A Project Report on

ONLINE PROCTORING SYSTEM-AI

Submitted by

B.Jayasree Lakshmi -R170542
B.Drakshayani -R170555
N.Bhumika -R170836

Submitted to

IIIT RK Valley Idupulapaya,Vempalli,
YSR kadapa Andhra Pradesh , India PIN 516330

Rajiv Gandhi University of Knowledge Technologies
RGUKT
R.K.Valley

Under the guidance of

N.Satyanandaram
Dept of Computer Science and Engineering

as a part of Partial fulfillment of the degree of
Bachelor of Technology in Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled “**Online Proctoring System-AI**” submitted by **B.JayasreeLakshmi [R170542], B.Drakshayani [R170555] N.Bhumika [R170836]** in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering isa bonafide work carried out by them under my supervision and guidance.

The report has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

N.Satyanandaram
Project Internal Guide,
Computer Science and Engineering,
RK Valley, RGUKT

N.Satyanandaram
Head of the Department,
Computer Science and Engineering,
RK Valley ,RGUKT

Acknowledgement

I would like to express my sincere gratitude to **N.Satyanandaram sir,** my project internal guide for valuable suggestions and keen interest throughout the progress of my course of research.

I am grateful to **N.Satyanandaram sir, HOD CSE**, for providing excellent computing facilities and a congenial atmosphere for progressing with my project

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work . I express my profound gratitude to all our friends and family members for this encouragement.

TABLE OF CONTENTS

1.Abstract

Different forms of remote education and massive online open courses are gaining reputation. The skill to proctor the online examination is a main essential factor for the scalability for promoting the students for next stage education. Existing manual monitoring is the most approaching method in education either by visually monitoring or by physically accompanying test takers to the examination center and monitoring them. verification and proctoring of online examinations is one of the main challenges in online learning systems. The migration and implementation of the online exam have been accelerated during the pandemic COVID-19. So the existing systems need a safe mechanism to authenticate and proctor online students. In this Project a remotely proctored exam is administered by experienced human proctors, an AI Algorithm, or both to maintain integrity. In this project, we present ways to improve reliability of proctoring by generating plots of each parameter measured during the online evaluation. The project makes use of a webcam and microphone connected to the PC/Laptop. Real-time video processing is enabled and the user is warned if he/she is suspected of cheating.

2.Introduction

In India, the number of internet users has nearly doubled in the past 6 years. This proved to be a boon for academics as many students could continue their education. This also facilitated examinations to go online which brought the concept of online proctoring at the academic level. A proctored exam allows the invigilators to invigilate remotely. They use video, audio, and various anti-cheating features to maintain the exam's credibility.

3.Objective

The objective of Remote Proctoring software is to supervise students while conducting exams. Thus, developing computer algorithms to Identify students cheating. Student camera access is taken then monitor them for unfair practices.

4.Scope

After using these technologies there can be significant reduction in cheating as compared to other means of taking tests remotely. Thus, an online examination system with an online proctoring system helps to hold fair online exams. The scope of secure online examinations is expanding day by day.

5.Existing System

Existing online systems have only one manual proctor for watching multiple students simultaneously which is not cost- effective and we have to depend on a manual proctor sitting at home or college for monitoring the students. If we continue like this traditional online proctoring systems then we need many proctors to conduct an examination or tests. when the proctor is focusing on one student, other students can cheat at that time. So, simultaneously proctoring the students is not Possible.

Disadvantages of Existing System

- Simultaneously Proctoring is not possible.
- If a student misses an exam rescheduling exam id not possible.
- Difficult to interface for students and it is not user friendly for students

6.Proposed System

We have proposed a web-based system to identify,and analyze the malpractices carried out by students during online examinations using Artificial intelligence. A webcam is installed into the computer of a student or the front camera if the student is giving an exam on a using face recognition the student is recognized and if the face matches with the registerd face image, then the student is verified and allowed to give the exam.

ADVANTAGES OF PROPOSED SYSTEM

- It verifies the student
- Face Recognition is to verify the students, which easily identify students faces and match them with their details.
- Voice Recognition helps pick up sounds and match it with the background noise to remove instances of cheating by recognizing speech patterns.
- This proctoring is User-friendly Interface for the students

7.REQUIREMENT SPECIFICATION

Hardware Configuration:

RAM	4 GB
HARD DISK	487.0GB
PROCESSOR	1AMD A9-9420 radeon r5,5 compute cores 2c+3g *2

8.Technologies Used

Front End

- ➢ Python

Back End

- ➢ Python

9.Libraries Used

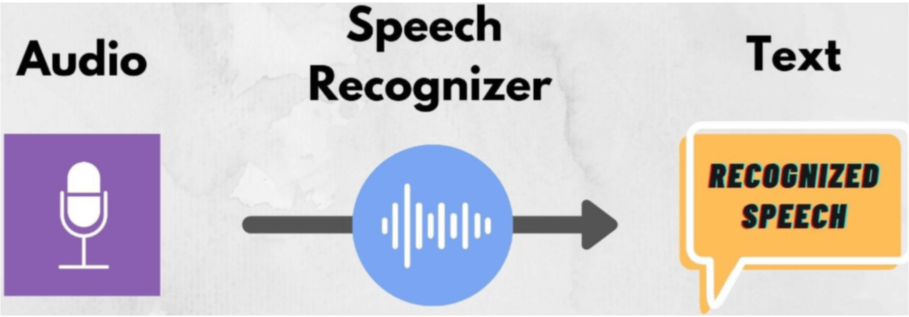
- ➢ PyAudio
- ➢ SpeechRecognition
- ➢ Media pipe
- ➢ Pillow
- ➢ NLTK
- ➢ Keras API.

9.1.PyAudio

PyAudio provides Python bindings for PortAudio v19, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms, such as GNU/Linux, Microsoft Windows, and Apple macOS.

9.2.Speech Recognition

Speech recognition is a machine's ability to listen to spoken words and identify them. You can then use speech recognition in Python to convert the spoken words into text, make a query or give a reply. You can even program some devices to respond to these spoken words.



9.3.NLTK

Natural language Processing(NLP) is a field that focuses on making natural human language usable by computer programs. NLTK, or Natural Language Tool Kit , is a Python package that you can use for NLP.

A lot of the data that you could be analyzing is unstructured data and contains human-readable text. Before you can analyze that data programmatically, you first need to preprocess it. In this tutorial, you'll take your first look at the kinds of text preprocessing tasks you can do with NLTK so that you'll be ready to apply them in future projects. You'll also see how to do some basic text analysis and create visualizations.

- ➢ StepWord
- ➢ Stemming
- ➢ Tokenization

Step Words

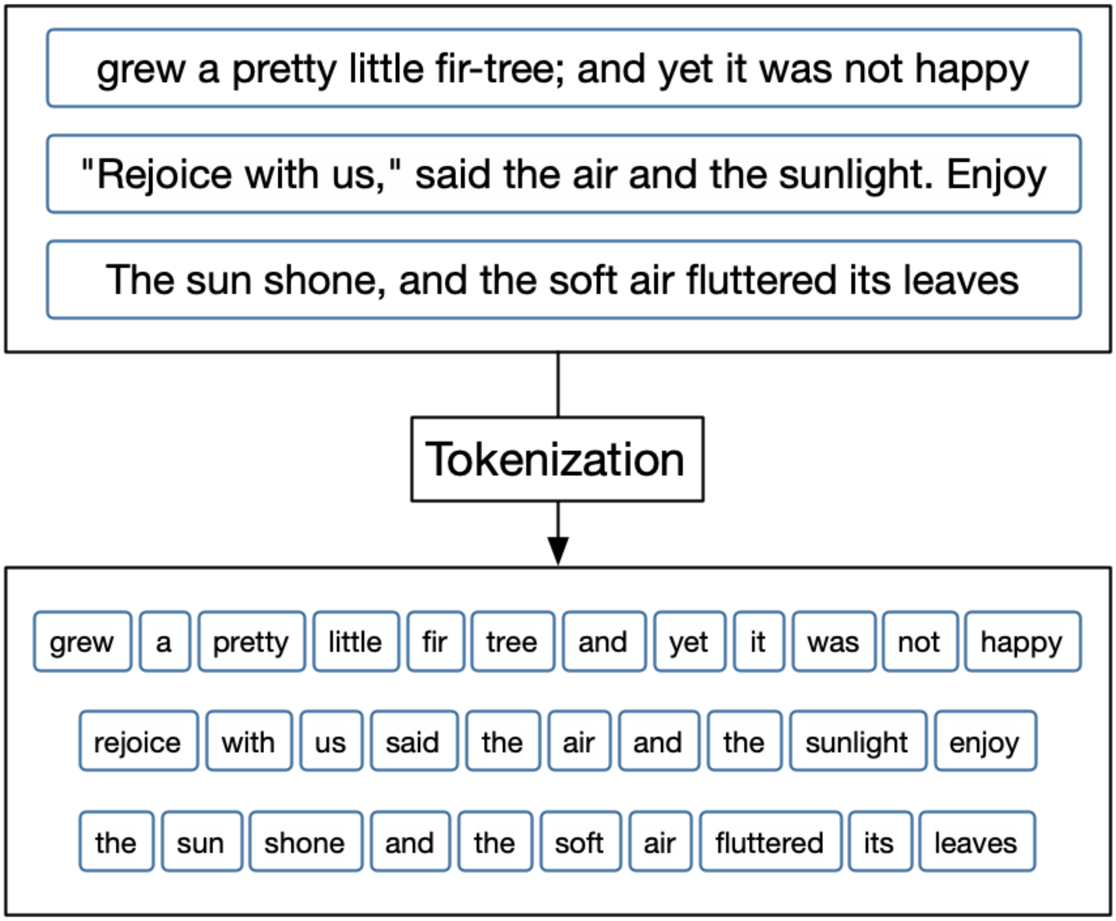
The words which are generally filtered out before processing a natural language are called stop words. These are actually the most common words in any language (like articles, prepositions, pronouns, conjunctions, etc) and does not add much information to the text.
Examples of a few stop words in English are "the", "a", "an", "so", "what".

Stemming

Stemming is a technique used to extract the base form of the words by removing affixes from them.It is just like cutting down the branches of a tree to its stem

Tokenization

Tokenization is the process of breaking a stream of text up into words,phrases,symbols,or other meaningful elements called Tokens.


grew a pretty little fir-tree; and yet it was not happy
"Rejoice with us," said the air and the sunlight. Enjoy
The sun shone, and the soft air fluttered its leaves
Tokenization
grew
a
pretty
little
fir
tree
and
yet
it
was
not
happy
rejoice
with
us
said
the
air
and
the
sunlight
enjoy
the
sun
shone
and
the
soft
air
fluttered
its
leaves

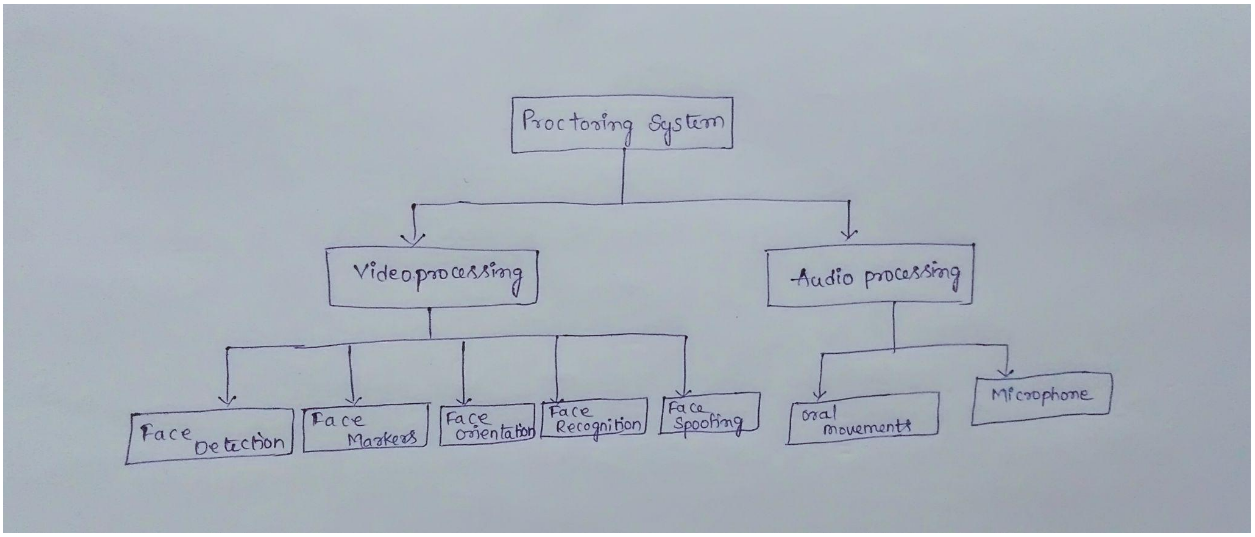
10.Methodology

The First step is to registration of students using their image for five times and face image on the platform. For every test a student get register with the latest face image which will be verified with an image stored in database. Face detection, Mouth open detection, Eye tracking, Multiple Face detection and no face detection will get detected.

Head Posing will be tracked, Voices will be detected and recorded as file,if any words that match with the question paper it will mark as detected . After clicking Esc the proctoring will be stopped,then finally report will be generated.

11.System Design

The design which is used to design the software related requirements. In this paper, complete system design is generated and shows how the modules, sub modules and the flow of the data between them are done and integrated. It is very simple phase that shows the implementation process. The errors done here will be modified in the coming processes.



The system design mainly consists of:

- **Video Processing**
  - **Face Detection**
  - **Face Markers**
  - **Face Orientation**
  - **Face Recognition**
  - **Face Spoofing**
- **Audio Processing**
  - **Oral Movements**
  - **Micro Phone**

Video Processing

- **FaceDetection:** we used MediaPipe(open source cross-platform, customizable ML solutions for live and streaming media) for detecting faces in real-time and drawing b boxes around the detected areas in the image

- **Face markers :** We make use of Mediapipe's "facemesh" for tracking eyes, mouth opening detection, and head pose estimation.

- **Face orientation :** The orienation that the face makes with the assumed axis is noted and orientations are classifed into Straight,right,left,up ordown.

- **Face Recognition :** The face detected from the face detector is compared with the registered faces(will talk about this below) and if an unknown face is detected, the user is suspected to be cheating. We use "Facenet" model for this purpose.

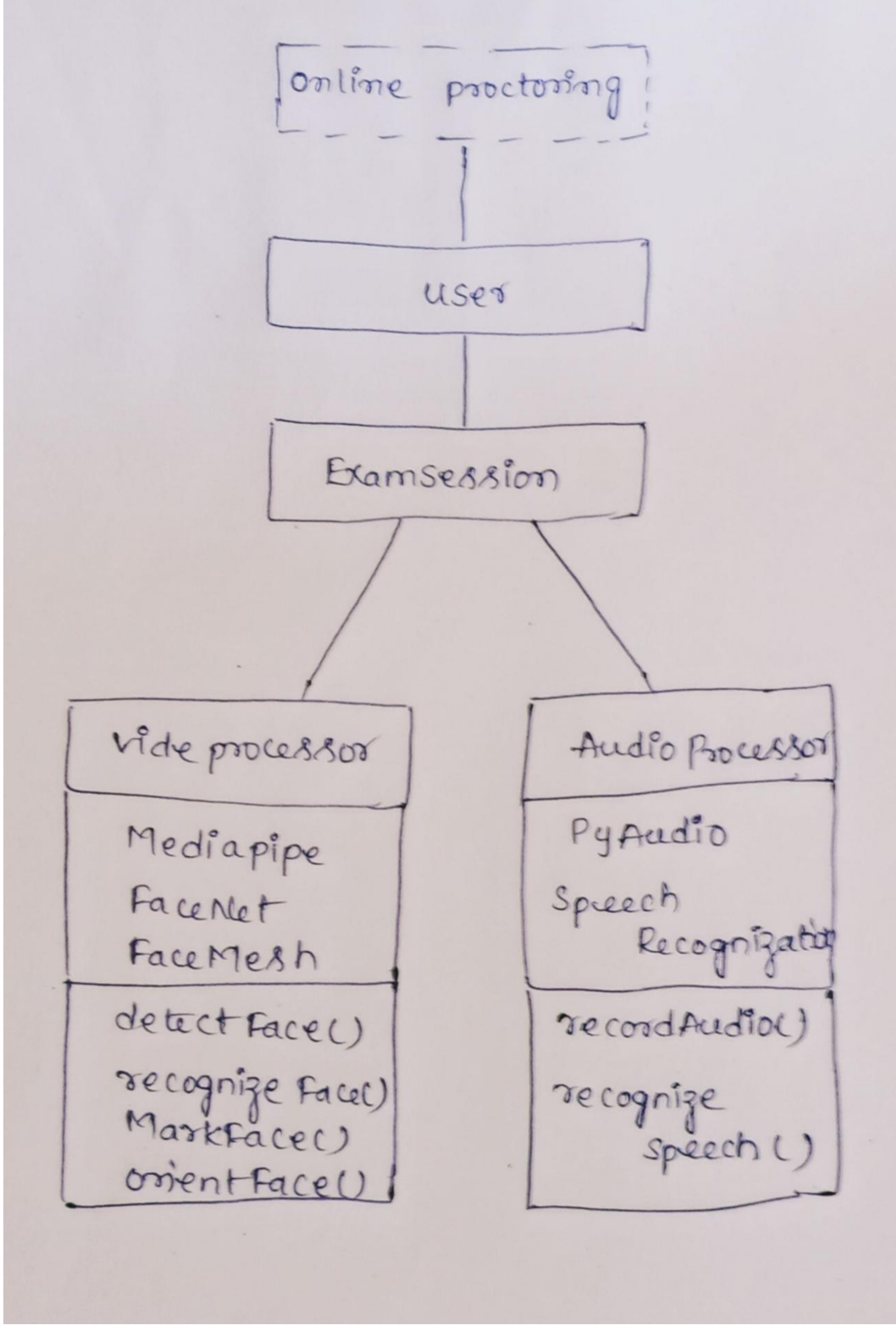
- **Face Spoofing :** It is used for finding whether the face is real or a photograph/image from a phone. The anti-spoofing system is implemented by using a pre-trained model with its weights. The models us

Audio Processing

1. **Oral movements :** The software detects whether the user is trying to speak by observing his oral movements. The landmarks detected on the face are used for detecting whether the mouth is open or closed.

2. **Microphone :** we made use of libraries like PyAudio ,speech_recognition. Audio from the microphone is recorded and converted to text using Google's speech recognition API. A different thread is used to call the API such that the recording portion is not disturbed a lot, which processes the last one, appends its data to a text file and deletes it. Using NLTK we removed the stopwords from that file. The question paper (in txt format) is taken whose stopwords are also removed and their contents are compared. Finally, the common words along with its number are presented to the proctor to determine whether the user cheated.

12.UML DIAGRAM



13.ALGORITHMS

CNN Algorithm:

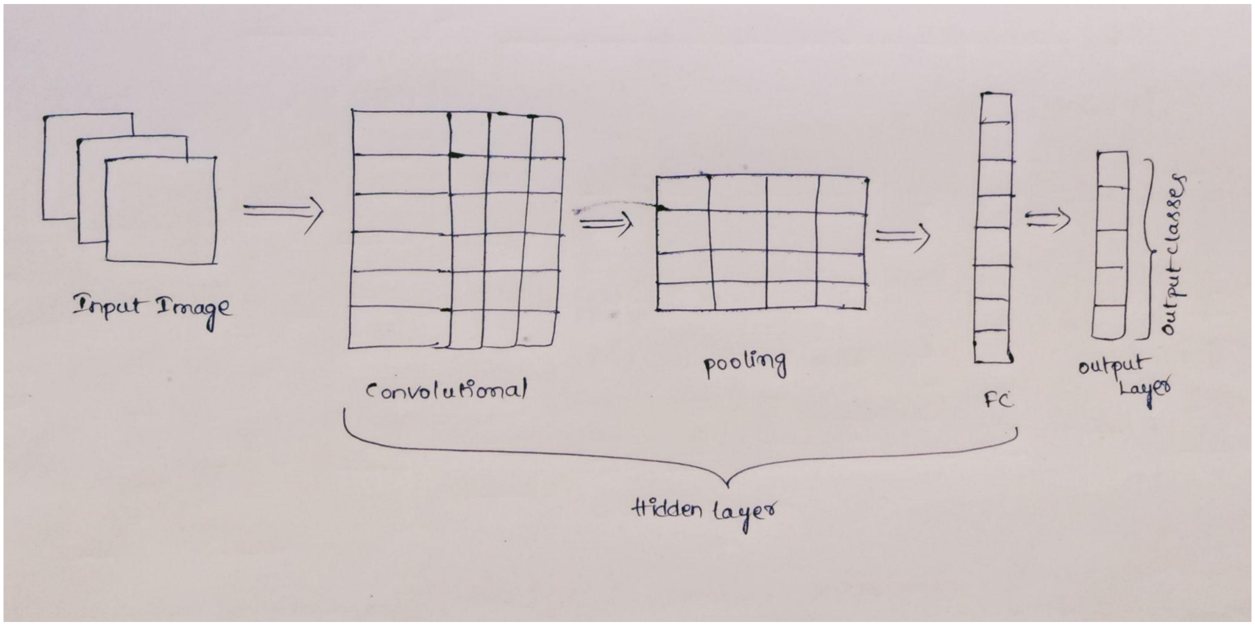
Convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data. Convolutional neural network is the special type of feed forward artificial neural network in which connectivity between the layers are inspired by visual cortex.

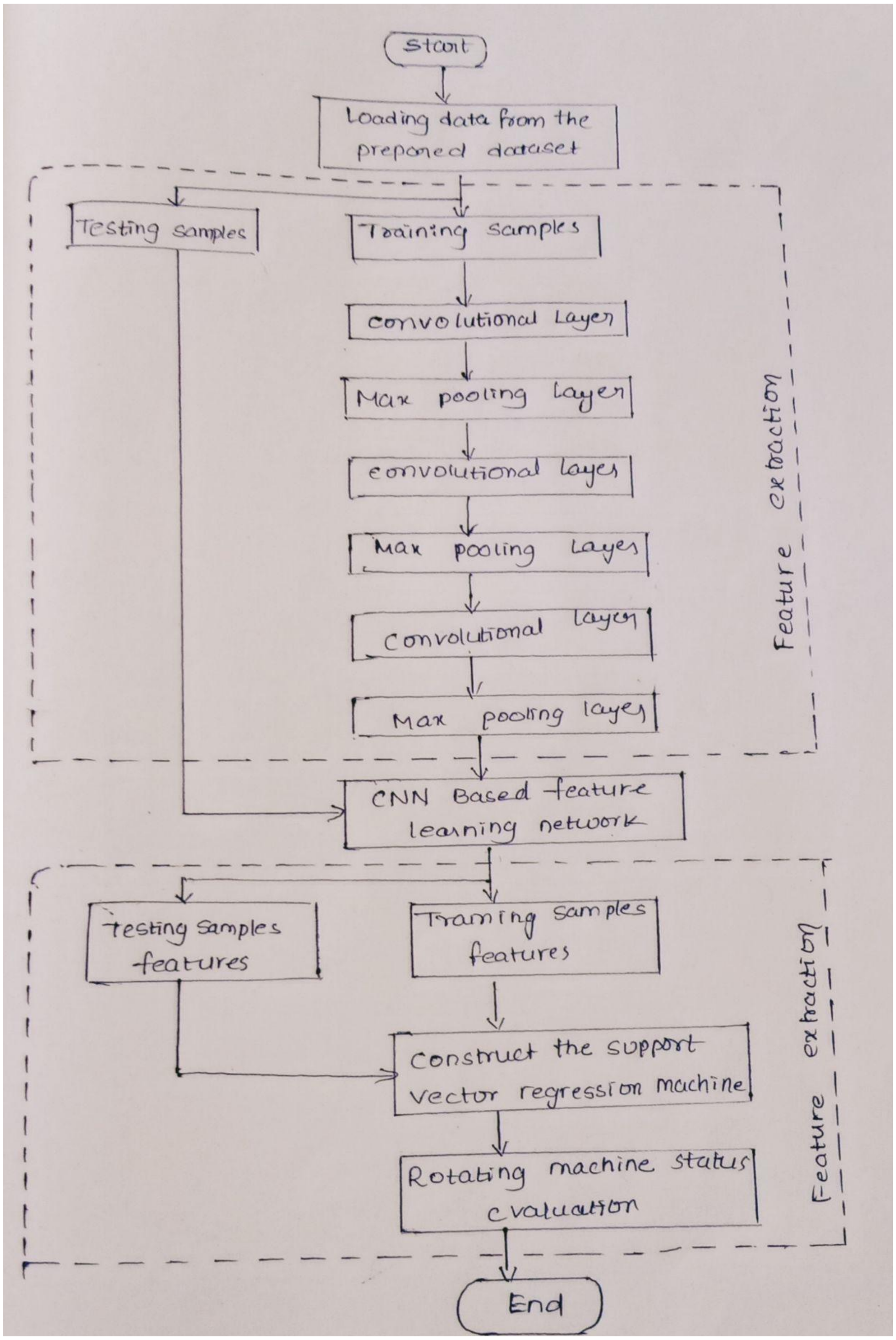
CNN is a class of deep neural network which is applied for analyzing visual imagery.In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications.

Objects detections, recognition faces etc., are some of the area where CNNs are widely used. Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data.It is a mathematical operation that takes two inputs such as image matrix and a filter.

The input to the fully connected layer is the output from the final pooling or convolutional layer, which is flattened and then fed into the fully connected layer.

It reduces the number of parameters to learn and the amount of computation performed in the network. The pooling layer summarizes the features present in the region of the feature map generated by a convolutional layer.


Input Image
Convolutional
pooling
FC
output Layer
output classes
Hidden layer

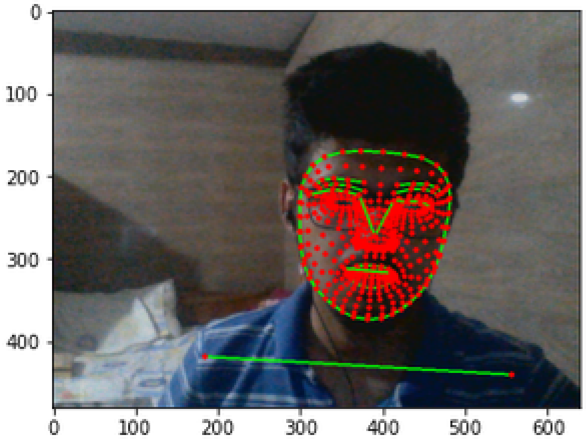

Start
Loading data from the prepared dataset
Testing samples
Training samples
convolutional Layer
Max pooling layer
convolutional layer
Max pooling Layer
Convolutional layer
Max pooling layer
Feature extraction
CNN Based feature learning network
testing samples features
Training samples features
Construct the support vector regression machine
Rotating machine status evaluation
Feature extraction
End

14.MediaPipe Library

MediaPipe is an open-source, cross-platform Machine Learning framework used for building complex and multimodal applied machine learning pipelines. It can be used to make cutting-edge Machine Learning Models like face detection, multi-hand tracking, object detection, and tracking, and many more.

MediaPipe basically acts as a mediator for handling the implementation of models for systems running on any platform which helps the developer focus more on experimenting with models, than on the system.

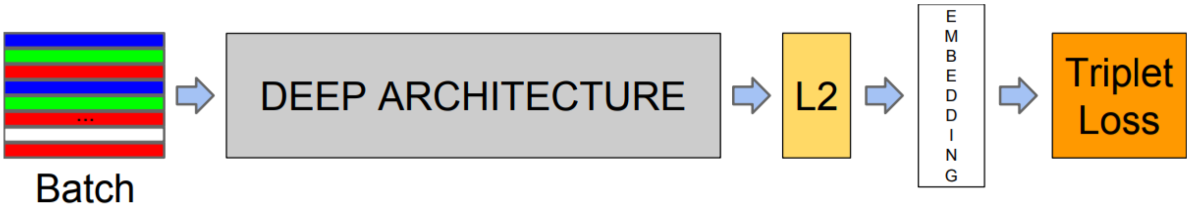
Facial LandMarks Using Mediapipe



15.FaceNet:

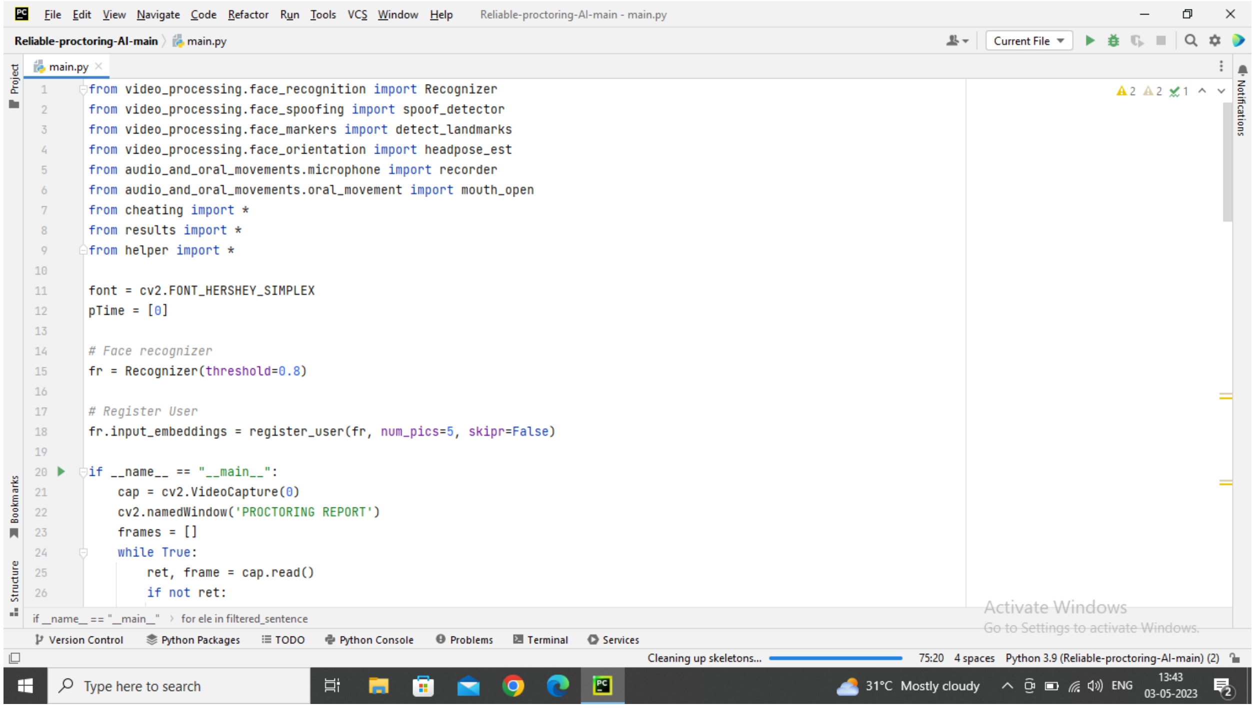
FaceNet is the name of the facial recognition system that was proposed by Google Researchers in 2015 in the paper titled *FaceNet: A Unified Embedding for Face Recognition and Clustering*. It achieved state-of-the-art results in the many benchmark face recognition dataset such as Labeled Faces in the Wild (LFW) and Youtube Face Database.

They proposed an approach in which it generates a high-quality face mapping from the images using deep learning architectures such as ZF-Net and Inception Network. Then it used a method called **triplet loss** as a loss function to train this architecture

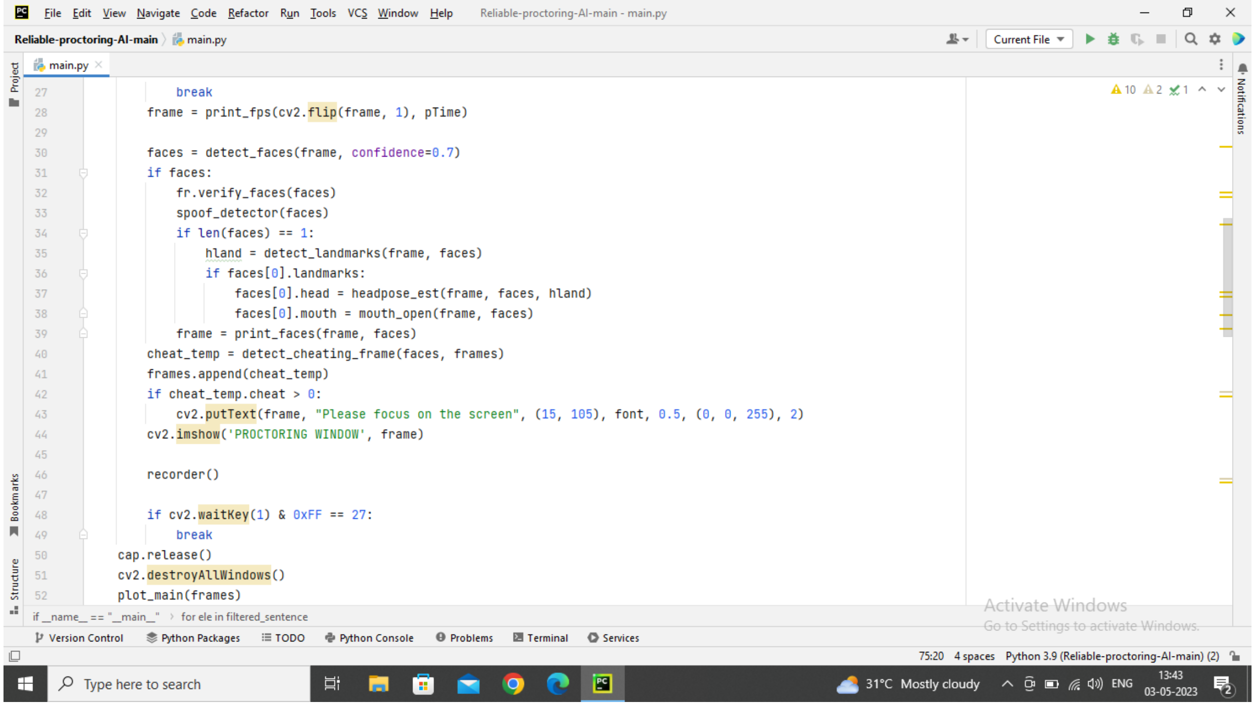


16.Evaluation

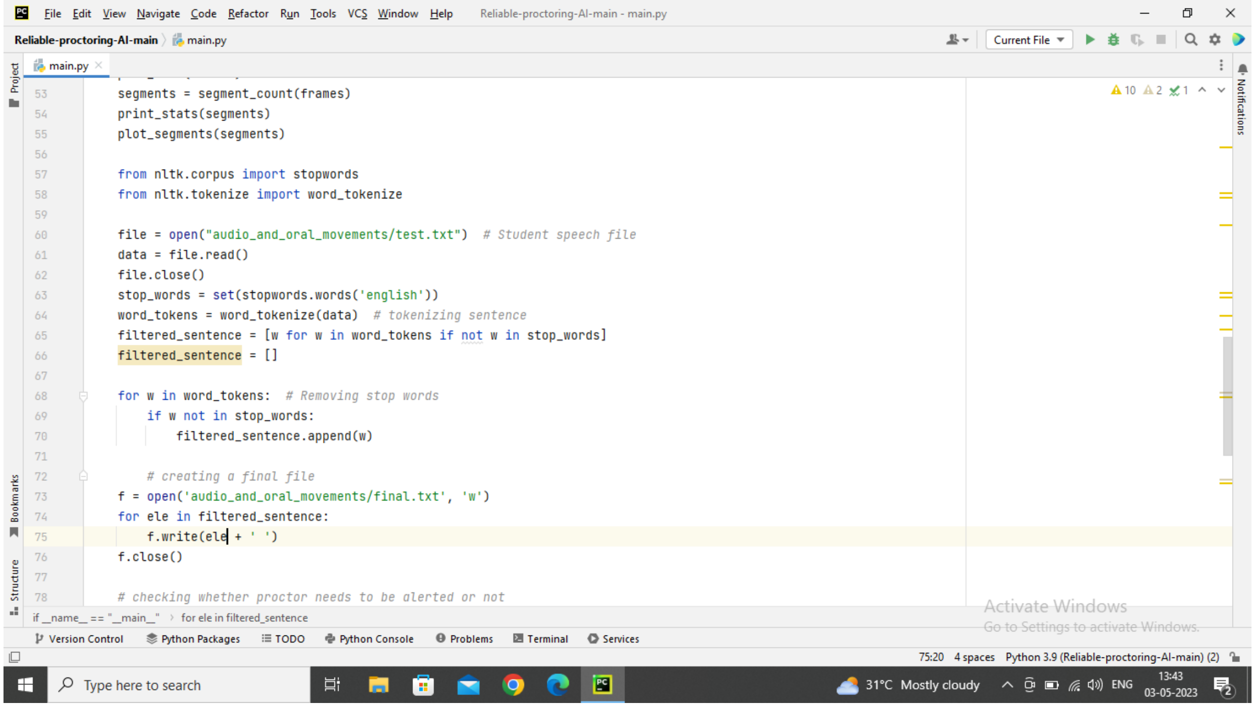
Sample Code:


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Reliable-proctoring-AI-main - main.py
Reliable-proctoring-AI-main
main.py
Current File
Project
Bookmarks
Structure
Notifications
if __name__ == "__main__"
for ele in filtered_sentence
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services
Cleaning up skeletons...
75:20
4 spaces
Python 3.9 (Reliable-proctoring-AI-main) (2)
Activate Windows
Go to Settings to activate Windows.
Type here to search
31°C Mostly cloudy
ENG
13:43
03-05-2023

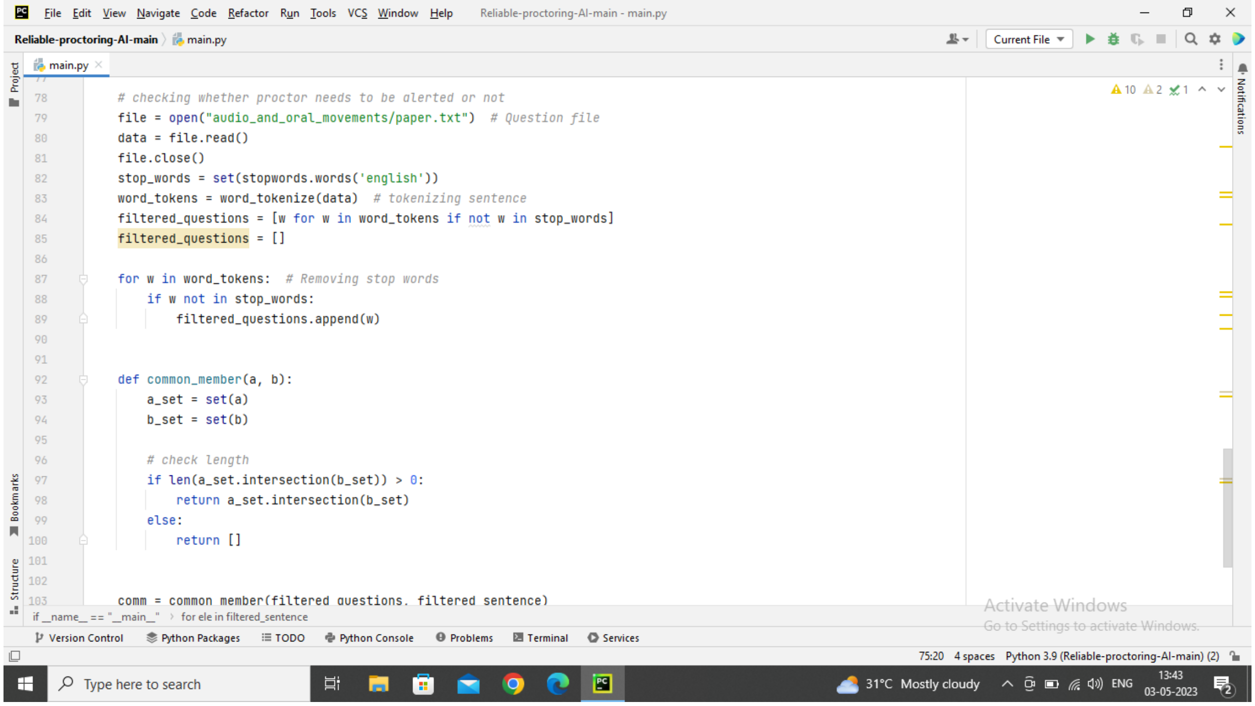
```python
from video_processing.face_recognition import Recognizer
from video_processing.face_spoofing import spoof_detector
from video_processing.face_markers import detect_landmarks
from video_processing.face_orientation import headpose_est
from audio_and_oral_movements.microphone import recorder
from audio_and_oral_movements.oral_movement import mouth_open
from cheating import *
from results import *
from helper import *

font = cv2.FONT_HERSHEY_SIMPLEX
pTime = [0]

# Face recognizer
fr = Recognizer(threshold=0.8)

# Register User
fr.input_embeddings = register_user(fr, num_pics=5, skipr=False)

if __name__ == "__main__":
    cap = cv2.VideoCapture(0)
    cv2.namedWindow('PROCTORING REPORT')
    frames = []
    while True:
        ret, frame = cap.read()
        if not ret:
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Reliable-proctoring-AI-main - main.py
Reliable-proctoring-AI-main 〉 main.py
Current File
main.py
if __name__ == "__main__" 〉 for ele in filtered_sentence
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services
75:20 4 spaces Python 3.9 (Reliable-proctoring-AI-main) (2)
Type here to search
31°C Mostly cloudy
ENG
13:43
03-05-2023

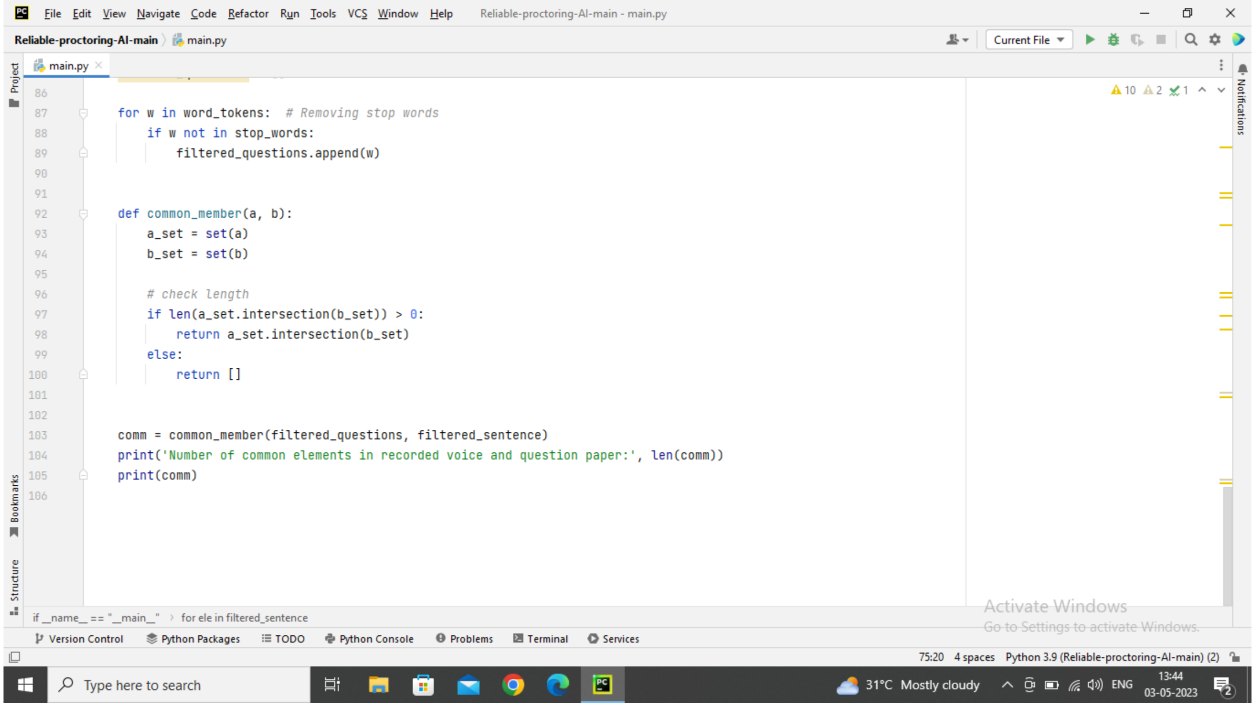
```
            break
        frame = print_fps(cv2.flip(frame, 1), pTime)

        faces = detect_faces(frame, confidence=0.7)
        if faces:
            fr.verify_faces(faces)
            spoof_detector(faces)
            if len(faces) == 1:
                hland = detect_landmarks(frame, faces)
                if faces[0].landmarks:
                    faces[0].head = headpose_est(frame, faces, hland)
                    faces[0].mouth = mouth_open(frame, faces)
            frame = print_faces(frame, faces)
        cheat_temp = detect_cheating_frame(faces, frames)
        frames.append(cheat_temp)
        if cheat_temp.cheat > 0:
            cv2.putText(frame, "Please focus on the screen", (15, 105), font, 0.5, (0, 0, 255), 2)
        cv2.imshow('PROCTORING WINDOW', frame)

        recorder()

        if cv2.waitKey(1) & 0xFF == 27:
            break
    cap.release()
    cv2.destroyAllWindows()
    plot_main(frames)
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Reliable-proctoring-AI-main - main.py
Reliable-proctoring-AI-main 〉 main.py
Current File
main.py
if __name__ == "__main__" 〉 for ele in filtered_sentence
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services
75:20 4 spaces Python 3.9 (Reliable-proctoring-AI-main) (2)
Type here to search
31°C Mostly cloudy
ENG
13:43
03-05-2023

```
    segments = segment_count(frames)
    print_stats(segments)
    plot_segments(segments)

    from nltk.corpus import stopwords
    from nltk.tokenize import word_tokenize

    file = open("audio_and_oral_movements/test.txt")  # Student speech file
    data = file.read()
    file.close()
    stop_words = set(stopwords.words('english'))
    word_tokens = word_tokenize(data)  # tokenizing sentence
    filtered_sentence = [w for w in word_tokens if not w in stop_words]
    filtered_sentence = []

    for w in word_tokens:  # Removing stop words
        if w not in stop_words:
            filtered_sentence.append(w)

        # creating a final file
    f = open('audio_and_oral_movements/final.txt', 'w')
    for ele in filtered_sentence:
        f.write(ele + ' ')
    f.close()

    # checking whether proctor needs to be alerted or not
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Reliable-proctoring-AI-main - main.py
Reliable-proctoring-AI-main 〉 main.py
main.py
Current File
Project
Bookmarks
Structure
Notifications
if __name__ == "__main__" 〉 for ele in filtered_sentence
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services
75:20 4 spaces Python 3.9 (Reliable-proctoring-AI-main) (2)
Activate Windows
Go to Settings to activate Windows.
Type here to search
31°C Mostly cloudy
ENG
13:43
03-05-2023

```
    # checking whether proctor needs to be alerted or not
    file = open("audio_and_oral_movements/paper.txt")  # Question file
    data = file.read()
    file.close()
    stop_words = set(stopwords.words('english'))
    word_tokens = word_tokenize(data)  # tokenizing sentence
    filtered_questions = [w for w in word_tokens if not w in stop_words]
    filtered_questions = []

    for w in word_tokens:  # Removing stop words
        if w not in stop_words:
            filtered_questions.append(w)


    def common_member(a, b):
        a_set = set(a)
        b_set = set(b)

        # check length
        if len(a_set.intersection(b_set)) > 0:
            return a_set.intersection(b_set)
        else:
            return []


    comm = common_member(filtered_questions, filtered_sentence)
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Reliable-proctoring-AI-main - main.py
Reliable-proctoring-AI-main 〉 main.py
main.py
Current File
Project
Bookmarks
Structure
Notifications
if __name__ == "__main__" 〉 for ele in filtered_sentence
Version Control
Python Packages
TODO
Python Console
Problems
Terminal
Services
75:20 4 spaces Python 3.9 (Reliable-proctoring-AI-main) (2)
Activate Windows
Go to Settings to activate Windows.
Type here to search
31°C Mostly cloudy
ENG
13:44
03-05-2023

```
    for w in word_tokens:  # Removing stop words
        if w not in stop_words:
            filtered_questions.append(w)


    def common_member(a, b):
        a_set = set(a)
        b_set = set(b)

        # check length
        if len(a_set.intersection(b_set)) > 0:
            return a_set.intersection(b_set)
        else:
            return []


    comm = common_member(filtered_questions, filtered_sentence)
    print('Number of common elements in recorded voice and question paper:', len(comm))
    print(comm)
```

17.RESULT:

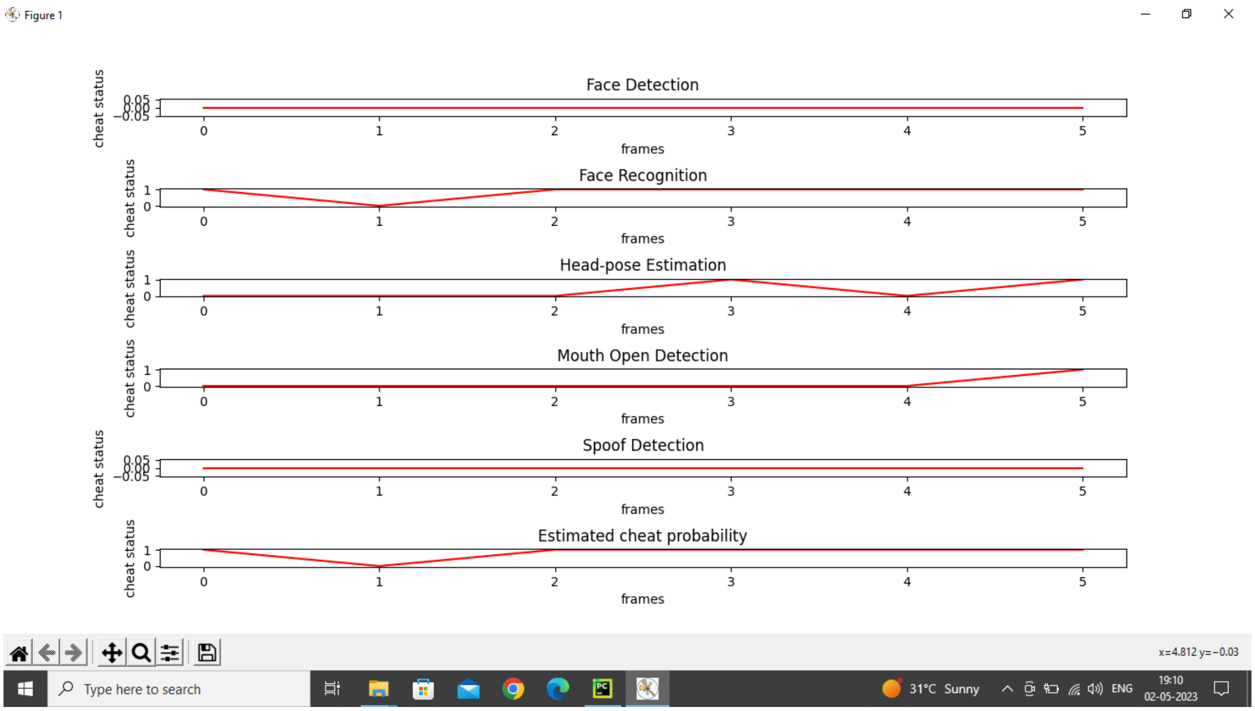
Registration of the User

Alerting the User(Looking Right)

Mouth Opening

Face Spoofing

Proctoring Report:



18.Conclusion:

Our system ensures the ethical examination environment without any malpractice activities. Our system keeps the track of the activities of the students from remote places. Our system can help in detection and prevention of cheating during the online exam process. Entire examination activity can be completed in a paperless manner. We can save a huge cost of exam administration and management by shifting tousing our system. Our system can be used along with traditional exam process

19.Future Enhancement

It is possible to create an AI proctoring system with high accuracy. Through this project, we will try to show that online proctoring is the future, and using online proctoring cheating in exams can be reduced drastically.