CrowdSurge: A Crowd Density Monitoring Solution Using Smart Video Surveillance with Security Vulnerability Assessment

Mary Jane C. Samonte, Andrea Camille Garcia, Jealine Eleanor E. Gorre, and Joshua Angelo Karl R. Perez
School of Information Technology, Mapua University, Makati City, Philippines
Email: {mjcsamonte@yahoo.com, andreaaacamille, jeegorre, jakrperez17}@gmail.com

***Abstract*—Overcrowding and crowd density monitoring in various places and establishments are being implemented since the pandemic, which helps observe social distancing. This study is about the development of a crowd density solution by utilizing YOLOv4 and Closed-Circuit Television (CCTV) called *CrowdSurge.* The practice of CCTV has been around for so many years with proven benefits. This has been combined with the state-of-the-art YOLOv4 algorithm that provides high video analytics and object detection performance. With the combination of the said technology and algorithm, it will serve as a smart surveillance system. A system and mobile application have been developed, and the YOLOv4 deep learning detection model was used to detect various set of scenarios considered to assess if the model executes according to the actions assigned in the experimental set-up. The browser-based application was tested using CVSS or Common Vulnerability Scoring system, which shows that the severity level of most vulnerabilities is low and has a minor impact on the system. Based on the overall usability testing and statistical results, the respondents are satisfied with both surveillance system and mobile applications developed in terms of functionality, usefulness, and aesthetics. Therefore, using the developed system in real-time surveillance can aid in crowd density reduction in an area.**

***Index Terms*—crowd density, surveillance, smart crowd monitoring, YOLO, CCTV, COVID-19**

I. Introduction

Overcrowding and exposure to large mass gatherings can inflict health problems and risks for individuals present in the area. People concentrated at a specific location on a certain period with a common purpose defines a mass gathering. It may be organized to occur in events such as sports fests or unplanned in daily scenarios such as transport terminals and marketplaces [1]. Crowd density, which pertains to the number of people in a given area, can increase with large crowds. High levels of crowd density can result to health risks related to the transmission of diseases within an area. During the pandemic, it is significant to exercise caution in entering public establishments where crowds may be present. However, it is inevitable for people to visit public places such as business areas like banks or shopping malls and transportation terminals, without expecting crowd formation at any time of the day [2].

Crowd monitoring is a process done in managing a crowd to avoid overcrowding and its underlying risks. It is a response to growing concerns regarding crowdedness in an area. Surveillance technology can be used to capture video footage to monitor an area where an intelligent system can analyze captured footage. It serves a variety of purposes for monitoring in areas where crowds are commonly formed [3]. Parameters in a crowd, such as its count and density, have been prominently considered in vision-based crowd monitoring through surveillance techniques. Other systems for crowd monitoring incorporate Wi-Fi and Bluetooth technologies; however, privacy concerns regarding MAC address collection are raised in such matters [4].

Smart crowd monitoring is often aided by deep learning techniques such as Convolutional Neural Networks (CNN) is used for object detection and analysis of video surveillance in a certain area. The technique can be used in video analytics to count the number of people in a crowded scene [5]. One CNN approach, You Only Look Once (YOLO), has been used in various smart surveillance systems. It is an open-source object detection model that utilizes CNN techniques for real-time scenarios [6]. YOLO can run on the GPU, which supports specialized computing processes through Compute Unified Device Architecture (CUDA) [7]-[9].

Crowd density levels inferred from smart surveillance systems can be employed in end-user platforms for decision support and risk mitigation through forecasts accumulated from the analyzed information. Existing crowd control and management applications provide insights in dealing with crowds through monitoring and alerts. These platforms provide early warnings regarding the crowdedness of an area on a given time, allowing a user to formulate decisions beforehand. For administrators of an establishment, it may also serve as a tool to mitigate the risks of crowding in an area. Alerts regarding the crowdedness of an area can reach the application's users for risk prevention, aiding instructed personnel in their job to manage the crowd [10], [11].

Other systems did not have end-user applications that can be accessed on a mobile device for quicker alerts to be

Manuscript received October 16, 2021; revised February 11, 2022.

doi: 10.12720/jait.13.2.173-180

received by a person-in-charge of monitoring a crowd [6], [12], [13]. Previous studies also recommended the use of real-time surveillance data to test the mobile application and its functions [10].

The developed solution in this study, which could determine crowd density in an area, can be utilized by the administrator of an establishment to monitor and manage a crowd. In addition to a web platform where an admin user can monitor the crowd density of an area, a mobile application was used by a monitoring personnel assigned to regulate the formation of crowds in an establishment, especially during the pandemic.

A. *Objective of the Study*

This study aims to develop a web and a mobile application, for crowd density monitoring solution using smart video surveillance called *CrowdSurge*. The develop system deals with monitoring and viewing of generated data and reports, and alerting watching personnel of an establishment to disperse and manage the crowd. Specifically, this study developed a web application that allows the admin to view the real-time video surveillance, people count, and crowd density per room within an area. An Android application was developed that allows a monitoring personnel to receive alerts from the admin regarding a room that has a medium to high crowd density. Experiments were conducted with sets of test groups.

B. *Scope and Delimitation of the Study*

The scope of the study focused on the development of a mobile application using Android Studio for Android 7.1 OS (Nougat) and a browser-based system application, which acts as a dashboard for displaying data, that runs on the Flask local server. A simulated establishment's administrator used the application for the overall monitoring of the crowd, while the mobile application served as a portable tool where the acting administrator could send information and alerts regarding a specific crowded place to monitoring personnel assigned in the area.

The features of the system application include: (1) a live view of footage captured from the surveillance cameras, (2) the raw count of people detected, (3) the crowd density levels of each room, and (4) an option to send alerts to the monitoring personnel if a medium to high density has been detected by the system among the two areas of an establishment. On the other hand, the mobile application used by the monitoring personnel is comprised of: (1) a receiving end where the alert from the admin can be viewed regarding the area where an increasing density level has been detected, and (2) a button that sends a confirmation to the database for logging purposes. This mobile application acted as a tool for the monitoring personnel to respond to the scene where the crowd had been detected. A login option was available for both applications to authenticate the users before using the platforms.

The solution presented in this study involved the development of a system that follows the process flow shown in Fig. 1. Two end-user applications were developed and utilized in the study: a browser-based system application for the admin and a mobile application for the monitoring personnel in charge of alerting the individuals in an establishment to exercise social distancing and maintain order in the crowd. Both users were required to log in for authentication to verify if they were authorized users of the platforms since the access is limited to a single establishment only.

For the admin, the system application was used for overseeing the overall surveillance of the establishment. The two areas were viewed separately on the interface. The system, which primarily works for crowd counting and density estimation, sends real-time alerts to the admin if a highly dense crowd had been detected in a certain quadrant. The admin alerted and instructed the assigned monitoring personnel to respond to the scene where the high crowd had been detected. An alert was sent to a mobile application held by the monitoring personnel roving in the establishment from the system application.

The monitoring personnel used the mobile application for receiving alerts from the admin regarding the specific areas that need attention. The areas were separated in the interface for a quicker response to be executed. Once the task has been completed, the monitoring personnel can confirm it through the 'Complete' button for data logging purposes.

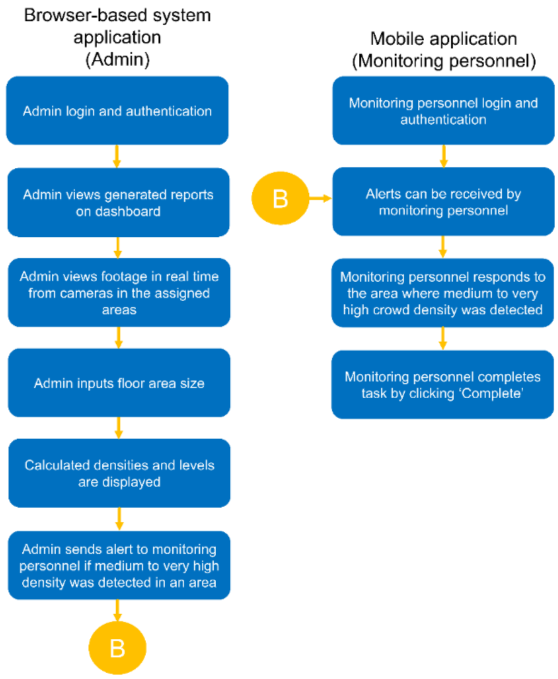


Figure 1. The process flow of the developed system *CrowdSurge*.

II. Review of Related Literatures

A mobile-based crowd management system was developed by Shalash *et al.* to monitor the crowd in an area through surveillance and send alerts to concerned parties regarding the crowdedness of a place. The system is comprised of two parts: (1) the server-side where crowd analysis occurs from surveillance footage captured from IP cameras, and (2) the client-side mobile application where various users of different access rights can view the crowdedness of an area and the corresponding alerts when the system has detected a high crowd density. In the study scenario, a pilgrimage site was considered where mass gatherings are expected to form. Pilgrims use the application for viewing the crowdedness of an area to aid in decision-making, whether to the location or not. The supervisor can send alerts using the application to subordinates for crowd risk mitigation.

On the other hand, a civil defense officer-in-charge of overseeing the crowded areas may receive alerts from the mobile application from the supervisor and view the crowded place that was indicated in the warning. This tool is used to respond quickly to crowdedness in a scene and for crowd monitoring [10]. In the context of prediction using intelligent surveillance, a research by Asmara *et al.* utilizes the YOLO object detection model for predicting future traffic conditions from accumulated data. Footage captured from surveillance is analyzed through Raspberry Pi, aided by an Intel NCS 2 performance accelerator due to the inability of the Raspberry Pi to handle large processes. These data are transmitted through a server which can then be viewed to a web platform. The web platform contains predicted data in the upcoming days to assist concerned parties in formulating decisions based on traffic [14]. A web platform connected to a cloud database through a Flask server was developed by Bura *et al.* to display graphical information regarding the vacancy in a parking area and the location of a specific vehicle for end-users. It utilizes Tiny YOLO and OpenCV to track and detect real-time surveillance footage from the cameras installed in the parking area [15].

YOLO can be used to count individuals in a crowd to determine crowd density in an area. In the study by Feng *et al.*, crowd density estimation is computed based on floor area and the crowd count provided by the YOLOv3 crowd count. The crowd density can be computed by simply dividing the number of people by the floor area, measured in square meters, on which the crowd is present. The results can be ranked based on the division of crowd density proposed by Polus *et al.* [16] where levels are assigned to a designated density derived from calculations, ranging from the lowest (very low/free) to the highest (very high/jammed) densities [17].

III. Methodology

A. *Experimental Set-up*

The setup for experimentation was conducted to answer the research question through the surveillance system and the developed mobile and a browser-based system application, in two areas where one IP surveillance camera was installed in each. Hence, a total of two (2) IP cameras were used for the entire experimental setup. Images captured were analyzed by the YOLOv4 CNN Model, which was pre-trained using MS COCO (Common Objects in Context) datasets. The YOLOv4 detection and people counting model was implemented on the GPU of a computer and was connected to a Flask server to display results in a system application. The footage from each camera were also shown on the system application and streamed through a Real Time Streaming Protocol (RTSP) for viewing. Crowd density was calculated from the counted number of people in a room divided by floor area. The results from calculation were ranked accordingly based on the crowd density rank division table [16]. All gathered and processed data from both applications were stored in the Microsoft Azure SQL Database. The system architecture for this study is shown in Fig. 2.

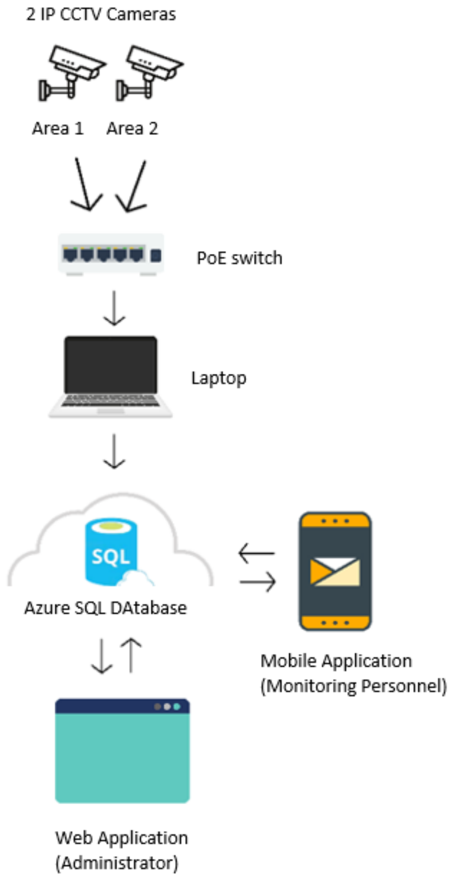


Figure 2. The system architecture of *CrowdSurge*.

Due to the pandemic, experimental setups were done in a private establishment. The size of area considered for the experiment was patterned with the minimum standard floor area size of a habitable room as stated in Section 806 of the National Building Code of the Philippines [18]. The two rooms in the private establishment served as the two areas where the setups were simulated. According to the minimum requirements for Closed-Circuit Television (CCTV) installations in business establishments, cameras should be installed at a 70-degree angle [19]. For this study, the cameras were situated at a 60-degree angle. The experiments were conducted for four (4) hours in one day according to the available hours designated by the establishment. Two hours were allotted for both the control and experimental setup respectively.

Two (2) IP cameras were connected to the port 1 and port 2 of the switch, while the laptop is connected via UP link port to access the IP cameras. Deep Learning was implemented for the computation of the crowd density level and crowd counting. The data were transmitted to the Azure SQL Database which serves as the external database of the developed applications. Flask was used in hosting the system application. The administrator handling the system application can send the crowd density level to the mobile application from the database. The mobile application can receive an alert from the administrator. This prompts the monitoring personnel to respond to the scene where a medium to high crowd density level has been detected. The assigned personnel can ask people in the scene to move away from each other or to another area. If the task is done, the personnel will send a confirmation to the database for data logging purposes by clicking a 'Complete' button.

B. System Design

There were two interfaces used in the study: a browser-based system application and a mobile application. The system application is used by the administrator of an establishment for viewing data on the dashboard. The login page of the system application is shown in Fig. 3. From here, the administrator can login to enter the main dashboard. The credentials are authenticated to ensure security and only the assigned administrator can login.

The main dashboard seen upon logging in is illustrated in Fig. 3. It is the primary page where all data is viewed by the administrator. On the left side of the screen, the footage from two cameras is streamed in real-time. The raw count of people is also displayed. The anchor boxes for the YOLOv4 detection are not shown on the system application because the surveillance feed displayed is only for viewing purposes. The right side of the page displays the crowd densities and their corresponding levels, which are ranked from very low to very high. Each area has a different table for its own data. The 'Send Alert' buttons are available for each designated area if the administrator will send an alert to the mobile application for the monitoring personnel to respond to the specified place.

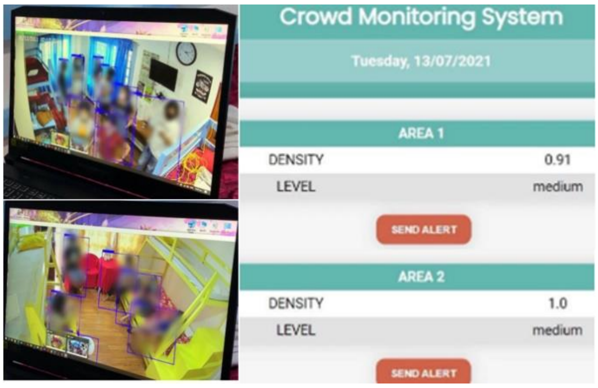


Figure 3. The dashboard of *CrowdSurge*.

C. System Implementation

The two-phase implementation of the developed system is stated as follows: **Phase 1:** In this phase, the YOLOv4 deep learning technique was implemented to the laptop. The two (2) IP cameras were connected to the device for the detection and counting process to take action. Once the back-end system was functioning properly, the crowd density detected was processed and be transferred to Azure SQL Database; **Phase 2:** In this phase, the mobile and system application were developed. The crowd density level was displayed on the system app for viewing and the mobile app after an alert was sent. Both applications were able to view the crowd density levels. The admin using the system app can view the real-time video surveillance and was able to send an alert to the monitoring personnel through the mobile app. The monitoring personnel using the mobile application can click a 'Complete' button if the task has been done.

D. Functionality Testing

Functionalities and performance of both system and mobile application, the applications had undergone several testing levels to determine their capability: (1) functional testing was conducted to assess the compliance of the application to the specified functional requirements; (2) usability testing was focused on ease of use when processing the features and functionality of the applications; (3) for cross-browser testing, the system application was tested in different browsers; (4) lastly is a vulnerability assessment, to identify if the system application was susceptible to known vulnerabilities and determine its possible flaws and weaknesses. Several scenarios were also tried to determine if the persons captured in surveillance can be detected individually to test the calibration of the model. A cross-browser test was also conducted to assess the compatibility of the system application with various system browsers. This study includes assessing the capability and severity of a vulnerability using the Common Vulnerability Scoring System or CVSS. The study used a vulnerability scanning tool called Owasp Zap for the system application.

E. Vulnerability Assessment

The developed application was tested using CVSS or Common Vulnerability Scoring system. It is composed of three metrics which are Base, Temporal and Environmental. The CVSS 3.0 Base Score Ratings was the basis of determining the level of severity of a specific vulnerability. The Base scoring system had a severity level ranging from None with a severity level of 0.0 to Critical with a severity level of 9.0 - 10.0. The Base Score has two metrics used to assess a vulnerability; the first metric is called exploitability metrics which is composed of Attack Vector (AV) is where the vulnerability is located (Network, Adjacent, Local or Physical), Attack Complexity (AC) are special conditions or requirements of an attack (Low or High), Privileges Required (PR) states the level of level of privileges an attacker possess before exploiting the vulnerability (None, Low or High), User Interaction (UI) if it an attack does or does not involve user interaction (None or Required) and Scope (S) if resources are changed or unchanged. The second base metrics is impact metrics which is composed of Confidentiality Impact (C) measures the impact to the of confidentiality of the information resource, Integrity Impact (I) measures the impact to integrity of an exploited vulnerability and Availability Impact (A) measures the overall impact to the availability of the impacted component caused by an exploited vulnerability [20].

The Temporal Testing and Environmental Testing are both dependent on External Factors in your environment. Temporal Testing is the characteristics of a vulnerability that evolves overtime. It is composed of three metrics which are Exploit Code Maturity (E) which measures the probability of the vulnerability being attacked (Not Defined, High, Functional, Proof of Concept or Unproven), Remediation Level (RL) the availability of a solution (Not Defined, Unavailable, Workaround, Temporary Fix and Official Fix), and Report Confidence (RC) which indicates the known technical details of a vulnerability (Not Defined, Confirmed, Reasonable, or Uknown). Moreover, Environmental Testing is more concern in the changes in the environment. This is composed of three metrics which are Confidentiality Requirement (CR), Integrity Requirement (IR), and Availability Requirement (AR). Also, there is a modified base metrics which is composed of eight metrics [20].

A severity level is indicated to assess the level of a vulnerability. First is severity level critical, which has a score ranging from 9.0 - 10.0 means that vulnerabilities can result to root-level compromise of servers or infrastructure devices, and attackers do not need any authentication or social engineering to exploit an attack successfully. Second is the severity level high which has a score ranging from 7.0-8.9; this indicates that vulnerabilities are difficult to exploit by an attacker, and exploitations may result in elevated privileges or data loss and downtime. The third is severity level medium ranging from 4.0-6.9, this is where attacks require social engineering tactics, vulnerabilities that require user privileges, and exploitation provides limited access to a system. Lastly is severity level low, which has a score ranging from 0.1-3.9, this are vulnerabilities that does not have a significant impact to a system and requires local and physical access to a system [21]. The CVSS 3.0 Base Score Ratings [20] is as follows: None = 0.0, Low = 0.1-3.9, Medium = 4.0-6.9, High = 7.0-8.9 and Critical = 9.0-10.0.

F. Usability Test

The study utilized System Usability Scale (SUS) for the usability testing of the system. Likert scale was used to calculate the 10-item questionnaire. The SUS scoring system includes: 1. For positive outcomes, add up the total score for all odd-numbered questions, then subtract five (5) from the total to get (X); 2. For negative outcomes, add up the total score for all even-numbered questions, then subtract that total from 25 to get (Y); and 3. Add up the total score of the new values (X+Y) and multiply by 2.5. All values are added and multiplied by 2.5, and the end result must at least equate to 68 for the average score. Calculated scores lower than 68 may indicate that the system may need further enhancement while scores of at least 80.3 indicate that the system is ranked as excellent.

IV. Experiment Process

A. Data Gathering

The data gathering process was done in two phases: the initial phase and the testing phase. Data gathering was conducted in an establishment with two rooms for the initial phase, each having one IP camera installed. The data was collected in two (2) hours for the controlled scenario or the no mobile application used, and two (2) hours for the experimental group using the developed system and mobile application used in monitoring the crowd. The computed crowd density per area observed has a time interval of 10 minutes [14].

B. Crowd Simulation

The floor areas are measured in square meters: Quadrant 1 = 8.75 sq/m and Quadrant 2 = 6 sq/m. There were two experimental scenarios and control groups. There were fourteen (14) people who participated in both groups. They were situated in the rooms, moving with speed varying at (0 – 1m/s) randomly [19]. Both setups contained the same group of people.

C. Confusion Matrix

A confusion matrix was used to evaluate the classification of the model and to determine its true values. The study's own test data was used from their video surveillance recordings. The confusion matrix shows the predicted values versus the actual values of the predictions by the algorithm. It includes True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). To conclude if the performance of the model can classify and detect people, accuracy and precision were calculated.

V. Results and Discussion

A. YOLO Detection

The YOLOv4 deep learning detection model was used to detect various set of scenarios considered to assess if the model executes according to the actions assigned in the experimental set-up and literatures [22], [23]. This test was done using one (1) camera only as it was only done for determining if the model can execute the operations. The results for the different YOLOv4 Object Detection and Person Counting scenarios in the study are shown in Table I and Fig. 4(a to f).

The scenario shown in Fig. 4(a) involves two people that are standing next to each other. Both persons are detected individually with their own bounding boxes. As Person A walked behind Person B, a total occlusion occurred, and Person A was undetected as shown in Fig. 4(b). This affected the person count as well because only one person was detected in the frame. However, upon walking past Person B, Person A was once again detected even with partial occlusion, as shown in Fig. 4(c). Person A was re-detected and counted.

For a scenario where a person is carrying another person (i.e. a parent carrying a child), both individuals were counted and detected despite the partial occlusion as persons were overlapped, as shown in Fig. 4(d).

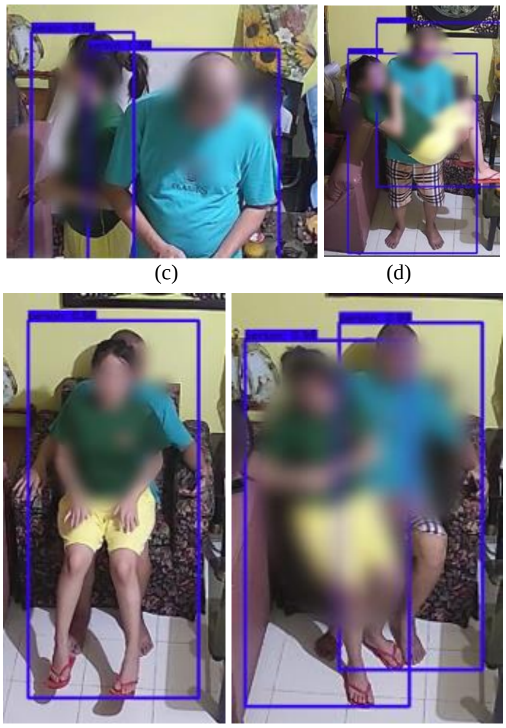
For the last test scenario, which involves a person sitting on another person's lap as shown in Fig. 4(e) and 4(f), it was determined that the sitting position affects the detection. A total occlusion occurred, causing Person B to stay undetected as Person A was sitting on their lap as shown in Fig. 4(e). However, on Fig. 4(f), both persons were then detected with the change in sitting position despite the partial occlusion.

The results have shown that total occlusion affects detection and the number count. A total occlusion that occurs for a short period of time can be resolved when the undetected person moves away from the object or person that caused them to be occluded. Partial occlusions where slight overlaps occur between people do not pose a problem as all persons are detected individually. For longer periods, total occlusion may hinder the detection rate, resulting in false detections and counts [23]-[26].

TABLE I. YOLOV4 OBJECT DETECTION AND PERSON COUNTING

Test Scenario	Action	Expected Response	Remarks
1- Two people standing next to each other	Two persons are shoulder to shoulder	Both Person A and Person B are detected and counted individually	Passed
2- Person A standing in front of Person B	One person stands in front of another person	Both Person A and Person B are detected and counted individually	Failed
3- Person A moving from behind Person B	Opposite set-up with scenario #2	Both Person A and Person B are detected and counted individually	Passed
4-Person A is carried by Person B	A person is being carried by another person B (i.e. if a father is carrying a child)	Both Person A and Person B are detected and counted individually	Passed
5-Person A is sitting on the lap of Person B	A person is sitting on the lap of another person with adjacent leg-aligned position	Both Person A and Person B are detected and counted individually	Failed
6-Person A is sitting on the lap of Person B	A person is sitting on the lap of another person with unparallel leg position	Both Person A and Person B are detected and counted individually	Passed

(a) (b)



(c) (d)

(e) (f)

Figure 4. Experimental YOLOv4 detection and counting.

B. Vulnerability Test Results

This study used Owasp Zap for Vulnerability Assessment. An automated scan was launched on the system application using the system URL. Upon completion of the scan, the Owasp Zap displayed the results, which showed all the relevant information related to the detected vulnerabilities. The information acquired from the scan results is analyzed to determine the severity of a vulnerability using CVSS.

Owasp Zap detected a total of four (4) vulnerabilities on the system application, which is shown in the table. The implication of each vulnerability in the system are as follows: (1) The first vulnerability is X – Frame Options Header Not set indicates with a Severity level of low and base score of 3.1. This shows that system is prone to Clickjacking attacks. (2) The second vulnerability is No Anti-CSRF tokens were found in an HTML submission with severity level of low and base score of 3.1. Having No Anti-CSRF tokens, Cross-site request forgery attacks can be used to redirect users to a malicious website, steal sensitive information, or execute other actions within a user's session. (3) The third vulnerability is Cross-Domain JavaScript Source File Inclusion with a severity level of medium with base score of 6.1. This indicates that one or more script files from a third-party domain are detected in the system. If a third-party domain intentionally or unintentionally holds a malicious content, it will also be added on the victim's system application. (4) Last vulnerability is X-Content-Type-Options Header Missing with a severity level of low and base score of 3.1, This is an issue where Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff' which allows older versions of a browser to perform MIME-sniffing, that may cause the response body to be interpreted and displayed as a content type other than the declared content type. Based on the results in the table, the severity level of most vulnerabilities is low, which means that it has minor impact on the system, and exploiting these vulnerabilities requires local or physical access to a system. The result of Owasp Zap and CVSS Metrics are shown in Table II.

TABLE II. OWASP ZAP AND CVSS METRICS RESULTS

Vulnerability	CWE ID	Severity Level	Base Score
1. X-Frame-Options Header Not Set	1021	Low	3.1
2. Absence of Anti-CSRF Token	352	Low	3.1
3. Cross-Domain JavaScript Source File Inclusion	829	Medium	6.1
4. X-Content-Type-Options Header Missing	693	Low	3.1

C. Usability Test Results

A usability test was conducted among thirty (30) respondents whose ages and occupations vary to evaluate the system and developed applications, a. The average age of respondents was twenty-six (26) years old. Majority of the respondents also indicated various occupations.

Upon calculating the scores from each question, the average SUS score was **84**. It indicates that the system users were satisfied with the overall usability and functionality of the system. The applications worked as expected, and the users were able to complete tasks. The users agree that the overall system can execute its purpose correctly.

D. Confusion Matrix Results

The confusion matrix was used to evaluate the detection model's overall performance. The results in Table III show the plotted and predicted values. Values obtained by counting each person from the surveillance footage were compared with the predicted values. Thirty-three (33) images were taken from the videos, and each person in the images were counted manually. True Positive (TP) is the number of correct predictions of the "Person" class, while True Negative (TN) encompasses objects that were correctly misclassified as persons. On the other hand, False Negative (FN) is the number of people that were not classified as persons, while False Positive (FP) pertains to the incorrect classifications of the "person" class. The sample of 415 resulted to 187 TP values that were able to classify people as "Persons" while 194 TN values were detected as objects aside from the person class. An FN value of 23 was also calculated, which represents the number of people that were not classified as "persons," and 11 FP values were the result of misclassifications. Overall, the model had an accuracy of 91.81% and a precision of 94.44%. The results imply that the model performed well in classifying and detecting people.

TABLE III. CONFUSION MATRIX RESULTS

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP (187)	FN (23)
	Negative	FP (11)	TN (194)

VI. CONCLUSION

Crowd density involves the formation of a large crowd in a specific area. The underlying concerns that overcrowding poses to a person's health, especially during the COVID-19 pandemic, can be alarming if crowds were not managed. In previous studies, visual-based crowd management solutions were implemented to avoid crowdedness in an area. Smart surveillance that uses deep learning techniques has been used in various object detection and person counting applications. Applying these techniques to real-time surveillance can aid in crowd density reduction in an area.

This study presents a crowd management solution that uses the YOLOv4 deep learning algorithm to count people in an area and calculate crowd densities. Computed data can be viewed through developed end-user platforms: a browser-based system application and a mobile application. Establishments can use the developed system to manage crowds within their premises to lessen crowd density levels. The usability test using the System Usability Scale (SUS) resulted to an overall score of 84. This means that users were satisfied with the system and its functionalities. The model utilized in this study has an accuracy rate of 91.81%and a precision rate of 94.44%. Based on the results, the model can properly detect and classify objects and it is effective to use in monitoring the crowd. It can be concluded that the system was effective in managing the crowd in an area.

The researchers recommend future studies to expand the detection by adding the number of cameras installed in different heights and angles. To further assess the system's recognition, the experimentation can be conducted in a larger area. Experiment can also be conducted in a bigger are Moreover, the testing of the system should be done with more participants and consider it in designing the system when simulating real-life situations [10].

CONFLICT OF INTEREST

This research paper presentation and registration was funded by Mapua University. All authors declare that they have no conflicts of interest.

AUTHOR CONTRIBUTIONS

Ms. Samonte supervised the whole research development and took the lead in writing the manuscript. Ms. Gorre developed the theoretical formalism and performed the analytic calculations. Ms. Garcia and Mr. Perez performed the numerical simulations. All authors provided critical feedback and helped shape the research, analysis, and manuscript.

References

[1] World Health Organization, *Public Health for Mass Gatherings: Key Considerations*, 2015.

[2] L. V. Seidlein, G. Alabaster, J. Dee, and J. Knudsen, "Crowding has consequences: Prevention and management of COVID-19 in informal urban settlements," *Building and Environment*, vol. 188, 2021.

[3] U. Singh, J. F. Determe, F. Horlin, and P. De Doncker, "Crowd monitoring: State-of-the-Art and future directions," *IETE Technical Review*, vol. 38, pp. 578-594, 2020.

[4] S. Lamba and N. Nain, "Crowd monitoring and classification: A survey," *Advances in Computer and Computational Sciences*, vol. 553, pp. 21-31, 2017.

[5] L. T. Hsien and I. Atmosukarto, "Video analytics in train cabin using deep learning," in *Proc. IEEE 4th International Conference on Intelligent Transportation Engineering*, 2019, pp. 94-98.

[6] J. Redmon, S. Divyala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779-788.

[7] A. P. Jana, A. Biswas, and Mohana, "YOLO based detection and classification of objects in video records," in *Proc. 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology*, 2018, pp. 2448-2452.

[8] C. Xue, P. Liu, and W. Liu, "Studies on a video surveillance system designed for deep learning," in *Proc. IEEE International Conference on Imaging Systems and Techniques*, 2019, pp. 1-5.

[9] G. Oltean, L. Ivanciu, and H. Balea, "Pedestrian detection and behaviour characterization for video surveillance systems," in *Proc. IEEE 25th International Symposium for Design and Technology in Electronic Packaging*, 2019, pp. 256-259.

[10] W. M. Shalash, A. A. Hazimi, and B. A. Zahrani, "A mobile based crowd management system," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 6, no. 6, pp. 205-215, 2017.

[11] M. Yamin, A. M. Basahel, and A. A. A. Sen, "Managing crowds with wireless and mobile technologies," *Wireless Communications and Mobile Computing*, article ID 7361597, 2018.

[12] Python Software Foundation, Flask 2.0.1.

[13] AllGoVision. (2009). [Online]. Available: https://www.allgovision.com/

[14] R. A. Asmara, B. Syahputro, D. Supriyanto, and A. N. Handayani, "Prediction of traffic density using YOLO object detection and implemented in Raspberry Pi 3b + and Intel NCS 2," in *Proc. 4th International Conference on Vocational Education and Training*, 2020, pp. 391-395.

[15] H. Bura, N. Lin, N, Kumar, S. Malekar, S. Nagaraj, and K. Liu, "An edge based smart parking solution using camera networks and deep learning," in *Proc. IEEE International Conference on Cognitive Computing*, 2018, pp. 17-24.

[16] A. Polus, J. L. Schofer, and A. Ushpiz, "Pedestrian flow and level of service," *Journal of Transportation Engineering*, vol. 109, no. 1, 1983.

[17] F. Feng, S. Liu, Y. Pan, X. He, J. Wei, and L. Wang, "Crowd density estimation method based on floor area," *Journal of Physics: Conference Series*, vol. 1651, no, 1, 2020.

[18] *National Building Code of the Philippines*, Department of Public Works and Highways, 2005.

[19] *Directing Cities/Capital Towns to require the Installation of Closed-Circuit Televisions (CCTV) for certain Business Establishments in accordance with Section 16 (General Welfare Clause) of Republic Act No. 7160 to Support the Maintenance of Peace and Order and Public Safety*, Department of the Interior and Local Government, 2014.

[20] R. Abraham. (2015). Common vulnerability scoring system v3.0: Specification document. [Online]. Available: https://www.first.org/cvss/v3.0/specification-document

[21] Atlassian. Severity levels for security issues. [Online]. Available: https://www.atlassian.com/trust/security/security-severity-levels

[22] Y. Yuan, "Crowd monitoring using mobile phones," in *Proc. Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2014, pp. 261-264.

[23] G. Bhola, D. Kathuria, and C. Das, "Real-Time pedestrian tracking based on deep features," in *Proc. 4th International Conference on Intelligent Computing and Control Systems*, 2020, pp. 1101-1106.

[24] D. Acharya, K. Khoshelham, and S. Winter, "Real-Time detection and tracking of pedestrians in CCTV images using a deep convolutional neural network," in *Proc. the 4th Annual Conference of Research@Locate*, 2017, pp. 31-36.

[25] M. I. H. Azhar, F. H. K. Zaman, N. M. Tahir, and H. Hashim, "People tracking system using DeepSORT," in *Proc. 10th IEEE International Conference on Control System, Computing and Engineering*, 2020, pp. 137-141.

[26] S. Kumar, P. Sharma, and N. Pal, "Object tracking and counting in a zone using YOLOv4, DeepSORT and TensorFlow," in *Proc. International Conference on Artificial Intelligence and Smart Systems*, 2021, pp. 1017-1022.

Mary Jane C. Samonte has a double bachelor's degree in computer education and IT. She also has two post graduate degree; Information Technology and Computer Science. She finished her Doctor in IT last 2019 with a study focusing on Deep Learning. She is now in her 20th year as a faculty member of Mapua University. She has a wide range of research interests that are centered around educational technologies, gamification, mobile and ubiquitous learning, digital game-based learning, artificial intelligence in education, e-health, assistive technology, natural language processing, green computing and data analytics-based studies.

Jealine Eleanor E. Gorre is a graduate of Bachelor of Science in Information Technology. She aims to become an IT specialist, with expertise in both front-end and back-end development. She encourages others to become the best that they can be and help them reach their fullest potential as an IT practitioner. During her senior year, she is a student leader who amplifies other student skills and aspirants to be involved in IT related events.

Andrea Camille Garcia is a graduate of Bachelor of Science in Information Technology. She aspires to become a full stack developer. She loves to express herself by playing flute and enjoys drinking coffee. She describes herself as adventurous. creative. and driven. She is part of a community that inspires others to express themselves through visual arts.

Joshua Angelo Karl R. Perez is a graduate of Bachelor of Science in Information Technology. He aims to become a Cybersecurity Specialist. He believes that miracles come true for those who work and study hard. He is enthusiastic not only about life, but also in solving real world problem that deals with cybersecurity and other digital challenges of our times.