A PROJECT REPORT ON

# "ONLINE EXAM PROCTORING"

SUBMITTED TO

SHIVAJI UNIVERSITY, KOLHAPUR

IN THE PARTIAL FULFILLMENT OF REQUIREMENT
FOR THE AWARD OF DEGREE

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY


| | |
|---|---|
| Mr. Koustubh Basavraj Wale | 17UCS12026XX |
| Mr. Shrenik Subhash Kusanale | 17UCS52028XX |
| Mr. Ganesh Sunil Mudhale | 17UCS12034XX |
| Mr. Nikhil Suresh Pandav | 17UCS12038XX |
| Mr. Sanket Sudesh Kamble | 18UCS72005XX |

UNDER THE GUIDANCE OF

Prof. A. B. Majgave

DKTE
Promoting Excellence in
Teaching, Learning & Research

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
D.K.T.E. SOCIETY'S TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI
(An Autonomous Institute, Affiliated to Shivaji University, Kolhapur)
Accredited with 'A+' Grade by NAAC, An ISO 9001: 2015 Certified


YEAR 2020-2021

D.K.T.E. SOCIETY'S

TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI

(An Autonomous Institute, Affiliated to Shivaji University, Kolhapur)

Accredited with 'A+' Grade by NAAC, An ISO 9001: 2015 Certified

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DKTE

Promoting Excellence in
Teaching, Learning & Research

# CERTIFICATE

**This is to certify that, project work entitled**

## "ONLINE EXAM PROCTORING"

**is a bonafide record of project work carried out by**

| | |
|---|---|
| **Mr. Koustubh Basavraj Wale** | **17UCS12026XX** |
| **Mr. Shrenik Subhash Kusanale** | **17UCS52028XX** |
| **Mr. Ganesh Sunil Mudhale** | **17UCS12034XX** |
| **Mr. Nikhil Suresh Pandav** | **17UCS12038XX** |
| **Mr. Sanket Sudesh Kamble** | **18UCS72005XX** |

**In the partial fulfillment of award of degree, Bachelor of Technology in Computer Science and Engineering prescribed by Shivaji University, Kolhapur for the academic year 2020-2021.**

**Prof. A. B. Majgave**
**(PROJECT GUIDE)**

**Prof. (Dr.) D.V. KODAVADE**
**(H.O.D. C.S.E.)**

**Prof.(Dr.) P.V.KADOLE**
**(DIRECTOR)**

**EXAMINER**

# DECLARATION

**We hereby declare that, the project work report entitled "ONLINE EXAM PROCTORING" which is being submitted to D.K.T.E. Society's Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.TECH.(CSE). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy / Cyber / IPR Act amended from time to time.**

| PRN | Name | Signature |
| --- | --- | --- |
| 17UCS12026XX | Mr. Wale Koustubh Basavraj | |
| 17UCS52028XX | Mr. Kusanale Shrenik Subhash | |
| 17UCS12034XX | Mr. Mudhale Ganesh Sunil | |
| 17UCS12038XX | Mr. Pandav Nikhil Suresh | |
| 18UCS72005XX | Mr. Kamble Sanket Sudesh | |

# ACKNOWLEDGEMENT

**With great pleasure we wish to express our deep sense of gratitude to Mr. A. B. Majgave for his valuable guidance, support and encouragement in completion of this project report.**

**Also, we would like to take opportunity to thank our head of department Dr. D. V. Kodavade for his co-operation in preparing this project report.**

**We feel gratified to record our cordial thanks to other staff members of Computer Science and Engineering Department for their support, help and assistance which they extended as and when required.**

**Thank you,**

| | |
|---|---|
| **17UCS12026XX** | **Mr. Wale Koustubh Basavraj** |
| **17UCS52028XX** | **Mr. Kusanale Shrenik Subhash** |
| **17UCS12034XX** | **Mr. Mudhale Ganesh Sunil** |
| **17UCS12038XX** | **Mr. Pandav Nikhil Suresh** |
| **18UCS72005XX** | **Mr. Kamble Sanket Sudesh** |

# ABSTRACT

This Online Examination Proctoring is a software solution, which allows any industry or institute to arrange, conduct and manage examinations via an online environment. It can be done through the Internet/Intranet and/ Local Area Network environments. Some of the problems faced during manual examination systems are the delays occurred in result processing, filing poses a problem, filtering of records is difficult. The chance of loss of records is high also record searching is difficult. Maintenance of the system is also very difficult and takes lot of time and effort. Online examination is one of the crucial parts for online education system. It is efficient, fast enough and reduces the large amount of material resource. An examination system is developed based on the web. This paper describes the principle of the system, presents the main functions of the system, analyzes the auto-generating test paper algorithm, and discusses the security of the system

# INDEX

# Introduction

In recent decades, information and communication technologies have become increasingly important in human life, especially in education. Traditional exam is paper-based and requires the presence of a supervisor on the institution's premises and forces him to walk around tables and keep an eye on any cheating. 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. This task can be automated by building a framework where teachers only receive a list of students that have shown suspicious behavior at certain time stamps. This framework could use different classification methods to detect a student cheating.

# Problem Description

## Problem definition:

Remote proctoring enables students to write a test online in a remote location, while maintaining the integrity of the test. Students must confirm their identity and they may be monitored through photos captured during exam. These photos are then used to flag any irregular student behavior.

Online Testing has been around for last 20 years in different formats. The most common type of online test is an objective test which tests a candidate on their subject knowledge or is used to understand his or her learning ability or behavioral profile.

## Aim of the project:

- This project aims to design, develop and implement virtual proctoring software. This helps to control the fraudulent activity of the student. An attempt to go even further to detect fraud in digital exams has been done by a tool known as Proctor Exam. This tool features which records the entire monitor of the student during the exam.
- The intended audience for this system is the students who appear for online examination

## Objectives of the project:

- To calculate suspicious activity percentage of a student who is trying to perform malpractices during exam.
- To capture the images of the student periodically while he/she giving the exam.
- To recognize the face of the student comparing with its reference images.
- To perform object detection as well as multi-faces detection from captured images.

## Scope and limitations of the project

### Scope:

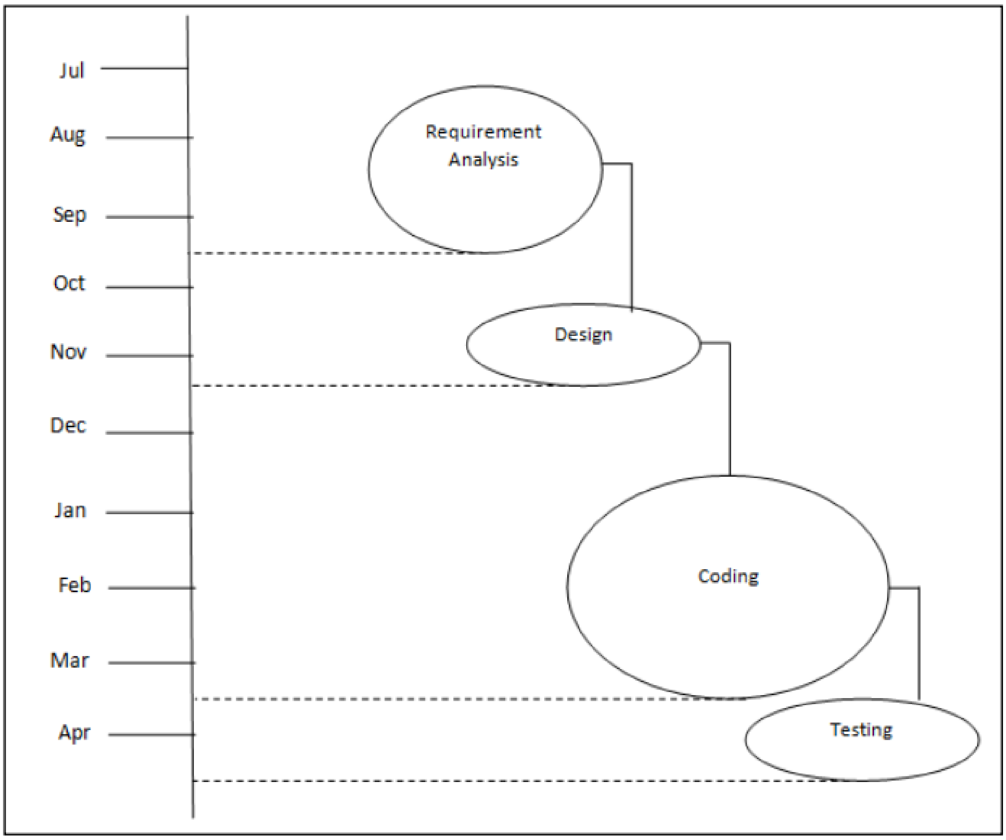
- The software called Digital Exam Proctor is used to monitor the students during examination.
- The software product will result the suspicious activity percentage of each student.
- This software can be used at many levels including universities, schools and at corporate level to control the student's fraud behavior.

### Limitations:

- Camera and Microphone is necessary
- Strong internet connection required.
- Candidate should be in a front of camera till the end of exam. No breaks are allowed.

## Timeline of the project:

We started the project by gathering the related documents to the project at the end of July 2020. Gathering the requirements and all the analysis tasks was done by mid of August 2020. After that System design was started in the month of September 2020 and completed by the start of November along with the UML diagrams and Synopsis with rough idea of the project. In November 2020 we started making the detailed SRS documents along with deciding the methodology for the project which was completed by mid-December 2020. By the start of January 2021, we started coding by dividing it into 3 modules and completed the 1st module by mid-January 2021. Other two modules were completed by the end of February 2021. By the end of March 2021, we started testing the project alongside designing the GUI which was completed in the first week of April 2021.



# Background study and literature overview:

- **Literature overview:**

Online examination system is one of the method of taking exams which is doesn't require any kind of a piece of paper or a pen.It is the fast growing method to take exams over online.Speed and accuracy is the reason behind the famous of this method because speed and accuracy is the backbone of this system.Many researchers have already researched about online examination system and we have developed a online examination system to keep an eye on this researches as a reference and these all are the following:

- Zhenming et al (2003):They developed an online examination system based on web browser/server framework.

- SIETTE Guzman and Cenejo (2005):They developed a online examination system called as SIETTE; (System of intelligent Evaluation using Tests for Tele education).

- Ayo at a (2007):They proposed a model of e-examination.The reason behind the developed such as software is to conduct the entrance examination for all Nigeria universities called JAMB(Joint Admission Matriculation Board).This software was designed and tested in Covenant university they were the private university in Nigeria. They found the software really helpful for conducting neat and clean with accuracy entrance examination. It is eliminates the problems that are associated with the traditional methods of entrance examination.

- Jim and seen (2006):They justified the assessment can be taken in different ways. First of all they made a e-assessment via internet and then the mindset has been done that we can enhance the e-assessment to a online examination system. And they thought that there will be a many different ways to take the assessment and examination.They continuously added the content and they took the shape of e-examination portal finally.

- Zhenming et al (2003):They developed an online examination system based on web browser/server framework. Which supports some premium basic features ,carriers out the examination and provide the auto grading system for objective questions and operating questions like programming, edit MS word,Power point,MS windows, Excel etc

- **Yousef Atoum, Liping Chen, Alex X. Liu, Stephen D. H. Hsu, Xiaoming Liu, "Automated Exam proctoring", [2017].**

  This IEEE paper was published in 2017 and proposes a solution to online exam proctoring [1]. This paper presents a multimedia analytic system that performs automatic exam proctoring. The system hardware includes one webcam, and a microphone, for the purpose of monitoring the visual and acoustic environment of the testing location. The system includes six basic components that continuously estimate the key behavior cues: user verification, text detection, voice detection, active window detection, gaze estimation and phone detection. By combining the continuous estimation components, and applying a temporal sliding window, they have designed higher level features to classify whether the test taker is cheating at any moment during the exam.

- **Asep, H. S. G., & Bandung, "A Design of Continuous User Verification for Online Exam Proctoring on M-Learning", [2019].**

  This IEEE paper was published in 2019 and proposes a solution to online exam proctoring [2]. In this paper, they have proposed a method to enhance the robustness for pose and lighting variations by doing an incremental training process using the training data set obtained from machine learning online lecture sessions. As a result, the design of the proposed method is presented in this paper.

- **Swathi Prathish, Athi Narayanan, S Kamal Bijlani, "An Intelligent System for Online Exam Monitoring", [2016].**

  This IEEE paper was published in 2016 and proposes a solution to online exam proctoring [3]. In this paper, they have presented a method to avoid the physical presence of a proctor throughout the exam by creating a comprehensive multi modal system. They have used hardware such as webcam to capture audio and video along with active window capture. This combination forms the inputs to an intelligent rule-based inference system which has the capability to decide whether any malpractices have happened. Examinee's face is detected and is used to extract feature points thereby estimating a head pose. Misconduct is detected based on yaw angle variations, audio presence and active window capture. This system has been tested in an e-learning scenario and we were able to make exam monitoring easy.

## ➢ Investigation of current project and related work

For the past few years, e-learning has become popular across countries because of its flexibility, availability and user friendliness. As far as online examinations are concerned, the major challenge faced by the research community is the proctoring techniques used. But Machine learning has considerable success in this area. The use of m-learning or other remote education continues to increase due to its ability to reach people who don't have access to campus. In an exam, a proctoring method to detect and reduce the cheating possibility is very important to ensure that the students have learned the material given. Various methods had been proposed to provide an efficient, comfortable online exam proctoring. Start with implementing an exam design with hard constraints in a no proctoring exam, a remote proctoring using a webcam, a machine-based proctoring and finally research on automated online proctoring. A visual verification for the whole exam session is needed in an online exam, therefore a face verification is needed. Along with face recognition, object detection as well as multiple faces detection is also needed to catch the malpractices conducted by an examinee. For this various powerful ML libraries like OpenCV, MTCNN are available.

# Requirement analysis

## Functional Requirements

- The system shall capture images of the student throughout the exam according to the algorithm.
- For processing the input, the system shall compare captured images to the reference image of student based upon four parameters mentioned earlier.
- The system shall compare each captured image to the reference image. If the image doesn't match to reference image then it will be considered as suspicious activity and every suspicious image will be saved by the system in order to process the output.
- The system shall provide the output in terms of suspicious activity percentage, i.e. for every suspicious image this percentage will increase.
- As there are four parameters for the comparison, there will be percentage for each parameter. Each suspicious image will be classified according to these parameters.
- The system shall provide all four suspicious activity percentage and each suspicious image in the form of profile of that particular student as final output.

## Performance Requirements

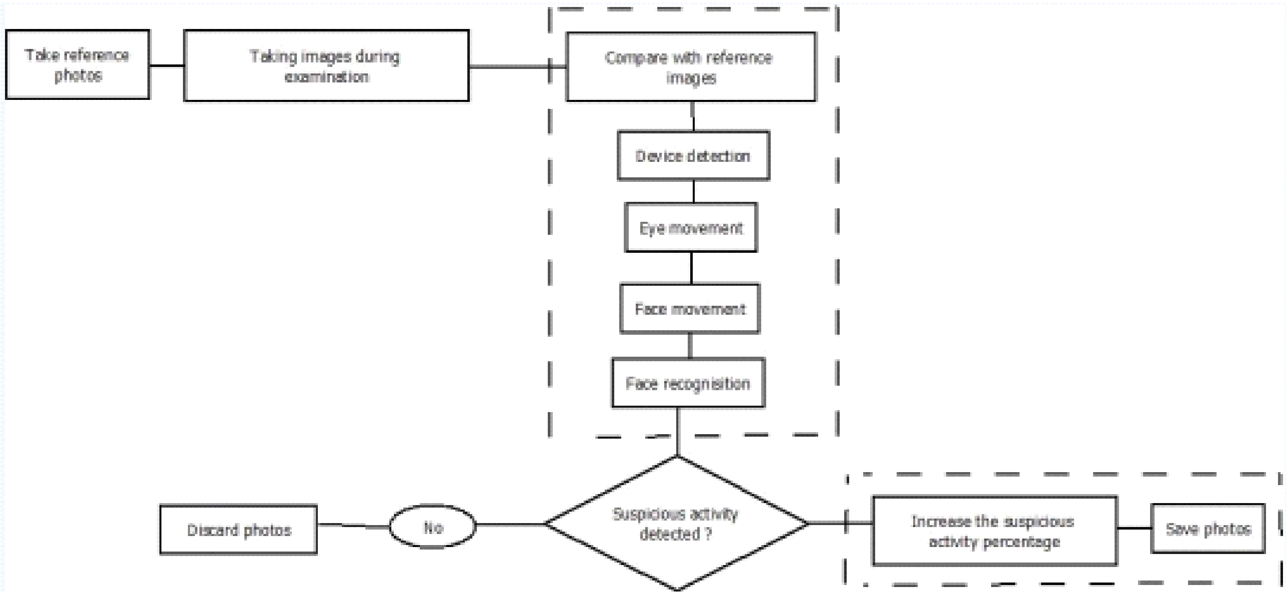
- Static numerical requirements

  a) This software system supports maximum two simultaneous users.

  b) The amount of information to be handled by the system is appropriately 400-500 Megabytes and this information will be in the form of images.

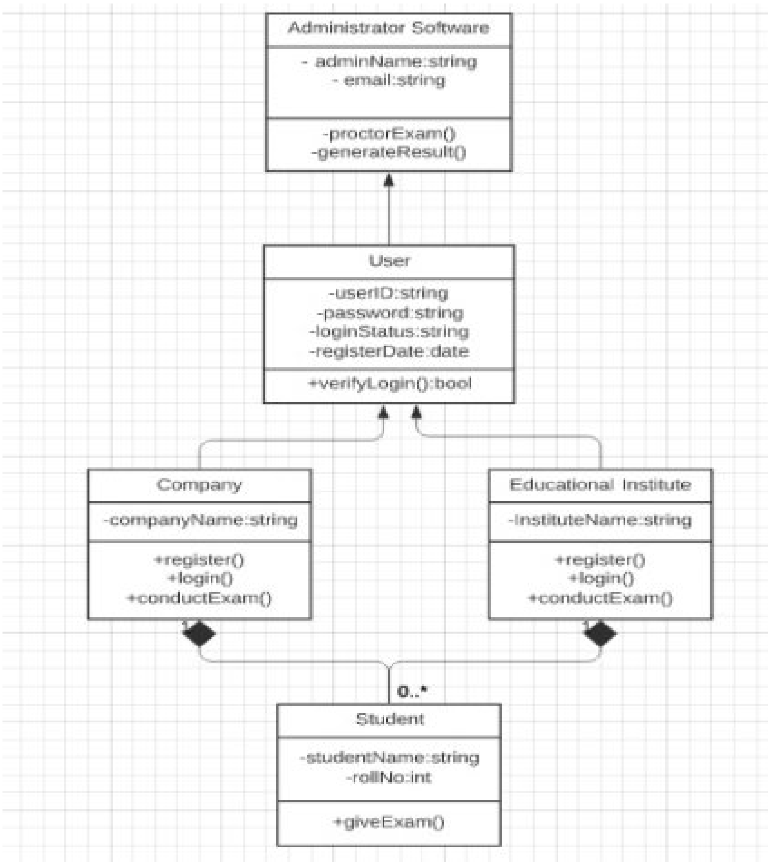
# System Design

## A. Architectural Design

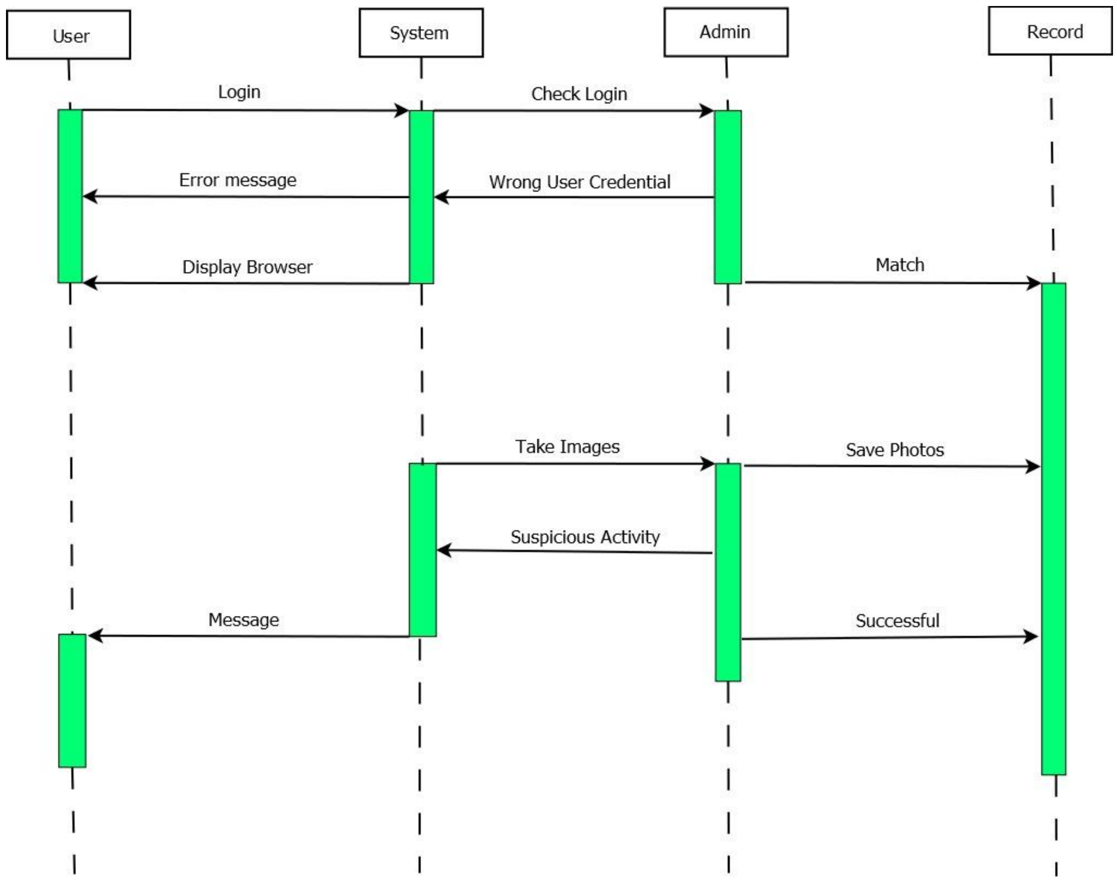


## B. Data Design

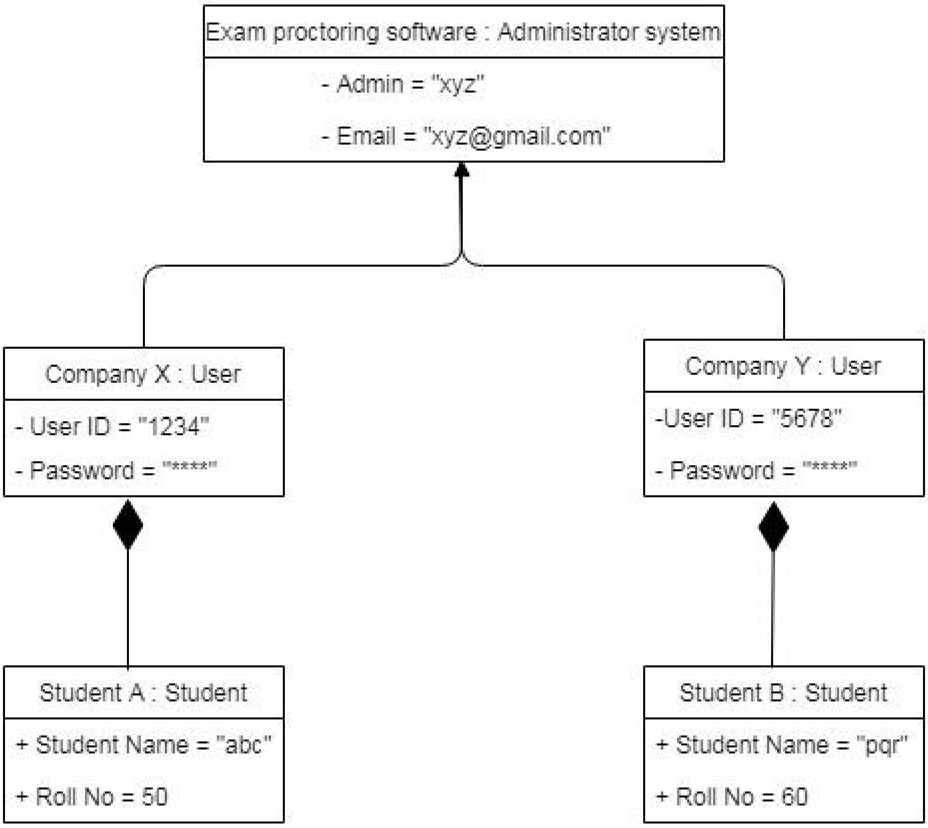
## a. Class Diagram



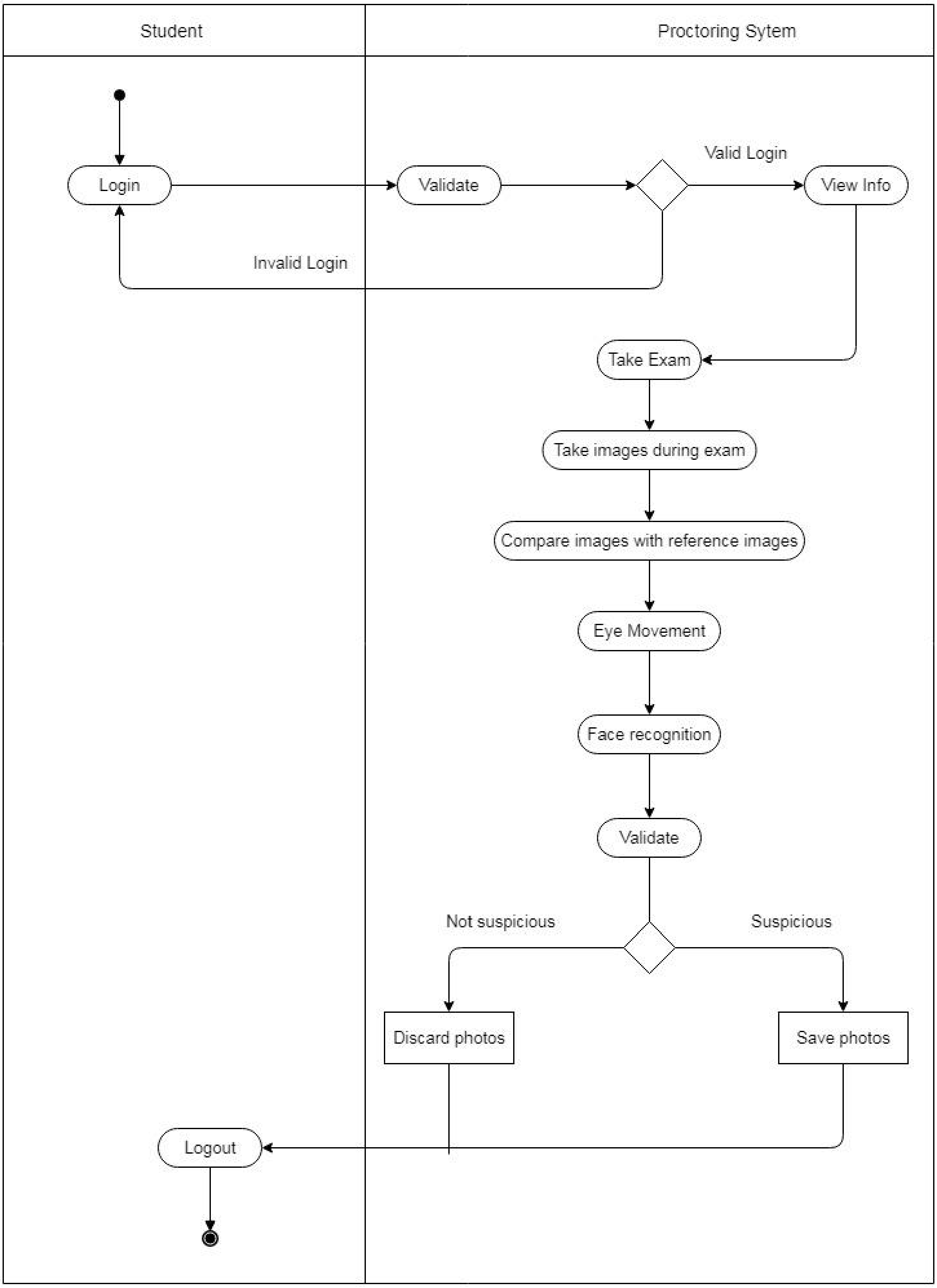
## b. Sequence Diagram



## c. Object Diagram



## d. Activity Diagram



# Implementation

- **Detailed Description of Method**

- ❖ Before user login into the system, he/she strictly maintain following guidelines:

The computer you are using to take the exam must not have more than one display or monitor. For example, if you usually use a laptop with a monitor connected, disconnect your monitor and use only the laptop screen.

You must close all other programs or windows on your testing computer before you begin the exam.

Examinations can be taken on devices such as Laptop/ Smart-Phone/ Tablet/Desktop. Charge the Laptop / Smart Phone / Tablet well in advance to last for at least 2 hours.

The device should have continuous internet connectivity.

Position the device in such a way that the front camera captures your face properly and you can sit for ONE HOUR to take up the examination conveniently without moving the device.

If your device is using a Wi-Fi router, make sure to sit near the Wi-Fi Router/Modem to avoid any signal related issues.

- ❖ Log in to the System with provided Credentials

To enter into the system first we have to enter our login credentials correctly as there is no other option to start the exam other than provided credentials.

The credentials are unique for every user including both username and password. There is no forgot password option as user already made the system check operation before one day of examination. If user get any problem during system check he/she can directly contact to system admin. Before user login into the system he/she strictly maintain following guidelines:

The computer you are using to take the exam must not have more than one display or monitor. For example, if you usually use a laptop with a monitor connected, disconnect your monitor and use only the laptop screen.

You must close all other programs or windows on your testing computer before you begin the exam.

Examinations can be taken on devices such as Laptop/ Smart-Phone/ Tablet/Desktop. Charge the Laptop / Smart Phone / Tablet well in advance to last for at least 2 hours.

The device should have continuous internet connectivity.

Position the device in such a way that the front camera captures your face properly and you can sit for ONE HOUR to take up the examination conveniently without moving the device.

If your device is using a Wi-Fi router, make sure to sit near the Wi-Fi Router/Modem to avoid any signal related issues.

❖ Log in to the System with provided Credentials

To enter into the system first we have to enter our login credentials correctly as there is no other option to start the exam other than provided credentials.

The credentials are unique for every user including both username and password. There is no forgot password option as user already made the system check operation before one day of examination. If user get any problem during system check he/she can directly contact to system admin

❖ Taking Reference Images

Once a user logs into the system with Username and Password, it will take you to the page to take a photograph with your details. Please allow camera access, and audio device access when prompted. If you do not give access to any of these, you will not be able to appear for the examination or the remote proctor may disable your examination.

1.The lighting in the room must be bright enough to be considered “daylight” quality. Overhead lighting is preferred.

2.You must sit at a clean desk or table.

3.The walls around you must not have any writing or hanging objects as algorithm detects objects in the images.

4. The reference photo must be bright and clear enough to recognize as a face.

❖ Remote Proctoring

The students appearing for terminal semester examination shall be remotely proctored throughout the examination. Login credentials such as login ID and password for students will be provided through registered e-mail ID / SMS to the registered mobile phone.

Rules for Online Examination:

If a student is violating any rules during the examination or trying to adopt any unfair means, the system will automatically collect data based on the following deviations and mark him/her as suspicious.

Face is not visible in the camera: Student is not looking into the camera.

Several faces in front of the camera: There are other people along with the examination taker.

Face does not match the profile: Student taking the examination is not the same person whose photo was captured before starting the.

Object detection: During examination ensure no objects / obstacles are behind you because proctor model is trained to detect objects.

## Methodology

### ➢ Face Recognition using LBPH algorithm

Face Recognition: with the facial images already extracted, cropped, re-sized and usually converted to grayscale, the face recognition algorithm is responsible for finding characteristics which best describe the image.

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

Using the LBP combined with histograms we can represent the face images with a simple data vector.

As LBP is a visual descriptor it can also be used for face recognition tasks, as can be seen in the following step-by-step explanation.

Step-by-Step

Now that we know a little more about face recognition and the LBPH, let's go further and see the steps of the algorithm:

Parameters: the LBPH uses 4 parameters:

Radius: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.

Neighbors: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.

Grid X: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

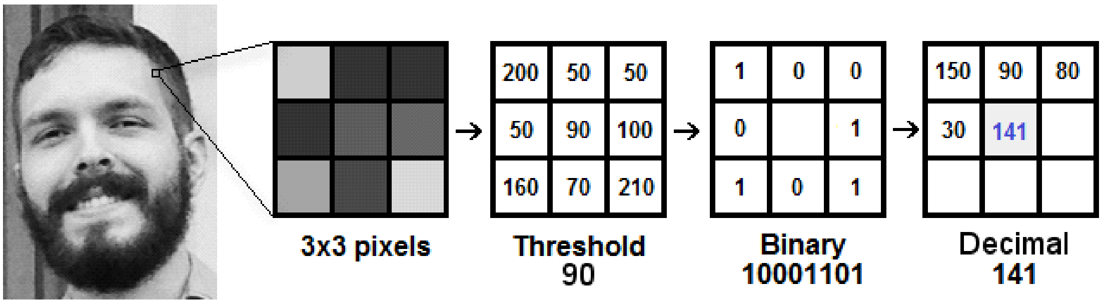
Grid Y: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

Don't worry about the parameters right now, you will understand them after reading the next steps.

2. Training the Algorithm: First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational steps.

3. Applying the LBP operation: The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameter's radius and neighbors.

The image below shows this procedure:




Based on the image above, let's break it into several small steps so we can understand it easily:

Suppose we have a facial image in grayscale.

We can get part of this image as a window of 3x3 pixels.

It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).

Then, we need to take the central value of the matrix to be used as the threshold.

This value will be used to define the new values from the 8 neighbors.

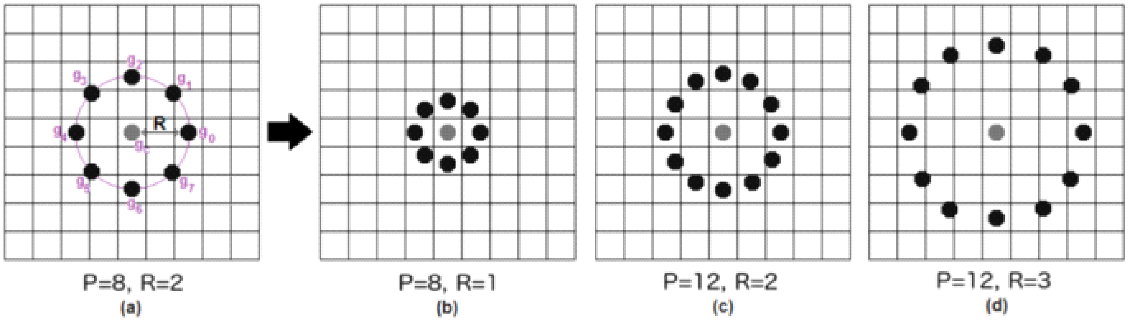
For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.

Now, the matrix will contain only binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g., 10001101). Note: some authors use other approaches to concatenate the binary values (e.g., clockwise direction), but the final result will be the same.

Then, we convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.

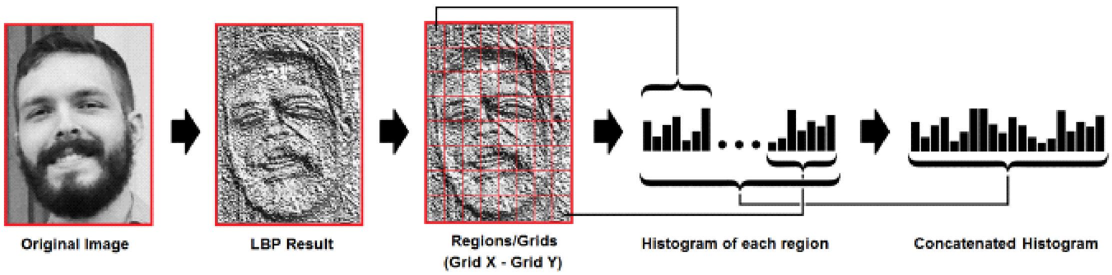
At the end of this procedure (LBP procedure), we have a new image which represents better the characteristics of the original image.

Note: The LBP procedure was expanded to use a different number of radius and neighbors, it is called Circular LBP.




It can be done by using bi-linear interpolation. If some data point is between the pixels, it uses the values from the 4 nearest pixels (2x2) to estimate the value of the new data point.

4. Extracting the Histograms: Now, using the image generated in the last step, we can use the Grid X and Grid Y parameters to divide the image into multiple grids, as can be seen in the following image:




Based on the image above, we can extract the histogram of each region as follows:

As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.

Then, we need to concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the image original image.

The LBPH algorithm is pretty much it.

5. Performing the face recognition: In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, we perform the steps again for this new image and creates a histogram which represents the image.

So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.

We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: Euclidean distance, chi-square, absolute value, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula:

So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a 'confidence' measurement. Note: don't be fooled about the 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.

We can then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

Used MT CNN for Multi face detection

Stage 1:

Obviously, the first thing to do would be to pass in an image to the program. In this model, we want to create an image pyramid, in order to detect faces of all different sizes. In other words, we want to create different copies of the same image in different sizes to search for different sized faces within the image.

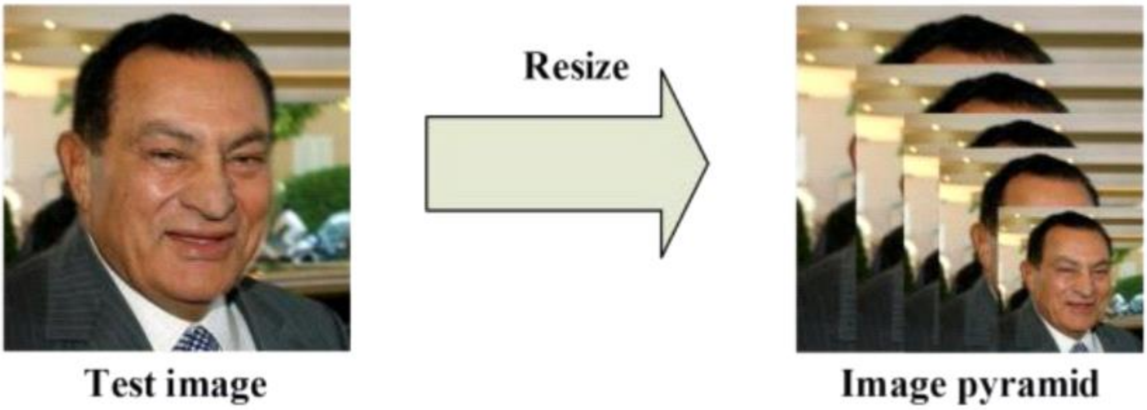



Image Pyramid // Source

For each scaled copy, we have a 12 x 12 stage 1 kernel that will go through every part of the image, scanning for faces. It starts in the top left corner, a section of the image from (0,0) to (12,12). This portion of the image is passed to P-Net, which returns the coordinates of a bounding box if it notices a face. Then, it would repeat that process with sections (0+2a,0+2b) to (12+2a, 12+2b), shifting the 12 x 12 kernel 2 pixels right or down at a time. The shift of 2 pixels is known as the stride, or how many pixels the kernel moves by every time.

Having a stride of 2 helps reduce computation complexity without significantly sacrificing accuracy. Since faces in most images are significantly larger than two pixels, it's highly improbable that the kernel will miss a face merely because it shifted 2 pixels. At the same time, your computer (or whatever machine is running this code) will have a quarter of the number of operations to compute, making the program run faster and with less memory.

The only downside is that we have to recalculate all indexes related to the stride. For example, if the kernel detected a face after moving one step to the right, the output index would tell us the top left corner of that kernel is at (1,0). However, because the stride is 2, we have to multiply the index by 2 to get the correct coordinate: (2,0).

Each kernel would be smaller relative to a large image, so it would be able to find smaller faces in the larger-scaled image. Similarly, the kernel would be bigger relative to a smaller sized image, so it would be able to find bigger faces in the smaller-scaled image.

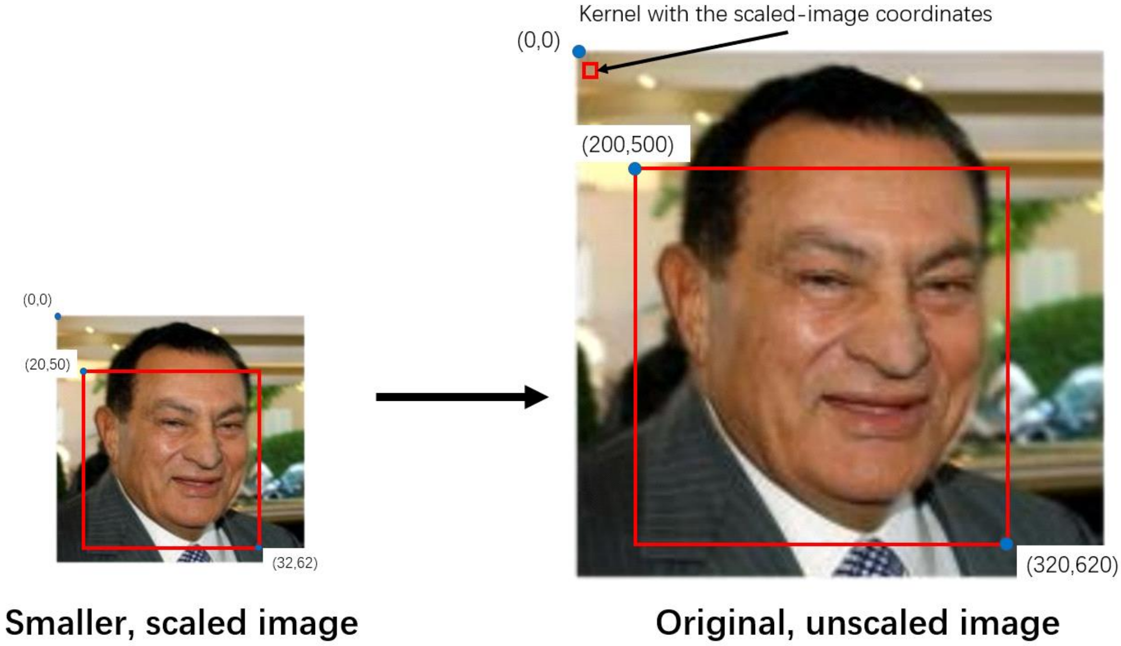
Video: Kernels can find smaller faces in larger images, and bigger faces in smaller images.

After passing in the image, we need to create multiple scaled copies of the image and pass it into the first neural net — P-Net — and gather its output.

| Kernel Coordinates | Bounding box: x1 | Bounding box: y1 | Bounding box: x2 | Bounding box: y2 | Confidence |
|---|---|---|---|---|---|
| (0,5) | 0.50 | 0.25 | 0.80 | 0.58 | 0.98 |
| (0,6) | 0.45 | 0.17 | 0.75 | 0.49 | 0.96 |
| (0,7) | 0.52 | 0.08 | 0.73 | 0.41 | 0.94 |
| (1,4) | 0.42 | 0.34 | 0.67 | 0.66 | 0.92 |
| (1,8) | 0.40 | 0.01 | 0.66 | 0.32 | 0.96 |
| (3,5) | 0.32 | 0.22 | 0.49 | 0.59 | 0.88 |
| (3,6) | 0.25 | 0.20 | 0.52 | 0.53 | 0.91 |
| (6,6) | 0.01 | 0.18 | 0.25 | 0.50 | 0.89 |
| (6,8) | 0.02 | 0.00 | 0.22 | 0.33 | 0.90 |

Sample output for P-Net. Note that the actual output has 4 dimensions, but for simplicity, I've combined it into a 2-dimensional array. Also, the coordinates for the bounding boxes are values between 0 and 1: (0,0) would be the top left corner of the kernel, while (1,1) would be the bottom right corner of the kernel.

The weights and biases of P-Net have been trained so that it outputs a relatively accurate bounding box for every 12 x 12 kernel. However, the network is more confident about some boxes compared to others. Thus, we need to parse the P-Net output to get a list of confidence levels for each bounding box, and delete the boxes with lower confidence (i.e., the boxes that the network isn't quite sure contains a face)



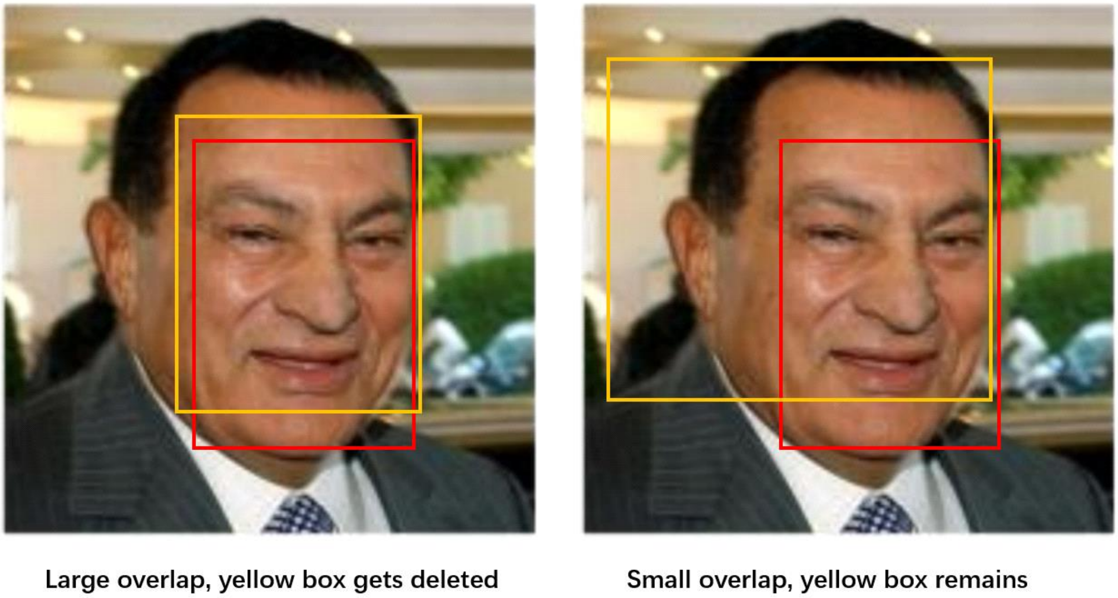

Standardizing kernel coordinates by multiplying it by the scale

After we've picked out the boxes with higher confidence, we will have to standardize the coordinate system, converting all the coordinate systems to that of the actual, "un-scaled" image. Since most kernels are in a scaled-down image, their coordinates will be based on the smaller image.

However, there are still a lot of bounding boxes left, and a lot of them overlap. Non-Maximum Suppression, or NMS, is a method that reduces the number of bounding boxes.

In this particular program, NMS is conducted by first sorting the bounding boxes (and their respective 12 x 12 kernels) by their confidence, or score. In some other models, NMS takes the largest bounding box instead of the one the network is most confident in.

Subsequently, we calculate the area of each of the kernels, as well as the overlapping area between each kernel and the kernel with the highest score. The kernels that overlap a lot with the high-scoring kernel get deleted. Finally, NMS returns a list of the "surviving" bounding boxes.

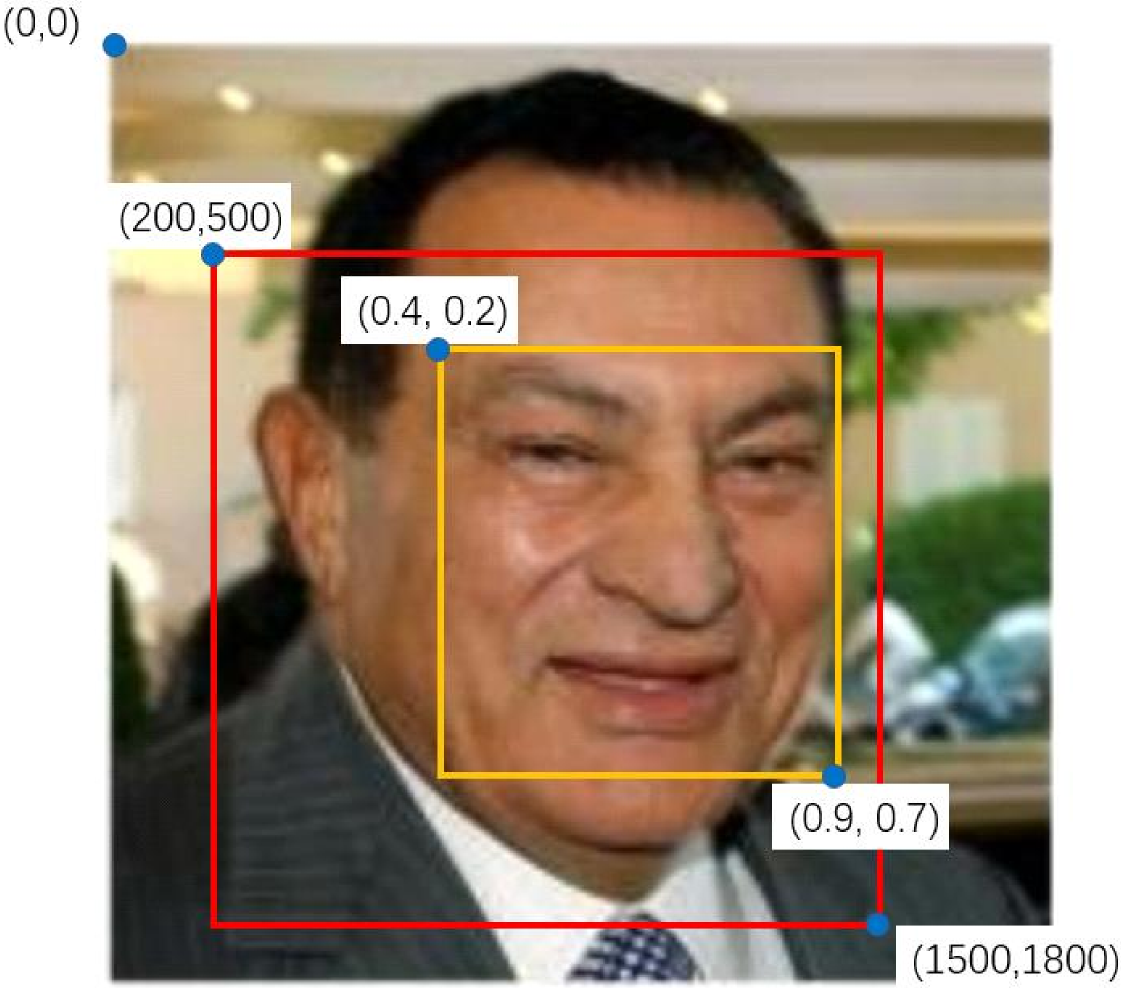



Non-Maximum Suppression

We conduct NMS once for every scaled image, then one more time with all the surviving kernels from each scale. This gets rid of redundant bounding boxes, allowing us to narrow our search down to one accurate box per face.

Why can't we just choose the box with the highest confidence and delete everything else? There is only one face in the image above. However, there might be more than one face in other images. If so, we would end up deleting all the bounding boxes for the other faces.

Afterward, we convert the bounding box coordinates to coordinates of the actual image. Right now, the coordinates of each bounding box is a value between 0 and 1, with (0,0) as the top left corner of the 12 x 12 kernel and (1,1) as the bottom right corner (see table above). By multiplying the coordinates by the actual image width and height, we can convert the bounding box coordinates to the standard, real-sized image coordinates.




Here, the red box is the 12 x 12 kernel, while the yellow box is the bounding box inside it.

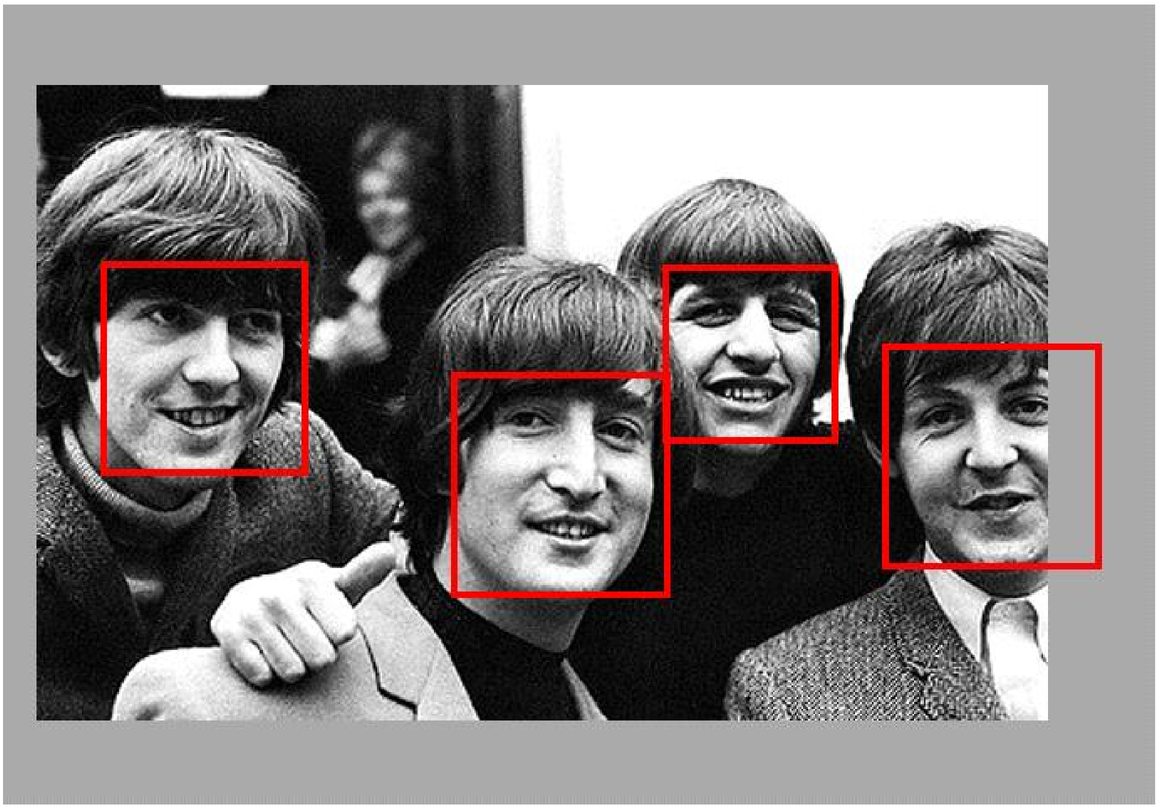
In this image, the red box represents the 24 x 24 kernel, resized back to the original image. We can calculate the width and height of the kernel: 1500–200 = 300, 1800–500 = 300 (Note how the width and height aren't necessarily 12. That is because we're using the coordinates of the kernel in the original image. The width and height we get here are the width and height of the kernel when scaled back to its original size.) Afterwards, we multiply the bounding box coordinates by 300: 0.4x300 = 120, 0.2x300 = 60, 0.9x300 = 270, 0.7x300 = 210. Finally, we add the top left coordinate of the kernel to get the coordinates of the bounding box: (200+120, 500+60) and (200+270, 500+210) or (320,560) and (470,710).

Since the bounding boxes may not be square, we then reshape the bounding boxes to a square by elongating the shorter sides (if the width is smaller than the height, we expand it sideways; if the height is smaller than the width, we expand it vertically).

Finally, we save the coordinates of the bounding boxes and pass it on to stage 2.

Stage 2:

Sometimes, an image may contain only a part of a face peeking in from the side of the frame. In that case, the network may return a bounding box that is partly out of the frame, like Paul McCartney's face in the photo below:



The Beatles and their bounding boxes. Paul McCartney's box is out of bounds and requires padding.

For every bounding box, we create an array of the same size, and copy the pixel values (the image in the bounding box) to the new array. If the bounding box is out of bounds, we only copy the portion of the image in the bounding box to the new array and fill in everything else with a 0. In the image above, the new array for McCartney's face would have pixel values in the left side of the box, and several columns of 0s near the right edge. This process of filling arrays with 0s is called padding.

After we pad the bounding box arrays, we resize them to 24 x 24 pixels, and normalize them to values between -1 and 1. Currently, the pixel values are between 0 to 255 (RGB values). By subtracting each pixel value by half of 255 (127.5) and dividing it by 127.5, we can keep their values between -1 and 1.

Now that we have numerous 24 x 24 image arrays (as many as the number of bounding boxes that survived Stage 1, since each of those bounding boxes has been re-sized and normalized into these kernels), we can feed them into R-Net and gather its output.

R-Net's output is similar to that of P-Net: It includes the coordinates of the new, more accurate bounding boxes, as well as the confidence level of each of these bounding boxes.

Once again, we get rid of the boxes with lower confidence, and perform NMS on every box to further eliminate redundant boxes. Since the coordinates of these new bounding boxes are based on the P-Net bounding boxes, we need to convert them to the standard coordinates.

After standardizing the coordinates, we reshape the bounding boxes to a square to be passed on to O-Net.

Stage 3:

Before we can pass in the bounding boxes from R-Net, we have to first pad any boxes that are out-of-bounds. Then, after we resize the boxes to 48 x 48 pixels, we can pass in the bounding boxes into O-Net.

The outputs of O-Net are slightly different from that of P-Net and R-Net. O-Net provides 3 outputs: the coordinates of the bounding box (out [0]), the coordinates of the 5 facial landmarks (out [1]), and the confidence level of each box (out [2]).

Once again, we get rid of the boxes with lower confidence levels, and standardize both the bounding box coordinates and the facial landmark coordinates. Finally, we run them through the last NMS. At this point, there should only be one bounding box for every face in the image.

Delivering results:

The very last step is to package all the information into a dictionary with three keys: 'box', 'confidence', and 'key points'. 'box' contains the coordinates of the bounding box, 'confidence' contains the confidence level of the network for each box, and 'key points' contains the coordinates of each facial landmark (eyes, nose, and endpoints of the mouth).

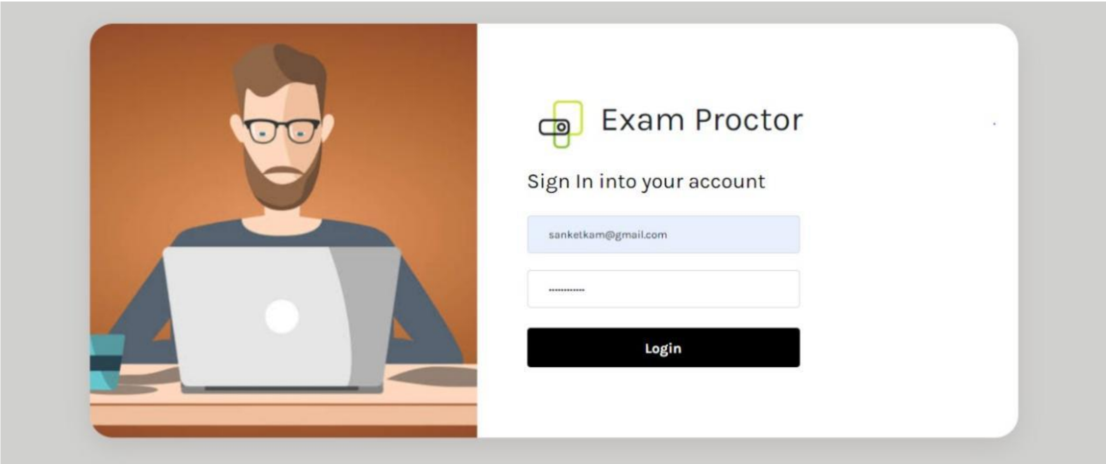
When we want to implement this model in our own code, all we have to do is call detect faces and we'd be given this dictionary with all the coordinates we need to draw the bounding boxes and mark the facial features.

# Integration and Testing

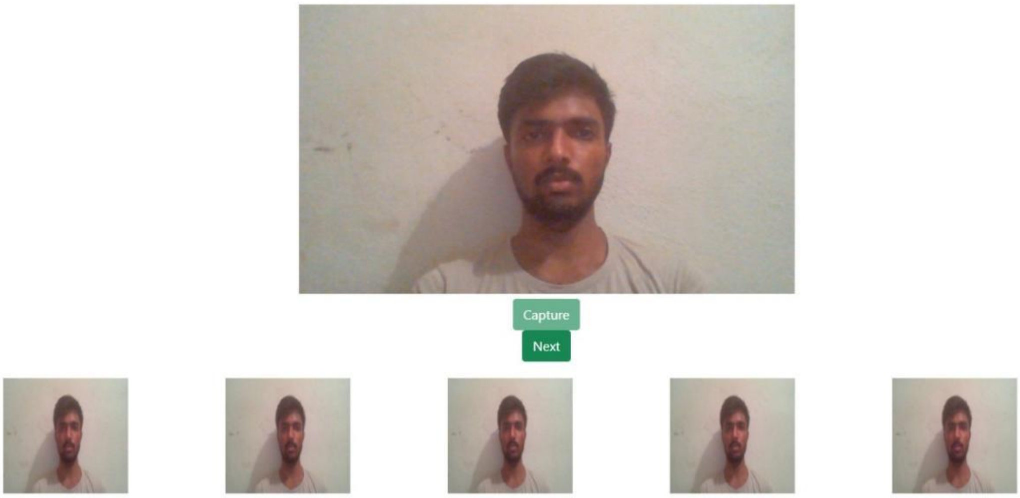
System is mainly divided into three different modules:

I. Face detection and recognition module
II. Multiple faces detection module
III. Object detection module

As mentioned in the requirements the system must have login pages for the student. Therefore, according to the requirement, we have performed the login testing.



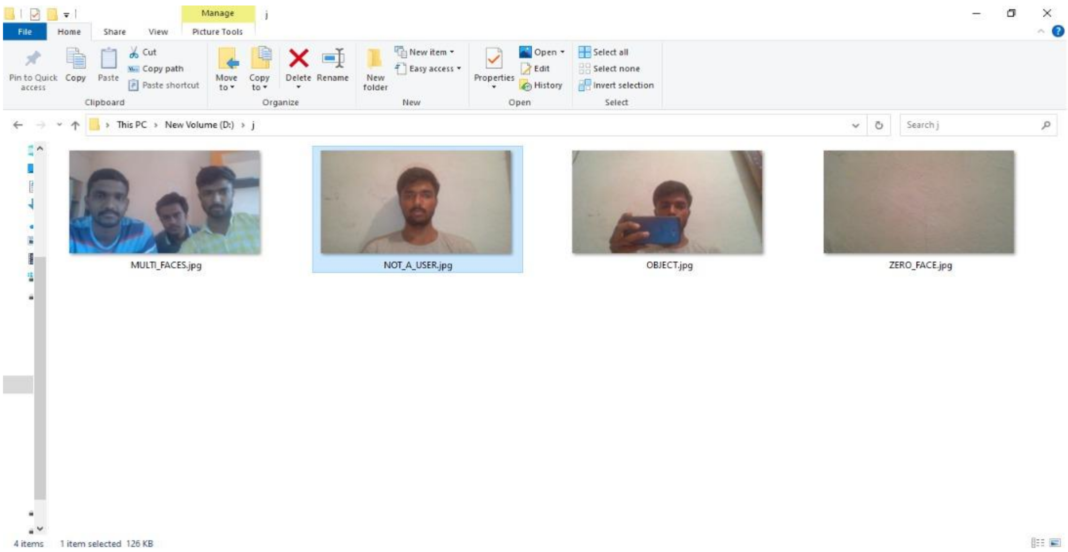

After login student will be asked to take five the reference images by clicking the ‘Capture’ button. After this student has to click the ‘Next’ button.



During entire period of the exam, system will capture images of the student and suspicious activity percentage will be calculated.

Suspicious images include: Images having object, Image with multiple faces or no faces at all, images with different person.

Following are the suspicious images captured by the system during the testing activity.



# Performance Analysis

Performance analysis of the system are as follows:

1. As per the requirements and the testing we can say that our system works in efficient manner.

2. The system can successfully register the student and store the images of students with their names.

3. The student can successfully capture the images and the system store the images in the database as per the name of the student.

4. As per the requirements the system shall capture the images after each 5 seconds throughout the exam and store those images in the database as per the name of the student.

5. The system shall compare each captured image to the refence image. If the image doesn't match to reference image then it will be considered as suspicious activity and every suspicious image will be saved by the system in order to process the output.

6. The system shall provide the output in terms of suspicious activity percentage, i.e., for every suspicious image this percentage will increase.

7. Finally the system successfully generates the report of suspicious activity percentage and each suspicious image in the form of profile of that particular student as final output.

8. Depending upon the requirements and objectives of the project our system works in efficient manner and performance is up to the mark.

# Applications

- For all test administrators, the digital tools that support remote proctoring provide a reasonable level of security for exams taken remotely.

- Also, for online universities and MNC, it provides validation for online education. For recruiters, it brings down the logistics requirement of entry level hiring significantly and hence time to hire and cost of hire.

- For test takers, these systems offer an opportunity to complete online assessments in a variety of times and places as per convenience.

- Remote proctored exams can be unscheduled and immediate, delivered with Internet-based testing. All of the data for the candidate, video & audio logs, screenshots, etc are available for future reference.

# Installation Guide and User Manual

**Installation:-**

1) First install Anaconda3.
2) Then install the Libraries.
3) Then download the Heroku CLI.

**Steps to install libraries -**

1) **steps to install TensorFlow**

On Windows open the Start menu and open an Anaconda Command Prompt. On macOS or Linux open a terminal window. Use the default bash shell on macOS or Linux.

Choose a name for your TensorFlow environment, such as “tf”. 3) To install the current release of CPU-only TensorFlow, recommended for beginners:

Use Command-

conda create -n tf TensorFlow ,conda activate tf,

To install the current release of GPU TensorFlow on Linux or Windows:

Use Command-

conda create -n tf-gpu TensorFlow-gpu

conda activate tf-gpu

**2) steps to install TensorFlow**

1.Open command prompt

2.Run the following command

3.pip install OpenCV-python

4.Successfully installed OpenCV on your system.

**Steps to download Heroku CLI -**

1.Heroku CLI requires git, first install git.

2.Download the appropriate installer for your Windows installation.

# Cost Estimation

| Hardware | Cost |
| --- | --- |
| Computer System | Rs. 45000/- |
| Internet | Rs. 800/- |
| Light Source | Rs. 400/- |
| Total | Rs. 46200/- |

# Ethics

As A Computer Science & Engineering Student, I believe it is unethical To,

1. Make a copy of software for personal or commercial use

2. Make a copy of software for a friend

3. Loan CDs of software to friends

4. Download pirated software from the internet

5.. Distribute pirated software from the internet

6. Buy software with a single user license and then install it on multiple Computers

7. Share a pirated copy of software

8. Install a pirated copy of software

9. Be unfair to and not supportive of their colleagues

# References

[1] E. M. Anderman et T. B. Murdock, « The Psychology of Academic Cheating », in Psychology of Academic Cheating, Elsevier, 2007, p. 1- 5.

[2] K. Curran, G. Middleton, et C. Doherty, « Cheating in Exams with Technology »:, International Journal of Cyber Ethics in Education, vol. 1, no 2, p. 54- 62, avr. 2011.

[3] B. Keresztury et L. Cser, « New Cheating Methods in the Electronic Teaching Era », Procedia - Social and Behavioral Sciences, vol. 93, p. 1516- 1520, oct. 2013.

[4] « Authenticating student work in an e-learning programme via speaker recognition - IEEE Conference Publication ». [En ligne]. Disponible sur: https://ieeexplore.ieee.org/document/5412484/.

[5] R. Bawarith, D. A. Basuhail, D. A. Fattouh, et P. D. S. GamalelDin, « E-exam Cheating Detection System », International Journal of Advanced Computer Science and Applications (IJACSA), vol. 8, n o 4, 2017

[6] Biedny, David, and Monroy, Bert, "Adobe Photoshop Handbook", 2.5 Edition, 1993

[7] Alba, J.W. and Barton Weitz, J.L. (1997), "Interactive home shopping: consumer, retailer, andmanufacturer incentives to participate in electronic marketplaces", Journal of Marketing , Vol.61, July, pp. 38-53.

[8] Eighmey, J., "Profiling user responses to commercial Web sites", Journal of Advertising Research, Vol.37, No. 3:59-66, 1997 [9] Bryan, J. (2006). Technology for physics instruction. Contemporary Issues in Technology

[9] Newton, Harry, "Newton's Telecom Dictionary", 14th Edition, 1998

[10] Ritchey, Tim, and Shobe, Matt, "JavaScript For Macintosh", 1996