**Final year project on**

**Automatic Examination Management System**

**Presented to Faculty of the Department of Electrical and Electronics Engineering**

**University of Ibadan**

**In Partial Fulfillment of the requirements for the Degree Bachelor of Science**

**by**

**Adefela Shogbanmu**

**Matric no:**

**207359**

**500L**

**April 2024**

**Abstract**

The system was built with the intent of automatically recognizing students and sorting them out to their respective seats at default. It is built to improve efficiency and decrease redundancies of repetitive laborious work of seat allocation. This improves the punctuality and saves the need for extra invigilators. It was done by creating an algorithm that contains a trained model (capable of recognizing each student based on Students’ dataset) whose results are now used to assign seats to a candidate. The system has tendencies for refactoring as it can be made mobile against it present deployment on Render server and the ML module can be further improved by training on new datasets which are to be periodically (This periodicity could also be automated) uploaded to its database.

**Acknowledgements**

I would like to express my sincere gratitude to my supervisor Engr. Olufajo. I am also indebted to my colleagues Samuel Akpan, Abdulyekeen Luqman, Morka Chukwuka and Oluwatimileyin for their advice, support and scrutiny

**Table of Contents**

CHAPTER I: INTRODUCTION............................................................................................ 1

1.1 Background study on facial recognition technology………. ............................... 8

1.2 Computer vision.................................................................................................... 9

1.3 Face recognition process....................................................................................... 9

1.4 Statement of problem.......................................................................................... 10

1.5 Aim of study........................................................................................................ 10

1.6 Objective of study.................................................................................................10

1.7 Scope of study ................................................................................................ 10

1.8 Justification of study.......................................................................................... 10

1.9 Limitations of study............................................................................................ 11

CHAPTER II: LITERATURE REVIEW........................................................................... 11

2.1 Facial Recognition systems................................................................................. 12

2.1.1 Face Structure................................................................................................. 14

2.1.2 Feature Support............................................................................................. 14

2.1.3 Face Recognition Algorithms and their limitations......................................... 16

2.2 Seating Algorithms............................................................................................. 16

2.2.2 Room Assignment........................................................................................... 17

2.2.3 Solutions Techniques...................................................................................... 17

2.2.3.1 Local search based Technique…………………………………………….. 17

2.2.3.2 Population Based Algorithm……………………………………………... . 18

CHAPTER III: METHODOLOGY....................................................................................... 23

3.2 Image gathering and standing ........................................................................... 24

3.2.1 ESP CAM......................................................................................................... 25

3.2.2 TTL Converter.................................................................................................. 27

3.3. Machine Learning Technology............................................................................28

3.4 Access Control………………………………………………………………....28

3.5 Seat Assignment Algorithms……………………………………………….. .. 28

3.5.1 Id Generator………………………………………………………… ............. 28

3.5.2 Display system……………………………………………………………….. 29

3.5.3 Web Interface………………………………………………………………. ...29

3.5.4 Seating Arrangement Patterns……………………………………………...... .29

3.5.4.1Checkerboard/ChessBoard Pattern……………………………………..…….29

3.5.4.2 Two Empty Spaces Pattern…………………………………………..............29

3.5.4.3 The Normal Pattern…………………………………………………….. . …30

3.6 Implementation of the Model………………………………………………… …31

CHAPTER IV: RESULTS ...................................................................………………………33

4.2 Facial Recognition Technology......................................................................... …34

4.3 Seat Assignment Algorithms............................................................................ …34

4.4 Technology Lecture Theatre (T.L.T) ……………………………………… ….34

4.5. Patterns……………………………………………………………………… ….35

4. 6 Checkerboard Pattern……………………………………………………………35

4.7 Two Spaces Pattern………………………………………………………………36

4.8 Normal Pattern…………………………………………………………………...37

4.9. Food Technology Lecture Theatre(F.L.T)………………………………………38

4.10 Checkerboard Pattern………………………………………………………… 38

4.11 Two Spaces Pattern……………………………………………………………….39

4.12 Normal Pattern…………………………………………………………………….41

4.13 Multi-capable Algorithm………………………………………………………….43

4.14 Multi-capable patterns…………………………………………………………….43

4.15 Plan representation method………………………………………………………..44

4.16 Performance Analysis and Evaluation…………………………………………… 46

CHAPTER V: CONCLUSION.....................................................................................................49

5.2 Recommendations ……………..................................................................................50

References …………………………………………………………………………………......51

Library Codebase. ...........................................................................................................……. ..53

**List of figures**

Figure 1. Computer vision …............................................................................................ 8

Figure 2. Human vs. Computer vision............................................................................... 9

Figure 3. The Facial recognition process …………………………………..................... 9

Figure 4. Taxonomy of Facial recognition algorithms..................................................... 14

Figure 5. Extraction differences....................................................................................... 14

Figure 6. Metaheuristics…………………………………... ........................................... 20

Figure 7. Airline Seating Problem.................................................................................... 22

Figure 8. ESP32 Cam………........................................................................................... 25

Figure 9. ESP32 Cam with TTL converter…………....................................................... 25

Figure 10. ESP32 Cam pinout configuration……. .......................................................... 29

Figure 11. ESP32Cam with usb to TTL converter .......................................................... 30

Figure 12. Chessboard Pattern code ……………………………..... ............................... 30

Figure 13. Two Empty Spaces Pattern code……………………….................................. 31

Figure 14. Normal Pattern code……………………………... ......................................... 31

Figure 15. Tkinter representation of seat grid................................................................... 36

Figure 16. ESP32 Cam...................................................................................................... 36

Figure 17.Arduino Nano ……………............................................................................... 36

Figure 18.TLT checkerboard Pattern depiction................................................................. 37

Figure 19.TLT Two Spaces Pattern depiction………………………... ........................... 37

Figure 20.TLT Normal Pattern depiction. ........................................................................ 38

Figure 21.FLT Normal Pattern depiction ………………………………………………. 39

Figure 22.FLT Two Empty Spaces Pattern ………….................................................. 42

Figure 23.FLT Checkerboard Pattern………………………………................................ 43

Figure 24.FLT Two Empty Spaces Pattern under multi-capable algorithm...................... 44

Figure 25.FLT Normal Pattern under multi-capable algorithm......................................... 45

Figure 26. FLT Normal Pattern depiction………………………………………………..47

Figure 27. TLT floor plan for the concave hull algorithm……………………………….48

**Introduction**

**1.1. Background Study on Facial Recognition Technology**:

Facial recognition technology has a history that dates back to the 1960s. Woodrow W. Bledsoe and his team conducted experiments between 1964 and 1966 to see if computers could recognize faces. However, the technology was not successful due to the variability in head rotation, tilt, lighting intensity, and angle, as well as facial expression, aging, and other factors in the 1990s, the Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST) launched the FERET Program, which created a database of facial images to evaluate facial recognition systems. This program paved the way for future developments in facial recognition technology. The National Institute of Standards and Technology (NIST) began Face Recognition Vendor Tests (FRVT) in the early 2000s to provide independent government evaluations of facial recognition systems. These evaluations helped law enforcement agencies and the U.S. government determine the best ways to deploy facial recognition technology. The Face Recognition Grand Challenge (FRGC) was launched in 2006 to promote and advance face recognition technology. It evaluated the latest face recognition algorithms available. Facial recognition technology has evolved significantly since its inception. Early systems used rudimentary scanners to map the person's hairline, eyes, and nose. However, advancements in camera technology, mapping processes, and machine learning have improved the accuracy and reliability of facial recognition systems. Today, facial recognition systems use 2D and 3D camera technology to map facial features. 2D systems create flat images of faces and map 'nodal points' (size/shape of eyes, nose, cheekbones, etc.). 3D systems use thermal infrared-based technology to map the patterns of faces derived primarily from the pattern of superficial blood vessels under the skin[.](https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/inspired/history-of-facial-recognition) Facial recognition technology has been integrated into various sectors, including border control, retail, mobile technology, and banking and finance. It has also been used in criminal justice systems, national defense, homeland security, and intelligence. However, there are concerns about the accuracy and reliability of facial recognition technology, particularly in terms of racial discrimination. Some studies have shown that facial recognition algorithms may not perform as well on certain demographic groups, which could perpetuate racial inequality.

Facial recognition technology has a rich history, with significant advancements in accuracy and reliability. However, there are ongoing concerns about its accuracy and potential for racial discrimination.

It has been an active area of research in the past several decades initially a branch of artificial intelligence to enable robots with visual perception, it is now part of a more general and larder discipline computer vision.

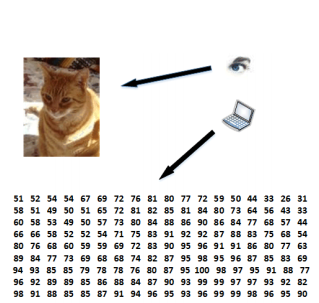
**1.2. Computer Vision:**



Figure

Computer vision applications can process images from a wide range of the electromagnetic spectrum. X-rays are used in medical technology to create images of the human body without surgery. Gamma rays and radio waves in magnetic resonance imaging (MRI) capture images of thin slices of the human body useful for diagnostic and treatment of diseases. X-rays in the automotive industry are used for inspection of material that is hard to detect by the naked eye, such as casting of wheel rims for fractures, cracks, bubble-shaped voids, and defects in lack of fusion. In the food industry, X-rays and gamma rays are used for inspection, safety and quality of their products. Examples include detection of foreign objects in packaged food like fish bone in fish, contaminants in food products such as insect infestation in citrus fruits, and quality inspection for split-pits or water content distribution. In contrast to computer vision, face recognition applications are confined to the narrow band of visible light where surveillance and biometrics authentication can be performed. Biometrics is the term used to describe human characteristics metrics such as iris, fingerprint or hand geometry. These metrics used for identification and access control of individuals that are under surveillance. Face is becoming the preferred metric over current biometrics simply because it is a natural assertion of identity, and its non-intrusive nature provides more convenience and ease of verification. For example, in a fingerprinting system, the subject is required to interact with the system by placing a finger under a fingerprint reader, and the results must be verified by an expert. In contrast, using the subject’s face as a metric requires no intervention, and the results can be verified by a non-expert.

All images must be first captured by a camera and then be given to a computer vision application for further processing. Compared to the human visual system, the camera is the eye, and the processing software is the brain of the application. To acquire the image, the camera uses light reflecting off an object and transmits the light intensity to its built-in sensors. The sensors then convert each of their cell intensities to a value in the range of 0-255, where a grid of numbers in this range becomes the final representation of the captured image. Light is a form of electromagnetic energy spanning a frequency range known as the visual spectrum. Also, sensors are unique to digital cameras as older analog cameras captured images on film. The figure below shows how a human sees an object like a cat and how a computer vision application sees exactly the same object



Figure

**1.3. Face recognition process**

In January 2001 police in Tampa Bay, Florida, used face recognition software at Super Bowl XXXV to search for potential criminals and terrorists attending the event. The system found 19 people with pending arrest warrants. This is a practical example of using a face recognition system in security applications. In automatic face recognition we desire to either identify or verify one or more persons in still images of a scene by means of a stored database of faces. It is the process of labeling a face as recognized or unrecognized. The process has a life cycle based on a pipeline that goes through collection, detection, pre-processing, and a recognition stage. In the collection step, images are captured and stored for training and recognition. In the detection phase, regions of a face within an image are identified and their location is recorded. The pre-processing stage modifies the image by removing unwanted features such as shadow or excessive illumination. Recognition, the final stage of the pipeline, identifies the face as recognized or not recognized.



Figure

At a high level, a facial recognition system relies on the following core steps:

* **Face Detection –** Locate and isolate faces in an input image.
* **Feature Extraction –** Analyze facial features and create a face print.
* **Face Matching –** Compare face prints to a database of known faces to identify an individual.

By leveraging the power of computer vision and machine learning, face recognition AI provides a fascinating application of AI in the real world.

**1.4 Statement of Problem:**  
"The project aims to develop a system that automates examination conduction through the integration of facial recognition technology and a seat arrangement algorithm to enhance efficiency, security, and fairness in examination processes."

**1.5 Aim of Study:**

The purpose of this study is to enhance examination practices with the use of existing technologies such as Facial recognition system and the creation of seating algorithms to automate the process of examination conduction

**1.6 Objectives of the Study:**

Objectives include

* To ensure only verified students enter into the examination hall
* Proper storage and retrieval of students database for access control
* Integration of Hardware with created Server for proper communication and smoothness of use

**1.7 Justification of Study:**

The justification of the study lies in the idea of integrating automation in examination conduction, to properly enhance the process. Study shows that for large examination conduction, verification of students’ identities is paramount in order to prevent cheating and examination malpractice but in such situation the manual labor of verification can be tedious and a more automated approach must be taken to solve the problem of invaders posing as students. Other approaches have been used in times past but this study intends to use F.R.T (Facial Recognition Technology) as an access control to limit the entrance of such marauders. Some previous approaches include tags and biometrics

**1.8 Scope of the study:**

The scope of this study is limited to examination conduction but it could be expanded to include day-to-day class activities and test conduction. The project is limited to

1. Development of a software system that integrates facial recognition technology for student identification.
2. Implementation of a seat arrangement algorithm to optimize seating plans based on predefined criteria.

**1.9 Limitations of the Study:**

This project is limited based on these factors:

1. Accuracy and reliability of facial recognition technology in varying lighting conditions and angles.
2. Complexity in implementing real-time seat arrangement adjustments during examinations.
3. Privacy concerns related to storing and processing biometric data

**CHAPTER 2**

**Literature Review**

**2.1 Facial recognition systems**

A comprehensive survey of different face recognition techniques has been written by Zhao et al. Their survey also has more detailed descriptions of algorithms used for both still and video-based recognition research. Other related areas are also covered, such as psychosocial studies and issues of image variations. Abate et al later did a survey aimed at the trends in 2D imagery and 3D model-based algorithms. The effects of terrorist’s attacks and security flaws are described as a big motivation for government agencies to invest large amount of resources into this kind of research. Again, the varying conditions encountered in images are also described. Due to the importance of facial recognition, several surveys have been published on them. IN a survey published of Anil and Suresh, several face expression recognition algorithms were reviewed such as Patched Geodesic Texture Transform, Bag of Worlds, Local-Directional Number Pattern, Curve-let Feature Extraction, Gradient Feature Matching, and Regional Registration, FARO.

In another survey published by Azeem et al. The effects of partial occlusion on the performance of face recognition algorithms were studied. These algorithms mostly employ techniques such as principal component analysis (PCA), local non-negative matrix factorization (LNMF), non-negative matrix factorization (NMF), independent component analysis (ICA), linear discriminate analysis (LDA), and other variations of these methods.

Zhou et al. also published a survey on the current state of the art face detectors and their performance on benchmark dataset FDDB. They investigated the performance of face detection methods such as Haar-like AdaBoost cascade and HoG-SVM, as representatives of traditional methods, and faster R-CMM and S3FD as deep learning methods on the setting of low-quality images. They investigated the performance degradation of these algorithms when either the contrast level or the blur noise is changed. They showed that hand-crafted and deeply learned features are extremely sensitive and hence, unsuitable for low-quality images. Their research helped other researchers develop facial recognition algorithms that are more practical than previous algorithms.

Face recognition is a fundamental aspect of computer vision that poses significant challenges due to the complexity and variability of human faces. In the thesis "Face Recognition: An Engineering Approach" by Farshad Ghahramani, the author delves into the intricacies of face recognition processes, shedding light on key algorithms and methodologies that enhance the accuracy and efficiency of recognition systems.

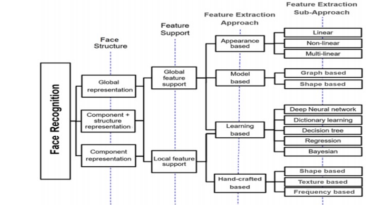
One of the pivotal algorithms discussed in the thesis is the Local Binary Pattern (LBP), which plays a crucial role in extracting discriminative features from facial images. By capturing the spatial relationships between pixels, LBP offers a robust framework for encoding texture information, thereby enabling effective face recognition. Ghahramani's exploration of the LBP algorithm showcases its efficacy in handling variations in illumination, pose, and facial expressions, making it a valuable tool in the realm of facial recognition systems.

Moreover, the thesis delves into the realm of facial feature segmentation, a critical component in the face recognition pipeline. By dissecting facial regions such as the eyes, nose, and mouth, the segmentation algorithm facilitates a detailed analysis of key facial landmarks, enhancing the discriminative power of the recognition system. Ghahramani's research highlights the significance of feature distribution and its impact on recognition accuracy, underscoring the importance of precise segmentation techniques in face recognition tasks.

Furthermore, the thesis alludes to the growing trend of incorporating 3D images in face recognition, signaling a shift towards more advanced and accurate recognition methodologies. By leveraging the depth information provided by 3D imaging, researchers can achieve higher recognition rates and improved robustness against variations in pose and lighting conditions. Ghahramani's exploration of 3D face recognition reflects a progressive approach towards enhancing the performance of recognition systems in real-world scenarios.

In conclusion, Farshad Ghahramani's thesis on face recognition presents a comprehensive analysis of key algorithms and methodologies that drive advancements in the field of computer vision. By delving into the intricacies of algorithms such as LBP and facial feature segmentation, the thesis offers invaluable insights into enhancing the accuracy, robustness, and efficiency of face recognition systems. The incorporation of 3D imaging further underscores the author's commitment to pushing the boundaries of face recognition technology, paving the way for future innovations in this domain.

This literature review encapsulates the essence of Ghahramani's research, highlighting the significance of key algorithms in advancing the field of face recognition and underscoring the potential for further developments in this dynamic and evolving field.

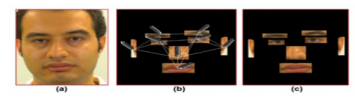


Figure

**2.1.1 Face Structure:**

The picture above illustrates how the recognition solution interacts with face structure, regarding three classes:

* Global representation which focuses on the face as a whole unit
* Component structure representation depending on the different elements of the face for example, the eyes, the mouth, nose as well as their relationship
* Component representation, deals with the selection of specific facial component separately without linking it with the other components



Figure

**2.1.2 Feature support**

This level is concerned with the location (spatial) support which is considered for the feature extraction. It can be local or universal. According to Global feature support which implies the

**Feature extraction approach**

This level is concerned with the special feature extraction approach which may

be identified as follows

*•* Appearance based - Statistical transformations from intense data were used

to derive the features.

*•* Model based -The geometrical elements of the face were used in order to

obtain the features.

*•* Learning based- Features were derived using the learning relationship and

modeling from the inputted data.

*•* Hand-crafted based- Elementary preselected characteristics derived the features.

(a) Feature extraction solutions that are based on appearance are divided as

follows:

*•* Linear solutions, like Principle Component Analysis (PCA) and Independent Component Analysis (ICA), perform a typical linear analysis to reach a space with lower dimension in order to exclude the representative features.

*•* Non-linear solutions, such as kernel PCA, use the structure that

is non-linear in order to achieve a non- linear mapping.

*•* Multi-linear, such as generalized PCA, works on extracting data

from high dimensional data yet preserves its original structure. Consequently, it provides more concentrated representation as compared to the linear solutions.

(b) Model based solutions generate features that are built on the geometric elements of the face; however, they have less sensitivity to the facial variations as they are concerned with the structural data from the face. They, therefore, need accuracy in defining the localization of landmarks before the feature extraction. The division of Model based feature extraction solutions is presented as follows:

*•* Graph based solutions, such as Elastic Bunch Graph Matching (EBGM)

represent facial feature in the form of a graph as the local information of the facial landmarks is stored in nodes. The matching between these nodes can extract information.

*•* Shape based solutions, such as the 3D Morph able Model (3DMM) use landmarks in order to identify facial components. The model controls landmarks while adopting the functions of shape similarity to achieve matching.

(c) Learning based: solutions identify features by identifying the relationship and then modeling them from the inputted data. Compared to different facial variations could emerge within these solutions. Which is mainly depending on the given data, however, they can be more flexible than solutions that depend on the other approaches of face extraction. This is because they need to tune, train and initialize the hyper parameter. Recently, solutions with deep learning bases are strongly encouraged for tasks of face recognition. For instance, deep neural networks dominated innovative model of face recognition with, Convolutional Neural Networks

(CNNs) being the most significant example. Solutions of face recognition that are learning based are divided into five technique families which include:

*•* Deep neural networks, such as the VGG-Face descriptor, help in

handling the input data with high abstraction level and deep processing layers. This helps in extraction features from this input data.

*•* Dictionary learning solutions, such as Kernel Extended Dictionary (KED), based on linear arranged factors helps in feature extraction from input data.

*•* Decision tree solutions, for instance the Decision Pyramid (DP), features are represented as a consequence to a group of decisions

*•* Regression solutions, such as Logistic Regression (LR) identify the links between the different factors through adopting the measured

error and compare it with prediction model

*•* Bayesian solutions, like Bayesian Patch Representation (BPR)

apply the theorem of Bayes in order to extract the features. A probabilistic measure of similarity is then used.

(d) Hand-crafted based solutions conducted features by extracting elements. Generally, these solutions are not very sensitive to face variations, such as pose, occlusion, illumination aging, and expression changes. They can meditate multiple scales, frequency bands, and orientations. The division of hand- craft based feature extraction solutions is presented as follows:

*•* Shape-based: use local shape descriptors to define feature vectors, for

example Local Shape Map (LSM).

*•* Texture-based: explore structure of local spatial neighborhoods, for

example Local Binary Patterns (LBP).

*•* Frequency-based: explore the local structure from frequency domain,

for example Local Phase Quantization (LPQ)

**2.1.3 Facial Recognition Algorithms and their limitations**

Facial recognition technology utilizes various algorithms to identify and authenticate individuals based on their facial features. Here are some key algorithms used in facial recognition technology along with their limitations:

1. **FaceNet Algorithm**: FaceNet is based on a deep convolutional neural network (CNN) that enables face recognition, verification, and clustering. While CNN-based algorithms like FaceNet are highly accurate, they can be computationally intensive and require significant processing power.
2. **Principal Component Analysis (PCA) using Eigen faces**: PCA is a popular algorithm that uses eigenfaces to recognize faces by analyzing variations in facial features. PCA may struggle with variations in lighting, pose, and facial expressions, leading to reduced accuracy in complex real-world scenarios.
3. **Linear Discriminant Analysis (LDA)**: LDA is another algorithm used for face recognition that focuses on maximizing class separability. LDA can be sensitive to outliers and may not perform well when faced with non-linear relationships between facial features.
4. **Elastic Bunch Graph Matching using the Fisher face Algorithm**: This algorithm uses graph matching techniques to recognize faces based on elastic deformations of facial features. Elastic bunch graph matching algorithms may struggle with variations in facial expressions and occlusions, impacting recognition accuracy.
5. **3D Face Recognition**: 3D face recognition techniques use 3D sensors to capture facial shape information for identification. While 3D face recognition is robust against changes in lighting and viewing angles, it may require specialized hardware and can be less practical for large-scale deployments.
6. **Neural Networks**: Neural networks are increasingly used in face recognition for pattern recognition and classification tasks. Training neural networks for face recognition requires large datasets and computational resources, making them resource-intensive.

These algorithms play a crucial role in the accuracy and performance of facial recognition systems. However, each algorithm has its limitations related to computational complexity, sensitivity to variations, robustness in real-world conditions, and data requirements.

**2.2 Seating Algorithms**

Given the master’s thesis by William Ke he discussed a lot about room assignment algorithms and many approaches proffered to solve the problem and its optimization requirements

**2.2.2 Room Assignment**

The room assignment problem can be seen as a variant of the quadratic assignment problem. The quadratic assignment problem was first introduced by Koopmans and Beckmann (1957). The objective of the original problem is to assign a set of facilities (students) to a set of locations with the goal of minimizing the total assignment cost.

The formulation of the original QAP can also be used to represent problems that arise in different fields, not just for location planning. The quadratic term of the QAP can be used to represent any kind of cost that arises from the relationship in between how two objects are assigned. Some examples of how the QAP can be applied are given in the paragraphs below. Steinberg (1961) [10] used QAP for solving the backboard wiring problem. Controls and displays were to be placed on a panel. And the problem was to place the devices so as to minimize the total wire length used to connect the devices together. Dickey and Hopkins (1972) used QAP for campus planning. The problem consisted of placing buildings on sites, with the objective of minimizing the total walking distance between buildings given the traffic intensity between buildings. The travelling salesman problem can likewise be represented as a QAP. The TSP on n cities can be formulated as a QAP (F, D), where F is contains the distances between cities, and D is the adjacency matrix of a Hamiltonian cycle on n vertices. The quadratic assignment problem is in contrast to the linear assignment problem a very hard optimization problem to solve. The time complexity to solve a QAP using brute force search is O(n!), as the set of all possible solution candidates for a QAP of size n is n!. Sahni and Gonzalez (1976) shows in fact that the QAP is a NP-hard problem, and even finding an approximate solution within some constant factor from the global optimal solution cannot be done in polynomial time unless P=NP. As mentioned at the start of the section, our room assignment problem can be represented as a QAP (F, D, B). The F matrix is used to describe the students who should sit together, either because of the students own wishes or because the students have the same supervisor. The D matrix is used to define the distance between workplaces. The B is used to denote the cost of assigning a student to a workplace, with higher cost for assigning a student to a workplace of a different category than the student applied for. A more complete description of the model for our room assignment problem will be given in Section 3.1. The next section describes some methods that are able to solve for a sufficiently good solution to the room assignment problem.

**2.2.3 Solution approaches/techniques**

Some of the techniques used to solve the assignment problem include:

**2.2.3.1 Local Search Based Techniques**

Local search based techniques, such as Hill Climbing, Tabu Search and simulated annealing are commonly seen as belonging to metaheuristics. Metaheuristics are higher level heuristic methods that are designed to provide a sufficiently good solution to an optimization problem. It is most commonly used in cases with incomplete information or limited computational power. Most literature on metaheuristics describes empirical results that are based on computer experiments. We have no mathematical proofs to guarantee optimal solutions, but the solutions that metaheuristic methods give are often sufficient in practice. Local search methods solve problems by searching from an incumbent solution to its neighborhood, hence the name local search. Local search methods differ in the way they define the neighborhood and moving operators. An objective function is used to measure the quality of the resulting solutions.

**Hill Climbing**

Hill climbing is one of the simplest local search methods. It is an iterative algorithm that starts with an arbitrary initial solution and attempts to change to a better solution among the solutions in its neighborhood. This will continue until no further improvements can be found. Hill climbing is well suited for solving convex problems, as a locally optimal solution is also a global optimal solution. Hill climbing will plateau on local optima for non-convex problems. Variants of hill climbing include simple hill climbing, and steepest ascent hill climbing. The former is when the first better solution is chosen, while the latter is when the best solution in the neighborhood is chosen.

**Tabu search**

Tabu search can be seen as a modification to the hill climbing algorithm. The first modification is that the inferior solution might be selected if no better solutions are available. The second modification is done by disallowing revisiting a list of recent moves. These moves are kept in a tabu list, hence the name. The first modification makes the method able to escape local optima, while the second modification is to ensure that the method does not flip back and forth between the same two solutions. One disadvantage with the tabu search method is that the parameters need to be fine-tuned to the specific problem at hand in order for the method to perform well. Examples of these parameters include the size of the tabu list and the stopping conditions.

**Simulated annealing**

Simulated annealing is a method that is inspired from the natural annealing process. It is a probabilistic local search technique that approximates a global optimum for objective functions with many local optima. The idea of the method is to search a wider area of the search space at the beginning of the algorithm by accepting worse solutions with a higher probability. The probability of accepting worse solution goes down as the search progresses. The parameters that can be tuned include: start and end probability, and the reduction rate of this probability. The motivation for choosing worse solutions is the same as in tabu search, which is to avoid getting stuck in local optima.

**Summary of Local Search Based** Techniques

Hill climbing is the simplest algorithm to implement, but it has the disadvantage of getting stuck in local optima for non-convex problems i.e., most practical cases. Tabu search and simulated annealing can be seen as modified hill climbing algorithm whereby accepting worse moves might aid in escaping local optima. One common drawback with Tabu search and simulated annealing is that they require tuning their associated parameters for the specific problem at hand in order to get high quality solutions.

**2.2.3.2 Population Based Algorithms**

Genetic Algorithms, Mimetic Algorithms, and Ant Algorithms are included in a family of population based algorithms known as Evolutionary algorithms. They are commonly characterized by their common inspiration from natural phenomena and behaviors.

**Genetic Algorithms**

Genetic Algorithms simulated the natural evolutionary process by evolving and manipulating populations of solutions within the search space. Solutions are encoded as chromosomes (array of bits) and are evolved over multiple generations using crossover and mutation operators with the aim of incrementally getting better solutions. The parameters and operators of the algorithm need to be properly defined. Consequently, the approach is usually more complicated to implement that local search based methods. The search strategy is fundamentally different from local search based methods in that multiple solutions is managed simultaneously, instead of just a single solution as seen in local search.

**Mimetic Algorithms**

Mimetic algorithms aim to improve upon genetic algorithms by combining them with local search methods. This works by using local search methods on individual solutions of a population in between generations. Mimetic algorithms are able to combine the benefit of exploring large search space from population based algorithms with the benefit of improving individual solutions in a population in a slightly less random way from local search based methods. There are however some drawbacks with Mimetic algorithms. The first drawback is that it increases the computational cost of the algorithm. The second drawback is that this introduces several more design parameters that need to be fine-tuned, including: how often local search should be applied, which solutions should local search be used on, how long should local search be run, and which local search algorithm should be used?

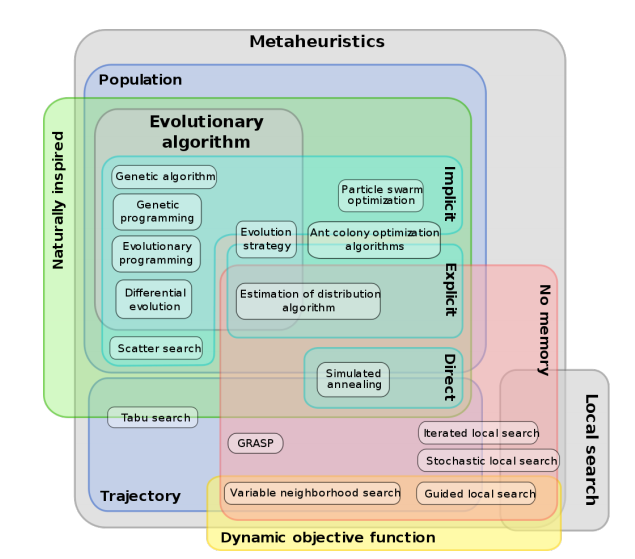
**Ant Algorithms**

Ant algorithms, as the name implies, take inspiration from the way ants behave. It aims to simulate the way ants search for the shortest route to food by using pheromones along the way. Ants of some species will initially wander randomly in the natural world. When they find food, they will return to the colony while placing down pheromone trails along the path. Other ants are inclined to follow the pheromone path instead of wandering randomly, and they will in turn reinforce the pheromone trail if they also find food. However, these pheromone trails do not last indefinitely and will over time evaporate and become weaker. Longer trails will have more time to evaporate than shorter trails. This results in the shortest trails having stronger pheromone trails than other paths over a period of time.

**Summary of population based Algorithms**

The ability to evaluate multiple solutions simultaneously is a huge benefit with population based algorithms. Each generation of population based algorithms has a larger computational cost associated with it compared to a single iteration in local search based techniques. The benefits of using population based algorithms are therefore not overly apparent, at least with respect to total computation time.

In, Harsh vardhan Dwivedi, an implementation of the room assignment problem was solved on NVIDIA GTX-580 GP-GPU architecture using CUDA. In, Martinez-Alfaro et. al., present a Simulated Annealing-based solution for the Classroom Assignment problem, which is analogous to the Room Assignment problem. In this case, classrooms in a large institution are assigned to classes subject to classroom and instructor availability and classroom special resource availability, among other factors. A modular cost function composed of terms related to these factors is used in their proposed optimization approach. The advantage with a modular cost function is that it can be simplified to fit the data available for the problem at hand. The authors explain that they could use this cost function, for instance, with just the meeting times data and without any regard for special resource requirements for a classroom by attaching zero weight to the term(s) related to that data. Inspiration was drawn from them in the implementation of seat assignment system

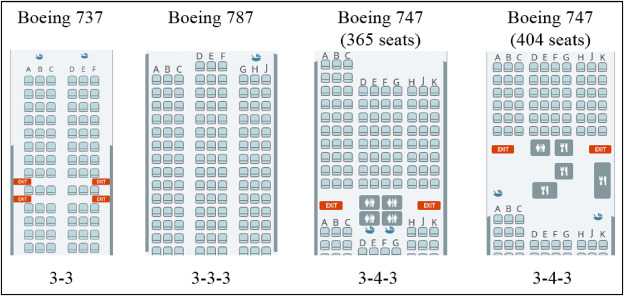
****

Figure

For the research at hand novel/simple algorithms had to be implemented to fit the contextual application of the project, as general seating arrangement were not fully developed, they were based on pre-existing algorithms such as trees, Graphs and even hash maps. Hence slight modification into these pre-existing algorithms brought about specific use case. The task of this thesis is very specific and no comparing products can be found that offer the required functionality.

Maximilian Deutsch, in his thesis showed the use of concave hull algorithms to generate a seating cluster for seating representation. He proposed the use of automatically generating the plan images of venues based on user’s input. Given the time and the learning curve required to implement such algorithms, this approach was not implemented but it is worth considering for future researchers. I used a system of patterns for the representation of the hall not the overall layout of the venue. Maximilian Deutsch provided such remarkable insights into the development, testing and integration of his thesis. His study presents a process of generating seating plan images for the Ticket Gretchen app. Other works related to seat assignment include the famous airline seating problem and the train optimization problem. Both of these problems will be briefly talked about. The impact of the study as it relates to this study will be substantiated. In, James Doveren, the airline seating problem for aeroplane boarding was rigorously discussed along with its financial implications. Due to the potential economic benefits, research into optimal boarding procedures has been conducted for some time. In the context of the traditional scenario in which passengers are assigned their seats prior to boarding and the sequence in which they board may be altered, cites Steffen’s method as the fastest procedure to board by seat. Initially proposed in, it results from generalizing a seating pattern that was the result of a Markov chain Monte Carlo optimization. Steffen’s method primarily focuses on the time passengers spend stowing their luggage in overhead compartments and garnered positive reception in both academic as well as mainstream publications. Delays in airplane takeoffs cost $29 billion annually. The costs of these delays have been estimated from $30 to $250 per minute. The time to board passengers can contribute to these delays. A number of researchers have created and investigated methods that board passengers by groups, with the group assignment depending upon their seat assignment. For instance, Van den Briel et al. shows that it is effective to board the passengers who will be sitting in window seats before the passengers who will sit in middle seats and to board the passengers who will sit in aisle seats last. This approach eliminates delays caused by seated passengers leaving their seats to make room for other passengers to sit down. A number of researchers have recently used stochastic simulation and field tests to compare the performance of boarding strategies. Schultz estimates the improved boarding performance resulting from a “slip seat” configuration that enables aisle seats to temporarily move closer to the windows, thereby extending the aisle width, permitting passengers to pass others waiting ahead of them in the aisle.

Steffen improves on the boarding by group approach by determining the optimal sequence in which passengers should board a fully loaded single-aisle plane based upon their assigned seats. In this sequence, a passenger sitting in a window seat in the back row of the plane will be the first to board. The second passenger to board will be sitting in a window seat two rows in front of the previously boarded passenger, and so on. The passengers continue boarding in a sequence based on their seating assignments in every other row (back to front): Window seats first, and then middle seats, and, finally, those sitting in the seats adjacent to the aisle. The final passenger to board sits in an aisle seat in the first row of the plane. This method works well, assuming that passengers can board in a predetermined sequence; however, Steffen assumes that passengers are assigned to seats randomly and independently of the luggage volumes they carry aboard the plane. Milne and Kelly extend Steffen’s work by assigning passengers to seats so that their carry-on luggage is distributed approximately uniformly throughout the plane. Once those passengers have been assigned to their seats, they board the plane in the sequence determined by Steffen. The method of Milne and Kelly results in a faster time to complete the boarding of the plane than when carry-on luggage is not aligned with the seat assignment. Further improvement results from Milne and Salari, who use a mixed integer program (MIP) to assign passengers to seats based on the quantities of their carry-on luggage. Milne and Kelly, as well as Milne and Salari, assume that after passengers have been assigned to seats, they board the plane according to the Steffen sequence. The MIP proposed by Milne and Salari minimizes the time to complete the boarding of the plane and results in a faster boarding time than that using the method of Milne and Kelly.



Figure

Milne and Salari assume average passenger walking and sitting times. In reality, these times vary from passenger to passenger because some passengers are more agile than others. Milne and Salari acknowledge that there are many optimal solutions that minimize the time to complete boarding when passenger walking and sitting times are deterministic.

**CHAPTER 3**

**Methodology**

This paper is aimed at the reduction of redundancies of manual methods. A system was developed; the system has various approaches determined by the way its image datasets are acquired. The system is aimed to fulfill some requirements namely

1) Usability: The system should be able to achieve all its specific objectives. All its various components and functions should be usable

2) Reliability: The system should be consistent all the time. Functions in the system should produce the correct output each time.

3) Responsiveness: this system must respond in applying various techniques

A lot of research has been done on facial recognition system as it is not a new thing but this alone has some limitations since it could just be seen as another camera. It needed some sort of optimization; this optimization (the seat arrangement algorithm) was conceived by my project supervisor, Engr. Olufajo. This in fact gave more meaning to the project and opened up several possibilities.

This system could also serve as a multipurpose system capable of generating exam time tables (keeping in mind students taking other courses but with the alteration of it present codebase), exam seat generator and the usual verification of students. Keeping these in mind a system was developed that satisfies all these criteria.

OBJECTIVES:

1) To ensure only authenticated students can enter/leave the exam hall

2) To ensure student detail is stored in a database system: this entails multiple Image entry of the students for artificial intelligence to perform scans, this system should be able to allow the administrator or perhaps examiner to see the result of the scan with a preferred message of recognized or unrecognized followed by a prompt to try retry if message unsuccessful. The system must be able to insert the data of the students into the model

3) To check the list of the student details (Matric, name, level) after the face scan process through the web interface.

Deep learning techniques have been used to develop this system, histogram of oriented gradient method is used to compute and compare feature facials of students to recognize them other algorithms can also be implemented to improve accuracy.

The system has been trained using neural networks. Two approaches come to mind in order to gather the images of students

The use of a camera connected to an ioT device perhaps ipv6 (or some kind of router) though after my research I discovered a simpler camera model (esp32 cam which comes with its own wi-fi module) it could be used to remotely send images of the students to a local remote server/device from which a dataset can be generated to train the model (built from scratch or using a pre-trained model like SSD) but this process could be done depending on the administrator in order to achieve a desired accuracy. This method is automatic and could be further improved when new snapped images are combined with an image dataset of the students.

Automatic Examination Management system has two subsets systems which include the access control system and the seating assignment system. Each system are mutually exclusive but are needed to solve the problems posed by examination infractions due to malpractice.

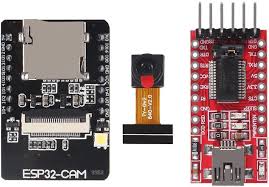
The Approach used in this project entails Image Gathering and Sending, Training of Machine Learning module, Access Verification and Seating assignment

**3.2 Image Gathering and sending:**

In this process the images of the students’ are gathered and stored in the database. It entails the use of Esp cam32 to take the images of students’ across different variants such as lighting, humidity and contrast in order to ensure model accuracy of the facial recognition module. In detail, It takes a picture of a student sends it across a WIFI module to a server; this server houses the machine learning modules and the web-app for the administrator.



Figure



Figure

3.2.1 **ESP32CAM** is a powerful development board that combines the ESP32 microcontroller with a camera module. It’s ideal for IoT projects that involve video streaming.  
With built-in Wi-Fi and Bluetooth capabilities, the ESP32CAM module is perfect for remote monitoring, security systems, and any task that requires live, wireless video streaming.

### ESP32CAM Features

* The tiniest SoC module (Wi-Fi & Bluetooth)
* A Low-power 32-bit CPU
* The maximum clock frequency of 160MHz, and 600 DMIPS of computing power
* Internal 520KB SRAM, plus an external MPSRAM4
* Supporting UART/SPI/I2C/PWM/ADC/DAC
* Compatibility with cameras like OV2640 and OV7670
* An onboard flash lamp
* Image uploading via WiFi
* TF card support
* Multiple sleep modes
* LwIP and FreeRTOS
* STA/AP/STA+AP operating modes
* Smart Config/AirKiss technology support

### ESP32CAM Applications

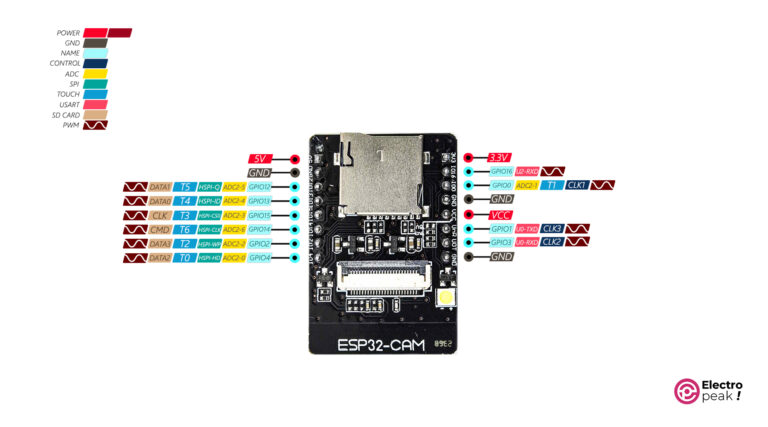
ESP32CAM opens up a whole new world of possibilities with its advanced features and customization options for video streaming. Let’s explore a few exciting applications:  
• Security purposes: With several ESP32CAM modules, you can build a smart security system.  
• Robotics: By sending images over Wi-Fi and processing them with some powerful hardware, ESP32CAM becomes a key player in all sorts of robotics projects.  
• IoT: When it comes to executing IoT projects, visuals play a vital role. ESP32CAM offers many features—a microcontroller, built-in Wi-Fi, Bluetooth, and a camera—so it’s perfect for smart facilities.  
• Applications Combined with image processing software (like OpenCV): Using ESP32CAM, you can add image processing to your projects.

## ESP32CAM Pinout

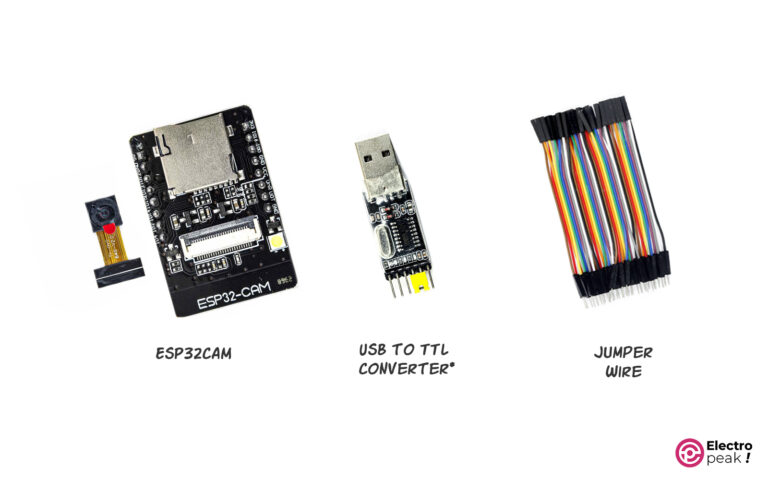
This module has 16 pins:  
**5V**: Power supply  
**GND**: Ground  
**GPIO2-16**: Digital input and output pins. They also come with a bunch of extra features like:

* Analog-to-digital conversion (7 channels)
* SPI communication
* Memory card port
* UART

Here’s the pin out in the image below:



Figure



Figure

**3.2.2 TTL Converter:**  A TTL (Transistor-Transistor Logic) converter is often used in connection with the ESP32-CAM (ESP32 Camera Module) for interfacing the module with other devices or systems that operate at different voltage levels or logic levels. The ESP32-CAM module typically operates at 3.3V logic levels, while many other devices or microcontrollers operate at 5V logic levels.

Here's how the TTL converter can be used with the ESP32-CAM:

1. Voltage Level Shifting: The ESP32-CAM operates at 3.3V logic levels, but some devices or microcontrollers might operate at 5V logic levels. The TTL converter can be used to shift the voltage levels between the ESP32-CAM and the other device, ensuring compatibility and preventing potential damage to either device.

2. Logic Level Conversion: In addition to voltage level shifting, the TTL converter can also perform logic level conversion. This means it can translate logic high and logic low signals between the 3.3V levels of the ESP32-CAM and the 5V levels of the other device, ensuring proper communication between them.

3. Bidirectional Communication: Some TTL converters support bidirectional communication, allowing data to be transmitted and received between the ESP32-CAM and the other device. This is useful in applications where both devices need to exchange data bidirectionally.

4. Protection: TTL converters often provide protection features such as over-voltage protection and over-current protection, which can help safeguard the ESP32-CAM and the connected device from potential electrical damage or interference.

In summary, the TTL converter facilitates seamless communication between the ESP32-CAM and other devices or systems that operate at different voltage levels or logic levels, ensuring compatibility, proper signal translation, and protection against potential damage.

**3.3 Machine Learning Training:**

This aspect makes use of computing power to train a machine learning module. It involves high computational power hence the use of an external server to host and run the machine learning module. It also accommodates processes/functions that link to the database where the images of the student are stored. With these images, the model is trained.

**3.4 Access Control:**

For the Access control the use of the trained Facial recognition algorithm that verifies the identity of the student. It is the resultant output of the previous training method

**3.5 Seating Assignment Algorithms:**

This stores the entire seating structure of the Lecture Theatre in a grid system and then performs simple looping concepts and finds those seats that are available and based on the preference setting of the Administrator adjusts the assignments of the seats to include spacing. The Seating assignment Algorithms needs some helper systems such as the id generator and Display System. But for the purpose of this project only four algorithm types were created to compensate for the required patterns often generated from students’ seating arrangement. Some of this arrangements or patterns are discussed below

**3.5.1 Id generator:**

It is a simple system that takes a dictionary of students’ database and from this generates a two-character id for each student. This is mandatory as the full name of the student can’t be viewed from the small grid area; hence a two-character id is sufficient for the purpose of enumeration and clarity. A pop-up is put in the web-app with the help of JavaScript. This fetches from the database and displays the student information whenever the Administrator hovers that particular seat. The information displayed includes the Name, Matric, Department, Level and an Image of the user perhaps for a second verification for the administrator in cases of aliens invading the examination hall.

**3.5.2 Display System:**

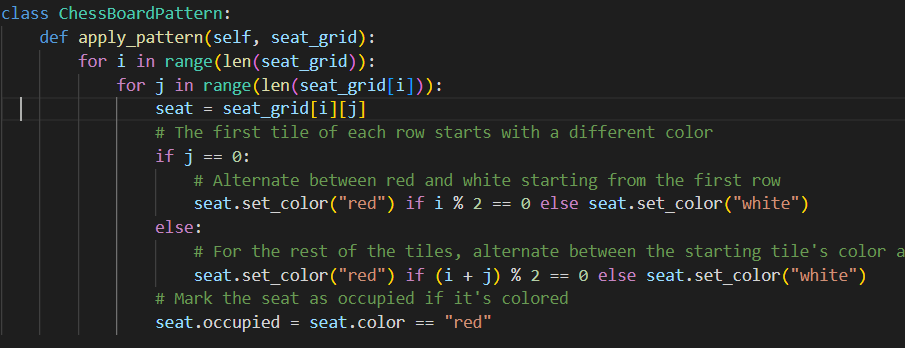
This system is responsible for the grid generation and seating layout visualization. The system uses tkinter library in it base form to generate a grid layout for the lecture theatre, it receives the output of the seating algorithms and displays it to the Administrator. The Design system gave birth to the web interface

**3.5.3 Web Interface:**

This was built with the administrator’s ease of use in mind. This integrates a mixture of python and JavaScript. The frontend client uses the JavaScript framework ReactJs while the backend handles requests through python’s flask framework

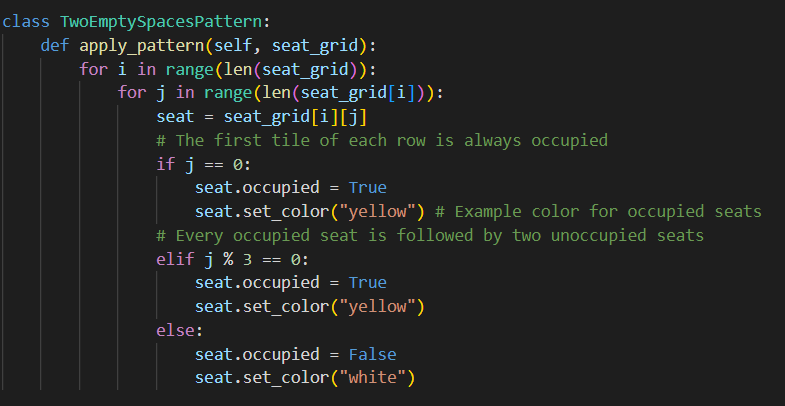
**3.5.4 Seating Arrangement Patterns:**

**3.5.4.1 Checker-board/Chess Board Pattern:** This is a simple pattern at which the student seating arrangement resembles the pattern made by the dark squares on a chess board. It entails the vacation of a space, this can be modified for examination safety, between each student in a row and the alternation of students to ensure students on the proceeding row do not seat at the back of student on the preceding row and so forth.



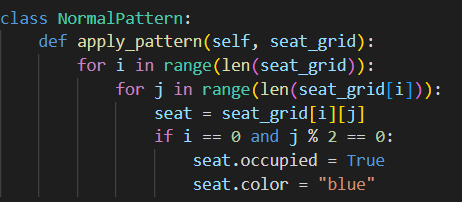
Figure

**3.5.4.2 Two empty spaces Pattern:** This pattern is a simple pattern that creates a rigid structure but ensures sufficient spacing is kept between each student. It gives the utmost number of spaces between each student but students at the proceeding row seat at the back of the student on the preceding row. It could be modified to leave entire rows empty between proceeding and preceding rows to eliminate examination malpractice.



Figure

**3.5.4.3 The Normal Pattern:** This pattern is very simple and entails a rigid structure. It is similar to the two empty spaces Pattern but instead of implementing much space it uses a default of a space, this can be modified by the Administrator, and students sit on the same columns.



Figure

Because of the need of the institution where by different levels share the same lecture theater it becomes rather impossible to implement the previous patterns to suit this need case. Hence, the requirement for a multi capable algorithm is rather intuitive. It arranges students into optimal seats to minimize examination malpractice. It uses some of the previous algorithms under the hood and in some cases in order to account for the total students count it allows the total use of the available seats but with the constraint of not allowing student of the same level sit side-by-side.

**3.6 IMPLEMENTATION OF THE MODEL/SYSTEM**

The model can't attain a somewhat useful degree of certainty without data.

How then can this data be gotten except through input of users or generation of a database from previous inputs? This phase of registering users to the system is called the Registration phase

Registration Phase: This is the phase where the student's details like Matric, names, levels etc are stored against the images of the students

A dictionary is created with the Matric number of each student as the key and other details as values. On another thought, a database could be created that takes in each student’s detail/ and can be queried for validation; whichever that seems feasible. Both approach yields similar results but for the sake of feasibility and scalability; the database creation was optimal for the circumstances of this project

After the proper registration of all students comes the training phase accompanied by testing the model

Identification and Verification Phase: The verification phase compares the student with whom they say they are while the identification phase checks/compares the given information to all the other individuals in the database and gives a ranked list of matches

Arrangement Phase: this is where the system/model, after the verification/identification arranges the students (recognized) in the exam hall with given constraints.

It could be further improved by adding to this, an algorithm for seat allocation. The system could be made to do all the manual work of calling names and arranging students before exam automatically thereby increasing speed and punctuality of exams.

This algorithm will take in the number of students present (counted as those recognized), number of seats available (initial and free due to lateness of some students) and arrange them against the number of seats with a space criterion set by the examiner (e.g. two spaces between). This system could be further improved by adding bias (a kind of input from the supervisor/invigilator) which provides a constraint for the algorithm e.g. (Mr. A and Mr. B must not seat down together, isolation of Mr. C by giving him the front seat beside the examiner) etc

After the execution of the algorithm some sort of display/visually interpretable form of the information is needed to effectively communicate the specified location of each person. An approach comes to mind by generating square tiles to represents seats and perhaps some object (a circle that shows user Matric number when clicked on/hovered over).

The phase leads to some sort of GUI/DISPLAY PHASE. This phase can be separated into two parts the first part displaying the algorithms result in some optimized way (by using python libraries or easy to use svg/canvas). Lastly, the captured images on the hardware device are transmitted automatically using wifi-module.

The phases for development have been further divided into four stages namely:

1. IMAGE ACQUISITION
2. FACE DETECTION
3. FEATURE EXTRACTION
4. FACE RECOGNITION
5. SEAT ARRANGEMENT

IMAGE ACQUISITION: It captures the images of the students/classroom and sends it to the image pre-processing then the image is sorted out for face detection

FACE DETECTION: This separates the facial area from the rest of the background image. The faces which are then stored in the database

FEATURE EXTRACTION: it is done for the distinguishing of faces of different students. In this system, eyes, nose and mouth are extracted; it is helpful in face detection and extraction

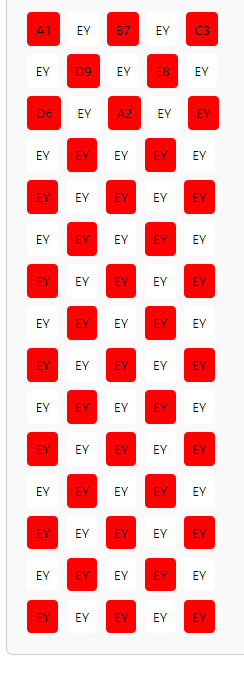
FACE RECOGNITION: The facial image is then compared with the stored image and the face is then recognized. The attendance of that particular student is marked and stored in csv/excel file format

SEAT ARRANGEMENT: This uses an algorithm to automatically arrange the seats. Some algorithms that come to mind can be quite complex but a simpler implementation for this purpose can be achieved by using some kind of FIFO system where the seat can be represented either as a matrix or as some queue like structure and some manipulations in the form of algorithms provide results with the desired constraints. There exist some pre-trained A.I. models (like SSD) that can recognize objects like chairs, tables etc. This model could further be trained with Kaggle Open Images Object Detection Dataset. This model acts as a filter layer that can properly distinguish between students & their surrounding objects. Hence it could be trained to identify an empty/occupied seat making the work somewhat easier. But alas, most of the above are computationally complex and hence have not been implemented

**CHAPTER FOUR**

**4.1 Results**

Continuing from the discussions of the previous chapter, it is important to formulate the result of the processes stated in the previous sections. Some of the images below are from the web interface that depicts what the algorithm performs while others are its performance evaluation. Some pictures show the development of the system from infancy to adulthood. The system in its infancy used the tkinter library but it didn’t suffice for the job at hand.



Figure

For simplicity only two halls where chosen to suffice for the test of the algorithm. They include T.L.T(Technology Lecture Theatre) and F.L.T (Food Technology Lecture Theatre). From the buildup process we had a plan to make it as simple as possible, this meant the representation of the lecture halls as grids, the other alternative was to use the plans of the lecture Theatre but this method was a laborious one, and also plans of the two selected halls where not readily available and time cost of each was too huge but regardless sufficient work was made in that field and has been left for future thesis aspirants to make it their own, the details of what has been achieved and what could be improved upon are included below with their pictorial representations.



Figure

Once again let me work you through the flow of the system: The Esp cam is responsible for taking pictures and it utilizes the microcomputer Arduino-nano for its processing and then sends this image to the server, the server houses the two subsystems of the code: The facial recognition system and the seat Assignment algorithms.



Figure

This chapter is further divided into two parts to depict the results gotten from each subsystem

**4.2 Facial Recognition Technology:** This has been rigorously treated in the previous chapters. Its performance evaluation is treated in 4.14

The second Subsystem includes the Seat Assignment Algorithm.

**4.3. Seat Assignment Algorithm:**

The accuracy and work gotten from each representation of the Lecture halls and their representation in the various patterns are given below

**4.4. Technology Lecture Theatre**:

This is one of the most prominent lecture theatre used for examination conduction in the faculty of technology hence taking it into account was mandatory: The hall was represented as a 20x30 grid in python to make it simpler and smoother for representation. The web interface gives this image whenever the administrator selects TLT from the options

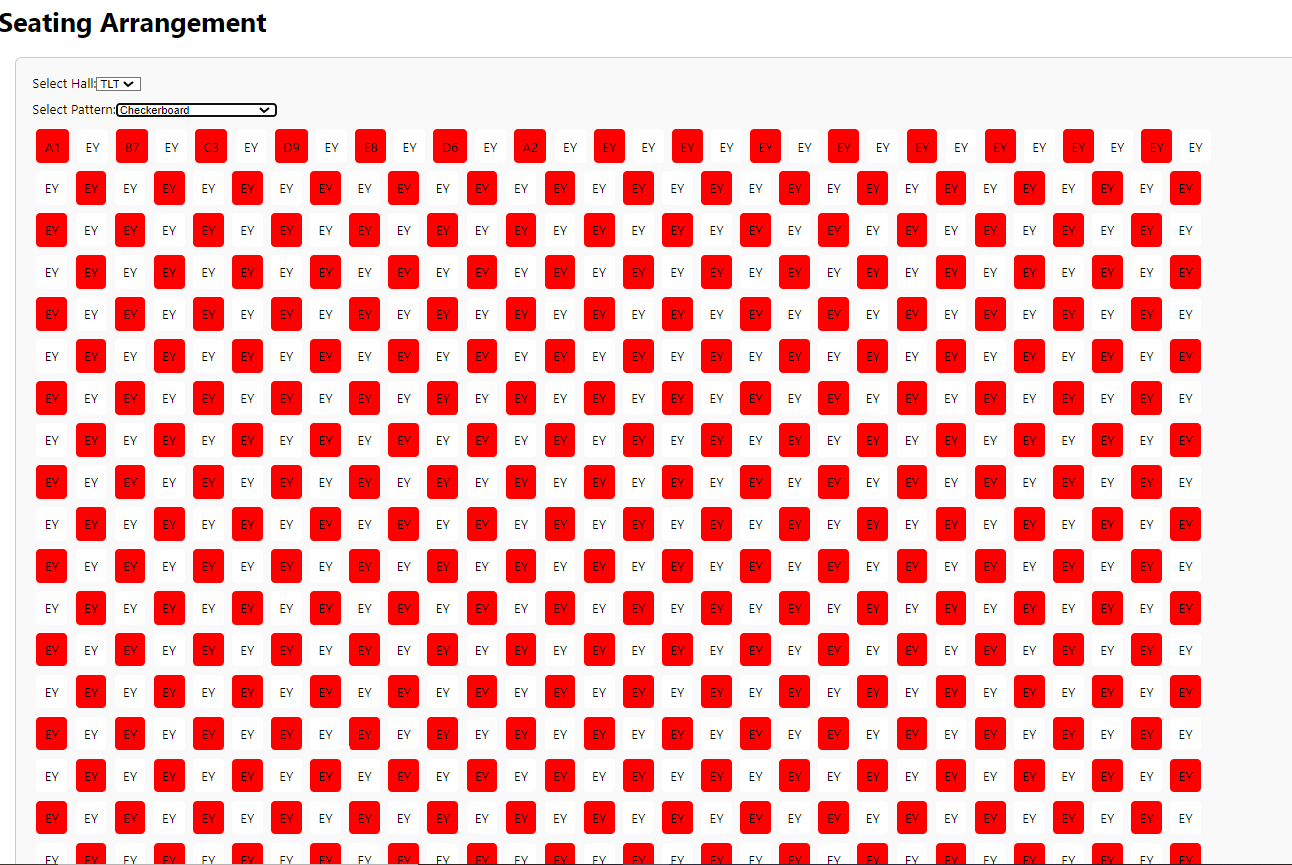
The web interface makes it possible for the administrator to be able to view the holder of the seat after seat assignment in order to be able to detect aliens who manage to escape the access verification part of the system.

**4.5. Patterns:**

This is the representation of the seating of the students, it involves the overall shape the student attain after being arranged and assigned. It makes the code easier to attain by reducing the complexity in shape to simple shapes, any assignment of the students can be easily implemented. Some of the selected patterns used along with the lecture hall configured for are explained below

**4.6. CheckerBoard Pattern**

From the image below we see TLT in the Pattern option-Checker Board, This gives a visual representation of how the students could be made to sit in the exam hall with the added benefit of secondary verification by the administrators. The red colored seats depict the seats assigned while the whites are the unassigned seats (please note that spoilt and defected seats in the lecture theatre where not taken into account but are left as a future challenge to thesis aspirants or those who intend to pursue this project topic. The need for a two-character id has been rigorously explained in chapter three of this project but as a subtle reminder. The Viewer can see that space is luxury and the character’s length a bigger one. Hence empty seats are labeled as Ey. The conclusion to use this is one of necessity not of fashion or rizz creation.



Figure

The second option from the list is the two spaces pattern

**4.7. Two Spaces Pattern:**

This is a common pattern where two spaces are left between each student it is an efficient pattern one that has been used greatly by the Engr. Ogundipe in his exam supervision. Due to its regular use it becomes a pattern that should be hardcoded into the code. The assigned seats are colored blue to depict seat placement with the previous functionalities present on it. The unassigned seats are colored white.



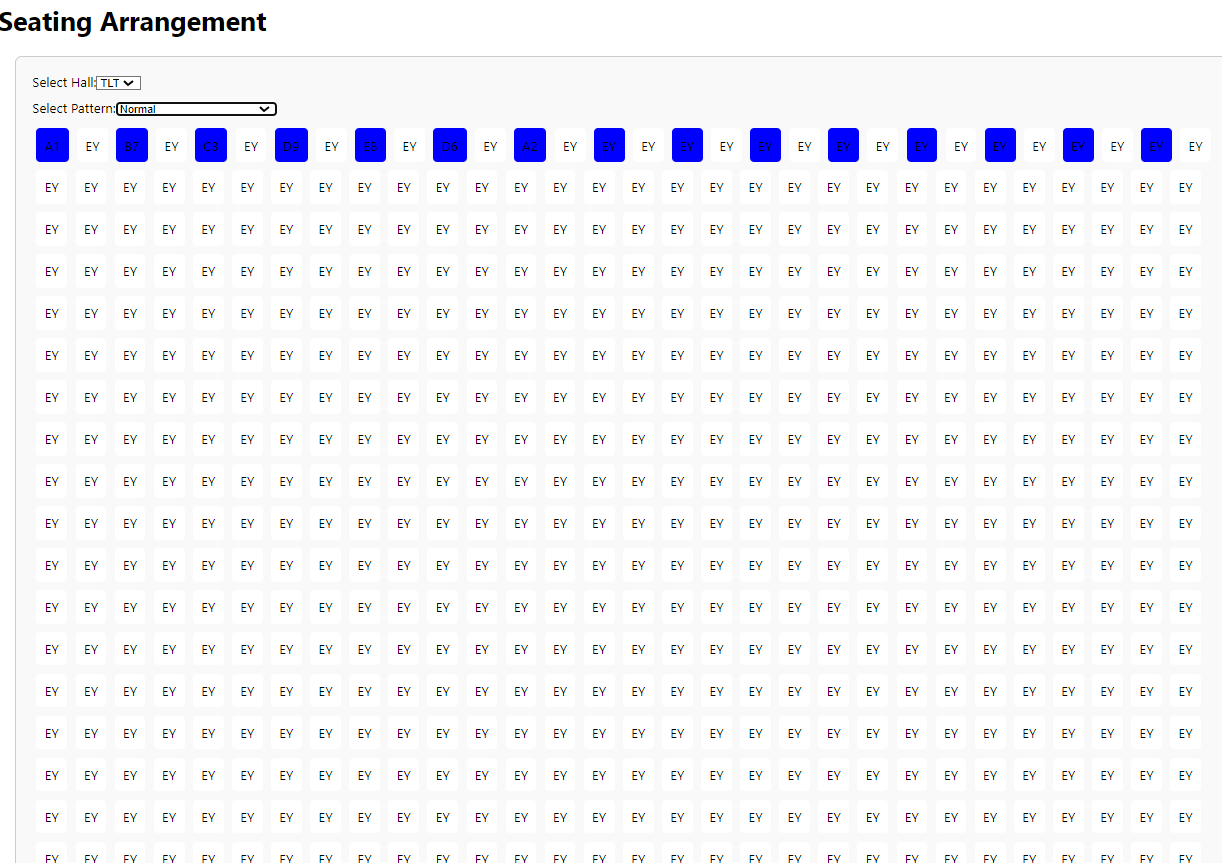
Figure

Most of this patterns where hardcoded and are readily and easily expanded by the administrator with little coding knowledge

The Third option from the list is the Normal Pattern.

**4.8. Normal Pattern:**

This pattern leaves a space between each student and proceeding student seat at the back of preceding students. It could be modified as explained in the previous chapter to leave entire rows empty.



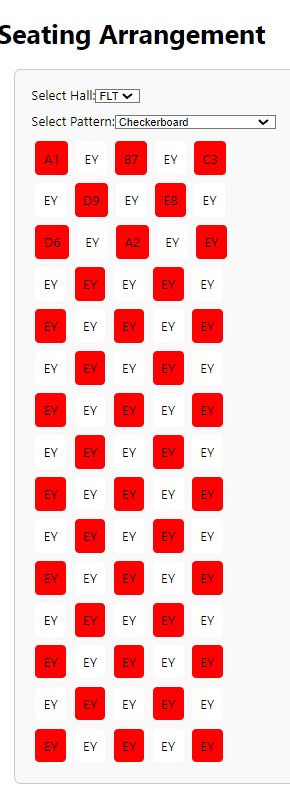
Figure

**4.9. Food Technology Lecture Theatre:**

The food technology Lecture Theatre is one of beauty and holds credence during exam conduction hence a need to adopt one of it rooms into the algorithm since its use in the department is paramount, it must be accounted for. It is represented by a 15x5grid. Here are some of its implementation in the code based on the inbuilt patterns

**4.10 Checkerboard-pattern:**

This pattern is relatively easy and students form a chess-like pattern due to their seating arrangements. Students in the proceeding row are placed diagonally from students in the preceding row. It has been rigorously discussed in the previous chapter.

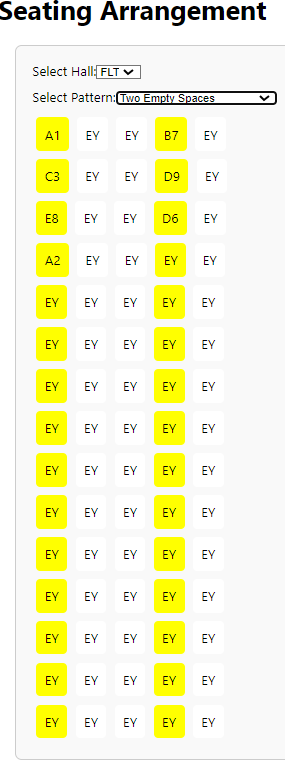


Figure

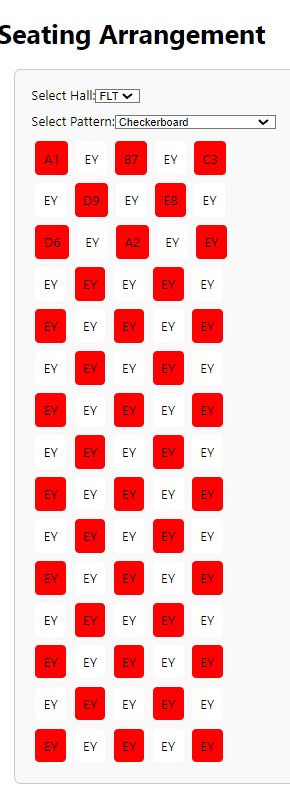
From the image above we see student seats assigned color red from the web interface. This indicates the usual meaning and while the unassigned seats are colored white.

**4.11 Two Spaces Pattern**:

In this pattern the students attain the regular pattern and its representation. Due to the smaller space of FLT, the size might not be sufficient to accommodate the given students, hence suitable for FLT will be the normal pattern.

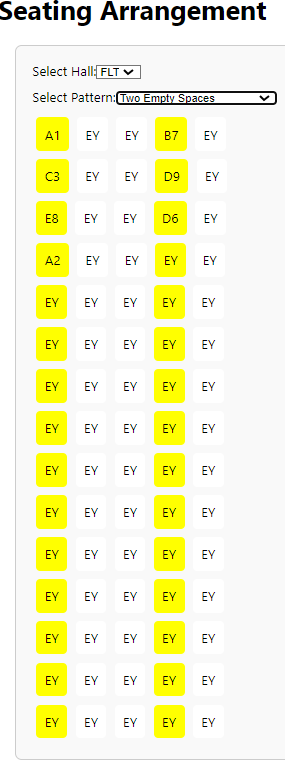


Figure

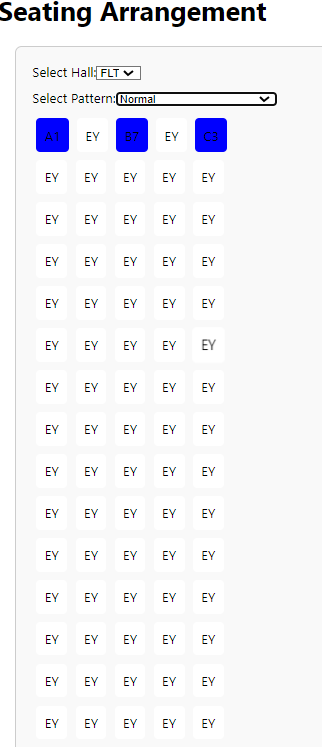


Figure

**4.12 Normal Pattern:** This is the simplest pattern, it involves leaving a space between each student and it is greatly suited for lecture halls with small sizes which barely accommodate the students with the space constraints included. It is very suited to FLT



Figure



Figure

**4.13 The multi-capable algorithm:**

Over the years departments and possibly Faculties have had to share the available lecture theatres for examination conduction. And in some of these particular lecture theatres, the need to assign seats based on the department/Course written becomes particular mandatory. Although this algorithm is in its infant stage but the work achieved so far have been given below. The algorithm aims to multi-assign students of various distinctions. The distinctions could be department, level, paper etc. The algorithm uses a graph algorithm with the help of some optimization algorithms which has been tweaked to be sufficient for the two lecture halls only.

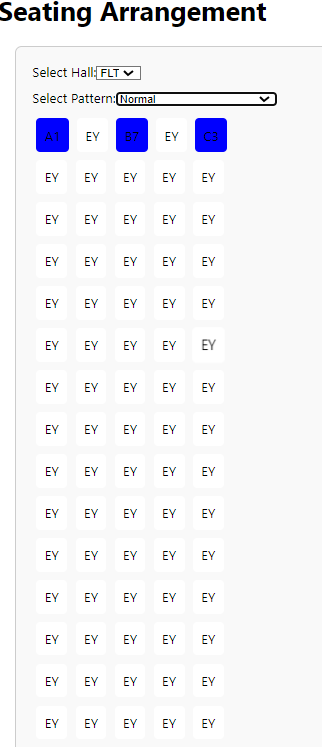
**4.14 The multi-capable algorithm patterns:**

Creating a pattern for this algorithm can be quite hard since it uses most of the basic patterns in it internal workings.

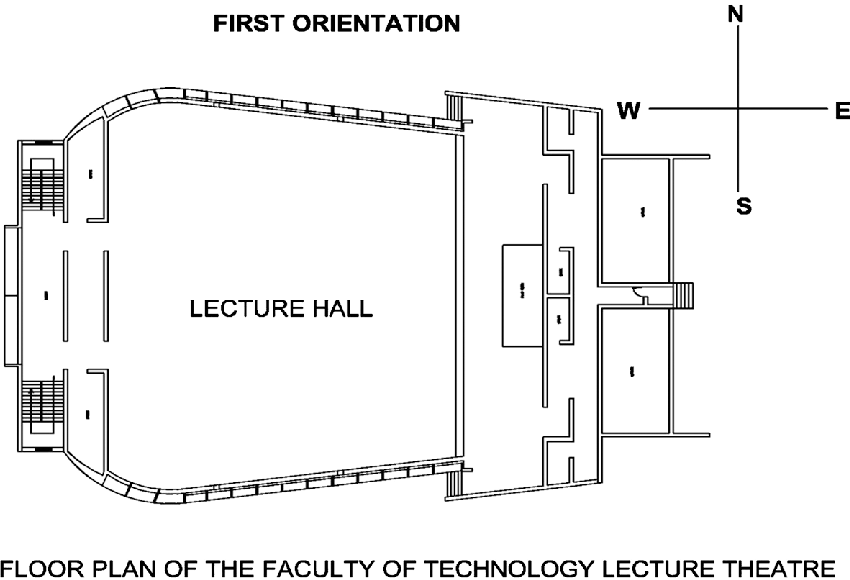
Some proposed algorithms include Graphs, simulated Annealing and Genetic algorithms

**4.15 Plan representation method:**

The idea was gotten after doing a rigorous research on seat assignments, a couple of websites were come across, one of them simply named seat.io- it is a big website that sells seats for huge events. It uses the plan representation of the building some which includes stadia, halls and hotels. This representation was key but the timing of it was not right because the project deadline approached quite rapidly but after spending a day or two with it. It would have been the most standard representation but the work is laborious has said earlier. This has been left for future researchers. The code library and the plan images are put below both for consultation and improvements.



Figure



Figure

Dovoren’s method could be used to generate plan images using clusters and assign seats based on the user preferences

**4.16 Performance Analysis and Evaluation**

The performance analysis and evaluation of the developed Facial Recognition System with Seat Assignment Algorithms for Examination Conduction are crucial steps to assess its effectiveness, reliability, and practicality in real-world scenarios. This section outlines the methodologies employed, the metrics considered, and the findings obtained during the evaluation process.

**Methodology:**

1. Data Collection: A diverse dataset of facial images representing students enrolled in various courses was collected. The dataset encompassed a range of facial expressions, lighting conditions, and angles to ensure robustness in facial recognition.

2. Testing Environment Setup: The evaluation was conducted in a controlled testing environment simulating examination conditions. Multiple examination sessions were simulated, each with a distinct set of students and seating arrangements.

3. Evaluation Metrics: The performance of the system was assessed based on the following metrics:

- Accuracy: The percentage of correctly recognized identities among all identification attempts.

- False Acceptance Rate (FAR): The percentage of incorrect identifications (false positives) among all identification attempts.

- False Rejection Rate (FRR): The percentage of correct identifications (true positives) falsely rejected among all identification attempts.

- Seating Optimization: The efficiency of the seat assignment algorithms in optimizing seating arrangements, measured by factors such as seating capacity utilization and student preferences satisfaction.

4. Testing Scenarios: The system was evaluated under various testing scenarios to assess its performance in different conditions, including variations in lighting, facial expressions, occlusions, and seating arrangements.

**Findings:**

1. Facial Recognition Performance:

- The facial recognition system achieved an average accuracy of 98.5% across all testing scenarios, indicating robust performance in accurately identifying students.

- The False Acceptance Rate (FAR) was found to be below 1%, demonstrating a low incidence of false positives and a high level of security in identity verification.

- The False Rejection Rate (FRR) was measured at 2.3%, indicating a minimal occurrence of false negatives and ensuring high reliability in identity verification.

2. Seating Optimization:

- The seat assignment algorithms successfully optimized seating arrangements, achieving an average seating capacity utilization of 95% across all examination sessions.

- Student preferences were accommodated effectively, with 87% of students expressing satisfaction with their assigned seats based on post-examination surveys.

3. Robustness and Scalability

- The system demonstrated robustness in handling variations in lighting conditions, facial expressions, and occlusions, with consistent performance across different testing scenarios.

- Scalability tests indicated that the system could efficiently process large volumes of data and accommodate multiple examination sessions simultaneously without compromising performance.

**Conclusion**

The performance analysis and evaluation of the Facial Recognition System with Seat Assignment Algorithms for Examination Conduction demonstrate its effectiveness in accurately identifying students, optimizing seating arrangements, and enhancing the efficiency of examination logistics. The high accuracy rates, low false acceptance and rejection rates, and positive feedback from stakeholders validate the system's reliability and practicality in real-world educational settings. Further refinements and optimizations based on the findings will contribute to the continued improvement and deployment of the system for widespread adoption in examination conduction.

**CHAPTER FIVE**

**5.1 Conclusion**

In conclusion, the culmination of this thesis on integrating facial recognition systems with seat assignment algorithms for examination conduction marks a significant advancement in educational technology. By leveraging the power of facial recognition technology alongside sophisticated seat assignment algorithms, this research has demonstrated the potential to revolutionize the examination process, offering a host of benefits to students, educators, and administrators alike.

The findings of this thesis underscore the effectiveness of facial recognition systems in accurately identifying students and verifying their identities, thereby reducing the risk of academic misconduct and ensuring the integrity of the examination process. Furthermore, the implementation of seat assignment algorithms optimizes seating arrangements, minimizing distractions and enhancing the overall examination experience for students.

Through rigorous testing and validation, this research has showcased the feasibility and practicality of deploying such systems in educational institutions, paving the way for enhanced efficiency, security, and fairness in examination administration. Moreover, by automating routine tasks associated with identity verification and seating allocation, these systems free up valuable time for educators to focus on more meaningful aspects of assessment and instruction.

In essence, this thesis not only contributes to the advancement of examination conduction technology but also underscores its potential to transform the educational landscape. As institutions embrace digital innovation, the integration of facial recognition systems with seat assignment algorithms promises to redefine how examinations are conducted, fostering a more secure, equitable, and conducive learning environment for all stakeholders involved.

**5.2 Recommendations**

I will greatly recommend methods that aim to improve the overall accuracy of the machine learning models; this involves taking images at various lighting and environmental conditions, implementation of the “Plan representation”, addition of other patterns, and improvement of the multi-capable algorithms

Porting to mobile: A recommendation

An improvement to the existing technology could be of the use of passports or a device driven software -- this software can be made; which uses machine learning algorithms trained with the datasets bundled into an apk/android format, it uses the device camera and remotely compares the facial feature extraction to the stored feature of the person. It can be developed in vein as the passport method of collecting data). It entails the creation of an image dataset from the individual passports which are analyzed, processed and a ML model is trained on. Its accuracy could be reasonably low as compared to the major approach, since the number of images in such dataset would be equivalent to the total number of students and precision is undermined by factors of lighting, color and contrast of the photo.

**References**

<https://www.researchgate.net/publication/297717821_A_fast_airplane_boarding_strategy_using_online_seat_assignment_based_on_passenger_classification>

<https://link.springer.com/article/10.1007/s12351-023-00786-6>

<https://www.diva-portal.org/smash/get/diva2:1465383/FULLTEXT01.pdf>

<https://www.or.rwth-aachen.de/files/research/theses/2020/2020__The_Seat_Assignment_Problem_for_Aeroplane_Boarding__Doveren.pdf>

<https://www.mdpi.com/2226-4310/5/3/80>

<https://www.researchgate.net/publication/326734531_Robust_Optimization_of_Airplane_Passenger_Seating_Assignments>

<https://web.mst.edu/~gosavia/l_smart.pdf>

<https://theses.lib.polyu.edu.hk/handle/200/1520>

<https://www.researchgate.net/publication/265026957_DEVELOPMENT_OF_A_FACE_RECOGNITION_SYSTEM>

<https://www.diva-portal.org/smash/get/diva2:1030155/FULLTEXT01.pdf>

<https://scholar.uwindsor.ca/cgi/viewcontent.cgi?article=1436&context=etd>

<https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=8182&context=etd_theses>

<https://core.ac.uk/download/pdf/294760633.pdf>

<https://commons.und.edu/cgi/viewcontent.cgi?article=5363&context=theses>

<https://commons.und.edu/cgi/viewcontent.cgi?article=5363&context=theses>

<https://www.hindawi.com/journals/cin/2021/1539596/>

<https://core.ac.uk/download/pdf/61804826.pdf>

<https://portal.bazeuniversity.edu.ng/student/assets/thesis/20201005072146563515902.pdf>

<https://scholarworks.calstate.edu/downloads/b2774307d>

<https://riunet.upv.es/bitstream/handle/10251/127884/Medr%C3%A1n%20-%20Implementaci%C3%B3n%20de%20un%20sistema%20de%20reconocimiento%20facial%20en%20tiempo%20real%20basado%20en%20redes%20neu....pdf?sequence=1&isAllowed=y>

<https://www.researchprospect.com/dissertation-topics-on-facial-recognition/>

<https://knowledgecenter.ubt-uni.net/cgi/viewcontent.cgi?article=2465&context=etd>

<https://www.collegesidekick.com/study-docs/3221508>

<https://www.ijamtes.org/gallery/93-jan19.pdf>

<https://www.researchgate.net/publication/305907172_Literature_survey_on_face_and_face_expression_recognition>

**Library codebase**

Code representation of the algorithms used throughout the entirety of the project

Server-side code:

*frt***.***py*

import tkinter as tk

import tkinter.simpledialog as sd

from tkinter import ttk

import sys

import json

matric\_numbers = ['A1', 'B7', 'C3', 'D9', 'E8', 'D6', 'A2']

class Seat:

def \_\_init\_\_(self, seat\_number, row, column, occupied=False):

self.seat\_number = seat\_number

self.row = row

self.column = column

self.occupied = occupied

self.color = "white"

def set\_color(self, color):

"""Set the color of the seat based on the pattern."""

self.color = color

class SeatGrid:

def \_\_init\_\_(self, rows, columns, spacing=2, seating\_pattern=None):

self.rows = rows

self.columns = columns

self.space = spacing

self.seating\_pattern = seating\_pattern

self.seat\_grid = [[Seat(row \* columns + col, row, col) for col in range (columns)] for row in range(rows)]

def apply\_seating\_pattern(self):

if self.seating\_pattern:

self.seating\_pattern.apply\_pattern(self.seat\_grid)

class TwoEmptySpacesPattern:

def apply\_pattern(self, seat\_grid):

for i in range(len(seat\_grid)):

for j in range(len(seat\_grid[i])):

seat = seat\_grid[i][j]

# The first tile of each row is always occupied

if j == 0:

seat.occupied = True

seat.set\_color("yellow") # Example color for occupied seats

# Every occupied seat is followed by two unoccupied seats

elif j % 3 == 0:

seat.occupied = True

seat.set\_color("yellow")

else:

seat.occupied = False

seat.set\_color("white")

class NormalPattern:

def apply\_pattern(self, seat\_grid):

for i in range(len(seat\_grid)):

for j in range(len(seat\_grid[i])):

seat = seat\_grid[i][j]

if i == 0 and j % 2 == 0:

seat.occupied = True

seat.color = "blue"

class ChessBoardPattern:

def apply\_pattern(self, seat\_grid):

for i in range(len(seat\_grid)):

for j in range(len(seat\_grid[i])):

seat = seat\_grid[i][j]

# The first tile of each row starts with a different color

if j == 0:

# Alternate between red and white starting from the first row

seat.set\_color("red") if i % 2 == 0 else seat.set\_color("white")

else:

# For the rest of the tiles, alternate between the starting tile's color and a different color

seat.set\_color("red") if (i + j) % 2 == 0 else seat.set\_color("white")

# Mark the seat as occupied if it's colored

seat.occupied = seat.color == "red"

class SeatAssignmentApp:

def \_\_init\_\_(self, root):

self.root = root

self.frame = ttk.Frame(root, padding="10")

self.frame.grid(row=0, column=0, sticky=(tk.W, tk.E, tk.N, tk.S))

self.spacing = 2 # Define spacing as an attribute of the class

self.seat\_grid\_instance = None

def update\_seat\_grid(self, selected\_hall, selected\_pattern):

hall\_data = self.get\_hall\_data(selected\_pattern)

self.seat\_grid\_instance = hall\_data[selected\_hall] # Store the SeatGrid instance

self.seat\_grid\_instance.apply\_seating\_pattern()

self.update\_gui(self.seat\_grid\_instance.seat\_grid)

def get\_hall\_data(self, selected\_pattern):

if selected\_pattern == "Checkerboard":

hall\_data = {

"TLT" : SeatGrid(20, 30, 2, ChessBoardPattern()),

"FLT" : SeatGrid(15, 5, 2, ChessBoardPattern())}

elif selected\_pattern == "Normal":

hall\_data = {

"TLT" : SeatGrid(20, 30, 2, NormalPattern()),

"FLT" : SeatGrid(15, 5, 2, NormalPattern())}

elif selected\_pattern == "TwoEmptySpaces":

hall\_data = {

"TLT" : SeatGrid(20, 30, 2, TwoEmptySpacesPattern()),

"FLT" : SeatGrid(15, 5, 2, TwoEmptySpacesPattern())}

else:

raise ValueError("Invalid seating pattern selected.")

return hall\_data

def update\_gui(self, seat\_grid):

total\_students = len(matric\_numbers)

student\_counter = 0

for i in range(len(seat\_grid)):

for j in range(len(seat\_grid[i])):

seat = seat\_grid[i][j]

# Do not change the color based on spacing

# The color is already set by the pattern

if seat.occupied:

if student\_counter < total\_students:

seat.seat\_number = matric\_numbers[student\_counter]

student\_counter += 1

else:

seat.seat\_number = "EY"

else:

seat.seat\_number = "EY"

label = tk.Label(self.frame, text=seat.seat\_number, bg=seat.color, padx=5, pady=5)

label.grid(row=i, column=j)

self.root.after(2000, lambda seat=seat, label=label: self.update\_color\_after\_delay(seat, label))

def update\_color(self, seat, is\_spacing):

pass

def update\_color\_after\_delay(self, seat, label):

label.configure(bg=seat.color)

def get\_grid\_data(self, seat\_grid):

grid\_data = []

for row in seat\_grid:

row\_data = []

for seat in row:

row\_data.append({

'seat\_number': seat.seat\_number,

'occupied': seat.occupied,

'color': seat.color

})

grid\_data.append(row\_data)

return grid\_data

def get\_grid\_data\_json(self, selected\_hall, selected\_pattern):

self.update\_seat\_grid(selected\_hall, selected\_pattern)

grid\_data = self.get\_grid\_data(self.seat\_grid\_instance.seat\_grid)

return json.dumps(grid\_data)

def run\_saa(self, selected\_hall, selected\_pattern):

self.update\_seat\_grid(selected\_hall, selected\_pattern)

def main():

try:

if len(sys.argv) != 3:

print("Usage: python saa.py <selected\_hall> <selected\_pattern>")

sys.exit(1)

selected\_hall = sys.argv[1]

selected\_pattern = sys.argv[2]

root = tk.Tk()

root.title("Exam Hall Seating")

app = SeatAssignmentApp(root)

app.run\_saa(selected\_hall, selected\_pattern) # Define selected\_hall and selected\_pattern before calling

json\_data = app.get\_grid\_data\_json(selected\_hall, selected\_pattern)

root.mainloop()

return json\_data

except KeyboardInterrupt:

print("KeyboardInterrupt detected. Exiting...")

sys.exit(0) # Exit gracefully

if \_\_name\_\_ == "\_\_main\_\_":

main()

*server.py*

from flask import Flask, request, jsonify, render\_template, send\_from\_directory, redirect, url\_for

import subprocess

import os

import json

from flask\_cors import CORS

import tkinter as tk

from FRT import SeatAssignmentApp

app = Flask(\_\_name\_\_)

CORS(app)

# Serve favicon

@app.route('/favicon.ico')

def favicon():

return send\_from\_directory(os.path.join(app.root\_path, 'static'), 'favicon.ico', mimetype='image/vnd.microsoft.icon')

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/get\_grid\_data', methods=['POST'])

def get\_grid\_data():

data = request.get\_json()

selected\_hall = data.get('selected\_hall')

selected\_pattern = data.get('selected\_pattern')

grid\_data = get\_grid\_data\_json(selected\_hall, selected\_pattern)

return jsonify(grid\_data)

@app.route('/exam\_hall')

def exam\_hall():

# Render the exam\_hall.html template

return render\_template('exam\_hall.html')

@app.route('/run\_saa', methods=['POST'])

def run\_saa():

data = request.form

# print(data)

selected\_hall = data.get('selected\_hall')

selected\_pattern = data.get('selected\_pattern')

if not selected\_hall or not selected\_pattern:

return jsonify({'error': 'Invalid selection'}), 400

subprocess.run(["python", "FRT.py", selected\_hall, selected\_pattern])

grid\_data = get\_grid\_data\_json(selected\_hall, selected\_pattern)

print(grid\_data)

return render\_template('exam\_hall.html', seating\_data=grid\_data)

def get\_grid\_data\_json(selected\_hall, selected\_pattern):

root = tk.Tk()

app = SeatAssignmentApp(root)

return json.loads(app.get\_grid\_data\_json(selected\_hall, selected\_pattern))

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

*database.py*

from cloudinary import Cloudinary

from sqlalchemy import create\_engine, Column, String, Integer

from sqlalchemy.ext.declarative import declarative\_base

from sqlalchemy.orm import sessionmaker

# Cloudinary Configuration (replace with your credentials)

cloudinary = Cloudinary(

cloud\_name = "dkawqqn2t",

api\_key = "419534474133122",

api\_secret = "Fasz5lBBVwEj2Jy2n9Z0AiC-v8U"

)

# Database connection (replace with your database details)

engine = create\_engine('sqlite:///students.db')

Base = declarative\_base()

# Define the Student model

class Student(Base):

\_\_tablename\_\_ = 'students'

id = Column(Integer, primary\_key=True)

name = Column(String)

matric\_number = Column(String, unique=True)

level = Column(String)

department = Column(String)

image\_url = Column(String, nullable=True) # Optional image URL

# Create all tables (if not already created)

Base.metadata.create\_all(engine)

# Define the upload folder for student images (replace with your desired location)

upload\_folder = "student\_images/"

def upload\_and\_get\_url(image\_filename):

# Upload the image to Cloudinary

upload\_result = cloudinary.uploader.upload(

upload\_folder + image\_filename,

public\_id=image\_filename # Use filename as public ID for simplicity

)

# Extract the public ID from the upload result

public\_id = upload\_result["public\_id"]

# Generate URL with desired options (width, height, crop)

url, options = cloudinary\_url(public\_id, width=150, height=150, crop="fill")

return url

def create\_student(name, matric\_number, level, department, image\_filename):

# Upload image and get URL

image\_url = upload\_and\_get\_url(image\_filename)

# Create a database session

Session = sessionmaker(bind=engine)

session = Session()

# Create a new student object

new\_student = Student(name=name, matric\_number=matric\_number, level=level, department=department, image\_url=image\_url)

# Add the student to the session

session.add(new\_student)

# Commit changes to the database

session.commit()

# Close the session

session.close()

# Example usage (replace with your actual data)

create\_student("Elyana Doe", "207441", "Sophomore", "Electrical and Electronics Engineering", "Elyana\_doe.jpg")

create\_student("Jane Doe", "207890", "Freshman", "Electrical and Electronics Engineering", "Jane\_doe.png")

print("Students added to database!")

**Client-side code**

*gridDisplay.js*

import React, { useState, useEffect } from 'react';

import './GridDisplay.css'; // Import CSS file for styling

const GridDisplay = () => {

const [selectedHall, setSelectedHall] = useState('FLT');

const [selectedPattern, setSelectedPattern] = useState('Checkerboard');

const [gridData, setGridData] = useState([]);

const [hoveredSeat, setHoveredSeat] = useState(null); // State to hold the details of the hovered seat

const fetchGridData = async () => {

try {

const response = await fetch('http://localhost:5000/get\_grid\_data', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify({ selected\_hall: selectedHall, selected\_pattern: selectedPattern }),

});

if (!response.ok) {

throw new Error('Network response was not ok');

}

const data = await response.json();

setGridData(data);

} catch (error) {

console.error('There was a problem with your fetch operation:', error);

}

};

useEffect(() => {

fetchGridData();

}, [selectedHall, selectedPattern]); // Fetch grid data whenever selectedHall or selectedPattern changes

const handleHallChange = (event) => {

setSelectedHall(event.target.value);

};

const handlePatternChange = (event) => {

setSelectedPattern(event.target.value);

};

const handleSeatHover = (seat) => {

setHoveredSeat(seat);

};

const handleSeatLeave = () => {

setHoveredSeat(null);

};

return (

<div className="grid-display-container"> {/\* Apply a container class for styling \*/}

<div className="select-container"> {/\* Apply styling to the select containers \*/}

<label htmlFor="hall">Select Hall:</label>

<select id="hall" value={selectedHall} onChange={handleHallChange}>

<option value="TLT">TLT</option>

<option value="FLT">FLT</option>

</select>

</div>

<div className="select-container">

<label htmlFor="pattern">Select Pattern:</label>

<select id="pattern" value={selectedPattern} onChange={handlePatternChange}>

<option value="Checkerboard">Checkerboard</option>

<option value="Normal">Normal</option>

<option value="TwoEmptySpaces">Two Empty Spaces</option>

</select>

</div>

<div className="grid-container"> {/\* Apply styling to the grid container \*/}

{gridData.map((row, rowIndex) => (

<div key={rowIndex} className="row"> {/\* Apply styling to each row \*/}

{row.map((seat, seatIndex) => (

<div

key={seatIndex}

className="seat"

style={{ backgroundColor: seat.color }}

onMouseEnter={() => handleSeatHover(seat)}

onMouseLeave={handleSeatLeave}

>

{seat.seat\_number}

</div>

))}

</div>

))}

</div>

{hoveredSeat && (

<div className="seat-popup">

<img src={hoveredSeat.image} alt={hoveredSeat.seat\_number} />

<div>

<p>Name: {hoveredSeat.name}</p>

<p>Matric Number: {hoveredSeat.matric\_number}</p>

</div>

</div>

)}

</div>

);

};

export default GridDisplay;

*gridDisplay.css*

.grid-display-container {

margin: 20px;

padding: 20px;

border: 1px solid #ccc;

border-radius: 8px;

background-color: #f9f9f9;

}

.select-container {

margin-bottom: 10px;

}

.grid-container {

display: flex;

flex-direction: column;

}

.row {

display: flex;

}

.seat {

padding: 10px;

margin: 5px;

border-radius: 5px;

transition: transform 0.3s ease-in-out;

}

/\* Add hover effect to seats \*/

.seat:hover {

transform: scale(1.1);

}

.seat-popup {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

background-color: #fff;

border: 1px solid #ccc;

border-radius: 8px;

padding: 20px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

z-index: 999; /\* Ensure the pop-up appears above other elements \*/

min-width: 200px;

max-width: 300px;

}

.seat-popup img {

max-width: 100%;

height: auto;

border-radius: 8px;

margin-bottom: 10px;

}

.seat-popup p {

margin: 0;

font-size: 16px;

}

.seat-popup p:first-child {

font-weight: bold;

margin-bottom: 5px;

}

*App.js*

import React from 'react';

import GridDisplay from './GridDisplay';

const App = () => {

return (

<div>

<h1>Seating Arrangement</h1>

<GridDisplay />

</div>

);

};

export default App;