2025 ASEE Northeast Section Conference, March 22, 2025, University of Bridgeport, Bridgpeort, CT, USA.

Automated Queueing System using Facial Recognition

Applying AI and Computer Vision to Queue Automation

Zebin Pepin
Department of Electrical and Computer Engineering
Wentworth Institute of Technology
Boston, Massachusetts

Douglas E. Dow
Department of Electrical and Computer Engineering
Wentworth Institute of Technology
Boston, Massachusetts

***Abstract*— Theme parks, concerts, and other large-crowd events often experience long queues that contribute to visitor dissatisfaction. This dissatisfaction arises not only from extended wait times but also from inaccurate wait-time estimates and perceived line-cutting incidents. This project aimed to develop and test prototype modules for an automated queuing system leveraging facial recognition to address these concerns. The system automates wait-time calculations, verifies individual positions in the queue, and detects queue violations. The design consists of three key phases: data collection, data validation, and resolution. Entry timestamps are recorded when individuals enter the queue, verified at an intermediate checkpoint, and finalized at exit to calculate total wait time. The system tracks individuals in real-time, identifying discrepancies such as line-cutting. Testing results demonstrated improved accuracy over manual queue monitoring, indicating potential for real-world implementation. Further refinement is necessary to enhance reliability across diverse operational environments. If fully developed, such an automated queuing system could significantly improve user experience by reducing frustration and enhancing fairness.**

Keywords— Facial recognition, queue automation, biometric identification, wait-time estimation, computer vision, machine learning, crowd management, real-time tracking.

I. Introduction

In large-scale events such as theme parks, concerts, and sports arenas, managing visitor flow is a significant logistical challenge. Inefficiencies in queue management lead to substantial visitor dissatisfaction, as lengthy wait times and inaccurate estimates of queue duration contribute to frustration. Additionally, queue violations such as line-cutting undermine the perceived fairness of the system.

Traditional queue management solutions, including manual monitoring, ticketing systems, and basic electronic monitoring, are insufficient in addressing these challenges. These methods are often labor-intensive, error-prone, and lack adaptability to real-time conditions. As event venues continue to increase in scale and complexity, a more sophisticated automated queue management solution is required to dynamically adjust queue regulation and improve fairness.

Numerous strategies have been implemented over the years to manage queues more efficiently and improve visitor satisfaction. However, each of these methods comes with its own set of limitations. In many cases, staff members are employed to monitor queues, ensuring that lines move smoothly and that no one cuts in line. While this approach can be effective in small settings, it is labor-intensive and prone to human error. The presence of staff may deter some individuals from cutting in line, but it cannot entirely prevent or accurately track such behavior, especially in large crowds [5]. For another method, some venues have adopted ticketing systems that assign visitors to specific time slots. This approach can help distribute visitors evenly throughout the day and reduce peak-time congestion. However, it does not account for real-time changes in queue dynamics, such as delays caused by technical issues or fluctuations in the number of visitors arriving at different times. Visitors often find it frustrating when the actual wait time deviates significantly from the time indicated by the ticket. For a third method, virtual queueing systems. Visitors join a queue through an app and are notified when it is their turn, which has gained popularity in recent years. While this approach allows visitors to enjoy other activities while waiting, it is not without flaws. The virtual queue system requires reliable internet connectivity, which may not always be available in certain areas. Additionally, these systems may struggle to maintain fairness if visitors are not present when their turn arrives or if they manipulate the system by joining multiple queues simultaneously [1, 7]. A fourth method that is suitable, GPS-based Tracking Systems. In outdoor events, GPS-based tracking systems can monitor the movement of visitors within the venue and provide real-time updates on queue lengths and wait times. However, the accuracy of GPS tracking diminishes

in indoor environments or areas with poor satellite visibility. Furthermore, these systems cannot prevent or detect line-cutting, as they are not designed to monitor individual location and behavior at such a fine-tuned level [8]. For a fifth method, Radio Frequency Identification (RFID) Technology. RFID technology has been used in some advanced queuing systems. Visitors carry RFID tags that allow the system to track their movements through various checkpoints. While RFID can provide more accurate data than GPS, it requires visitors to always carry a tag, which can be inconvenient. Visitors can exchange RFID tags complicating accurate tracking practices. Additionally, the system's ability to detect and prevent line-cutting is limited [9].

Given the limitations of existing methods, there is a clear need for a more advanced, automated solution that can dynamically respond to the real-time conditions of a queue [10]. An ideal system would not only provide accurate, real-time wait time estimations, but also ensure fairness by detecting and preventing line cutting [1].

The development of facial recognition technologies offers a promising avenue for addressing these challenges. Unlike traditional methods, facial recognition can track individuals throughout their journey in a queue without requiring them to carry any additional devices or tags [1]. By analyzing facial features, the system can maintain an accurate log of each individual's position in the queue, detect any deviations from the expected order, and calculate wait times based on actual movement through the queue [3]. Moreover, the integration of facial recognition with multi-factor authentication and object tracking technologies can further enhance the system's accuracy and reliability. This combination allows the system to verify individuals at multiple checkpoints, reducing the likelihood of errors and ensuring that each person is correctly identified throughout their queuing experience.

The objective of this project was to design, develop, and test a prototype for an automated queueing system that leverages facial recognition technology. The system aims to do the following. Accurately track individuals as they move through a queue, ensuring that their position is maintained and that any deviations are detected promptly. Provide real-time, automated wait time calculations that reflect the current conditions of the queue. This includes accounting for variations in queue length, processing times, and any disruptions such as line cutting. Detect and address instances of line-cutting by comparing the order in which individuals enter and move through the queue with their expected positions. The system should flag potential line-cutters and alert staff for further action.

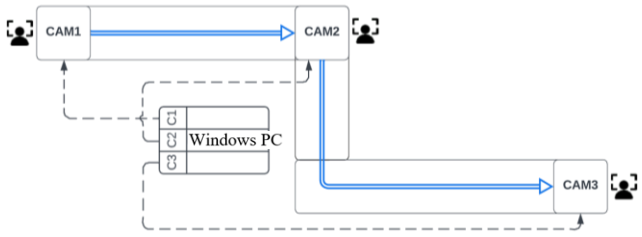
The implementation of these objectives should improve the overall visitor experience by reducing the frustration associated with long wait times and perceived unfairness in the queuing process.

II. Materials and Methods

A. *System Overview*

The overall system was designed to utilize facial recognition for real-time monitoring and tracking of individuals within a queue. The system's primary architecture has as a multi-layered software approach consisting of three key stages as shown in Fig I. The three stages are, data collection, data validation, and resolution. The system captured facial images of individuals entering the queue. This was done through cameras placed at entry points. One challenge in this stage was ensuring accurate facial detection in varied lighting conditions and different angles. To overcome this, multiple cameras from different viewpoints could be used with associated algorithms for face detection and recognition. Once data was collected for each visitor entering the queue, it was validated to ensure the individual remained in the correct position within the queue. This involved recognizing the same face at multiple checkpoints within the queue and comparing their order of appearance with the initial data. This stage required the integration of robust algorithms capable of handling varying distances and movements. The final stage involved analyzing the data collected and validated to detect any discrepancies such as line cutting. If line cutting was suspected, an appropriate response was made such as alerting staff or adjusting the wait time estimations for the affected individuals.

Figure I. Queue Camera Placement

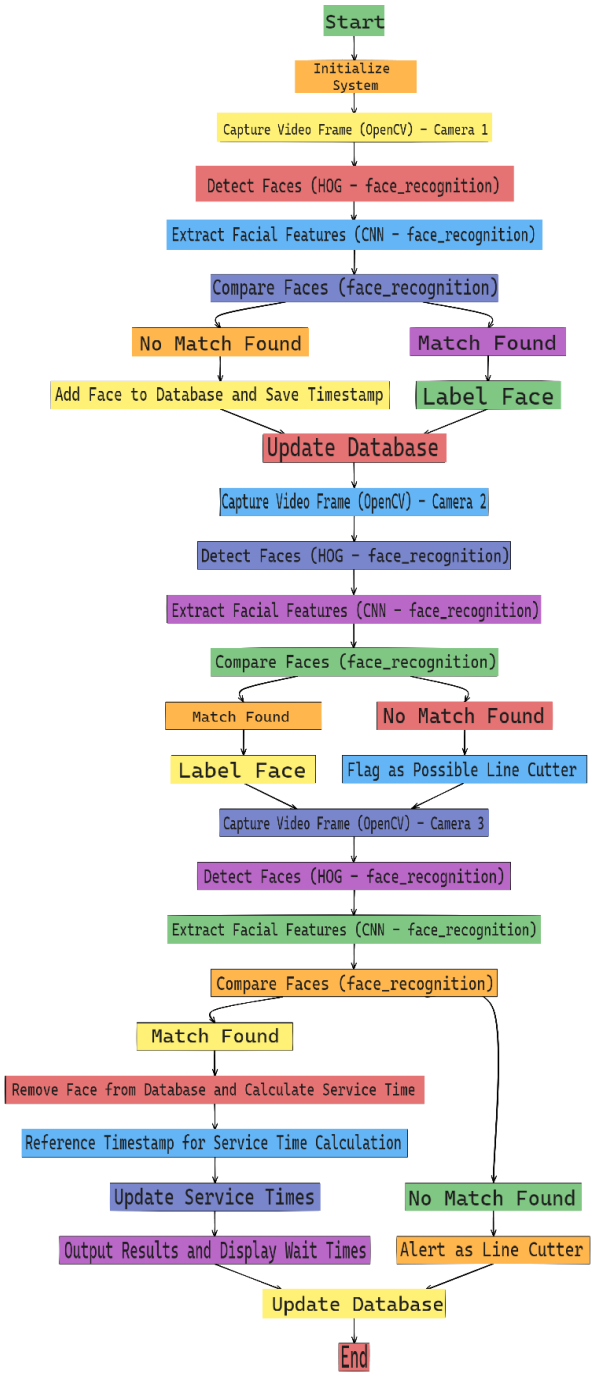


[a] Placement is based on optimal facial recognition angles to ensure accurate tracking throughout the queue.

Fig. 1. A diagram showing the strategic placement of cameras in the queue to track individuals, detect line-cutting, and automate wait-time calculations.

In Figure I, each box labeled ‘CAMX’ represents a camera in the system along the queue line. Arrows simulate the flow of traffic through the queue path. All cameras in the prototype were linked centrally to the Windows PC running all 3 cameras simultaneously. Figure I is a flowchart representing the process of the automated queueing system. The flow includes steps for labeling faces, updating the database, and calculating service times. It also featured mechanisms to flag possible line cutters and remove faces from the database after the service. This helped to ensure the calculation of accurate wait time estimations for visitors. The output included a display of wait times and any alerts for detected issues, such as line cutting.

FIGURE II. AUTOMATED QUEUEING DESIGN FLOW



b. The workflow consists of sequential phases: data collection, validation, and resolution to ensure queue integrity.

Fig. 2. A diagram illustrating the automated queue management process, from facial recognition at entry to final wait-time calculation and queue violation detection.

B. Facial Recognition Technology

The core of the system design relied on facial recognition technology. Two primary methods were considered: Histogram of Oriented Gradients (HOG) [6] and Convolutional Neural Networks (CNN) [6]. HOG is a feature descriptor used in computer vision for object detection. It is considered to be particularly effective in detecting faces in images. The system used the HOG method due to its efficiency and accuracy in real-time applications. HOG works by capturing the gradients or edges within an image and using these to create a feature vector that represents the object (in this case, a face). CNN is a class of deep neural networks that is considered to be effective in image recognition tasks. The design incorporated CNNs to enhance the accuracy of face recognition, especially in challenging conditions like varied lighting and occlusions. CNNs were used to process the facial images, learning to identify distinguishing features automatically.

To further enhance the system's accuracy and reliability. The identity of individuals at multiple points within the queue was monitored. Sample images of the same person were correlated with one another. A new face or identified face detected far ahead of where it should have been in the queue was identified and alerted as a suspected line cutter.

C. Facial Rec Hardware Configuration

The prototype required a combination of cameras, processing units, and storage devices.

The prototype used three cameras placed at strategic points within the queue: the entry point (Camera 1), an intermediate checkpoint (Camera 2), and the exit point (Camera 3) as shown in Fig I. These cameras were connected to a processing unit that handled the facial recognition and tracking tasks, a Razer Blaze 15 Advanced Model Laptop, running PyCharm. The cameras used in the prototype were as follows: (Camera 1) Microsoft LifeCam Studio 1080p HD Webcam, (Camera 2) Logitech Brio 4k Webcam, and (Camera 3) Logitech C615 1080p Webcam.

Processing unit, the facial recognition and tracking tasks were performed using a Windows PC unit capable of running the image processing algorithms. The system was built using a Python-based environment, utilizing libraries including OpenCV (OpenCV.org) for image processing.

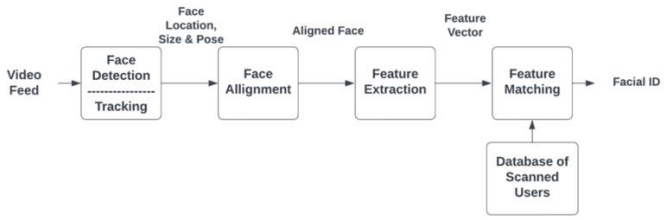
To handle the data captured by the cameras, the system included both temporary storage for real-time processing and permanent storage for logging and offline analysis. The data was managed using a local database that stored facial recognition data, timestamps, and service times to a USB flash memory storage unit.

D. Software Implementation

The software was implemented in Python, specifically on PyCharm, with several key components. Facial Recognition Module: The system's facial recognition was built using a pretrained face_recognition library [11], which simplified the process of detecting and encoding faces in images. This module handled the initial detection of faces from the video images that were captured. The module was also used at the

entry point and at the subsequent recognition checkpoints. Figure II shows a flow chart of the image processing identification algorithm. The process began with face detection and tracking, where the system identified the location, size, and pose of faces within the video. The identified face was then aligned to standardize its orientation before moving to feature extraction. In feature extraction, unique facial characteristics were extracted and converted into a feature vector. This vector was compared to a database of scanned faces of visitors in the feature matching step to identify or verify the individual, resulting in a facial ID output.

FIGURE III. FACIAL RECOGNITION PROCESSING FLOW



c. The system processes video input in sequential stages, converting raw footage into identifiable facial encodings for queue tracking.

Fig. 3. A flowchart illustrating the facial recognition pipeline, from initial video capture to feature extraction, encoding, and identity matching for queue management.

The queue management algorithm was responsible for calculating the estimated wait times based on the number of people in the queue and their movement. The algorithm took into account the rolling average of service times, adjusting the estimations dynamically as new data was collected. The algorithm utilized three primary equations: throughput, rolling average service time, and estimated wait time.

FIGURE IV. ROLLING AVERAGE SERVICE TIME EQUATION

$$T = \frac{N}{\sum_{i=1}^{N} t_i}$$

d. The equation represents system throughput, calculated as the total number of individuals processed divided by the sum of their wait times.

Fig. 4. Throughput equation used to determine the efficiency of the queue processing system, where T is throughput, N is the number of individuals, and t_i represents individual wait times.

FIGURE V. THROUGHPUT EQUATION FOR QUEUE ANALYSIS

$$\bar{t} = \frac{\sum_{i=N-M+1}^{N} t_i}{M}$$

e. This equation calculates the rolling average service time by taking the mean of the most recent M service times.

Fig. 5. Equation for rolling average service time, where t represents the average service time over the last M individuals, and t_i denotes individual service times.

FIGURE VI. ESTIMATED WAIT TIME CALCULATION

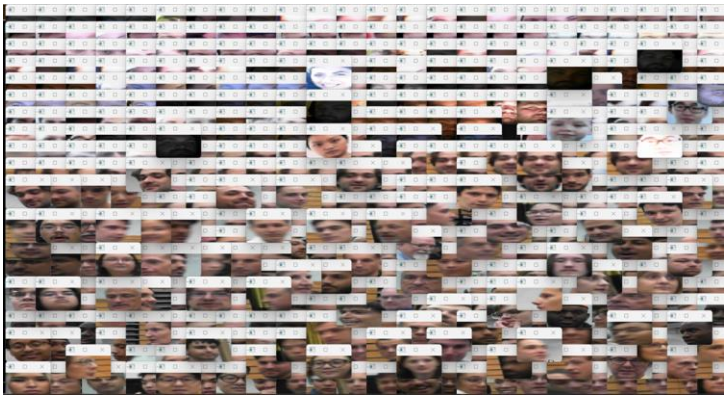
$$W = (L \times \bar{t}) + \max\left(0, C - (T_{\text{current}} - T_{\text{last\_exit}})\right)$$

f. This equation determines the estimated wait time based on queue length, rolling average service time, and remaining countdown duration.

Fig. 6. Equation for estimating wait time W, where queue length L, rolling average service time t, and countdown duration C contribute to the final wait time estimation.

A crucial part of the software algorithm was the ability to detect line-cutting. The system compared the order of individuals at each checkpoint with their initial order of entry. If discrepancies were detected (within 5 people), the system flagged potential line-cutters and sent an alert to the management interface.

FIGURE VII. CAPTURED IMAGES FROM HIGH VOLUME TESTCASE

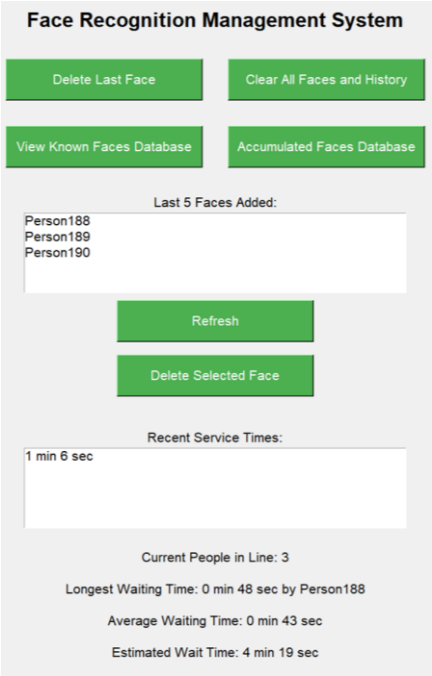


g. Illustrates system performance when processing over 450 captured images, including some duplicates resulting from queue resets and system restarts.

Fig. 7. High-volume dataset showcasing over 450 captured images used for real-time facial recognition analysis. Some duplicates appear due to queue clearing and system restarts, reflecting the system's dynamic processing in real-world scenarios.

The prototype featured a user-friendly GUI for both management and guest information. The management GUI provided a real-time overview of the queue, including the number of people in line, estimated wait times, and alerts for potential line-cutters. The interface allowed management to view detailed information about each person in the queue, including their entry and exit times. The management GUI also allowed for database management such as deleting or adding guests, as well as viewing most recent and longest waiting times.

FIGURE VIII. MANAGEMENT SYSTEM GUI



h. The interface provides real-time updates on facial recognition processes, enabling efficient queue management and data control.

Fig. 8. The graphical user interface (GUI) for the Face Recognition Management System, displaying options for managing facial data, refreshing records, and monitoring queue statistics such as estimated wait time and recent service times.

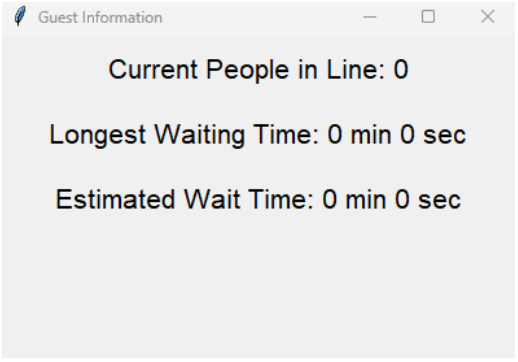
FIGURE IX. DETECTION OF POTENTIAL LINE CUTTER

i. The detection is based on facial recognition timestamps and movement tracking.

Fig. 9. An individual flagged as a potential line cutter based on system analysis. The system monitors queue order and identifies anomalies where a person may have skipped ahead.

The guest-facing GUI displayed relevant information such as the current wait time and queue length. This interface was designed to be simple and easy to read, ensuring that guests could quickly understand their expected wait times. After estimated wait time is calculated with throughput and taking into account the rolling average service time of the last 5 people, the wait time is displayed to an external monitor that is guest facing. Figures X and XI are the guest facing GUI.

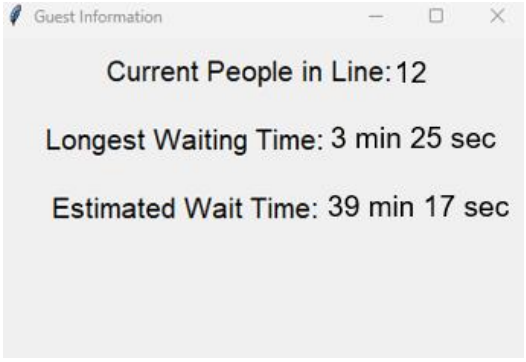
FIGURE X. GUEST FACING GUI: EMPTY QUEUE



j. This interface dynamically updates as guests enter the system, transitioning to an active queue display when a wait is expected.

Fig. 10. The guest-facing interface when no guests are currently in the queue. The screen prominently displays a message indicating that there is no wait, allowing guests to proceed immediately. Other elements, such as recent service times and estimated wait times, remain inactive until the queue becomes populated.

FIGURE XI. GUEST FACING GUI: ACTIVE QUEUE



k. The estimated wait time is computed using the throughput equation and rolling average service time, ensuring guests have real-time updates on their expected wait duration.

Fig. 11. The estimated wait time is computed using the throughput equation and rolling average service time, ensuring guests have real-time updates on their expected wait duration.

III. TESTING AND RESULTS

The final stage of prototype development involved a series of tests designed to validate the functionality, accuracy, and reliability of the prototype modules for an automated queuing system. Both individual components and modules were tested to evaluate functionality. Two sets of testing were done, sections 1) Initial Testing (pre-showcase) and 2) Field Testing (showcase presentation). This allowed for testing in controlled and uncontrolled environments to evaluate system performance.

A. *Initial Testing: Functional Testing – Video Capture*

The video capture functionality was tested to ensure that the cameras provided clear and consistent video feeds under varying conditions. The tests focused on validating the system's ability to maintain high-quality video streams essential for subsequent processing. This was tested by attaching all three cameras into the Microsoft PC and ran the program to ensure all windows were active at once.

B. *Initial Testing: Functional Testing – Face Detection*

Face detection was evaluated by assessing the system's ability to accurately outline and identify faces within video frames. This testing was not performed with respect to facial encodings; the testing performed was strictly to assist in optimal camera placement within the system derived from Figure XII data. Testing included capturing varying angles of facial orientation. A small algorithm was utilized to give facial detection confidence values on detecting the face within frame. A face was shown in four different positions 3 feet away from the camera 10 different times. Front Image (FI), looking directly into camera iris. Image looking down (ILD), chin to chest, looking directly into camera iris. Image looking up (ILU), head tiled until resistance, looking directly into camera iris. Imagine horizontally (IH), head swiveled both left and right, 5 each, looking direct into camera iris. Each test was averaged and found in early testing, overall, in every position, the camera detected a face with 53.51% accuracy. The low accuracy percentage in initial testing was likely due to suboptimal camera placement, as the test was primarily conducted to determine the best positioning for facial detection. Additionally, extreme head positions, such as looking up or down, may have introduced occlusions and distortions that made it difficult for the algorithm to consistently detect faces.

FIGURE XII. INITIAL TESTING: FACIAL ORIENTATION CONFIDENCE

Data Set	Confidence
Front Image (FI)	68.02%
Image Looking Down (ILD)	51.33%
Image Looking Up (ILU)	49.11%
Image Horizontally (IH)	45.59%
Average	53.51%

[l] Early testing focused solely on face detection without considering facial encodings. The results informed optimal camera placement and system adjustments to improve overall detection accuracy.

Fig. 12. The table illustrates the confidence percentages of face detection based on different head positions relative to the camera. The highest accuracy was observed when the subject faced directly forward, while significant drops occurred when the head was tilted or turned.

FIGURE XIII. INITIAL TESTING: FACIAL ORIENTATION OPTIMAL ANGLE DETERMINATION

[k] This figure illustrates the various head positions tested to evaluate face detection accuracy, including front-facing, looking down, looking up, and horizontal orientations.

Fig. 13. Initial testing revealed that face detection accuracy was highly dependent on the angle of orientation, with front-facing images yielding the highest confidence scores (68.02%) and horizontal orientations producing the lowest (45.59%). These results informed optimal camera placement in the system

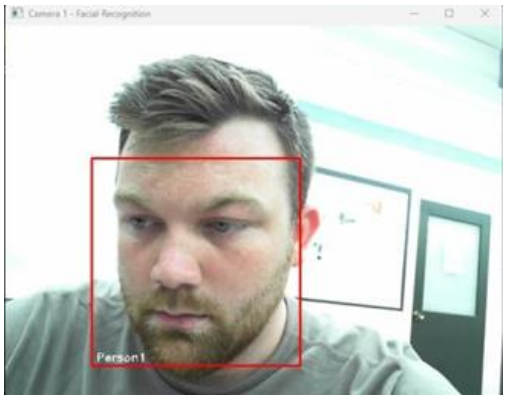
C. *Initial Testing: Functional Testing – Face Encoding*

The encoding process was tested to confirm that facial features were accurately represented as unique identifiers for recognition tasks. These tests validated that the system could reliably encode key facial features necessary for accurate recognition. Facial encoding testing was performed by showing 5 different faces to the system, assuring each were identified independently with a unique identifier.

D. Initial Testing: Functional Testing – Face Recognition

Facial recognition was tested by matching detected faces against a database of known individuals. The focus was on ensuring that the system could accurately identify individuals, even under varied conditions, and reliably match them to stored profiles. These tests were performed after data collection. Once a face was assigned a unique identifier and updated to the database, persons were taken out of frame and brought back to assuring each unique persons were recognized correctly from the known database. Figures XIV, XV and XVI support correct functionality for the facial encoding and recognition modules.

FIGURE XIV. SUCCESSFUL FACIAL RECOGNITION AND IDENTIFICATION



l. This figure demonstrates the system accurately recognizing a previously stored face and correctly labeling it as "Person 1" in the live video feed.

Fig. 14. Facial recognition functionality was verified by matching detected faces to stored encodings within the database. The system successfully identified "Person 1," confirming its ability to reference known individuals in real time.

E. Initial Testing: Integration Testing – Hardware/Software Fusion

Integration tests were performed to ensure that the hardware (cameras) and software (facial recognition algorithms) components interacted smoothly. The system was monitored for processing errors and lag, ensuring that real-time video feeds were accurately processed. Testing for hardware/software interaction was primarily stress testing. The system was shown up to 15 people at a time to test whether the processing power was enough to 1) detect 15 people at one time, and 2) process and recognize 15 people in real time. The system worked well with no faults and freezing.

F. Initial Testing: Integration Testing – Database Synchronization

The synchronization between the facial recognition system and the database was tested to ensure that new and updated information was accurately reflected in real-time. The system was monitored for its ability to handle database updates promptly and maintain accurate records. Database synchronization was primarily a guess and check scenario in testing. Once the management GUI was developed, the database was available to be displayed live and edited without disrupting the program.

FIGURE XV. STORED FACIAL ENCODINGS IN THE DATABASE

id	name	face_encoding
1	Zebin	0,0,0,64,139,47,174,191,0,0,0,0,117,134,113,63,0,0,
2	Person2	0,0,0,64,38,162,174,191,0,0,0,224,157,30,181,63,0,(
3	Person3	0,0,0,160,111,193,192,191,0,0,0,32,176,205,169,19
4	Person4	0,0,0,64,174,105,197,191,0,0,0,224,47,193,191,63,0
5	Person5	0,0,0,96,35,138,185,191,0,0,0,192,58,25,154,191,0,(
6	Person6	0,0,0,128,106,159,184,191,0,0,0,192,246,65,156,63,
7	Person7	0,0,0,224,30,219,186,191,0,0,0,192,99,243,188,63,0
8	Person8	0,0,0,32,209,224,175,191,0,0,0,192,6,147,126,63,0,(
9	Person9	0,0,0,0,15,91,182,191,0,0,0,192,66,209,136,63,0,0,0
10	Person10	0,0,0,96,112,117,180,191,0,0,0,96,207,55,150,191,0
11	Person11	0,0,0,64,108,113,158,191,0,0,0,0,118,46,186,63,0,0,
12	Person12	0,0,0,192,177,214,195,191,0,0,0,0,147,242,170,63,0
13	Person13	0,0,0,128,224,96,195,191,0,0,0,0,67,70,181,63,0,0,0

m. This figure presents a sample of face encodings stored in the system's database, with each entry linked to a unique identifier and name label.

Fig. 15. The encoding values represent numerical feature vectors derived from facial characteristics, allowing for efficient comparison and recognition. These stored encodings enable the system to match detected faces against previously recorded identities, facilitating real-time identification in live video feeds.

G. Field Testing: Environmental Adaptability

The system was tested under various environmental conditions to ensure it could adapt effectively to real-world scenarios. This testing phase focused on assessing the system's consistency in detecting and recognizing faces under different lighting conditions and in varying levels of crowd density. The system was tested during a large showcase event where hundreds of people walked by a booth where the 3-camera system was configured. This showcase event was used to stress test the system in a high-volume situation. Due to the nature of the showcase, people were being identified in the background up to 30 feet away. People gazing at the system whilst walking by would be entered into the system. The system was designed to be traversed from left to right (camera 1 – 3) to simulate the queue correctly. People walking from right to left (camera 3 – 1) would first be identified as a line cutter, then a potential line cutter, then added to the database, causing disturbances in the wait-time calculations for system demonstrations if not caught immediately.

H. Field Testing: Real-Time Processing

Real-time processing capabilities were evaluated to ensure that the system could promptly process data as individuals moved through the queue. The primary goal was to verify the system's responsiveness and ability to update the database and recognition results in a dynamic environment. Although there was an excess of accidental identified individuals from the background, the system did manage to run for 3 hours straight without a hiccup. There were no freezes or downtimes in the system as it swiftly processed every individual that entered its crosshairs.

I. *Field Testing: Unique Identification*

Unfortunately, one of the most significant issues encountered with the system is its difficulty in accurately identifying individuals of color. During field testing, two separate instances were observed where groups of 3-4 individuals of color were presented to the system, both individually and as a group, to assess its facial detection capabilities. In both cases, the system failed to accurately identify all individuals independently. This was not the case for people that were not persons of color. The system seemed to more accurately identify Caucasians throughout showcase testing. This issue is particularly concerning given that OpenCV, an open-source tool widely used for facial recognition, is expected to provide reliable and unbiased results across all demographics. The presence of such bias undermines the fairness and inclusivity of the system, emphasizing the need for more rigorous testing and refinement to ensure that the technology works equally well for everyone.

In addition to algorithmic bias, hardware limitations can also contribute to the problem. For instance, the quality of the camera, lighting conditions, and image resolution can all affect the system's ability to detect and recognize faces accurately. If the hardware is not capable of capturing detailed and well-lit images, especially in varying environmental conditions, the system might struggle more with recognizing faces that have less contrast with the background. So, while algorithmic bias is a significant factor, hardware constraints can exacerbate these issues, making it a multifaceted problem that requires attention to both software improvements and hardware upgrades.

J. *Field Testing: User Experience*

Field testing also involved collecting feedback from users who interacted with the system in a live setting. This feedback was important for understanding the system's impact on queue management and its effectiveness in reducing wait times. The insights gained were used to identify areas where the system could be further refined. In general, there was concern about potential data leakage. Facial data is being collected and stored at the discretion of the operator/owner of the system. Many acknowledged that their smartphone collected data on them without knowledge and or consent. The ease of giving consent and being transparent with the facial data was perceived as better than not being told.

IV. Future Direction

While the prototype demonstrated promising improvements in queue management, several areas warrant further research and development to enhance its functionality and applicability.

One of the primary areas for improvement was the implementation of a consistent camera system. The current prototype utilizes cameras with varying specifications, which can introduce inconsistencies in the data collected at different stages of the queue. Standardizing the cameras across all stages of the queue management process would likely improve overall system performance. A uniform camera setup would ensure better integration, reduce compatibility issues, and provide consistent video quality, thereby enhancing the accuracy of face detection and recognition.

To fully leverage the data collected by the system, developing advanced real-time analytics and reporting tools is essential. Currently, the system processed data in real-time but lacked the capability to provide deeper insights into crowd behavior, peak times, and other valuable metrics. By incorporating real-time analytics, event organizers could gain actionable insights that would help optimize operations, manage crowds more effectively, and make data-driven decisions to improve the overall visitor experience. Additionally, these tools could provide predictive analytics, allowing for better preparation and management of future events.

Future improvements to the queue management system could involve the integration of multi-factor authentication and advanced object detection to enhance identification accuracy. Instead of relying solely on facial recognition, the system could incorporate additional biometric and non-biometric identifiers such as clothing color, gait analysis, and shoe recognition.

By tracking multiple aspects of an individual’s appearance, the system can improve its ability to accurately identify and track each person through the queue, even in cases where facial recognition might be less effective due to obstructions, lighting conditions, or other factors. For instance, recognizing a unique combination of clothing color and patterns, or analyzing a person’s gait, would provide additional layers of verification. This multi-factor approach would reduce the likelihood of misidentification and improve the overall reliability of the system.

Moreover, incorporating object detection for items closely associated with individuals, such as backpacks or strollers, could further enhance the system's capability to manage complex queuing environments. By tracking these items along with individual characteristics, the system could better understand and predict queue dynamics, leading to more accurate wait time estimations and better management of the queue flow.

Another critical area for improvement is the user interface. Both the management and guest information displays could benefit from a more intuitive and user-friendly design. For management, a customizable dashboard with better visualizations of queue data, including real-time alerts and notifications, would enhance usability and make it easier for staff to monitor and manage queues effectively. For guests, a more engaging and informative display could improve their

overall experience by providing clear and accurate wait time estimates and other relevant information.

Looking forward, integrating the queue management system with other emerging technologies could further enhance its capabilities. For instance, combining the system with mobile applications could allow visitors to receive real-time updates on queue status directly on their smartphones. Additionally, integrating the system with other event management tools, such as ticketing systems or crowd control measures, could provide a more comprehensive solution for managing large-scale events.

While the prototype has shown promise in initial testing, more extensive testing in diverse operational environments is necessary to ensure its reliability and effectiveness. Future work should focus on testing the system in different settings, such as indoor and outdoor venues, under various conditions, including different lighting and crowd densities. It's essential the system approximates 100% accuracy amongst people of color before any distribution of the product. Additionally, pilot deployments at actual events would provide valuable feedback and insights, allowing for further refinement and optimization of the system.

V. Conclusions

This project successfully developed and tested a prototype for an automated queueing system that leverages facial recognition technology to improve the accuracy, efficiency, and fairness of queue management in small-scale events. The system demonstrated significant potential in reducing visitor frustration associated with long wait times and line-cutting, thereby enhancing the overall user experience. The integration of facial recognition with multi-factor authentication provided a robust solution for accurately tracking individuals and estimating wait times. However, the project also highlighted several areas for improvement, including the need for a consistent camera system, advanced real-time analytics, and a more user-friendly interface. Moving forward, further development and testing are necessary to refine the system and ensure its reliability in diverse operational environments. By continuing to build on the work done in this project, there is potential to create a fully implementable solution that addresses the challenges of queue management in large-scale events, ultimately contributing to a more satisfying and efficient experience for both visitors and event organizers. This project represents a small advancement in the field of queue management and provides a solid foundation for future research and development. The lessons learned and insights gained from this work offer valuable guidance for future efforts aimed at improving queue management systems and enhancing the overall visitor experience at large-scale events.

References

[1] B. Handaga, B. Murtiyasa, and J. Wantoro, "Attendance System based on Deep Learning Face Recognition without Queue," in *Proc. 2019 Fourth International Conference on Informatics and Computing (ICIC)*, Semarang, Indonesia, 2019, pp. 1-4, doi: 10.1109/ICIC47613.2019.8985697.

[2] V. Dwivedi, M. Bhatnagar, J. Venjarski, G. Rozinaj, and Š. Tibenský, "Multiple-camera System for 3D Object Detection in Virtual Environment using Intelligent Approach," in *Proc. 2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Sofia, Bulgaria, 2022, pp. 1-5, doi: 10.1109/IWSSIP55020.2022.9854487.

[3] Y. J. Wong, K. Huang Lee, M. -L. Tham, and B. -H. Kwan, "Multi-Camera Face Detection and Recognition in Unconstrained Environment," in *Proc. 2023 IEEE World AI IoT Congress (AIIoT)*, Seattle, WA, USA, 2023, pp. 0548-0553, doi: 10.1109/AIIoT58121.2023.10174362.

[4] S. Nirenberg, A. Daw, and J. Pender, "The Impact of Queue Length Rounding and Delayed App Information on Disney World Queues," *Proc. of the IEEE*, vol. X, no. X, pp. XX-XX, 202X.

[5] D. Almeida, K. Shmarko, and E. Lomas, "The Ethics of Facial Recognition Technologies, Surveillance, and Accountability in an Age of Artificial Intelligence: A Comparative Analysis of US, EU, and UK Regulatory Frameworks," *Journal of Ethics in Information Technology*, vol. 2, no. 3, p. 377, Jul. 29, 2021, doi: 10.1007/s43681-021-00077-w.

[6] M. Khan, R. Astya, and S. Khepra, "Face Detection and Recognition Using OpenCV," *Journal of Artificial Intelligence*, vol. X, no. X, pp. XX-XX, 202X.

[7] S. Khan, A. Akram, and N. Usman, "Real-Time Automatic Attendance System for Face Recognition Using Face API and OpenCV," *Wireless Personal Communications*, vol. 113, no. 1, p. 469, Mar. 31, 2020, doi: 10.1007/s11277-020-07224-2.

[8] M. P. Dattani, "Face Detection Based on Image Processing Using Raspberry Pi 4," *Journal of Computing*, vol. X, no. X, pp. XX-XX, 202X.

[9] E. C. Daniels, J. B. Burley, T. Machemer, and P. Nieratko, "Theme Park Queue Line Perception," *Journal of Entertainment Technology*, vol. X, no. X, pp. XX-XX, 202X.

[10] R. H. Ahmadi, "Managing Capacity and Flow at Theme Parks," *Operations Research*, vol. 45, no. 1, p. 1, Apr. 13, 2024, doi: 10.1287/opre.45.1.1.

[11] "Face recognition with OpenCV," *OpenCV Documentation*, 2023. [Online]. Available: https://docs.opencv.org/4.x/da/d60/tutorial_face_main.html. [Accessed: Jan. 10, 2024].