[P-1]

A PRELIMENERY REPORT ON

CONTINUOUS USER VERIFICATION FOR ONLINE EXAM PROCTORING BASED ON MACHINE LEARNING AND DEEP LEARNING

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

Hemant Vijay Mhapadi	Exam No :	B150444229
Yash Bhausaheb Hure		B150444218
Nitin Dilip Dange		B150444207

DEPARTMENT OF COMPUTER ENGINEERING

SIDDHANT COLLEGE OF ENGINEERING

SUDUMBARE, TALEGAON CHAKAN HIGH WAY, PUNE 412109
SAVITRIBAI PHULE PUNE UNIVERSITY
2020 -2021

[P-2]

CERTIFICATE

This is to certify that the project report entitles

“CONTINUOUS USER VERIFICATION FOR ONLINE EXAM PROCTORING BASED ON MACHINE LEARNING AND DEEP LEARNING”

Submitted by

Hemant Vijay Mhapadi	Exam No : B150444229
Yash Bhausaheb Hure	B150444218
Nitin Dilip Dange	B150444207

Is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Prof. Priyanka Kale** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

(Prof. Priyanka Kale)
Guide
Department of Computer Engineering

(Dr. S.Gupta)
Head,
Department of Computer Engineering

(Dr. J. U. Panchal)
Principal,
Siddhant College of Engineering Pune – 412109

Place : Pune

Date : 26/4/2021

ACKNOWLEDGEMENT

Without the help of guide we would have never succeeded in completing our Project. Their cooperation, encouragement and help, helped us a lot. First and foremost, we wish to record our sincere gratitude to Management of this college and to Dr. J**.** U**.** Panchal Principal, Siddhant College of Engineering, Pune for his constant support and encouragement.

Our sincere thanks to Prof. S.Gputa Head, Department of Computer Engineering, for her valuable suggestions and guidance throughout the period of this report. We express our sincere gratitude to our internal and external guides, Prof Kamal Reddy for guiding us in investigations for this project. Our numerous discussions with them were extremely helpful. We hold them in esteem for guidance, encouragement and inspiration received from him.

Our sincere thanks to Prof. Priyanka Kale, Project Coordinator for having supported us with the work related to this project. Last but not the least, we wish to thank our parents for helping in our studies in the college as well as for constantly encouraging us to learn engineering.

NAME OF THE STUDENTS

Nitin Dilip Dange

Yash Bhausaheb Hure

Hemant Vijay Mhapadi

ABSTRACT

Educational institutions are using online examination system to improve the quality of instructions by using a supervised measure for their students to self -paced learning environment. E-learning became so popular because of its fast feedback to the candidates. An online examination system has capability to address academic malpractice. One of the perks of online examination system is saving time and also have limitations on dependency to the quality of internet services. The research looked into interviewing through a focus group the proctors of online exams to identify root causes of academic malpractice at the same time interview exam content creators on possible approaches on exam questions generators that allow a validity of measure of outcomes. Generally, a final validation done by the focus group respondents and end users for effectivity and usability.

[P3]

TABLE OF CONTENTS

CHAPTER **TITLE**

LIST OF ABBREVATIONS

ABBREVIATION	ILLUSTRATION
VPN	Virtual Private Network
IP	Internet Protocol
IDS	Intrusion Detection System
TCP	Transmission Control Protocol

Contents

Appendices

List of Figures

List of Tables

Introduction

Exams are a critical component of any educational program, and online educational programs are no exception. In any exam, there is a possibility of cheating, and therefore, its detection and prevention are important. Educational credentials must reflect actual learning in order to retain their value to society. A typical testing procedure for online learners is the students come to an on-campus or university-certified testing centre and take an exam under human proctoring. New emerging technologies which allow students to take tests anywhere as long as they have an Internet connection.

However, they still rely on a person “watching” the exam-taking. Exposing the long-established method might prove to be unsuccessful to fully prevent academic malpractice during examinations. E-learning has its vital and integral assessment component using online examination. Submitting exams in E-learning has already been done without a proctor present. As a result, students can easily commit academic malpractice during exams, educational institutions with E-learning depend on an examination process on which students take the exam in a physical controlled environment at the institution under a supervised condition, however, this contradicts the concept of the live E-learning environment.

The ability to efficiently proctor remote online examinations is an important limiting factor to the scalability of this next stage in education. Presently, human proctoring is the most common approach of evaluation, by either requiring the test taker to visit an examination centre, or by monitoring them visually and acoustically during exams via a webcam. How-ever, such methods are labour-intensive and costly. Saving time is one of the perks in having an Online examination system, but it also had limitations on dependency to the quality of Internet service leaving both the proctor and the examiners not being able to use the system. Use of m-learning or other remote education continue to increase due to its ability to reach people who don’t have access to campus. A visual verification for the whole exam session is needed in an online exam, therefore a face verification is needed. A remaining problem in face recognition area is the system robustness.

Problem Definition:

Enormous open online courses offer the potential to significantly expand the reach of today’s educational system, by providing a wider range of educational resources to enrolled students and by making educational resources available to people who cannot access a campus due to location or schedule constraints. Instead of taking courses in a classes on campus, now students can take courses anywhere in the world using a computer, where educators/teachers deliver knowledge via various types of multimedia content. Exams are a critical component of an educational program, and online educational programs are no exception. In any exam, there is a possibility of cheating, and therefore, its detection and prevention are important. A typical testing procedure for online learners is the following: students come to an on-campus or university-certified testing centre and take an exam under human proctoring. The proctors are trained to watch and listen for any unusual behaviours of the test taker(candidates), such as unusual eye movements, or removing oneself from the field of view. They can alert the test taker or even stop the test. In this paper, we introduce a multimedia analytics system to perform automatic and continuous online exam proctoring (OEP). The overall goal of this system is to maintain academic integrity of exams, by providing real-time proctoring for detecting the majority of cheating behaviours of the test taker. To achieve such goals, audio-visual observations about the test takers are required to be able to detect any cheat behaviour. This system monitors such cues in the room where we will moniter candidate using camera and microphones.

Purpose of System:

In this project, we focus to develop a multi-tasking analysis system to detect a broad variety of cheating during online examination. System include user verification, text detection, speech detection, active window detection, gaze estimation, face detection, person detection etc. This phase system includes 3 phases:

a)Preparation Phase: In this phase, the candidate has to authenticate himself/herself before beginning the exam, by using a password and face recognition. This phase further includes calibration steps to ensure that all sensors are connected and functioning properly. Further, the candidate learns and verbally acknowledges the rules of using the online exam proctoring (OEP) system, such as, he/she has to rotate his/her webcam to 270-360 degree so that we can get the all information about the candidate surroundings. Secondly, no second person is allowed in the same room. There should be light in candidate room so that we clearly perceive his actions, etc

b)The exam Phase: The candidate takes the exam, under the continuous monitoring through our OEP system for real time cheating detection. Webcam is used to monitor the user. Using it, we capture images and video used for user verification, gaze estimation, speech detection, text detection, active window detection and phone detection.

c)Submission Phase: If user do not found violating any rules and condition then test will be submitted normally after time ends. If user found violating rules, He will be given 3-5number of chances for betterment. If he continue to violating rules again and again then he will be terminate by the system or admin.

Flow Diagram shows the working of Online Proctor System,

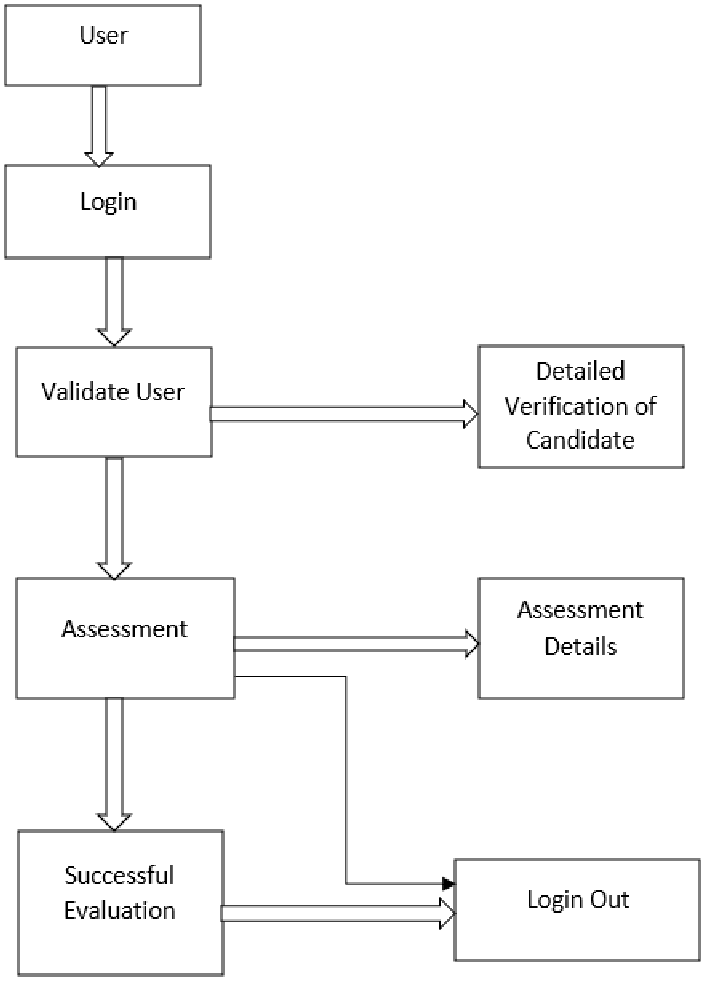


Figure. Flow Diagram of System

Literature Survey

Sr.no.	Literature	Methodology
1.	A Deep Learning approach for face detection using YOLO.	YOLO framework for detecting objects.
2.	Automated online exam proctoring.	User verification, speech detection, active window detection, phone detection.
3.	Gaze tracking system using structure sensor and gaze camera.	Gaze tracking, face and eye detection.
4.	Online examination system with cheating prevention using question bank randomization and tab locking.	Royce model for preparation, development, validation, modification and evaluation.
5.	An intelligent system for online proctor monitoring.	Time duration for face disappearance action.
6.	U2Eyes: a binocular dataset for eye tracking and gaze estimation	Binocular images created using UnityEyes
7.	Heuristic based approach for online exam proctoring.	Face detection using inference of activities by user
8.	Face detection and recognition system using digital image processing.	Process of Face Detection System.
9.	Identity aware face super resolution for low resolution face recognition.	Face super resolution network architecture and identity aware loss.
10.	Google API Speech to Text	The conversion of speech to text using Google API

Table. Literature Survey

Problem Statement

The title of this project is “**Continuous user verification for online exam proctoring using Machine Learning and Deep Learning**”. It is an online proctoring system which helps organisations, test- takers and various others professional to assess individuals for any specific reason without actual presence of a physical proctor. It can also be used for distance learning and assessment as it is not required for the candidate to be physically present at the assessment centre.

In today’s modern world time is everything. No one can afford any heavy expense of time for a particular work, such as candidate assessment which can be done online. For, this purpose the plan of the project is to develop a system which will continuously proctor the candidate undergoing the assessment and prevent any malpractice that the user may perform and maintain the dignity of candidate assessment.

The user can only undergo the online proctoring assessment on a desktop or a laptop. In this process the user will be prompted to allow mic and front cam to be active throughout the session and the system will keep an eagle’s eye on the candidate. The candidate if performs any malpractice will be warned about the followed misconduct, and if the warning prompts a limited number of times, then user will automatically be logged out of the assessment and won’t be accepted for any further assessment.

PROJECT DESIGN

6.1 Software Requirements:

Operating System - Any stable version of Browser.

6.1.1 Security Requirements:

To access a new level authentication and authorization by passwords.

- **Perimeter Security**: Network security, firewalls, and, ultimately, authentication to confirm user identities is guided.
- **Data Security**: Data is protected from unauthorized user by providing user login and password credentials.
- **Access Security**: Defining what authenticated users and applications can do with the data in the cluster through providing access to staff via there login and password credentials.

6.1.2 Technologies Used

- Python IDE.
- OpenCV.
- Flask.
- MySQL.

6.1.3 Requirements

- Internet Explorer 9.0 and higher
- Google Chrome latest stable release
- Mozilla Firefox (All Versions)

6.1.4 Performance Requirements

- Accuracy.
- Processing Time
- Memory Usage

6.1.5 Safety Requirements

Login is made successful.

6.1.6 Security Requirements

Authentication and authorization are done strongly, which has data integrity and confidentiality.

6.2 Data Flow Diagrams

6.2.1 Level 0:

It is also known as context diagram. It’s designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as single bubble with input and output data indicated by incoming-outgoing arrows.

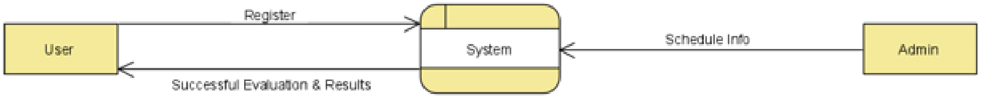


Figure. Level 0

6.2.2 Level 1:

Context diagram is decomposed into multiple bubbles/processes.in this level we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses.

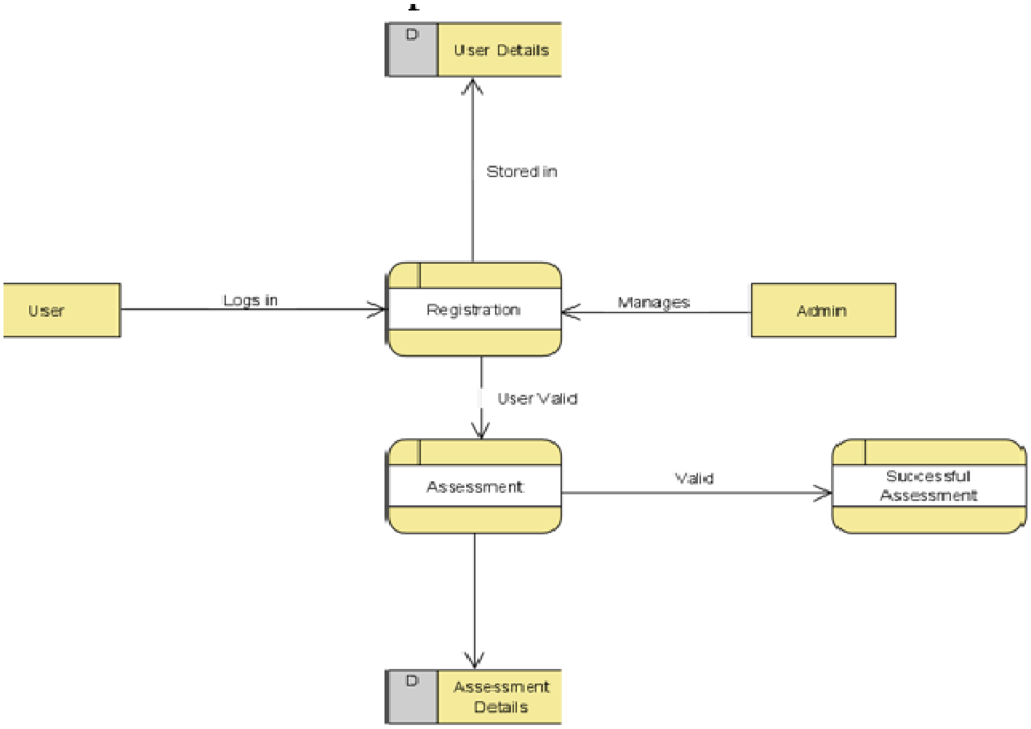


Figure. Level 1

6.2.2 Level 2:

Level-2 DFD goes one step deeper into parts of level-1 DFD. It can be used to plan or record the specific/necessary detail about the system's functioning.

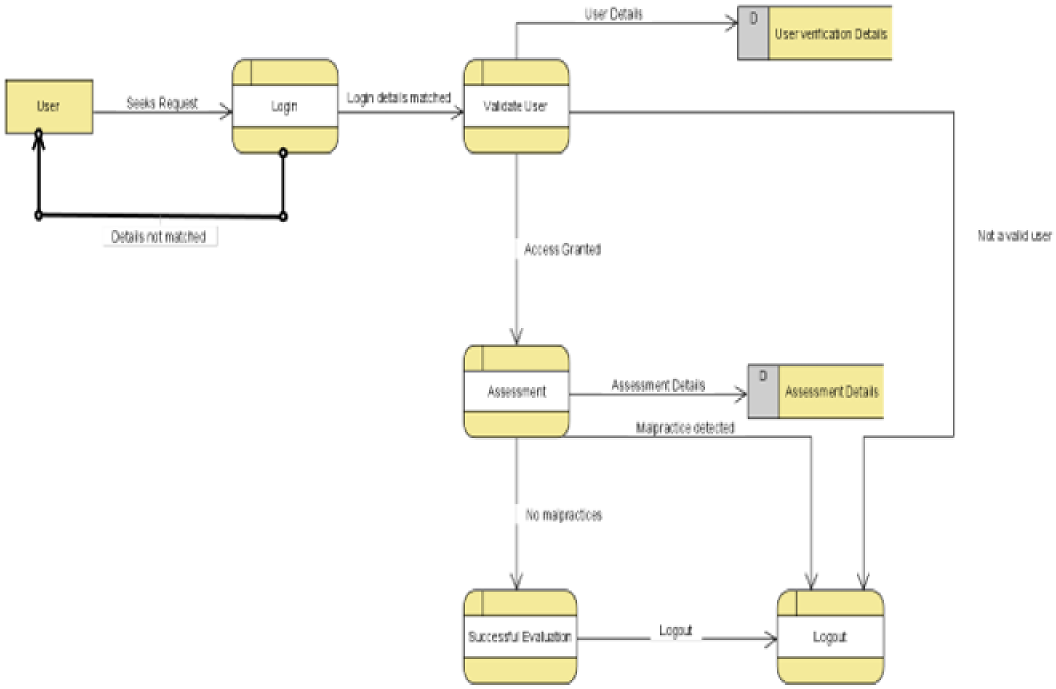


Figure. Level 2

6.3 UML Diagrams

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

6.3.1 Activity diagram 1:

This diagram depicts the entire system in a modelled way such that they are in an outer look.

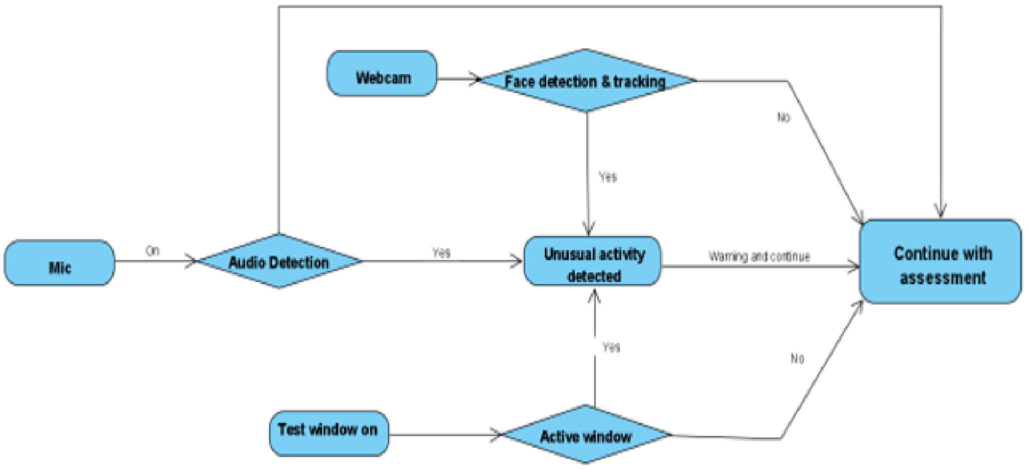


Figure. Activity Diagram 1

6.3.2 Activity diagram 2:

It models the entire process of user verification throughout the system.

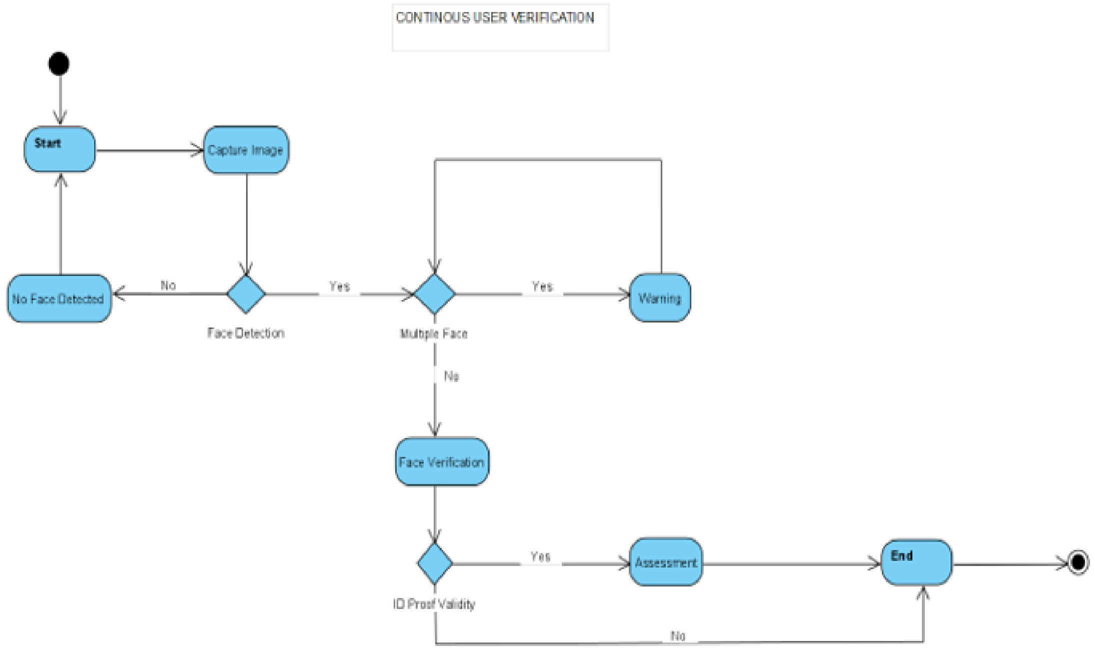


Figure. Activity Diagram 2

6.3.3 Class Diagram:

It is the main building block of object-oriented modelling. It is used for detailed conceptual modelling and building of the system. Almost all of the UML models are developed by considering this model as a base model. A class diagram can be used to display logical classes, which are typically the kinds of things. It can also be used to show implementation classes, which are the things the programmers typically deal with.

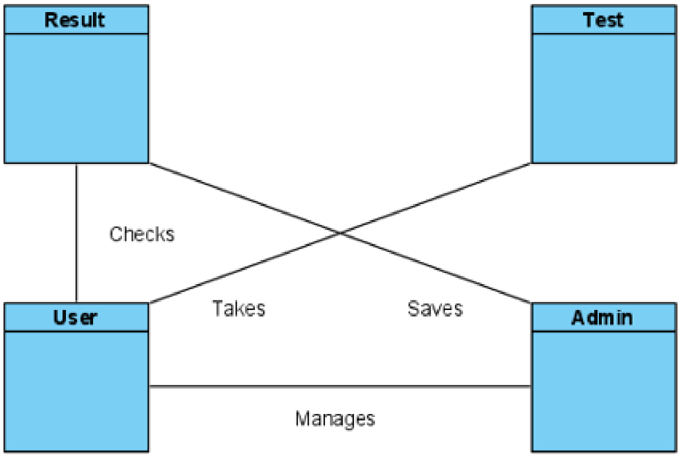


Figure. Class Diagram

6.3.4 Use Case:

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. It provides the simplified and graphical representation of what the system must actually do.

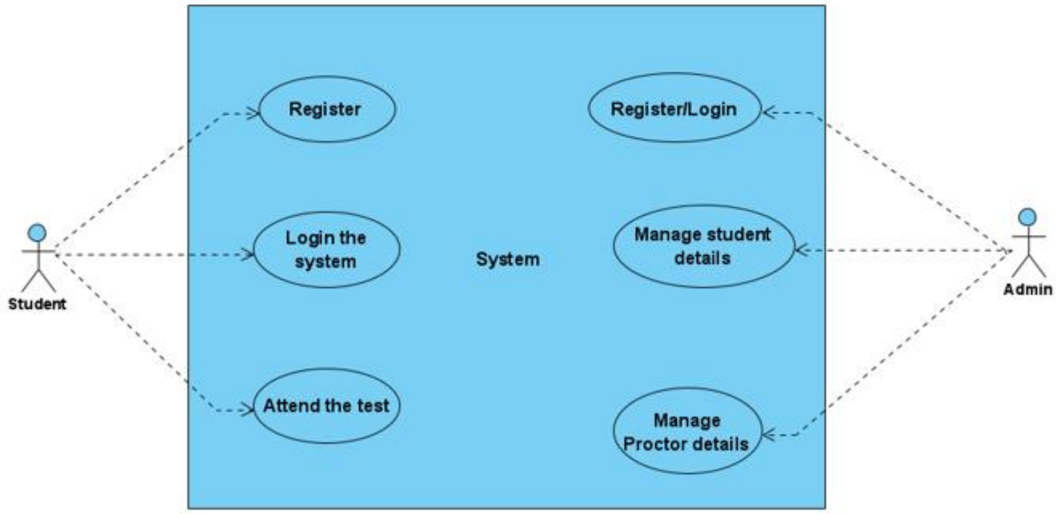


Figure. Use Case

6.3.5 Sequence Diagram:

Sequence diagram show a detailed flow for a specific use case or even just a part of it. It explains all the details of the objects in the sequence in which they are executed in a detailed level.
It has two dimensions

1. Vertical Dimension: It shows the sequence of messages in the order in which they occur.

2. Horizontal Dimension: It shows the object instances to which messages are sent.

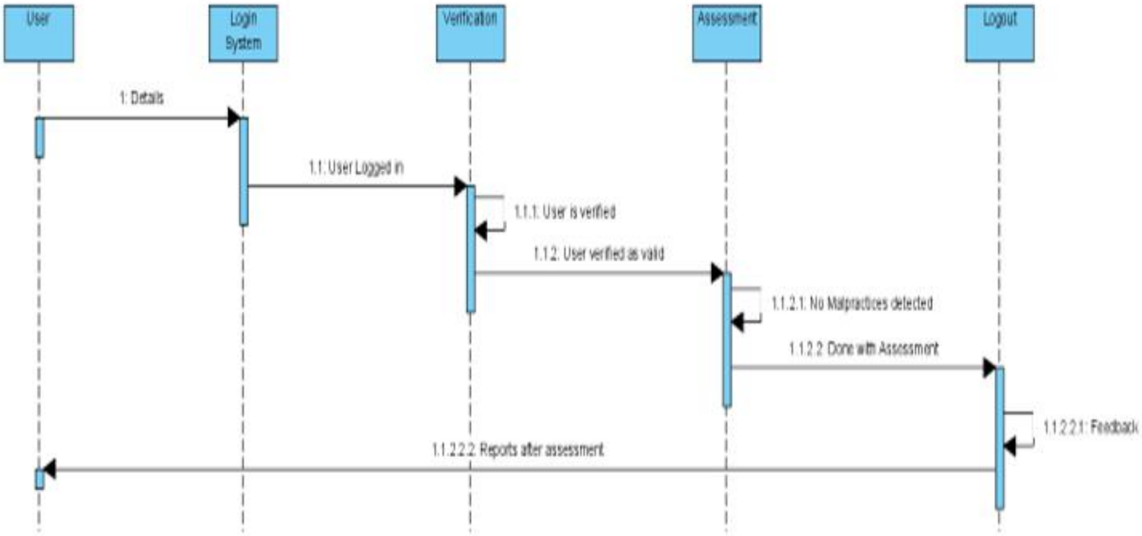

User
Login System
Verification
Assessment
Logout
1: Details
1.1: User Logged in
1.1.1: User is verified
1.1.2: User verified as valid
1.1.2.1: No Malpractices detected
1.1.2.2 Done with Assessment
1.1.2.2.1: Feedback
1.1.2.2.2: Reports after assessment

Figure. Sequence Diagram

6.3.6 State Diagram:

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It is a behavioural diagram and it represents the behaviour using finite state transitions**.**

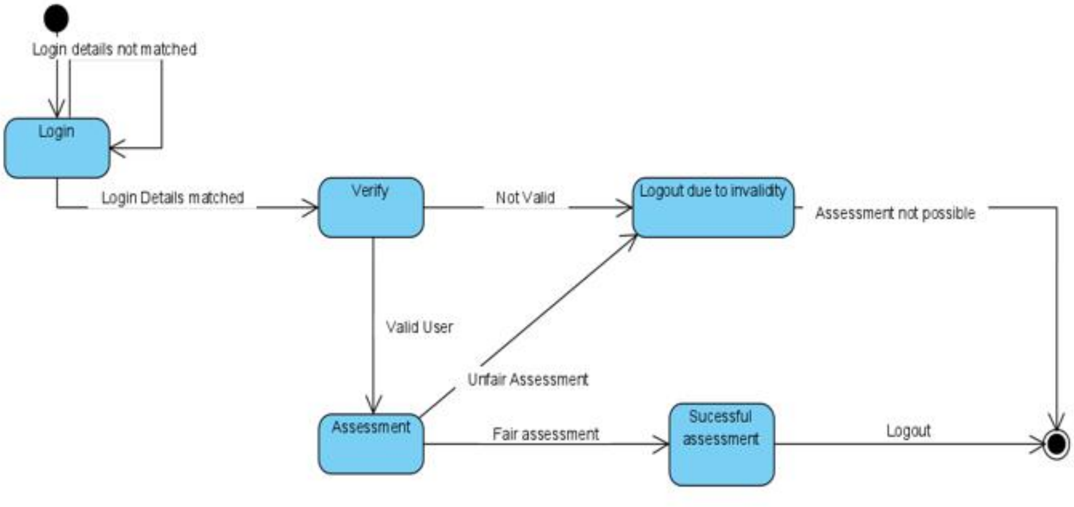


Figure. State Diagram

6.3.7 Deployment Diagram:

A deployment diagram models the physical deployment of artifacts on nodes. To describe a system a deployment diagram would show what hardware components exist, what software components run on each node, and how the different pieces are connected.

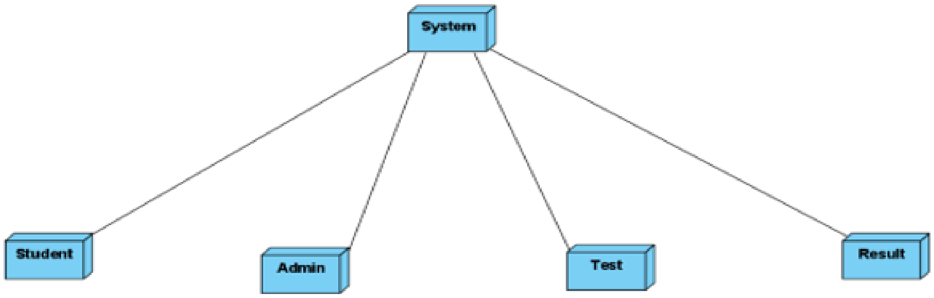


Figure. Deployment Diagram

6.3.8 Component Diagram:

A component diagram allows verification that a system's required functionality is acceptable. These diagrams are also used as a communication tool between the developer and stakeholders of the system. The diagrams formalize a roadmap for the implementation, allowing for better decision-making about task assignment or needed skill improvement.

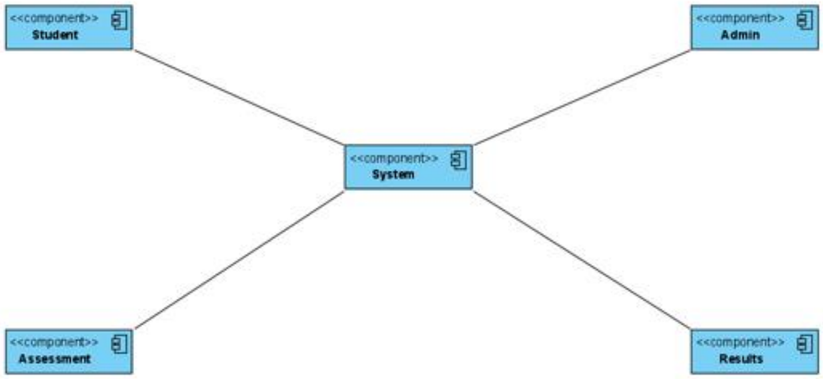


Figure. Component Diagram

Methodologies

The seven phases for Software Development are:

1.Requirement Analysis

2.Feasiblity Study

3.Design

4.Coding

5.Testing

6.Installation and Deployment

7.Maintenance

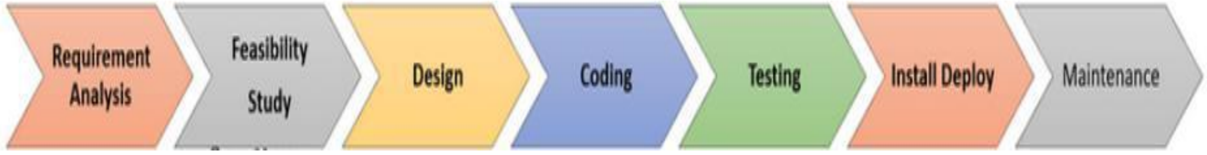


Figure. Agile Software Development

What is Agile?

Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment. It approaches development requirements and solutions through the collaborative effort of self-organizing and cross-functional teams and their customers/end users. It advocates adaptive planning, evolutionary development, early delivery and continual improvement and it encourages flexible responses to change.

Why agile?

It gives more capabilities than just Extreme Programming and Feature Driven Development. It widely uses the set of frameworks and practice based on the values in Manifesto for Agile Software development. By breaking down the project into manageable units, the project team can focus on high-quality development, testing, and collaboration. Agile helps project teams deal with many of the most common project pitfalls.

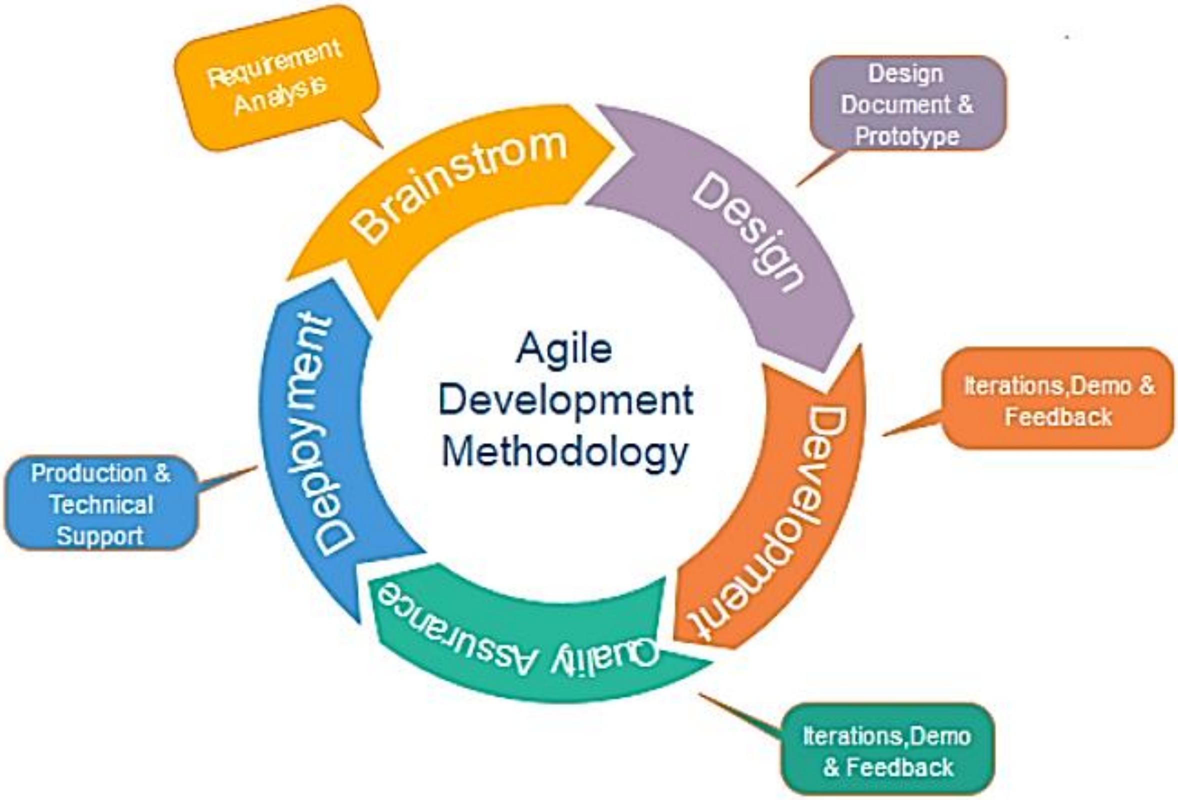
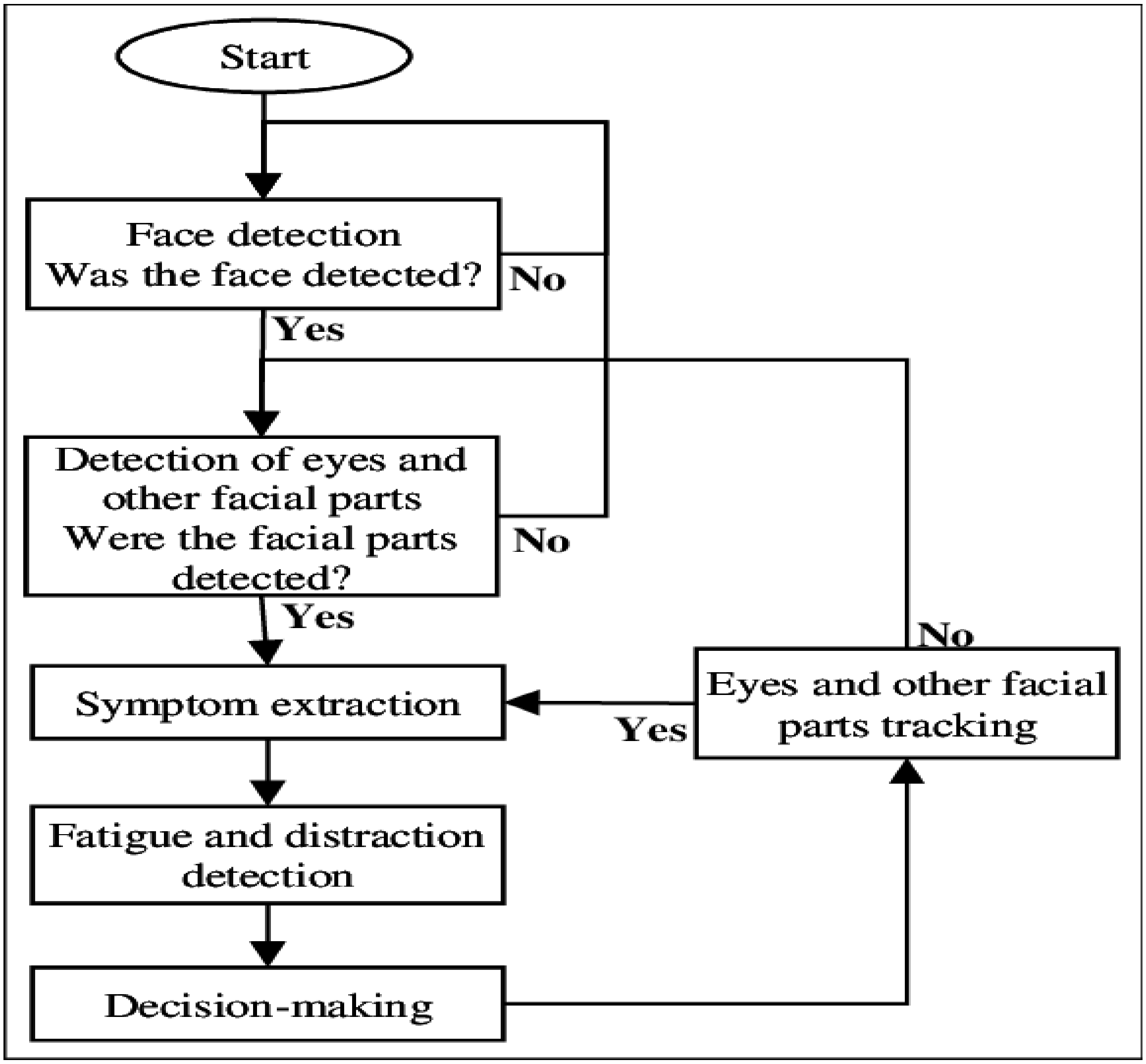

Requirement Analysis
Brainstrom
Design
Design Document & Prototype
Agile Development Methodology
Development
Iterations,Demo & Feedback
Quality Assurance
Iterations,Demo & Feedback
Deployment
Production & Technical Support

Figure. System Life-Cycle

Algorithms


Start
Face detection
Was the face detected?
No
Yes
Detection of eyes and other facial parts
Were the facial parts detected?
No
Yes
No
Symptom extraction
Yes
Eyes and other facial parts tracking
Fatigue and distraction detection
Decision-making

The system will have five vision-based capabilities which are combined using multithreading so that they can work together:

1.Eye detection and Gaze tracking:

Gaze tracking algorithm first detects the face of a user on the RGB images captured by the webcam based on the Viola- Jones algorithm. But because the user is far away from the webcam, the face image is understandably low resolution. Therefore, the proposed gaze tracking algorithm detects eyes on the face candidate images, instead of repeatedly detecting a face in reducing the expected size of a face. In addition, it improves the eye detection performance by expanding the size of the face image with interpolation. After the face and eye detection, it tracks them using

Kalman filter to reduce the processing time. Data collection by either a remote, head-mounted or VR based ‘eye tracker’ connected to a computer. Eye trackers generally include two common components: a light source and a camera. The light source (usually infrared) is directed toward the eye. We shall aim to track the eyeballs of the test-taker and report if he is looking to the left, right, or up which he might do to have a glance at a notebook or signal to someone.

This can be done using D-lib’s facial key-point detector and OpenCV for further image processing. Among the obstacles that eye tracking community encounters when facing the challenge of low-resolution gaze estimation, the lack of large-scale labelled datasets to be used for these purposes is remarkable. Ideally, datasets including images of the eye/face area are required where not only face but also eye area landmarks (eyelids, iris, pupil) are included. Moreover, images should be annotated with gaze information and, preferably, head pose should be also labelled. Many efforts have been made by researchers in order to generate large datasets containing the corresponding labels. However, although deep learning techniques proved to be successful in most areas of research, the accuracies obtained using these datasets in terms of gaze estimation are insufficient. One of the hypothesis is that the models do not learn to generalize because datasets employed for training purposes lack of enough variability. In order to enlarge these datasets size and trying to avoid the burdensome manual labelling option other possibilities have been proposed, such as image augmentation techniques or synthesizing images. Gaze tracking algorithm detects glints which are corneal reflections generated by the two NIR illuminators and utilized as a reference point for gaze position calculation.

It first detects bright regions in the IR image using adaptive threshold method. And then, it differentiates them between square regions and long rectangle regions where two glints are located closely. Especially in the case of square regions, it finds another square region connected from side to side, and then make the two regions into a long rectangle region. Next, it finds the centre of each long rectangle region using Gaussian blurring. The long rectangle region whose centre is closest to the pupil centre is decided as real glints. Gaze tracking algorithm also detects pupil centre. It first specifies pupil candidate regions of the foregoing bright regions using morphology dilation and pupil extraction filter that has a square shape and compares all the border pixels are brighter than the centre. And it verifies all the pupil candidate regions by checking the existence of the glints. Finally, it extracts the pupil centre using rough circle estimation based on sobel filter and weighted ellipse fitting which controls contour points before applying ellipse fitting especially
for excluding glints eyelid.

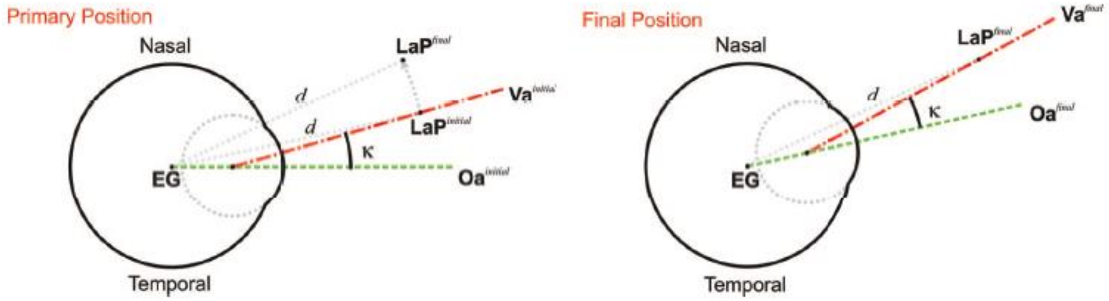


Figure. Primary Position of Gaze

In the upper part of the figure the primary position is shown together with the LaP in the final position. An imaginary axis is calculated in both, initial and final positions connecting EG to LaP$_{initial}$ and LaP$_{final}$ respectively. After estimating the rotation between the imaginary axis it is applied to the whole eyeball. In the lower part, both optical and visual axes have been rotated accordingly to the final position. Now, the visual axis points to the pursued LaP$_{final}$.

2.Facial Landmark Detection

One of the sub divided image frame makes one class i.e. the one consisting the faces in the image, which marks the first step towards the process of face detection. It is inconvenient because in spite of the congruity exist among faces but several factors like age, skin colour and facial expression can vary considerably. Then this problem is furthermore intricate by the arrival of factors like environment factors affecting light, risk of imitation and also

probability of limited obstruction in image. The face detection system that can easily recognize any face from a given image that too under any circumstance

with any kind of lighting environment is thus considered as the finest face detection system. The function of the face detection system can be further

bifurcated into two phases. Phase one consists of classification, in which the system based on the input that was in the form of some random images and if the face is present in the image the output comes in the form of yes or no. Face localization is the second phase in which for a given input image it shows a bounding box which comprise the dimensions of exact location of the face in the image.

The process of face detection system is sub-divided as follows:

1) **Pre-Processing:** Before feeding any image to the network it is processed properly to lower down the variability. Frontal faces that are comprised in the front view of the image is thus obtained by cropping the images that contain the human faces. On completion of the above step, standard algorithms are used to correct the lighting of the cropped images.

2) **Classification:** To categorize any image as faces or non faces, neural networks are implemented by training on these examples. For the process of

classification, we have combined the MATLAB NN toolbox along with the basic implementation of the neural networks.

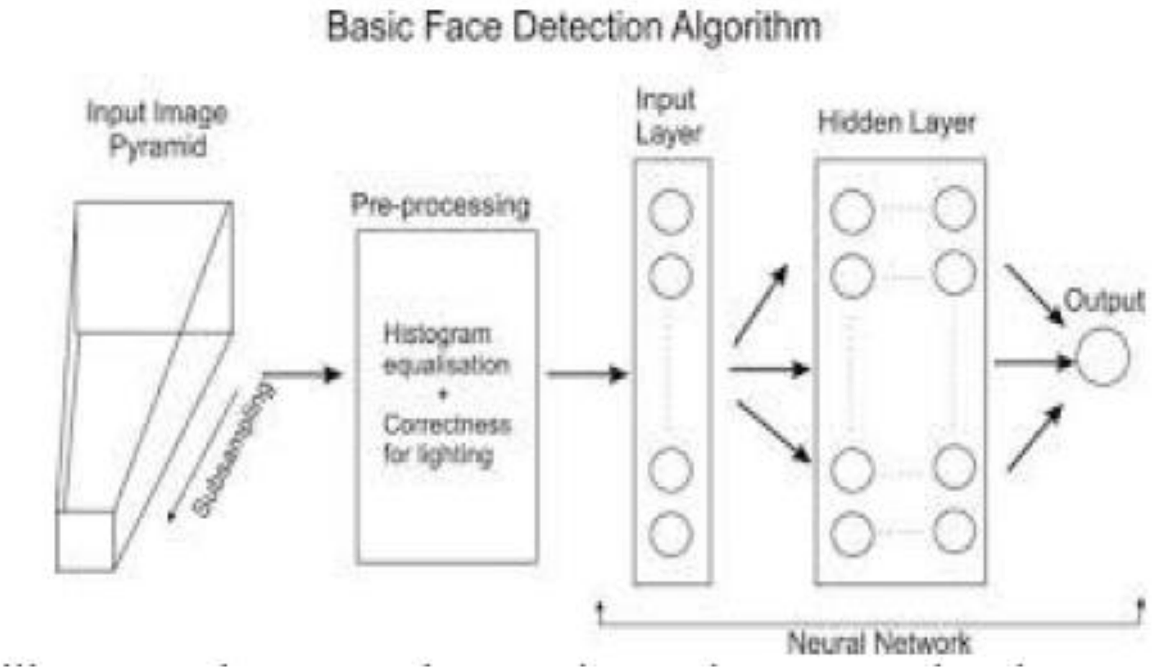


Figure. Face Detection Algorithm Architecture

The facial landmark detector which is pre-trained inside the d-lib library of python for detecting landmarks, is used to estimate the location of 68 points or (x, y) coordinates which map to the facial structures. These indexes of 68 coordinates or points can be easily visualized on the image.

The Locations of the Facial Parts are shown in table:

Sr. No.	Facial Part	(x Co-ordinate)	(Y Co-ordinate)
1	Left Eye	42	48
2	Mouth	49	67
3	Left Eyebrow	22	26
4	Nose	27	34
5	Right Eye	17	21
6	Right Eyebrow	36	41
7	Jaw	0	16

Table. Location of Facial Parts

The first part of proctoring is face detection. To evaluate face detection in our system we first use face video dataset to train our system. After detecting face the next step is to track face and ensure continuous presence of student in the exam. Since there are a lot of face movements of face during online exam we use dataset for evaluating face features. The following table gives set of rules that can be used for evaluating face tracking.

Sr. No.	Rules	Decision
1	Detected any face missing from frame at any point of duration	Malpractice
2	Face moving far away from webcam more than 2 times	Warning
3	Face moving far away from webcam more than 4 times	Malpractice
4	Multiple face detected	Malpractice
5	Sound more than mean amplitude for more than 1 time	Warning
6	Sound more than mean amplitude for more than 3 times	Malpractice
7	Any window active other than browser	Malpractice

Table: Inference system rules to classify actions by student

3.Mouth Detection

- This is very similar to eye detection. D lib’s facial key-points are again used for this task and the test-taker is required to sit straight (as he would in the test) and the distance between the lips key-points (5 outer pairs and 3 inner pairs) is noted for 100 frames and averaged.
- If the user opens his/her mouth the distances between the points increases and 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.
- The mouth can be accessed through points [49, 68].

4.Person Counting and Mobile Phone Detection

In this, we are focusing on the problem of detecting each instance of a specific category of person. A new technique use for person detection is proposed based on a deep counting model. The feature extractor in YOLOv3 of the deep counting model is extended with additional layers for segmenting specific instances and also focus on the persons in the scene. The segmentation layers help to get a more accurate estimation of the foreground with persons. The instance segmentation is able to estimate separate instances of persons.

YOLOv3 (You Only Look Once, version 3) will perform multiple image analysis tasks as person counting, person segmentation and instance segmentation. A deep counting model have features extensions, use to obtain a comprising small number of layers to multi task network which helps to achieves segmentation combination with tasks. By using frame-work we obtain the output for segmentation maps and semantic boundaries to use in post processing steps. YOLO results in the feature extractor that useful for analyse applications as shown in figure. Figure shows the feature extractor of YOLO Algorithm trained for counting persons, able to focus on persons in the different scenarios.

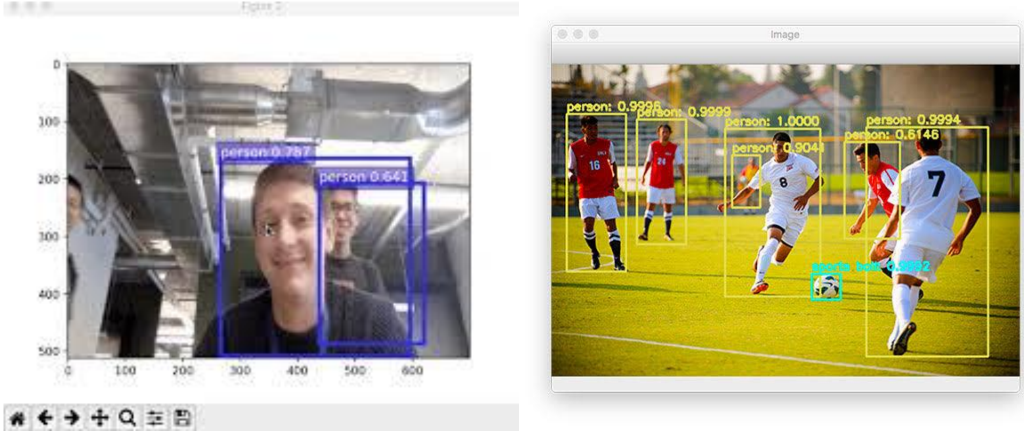


Fig. Object Detection in Instance Segmentation

The architecture of a comprises of a common feature extractor that followed by three layers that are, 1) A specific layer for object counting 2) A second layer for segmentation, and 3) A third for instance separation.

The feature extractor of the deep counting model is able to focus on the foreground region. We extending it to few layers to achieve the segmentation and instance separation instead of a full-fledged decoder comprising of a series of de-convolution and un-pooling layers. This helps to reduce the number of computations during inference. The probabilistic segmentation map drives the head for instance segmentation. The instance separator function is to create an embedding in a feature space such as the pixel representation corresponding to the same instances that are close together and others are separated. Due to the number of instances in each frame vary and no clear discriminant, Simple classifier or cross entropy loss is not possible to achieve instance segmentation directly. Also, the ordering of the instances is not important. The task of instance segmentation is broken down into background removal, foreground localization and distance separation that followed by post-processing steps for instance segmentation.

Instead of using separate networks for each of these sub-tasks, using a combined network that comprises of a common feature extractor is beneficial. The different components of the network are synergistic and assist each other during the training resulting in quicker convergence.

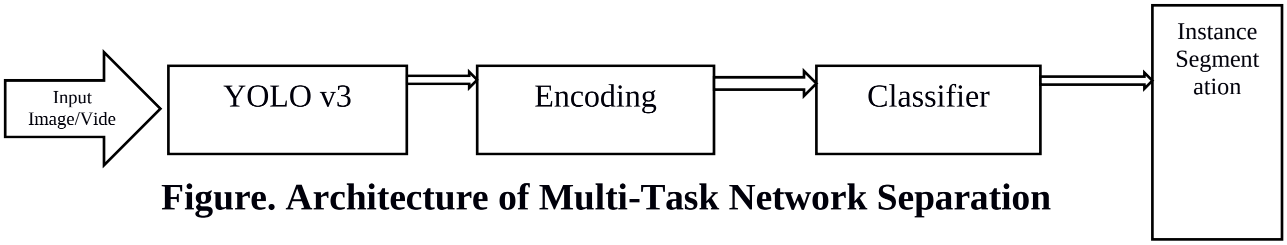


Figure. Architecture of Multi-Task Network Separation

During inference, the trained model can describe the prediction of the count of instances and the instance separated output embedded in an n-dimensional space. To obtain the instance segmentation, clustering techniques like mean shift algorithm or K-means can be used in which the predicted count can be used to initialize the number of clusters. A deep network with few layers that has been trained for obtaining instance segmentation from the embedded space can be used. The predicted object count will be use to select the appropriate network.

5.Audio to Text

The idea is to record audio from the microphone and convert it to text using Google's speech recognition API. The API needs a continuous voice from the microphone which is not plausible so the audio is recorded in chunks such there is no compulsory space requirement in using this method (a ten-second wave file had a size of 1.5 Mb so a three-hour exam should have roughly 1.6 Gb). A different thread is used to call the API so that a continuous recording can without interruptions, and the API processes the last one stored, appends its data to a text file, and then deletes it to save space.

Speech to text has three main methods to perform speech recognition:

1.Synchronous recognition send audio data to the speech to text API, performs recognition on the data and results after all audio has been processed. Synchronous recognition requests are limited to audio data of 1 minute or less duration.

2.Asynchronous recognition sends audio data to speech to text API and initiates a long running operation. Using this operation, you can periodically poll for recognition results. Use asynchronous results for audio data of any duration up to 480 minutes.

3.Streaming Recognition performs recognition on audio data provided within a gPRC bi-directional stream. Streaming requests are designed for real time recognition purposes, such as capturing live audio from a microphone. Streaming recognition provides interim results while audio is being captured, allowing result to appear. Requests contain configuration parameters as well as audio data.

SPEECH TO TEXT API RECOGNITION:

A speech to text API recognition request is the simplest method for performing recognition on speech to audio data. Speech to text can process up to 1 minute of speech data audio sent in a synchronous system. After speech-to-text processes and recognizes all of the audio, it returns a response.

A synchronous request is blocking, meaning that Speech to Text must return a response before processing the next request. Speech to Text typically processes audio faster than real-time, processing 30 seconds of audio in 15 seconds on average. In cases of poor audio quality, your recognition request can take significantly longer.

Synchronous speech recognition requests

A sample speech to text API request:

```
{
        “config”: {
                  “encoding” : ”LINEAR16”,
                “sampleRateHertz”:16000,
                “languageCode”: ”en-US”,
                },
                “audio”: {
                “uri” : ”gs://bucket-name/path/path-to-audio-file”
                }
}
```

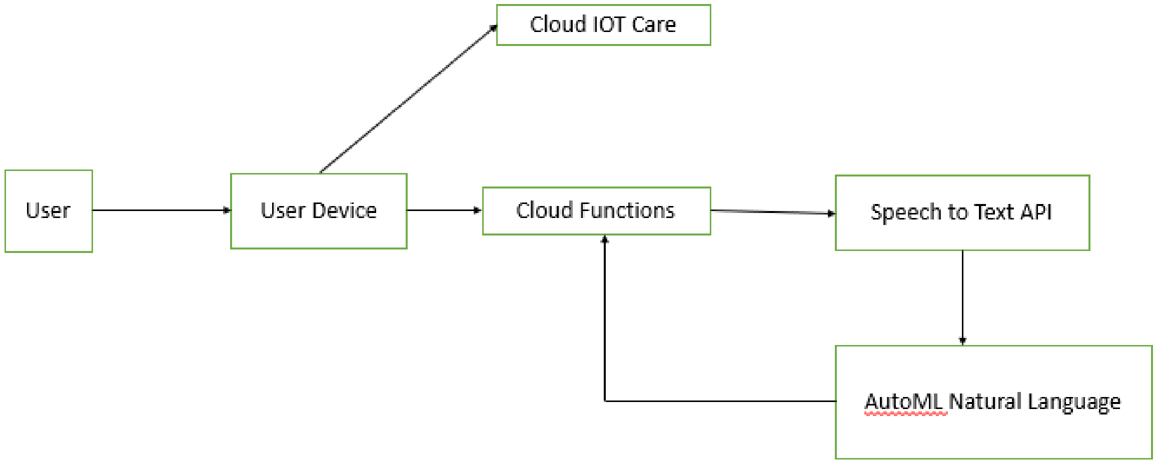


Figure. Architecture for Speech to Text Converter

Conclusions

Proposed system gives enough idea about design of organizing online test using web-based application and monitor the user throughout the session using different deep learning and computer vision algorithms and libraries. Using Face detection, system can easily check whether user is real or not, also it can check whether one or more people present in the frame. Using eye detector or gaze tracking, we can check that whether the user looking outside the screen or not. Speech to text conversion will check if person is speaking with somebody. Conducting exams online is very important in current time. It enables user to give the test at any time and from anywhere and reduces transportation as well. This will not only save time and cost but it will also create transparency in the examination process.

References

[1] Y. Atoum, L. Chen, Alex X. Liu, Stephen D. H. Hsu, and X.Liu, “Automated Online Proctoring”, IEEE Transaction on Multimedia, 2015.

[2] Asep Hadian S. G., Yoanes Bandung, “A Design of Continuous User Verification for Online Exam Proctoring on M-Learning”,2019 International Conference on Electrical Engineering and Informatics (ICEEI)

[3] Sanjukta Ghosh, Peter Amon, Andreas Hutter, André Kaup, “DEEP COUNTING MODEL EXTENSIONS WITH SEGMENTATION FOR PERSON DETECTION”, ICASSP 2019

[4] Swathi Prathish, Athi Narayanan S, Kamal Bijlani, “An Intelligent System For Online Exam Monitoring”, 2016 IEEE

[5] Dweepna Garg. Parth Goel, Sharnil Pandya, Amit Ganatra, Ketan Kotecha, “A Deep Learning Approach for Face Detection using YOLO”, ©20XX IEEE

[6] Vishnu R R, Athi Narayanan S, kamalbijlani, “Heuristiv-based Automatic online proctoring system”, 2015 IEEE 15th International Conference on Advanced Learning Technologies

[7] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems 25, pp. 1097–1105. 2012.

[9] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, Cambridge, MA, USA, 2015, NIPS’15, pp. 91–99, MIT Press.

[10] I. E. Allen and J. Seaman. Grade change: Tracking online education in the united states, 2013. Babson Survey Research Group and Quahog Research Group, LLC. Retrieved on, 3(5), 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904–1916, 2015.

[12] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” CoRR, vol. abs/1506.02640, 2015.

[13] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481–2495, 2017.

[14] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440.

[15] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” in 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[17] C. Szegedy, A. Toshev, and D. Erhan, “Deep neural networks for object detection,” Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. Curran Associates Inc., pp. 2553–2561, 2013.

[18] A. Trivedi, A Relevant Online Examination System, Proceedings of the 1st IEEE International Conference on Technology for Education (IEEE T4E), Mumbai, India, pp. 32-35, July 2010