International Journal of Technology and Engineering Studies
volume 5 issue 2 pp. 40-46 doi: https://dx.doi.org/10.20469/ijtes.5.40002-2

Based on Frond-End and Back-End Platfrom and Image Processing Algorithm to Design People Counting Analysis System

Po-Hsiang Liao*
Automotive Research and Testing Center, Changhua, Taiwan

De-Ciang Ye
Automotive Research and Testing Center, Changhua, Taiwan

Hung-Pang Lin
Automotive Research and Testing Center, Changhua, Taiwan

Hsuan-Ta Lin
Automotive Research and Testing Center, Changhua, Taiwan

Abstract: This paper presents a people counting method that is derived from traditional machine learning and deep learning algorithms. The proposed design mainly provides cognition information of a period of time, peak hour or off hour in specific public places, such as transportation tools, hotel lobby, and bus shelter. In previous literature, the traditional machine learning technique, such as Support Vector Machine (SVM) was adopted for the people counting. However, the pedestrian recognition rate of the previous means is lower than the deep learning method. Hence, the Convolutional Neural Network (CNN) is derived to the improved drawback of a worse recognition rate. However, given its computation task is very heavy when the processing of operating the system. Therefore, the proposed system is designed based on a two-stage architecture that contains the previous two methods in front-end and back-end, respectively. Among these, the first stage, which is front-end that mainly be used for pedestrian recognition. According to the above results, the people number counting could be executed. After that, the statistics consequence is classified into two-level, and then the back-end stage only needs to process pedestrian recognition of level two. Finally, the experimental results show that pedestrian recognition is increased and computational complexity is reduced when comparing with traditional machine learning and deep learning, respectively. The experimental results indicated that the proposed front-end design had 84.56% accuracy for detection performance. The other proposed architecture, which is the back-end, can obtain a detection accuracy of 93.59%. On the other hand, the proposed method also improves the average 29% execution time compared to the related designs. This system could be implemented to save management costs.

Keywords: *People counting, deep learning, machine learning*

Received: 15 February 2019; ***Accepted:*** 26 March 2019; ***Published:*** 26 April 2019

I. INTRODUCTION

In general, the surveillance system can be applied for several people counting applications, such as pedestrian traffic [1, 2] in public transportation, tourists flow estimation on bridge and people flow monitoring in building entrance [3, 4]. For previously, the people counting method includes manual statistics, rotary gate of mechanical equipment, infrared optoelectronic equipment. However, these traditional methods have some problems which contain labor costs increasing, inconvenient passing through of pedestrian, and inexact counting in high-flow people place. Therefore, image analysis methods are derived to implement an intelligent surveillance system. First, the machine learning theory which can be used to pedestrian detection is adopted in many literatures [5, 6, 7]. Among these, the SVM, SVM and adaptive

*Correspondence concerning this article should be addressed to Po-Hsiang Liao, Automotive Research and Testing Center, Changhua, Taiwan. E-mail: hsiang@artc.org.tw

boosting, AdaBoost are two of the famous methods based on the above theory to be accomplished. Furthermore, their statistical model classifiers are integrated with one kinds of the pedestrian feature extraction method, such as Histograms Oriented Gradients (HOG) or Local Binary Pattern (LBP). For precise obtaining feature of obstacle, a histogram representation is used by HOG which shows the computation and statistics results of the gradients data or edge directions in several region masks. Next, the LBP adopted a method which computes the relationship between a pixel point and its surroundings of point data, such that a part of texture features of obstacles can be an extrication data.

In addition to the aforementioned theories, deep learning has been used for image recognition. Two types of method are prevalent for image recognition. One is constituted of one-stage methods, such as You Only Look Once (YOLO), which directly predict the categories and positions of different targets by using solely CNN networks [8]; this type of method exhibits faster computation speed but lower accuracy [9]. The other type is two-stage, such as R-CNN [10], Fast R-CNN [11], Faster R-CNN [12], and Mask R-CNN [13], which first generates region proposals and then performs classification and regression. This type of method are more accurate but exhibits lower computation speed [9]. In this study, a Mask R-CNN was selected because of its high accuracy, fine segmentation of objects, and expandability for posture analysis.

After analyzing two intelligent recognition methods, we found that the architecture of deep learning is more complex than machine learning, but its hardware cost and power consumption are also higher for accomplishing superior detection accuracy. Hence, this paper considers both of hardware cost and detection accuracy, so people counting architecture of the proposed is consisted of two stages which concludes front-end and back-end, respectively. Among these, the front-end is responsible for preliminary counting people based on SVM algorithm of machine learning. Next, according to the previous statistics results, the second stage is enable when operating on the higher high-flow people condition. At this time, the HOG algorithm which belongs to deep learning method is started up computation, such that people counting can be implemented based on lower complexity computation and higher detection accuracy performance.

A. *System Architecture*

This paper proposes the integration of front-end and back-end image processing and image identification systems with cameras at two different angles. The system flowchart is presented in Fig. 1. First, images captured by the two cameras at different angles are transferred to the front-end image processing unit to process the signals. The images from Camera 1 are analyzed according to image morphology by applying a discrimination method to separate the foreground and background. The images from Camera 2 undergo pedestrian detection through feature extraction and application of a classifier technique. Both images are assessed and analyzed by targeting the region surrounding the bus stop areas according to a predetermined range, and the analysis results of the two images are further analyzed to obtain the stream density of people in the images. When the density is relatively high, the image signals and front-end image detection results are output to the back-end system for deep learning computation to obtain statistical information concerning the number of people. This process is performed for several frames before reconducting stream density analysis on the front-end system.

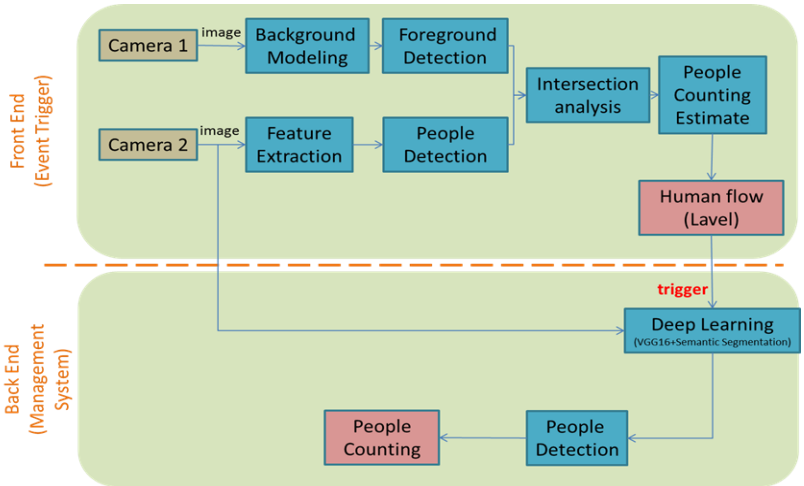


Fig. 1. System flowchart

B. Frond-End

Two cameras are used to capture images at two angles, namely the birds-eye view and side view, as illustrated in Fig. 2. For the birds-eye view, the objects of interest in the image are extracted and the ratios of object areas to the total area are computed. For the side view, pedestrian detection is developed using the classifier algorithm technique. Finally, the two results are combined for pedestrian stream flow classification.

C. Bird's Eye View (Camera1)

This system uses a Gaussian Mixture Model (GMM) to separate the background and foreground, as shown in Fig. 3. Subsequently, the Regions of Interest (ROI) are analyzed; the results are depicted in Fig. 3(c).

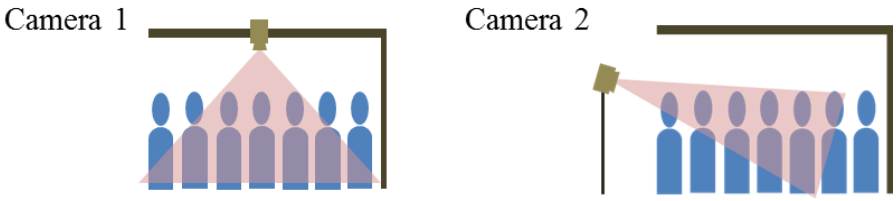
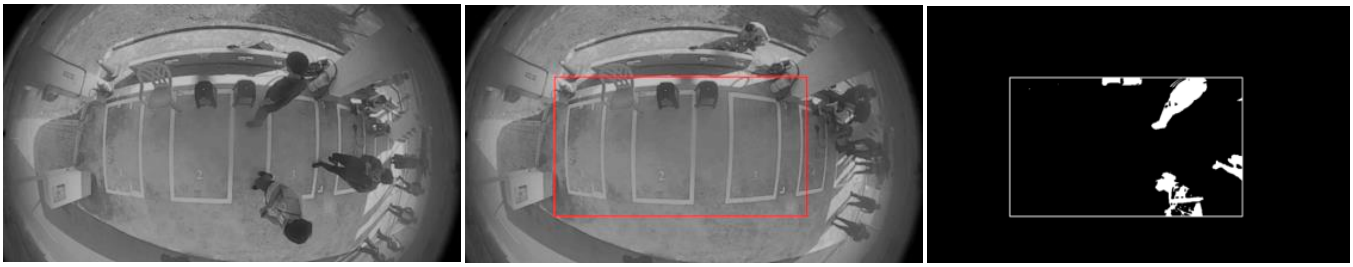


Fig. 2. Camera setup (Camera1: bird's eye view, Camera2: side-view)



(a) Raw image (b) Background image (c) Foreground image

Fig. 3. Background/Foreground image

D. GMM

This study mainly focused on bus stop areas, which are outdoor spaces that are easily influenced by the time of day (daytime, afternoon, and night). Therefore, the GMM [14] was adopted to establish the background, and this method can adjust the background according to the time of day. C number of Gaussian distributions is used to describe the pixel coordinates x with time n and grey scale values $I_{o,x}, I_{1,x}, \ldots, I_{t,x}$. The mixture model consisting of Gaussian distributions at time t can be denoted by

$$P(I_{tx}) = \sum_{k=1}^{c} \omega_{t-1\times x} N\left(I_{t,x}; \mu_{t-1,x}, \sigma^2_{t-1,x,k}\right) \quad (1)$$

$$N\left(I_{t,x}, \mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{(I-\mu)}{2\sigma^2}\right)} \quad (2)$$

Where $N\left(I_{t,x}; \mu_{t-1,x,k}; \sigma^2_{t-1,x,k}\right)$ is the Gaussian probability density function as shown in Equation 2. $\omega_{t-1,x,t}$ is the weight value of the kth Gaussian model, $\mu_{t-1,x,k}$ is the mean value, and $\sigma^2_{t-1,x,k}$ is the standard deviation. The pixel is updated only when it conforms to the background model, and the update methods are represented by Equations 2 and 3.

$$\mu_{t,x,k} = (1-\rho)\mu_{t-1,x,k} + \rho I_{t,x} \quad (3)$$

$$\sigma^2_{t,x,k} = (1-\rho)\sigma^2_{t-1,x,k} + \rho\left(I_{t,x} - \sigma^2_{t-1,x,k}\right) \quad (4)$$

where ρ is the learning rate for determining the mean and standard deviation, and the weight is updated according to whether the pixel conforms to the model, as in shown Equation 5.

$$\omega_{t,x,k} = (1-\alpha)\omega_{t-1,x,k} + \alpha M_{k,t} \quad (5)$$

where α is the learning rate for determining the weight, and $M_{k,t}$ is 1 for the matched model and 0 for the remaining models.

E. Side-Looking(Camera2)

This system adopts the pedestrian classifier algorithm technique and uses the HOG [5] to describe the correlation of pedestrian edges, capture object features, and perform training and classification by using classifiers based on an SVM.

The method for computing direction gradient histogram features involves first computing the image gradient vectors using masks $\text{Mask}_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ and $\text{Mask}_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T$:

$$\begin{aligned} G_x(x,y) &= I(x+1,y) - I(x-1,y) \\ G_y(x,y) &= I(x,y+1) - I(x,y-1) \end{aligned} \tag{6}$$

$$\begin{aligned} G(x,y) &= \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \\ \theta &= \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \end{aligned} \tag{7}$$

where x and y are the pixel coordinates, $G_y(x,y)$ is the vertical gradient vector, $G_x(x,y)$ is the horizontal gradient vector, $G(x,y)$ is the gradient magnitude, and θ is the gradient angle. Next, the cell histograms are examined. A cell comprises 8*8 pixels, and a block comprises 2*2 cells. All gradient angles 0°180°, and the gradient angles scale is 20 degree. The histogram is a vector of 9 bins corresponding to angles. The gradient vector of each cell is documented. Features are computed according to the size of the Sliding Window, and HOG descriptor blocks are established, as illustrated in Fig. 4.

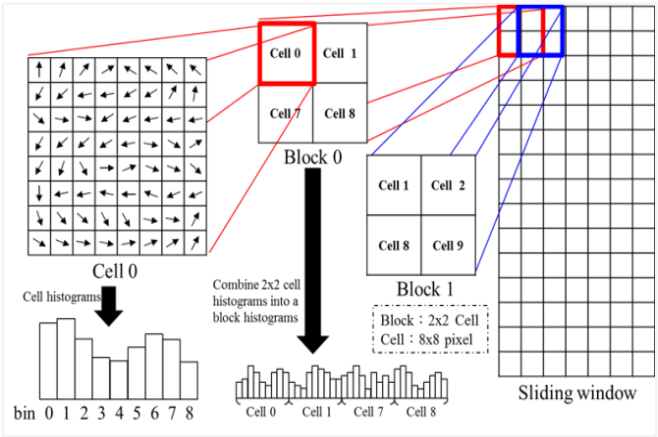


Fig. 4. HOG descriptor block

The SVM mainly divides the learning samples into positive samples and negative samples with a hyperplane, which is given in Equation 8. The distance from any sample point to the plane is set as ≥ 1, as shown in Equation 9.

$$w^T X + b = 0 \tag{8}$$

$$y_i\left(w^T X_i + b\right) = 0 \tag{9}$$

Where w and b are the normal vector and intercept of the hyperplane, respectively. Next, the constraint conditions of w and b for Equation 10 are determined.

The main condition is that the distance of the hyperplane to the positive and negative sample is the maximum.

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} L_i, L_i = \max\left[0, 1 - y_i\left(w^T X_i + b\right)\right] \\ &\text{s.t. } y_i\left(w^T X_i + b\right) \geq 1 - L_i, L_i \geq 0 \end{aligned} \tag{10}$$

Where C is the normalization coefficient. Finally, a set of w and b is obtained, and the feature points can be categorized into positive and negative categories by inputting them into this set of parameters, as given in Equation 11.

$$\begin{aligned} \left(w^T X_i + b\right) &\geq +1, \quad \Rightarrow y_i = +1 \\ \left(w^T X_i + b\right) &\leq -1, \quad \Rightarrow y_i = -1 \end{aligned} \tag{11}$$

F. *Intersection Analysis(Camera1 & Camera2)*

In the same corresponding ROI area of the overhead camera and side camera, the coverage C_1 of the birds-eye view in the ROI area can be computed using the background and foreground separation method. The detected number of people n in the side-view image can be obtained using the pedestrian classifier. Assume that the coverage of one person is v, as shown in Equation 13; the coverage C_2 of the side-view image can then be estimated. Subsequently, the total coverage C is summed according to the weights of different views, as in Equation 12. Finally, the sum is classified for the stream density of people.

$$C = w_1 C_1 + w_2 C_2 \tag{12}$$

$$C_2 = n \bullet v \tag{13}$$

G. *Back-End*

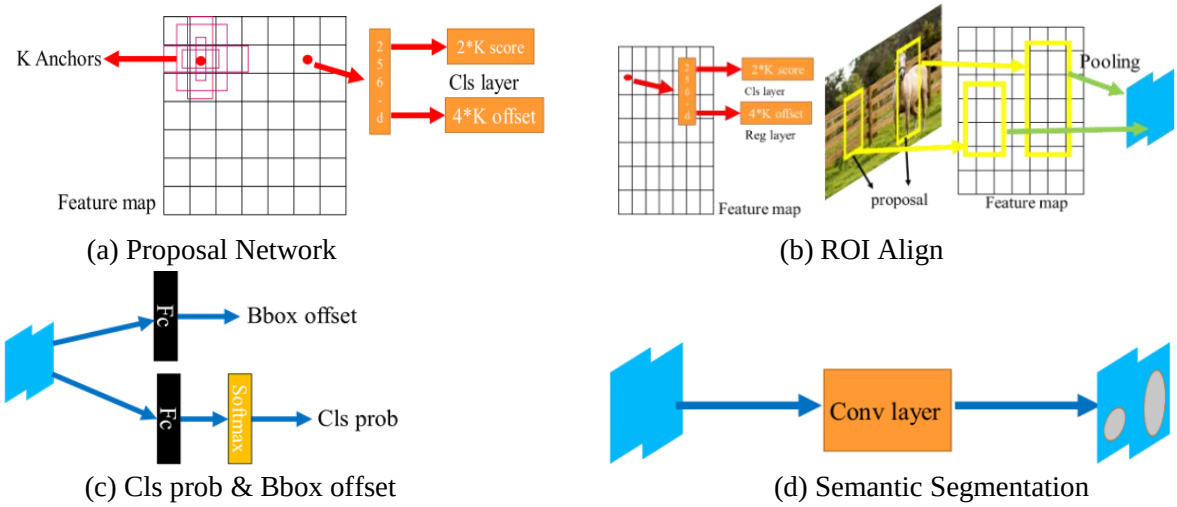
The back end receives classification information from the front end, and then the need for back-end computation is determined. Accurate computation of the number of people is performed mainly by using a deep learning method, and the method applied in the present study was the Mask-RCNN [13].

H. *Mask-RCNN Algorithm*

First, the image features are captured, and then the feature maps are input into the region proposal network to extract obstacle information. Next, information of each obstacle is categorized, and the candidate frames are adjusted slightly. Finally, every candidate frame generates a semantic segmentation

The algorithm steps are as follows:

- Feature capture: ResNet-101 is used to capture features.
- Proposal Network: Feature maps are input into the region proposal network to extract obstacle information. Several K anchors are placed on the cells of feature maps, and each anchor predicts the foreground probability (Cls layer) and bounding box displacement (Reg layer), as shown in Fig. 5(a).
- ROI Align: The images are input back to corresponding predicted positions of proposal frames(Region proposal) and then projected onto the feature maps. Finally, the feature size of candidate bounding boxes is fixed, as shown in Fig. 5(b).
- Categorization and candidate bounding boxes adjustment: A fully connected layer is used to predict every proposal category (Cls prob) and fine-tune the value of the frame (Bbox offset), as shown in Fig. 5(c).
- Semantic Segmentation: Convolution layers are used for semantic segmentation (1 or 2 convolution layers) and are categorized into the foreground and background, as shown in Fig. 5(d)



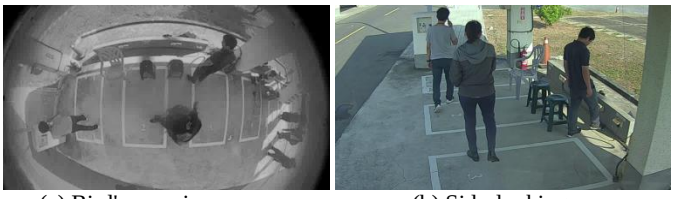
(a) Proposal Network (b) ROI Align

(c) Cls prob & Bbox offset (d) Semantic Segmentation

Fig. 5. The mask-RCNN algorithm [13]

II. RESULTS AND DISCUSSION

The resolution of the side-view camera used in testing and the resolution of the birds-eye view camera were both 1280*720; the images are presented in Fig. 6. Details of the computation platform adopted are provided in Table 1. For the back end, the performance of the computation platform was higher because of the larger computation capability of the deep learning method.



(a) Bird's eye view (b) Side-looking

Fig. 6. Difference visual angle of image

TABLE 1
FROND-END & BACK-END HARDWARE PERFORMANCE

	Frond-End	Back-End
GPU	NVIDIA GeForce GTX 1050	NVIDIA GeForce GTX 1660Ti
CPU	Intel Core i7-7700HQ 2.80GHz	Intel Core i7-8700 3.20GHz
Memory	8GB	64GB

A. *Effectiveness Analysis*

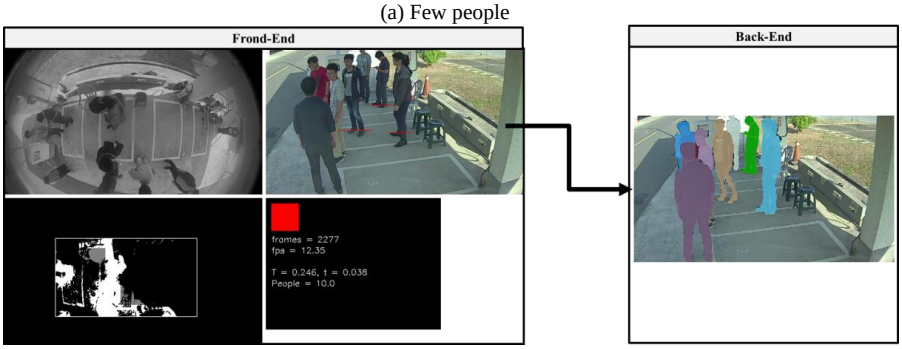
A total of three videos and 5581 images were tested, and the videos contained different numbers of people. A comparison of execution speed was conducted between the method proposed in this paper and the method using only the Mask-RCNN, and the results are provided in Table 2. On average, the proposed method can reduce 29% execution time when comparing with Mask-RCNN. The execution time of the proposed method when processing images with relatively few people was approximately 0.1 seconds. The execution time for images with more people was longer, at approximately 0.27 seconds due to time required for triggering the processing of the back end according to the information sent from the front end. After the trigger, several frames took approximately 0.2 seconds to process. However, our trigger cost performance is as same as the Mask-RCNN.

Front-end pedestrian flow is separated into two levels: low (zero to four people) and high (more than four people). The front-end detection results are presented in Fig. 7. The correct pedestrian flow ratio was computed with 2636 images, and the accuracy rate was 84.56% (items with correct flow classification divided by the total number of items). The detection results for back-end deep learning are shown in Fig. 7(b). The performance of the back-end number-counting system was analyzed with 2636 images. The accuracy rate was 93.59% and the error rate was 1.53% (number of misjudged items divided by the total actual number).

TABLE 2
OPERATION TIME

	Video 1	Video 2	Video 3	Total
Number of images (frame)	2636	2218	727	5581
The proposed method (second)	359.6	319.4	94.8	773.8
Mask-RCNN (second)	503.4	431.3	145.1	1079.8
Saving(%)	29%	26%	35%	29%

(a) Few people



(b) More people

Fig. 7. Experimental results

III. CONCLUSION AND RECOMMENDATIONS

The proposed system is designed based on two-stage architecture which contains previous two methods in front-end and back-end. The front end classifies the flow of people into two levels by using images captured from two angles. The back-end system, which is used only when the flow reaches level 2, determines the number of people by using the Mask-RCNN. The experimental results indicated that the front-end classification had an accuracy of 84.56% and that back-end counting achieved an accuracy of 93.59%. A comparison of the method proposed in this paper with the detection method using only the Mask-RCNN revealed that the proposed method had a 29% lower execution time. Bus shelters have off-peak periods and peak periods; accordingly, the proposed method adjusted its computation process according to the number of people counted to effectively reduce the execution time. In the future, multiple front-end systems can be integrated with a back-end system, and the front-end systems can be integrated into an embedded platform to effectively reduce cost.

Declaration fo Conflicting Interests

This has no financial or non-financial competing interests.

ACKNOWLEDGEMENT

This study is part of the results of the TDP for Nonprofit Research Organization plan (No. 108-EC-17-A-25-1588) of the Automotive Research & Testing Center (ARTC).

REFERENCES

[1] C. Hong, J. Yu, J. Wan, D. Tao, and M. Wang, "Multimodal deep autoencoder for human pose recovery," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5659–5670, 2015. doi: https://doi.org/10.1109/tip.2015.2487860

[2] R. Suryanita, H. Maizir, and H. Jingga, "Prediction of structural response based on ground acceleration using artificial neural networks," *International Journal of Technology and Engineering Studies*, vol. 3, no. 2, pp. 74–83, 2017. doi: https://doi.org/10.20469/ijtes.3.40005-2

[3] N. Bernini, L. Bombini, M. Buzzoni, P. Cerri, and P. Grisleri, "An embedded system for counting passengers in public transportation vehicles," in *IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, Senigallia, Italy, 2014. doi: https://doi.org/10.1109/mesa.2014.6935562

[4] J. Jeon, D.-H. Kim, B. Choi, G. Kim, and Y.-S. Kim, "A construction of vehicle image and ground truth database for developing vehicle maker and model recognitions," *International Journal of Technology and Engineering Studies*, vol. 3, no. 6, pp. 229–235, 2017. doi: https://doi.org/10.20469/ijtes.3.40002-6

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, 2005.

[6] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proceedings of 12th International Conference on Pattern Recognition*, Jerusalem, Israel, 1994. doi: https://doi.org/10.1109/icpr.1994.576366

[7] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *IEEE 12th International Conference on Computer Vision*, Kyoto, Japan, 2009. doi: https://doi.org/10.1109/iccv.2009.5459207

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016.

[9] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019. doi: https://doi.org/10.1109/TNNLS.2018.2876865

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014. doi: https://doi.org/10.1109/cvpr.2014.81

[11] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, Las Condes, Chile, 2015. doi: https://doi.org/10.1109/iccv.2015.169

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, Montreal, Canada, 2015.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 2017.

[14] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, Fort Collins, CO, 1999. doi: https://doi.org/10.1109/cvpr.1999.784637