INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

Real-Time Crowd Monitoring System

Hrishikesh Gaikwad, Sumit Jadhav, Nirbhaya Gunjal, Sushant Survase, Mr. Kaustubh Shinde

Student, Dept of Computer Engineering, SITS, Narhe, Pune, Maharashtra, India

Student, Dept of Computer Engineering, SITS, Narhe, Pune, Maharashtra, India

Student, Dept of Computer Engineering, SITS, Narhe, Pune, Maharashtra, India

Student, Dept of Computer Engineering, SITS, Narhe, Pune, Maharashtra, India

Professor, Dept of Computer Engineering, SITS, Narhe, Pune, Maharashtra, India

ABSTRACT: The use of video-based monitoring systems for crowd analysis is becoming increasingly important due to population growth and the high cost of human monitoring. This paper proposes a framework for detecting, tracking, and counting crowds using video-based monitoring systems. Compared to sensor-based and human-based solutions, video-based systems offer more flexible functionalities, better performance, and lower costs.Crowd management is a crucial research area that requires attention to prevent potential losses, disasters, and accidents. The integration of different crowd detection and monitoring techniques can enhance performance and control compared to limited stand-alone techniques. Crowd management involves accessing and interpreting information sources, predicting crowd behavior, and deciding on the use of interventions based on context. The paper concludes that more investigative work is needed to further advance the field of crowd management.

KEYWORDS: Crowd Detection, Tracking, Image Processing.

1.INTRODUCTION

Crowd management has become an increasingly important research area due to the potential losses, disasters, and accidents that could occur if it were neglected. The rise in video-based monitoring systems has led to a growing interest in the field of computer vision. The collection of information of people passing by surveillance cameras has opened up new avenues for studying crowd control, detection, and tracking, based on the acquired dataset. This dataset can be used for various purposes, such as counting, surveying, and monitoring the population in a specific area or controlling traffic in the same area. With the aid of different methods and algorithms developed for image and video processing, video sequences obtained through observation cameras can be analyzed to extract the required information. Automated tools based on computers and recent technologies like laser, RFID, Wi-Fi, Bluetooth, and AI have been developed to detect and recognize crowds. The second phase of managing crowds is to monitor, track, and analyze the detected crowd to obtain reliable insights. Many researchers have investigated this topic using theoretical, statistical, data mining, machine learning, and prediction techniques. The detection, tracking, counting, etc. tools are tallied by advanced programmed software, making crowd management an active and flourishing research area that needs attention. In recent years, the need for effective crowd monitoring has increased due to the growing concerns of security, safety, and efficiency in various public places, such as airports, railway stations, and sports arenas. The use of video-based monitoring systems for crowd analysis has gained popularity as they offer more flexible functionalities, enhanced performance, and lower costs compared to sensor-based and human-based monitoring systems.

2. PROBLEM

The problem statement highlights the need for an automated system that can count and monitor people in various public places such as universities, shopping malls, railway stations, and airports. Traditional methods of manual counting and monitoring are time-consuming and impractical, especially in areas with a high volume of foot traffic. This is where an automated system that uses video surveillance can provide valuable insights. Such a system can be developed using deep learning techniques that enable the detection and tracking of individuals in real-time. By analyzing the video footage, the system can count the number of people present in a specific area and monitor their movements.

This information can be used for various purposes such as improving crowd control, optimizing resource allocation, and enhancing security measures. For instance, in a university setting, the system can monitor the number of students present in a lecture hall or a library. This information can be used to optimize the use of space and resources and prevent overcrowding. Similarly, in a shopping mall or an airport, the system can help to monitor the movement of people and

prevent the formation of queues or overcrowding in certain areas. Overall, an automated crowd detection and monitoring system using video surveillance and deep learning techniques can provide valuable insights into the behavior and movement of people in public places. This can lead to more efficient resource allocation, better crowd control, and enhanced security measures.

3. SOLUTION

The proposed methodology for crowd detection and tracking involves a framework that can be used to analyze and count human beings in a crowd. The first step in this process is to subtract the background and remove unwanted pixels from the image. This method helps to identify and count people in the crowd. The database used to test the performance of the proposed system is also described in detail, which helps to identify the limitations of the model. By using this methodology, it will be possible to develop an automated system that can provide meaningful information fromrecorded videos, thereby reducing the need for manual monitoring and counting of people in various places such as universities, shopping malls, railway stations, airports, and other crowded areas. The proposed system will be able to detect and track crowds in real-time using Frame by Frame analysis, advanced algorithms and image processing techniques.

4. PROPOSED SYSTEM

Generate Train set and Test set: In this this phase we first create training and testing dataset for proposed system. The basic objective of this module to generate the ground truth values for both training and testing dataset.
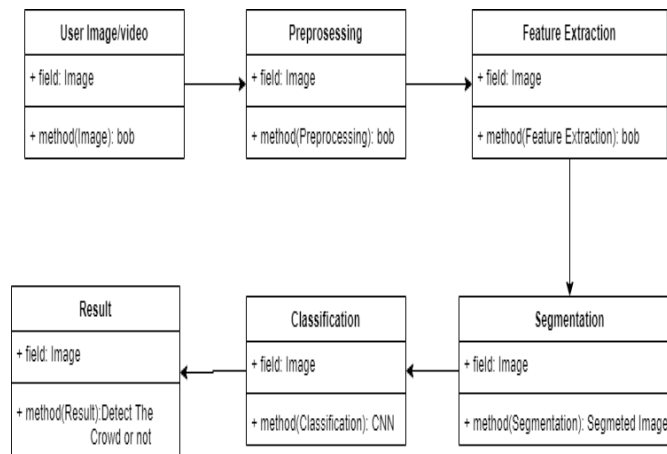
Three different features have been extracted from each image like height, width and channel. It extracts the actual pixel values of each image during data creation. The outcome this process the .csv files both training and testing respectively. The crowd monitoring system will detect the number of people count them and it will show to count as frame count.

Python: Python is a popular high-level programming language that is widely used in machine learning and computer vision applications, including crowd monitoring detection systems. Python's popularity in these domains is mainly due to its simplicity, readability, and ease of use, as well as the availability of many open-source libraries and tools that can be used for developing such systems. In the context of crowd monitoring detection systems, Python can be used to perform a range of tasks, including image processing, data analysis, and machine learning. For example, Python libraries such as OpenCV, NumPy, and SciPy can be used for image processing tasks such as image segmentation, feature extraction, and classification. Similarly, machine learning libraries such as TensorFlow, Keras, and PyTorch can be used for training and deploying machine learning models, for crowd detection and analysis.

CNN (Training and Testing): Convolutional Neural Networks (CNNs) are a type of deep learning algorithm that is particularly well-suited for image processing tasks, such as crowd detection. CNNs are designed to learn features from images in a hierarchical manner, where lower layers learn simple features, such as edges and corners, and higher layers learn more complex features, such as shapes and objects. In the context of crowd detection, CNNs can be trained on a large dataset of images and videos, such as aerial photographs or security camera footage, to learn how to identify patterns and features that are indicative of a crowd. The CNN algorithm works by processing the input image through a

series of convolutional layers, pooling layers, and fully connected layers. The convolutional layers use a set of filters to scan the input image and extract features that are relevant to the task of crowd detection. These filters learn to identify patterns and edges in the image, such as the shapes of individuals or groups of individuals in the crowd. The pooling layers down sample the output of the convolutional layers by selecting the most relevant features and reducing the resolution of the image. This helps to reduce the computational complexity of the algorithm and prevent overfitting to the training data. The fully connected layers take the output of the convolutional and pooling layers and use it to classify the input image as containing a crowd or not. This is done by mapping the extracted features to a set of output classes, such as "crowd" or "no crowd," using a set of learned weights. To train the CNN algorithm for crowd detection, a large dataset of labeled images is required. The images in the dataset are first preprocessed to normalize the pixel values and apply data augmentation techniques to increase the size of the dataset. The CNN is then trained using a loss function, such as binary cross-entropy, which measures the difference between the predicted output and the true label. Once the CNN is trained, it can be used to detect crowds in new images by applying the same convolutional filters and fully connected layers to the input image. The output of the CNN is a probability score indicating the likelihood that the input image contains a crowd. Overall, CNNs are a powerful tool for crowd detection systems, as they can learn to extract useful features from images and classify them with high accuracy. By training CNNs on large datasets of labeled images, developers can create robust and effective crowd detection systems that can be used for various applications, such as crowd control, security, and event planning.

TensorFlow Library Module: In the first module we implement the access interfaces and should be customized for every deep learning tool called TensorFlow. With the help of this APIs often need to be compatible with application's source code.

Deep Learning: Deep learning is a subset of machine learning that uses neural networks with multiple layers to learn representations of data. It has been shown to be highly effective in computer vision tasks, such as image classification, object detection, and segmentation, making it a useful tool for crowd detection systems. Deep learning algorithm, such as Convolutional Neural Networks (CNNs), can be trained on large datasets of crowd images to learn how to identify patterns and features that are indicative of a crowd. These algorithms can extract useful information from images, such as crowd size, density, and movement patterns, which can be used for various applications, such as crowd control, security, and event planning. In the context of crowd detection, deep learning algorithms can be used to perform a range of tasks, including crowd counting, crowd segmentation, and crowd behavior analysis. For example, a CNN can be trained to detect the presence of crowds in images by learning to identify patterns of individuals and their distribution across the image. Similarly, Deep learning algorithms can be trained using a supervised or unsupervised learning approach. In supervised learning, the algorithm is trained on a labeled dataset of crowd images, where each image is labeled with the number of individuals in the image or whether it contains a crowd or not. In unsupervised learning, the algorithm is trained on an unlabeled dataset of crowd images and is tasked with learning to identify patterns and features in the data without any explicit guidance.

5. CROWD DETECTION
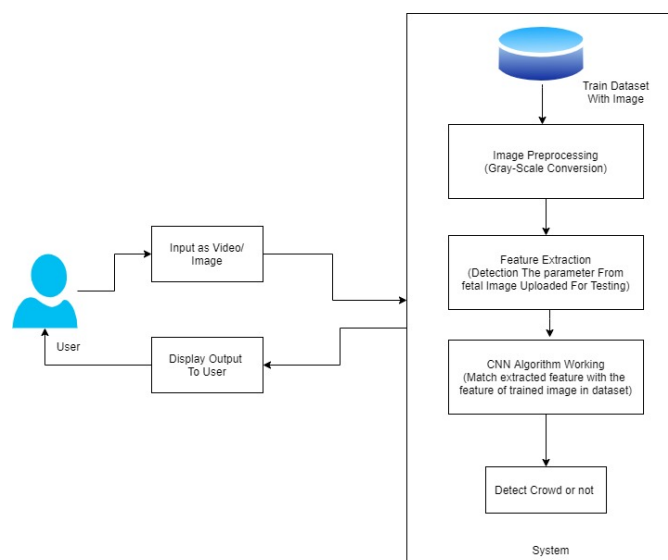
A crowd detection system is a system designed to detect and monitor the presence and movement of crowds in real-time using computer vision and machine learning techniques. These systems are used in a variety of applications, including public safety, event management, and transportation. The main purpose of a crowd detection system is to provide real-time insights into crowd behavior and movement that can help prevent and respond to potential crowd-related incidents such as overcrowding, stampedes, and unauthorized gatherings. These systems use various techniques such as object detection, segmentation, and tracking to identify individuals and groups within the crowd and estimate crowd density. A typical crowd detection system consists of multiple cameras and sensors placed strategically in public spaces to capture visual and environmental data. The data is then processed using algorithms and techniques such as convolutional neural networks (CNNs) to extract features from images and classify them with high accuracy.

Image Processing: Public detection using image processing frame by frame and using gray scale conversion is a commonly used technique in crowd detection. The basic idea is to convert the video frames to grayscale, which helps in reducing the amount of information that needs to be processed. Grayscale images only contain luminance information, which makes them easier to work with than full-color images. Once the video frames are converted to grayscale, image processing techniques can be used to detect and track people in the scene. One approach is to use a technique called background subtraction, which involves subtracting the current frame from a background model to detect moving objects in the scene. The basic idea behind image processing frame by frame is to analyze each frame of the video separately, and extract features that can be used to identify and track people in the scene. These features can include properties such as color, texture, and shape, which can be used to distinguish people from the background and other objects in the scene. One common technique used in frame-by-frame image processing for crowd detection is background subtraction. This involves creating a model of the scene's background, and then subtracting the current frame from this model to detect moving objects in the scene. By comparing each frame to the background model, it is possible to detect and track people as they move through the scene. Another technique used in frame-by-frame image processing for crowd detection is blob detection. This involves identifying regions of pixels in the image that have similar properties, such as color or intensity, and grouping them together into blobs. By analyzing the properties of these blobs, it is possible to identify and track individual people in the scene.

Feature Extraction:Feature extraction is a fundamental step in computer vision and machine learning, and it involves identifying and extracting important information or features from raw images that can be used for further analysis, such as classification, detection, or segmentation. These features can be statistical or structural, and can capture information such as texture, shape, or color.

The process of feature extraction can be broken down into several elements, including:

Preprocessing**:** The first step in feature extraction is often preprocessing, which involves preparing the image for analysis. This can include steps such as resizing, cropping, or color normalization, which help to improve the quality and consistency of the image.

Feature Selection**:** Once the image is preprocessed, the next step is to select the most relevant features that can help to distinguish between different objects or classes. This can be done manually, by identifying important visual cues or properties that are specific to the problem at hand, or automatically, using machine learning techniques such as principal component analysis (PCA) or linear discriminant analysis (LDA).

Feature Representation: After the relevant features have been selected, the next step is to represent them in a suitable format that can be used for analysis. This can involve transforming the raw image data into a more compact or meaningful representation, such as a histogram of color values, a texture descriptor, or a shape model.

Feature Extraction: Once the features have been selected and represented, the final step is to extract them from the image. This can involve applying a set of operations or filters to the image data, such as convolution or wavelet transforms, to identify and extract the relevant features.

Feature Normalization: In some cases, it may also be necessary to normalize the extracted features to improve their consistency and robustness. This can involve techniques such as z-score normalization or min-max normalization, which help to ensure that the features are scaled and centered appropriately.

Image Segmentation:Image segmentation is a process of dividing an image into multiple segments or regions, each of which corresponds to a different object or part of the image. Image segmentation plays a critical role in computer vision

applications, such as object detection, tracking, and recognition. The goal of image segmentation is to partition an image into meaningful regions that can be analyzed and processed independently. This is typically achieved by applying a set of image processing techniques to identify regions that share similar visual characteristics, such as color, texture, or intensity.

There are several techniques for image segmentation, including:

Thresholding: This technique involves selecting a threshold value and partitioning the image into regions based on the intensity values of each pixel. Pixels with intensity values above the threshold are assigned to one region, while pixels with intensity values below the threshold are assigned to another region.

Edge Detection: This technique involves detecting the edges or boundaries between different regions in an image. This can be done using techniques such as the Canny edge detector or the Sobel edge detector.

Region Growing: This technique involves starting with a seed pixel and iteratively adding neighboring pixels to the region based on some similarity criterion. This process continues until all pixels in the region have been added.

Clustering: This technique involves grouping pixels into clusters based on their visual similarity. This can be done using algorithms such as k-means clustering or hierarchical clustering.

Classification: In a crowd detection system, classification of images typically involves identifying and categorizing different objects or groups within the crowd. This process can be performed using various techniques, including traditional computer vision methods and deep learning-based approaches. In traditional computer vision methods, feature extraction algorithms are used to identify and extract relevant features from the images, such as edges, corners, and texture patterns. These features are then used to train a classifier, such as a support vector machine (SVM) or k-nearest neighbor (KNN), to recognize different objects or groups within the crowd, such as individuals or clusters. In deep learning-based approaches, Convolutional Neural Networks (CNNs) are commonly used for image classification tasks. CNNs are a type of neural network that are specifically designed for image processing tasks, as they can automatically learn and extract relevant features from images by using multiple layers of convolutional filters.

The process of image classification in a crowd detection system using CNNs typically involves the following steps:

Data Preparation: Collecting and preparing a large dataset of labeled images for training the model.

Training the Model: Using the prepared dataset to train the CNN model to recognize different objects or groups within the crowd.

Validation: Testing the trained model on a separate dataset of images to evaluate its accuracy and performance.

Deployment: Using the trained model to classify new images in real-time.

During the training process, the CNN model learns to recognize patterns and features in the crowd images by adjusting the weights of the neural network connections based on the errors between the predicted and actual class labels. Once the model is trained and validated, it can be used to classify new crowd images by feeding them into the model and predicting the corresponding class labels based on the learned features and patterns.

6.RESULTS
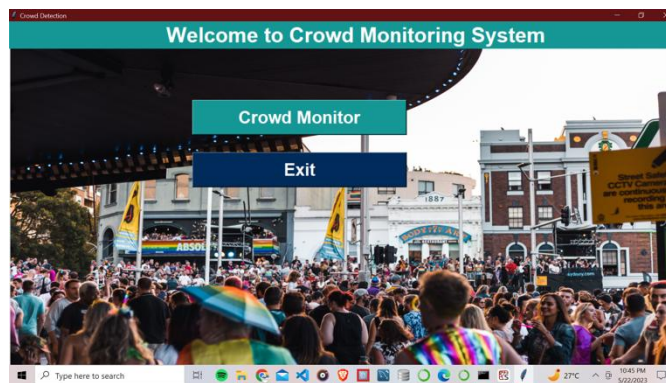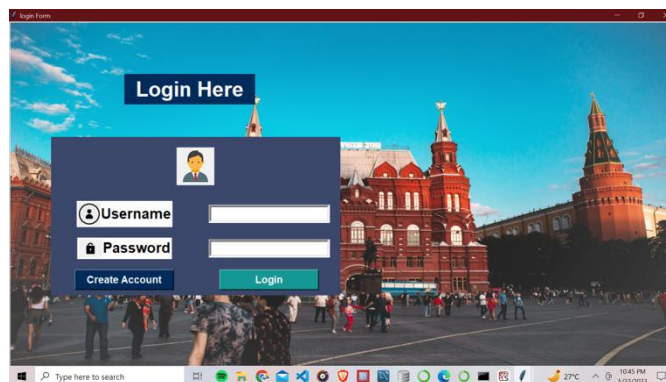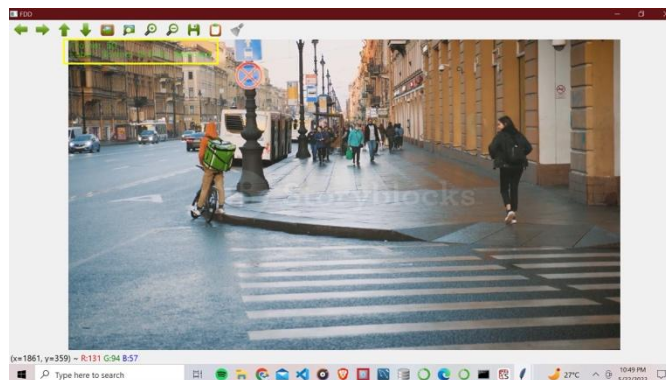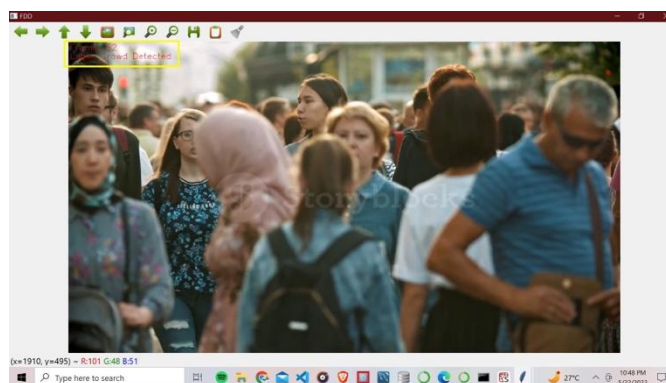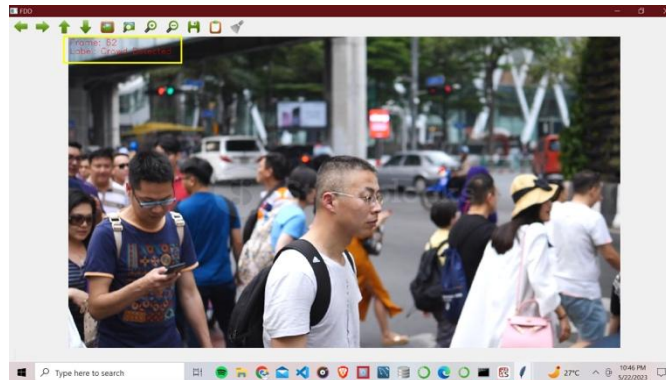
TEST CASES

Test Case 1 -

Test Case ID	Test Case	Test Case I/P	Actual Result	Expected Result	Test case criteria(P/F)
001	Enter The Wrong username or password click on submit button	Username or password	Error comes	Error Should come	P
002	Enter the correct username and password click on submit button	Username and password	Accept	Accept	P

Test Case 2 –

Test Case ID	Test Case	Test Case I/P	Actual_Result	Expected Result	Test case criteria(P/F)
001	Enter the number in username, middle name, last name field	Number	Error Comes	Error Should Comes	P
001	Enter the character in username, middle name, last name field	Character	Accept	Accept	p
002	Enter the invalid email id format in email id field	Kkgmail.com	Error comes	Error Should Comes	P
002	Enter the valid email id format in email id field	kk@gmail.com	Accept	Accept	P
003	Enter the invalid digit_no in phone no field	99999	Error comes	Error Should Comes	P
003	Enter the 10 digit no in phone no field	9999999999	Accept	Accept	P

Test Case 3 –

Test Case ID	Test Case	Test Case I/P	Actual Result	Expected Result	Test case criteria(P/F)
001	Store csv File	csv_file	csv file store	Error Should come	P
002	Parse the csv file for conversion	parsing	File get parse	Accept	P
003	Attribute identification	Check individual Attribute	Identify Attributes	Accepted	P
004	Weight Analysis	Check Weight	Analyze Weight of individual Attribute	Accepted	P
005	Tree formation	Form them-Tree	Formation	Accepted	P
006	Cluster Evaluation	Check Evaluation	Should check Cluster	Accepted	P
007	Algorithm Performance	Check Evaluation	Should work Algorithm Properly	Accepted	P
008	Query Formation	Check Query Correction	Should check Query	Accepted	P

7. CONCLUSION

In conclusion, the proposed CNN-based method provides a promising approach for crowd detection and counting in still images from various scenes. The use of features derived from a CNN model trained for other computer vision tasks enables accurate representation of crowd density. The neighboring local counts are also strongly correlated, and feature extraction techniques contribute to good detection accuracy. The system uses the RESNET deep convolutional network, which provides up to 152 hidden layers, and can be extended with multiple convolutional layers with an ensemble deep learning model for even higher accuracy. Experimental findings demonstrate that the proposed method outperforms other recent related methods and can be extended to work on image and video datasets. Overall, this system offers better accuracy for crowd detection from heterogeneous images, making it a valuable tool for various applications.

REFERENCES

[1]Crowd Detection And Tracking In Surveillance Video Sequences, Sohail Salim; Othman O Khalifa; Farah Abdul Rahman; AdidahLajis 2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)

[2] Multi-UAV Based Crowd Monitoring System Rodrigo Saar de Moraes; Edison Pignaton de Freitas IEEE Transactions on Aerospace and Electronic Systems

[3] In-Depth Survey to Detect, Monitor and Manage Crowd Ali M. Al-Shaery; Shroug S. Alshehri; Norah S. Farooqi; Mohamed O. Khozium IEEE Access

[4] Analysis and Design of Public Places Crowd Stampede Early-Warning Simulating System Shangnan Liu; Zhenjiang Zhu; Qiang Cheng; Hao Zhang 2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)

[5] Crowd Monitoring Using Mobile Phones Yaoxuan Yuan 2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics

[6] A Real-Time Crowd Detection and Monitoring System using Machine Learning Pooja Shrivastav; Vakula Rani J 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)

[7] Crowd detection and management using cascade classifier on ARMv8 and OpenCV Python S. Syed Ameer Abbas; P. Oliver Jayaprakash; M. Anitha; X. VinithaJaini 2017 International Conference on Innovations Information, Embedded and Communication Systems (ICIIECS)

[8] Multi-Person Tracking in Smart Surveillance System for Crowd Counting and Normal/Abnormal Events Detection Ahsan Shehzed; Ahmad Jalal; Kibum Kim 2019 International Conference on Applied and Engineering Mathematics (ICAEM)

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462　6381 907 438　ijircce@gmail.com

Scan to save the contact details