CAIT-UTC-REG 26

Passenger Flow Modeling and Simulation in Transit Stations

FINAL REPORT
February, 2022

Submitted by:

Xiang Liu, Ph.D.
Associate Professor
Rutgers, The State University
Department of Civil & Environmental Eng.
500 Bartholomew Road
Piscataway, NJ, 08854

Yadi Zhu, Ph.D.
Postdoctoral Research associate
Rutgers, The State University
CAIT
100 Brett Road
Piscataway, NJ 0885

External Project Manager
Brad Mason, Director, Capital Resiliency and Continuity
NJ Transit
One Penn Plaza East
Newark, NJ 07105

In cooperation with

Rutgers, The State University of New Jersey
And
New Jersey Transit
And
U.S. Department of Transportation
Federal Highway Administration

Disclaimer Statement

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

The Center for Advanced Infrastructure and Transportation (CAIT) is a Regional UTC Consortium led by Rutgers, The State University. Members of the consortium are Atlantic Cape Community College, Columbia University, Cornell University, New Jersey Institute of Technology, Polytechnic University of Puerto Rico, Princeton University, Rowan University, SUNY - Farmingdale State College, and SUNY - University at Buffalo. The Center is funded by the U.S. Department of Transportation.

1. Report No.
CAIT-UTC-REG 26

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Passenger Flow Modeling and Simulation in Transit Stations

5. Report Date
February, 2022

6. Performing Organization Code
CAIT/Rutgers University

7. Author(s)
Xiang Liu https://orcid.org/0000-0002-4348-7432
Yadi Zhu https://orcid.org/0000-0003-4906-5916

8. Performing Organization Report No.
CAIT-UTC-REG 26

9. Performing Organization Name and Address
Center for Advanced Infrastructure and Transportation
Rutgers, The State University of New Jersey
100 Brett Road, Piscataway, NJ 08854

10. Work Unit No.

11. Contract or Grant No.
69A3551847102

12. Sponsoring Agency Name and Address
center for Advanced Infrastructure and Transportation
Rutgers, The State University of New Jersey
100 Brett Road, Piscataway, NJ 08854

13. Type of Report and Period Covered
Final Report
11/01/2019-10/31/2020

14. Sponsoring Agency Code

15. Supplementary Notes
U.S. Department of Transportation/OST-R
1200 New Jersey Avenue, SE
Washington, DC 20590-0001

16. Abstract

Crowd analysis and management is key area of study for transit agencies to maximize operational efficiency and minimize risk. Passenger flow volume, crowd density and walking speed are key features of crowd analytics. This study proposes a generalized Artificial Intelligence (AI)-based crowd analytics model framework for rail transit stations, by analyzing high-density crowd video data. We propose a generalized triple-layer AI-aided methodological framework (named AI-Crowd) for calculating the flow volume, crowd density and walking speed. For the pedestrian detection and tracking layer, the You Only Look Once (YOLO) and Simple Online and Realtime Tracking with a Deep Association Metric (Deep SORT) are integrated to detect and track the dynamic position of each individual. Then, we adjust the original image from the camera to real scale in the camera calibration layer. Next, the calibrated results used by the metric calculation layer to implement a comprehensive calculation of crowd metrics. Several video records from stair and hallway scenarios in a major rail transit station in China are used to validate the model framework. Based on the example video data samples, the pedestrian counting accuracy can be 95% - 98%; the fundamental diagrams of density-speed are shown to be consistent with empirical studies. Furthermore, the methodology has practical applications such as automatic passenger counting, level of service evaluation, and social distancing monitoring in the era of COVID-19.

17. Key Words
Transit, Station, Passenger Flow, Simulation, Modeling

18. Distribution Statement

19. Security Classification (of this report)
Unclassified

20. Security Classification (of this page)
Unclassified

21. No. of Pages
Total #38

22. Price

Acknowledgments
This project was financially supported by the Center for Advanced Infrastructure and Transportation (CAIT), a University Transportation Center (UTC) supported by USDOT-OST-R. We are grateful to the anonymous rail agency for providing their infrastructure asset data for this research.

Contents

1. Introduction

Transit providers must understand crowd dynamics to meet passenger demand efficiently and safely (Reuter, 2003). For example, flow volume in stations can be correlated to system capacity, which is an essential metric for transit agencies to evaluate the quality of service and make critical scheduling decisions (Sørensen et al., 2019). Crowd density is another essential metric for evaluating transit service and safety (Helbing et al., 2015), and allows transit agencies to monitor the safety of their passengers (Song et al., 2019). Individual movement trajectories and walking speeds are also important for analyzing passenger behavior, which provides insight into facility design to improve customer experience (Ye et al., 2008).

To acquire critical crowd data, various approaches and solutions have been used (Table 1). Many passenger counting technologies have been developed to obtain flow volume. According to a survey from over 50 city metro authorities, commuter railroads, and surface transport providers around the world (Reuter, 2003), manual methods (e.g., staff counting, manual estimation based on train arrivals/departures) still dominate the practice and are rated as most useful at disaggregate data analysis on individual level (Boyle, 1998). However, acquiring and analyzing these data are labor-intensive and time-consuming. Automatic fare collection devices (AFCs) (Pinna et al., 2010), have been widely used in some countries (e.g., China). Nevertheless, card penetration is crucial to obtain accurate ridership. Field surveys (Song et al., 2019) or manually counting people at an area under observation from closed-circuit television (CCTV) records (Saleh et al., 2015) are commonly used manual methods for acquiring the crowd density. For analyzing the passenger movement, like walking speed, field surveys with stopwatches or manually review of walking length and duration from video records (Ye et al., 2008) are the commonly employed manual methods. Similarly, these methods are also labor intensive and time consuming.

Computer vision-based techniques show promise at overcoming the limitations of manual methods with low costs and high efficiency and receive much attention from academia and industry for crowd analytics (Junior et al., 2010; Saleh et al., 2015; Sindagi and Patel, 2018).

Table 1.1. Approaches Comparison Matrix for the Crowd Dynamic Metrics

Metrics	Approaches	Advantages	Limitations
Flow volume	Manual methods (Boyle, 1998; Reuter, 2003)	Easy to implement without any professional devices	Labor-intensive, time-consuming; reliability is impacted by the checkers
	Electronic registering fareboxes (Pinna et al., 2010; Reuter, 2003)	Aggregate data on route-wide or system-wide level	Reliability is impacted by operator compliance and attitudes

	Automatic fare collection devices (Pinna et al., 2010; Reuter, 2003)	Individual level data, easy to operate for passengers and operators	Card penetration impacts the accuracy
	Infrared sensors (Pinna et al., 2010; Reuter, 2003)	Capability of managing fast and compact flows	Incorrect activations by luggage, multiple sensors on each door
	Treadle mats (Pinna et al., 2010; Reuter, 2003)	Less issues of incorrect activations, high accuracy	Need of slow passenger flows
	Computer vision-based methods (Junior et al., 2010)	Less issues of incorrect activations, high accuracy, capability of managing fast and compact flows	Robustness in multi-scenarios, large storage and computational capacity requirement
Crowd density	Manual methods (Saleh et al., 2015; Song et al., 2019)	Easy to implement without any professional devices	Labor-intensive, time-consuming
	Computer vision-based methods (Ye et al., 2008)	Low cost, high efficiency, visual display	Robustness in multi-scenarios, large storage and computational capacity requirement
Walking behavior	Manual methods (Ye et al., 2008)	Easy to implement without any professional devices	Labor-intensive, time-consuming
	Computer vision-based methods (Ye et al., 2008)	Low cost, high efficiency, visual display	Robustness in multi-scenarios, large storage and computational capacity requirement

However, existing studies mainly focus on counting passengers in a static image to estimate crowd density and cannot acquire flow volume, monitor crowd density or analyze passenger movement. Meanwhile, most commercially available CV-based products focus on counting, not estimating crowd density monitoring or analyzing walking behavior. Moreover, these products are targeted to scenarios of low-density and small-flow-volume crowds such as shopping malls and office buildings, which contrasts with the typical environment of a high-density rail transit station.

These studies and products mainly utilize video data from cameras with high-angle views for counting people and crowd density estimation (Li et al., 2016; Punn and Agarwal, 2019). However, the cameras in transit stations are more commonly top-view fisheye lens cameras, to accommodate for low overhead clearances. Top-view cameras capture more extensive images with line distortion than normal lens cameras, as shown in Figure 1.1. An added benefit of these cameras is that they do not capture facial information, which addresses privacy concerns of big video data collection. These specific features make the analysis of video data from top-view fisheye lens cameras more difficult from that from normal lens cameras.

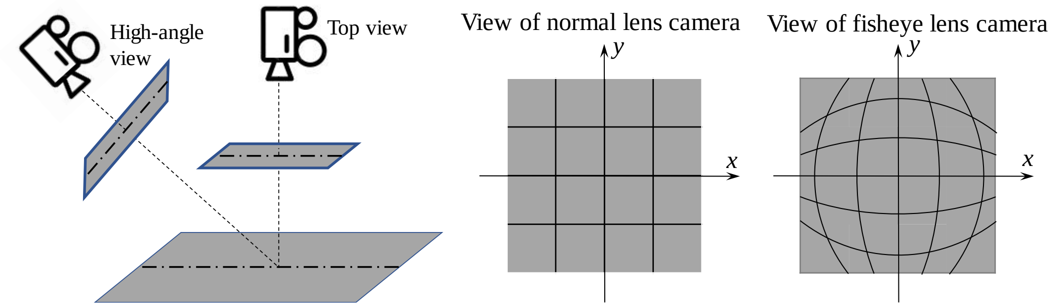


Figure 1.1. Illustration of Top View and High-angle View, View of Normal Lens Camera and View of Fisheye Lens Camera

From the above, there are three challenges to develop a practicable, comprehensive CV-based high-density crowd analytics model framework for the videos from top-view fisheye lens cameras. This study aims to provide implementable and generalized solutions for these challenges, as shown in Table 1.2.

Table 1.2. Challenges and Solutions for AI-aided Crowd Analytics in Rail Transit Station

	Challenges	Solutions
I	How to acquire individual dynamic trajectory information from video records of high-density crowds in rail transit station?	We propose a head detection + tracking model framework to get trajectories.
II	How to measure the real distance from video frames, especially for the distorted images?	We formulate a camera calibration method with extrinsic parameters and intrinsic parameters extracted from the video frames.
III	How to calculate three essential metrics of crowd analytics using the trajectories from the proposed model framework?	We develop the calculation methods of three metrics (i.e., flow volume, crowd density, walking speed) with individual trajectory.

The remainder of this report is organized as follows. Chapter 2 shows a comprehensive literature review on pedestrian counting and crowd density calculation using computer vision techniques, as well as studies on top-view fisheye lens video data analysis. Chapter 3 specifies the methodology proposed in this study. Chapter 4 describes the model implementation and validation using video data from Beijing Subway stations. Chapter 5 discusses the potential practical applications of the proposed model framework. Chapter 6 concludes this report and proposes future improvements

2. Literature Review

2.1. Crowd Analytics with Computer Vision

Flow low volume, crowd density and walking speed are the three basic metrics for crowd analysis, and their relationships are defined as fundamental diagrams. These diagrams are used to estimate capacity and Level of Service (LOS) of various facilities in transit stations. With the improvement of computer vision techniques, they can be acquired by multiple methods, including computer vision and convolutional neural network (CNN)-based methods.

The computer vision based methods are categorized into two categories, (1) regression-based methods, (2) detection-based methods (Junior et al., 2010; Saleh et al., 2015; Sindagi and Patel, 2018). Regression-based methods are proposed to estimate the number of people for extremely dense crowds in the images (Sindagi and Patel, 2018). These methods formulate the relationship between the image features (e.g., blob area) of people and crowd density using regression models. This method is mainly used to estimate total crowd density in an image but cannot be used to segment each individual. In the methods, features contain low-level features (e.g., blob area, perimeter-area ratio) (Ryan et al., 2009) and texture features (e.g., contrast, homogeneity, entropy) (Chan et al., 2008), which are also commonly used in detection-based methods.

Detection-based methods count the number of people by detecting each person in an image and calculate the crowd density using counted people number with region area (Li et al., 2016). Monolithic detection is a commonly used technique, which identifies the crowd with extracted features from a whole-body (Saleh et al., 2015; Sindagi and Patel, 2018). For example, binary classifier feature extraction methods for a scanning window detector shows a detection rate of about 93% in typical surveillance scenarios (Jones and Snow, 2008). However, occlusion in high-density crowds adversely impacts the performance of whole-body detection models. Three head-like detection methods are proposed to tackle this challenge; Haar wavelet transform for feature extraction of the head-like contour (Sheng-Fuu et al., 2001), omega shape (Ω) feature for the head-shoulder part (Li et al., 2009), and 3D shape model using three ellipsoids (Zhao et al., 2008). These methods excel in high density environments as they are robust to partial occlusions and atypical part appearances.

CNN is a state-of-the-art machine learning method and has superior capabilities for learning non-linear functions from the input data (Zhao et al., 2019). A CNN was used to formulate a deep CNN regression model for estimating people's numbers from images by determining the relationship between image features and the number of people in the image (Wang et al., 2015).

In addition, studies have used CNNs for feature extraction (Sheng et al., 2018), which demonstrated CNNs suitability for pedestrian detection. R-CNN is one of these features based CNNs with superior performance (Girshick et al., 2014) that extracts a large number of region proposals from an input image and computes features for each proposal using CNN. Objects are detected by classifying each region with features. Following R-CNN, Fast-RCNN (Girshick, 2015) and Faster-RCNN (Ren et al., 2017) were developed to improve the computational efficiency and model performance. You Only Look Once (YOLO) is a recent development in CNN detectors that isolates objects using CNN in a single analysis of an image. This methodology has greatly improved computational efficiency while maintaining a high detection accuracy (Redmon et al., 2016; Redmon and Farhadi, 2018). Although the detection accuracy of YOLO is lower than that of Faster-RCNN, the superior computational efficiency makes YOLO a promising solution for the application in practice.

Most previous studies focus on pedestrian detection in an image, static counting, and density estimation but do not implement trajectory-based analyses. Some studies have attempted to overcome this challenge through methods such as the Kanade-Lucas-Tomasi (KLT) tracker, which is a commonly used tracking methodology (Rabaud and Belongie, 2006; Sidla et al., 2006). As an optical flow-based method, KLT tracks people using displacements of the dominant points and acquires people’s walking trajectories. In these studies, a “virtual gate” was defined to count people with trajectories (Sidla et al., 2006), and the trajectories can also be employed to calculate walking speed (Hediyeh et al., 2014a; Sultan and Khan, 2013). However, these methods are susceptible to occlusion or illumination changes. Recently, a simple online real-time tracking with a deep association metric (or Deep SORT) (Wojke et al., 2017) was developed. In Deep SORT, persons are detected in each frame, and the detections are matched based on feature similarity to acquire their tracks. Deep SORT solves occlusion tracking problems effectively and is more suitable for high-density tracking as in the context of rail transit station.

2.2. Top-view Fisheye Lens Video Analysis

Fisheye lens cameras capture larger areas than conventional cameras. They are widely used in security surveillance system (Tai et al., 2018; Wang et al., 2019). Furthermore, top-view fisheye lens cameras benefit from less mutual occlusion among objects in a scene and protecting individual privacy without facial information (Krams and Kiryati, 2017). However, the lenses capture wide visual fields by distorting the lines in images, which makes the pedestrian detection and tracking challenging. Most studies using videos from top-view fisheye lens cameras focus on pedestrian detection, while studies on pedestrian tracking are scarce.

A significant difference between conventional camera analysis and fisheye lens camera analysis is the camera calibration that extracts camera parameters from the video images. Except for extrinsic and intrinsic camera parameters, distortion parameters are the crucial parameters for fisheye lens camera. For acquiring the distortion parameters, a geometric-mathematical model for describing the physical and optical behavior of the sensor is essential (Puig et al., 2011). To extract parameters quickly and automatically, generalized parametric fisheye lens models were proposed using a polynomial form with distortion parameters (Kannala and Brandt, 2006; Scaramuzza et al., 2006). These models calculate extrinsic, intrinsic parameters and distortion parameters using several checkerboard images captured by the cameras. Because of the simple expression and robust performance, these models are widely used to calibrate fisheye lens cameras.

With the calibrated parameters, undistorting the deformed video frames as well as converting image scale to real-world scale need to be conducted for further analysis. For pedestrian detection, there are two different approaches (Wang et al., 2019): the first is detecting people directly on distorted video frames and then undistorting images (Tamura et al., 2019; Wang et al., 2019). For example, blobs extracted by background subtraction were used to detect people; undistorted images using intrinsic and extrinsic camera parameters were employed to track people (Meinel et al., 2014). Images of the COCO dataset were rotated to train the YOLO model for detecting the rotated people (Tamura et al., 2019). Mask-RCNN was utilized to detect the deformed shape of people, and tacking was implemented based on the detected masks (Wang et al., 2019). The second one is undistorting fisheye video frames into normal images and then detecting people in them (Seidel et al., 2018). For example, projecting distorted images to normal images was implemented by the camera calibration model, and YOLO was used to detected people in research by Seidel et al. (Seidel et al., 2018).

Converting image scale to real-world scale is essential for crowd dynamic analysis by computer vision techniques and conversions based on camera parameters are common practice (Hediyeh et al., 2014b). Scale conversion is based on the similarity triangles that are formulated by points in real world, corresponding points in the image and the central point of camera lens. After getting the real length of trajectories, crowd dynamics (e.g., walking speed) can be analyzed. For example, we can calculate the instantaneous walking speed by using the displacement of tracked points over successive frames (Sultan and Khan, 2013). We can also calculate average walking speed by setting two parallel screen lines and calculating the shortest distance between two intersecting points of trajectory with two screen lines (Hediyeh et al., 2014a).

2.3. Knowledge Gaps and Intended Contribution

A Taken together, there are two gaps in existing studies on computer vision based crowd analytics: (1) few studies focus on individual dynamic trajectory extraction from distorted video records, especially for high density crowd videos; (2) existing studies ignored calculating basic metrics for crowd analysis from the on-site detection data, which limits the application for crowd safety monitoring and control in transit stations.

This study proposes a novel approach to fill these gaps in two major ways.

(1) We integrate pedestrian detection, people tracking, and camera calibration technique to acquire individual trajectories from distorted video records; moreover, an individual's head is used to isolate a person's visible features under high density crowd conditions. Formulating three methods for crowd analysis in video,

a. An IO (In and Out) Matching flow volume counting algorithm.

b. A Voronoi diagram based crowd density calculation algorithm, and a trajectory based walking speed calculation algorithm, to extract the three metrics from on-site video records.

c. A prediction method is employed to get the positions of "missing" people.

3. Methodology

We developed a comprehensive model framework with three layers to implement crowd dynamic analysis with computer vision techniques, as shown in Figure 3.1: (1) head detection and tracking for individual trajectory extraction, (2) camera calibration for undistorting and scale converting, (3) crowd analysis. Video feeds are input into the analysis framework and are detected and tracked to acquire individual trajectories. Head detection and a tracking-by-detection paradigm is employed to adapt to high-density crowd scenarios. Camera calibration is used to rectify barrel distortion of video frames caused by fisheye lens and convert an images' scale to real-world scale. We propose a trajectory-based person counting method, an individual-based crowd density calculation method, and a walking speed calculation method to obtain the basic crowd metrics. These results can be displayed or archived for specific application or further analysis.

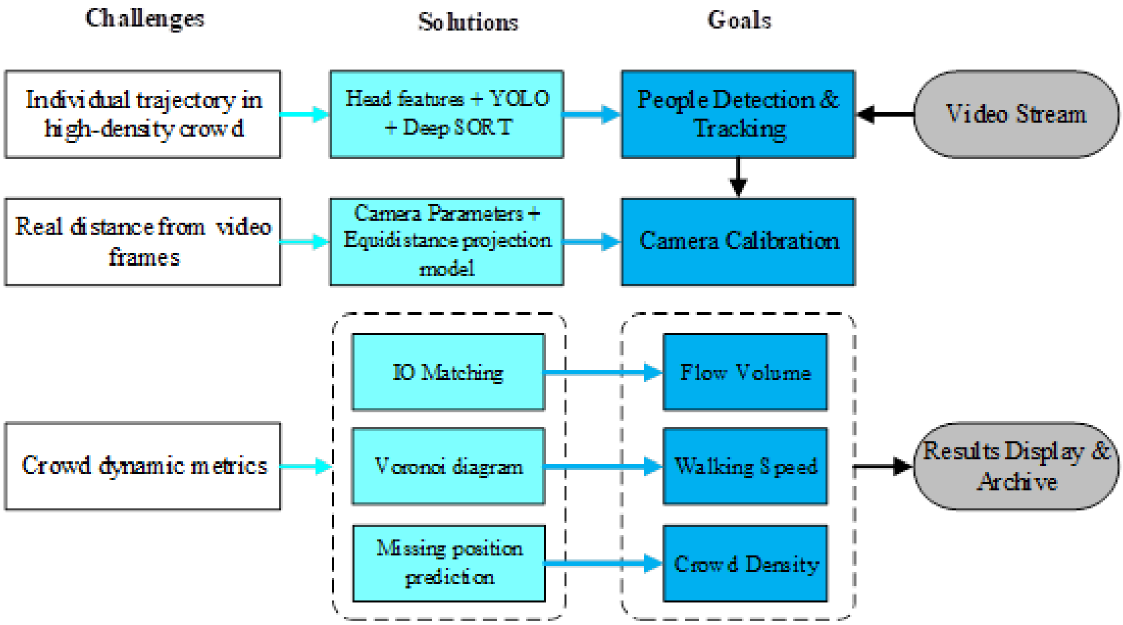

Challenges
Solutions
Goals
Individual trajectory in high-density crowd
Head features + YOLO + Deep SORT
People Detection & Tracking
Video Stream
Real distance from video frames
Camera Parameters + Equidistance projection model
Camera Calibration
IO Matching
Flow Volume
Crowd dynamic metrics
Voronoi diagram
Walking Speed
Results Display & Archive
Missing position prediction
Crowd Density

Figure 3.1. Research Architecture

3.1. People Detection and Tracking

To achieve the goal of monitoring high-density crowds in transit station with crowd analytics metrics, we firstly extract individual dynamic positions and trajectories from video records. In high density scenarios, head detection is a commonly used technique. Additionally, computational efficiency and detection accuracy are essential for use of this technology in practice. Based on these requirements, we propose a generalized head detection and tracking framework by integrating techniques of YOLO (You Only Look Once) and Deep SORT (Simple Online and Realtime Tracking with a Deep Association Metric), as shown in Figure 3.2. People are detected by YOLO algorithm which is a state-of-the-art real-time object detection system in both detection accuracy and computational efficiency. Deep SORT is employed to track the previously detected people as an extension of the SORT algorithm. Deep SORT excels over its predecessor by performing better under occlusion conditions

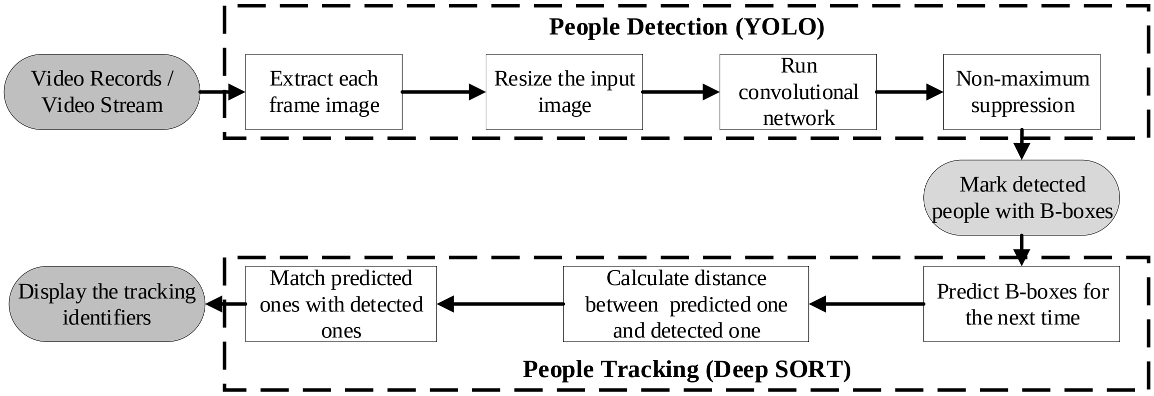

People Detection (YOLO)
Video Records / Video Stream
Extract each frame image
Resize the input image
Run convolutional network
Non-maximum suppression
Mark detected people with B-boxes
Predict B-boxes for the next time
Calculate distance between predicted one and detected one
Match predicted ones with detected ones
Display the tracking identifiers
People Tracking (Deep SORT)

Figure 3.2. People Detection and Tracking Framework

3.1.1. People Detection

Video feeds are input into YOLO, and each image frame is extracted to detect all identifiable people within it. YOLO resizes the image into a square size (e.g., 448 × 448) and divides the resized image into multiple grid cells (e.g., 7 × 7). Bounding boxes are predicted for each cell based on a convolutional neural network, as shown in Figure 3.3. The bounding box is a rectangular box that can be determined by *x* and *y* coordinates of the central point, width w, and height *h*. It marks the detected object with a confidence score conf. Therefore, each predicted bounding box contains five values (*x, y, w, h, conf*). The confidence score conf is calculated by Equation (1).

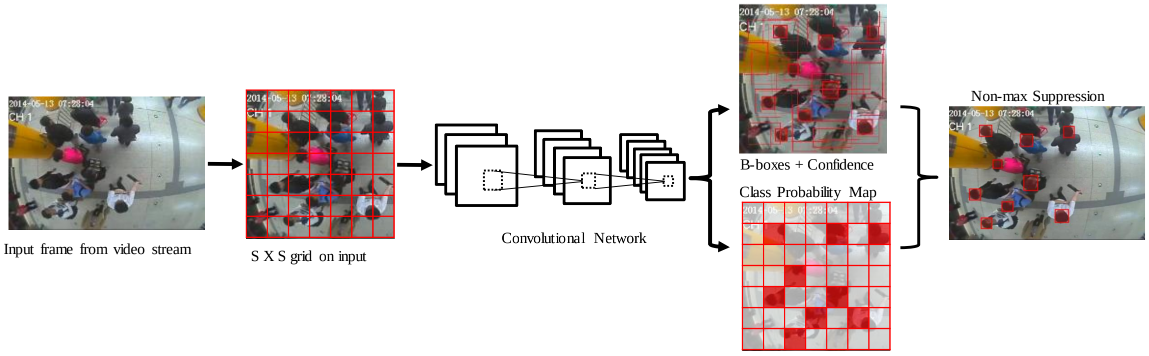


Figure 3.3. People Detection Architecture

$$conf = P\left(\text{Object}\right) \times \text{IOU}_{\text{pred}}^{\text{truth}} \tag{1}$$

In this equation *P*(Object) is the probability of object existing in the cell. If no object exists in the cell, *conf* is 0. Otherwise, *conf* equals the IOU (intersection over union) between the predicted box and the ground truth.

Meanwhile, each grid cell also predicts *C* conditional class probabilities *P*(Class|Object) for the detected object and we can get a class probability map, in which *C* is the number of classes. In this study, we mainly focus on the class of people's heads.

Consequently, the class-specific confidence scores confc for each bounding box are calculated by Equation (2). The bounding boxes with class-specific confidence scores are filtered out by a predefined threshold. A non-maximum suppression algorithm is used to filter out the bounding boxes with high values of IOU between these bounding boxes. The remaining ones are detected bounding boxes for the people. More details can be found in the work of Redmon et al. (2016).

$$conf_c = P\left(\text{Class} \middle| \text{Object}\right) \times P\left(\text{Object}\right) \times \text{IOU}_{\text{pred}}^{\text{truth}} \tag{2}$$

3.1.2. People Tracking

The detected bounding boxes, corresponding confidence scores and features are input into the Deep SORT module, and Figure 3.4 shows the tracking architecture. A Kalman filter is employed to predict positions of the detected bounding boxes for the next time step. At the next time step, new detections are input, and the predicted tracks are processed following their status (i.e., confirmed or unconfirmed) to match with new detections.

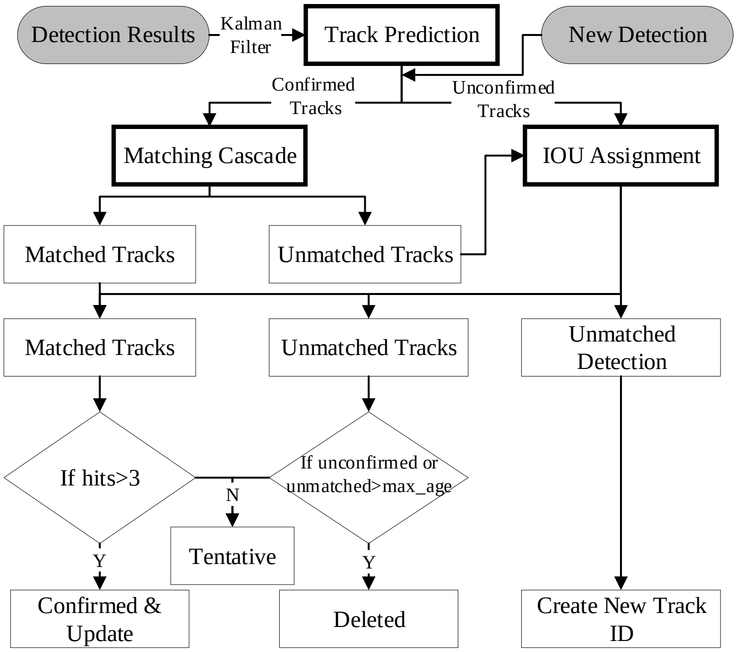


Figure 3.4. People Tracking Architecture

The confirmed tracks are matched by using matching cascade algorithm. The matching cascade algorithm records the time duration of each track since the last update, and the shorter track are prioritized first. For matching, the similarity of motion information (e.g., the position of bounding boxes) and the similarity of appearance features between predicted Kalman tracks and newly detected tracks are measured to get weighted similarities. The weighted similarities are input into the Hungarian algorithm, which is a widely used combinatorial optimization algorithm, to get a matching matrix. A matching threshold *max_threshold* is used to get matched tracks and unmatched tracks. This method is resilient to missed tracking in sequential frames due to partial occlusion.

Then, unconfirmed tracks and the unmatched tracks from the matching cascade phase are input into IOU association algorithm to conduct matching with unmatched newly detected ones. IOU values between unmatched tracks and newly detected tracks are calculated as matching metric, as shown in Figure 3.5. IOU values are input into the Hungarian algorithm to get a matching matrix. A matching IOU threshold *max_iou_threshold* is used to acquire the matching results.

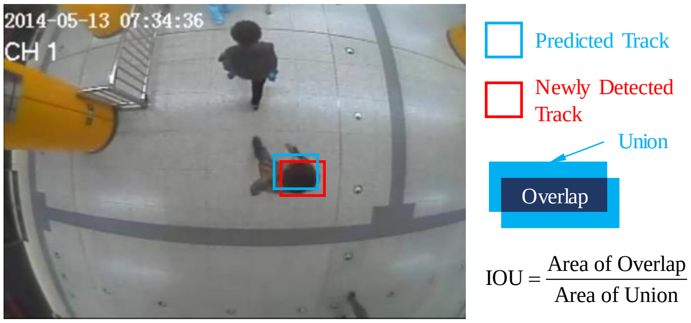


Figure 3.5. Definition of IOU

After the process, we get three result sets: matched tracks, unmatched tracks, and unmatched detections. If the tracks are matched in n frames, we set them as confirmed tracks and update their information. Otherwise, their statuses are still tentative. For the unmatched tracks, if they are unconfirmed tracks, we delete them from the track list. If they are confirmed, but unmatched time is larger than a threshold *max_age*, they are also deleted. For the unmatched detections, we create new track identifiers for them. More details can be found in the work of Wojke et al. (2017).

3.2. Camera Calibration

To ascertain critical crowd metrics, accurate trajectories in a real-world coordinate system are needed. A camera calibration is used to convert the distorted images caused by the fisheye lens and remap the image coordinate system to the real-world coordinate system in this study.

3.2.1. Camera Parameters

For a pinhole camera, we can use two parameter sets to project a point in real-world coordinates to pixel coordinates in the image frame. Extrinsic parameters transform world coordinates to camera coordinates, which is a rigid transformation from 3D to 3D. Intrinsic parameters transform camera coordinates to pixel coordinates in the image frame, which is a projective transformation from 3D to 2D.

Extrinsic parameters include a rotation matrix **R** and a translation vector **t**. The rotation matrix describes the camera rotation information relative to three coordinate axes of the world coordinate system; the translation vector describes the translation information of the camera optical center relative to the origin of the world coordinate system. A point $p_w\,(x_w, y_w, z_w)$ in the world coordinate system is transformed to a point $p_c\,(x_c, y_c, z_c)$ in the camera coordinate system, as shown in Equation (3)

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \mathbf{R} \times \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \mathbf{t} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \times \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \tag{3}$$

The point p_c projects to point p (x, y) in image plane followed by Similar Triangles rule for a simple pinhole camera, as shown in Figure 3.6. This projection can be represented by Equation (4). Next, we transform the point p to p_i (u, v) in pixel coordinate system with image resolution, as shown in Equation (5). Integrating the two equations in Equation (4), we get the camera intrinsic matrix **K**, as shown in Equation (5).

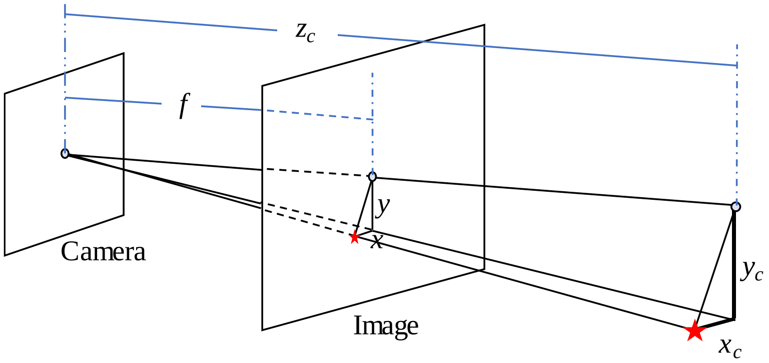


Figure 3.6. Projection from Camera Coordinate System to Pixel Coordinate System

$$z_c \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}, \quad \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/dx & 0 & u_0 \\ 0 & 1/dy & v_0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4}$$

In this equation, f is focal length in world units, which is one of the intrinsic parameters. dx, dy are the length of one pixel in world units; u_0, v_0 are the optical center in pixel units, which are all the intrinsic parameters

$$\mathbf{K} = \begin{bmatrix} f/dx & 0 & u_0 \\ 0 & f/dy & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

For the fisheye lens camera, the position of a point will be distorted when it is projected to image plane. We assume the fisheye lens obeys the equidistance projection model that is the most commonly used (Kannala and Brandt, 2006), and this distortion can be modeled by using distortion parameters. In this study, we employ a general projection form as shown in Equation (6) (Kannala and Brandt, 2006). Therefore, point *p* is distorted to point *p'* (*x'*, *y'*), as shown in Equation (7).

$$\theta_d = \theta + k_1\theta^3 + k_2\theta^5 + k_3\theta^7 + k_4\theta^9 \tag{6}$$

Where: $k_{1\text{-}4}$ are the distortion parameters, θ is the angle between the principal axis and the incoming ray, θ_d is the angle between the principal axis and the projection ray.

$$x' = \frac{\theta_d \times z_c}{\sqrt{x_c^2 + y_c^2}} \times \frac{x_c}{z_c}, \quad y' = \frac{\theta_d \times z_c}{\sqrt{x_c^2 + y_c^2}} \times \frac{y_c}{z_c} \tag{7}$$

Where: (*x'*, *y'*) is the coordinates of the distorted point, other indicators are the same as mentioned before.

For the fisheye lens camera used in this study, we ascertain all 20 parameters, including 12 extrinsic parameters, 4 intrinsic parameters, and 4 distortion parameters. They are used to undistort the video frames and convert the image scale.

3.2.2. Video Calibration

After getting the camera parameters, intrinsic parameters and distortion parameters can be used to undistort video image frames with Equation (4) – (7). For image scale conversion, we assume the tracking point marked on individual head is his/her position point on the ground. When a tracking point moves a distance of x_c, as shown in Figure 3.7, the displacement on the undistorted video image frames will be *x*. We obtain the camera installation height *H* from extrinsic parameter z_c, and the focal length from intrinsic parameter *f*. With the body height *h* of human being, we can determine Equation (8) followed by the Similar Triangles rule and calculate the tracking point moving distance x_c. We employ this distance to calculate walking speed for each person.

$$\frac{f}{x} = \frac{H-h}{x_c} \Rightarrow x_c = (H-h)\square x/f = (z_c - h)\square x/f \tag{8}$$

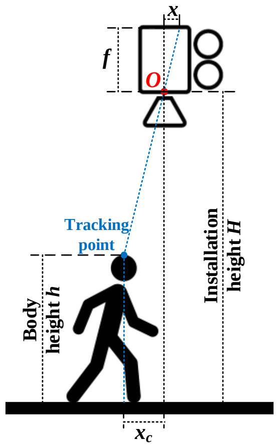


Figure 3.7. Illustration of Image Scale Conversion

3.3. Crowd Metrics Calculation

Flow volume, crowd density and walking speed are the essential metrics to describe crowd status. Therefore, we propose an IO (In and Out) Matching method to identify the people walking direction and count the people passing through a specific cross-section (e.g., a door of bus or train) for acquiring flow volume. We formulated an individual-based crowd density calculation method to measure the accurate crowd density. We also used the trajectory information to calculate individual walking speeds.

3.3.1. Flow Volume

We designated a "virtual gate" (VG) as a cross-section for counting flow volume, which is a rectangle region, as shown in Figure 3.8. Moreover, we defined four walking directions for flow counting, and an IO Matching method to count people walking through the VG.

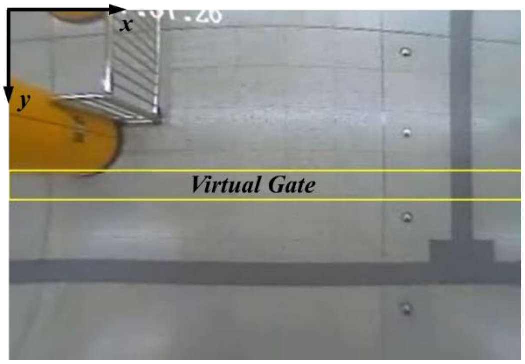


Figure 3.8. Designated Virtual Gate in One Scenario

We established two real-time attributes for each tracked pedestrian: walking direction and IO status. We calculated the walking direction on vertical and horizontal levels using the latest *n* track information for each tracking ID at each time frame. Firstly, for each detected person *i* their positions are ascertained. The counter will be triggered only when the number of position records is larger than threshold *min_last* = 5; or it will go to the next pedestrian *i*+1. The benefits of this process are twofold. The first is to mitigate the counting disturbance of some misidentified individuals, and the second one is to avoid the misjudgment of walking direction by using limited position information. The threshold can be set a suitable value based on test outcomes. IO status is updated following spatial relations between the individual trajectory and edges of the VG. If the trajectory of people *i* intersects two edges of the VG, IO status of the people will be set as passed, and people *i* will be counted in the flow volume of his/her walking direction. If people *i* is walking to the VG or walking in the VG, his/her the IO status will be set as tentative for further analysis. If people *i* is walking out VG, he/she will be matched with people with tentative status. We calculate the similarity of their features as the matching index; a predefined threshold is used to get the matched one. They are counted as one person by the counter of the corresponding walking direction. Both of the IO statuses of the matched people are set as passed. This method addresses the emergent challenge of incorrectly changing tracking IDs. In this study, we use walking speed as the matching index

3.3.2. Crowd Density

In a high-density crowd, people can only perceive the local density without a whole picture of the entire crowd. To reflect the individual heterogeneity, we use the Voronoi diagram to formulate the occupancy area for each people and calculate individual-based crowd densities.

Voronoi diagram (Longley et al., 2005) is a plane partition method that divides a plane into regions based on a given set of points in the plane. At first, we formulate a triangulated irregular network that meets the Delaunay criterion using the points (Longley et al., 2005). Then, the perpendicular bisectors for each triangle edge are generated to form the edges of the Voronoi cells. Then, the perpendicular bisectors for each triangle edge are generated to form the edges of the Voronoi cells. The occupancy area A_i for a point *i* is the region containing the point. Consequently, we calculate the individual-based crowd density D_i by Equation (9).

$$D_i = 1.0/A_i \tag{9}$$

However, some people in the image frames may not be detected due to occlusion or view angle, and we cannot get their positions in some frames. Consequently, failing to detect these occluded individuals will negatively impact the accuracy of crowd density. To overcome this challenge, we propose a position prediction method to get coordinates of the "missing" people.

For the unmatched tracking identifiers at frame *t*, we regard them as the "missing" identifiers. We assume that they will remain their walking speed $\boldsymbol{v}_t$ (v_{xt}, v_{yt}) at frame *t*; their positions (*x*, *y*) at frame *t*+*n* are predicted by Equation (10)

$$x = v_{xt}\left(t+n\right)\times\Delta t + x_t; \quad y = v_{yt}\left(t+n\right)\times\Delta t + y_t \tag{10}$$

The (x_t, y_t) is position of the "missing" people at frame *t*; *Δt* is the time interval of one frame; other variables are same as above.

Considering that the prediction error increases over time, a maximum effective time threshold is predefined. If the update time of an unmatched tracking identifier is larger than the threshold, the identifier will be removed from the prediction process and be considered as lost forever. In this study, we set the value of threshold *max_age* in tracking model as the maximum effective time threshold.

3.3.3. Walking Speed

Walking speed is an essential indicator for flow volume counting and crowd density calculation, as well as a crucial metric for crowd analytics. For each tracked person, we record the coordinates in real-time. Therefore, we can calculate the instantaneous walking speed $\boldsymbol{v}_t$ (v_{xt}, v_{yt}) at frame *t* by Equation (11).

$$v_{xt} = \left(x_t - x_{t-1}\right)/\Delta t; \quad y = \left(y_t - y_{t-1}\right)/\Delta t \tag{11}$$

In the above equation (x_{t-1}, y_{t-1}), (x_t, y_t) are positions of the tracked people at frame *t*-1 and *t*;*Δt* is the time interval of one frame.

As stated in crowd density calculation section, we will remain the walking speed $\boldsymbol{v}_t$ (v_{xt}, v_{yt}) for "missing" people. The walking speed will be recorded until the "missing" time reaches maximum effective time threshold *max_age*.

4. Model Implementation

To validate the model framework (Figure 10) proposed in this study, we used the system to evaluate video records from two major subway stations in China. Pedestrian detection was implemented based on the “tiny” version of YOLOv3 (YOLO3-tiny) which is faster than other models and more suitable for practical application (Redmon and Farhadi, 2018). Pedestrian tracking is implemented with the standard version of Deep SORT (Wojke et al., 2017).

4.1. Data Description

To verify the generalization and robustness for various scenarios of the proposed model framework, we utilized surveillance video data of two different scenarios to implement the model, as shown in Figure 4.1. The first scenario (Scenario A) is alighted passengers walking to a stair and transferring to another line. A top-view fisheye lens camera shows the walking behavior of passengers in front of the entrance of the stair. The second scenario (Scenario B) is passengers walking in a passage while transferring to another line. Similarly, a top-view fisheye lens camera captures the walking behavior of passengers. For video records in these scenarios, the width and height of frame are 352 × 240 pixels and the frame rate is 30 frames per second. In this study, we selected two video records for each scenario, and each video record lasts for about 30 minutes. One video record for each scenario is used to train the model, and another one is used to test the model.

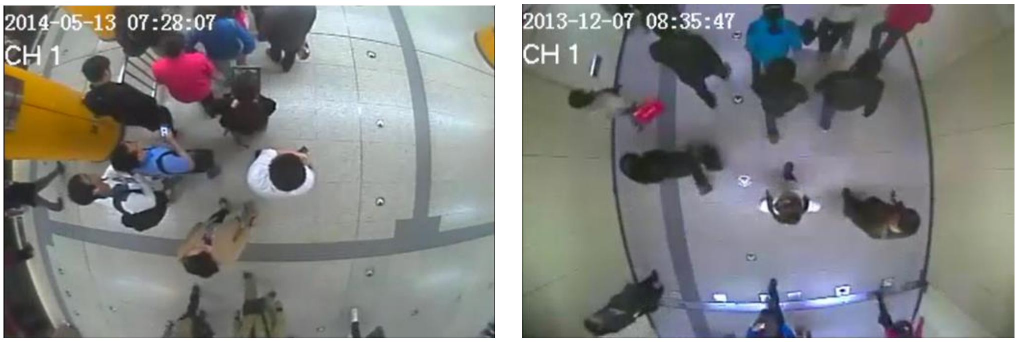


Figure 4.1. Surveillance Video Records of Two Scenarios (left: Scenario A, right: Scenario B) from Beijing Subway

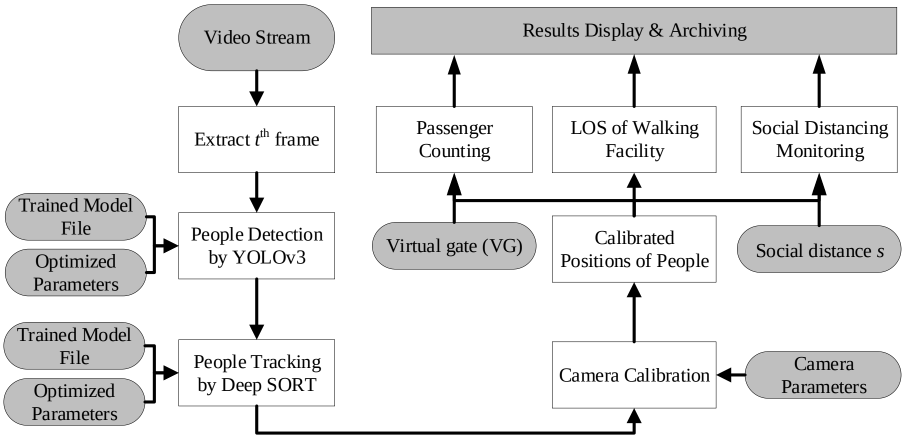


Figure 4.2. Implementation of the Entire Model Framework

4.2. Model Training

To maximize model performance for detection and tracking, we trained the detection and tracking model to get the weights files and optimized parameters. We extracted 500 frame images from each training video and labeled the heads of all identifiable people in images with rectangle boxes, as shown in Figure 4.3. Coordinates of label boxes and the classification (i.e., Head class) in an image are recorded into an annotation file. The labeled images and annotation files are randomly split into a training set, a testing set, and a validation set.

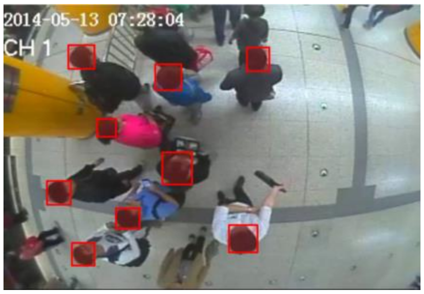
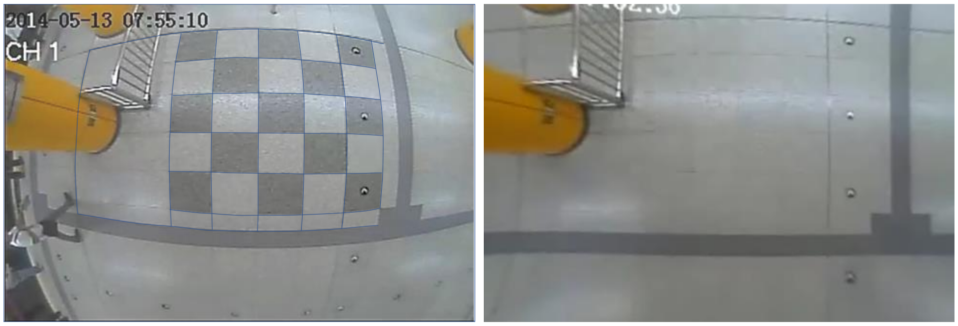


Figure 4.3. Extracted Images from Top-view Videos

These data are employed to tune the training hyper-parameters at first, and then train the model with the optimized hyper-parameters to acquire the best model weights file. After getting a well-performed detection model, we refine tracking model parameters *max_dist* = 0.7, *max_iou_dist* = 0.7 and *max_age* = 5. The original weights file provided by Wojke et al. (2017) is employed as the cosine metric feature representation for person re-identification in this study.

Meanwhile, we extract camera parameters from video records. Checkerboard images should be captured by the camera to calibrate the parameters. Note that the extrinsic parameters are calculated based on the world coordinate system in the checkerboard plane, we choose the ground floor as the checkerboard plane. For reducing the labor work further, we employ the floor tiles shown in the videos as the checkerboard for camera calibration. We extract several images with floor tiles and draw grid cells covering the floor tiles exactly, as shown in Figure 12; the floor tiles are square shape with 0.65 *m* of edge. The checkerboard images and the edge length are input into the camera calibration tool; camera parameters and the undistorted images are output as shown in Table 3 and Figure 12



(a) Checkerboard with floor tiles (b) Undistorted image frame

Figure 4.4. Camera Calibration

Table 4.1. Camera Intrinsic Matrix and Distortion Coefficients

Camera intrinsic matrix			Fisheye lens distortion coefficients	
147.00	0.00	197.20	-0.13399 (k_1)	0.00933 (k_2)
0.00	106.50	128.06	-0.02622 (k_3)	-0.00638 (k_4)
0.00	0.00	1.00	0.0000 (k_5)	0.0000 (k_6)

We measured the size of checkerboard in undistorted image frames, 150 *pixels* × 150 *pixels*. From Figure 3.6, z_c is calculated in accordance with Similar Triangles rule, z_c = 3250 *mm* × 147 *pixels* / 150 *pixels* = 3185 *mm*. Individual body height is set as h = 1600 *mm* ((CNHFPC), 2015), so we can convert image scale to real scale using x_c = (3185 *mm* – 1600 *mm*) ×*x pixels* / 147 *pixels* = 10.78*x mm*

4.3. Model Validation

We configured the model framework with trained models and optimized parameters. A "virtual gate" was set up in the middle of the frame with a width of 20 pixels. A new video record of Scenario A was input into the model framework and we selected a clip that records passengers' walking behavior in a time-headway within 250 seconds to validate model results. Utilizing the CV-based flow volume counting method, we obtained the flow volume in every 5-second as shown in Figure 13. The ground truth of the flow volume in every 5-second was counted manually and the error between ground truth and model results was calculated as shown in Figure 4.5. The total number of passengers from ground truth is 200, while total number counted by the proposed method is 190. Therefore, the counting accuracy of the proposed CV-based people counting method is 95%. We analyzed the negative error, and they are mainly caused by occlusion in a high density crowd.

We recorded individual-based crowd density and walking speed of each frame and validated the results by the fundamental diagram of density-speed. We reorganized the data by averaging walking speed with crowd density interval of 0.01 people/m^2 to reduce noise. The CV-based model results are shown in Figure 4.6 as well as the benchmark survey results of Older (1968) and Mōri and Tsukaguchi (1987). Older surveyed the crowd density and walking speed at an open boundary shopping street in Slough, England; Mōri and Tsukaguchi conducted the survey at open boundary footpaths in Osaka, Japan. From the figure, the relationship between crowd density and walking speed from the CV-based model result is consistent with the benchmark datasets. When crowd density is less than 3 people/m^2, the model result is consistent with Older's survey result and when crowd density is larger than 3 people/m^2, the model result is consistent with Mori and Tsukaguchi's survey result. As the benchmark datasets, the scenarios are different from the studied scenario, so they only get the same values on some sections. But this result validates that our model performs well for crowd density and walking speed calculation.

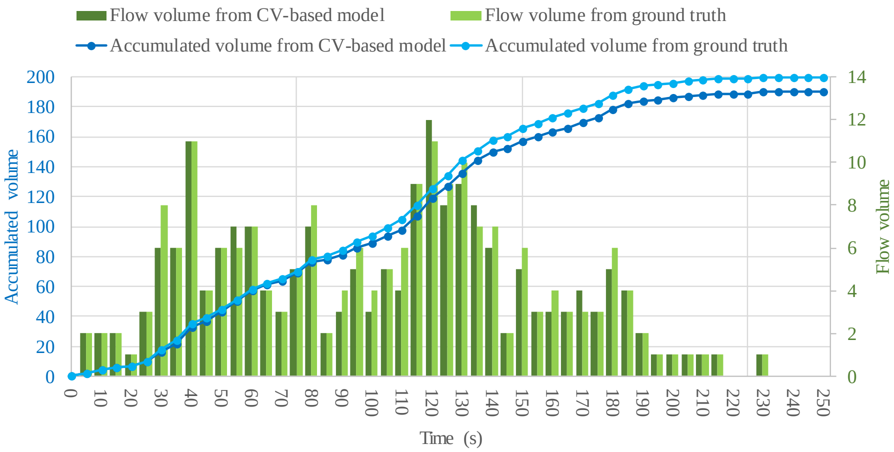


Figure 4.5. Validation Results for People Counting by the Proposed Model Framework

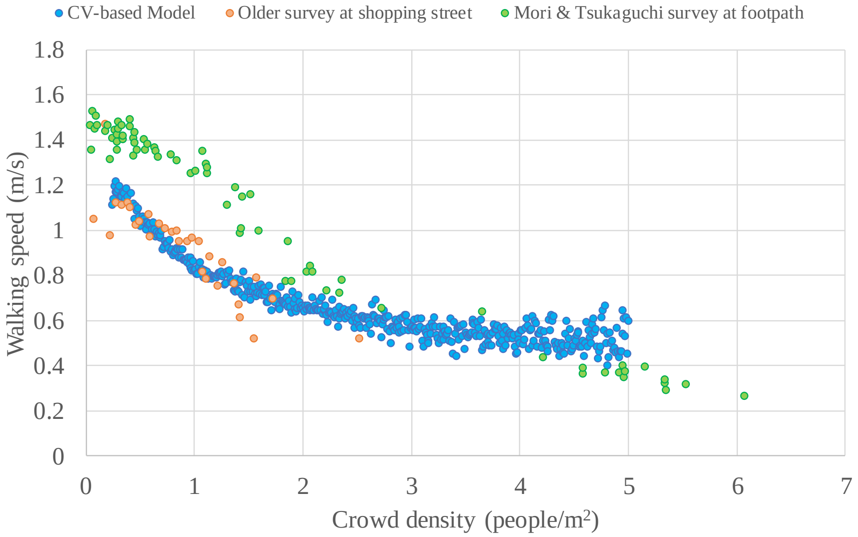


Figure 4.6. Fundamental Diagram of Stair Scenario

We used a new video record from Scenario B as the model framework input and selected a clip that records passengers' walking behavior in a time-headway within 180 seconds to validate model results. The CV-based flow volume counting results as well as the flow volume ground truth in every 5-second are shown in Figure 4.7. The number of people passing through the "virtual gate" counted by the model framework is 191, while the ground truth is 187. The accuracy is 97.87%, which is higher than the accuracy of the stair scenario because the crowd densities in this scenario are smaller than the ones of the stair scenario.

Results of individual-based crowd density and walking speed are validated by the fundamental diagram of density-speed. We reorganized the data by averaging walking speed with crowd density interval of 0.01 people/m^2 to reduce noise. The CV-based model results are shown in Figure 4.8 as well as the benchmark survey results of Older (1968) and Mōri and Tsukaguchi (1987). From the figure, the amplitudes are different from each other for three datasets. This is primarily due to the natural variation caused by different countries, scenarios, and walking habits in the datasets. However, the relationship between crowd density and walking speed from the CV-based model result is consistent with those from the benchmark datasets, which indicates walking speed decreases as crowd density increases. This result shows the proposed model framework also works well for crowd density and walking speed calculation in passage scenario. Above all, our model framework performs well for crowd dynamic metrics calculation in multiple scenarios.

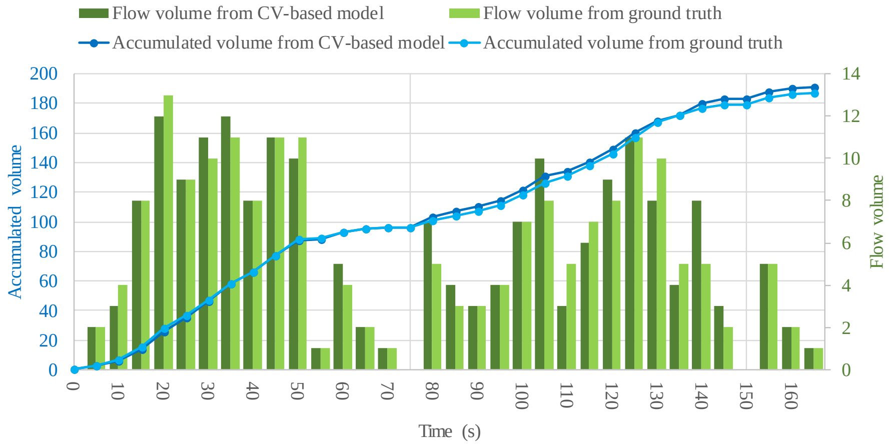


Figure 4.7. Counting Results of New Video Data from Camera in Transfer Passage

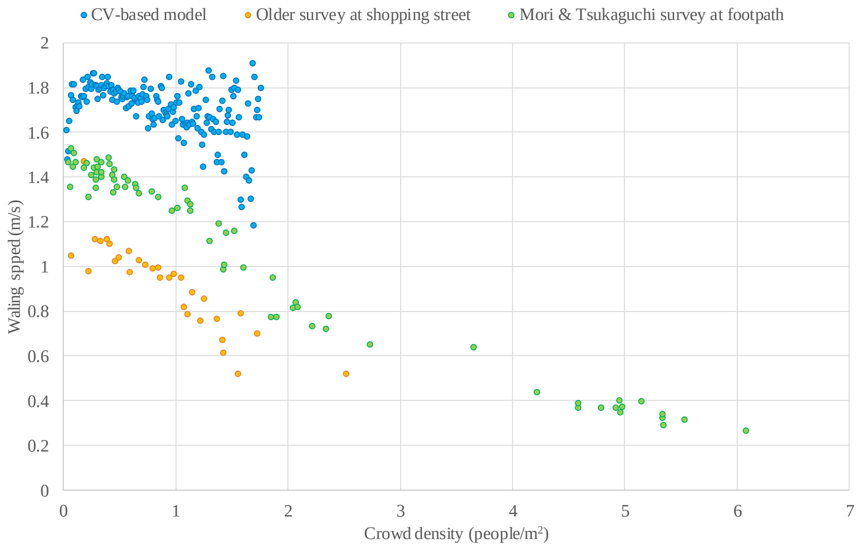


Figure 4.8. Fundamental Diagram of Passage Scenario

5. Application and Discussions

In this study we developed practical applications including automatic passenger counting, crowd safety monitoring, and social distancing monitoring for COVID-19.

5.1. Automatic Passenger Counting

Passenger counting is the application of flow volume in public transit system. In most metro systems, ticket records only provide origin and destination (OD) information to transit providers. However, there are several potential paths between an OD pair in a large metro network, and several different operation companies may provide service in the network. Under this situation, clearing and settling of ticket fares is a crucial issue for the joint operation companies. The main solution in practice is estimating path choice of passengers based on individual attributes and travel cost of each path. However, the solution needs a large amount data to tune the model parameters and network extension will impact the model performance.

CV-based flow volume counting method proposed in this study provide another promising solution for transferring passenger counting. The validation results show the counting accuracy of 95% - 98% and it also records the transferring direction (e.g., Line A to Line B and Line B to Line A). Furthermore, the real-time display function is developed for monitoring, as shown in Figure 5.1.

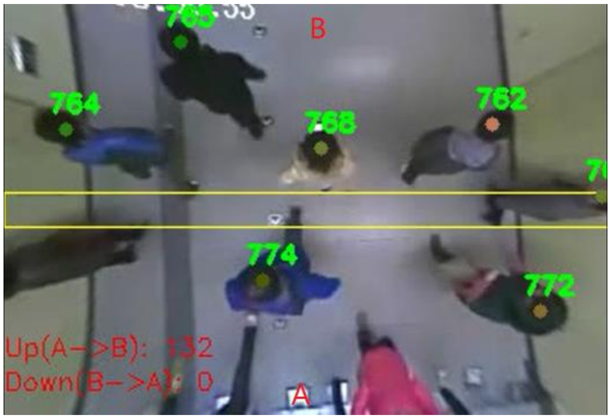


Figure 5.1. Transferring Passenger Counting

5.2. Level Of Service (LOS) Evaluation

Level of service (LOS) is an important metric of performance in analysis of existing pedestrian facility conditions. Evaluating LOS of facilities contributes to identify any potential problems at an early stage (Cepolina et al., 2018). Highway Capacity Manual (HCM) provided a guidance of LOS criteria for walkway based on crowd density in six levels (A-F) (Rouphail et al., 1998), as shown in Table 4.

Table 5.1. Recommended HCM walkway Level of Service (LOS) criteria (Rouphail et al., 1998)

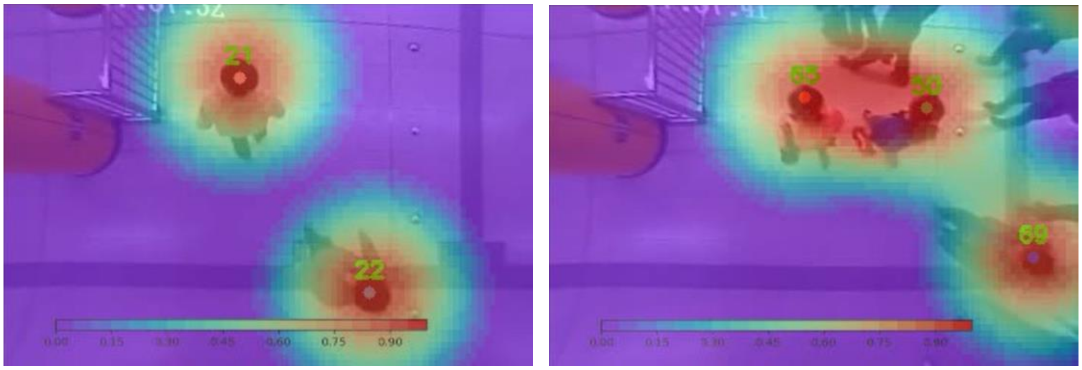
LOS	A	B	C	D	E	F
Density(people/m^2)	≤0.18	0.18-0.27	0.27-0.45	0.45-0.71	0.71-1.33	≥1.33
Status	Free flow	Reasonably free flow	Stable flow	Approaching unstable flow	Unstable flow	breakdown flow

Based on the criteria, we can evaluate the LOS of the walking facilities in this study. For the walking facility in Scenario 1, the average crowd density is 0.574 ped/m^2 and the LOS is D which means passengers walk in crowded surroundings. For the walking facility in Scenario 2, the average crowd density is 0.228 ped/m^2 and the LOS is B which means passengers can walk freely as they desired, and the facility can service more people. Furthermore, we can use LOS to evaluate the effectiveness of infrastructure improvement quantificationally.

5.3. Social Distancing Monitoring

The COVID-19 pandemic has impacted people’s travel behaviors and public transit operations. Many cities are developing solutions to maintain social distancing in public spaces for a phased reopening, especially for passengers in transit stations. Considering the dynamic position tracking technique in this study, the proposed model framework can be utilized to monitor the social distancing in transit stations.

We set an individual safety area for each tracked person based on their tracked positions and set the half-length of social distance (for example, 6 feet in the U.S.) as the radius of the safety area. The safety area is displayed by different colors with the distance to the tracking position to show the emergency levels, as shown in Figure 5.2. We choose blue (0, 0, 255), green (0, 255, 0), yellow (255, 255, 0) and red (255, 0, 0) to indicate 1, 2/3, 1/3, 0 of the half-length of social distance, and the color of other distance is gotten by interpolation. If the safety areas of passengers are separated, it means they keep the social distancing rules, as shown in the left of Figure 5.2. While, if the safety areas integrate with each other, it means these passengers don’t keep the social distancing rules, as shown in the right of Figure 5.2. Green color integration means that their distance is about 50% of social distance (3 feet) (e.g., passenger 69 and 50); red color integration means that their distance is about 10% of social distance (0.6 feet) (e.g. passenger 65 and 50).



(a) keeping social distancing (b) violating social distancing

Figure 5.2. Social Distancing Monitoring

6. Conclusions

This study aims to formulate a generalized CV-based crowd analytics model framework. We focus on solving three challenges to achieve this goal: (1) acquiring dynamic information from the video data, (2) projecting distorted images to the real-world reference system, (3) calculating essential metrics from trajectory information. We implemented pedestrian detection and tracking with head features to solve the first challenge. We trained the model with our data and the detection accuracy was 0.80 of precision and 0.75 of recall ratio. The performance provides the model a opportunity for application in practice. For the second challenge, we employ equidistance projection model to calibrate the distorted image frame and convert the image scale to real scale. Based on the calibrated results, we formulate an IO Matching counting method, an individual-based crowd density calculation method based on Voronoi diagram, and a trajectory-based walking speed calculation method to solve the third challenge. Video records from two different scenarios to validate the model framework, and results show high flow volume counting accuracy of 95%-98% and reasonable density-speed fundamental diagrams which are consistent with empirical studies.

Based on the crowd analytics, we also developed several practical functions, including automatic passenger counting, crowd safety monitoring, and social distancing monitoring for COVID-19. Additionally, in future works, we will improve the person tracking model to solve the challenge of lost tracking people fundamentally. Furthermore, more functions, such as trajectory analysis, will also be added in the future.

REFERENCES

[1] (CNHFPC), T.C.N.H.a.F.P.C., 2015. The Nutrition and Health Status of the Chinese People (2015 Report). The Chinese National Health and Family Planning Commission, Beijing.

[2] Boyle, D.K., 1998. *Passenger counting technologies and procedures*. National Academy Press, Washington, D.C.

[3] Cepolina, E.M., Menichini, F., Gonzalez Rojas, P., 2018. Level of service of pedestrian facilities: Modelling human comfort perception in the evaluation of pedestrian behaviour patterns. *Transportation Research Part F: Traffic Psychology and Behaviour* 58, 365-381.

[4] Chan, A.B., Zhang-Sheng John, L., Vasconcelos, N., 2008. Privacy preserving crowd monitoring: Counting people without people models or tracking, *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-7.

[5] Girshick, R., 2015. Fast R-CNN, *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440-1448.

[6] Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587.

[7] Hediyeh, H., Sayed, T., Zaki, M.H., Ismail, K., 2014a. Automated analysis of pedestrian crossing speed behavior at scramble-phase signalized intersections using computer vision techniques. *International journal of sustainable transportation* 8(5), 382-397.

[8] Hediyeh, H., Sayed, T., Zaki, M.H., Mori, G., 2014b. Pedestrian gait analysis using automated computer vision techniques. *Transportmetrica A: Transport Science* 10(3), 214-232.

[9] Helbing, D., Brockmann, D., Chadefaux, T., Donnay, K., Blanke, U., Woolley-Meza, O., Moussaid, M., Johansson, A., Krause, J., Schutte, S., Perc, M., 2015. Saving Human Lives: What Complexity Science and Information Systems can Contribute. *Journal of Statistical Physics* 158(3), 735-781.

[10] Jones, M.J., Snow, D., 2008. Pedestrian detection using boosted features over many frames, *2008 19th International Conference on Pattern Recognition*, pp. 1-4.

[11] Junior, J.C.S.J., Musse, S.R., Jung, C.R., 2010. Crowd Analysis Using Computer Vision Techniques. *IEEE Signal Processing Magazine* 27(5), 66-77.

[12] Kannala, J., Brandt, S.S., 2006. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(8), 1335-1340.

[13] Krams, O., Kiryati, N., 2017. People detection in top-view fisheye imaging, *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1-6.

[14] Li, J., Wang, L., Tang, S., Zhang, B., Zhang, Y., 2016. Risk-based crowd massing early warning approach for public places: A case study in China. *Safety Science* 89, 114-128.

[15] Li, M., Zhang, Z., Huang, K., Tan, T., 2009. Rapid and robust human detection and tracking based on omega-shape features, *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 2545-2548.

[16] Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W., 2005. *Geographic information systems and science*. John Wiley & Sons.

[17] Meinel, L., Findeisen, M., Heß, M., Apitzsch, A., Hirtz, G., 2014. Automated real-time surveillance for ambient assisted living using an omnidirectional camera, *2014 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 396-399.

[18] Mōri, M., Tsukaguchi, H., 1987. A new method for evaluation of level of service in pedestrian facilities. *Transportation Research Part A: General* 21(3), 223-234.

[19] Older, S.J., 1968. *Movement of Pedestrians on Footways in Shopping Streets*. Traffic engineering & control.

[20] Pinna, I., Dalla Chiara, B., Deflorio, F., 2010. Automatic passenger counting and vehicle load monitoring. *Ingegneria Ferroviaria* 65(2), 101-138.

[21] Puig, L., Bastanlar, Y., Sturm, P., Guerrero, J.J., Barreto, J., 2011. Calibration of Central Catadioptric Cameras Using a DLT-Like Approach. *International Journal of Computer Vision* 93(1), 101-114.

[22] Punn, N.S., Agarwal, S., 2019. Crowd Analysis for Congestion Control Early Warning System on Foot Over Bridge, *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pp. 1-6.

[23] Rabaud, V., Belongie, S., 2006. Counting Crowded Moving Objects, *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pp. 705-711.

[24] Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788.

[25] Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767.*

[26] Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6), 1137-1149.

[27] Reuter, L.G., 2003. Passenger Counting and Service Monitoring: A Worldwide Survey of Transportation Agency Practices, in: New York City Transit, A. (Ed.). New York City Transit Authority, New York.

[28] Ryan, D., Denman, S., Fookes, C., Sridharan, S., 2009. Crowd Counting Using Multiple Local Features, *2009 Digital Image Computing: Techniques and Applications*, pp. 81-88.

[29] Saleh, S.A.M., Suandi, S.A., Ibrahim, H., 2015. Recent survey on crowd density estimation and counting for visual surveillance. *Engineering Applications of Artificial Intelligence* 41, 103-114.

[30] Scaramuzza, D., Martinelli, A., Siegwart, R., 2006. A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion, *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, pp. 45-45.

[31] Seidel, R., Apitzsch, A., Hirtz, G., 2018. Improved person detection on omnidirectional images with non-maxima suppression. *arXiv preprint arXiv:1805.08503.*

[32] Sheng-Fuu, L., Jaw-Yeh, C., Hung-Xin, C., 2001. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 31(6), 645-654.

[33] Sheng, B., Shen, C., Lin, G., Li, J., Yang, W., Sun, C., 2018. Crowd Counting via Weighted VLAD on a Dense Attribute Feature Map. *IEEE Transactions on Circuits and Systems for Video Technology* 28(8), 1788-1797.

[34] Sidla, O., Lypetskyy, Y., Brandle, N., Seer, S., 2006. Pedestrian Detection and Tracking for Counting Applications in Crowded Situations, *2006 IEEE International Conference on Video and Signal Based Surveillance*, pp. 70-70.

[35] Sindagi, V.A., Patel, V.M., 2018. A survey of recent advances in CNN-based single image crowd counting and density estimation. *Pattern Recognition Letters* 107, 3-16.

[36] Song, J., Chen, F., Zhu, Y., Zhang, N., Liu, W., Du, K., 2019. Experiment Calibrated Simulation Modeling of Crowding Forces in High Density Crowd. *IEEE Access* 7, 100162-100173.

[37] Sørensen, A.Ø., Olsson, N.O.E., Akhtar, M.M., Bull-Berg, H., 2019. Approaches, technologies and importance of analysis of the number of train travellers. *Urban, Planning and Transport Research* 7(1), 1-18.

[38] Sultan, D., Khan, S., 2013. Estimating Speeds and Directions of Pedestrians in Real-Time Videos: A solution to Road-Safety Problem, in: Bandini, S., Aiello, L.C., Vizzari, G. (Eds.), *AgeingAI 2013 The Challenge of Ageing Society: Technological Roles and Opportunities for Artificial Intelligence*, Turin, Italy.

[39] Tai, Y., Hsieh, Y., Chuang, J., 2018. A Fully Automatic Approach for Fisheye Camera Calibration, *2018 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1-4.

[40] Tamura, M., Horiguchi, S., Murakami, T., 2019. Omnidirectional Pedestrian Detection by Rotation Invariant Training, *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1989-1998.

[41] Wang, C., Zhang, H., Yang, L., Liu, S., Cao, X., 2015. Deep People Counting in Extremely Dense Crowds, *Proceedings of the 23rd ACM international conference on Multimedia*. Association for Computing Machinery, Brisbane, Australia, pp. 1299–1302.

[42] Wang, T., Hsieh, Y., Wong, F., Chen, Y., 2019. Mask-RCNN Based People Detection Using A Top-View Fisheye Camera, *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp. 1-4.

[43] Wojke, N., Bewley, A., Paulus, D., 2017. Simple online and realtime tracking with a deep association metric, *2017 IEEE international conference on image processing (ICIP)*. IEEE, pp. 3645-3649.

[44] Ye, J., Chen, X., Yang, C., Wu, J., 2008. Walking Behavior and Pedestrian Flow Characteristics for Different Types of Walking Facilities. *Transportation Research Record* 2048(1), 43-51.

[45] Zhao, T., Nevatia, R., Wu, B., 2008. Segmentation and Tracking of Multiple Humans in Crowded Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(7), 1198-1211.

[46] Zhao, Z.-Q., Zheng, P., Xu, S.-t., Wu, X., 2019. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems* 30(11), 3212-3232.