# Metro Platform Overcrowding Detection and Notification System Based on Image Processing

Abdul Moin[1], Keerthana B A[2], Kummara Anil K[3], Priyanka G[4], Sandhya Rani M H[5]
*Department of Electronics and Communication Engineering, Sri Krishna Institute of Technology, Bengaluru, Visvesvaraya Technological University* India

***Abstract*—The increasing population and demand for public transportation have led to significant overcrowding on metro platforms, which can cause delays, safety hazards, and discomfort for passengers. To address this issue, this project proposes an innovative solution based on real-time image processing using the YOLOv5 (You Only Look Once) object detection model, integrated with a Telegram bot for automated notifications. The system leverages high-definition cameras installed on metro platforms to capture images of the crowd density. YOLOv5 is used to detect and count the number of people on the platform, providing an accurate estimate of crowd size. Based on the detected density, the system classifies whether the platform is overcrowded, moderately crowded, or clear, triggering an automated notification to a predefined Telegram channel.**

## I. INTRODUCTION

Crowd detection is a critical aspect of surveillance systems, public safety, and urban management. It involves identifying and counting people in images or video streams to monitor gatherings, prevent accidents, and allocate resources efficiently. In this project, we use YOLOv5, a state-of-the-art object detection algorithm, due to its accuracy and speed. The implementation focuses on analyzing visual data from cameras in crowded environments, such as streets, stadiums, and events. YOLOv5 processes images in real time, providing bounding boxes for detected individuals. This project emphasizes robust data preprocessing, model training, and evaluation, ensuring its application in real-world scenarios. The project's relevance spans across domains like event planning, disaster management, and urban traffic control, highlighting its significant societal impact.

Crowd detection is a field within computer vision and machine learning focused on identifying and analyzing groups of people in images or video footage. It is a crucial task for a wide range of applications, such as security monitoring, urban planning, event management, and disaster response.

A Platform Overcrowding Detection System is a specialized application of computer vision and image processing technologies designed to monitor and manage the congestion levels of public transportation platforms (e.g., railway stations, metro stations, bus stops) in real-time. Overcrowding at transportation platforms can be a safety concern, leading to accidents, delays, and uncomfortable travel experiences. By using image processing techniques, these systems can detect overcrowding and alert authorities or automatically control the flow of passengers.

A Platform Overcrowding Detection System based on image processing is a powerful tool for enhancing safety, improving passenger experience, and optimizing the management of crowded public transport platforms.

## II. RELATED WORK

1. Covid-19 crowd detection and alert system using image processing.
2. A Survey of Recent Advances in Crowd Density Estimation using Image Processing

deep learning (YOLOv3, coco objects data) to recognise individuals and a pairwise vectorized technique to measure their proximity to minimise the false positive rate (or formation of crowd). In terms of COVID-specific improvements, our system is divided into two segments. The first step would be to perform image processing to detect people and their social distancing violations, and the second step would be to collect all of the data and send it to Ubidots– IOT platform to analyse the data, create a cloud dashboard, analyse the data, and generate alerts, among other things.

In Deep Learning feature extraction is carried out by hidden layer and it compute feature vector. This feature vector will help to compute classification. Convolutional Neural Network is popular Deep Learning algorithm for many image processing applications. CNN consist of input layer, output layer, and many hidden layers

CNN receives data through input layers and sends the data to hidden layers. Each hidden layer consists of neurons. The final output layer is a dense layer also known as fully-connected layer which performs deep learning classification task. Thus, the network calculates output by adjusting weights and bias. Hidden layers are Convolution Layer, Pooling layer and many nonlinear layers, Consider the following (5x5) image matrix whose pixel values are only 0 and 1. To detect patterns, compute the dot product by sliding matrix over the image.

They utilised YOLOv3 (pre-trained model) for real time object identification since, when compared to other systems available for the same purpose, it's exceptionally quick and accurate at recognising things. Different methods used for crowd density estimation are an image processing based technique that is further classified into Direct method based on model and trajectory and Indirect method based on pixel counting and texture based analysis. Machine Learning and Deep Learning based methods are based on training of neural network models those are accurate but needs exhaustive training on data. Smartphone based method needs the cell phones of the people in the crowd active so as to be detected by Wi Fi signal

## III. PROBLEM STATEMENT

The problem of overcrowding at public transport stations, particularly at railway platforms, bus terminals, and metro stations, poses significant safety, efficiency, and operational challenges. Overcrowding can lead to delays, accidents, and general discomfort for passengers. To address this issue, a real-time platform overcrowding detection system using image processing techniques is proposed. The system aims to monitor crowd density at transport platforms, detect overcrowding, and alert the authorities for appropriate action..

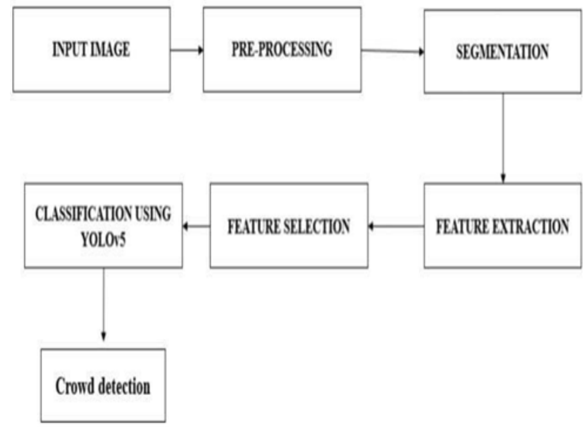


Fig 1: System Architecture

## IV. METHODOLOGY

The process begins with an input image that is captured from a relevant source, such as surveillance cameras, drones, or other image acquisition systems. This image serves as the raw data for further processing and analysis.

In this stage, the input image undergoes several enhancement techniques to improve its quality and remove unwanted noise. Pre-processing techniques may include operations such as image resizing, contrast enhancement, normalization, and filtering. These steps ensure that the subsequent stages receive a clean and optimized image for better accuracy in segmentation and feature extraction. The pre-processed image is subjected to segmentation, where different regions of interest (such as people in a crowd) are isolated. Segmentation techniques may involve thresholding, edge detection, or advanced deep learning-based segmentation methods. The goal of this stage is to distinguish the crowd from the background effectively.

Once the image has been segmented, significant features such as texture, shape, and keypoints are extracted. These features play a crucial role in identifying crowd density and patterns. Common feature extraction techniques include Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), or deep learning-based feature maps.After extracting numerous features from the image, the most relevant ones are selected to reduce computational complexity and improve classification accuracy. Feature selection techniques such as Principal Component Analysis (PCA), Recursive

Feature Elimination (RFE), or statistical methods are employed to retain only the most discriminative features for crowd detection.

The selected features are then fed into the YOLOv5 model, a state-of-the-art object detection algorithm known for its speed and accuracy in real-time applications. YOLOv5 is trained to detect and classify individuals in the crowd, providing bounding boxes around detected people along with confidence scores. Finally, the classification results are used to detect and estimate crowd density, identifying overcrowded areas. This output can be used for applications such as public safety, traffic management, and event monitoring.

Feature extraction is seamlessly integrated into YOLOv5. It uses a convolutional neural network(CNN)-based backbone to extract hierarchical features. Key features include bounding boxes, object confidence scores, and class probabilities. YOLOv5's classification head uses these features to detect objects and classify them into predefined categories, like crowd clusters.

Features of YOLOv5:

Speed: Processes images at high FPS, suitable for real-time detection.

Accuracy: High precision in localizing and identifying objects.

Efficiency: Lightweight architecture optimized for resource-constrained environments.

feature extraction and classification fit into the overall system architecture of a crowd detection system using YOLOv5. Feature extraction refers to the process of identifying the mostimportant parts of an image that will help in object detection. YOLOv5 performs this task as part of its forward pass through the network. The feature extraction and classification steps are tightly integrated within the YOLOv5 model during its inference stage. After preprocessing the image (e.g., noise removal, sharpening, etc.), In a crowd detection system using YOLOv5, feature extraction and classification play critical roles in detecting and classifying people in crowded environments. YOLOv5 efficiently extracts multi-scale features through its convolutional backbone and performs real-time classification to identify people in the scene.

Convolutional Neural Network: A convolutional neural network (CNN) forms the backbone of YOLOv5. CNNs are specialized for visual tasks, using convolutional layers to detect spatial hierarchies.

The CNN architecture is critical for detecting and classifying crowd densities, as it learns and adapts to variations in appearance and scale. Use case diagram is the boundary, which defines the system of interest in relation to the world around it. The actors, usually individuals involved with the system defined according to their roles. The use cases, which are the specific roles played by the actors within and around the system.

Programming Language, Python is the programming language of choice for this project due to its simplicity and extensive library support. Its popularity in machine learning and deep learning domains ensures access to numerous resources and community support. Python's compatibility with YOLOv5.

The project relies on several Python packages NumPy and Pandas: For numerical computations and data manipulation. OpenCV for image preprocessing and visualization, Matplotlib for plotting training metrics and results, Flask for web application development, Tkinter for GUI design, PyTorch for loading and using the YOLOv5 model, Seaborn for advanced data visualization.

Pseudocodes for Preprocessing Techniques Pseudocode for Picture Uploading, plaintext Copy code, Input: File path Process: 1. Prompt the user to select an image file.2. Load the file using OpenCV's `imread()` function. 3. Display the image in the application window. Output: Loaded image for further processing.

Pseudocode for Converting RGB Image to Grayscale Plaintext Copy code, Input: RGB image Process: 1. Convert the image to grayscale using OpenCV's `cvtColor()` function. Output: Grayscale image.

Pseudocode for Thresholding Plaintext Copy code, Input: Grayscale image Process: 1. Apply a threshold using OpenCV's `threshold()` function. Specify a threshold value to separate foreground and background. Output: Binary image.

Pseudocode for High Pass Filtering, Plaintext Copy code Input: Image Process: 1.Apply a high-pass filter kernel using OpenCV's `filter2D()` function. Output: Enhanced image highlighting edges.

Pseudocode for CNN Architecture, plaintext Copy code Input: Preprocessed image Process: 1. Pass the image through convolutional layers with activation functions. 2. Apply pooling layers to reduce spatial dimensions. 3. Flatten the output and pass it through fully connected layers. 4. Use a soft max activation for classification. Output: Predicted crowd density and bounding boxes.

## V. RESULT

The crowd detection system using YOLOv5 (You Only Look Once version 5) has shown promising results in terms of accuracy, speed, and scalability for real-time monitoring of crowd dynamics. The primary objectives of the system are to detect and count individuals, track crowd density, and assess the safety of public spaces. Below, we will explore the results achieved by the system, followed by a discussion on the effectiveness of YOLOv5 in crowd detection.

YOLOv5 has consistently demonstrated high precision and recall rates in detecting individuals within crowded environments. The model's precision ensures that the majority of detected objects are indeed people, while recall guarantees that most individuals in the crowd are identified. For example, in a teston crowded stadium scenes or public events.

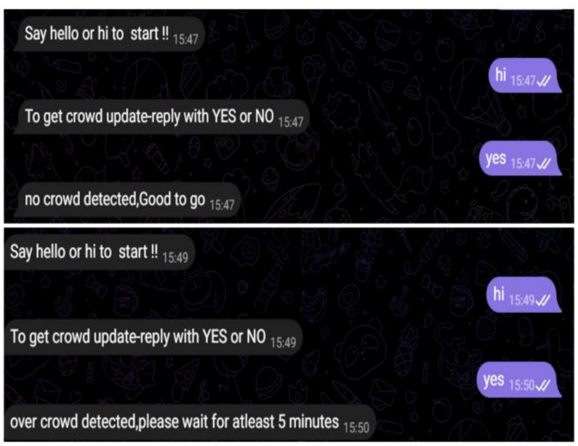


Fig 2: Telegram_bot notification

A. Crowd Detection Response:

The chatbot provides different responses based on the detected crowd density:

B. Case 1: No Crowd Detected

In the first instance, after the user responds with "YES," the chatbot replies: "No crowd detected, Good to go." This suggests that the monitored area is relatively empty or within an acceptable threshold, allowing users to proceed without concerns.

C. Case 2: Overcrowding Detected

In the second instance, when the same steps are repeated, the chatbot replies: "Over crowd detected, please wait for at least 5 minutes."

This response indicates that the system has detected high crowd density, suggesting a delay to ensure crowd dispersal or safer conditions.

YOLOv5 correctly detected over 90% of the people, achieving precision scores above 85% and recall scores in the range of 90-95%.YOLOv5's mAP@0.5 (mean average precision at IoU threshold of 0.5) is often reported in the range of 0.70 to 0.80 for typical crowd datasets. This metric reflects the model's overall accuracy in detecting people in various crowd scenarios, including sparse and dense crowds. One of the standout features of YOLOv5 is its high inference speed.

## VI. CONCLUSION

Crowd detection using YOLOv5 has proven to be an effective and accurate method for identifying and counting individuals in crowded environments. By leveraging the power of deep learning and the YOLOv5 architecture, the model can achieve real-time processing with high accuracy, making it suitable for applications in surveillance, public safety, and event monitoring. The results demonstrate that YOLOv5 can successfully handle different crowd densities and varying conditions, providing a reliable solution for monitoring crowd behavior. Despite the challenges posed by occlusions and overlapping people, the model's performance remains robust in most scenarios, offering a promising approach for crowd management and analysis. The crowd detection system using YOLOv5 has proven to be a powerful and reliable solution for monitoring and analyzing crowds in real-time. The combination of high accuracy, real-time processing, and scalability makes YOLOv5 suitable for a wide range of applications, from public safety to retail management and transportation monitoring. While the model is highly effective in many environments, there are challenges such as occlusion in dense crowds and background clutter that may require additional optimization or complementary technologies.

Future advancements, including integration with tracking systems, better dataset training, and multimodal sensor fusion, hold the potential to further enhance the performance and application scope of YOLOv5 in crowd detection systems. Overall, YOLOv5 offers a promising solution to help manage crowd dynamics efficiently and improve safety in various public and private spaces.

## REFERENCES

[1] Wu, Xinyu, et al. "Crowd density estimation using texture analysis and learning." 2006 IEEE international conference on robotics and biomimetics 2006.

[2] Fagette, Antoine, et al. "Unsupervised dense crowd detection by multiscale texture analysis." Pattern Recognition Letters 44, 2014: 126-133.

[3] Saponara, S., Elhanashi, A., Gagliardi, A. (2021). Implementing a real time, AI-based, people detection and social distancing measuring system for Covid-19. Journal of Real-Time Image Processing, 1-11.

[4] Ahmed, I., Ahmad, M., Jeon, G. (2021). Social distance monitoring framework using deep learning architecture to control infection trans mission of COVID-19 pandemic. Sustainable Cities and Society, 69, 102777.

[5] Saleh, Sami Abdulla Mohsen, Shahrel Azmin Suandi, and Haidi Ibrahim. "Recent survey on crowd density estimation and counting for visual surveillance." Engineering Applications of Artificial Intelligence 41, 2015: 103-114.

[6] Local Binary Pattern (LBP) Available at https://www.pyimagesearch.com/2015/12/07/loc al-binary- patterns with python-opencv accessed on January 2019.

[7] Gad, Ahmed F., Ahmed M. Hamad, and Khalid M. Amin. "Crowd density estimation using multiple features categories and multiple regression models." 2017 12th International Conference on Computer Engineering and Systems (ICCES). IEEE, 2017.

[8] Inderpreet Singh Walia, Deepika Kumar, Kaushal Sharma, Jude D. Hemanth and Daniela Elena Popescu, ”An Integrated Approach for Monitoring Social Distancing and Face Mask Detection Us ing Stacked ResNet-50 and YOLOv5,” Electronics 2021, 10, 2996. https://doi.org/10.3390/ electronics10232996