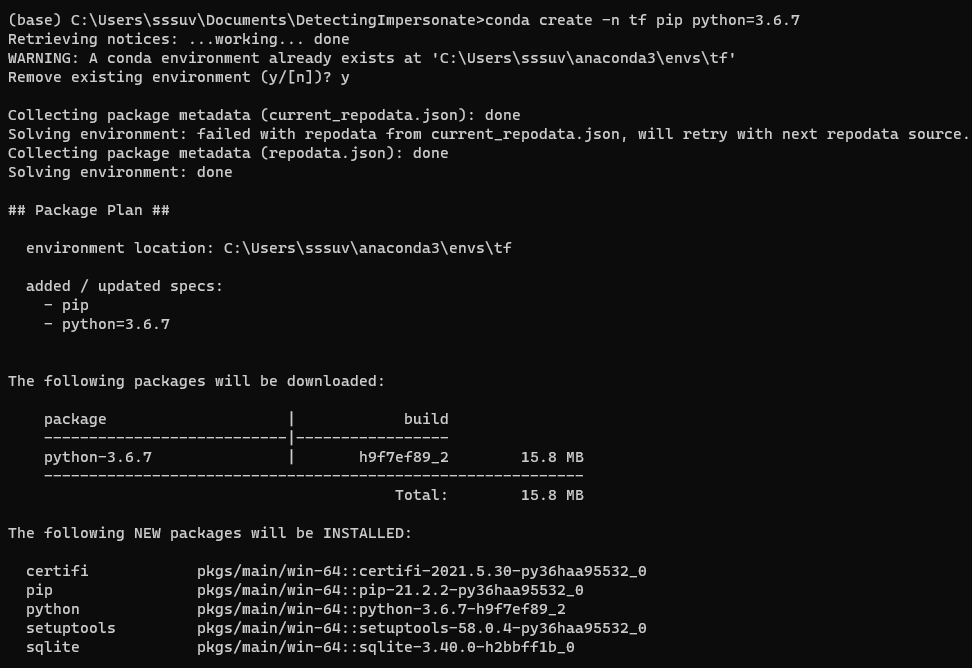
**Step-1**-Open Anaconda Command Prompt and move to the folder you have stored the python files.

**Graphical user interface, text, application

Description automatically generated**

**Step-2** **conda create -n tf**, this command will be used to create a new virtual environment with name TensorFlow, and **pip python=3.6.7** will help us in downloading the module which lends us a way to reliably generate cryptographically strong pseudo-random values. Using these, we can manage secrets like account authentication, tokens, and so.

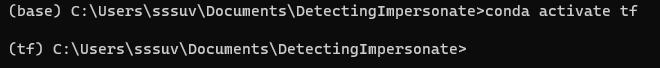
It will state the packages it will install and ask if we must proceed, press Y

****

**Text

Description automatically generated**

**Step-3** Now we will activate conda tf, with conda, you can create, export, list, remove, and update environments that have different versions of Python and/or packages installed in them. Switching or moving between environments is called activating the environment.



**Step-4** Now we are in tf (tensorflow) environment, we need to install the libraries and packages of python namely imutils, NumPy, open cv2, cmake scikit-learn and face-recognition.

**Imutils** A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edge**s.**

**NumPy** is usedto perform a wide variety of mathematical operations on arrays.

**OpenCV** is a Python library that allows you to perform image processing and computer vision tasks

**CMake** is an open-source, cross-platform family of tools designed to build, test and package software.

The **sklearn library** contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction.

**Facial recognition** is a way of identifying or confirming an individual's identity using their face. Facial recognition systems can be used to identify people in photos, videos, or in real-time.

Text

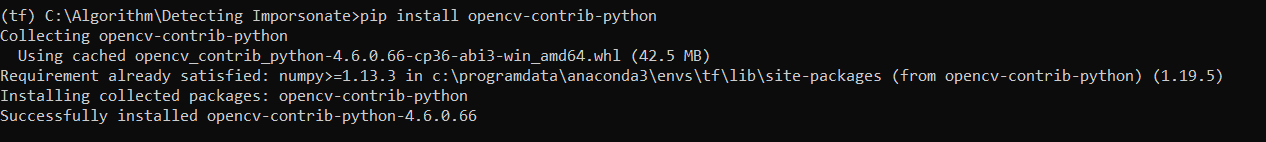
Description automatically generated

A computer screen capture

Description automatically generated with medium confidence

A picture containing text

Description automatically generated



Text

Description automatically generated with medium confidence

Text

Description automatically generated

A computer screen capture

Description automatically generated with low confidence

**Step-5**- As all the packages and libraries are installed, now we will be training the model we are using, the hog model.

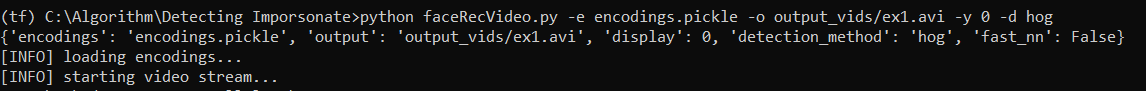
This is based on the **HOG** (Histogram of Oriented Gradients) feature descriptor with a **linear SVM** machine learning algorithm to perform face detection.

HOG is a simple and powerful feature descriptor. It is not only used for face detection but also it is widely used for object detection like cars, pets, and fruits. HOG is robust for object detection because object shape is characterized using the local intensity gradient distribution and edge direction.

Text

Description automatically generated

**Step-6**- Now we will start the video stream.



A picture containing text, person, green

Description automatically generated Graphical user interface, application

Description automatically generated A person wearing glasses

Description automatically generated with medium confidenceA person with a mustache

Description automatically generated with low confidence

A picture containing person, indoor, looking, colorful

Description automatically generated

**SYSTEM IMPLEMENTATION**

**DATA SET COLLECTION**:  
Students data set can be collected manually while issusing halltickets and store in each folder. Minimum 5-10 images need to be collected room each student.  
  
**ENCODING DATASET**:  
In this stage face recognition, pickle, OpenCV, sklearn libraries are initialized and each image is looped from the folder using batching and multiprocessing.

•load the input image and convert it from BGR

•detect the (x, y)-coordinates of the bounding boxes

•corresponding to each face in the input image compute the facial embedding for the face

**K-D TREE ALGORITHM:**Encoded images are given to kd tree algorithm and model is saved in pickle format.K-d tree data structure has been used as a data structure for overcoming increased processing times caused by the addition of features into the database. An alternative approach is establishing a balanced k-d tree

**PREDICTION:**In this stage input values are taken from image or video and compared with model and results are predicted and output is stored in image or video.

**SYSTEM TESTING**

**TESTING CODE**

* Code is usually developed in a file using an editor.
* To test the code, import it into a Python session and try to run it.
* Usually there is an error, so you go back to the file, make a correction, and test again.
* This process is repeated until you are satisfied that the code works.
* The entire process is known as the development cycle.
* There are two types of errors that you will encounter. Syntax errors occur when the form of some command is invalid.
* This happens when you make typing errors such as misspellings, or call something by the wrong name, and for many other reasons. Python will always give an error message for a syntax error.

**FUTURE WORKS ENHANCMENT**

* To make this more efficient, we will need more data.
* In future we will build a complete open source working platform with huge amounts of image data.
* In future, we would be able to detect all the students present in the classroom at once.
* We can make this system as attendance face recognition system where we will use a webcam to record the attendance live in an excel sheet.

**REFERENCES**

* T.-H. Kim, D.-C. Park, D.-M. Woo, T. Jeong, and S.-Y. Min, “Multi-class classifierbased adaboost algorithm,” in Proceedings of the Second Sinoforeign-interchange Conference on Intelligent Science and Intelligent Data Engineering, ser. IScIDE’11. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 122– 127.
* P. Viola and M. J. Jones, “Robust real-time face detection,” Int. J. Comput. Vision, vol. 57, no. 2, pp. 137–154, May 2004.
* P. Viola and M. Jones, “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, Dec 2001, pp. I–I.
* J. Li, J. Zhao, Y. Wei, C. Lang, Y. Li, and J. Feng, “Towards real world human parsing: Multiple-human parsing in the wild,” CoRR, vol. abs/1705.07206.
* J. Li, J. Zhao, Y. Wei, C. Lang, Y. Li, and J. Feng, “Towards real world human parsing: Multiple-human parsing in the wild,” CoRR, vol. abs/1705.07206.
* https://aws.amazon.com/what-is/facial-recognition/#:~:text=It%20is%20faster%20and%20more,authentication%20for%20additional%20security%20verification