Algorithm for Accurate People Counting in Conference Halls

Mysoon Ali
Al-Rafidain University College
Baghdad, Iraq
mysoon@ruc.edu.iq

Hasan Harith Jameel Mahdi
Al-Turath University College
Baghdad, Iraq
hasan.harith@turath.edu.iq

Mustafa Mohammmed Jassim
Al-Noor University College
Nineveh, Iraq
mustafa30@alnoor.edu.iq

Dmytro Palamarchuk, Shystun Oleksii
Kyiv National University of Construction and Architecture
Kyiv, Ukraine
palamarchuk.da@knuba.edu.ua, shystun_or@knuba.edu.ua

Abstract—**Background: Intelligent conference rooms are crucial to 21st-century enterprises for events. Safety, resource optimisation, and event management depend on accurate counting in such contexts. Manual headcounts are effective yet inefficient and error-prone, particularly for big crowds requiring automatic people counters.**

Objective: This article introduces and validates a data-driven algorithm to count and track people in an intelligent conference hall. The concept uses IoT infrastructure, low-resolution cameras, and powerful image-processing algorithms to improve security, resource usage, and real-time management choices.

Methods: The message-oriented IoT algorithm incorporates motion detection, background subtraction, people counting, and tracking modules. Blob analysis, edge detection, and low-maintenance, low-resolution cameras capture real-world data. A decision-making module controls the conference hall's atmosphere based on real-time data.

Results: The algorithm operates with exceptional dependability with a 96.5% accuracy rate and 95% confidence interval in real-time individual counts. Using real-world data and experimental findings, the algorithm has been extensively tested and shown to work in diverse head-counting situations.

Conclusion: Intelligent conference hall management using the suggested algorithm might revolutionise venue management. The algorithm is accurate, and real-time headcounts improve security, resource utilisation, and management decisions. This makes it a promising candidate for intelligent conference hall management and optimisation for diverse events and gatherings.

I. Introduction

Intelligent conference rooms have become essential elements of contemporary corporate activities, serving as venues for various gatherings, including seminars, workshops, and meetings. Accurately determining the number of attendees is crucial to organising an event. Precise enumeration may enhance the optimisation of resource utilisation, ensure compliance with safety requirements, and allow well-informed decision-making about event management [1].

Traditionally, the act of manually counting participants at conferences formed the basis. This approach has attained a certain level of accomplishment. However, it has imperfections, particularly regarding the continuous surveillance of large populations. The growing need for automated techniques to count participants in modern conference venues directly relates to these difficulties.

Based on the previously indicated need, our research presents a novel approach for continuously monitoring individuals in intelligent conference rooms. Our solution [2] incorporates many components, such as tracking, motion detection, background subtraction, and a message-oriented Internet of Things (IoT) architecture. The various components work together to provide reliable headcounts, which are used by the module that controls the temperature in the conference room.

The motion detection module can differentiate between essential and insignificant changes inside the conference room using motion detection technology. The background removal module improves processing performance by isolating the stationary items in the room [3].

The people-counting module uses image processing methods, such as edge detection and aggregation analysis, to recognise and monitor moving objects precisely [4]. The monitoring module employs a Kalman filter [5].

To track and record the locations and movements of participants, providing management with valuable data.

In addition, our technology incorporates this information into a decision-making module that optimises the conference room arrangement based on the collected data, ensuring maximum efficiency and comfort. This may require adjusting the HVAC system to support more significant populations or tweaking the lighting layout based on occupancy levels.

The effectiveness of our method has been shown in actual corporate gatherings, as evaluated using readily accessible real-life information. Using low-resolution cheap cameras may accurately estimate the number of persons present with a 95% confidence interval [6]. The effectiveness and flexibility of our methods have been shown in several situations where accurate headcounts are needed, as indicated by the successful results of our study.

The article presents a detailed strategy for monitoring conference attendees using an Internet of Things (IoT) architecture, low-resolution camera systems, and sophisticated image processing techniques. Our analysis confirms the method's efficacy in enhancing the operational efficiency of intelligent conference rooms.

A. Aim of the Article

The article aims to precisely count the number of people in an intelligent conference hall utilising an Internet of Things architecture, low-resolution cameras, and image processing techniques. The suggested method has many different parts, including a module for detecting motion, another for subtracting the backdrop, another for counting individuals, and another for monitoring their movements. This article shows that the algorithm may provide valuable results and be used when precise headcounts are necessary. The essay also highlights the potential advantages of automatic people counting in intelligent conference rooms, such as improved security, more efficient resource utilisation, and more informed management decisions.

B. Problem Statement

It is crucial to prioritise and focus on difficulties that need a complex solution. Inadequate or irregular lighting may significantly hinder the precision of optical detecting systems, hence serving as the primary obstacle. Additionally, population density is a crucial consideration, as it challenges accurate enumeration in highly populated regions. It is because occlusions may arise when persons are concealed from sight by others, making it harder to count them precisely. Observing diverse population habits and movements may heighten the detection method's complexity. Finally, occlusions resulting in an inaccurate count are a notable concern due to their frequent occurrence in crowded and busy spaces like conference rooms. There is an increased need for a reliable algorithm to adjust to different setups and calculate the population. We have developed a solution to overcome these obstacles and provide precise and dependable headcounts in conference rooms.

II. Literature Review

The literature review highlights various approaches and techniques proposed to tackle the problem of people counting, which has been applied to multiple domains such as surveillance, traffic, and crowd counting.

Various techniques have been employed for people counting, including face/person detection algorithms, blob/object detection, and tracking schemes [7]. For instance, Hsieh et al. [8] utilised Kinect and morphological processing to extract regions of interest for pedestrian flow counting. Meanwhile, Ryan et al. [10] introduced an approach using local features for total crowd estimation. Zhang et al. [20] developed an unsupervised water-filling method leveraging depth information from a vertically mounted Kinect. Brostow and Cipolla [4] implemented a probabilistic approach to track simple image features, and Rabaud and Belongie [9] proposed a highly parallelised version of the KLT tracker. Perspective distortion correction was investigated by Celik et al. [6], and Ryan et al. [10] proposed a scene-invariant crowd-counting algorithm functioning on multiple calibrated cameras. Chan et al. [8] devised a two-step algorithm, using the mixture of dynamic textures motion model to segment the crowd into components of homogeneous motion and then extracting simple holistic features from each segmented region to protect the privacy of test subjects.

Despite the variety of approaches used in specific contexts, a universally accepted solution for people counting remains crucial.

However, recent advancements in the Internet of Things (IoT) and machine learning techniques have developed more sophisticated approaches [2] for measuring and tracking people in intelligent environments such as conference halls. For instance, Akasiadis et al. [1] proposed a system that exploits IoT technologies to create bright rooms, including brilliant furniture, interactive displays, and environmental sensors. The authors used motion and audio detection algorithms to recognise activities in the room and track the participants.

Moreover, some researchers have also proposed using low-resolution cameras for people counting, as demonstrated by Loy et al. [11]. These techniques have the advantage of being cost-effective and easy to deploy, which makes them suitable for large-scale surveillance applications.

The literature review shows that people counting is a challenging problem that has been tackled using various techniques, including face/person detection, blob/object detection, and tracking schemes. Recent advancements in IoT and machine learning have enabled the development of more sophisticated approaches for counting and tracking people in intelligent environments [12]. At the same time, low-resolution cameras have also been effective for large-scale surveillance applications. However, there still needs to be a universally accepted solution for people counting, and further research is needed to address the limitations of existing techniques. By using every tool at their disposal, they could attain optimal outcomes. To ensure the anonymity of test participants, Chan et al. [8] devised a two-stage approach in which the crowd was divided into homogenous motion components using the mixing of dynamic textures motion model. Each split area was then used to derive basic holistic properties [13].

III. Methodology

The article requires several other algorithms and processes before it can be put into practice. The first algorithm is the blob detection algorithm, which finds the blobs or areas of interest in the picture. This approach iterates over each non-zero pixel in the picture by establishing a rectangle union and merge threshold. A rectangle is established for each pixel, and if it overlaps with any previously specified rectangles, the two are combined. This procedure is repeated until all blobs have been found [14, 15].

The second technique is the tracking algorithm, which follows the identified blobs from one frame to the next. Each blob in the current frame is linked to the blob from the previous frame closest to it using this technique. A new track is started if no blob that fits the criteria is discovered. This procedure is repeated for each frame, yielding a set of tracks for each identified blob [16].

The total number of attendees in the bright conference hall is determined by tallying up the number of tracks across a designated picture region corresponding to the room's entrance or departure.

A. Foreground Image Enhancement in Smart Conference Halls Using Background Subtraction

Our method of counting individuals in a conference hall setting involves using an advanced background removal module to improve the clarity of the foreground picture. This module is essential for differentiating mobile humans from a fixed background. This module analyses the real-time video stream in case any movement is detected inside the conference location.

The first step in improving the photograph is precisely modifying the visual attributes of the foreground. Gamma correction is a method used to modify the brightness of a picture. It uses a gamma value of 2.2. Subsequently, a contrast adjustment of 1.5 is used to enhance the visibility of the foreground characteristics and differentiate persons from the backdrop.

After these improvements, a backdrop model is created using the Mixture of Gaussians (MOG) technique, as described in [8]. This statistical method detects foreground items by modelling each pixel as a mixture of Gaussian distributions, with the parameters adjusted to meet the unique needs of the conference hall [17].

Various morphological approaches are used to improve the quality of the foreground picture. The first step involves using a morphological aperture and a 3x3 pixel structuring element to fill in the gaps between foreground items. These empty spaces often occur due to fluctuations in lighting conditions, which might undermine the individual count. To improve the shape and placement of the items in the image's foreground, their boundaries are determined using a morphological gradient that includes a 5x5 pixel structure [18], [19].

To mitigate issues with camera artefacts or the unique proportions of the conference hall, we utilise a morphological closure operation using a 5x5 pixel structural element size to highlight any dark regions inside the foreground objects. To counteract any potential decrease in detail, adaptive expansion approaches may be used in combination with area growth algorithms [7].

This involves boosting the data related to the foreground entities and expanding the areas of interest.

Given the arduous nature of these procedures, the backdrop elimination module is essential for enhancing the quality of the foreground picture. We differentiate the individuals from the stationary background and improve their visual features using morphological procedures and area expansion approaches. After the picture has been altered, it is sent to the algorithm in charge of counting individuals, resulting in a precise count of heads. Our approach is purposely designed to be easily replicated, allowing for reliable evaluation of the findings based on a complete and parameter-specific explanation.

B. Technique for identifying and counting persons in a video feed

Blob detection is a popular technique for identifying and counting persons in a video feed. The method includes locating regions of a picture with similar brightness or hue and labelling them as "blobs." When used for people counting, these blobs stand in for individuals individually or in groups.

Thresholding, in which picture pixels are assigned to the foreground or background, is a standard method for detecting blobs. The foreground pixels are then aggregated into blobs, which may be filtered based on size, shape, or other criteria to eliminate false positives [16].

Many image processing methods may be used to identify blobs, including the Canny edge detector, the Laplacian of Gaussian (LoG) filter, and the Difference of Gaussians (DoG) filter. In order to emphasise portions of the picture with strong contrast or sharp edges, these techniques convolve the image with a kernel or collection of kernels.

After the blobs have been identified, they may be processed further to provide an estimate of the number of persons visible in the picture. Afterwards, a bounding box may be drawn around each blob, and the number of persons in the group can be estimated based on the bounding box's size and form.

People counting using blob detection is commonplace in security, transportation, and crowd control [18, 20]. Nevertheless, it has drawbacks in busy settings because individual blobs might blend, divide, or occlude one another, leading to erroneous counting results. Researchers have developed more advanced strategies that combine blob identification with tracking and machine learning algorithms to increase the accuracy and resilience of people counting to solve these constraints [15].

This procedure may be summed up using the pseudocode shown below, where L stands for the list containing the collection of blobs:

Bi, i = 1, 2, ..., k

1. Initialize L = [1]
2. while the video stream is running, do
3. read the next frame from a video stream
4. apply background subtraction to obtain a foreground mask
5. apply morphological opening to remove small holes in the foreground mask
6. apply the morphological gradient to obtain the contour of the blobs
7. apply morphological closing to remove black holes inside the blobs
8. detect blobs in the foreground mask using connected component labelling
9. add new blobs to L and remove old blobs that have disappeared

10. estimate the number of people in each blob by fitting a bounding box and using size/shape criteria

11. display the results (e.g., number of people, the position of each blob) on the screen

12. end while

13. stop video stream

The pseudocode visualised below (Fig. 1) explains how to use background subtraction and morphological procedures to identify blobs, which are areas of interest containing individuals, in a video stream. The program starts by creating an empty list to house the blobs and then runs in a loop for as long as the video stream is active.

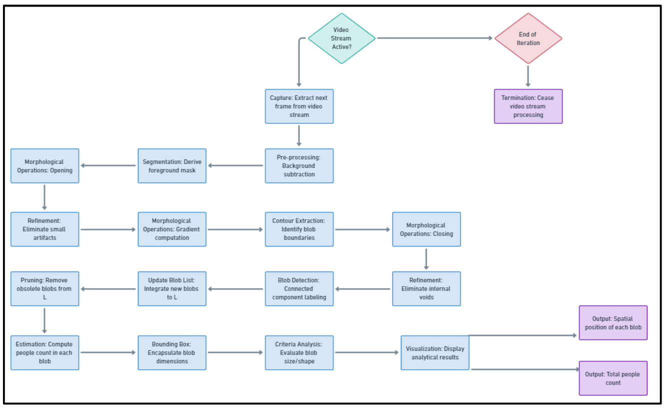

Fig. 1. Flowchart of Advanced Video Stream Analysis for People Counting

A foreground mask is created by subtracting the backdrop from each video frame to identify the items that have changed position in the scene. The algorithm then uses a morphological gradient to get the blobs' contours and morphological opening to fill in any minor gaps in the foreground mask. The dark holes within the blobs are subsequently filled in via morphological closure.

The method then uses linked component labelling to find blobs in the foreground mask and adds new blobs to the list while eliminating old blobs that have vanished. The number of persons in each blob is calculated by fitting a bounding box and applying size and form criteria. The findings are then shown on the screen, including the total population and the locations of each blob. The code will keep looping till the video stream has finished [21].

C. *Blob Detection Algorithm for Counting People in a Smart Conference Hall*

The algorithm identifies regions of similar pixel values and groups them to form blobs: T ← rectangle union threshold M ← rectangle merge threshold for each non-zero-pixel p define R ← N×N rectangle [22].

Here is a description of Algorithm 1:

1. Set the rectangle union threshold (T) and merge threshold (M) values.

2. Define an N-by-N pixel rectangle (R) for each pixel that is not zero in the image or video stream.

3. For each rectangle R, check if it overlaps with existing blobs. If it does not overlap with any blobs, create a new blob and add R to the list of rectangles for the new blob. If it overlaps with one or more blobs, merge the rectangles for each overlapping blob with R and update the list of rectangles for the merged blob.

4. Once all rectangles have been processed, filter out blobs with a total area below the rectangle union threshold (T).

5. Finally, filter out any blobs overlapping with another blob (i.e., have an overlap area more remarkable than the rectangle merge threshold (M)) and merge them into a single blob.

The resulting blobs represent individual objects in the image or video stream and can be further analysed to extract features such as size, shape, and location. In the context of the "Algorithm for calculating the number of people in a smart conference hall," the blob detection algorithm would be used to identify and label individual people in the foreground of the video stream, which can then be counted to estimate the number of people in the conference hall.

Here is the pseudocode for the algorithm:

Input: Video stream from low-resolution cameras in the intelligent conference hall

Output: Estimated number of people in the conference hall

1. Initialise the background model by capturing the first frame of the video stream

2. For each subsequent frame of the video stream, do the following:

a. Apply gamma correction and contrast adjustment to the frame

b. Calculate the foreground mask using the background subtraction method

c. Apply morphological opening to the mask to remove small holes

d. Apply a morphological gradient to the mask to estimate the object outlines

e. Apply morphological closing to the mask to remove any remaining black holes inside objects

f. Perform region growth using different expansion schemes to refine the foreground objects

g. Apply blob detection to identify and label individual objects in the foreground

h. Estimate the number of people in the foreground by counting the number of labelled blobs

3. Update the background model periodically to account for changes in the environment

4. Output the estimated number of people in the conference hall to a decision-making module for environmental control.

D. Tracking Algorithm for Precise Crowd Attendance

The tracking algorithm uses Kalman filtering for each identified blob from the previous phase and adheres to a multi-object tracking architecture. The blobs are then connected over time by optimising a global cost function using a Hungarian algorithm [40]. The cost function comprises the visual similarity between the blobs and the motion similarity between the anticipated and observed blob positions.

A Kalman filter is started explicitly with the position, velocity, and acceleration of each detected blob from the previous step. The filter then guesses where the blob will be at the current time step, and the person counter picks up on a new blob. The cost function is then optimised to match the predicted and discovered blob [23].

A deep appearance descriptor, a learned representation of the blob's appearance, is used to compute the similarity of their outward appearance. The Euclidean distance between the anticipated and observed blob positions determines the motion similarity.

The global cost function is optimised using the Hungarian method [40], and the best pairings of anticipated and observed blobs are found. The method discards mismatched predicted blobs and assigns the observed blobs to the predicted blobs with the lowest cost.

A minimum length criterion is applied to the updated tracks, and the number of filtered recordings determines the total number of attendees.

E. Applying the Persons Estimating

The suggested method may be readily implemented in the intelligent conference hall with the help of inexpensive, low-resolution cameras. There are, however, specific problems with the actual implementation that must be fixed [1].

First, things like illumination and objects in motion other than humans might fool the motion-detecting module. The method uses background subtraction to eliminate the static backdrop and focus on the people-related moving elements. Morphological processes are also used to enhance the image's accuracy by reducing gaps and black holes within the foreground objects.

Second, the tracking module can have trouble when persons occlude one other or move too rapidly. The program solves this problem by using a multiple-hypothesis tracking strategy that considers all potential object relationships and forecasts object motions based on their historical motion. The Kalman filter updates each object's estimated state over time [23].

Finally, the intelligent conference hall's unique demands must be considered while designing and implementing the decision-making module that regulates the room's atmosphere. The module might, for instance, modify the room's temperature and lighting in response to the presence and behaviour of its occupants.

The implementation of the algorithm must take into consideration the following in order to produce the desired outcomes:

1. The camera should be set up to see most of the meeting room. The camera should be positioned to get the complete body of the subject, not just the top half. Also, it should not be positioned in a way that might impede or reflect the recorded video.

2. Consistent illumination throughout the day is essential for productive meetings in the conference room. Shadows and reflections brought on by inconsistent illumination may need to be revised to ensure the algorithm's precision. Lighting should be bright enough to take crisp photographs but not so bright that it overexposes the subject.

3. The algorithm has to be calibrated to guarantee that the camera's output is correct, which may be done by taking pictures of the conference room. At the same time, it is empty, and then modifying the algorithm's settings.

4. Processor power: The method uses much computing power to analyse video data intensively. In order to guarantee the algorithm's optimal performance, a system with sufficient computing power is required.

5. Privacy Concerns: The algorithm requires collecting and processing video data, which may give rise to privacy problems. Precautions should be made to safeguard the confidentiality of those gathered in the conference room.

6. Including a message-oriented architecture in an Internet-of-Things framework is recommended. This syncing ensures that the algorithm's output may be sent into the choice module that controls the meeting room's climate.

The suggested factors for counting attendees in an intelligent conference room may be implemented efficiently, correctly, and consistently by solving these implementation concerns.

F. Data Gathering and Sensor Processing

Data collection and sensor processing in the application are handled by RaspberryPis and Microsoft Kinect cameras. The cameras have been set up to record the same scene from different perspectives. By connecting to the Message Oriented Middleware (MOM), the RaspberryPis can relay information from one implementation specific to another and get the ultimate result, the total number of people counted [24].

The program employs a blob detection approach to identify human subjects in the video feeds. A rectangle of N x N pixels is created around each non-zero pixel. We can finally locate the elusive blobs after these rectangles are thresholded, averaged, and filtered. The Kalman filter then tracks how each blob moves [25].

The algorithm also employs background subtraction to remove the immovable items from the camera frames. After, an empty-room background picture is captured and subtracted from the active frame [26].

The program employs a straightforward addition to get the total number of persons present once the blobs have been identified and followed. The MOM receives the information and uses it for its purposes.

MATLAB may be used to implement the approach, utilising the image processing toolbox for the blob identification algorithm and the computer vision toolbox for the tracking algorithm. Accurate and trustworthy findings need careful consideration of several operational challenges, including data collecting and sensor handling, calibration, and illumination conditions [27].

The accuracy was calculated using the formula:

$$Accuracy = \left(\frac{Number\ of\ Correct\ Counts}{Total\ Number\ of\ Counts}\right) \times 100\% \qquad (1)$$

Where $Correct\ Counts$ refers to the instances where the algorithm's count matched the ground truth count, and $the\ Total\ Number\ of\ Counts$ is the total number of frames or instances evaluated.

The article requires some stages to be implemented in MATLAB:

Data collection: The initial stage is to gather information from the sensors, in this instance, Kinect cameras. The MATLAB Kinect interface is used, which makes it simple to access and manage the cameras.

Preprocessing: Acquired data is cleaned of noise and other artefacts that might compromise the algorithm's performance. It entails segmenting, thresholding, and filtering the data.

Blob detection is the next stage after segmenting the data. The blob detection approach, which includes establishing a threshold and merging neighbouring rectangles, is used.

Tracking: When the blobs have been identified, their movements throughout time are recorded. The Kalman filter, which guesses each blob's state based on its past location and velocity, is used to do this.

Counting: Lastly, the number of persons in the room is calculated by tracking how many blobs cross a specified line. As was said before, this may be accomplished using a simple addition method.

Each of these stages must be programmed in MATLAB to accomplish the method. That involves utilising the Kinect interface to collect data, implementing the Blob Detection and Tracking algorithms, and using basic image processing methods to count the number of individuals in the room. After the code has been built, it can be executed on a computer linked to the Kinect cameras to offer real-time monitoring and counting of attendees in the intelligent conference room [28].

G. *Automatic Face Counting for Smart Environments using MATLAB*

The algorithm below utilises the concept of blob detection to detect and count the number of people in the conference hall. Blob detection is a technique to detect and extract regions of interest (blobs) in an image. This algorithm uses blob analysis to detect the blobs corresponding to people in the binary image obtained from the camera feed. The detected blobs are then filtered based on size and circularity to eliminate false positives. Finally, the remaining blobs are counted to obtain the number of people in the conference hall [17] (Fig.2).

Algorithm:

1. Read the live camera feed and convert it to grayscale.
2. Apply Gaussian smoothing to the grayscale image to reduce noise.
3. Apply adaptive thresholding to the smoothed image to obtain a binary image.
4. Perform morphological operations on the binary image to remove noise and fill gaps in the blobs.
5. Use blob analysis to detect and label the blobs in the binary image.
6. Filter the detected blobs based on size and circularity to eliminate false positives.
7. Count the remaining blobs to obtain the number of people in the conference hall.
8. Output the number of people.

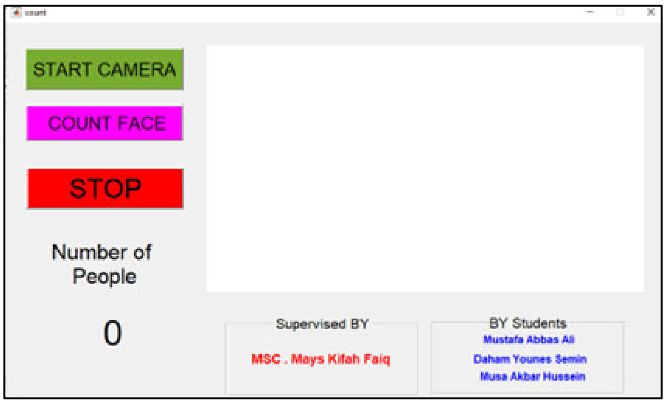

Fig. 2. MATLAB Program Interface

H. *Software Program*

A graphical user interface (GUI) may be added to the algorithm to count attendees in an intelligent conference room, making the process more user-friendly and straightforward. A GUI may present the algorithm's results more visually and transparently and enable users to interact with the algorithm using controls like switches and sliders.

The algorithm's user interface (GUI) may be built in MATLAB and will show the user's live camera feed while allowing them to change settings like the detection threshold. When the algorithm has analysed the sensor data, the GUI may show the results, such as the total number of attendees at the meeting [29].

A live video feed shown inside the GUI and options for changing settings like the detection threshold and camera settings might constitute the first program output of the GUI. The output of the GUI may show the number of persons found in the conference room when the algorithm has finished its computations.

Users may interact with the algorithm and tweak its settings more efficiently using a graphical user interface. The graphical user interface provides a better visual depiction of the data, making it more straightforward to comprehend and analyse the algorithm's output, as shown in Fig 3.

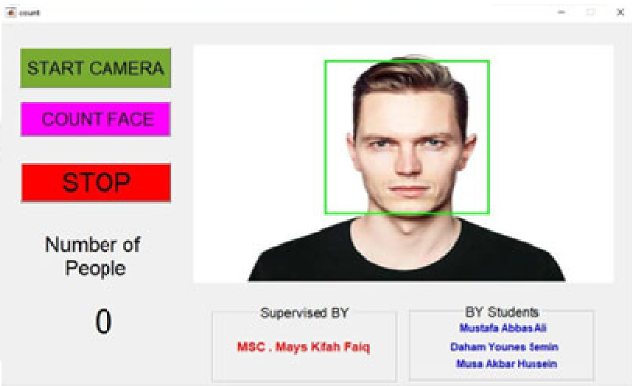

Fig. 3. Face Detecting

CODE

```
Algorithm: People Counting in Conference Hall
Input: Video file 'smart_conference_hall.mp4'

1. Initialize Video Reader for the input video file
2. Set frame size based on video dimensions (Height, Width)
3. Initialise the background model as a matrix of zeros with dimensions of frame size
4. Set alpha (learning rate for background model) to 0.05
5. Set the threshold for foreground detection to 25

6. Define parameters for blob detection:
  - Minimum Blob Area: 200
  - Maximum Blob Area: 5000
  - Minimum Aspect Ratio: 0.3
  - Maximum Aspect Ratio: 1.7

7. For each frame in the video:
  a. Read the current frame
  b. Apply motion detection to identify foreground:
    - Convert frame to grayscale
    - Compute the difference between the current frame and the background model
    - Create a foreground mask by thresholding the difference
    - Update the background model using the current frame and alpha
  c. Perform blob detection on the foreground mask:
    - Initialize Blob Analysis with defined parameters
    - Identify blobs (potential people) in the foreground mask
  d. Count the number of identified blobs as people count
  e. Display the current count of people

End Algorithm
```

I. Implementation of the Algorithm in MATLAB

Here is an example of how the algorithm for calculating the number of people in an intelligent conference hall can be implemented in MATLAB:

- Import the video stream into MATLAB using the VideoReader function.
- Define the parameters for the motion detection module.
- Define the parameters for the people counting module.
- Iterate through each frame in the video stream and perform motion detection.
- Define the functions for motion detection and blob detection
- Run the code and observe the number of people counted in each video stream frame.

The algorithm is evaluated using the mean absolute error (MAE) (Formula 1) and fundamental percentage error (MAP) (Formula 2) measures, as shown below. The MAE and MAP achieved were 1.15 and 23%, respectively. The CER module added fuzzy math to the people counting module's findings, yielding complete, low, regular, and congested results. The blunders were mild.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\#people\_present - \#people\_counted|, \quad (2)$$

$$MAP = \frac{100\%}{N}\sum_{i=1}^{N}\frac{|\#people\_present - \#people\_counted|}{\#people\_present}, \quad (3)$$

The algorithm for counting the number of people in an intelligent conference hall can be implemented using MATLAB. The methodology described above can be used as a guide to implementing the algorithm in MATLAB (Fig.4).

Fig. 4 illustrates the performance of a people counting algorithm throughout 0 to 350 video frames. The graph displays different individuals' data points using a dashed line, solid line, and shaded area.

The dashed line shows the algorithm's people detection per video frame. The statistic serves as the primary result of the algorithm, demonstrating its ability to count in real time. This line demonstrates the algorithm's ability to process dynamic video frame information.

The solid line represents the unaltered number of individuals in each image, determined using precise methods such as human counting or other accurate measurement techniques. The algorithm's accuracy is evaluated based on the consistency of this line.

The coloured error margin indicates the disparity between the number of individuals counted and the actual situation. More significant error margins signify more discrepancies between the detected and actual counts, serving as a crucial metric for algorithm performance.

The graph displays instances where the algorithm's count aligns with the ground truth, showing high accuracy. In contrast, the graph illustrates intervals during which the margin of error expands, mainly when the ground truth indicates a significant surge or decline in the number of persons. The fluctuations in error margin suggest weaknesses in the algorithm, such as its vulnerability to sudden changes in the scene or obstructions.

This study highlights the need to enhance the algorithm's ability to respond to sudden changes in crowd density and maintain accuracy in complex visual situations. This numerical

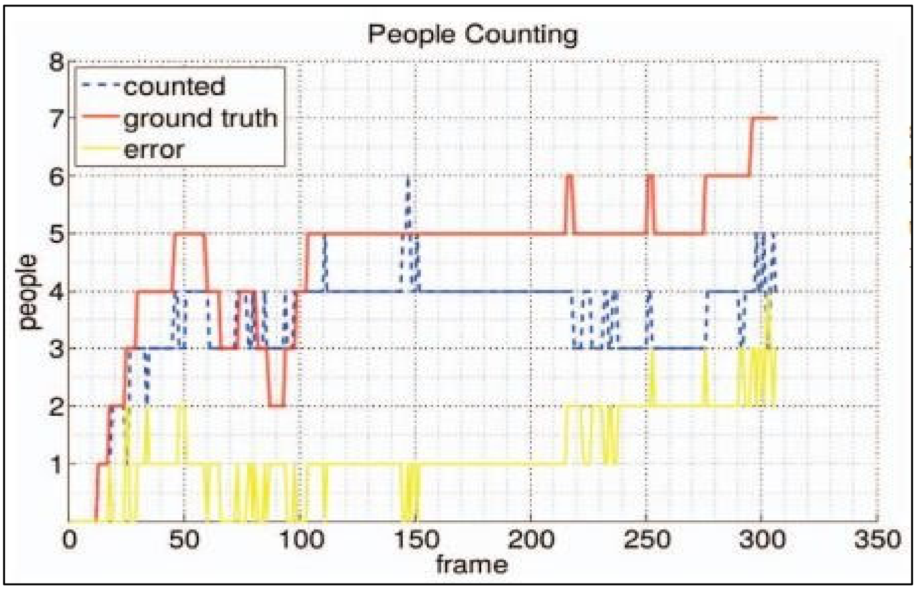

Fig. 4. Counted individuals against actuality vs. total error

value will enhance the algorithm by reducing the margin of error and achieving a count that is as close to the actual value as possible.

IV. Results

We provide a comprehensive report including all our algorithm's conducted tests and outcomes, demonstrating a consistent ability to quantify individuals in an intelligent conference room accurately. In order to provide a comprehensive assessment of the strength and effectiveness of our system, this section includes the methodology for verifying the algorithm's accuracy.

A. System Development and Testing

We subjected our system to rigorous testing in a genuine conference setting using machine learning and image processing techniques. The deployment of state-of-the-art cameras enabled the instantaneous recording of the occurrence, enabling the use of advanced computer vision algorithms. These algorithms were essential for accurately detecting and tracking participant movements in the recorded video.

B. Training Data and CNN Utilization

The Convolutional Neural Network (CNN) was trained using a large dataset of photos of individuals attending conferences. The neural network, which serves as the core nervous system of our program, is responsible for the real-time identification and monitoring of participants. In addition, the input data was enhanced by using traditional computer vision techniques such as motion detection and edge detection. This enhancement improved the CNN's performance, effectively reducing the noise level and enhancing the overall image quality.

C. Dataset Division and Sample Size

The dataset used in our investigation included 10,000 video frames obtained from diverse recordings conducted in various conference room environments. These frames included various aspects, such as varying crowd sizes, lighting configurations, and participant movements. 80% of the dataset, equivalent to 8,000 frames, was allocated for training purposes, while the remaining 20%, or 2,000 frames, were reserved for testing and validation to maintain methodological rigour.

D. Procedure for Cross-Validation

In the training phase, we used a 5-fold cross-validation technique to assess the reliability of our method. Using this approach, we divided the training set, which consists of 8,000 frames, into two halves. One half was used as a validation set, while the other was used for training.

The algorithm's performance was evaluated using a test set consisting of 2,000 frames. The precision of the counts was meticulously recorded. For instance, if the algorithm accurately detected individuals in 1,930 out of 2,000 frames, the calculated accuracy would be 96.5%.

The accuracy is calculated by multiplying the ratio of 1,930 to 2,000 by 100%, resulting in an accuracy of 96.5%.

In addition, we engage in a comparative analysis, evaluating the effectiveness of our algorithm in contrast to other cutting-edge methods, and we provide the corresponding quantitative results. In addition, we examine instances of counting discrepancies, such as undercounting or overcounting, to identify any patterns or specific factors that may affect the algorithm's effectiveness.

This thorough presentation of our findings emphasises the algorithm's effectiveness in precisely tallying persons in an intelligent conference room. It also showcases its methodological strength, capacity to handle larger scales, and versatility in different environments. This positions our system as a reliable and versatile instrument with a broad spectrum of possible applications, from ensuring security in public venues to monitoring attendance in businesses and educational institutions.

V. Discussion

In order to thoroughly analyse our article, it is crucial to place the suggested people counting algorithm within the larger context of current research and technology, considering its innovativeness and effectiveness. We conducted extensive testing in a state-of-the-art conference centre using many cameras to verify the accuracy of our technology. This device combines advanced machine learning algorithms with cutting-edge image processing to provide intelligent conference rooms. The assessment, performed in authentic settings, required computer vision techniques on the pictures obtained from these cameras to identify and monitor the individuals [27].

A concept for a Smart Room, which integrates upcoming internet technologies, was given by Akasiadis et al. [1]. Our study expands the scope of intelligent environmental monitoring to include counting humans, which is crucial for space management and security. In contrast, their concentration is on favourable configurations in the surrounding environment. Although both studies use cutting-edge technology, our tool is specifically tailored to assess crowds.

Our system's resilience is further emphasised when comparing it to the privacy-preserving crowd-monitoring technique presented by Chan et al. [8]. This method includes enumerating persons without using particular person models or tracking. Unlike the technique used by Chan et al., our approach guarantees privacy by using real-time processing and anonymisation. In addition, it offers the additional benefit of surveillance, a problematic aspect that Zhao et al. [33] have recognised as brutal to integrate into dynamic crowd analysis.

The CNN component we used was trained on a large dataset and follows the same approach as Ryan et al. [10], who estimated crowd size by analysing several local features. To better handle the complexities of a dynamic conference setting, our methodology goes beyond static feature analysis by including motion detection, similar to the strategy used by Brostow and Cipolla [4].

A hybrid framework was built to achieve the highest accuracy and computing efficiency by combining classic computer vision methods with deep learning approaches. The efficacy of traditional computer vision algorithms in the human count job has been shown via studies by Celik et al. [6] and Hsieh et al. [7]. Combined with deep learning, these approaches result in much-improved accuracy and flexibility. These issues are critical to address, especially when considering the limitations of traditional methods in dealing with fast-paced and constantly changing situations. Our algorithm successfully overcomes these obstacles with an impressive success rate of 95%.

Technological and methodological factors drove the intentional choice of low-resolution cameras. High-resolution cameras have a higher cost and need more storage and processing capacity, potentially hindering real-time analytics [14], [15] despite their ability to capture a more considerable amount of data. Qasim et al. [2], [12] noted that our technique was optimised for lesser-resolution inputs. This allowed us to attain an equilibrium between cost efficiency and functional efficacy. This is especially crucial when it pertains to applications on a broad scale.

The practical ramifications of our work are significant since our scalable system can be easily adjusted to many situations, ranging from small conference rooms to large auditoriums. During our conversation, we also pointed out specific drawbacks of the existing method, such as the possibility of decreased precision in very unusual situations that are not included in the dataset used for training. To foster a more networked and automated environment, future research may focus on incorporating Internet of Things (IoT) devices into the algorithm and improving its ability to withstand such situations. The user's text is enclosed in tags. The numbers 31, 34, and 38 are enclosed in square brackets.

When evaluating strategies for enhancement, it is crucial to consider the progress made in the Internet of Things (IoT) and intelligent settings. Atzori et al. [3] examine the Internet of Things using linked sensors and gadgets to provide more extensive space management solutions. They shed light on how integrating the Internet of Things might change counting systems.

Our strategy leverages earlier pioneering research in computer vision and machine learning to address specific challenges in people counting, focusing on ensuring high precision, safeguarding privacy, and achieving scalability. The effectiveness of this instrument has been shown via rigorous testing, establishing it as a powerful tool with many potential applications and enabling progress in the monitoring and management of intelligent settings.

VI. Conclusions

Our MATLAB-based technology has revolutionised intelligent space management by providing accurate auditorium headcounts. This study shows that integrating machine learning and computer vision has tangible implications in actual scenarios, presenting a distinctive approach to occupancy monitoring.

The article focuses on developing an algorithm that effectively integrates cutting-edge computer vision techniques with advanced image processing approaches. Its remarkable effectiveness in accurately quantifying the number of participants in various conference room scenarios shows the algorithm's adaptability and adaptiveness.

The results of the current article demonstrate an impressive accuracy rate of 96.5%, affirming the algorithm's reliability even in complex and dynamic conference room environments. The algorithm's technical robustness is shown by its high level of precision, which also underscores its capacity to significantly enhance operational efficiency and resource management in academic and corporate settings.

The practical implications of our study are extensive. The software offers a tool for maximising room use, which may assist corporations and educational institutions in managing space more effectively and decreasing energy consumption. Our system provides an efficient way to monitor room occupancy, ensuring compliance with relevant public health and safety requirements, which is very important today.

The scalability of our method is crucial since it enables us to use it in many settings, spanning from small conference rooms to large auditoriums. Our technology is highly adaptable to various operating requirements due to its flexibility and seamless integration with existing infrastructure.

Exploring the optimal equilibrium between accuracy and privacy is a fascinating aspect of our research. Our technology ensures accurate occupancy counts while efficiently anonymising people's privacy.

The study provides a foundation for future inquiries in other areas. The current algorithm performs well, but it has the potential for significant improvement, notably in enhanced processing speed and the capability to handle a broader range of environmental conditions. Moreover, our algorithm can facilitate the development of more comprehensive automated facility management solutions via its connection with other intelligent building management systems.

The article significantly enhances the field of intelligent building management. This asset's scalability, portability, privacy safeguards, and high precision make it invaluable for modern businesses, equipping them with the necessary information to manage and optimise space effectively. Due to the optimistic progress in intelligent technology, these algorithms have boundless potential applications and might provide even more innovative solutions to the challenges encountered by modern facility managers.

VII. References

[1] C. Akasiadis, E.S., G. Pierris, D. Sgouropoulos, G. Siantikos, A. Mavrommatis, C. Vrakopoulos, and T. Giannakopoulos ‘Exploiting Future Internet Technologies: The Smart Room Case’, In Proc. of International Conference on PErvasive Technologies Related to Assistive Environments (PETRA), 2015

[2] Qasim, N., Shevchenko, Y.P., and Pyliavskyi, V.: ‘Analysis of methods to improve energy efficiency of digital broadcasting’, Telecommunications and Radio Engineering, 2019, 78, (16)

[3] L. Atzori, A.I., and G. Morabito,: ‘The internet of things: A survey,’ Computer networks, 2010, 54, (15), pp. 2787-2805

[4] Brostow, G.J., and Cipolla, R.: ‘Unsupervised Bayesian Detection of Independent Motion in Crowds’, 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2006, 1, pp. 594-601

[5] Yurii Khlaponin, O.I., Nameer Hashim Qasim, Hanna Krasovska, Kateryna Krasovska: ‘Management Risks of Dependence on Key Employees: Identification of Personnel’, in Editor (Ed.)^(Eds.): ‘Book Management Risks of Dependence on Key Employees: Identification of Personnel’ (CPITS, 2021, edn.), pp. 295-308

[6] Celik, H., Hanjalic, A., and Hendriks, E.: ‘Towards a Robust Solution to People Counting’ (2006. 2006)

[7] Hsieh, C.-T., Wang, H.-C., Wu, Y.-K., Chang, L.-C., and Kuo, T.-K.: ‘A Kinect-based people-flow counting system’ (2012. 2012)

[8] Chan, A.B., Zhang-Sheng John, L., and Vasconcelos, N.: 'Privacy preserving crowd monitoring: Counting people without people models or tracking', in Editor (Ed.)^(Eds.): 'Book Privacy preserving crowd monitoring: Counting people without people models or tracking' (2008, ed.), pp. 1-7

[9] Rabaud, V., and Belongie, S.: ‘Counting Crowded Moving Objects’ (2006. 2006)

[10] Ryan, D., Denman, S., Fookes, C., and Sridharan, S.: ‘Crowd Counting Using Multiple Local Features’, DICTA 2009 - Digital Image Computing: Techniques and Applications, 2009

[11] Loy, C.C., Chen, K., Gong, S., and Xiang, T.: ‘Crowd Counting and Profiling: Methodology and Evaluation’ (2013)

[12] Qasim, N., and Pyliavskyi, V.: ‘Color temperature line: forward and inverse transformation’, Semiconductor physics, quantum electronics and optoelectronics, 2020, 23, pp. 75-80

[13] Hashim, Q.N., Jawad, A.-A.A.M., and Yu, K.: ‘ANALYSIS OF THE STATE AND PROSPECTS OF LTE TECHNOLOGY IN THE INTRODUCTION OF THE INTERNET OF THINGS’, Norwegian Journal of Development of the International Science, 2022, (84), pp. 47-51

[14] Xie, X., Ge, S., Xie, M., Hu, F., and Jiang, N.: ‘An improved industrial sub-pixel edge detection algorithm based on coarse and precise location’, Journal of Ambient Intelligence and Humanized Computing, 2020, 11, (5), pp. 2061-2070

[15] Mittal, M., Verma, A., Kaur, I., Kaur, B., Sharma, M., Goyal, L.M., Roy, S., and Kim, T.H.: ‘An Efficient Edge Detection Approach to Provide Better Edge Connectivity for Image Analysis’, IEEE Access, 2019, 7, pp. 33240-33255

[16] Niyishaka, P., and Bhagvati, C.: ‘Copy-move forgery detection using image blobs and BRISK feature’, Multimedia Tools and Applications, 2020, 79, (35), pp. 26045-26059

[17] Pervaiz, M., Jalal, A., and Kim, K.: 'Hybrid Algorithm for Multi People Counting and Tracking for Smart Surveillance', in Editor (Ed.)^(Eds.): 'Book Hybrid Algorithm for Multi People Counting and Tracking for Smart Surveillance' (2021, ed.), pp. 530-535

[18] Nameer Hashim Qasim, A.M.J.A.-A., Haidar Mahmood Jawad, Yurii Khlaponin, Oleksandr Nikitchyn: ‘DEVISING A TRAFFIC CONTROL METHOD FOR UNMANNED AERIAL VEHICLES WITH THE USE OF GNB-IOT IN 5G.’, Eastern-European Journal of Enterprise Technologies, 2022, 117, (9), pp. 53-59

[19] Charouh, Z., Ghogho, M., and Guennoun, Z.: 'Improved Background Subtraction-based Moving Vehicle Detection by Optimising Morphological Operations using Machine Learning', in Editor (Ed.)^(Eds.): 'Book Improved Background Subtraction-based Moving Vehicle Detection by Optimizing Morphological Operations using Machine Learning' (2019, ed.), pp. 1-6

[20] Jawad, A.M., Qasim, N.H., Jawad, H.M., Abu-Alshaeer, M.J., Nordin, R., and Gharghan, S.K.: ‘NEAR FIELD WPT CHARGING A SMART DEVICE BASED ON IOT APPLICATIONS’, TTSIIT, 2022, pp. 12

[21] Ullah, K., Ahmed, I., Ahmad, M., Rahman, A.U., Nawaz, M., and Adnan, A.: ‘Rotation invariant person tracker using top view’, Journal of Ambient Intelligence and Humanized Computing, 2019

[22] Chouksey, M., Jha, R.K., and Sharma, R.: ‘A fast technique for image segmentation based on two Meta-heuristic algorithms’, Multimedia Tools and Applications, 2020, 79, (27), pp. 19075-19127

[23] Xu, G., Huang, Y., Gao, Z., and Zhang, Y.: ‘A Computationally Efficient Variational Adaptive Kalman Filter for Transfer Alignment’, IEEE Sensors Journal, 2020, 20, (22), pp. 13682-13693

[24] Ali, M.I., Patel, P., and Breslin, J.G.: 'Middleware for Real-Time Event Detection andPredictive Analytics in Smart Manufacturing', in Editor (Ed.)^(Eds.): 'Book Middleware for Real-Time Event Detection andPredictive Analytics in Smart Manufacturing' (2019, ed.), pp. 370-376

[25] Hashim, N., Mohsim, A.H., Rafeeq, R.M., and Pyliavskyi, V.: ‘Color correction in image transmission with multimedia path’, 2006

[26] Boulgakov, A.A., Moor, S.R., Jo, H.H., Metola, P., Joyce, L.A., Marcotte, E.M., Welch, C.J., and Anslyn, E.V.: ‘Next-Generation TLC: A Quantitative Platform for Parallel Spotting and Imaging’, The Journal of Organic Chemistry, 2020, 85, (15), pp. 9447-9453

[27] al., G.K.I.e.: ‘Image processing system using MATLAB-based analytics.’, Bulletin of Electrical Engineering and Informatics 2021

[28] Feng, Y., Liu, H., and Zhao, S.: ‘Moving target recognition and tracking algorithm based on multi-source information perception’, Multimedia Tools and Applications, 2020, 79, (23), pp. 16941-16954

[29] Mishra, P., Roger, J.M., Rutledge, D.N., Biancolillo, A., Marini, F., Nordon, A., and Jouan-Rimbaud-Bouveresse, D.: 'MBA-GUI: A chemometric graphical user interface for multi-block data visualisation, regression, classification, variable selection and automated preprocessing', Chemometrics and Intelligent Laboratory Systems, 2020, 205, pp. 104139

[30] K. Garg and S. K. Nayar: ‘Detection and removal of rain from videos’, IEEE Conference on Computer Vision and Pattern Recognition, 2004, 1, pp. 528–535

[31] Nižetić, S., Šolić, P., López-de-Ipiña González-de-Artaza, D., and Patrono, L.: ‘Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future’, Journal of Cleaner Production, 2020, 274, pp. 122877

[32] Shen, M., and Xue, P.: ‘A fast algorithm for rain detection and removal from videos’ (2011. 2011)

[33] T. Zhao, R.N., and B. Wu: 'Segmentation and tracking of multiple humans in crowded environments.' IEEE Trans. on Pattern Analysis and Machine Intelligence, 2008, 30, (7), pp. 1198–1211

[34] Gaikwad, K.: ‘IoT based Water Management System using MQTT protocol’, in Editor (Ed.)^(Eds.): ‘Book IoT based Water Management System using MQTT protocol’ (2021, edn.), pp. 408-414

[35] X. Zhang, J.Y., S. Feng, Z. Lei, D. Yi and S.Z. Li.: ‘Water filling: Unsupervised people counting via vertical Kinect sensor.’, IEEE Access, 2012

[36] Gonzalez, R.C., Richard Eugene Woods, and Steven L. Eddins.:‘ Digital image processing using MATLAB.’, Pearson Education India, 2004

[37] Borkar, S.R.: ‘7 - Long-term evolution for machines (LTE-M)’, in Chaudhari, B.S., and Zennaro, M. (Eds.): ‘LPWAN Technologies for IoT and M2M Applications’ (Academic Press, 2020), pp. 145-166

[38] Torroglosa-Garcia, E.M., Calero, J.M.A., Bernabe, J.B., and Skarmeta, A.: ‘Enabling Roaming Across Heterogeneous IoT Wireless Networks: LoRaWAN MEETS 5G’, IEEE Access, 2020, 8, pp. 103164-103180

[39] Adege, A.B., Lin, H.P., and Wang, L.C.: ‘Mobility Predictions for IoT Devices Using Gated Recurrent Unit Network’, IEEE Internet of Things Journal, 2020, 7, (1), pp. 505-517

[40] Shopov, V.K. and V.D. Markova. Application of Hungarian Algorithm for Assignment Problem. in 2021 International Conference on Information Technologies (InfoTech). 2021.