ISSN: 0067-2904

Robust Pedestrian Detection and Tracking Using CNN and SORT Algorithms

Tuqa Hani Abd-Alamir*, Mohammed Sadoon Hathal

Department of Computer Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq.

Received: 19/9/2023 Accepted: 16/2/2024 Published: 28/2/2025

Abstract

In recent years, there has been a growing popularity of autonomous vehicles due to their significant impact on society. One of the key tasks of autonomous vehicles is accurate pedestrian detection, which plays a vital role in preventing accidents. However, accurately detecting and tracking pedestrians under various environmental circumstances poses a significant challenge. In this paper, an efficient model for pedestrian detection is proposed by integrating three modules: You Only Look Once version 8 (YOLOv8) for object segmentation, Histogram of Oriented Gradients (HOG) for feature extraction, and Custom Convolutional Neural Network (CNN) for classification and detection. For tracking purposes, a simple online and real-time tracking (SORT) algorithm is used to track pedestrians in consecutive frames. Extensive experiments were conducted using the EPFL dataset. By leveraging the strengths of these modules, the proposed model aims to improve the accuracy and performance of pedestrian detection and tracking. The experimental results demonstrate the remarkable capability of the suggested model to detect and track pedestrians, achieving an accuracy rate of 93.34% even in challenging weather scenes.

Keywords: Pedestrian detection, Pedestrian tracking, HOG, YOLO, CNN.

الكشف والتتبع المتين للمشاة باستعمال خوارزميات *CNN* و *SORT*

تقى هاني عبد الأمير* , محمد سعدون حثيل

قسم هندسة الحاسبات, كلية الهندسة, جامعة بغداد, بغداد, العراق

الخلاصة

في السنوات الأخيرة ، كانت هناك شعبية متزايدة للمركبات ذاتية القيادة نظرًا لتأثيرها الكبير على المجتمع. تتمثل إحدى المهام الرئيسية في المركبات ذاتية القيادة في الكشف الدقيق عن المشاة ، والذي يلعب دورًا حيويًا في منع وقوع الحوادث. ومع ذلك ، فإن الكشف الدقيق عن المشاة وتتبعهم في الظروف البيئية المختلفة يشكل تحديًا كبيرًا. في هذا البحث ، تم اقتراح نموذج كفوء لاكتشاف المشاة من خلال دمج ثلاث وحدات: خوارزمية YOLOv8 لتجزئة الكائن ، خوارزمية HOG لاستخراج الميزات ، والشبكة العصبية التلافيفية المخصصة CNN للتصنيف والكشف. لأغراض التتبع ، يتم استعمال خوارزمية SORT لتتبع المشاة في إطارات متتالية. أجريت تجارب مكثفة باستعمال مجموعة بيانات EPFL. من خلال الاستفادة من نقاط القوة في هذه الوحدات ، يهدف النموذج المقترح إلى تحسين دقة وأداء اكتشاف المشاة وتتبعهم. توضح النتائج التجريبية القدرة الجيدة

*Email: tuqa.hani2105m@coeng.uobaghdad.edu.iq

للنموذج المقترح لاكتشاف وتتبع المشاة ، وتحقيق معدل دقة يبلغ 93.34٪ ، في ظل تحديات ظروف الطقس
المختلفة.

1. Introduction

The task of pedestrian detection in videos and images involves identifying and localizing pedestrians within a scene. In contrast, pedestrian tracking of video sequences includes keeping track of the spatial and temporal movements of pedestrians as they traverse through frames [1].

Pedestrian detection and tracking have gained significant attention as essential research topics due to their widespread applications in the field of computer vision [2]. These applications encompass various domains, such as driver assistance systems, where the goal is to enhance pedestrian safety as well as monitor human activities in both indoor and outdoor environments [3]. Additionally, pedestrian detection and tracking contribute to security surveillance, facilitate efficient operations at bus stops and shopping centers [4], enable the monitoring of human-robot interactions, etc. [5].

Automated systems for pedestrian detection and tracking surpass human capabilities, particularly in scenarios requiring extensive monitoring. Human operators often face challenges in effectively managing controlled areas, especially when multiple cameras are involved. Moreover, in medical applications, analyzing videos captured by devices can be a complex task [6]. Therefore, the development of automated systems that ensure efficient and accurate pedestrian detection and tracking becomes essential.

The task of pedestrian detection and tracking poses significant challenges due to the diverse heights and body shapes of people. When two pedestrians have identical height, body shape, and similar clothing, it becomes challenging for a computer to differentiate between them accurately. Additionally, changes in illumination within the scene can further complicate the pedestrian appearance, making it difficult to design an effective real-time algorithm that addresses these issues [7]. In the proposed model, You Only Look Once version 8 (YOLOv8) was utilized for object segmentation, and the Histogram of Oriented Gradients (HOG) was employed as a feature descriptor to capture the shape information of pedestrians. Moreover, a custom Convolutional Neural Network (CNN) model was designed to successfully categorize pedestrians based on their unique features. In tracking, the Simple Online Real-Time Tracking (SORT) algorithm was applied, allowing the efficient tracking of multiple pedestrians in the same captured video frame. By combining the strengths of traditional computer vision techniques and deep learning (DL) algorithms, the proposed model aims to enhance the accuracy of pedestrian detection and tracking systems, taking into consideration the complexities and challenges discussed earlier.

2. Related works

Over the past few decades, numerous academics have presented research on pedestrian detection and tracking. These studies vary in computational complexity: feature descriptors, classification schemes, detection algorithms, and tracking algorithms. Traditional techniques were used by some researchers, while DL was utilized by others. In [1], they proposed a method aimed at enhancing the processing time of HOG feature extraction. Their approach used two classifiers: support vector machines (SVM) and boosted classifiers, and they used a Kalman filter (KF) for tracking in both classifiers. Through experimental evaluations, the second approach has shown promising results in terms of speed and accuracy. The paper [2] projected a model that combines cluster segmentation, Gabor feature extraction, and an SVM classifier for person detection. While HOG feature descriptor approaches are used for action recognition, for tracking, they use temporal information. The experiment results proved the superiority of the proposed methods, with an accuracy rate of 89.59%. The detection model in [3] was done by integrating a set of modules including a combination of aggregate channel features (ACF) for proposing regions of interest (ROIs) and a deep CNN as the classifier. For tracking, the model incorporates the Nearest Neighborhood Probabilistic Joint Data Association (NNJPDA) and KF techniques. The experimental results illustrate the robustness of the proposed model in applications such as robotic navigation. The paper [8] presents a new framework for pedestrian detection using a moving camera. Motion vectors are extracted from the ROI using the Block Matching Algorithm (BMA). The adaptive threshold value is used to determine the foreground and background. Features are extracted using the LBP descriptor, and then SVM is used for the classification task. The results prove that the proposed model has higher accuracy, but it increased the time slightly. The study [9] presented a comparative analysis of complete occlusion in pedestrian detection and tracking systems. For detection, they compared two approaches: HOG-SVM and CNN. The video used in this work was captured in a noisy environment, and some frames exhibited complete occlusion. The results showed that in the case of noise and complete occlusion, CNN gives better results. However, in the absence of occlusion, HOG-SVM achieves better results. For tracking, three types of filters are compared, represented by KF, particle filter (PF), and a proposed hybrid Kalman-Particle Filter (KPF). The findings show that using the KF filter with CNN provides more accurate results, considering all cases. The proposed KPF filter gives a similar result to the KF filter but is more time-consuming due to intensive calculations. The work presented in [4] successfully improved the Darknet53 backbone of YOLOv3 by introducing the Convolutional Block Attention Module (CBAM) for pedestrian detection and developing the DeepSort method for pedestrian tracking. Their work produced enhanced accuracy in training and testing, along with improved processing time.

After reviewing some of the published research on pedestrian detection and tracking, it is evident that the current algorithms focus on improving their effectiveness. Nonetheless, creating an algorithm that achieves high accuracy while minimizing computational cost is a challenging task.

3. The proposed system

The proposed model aims to address the challenges of pedestrian detection and tracking under different weather conditions by combining traditional techniques with DL approaches. Figure 1 illustrates the key steps involved in the proposed system, which consists of two main parts: detection and tracking. The detection part incorporates Yolov8 for object segmentation, HOG for feature extraction, and CNN for classification. Conversely, the tracking part was implemented using the SORT algorithm. A comprehensive explanation will be presented in the subsequent sections.

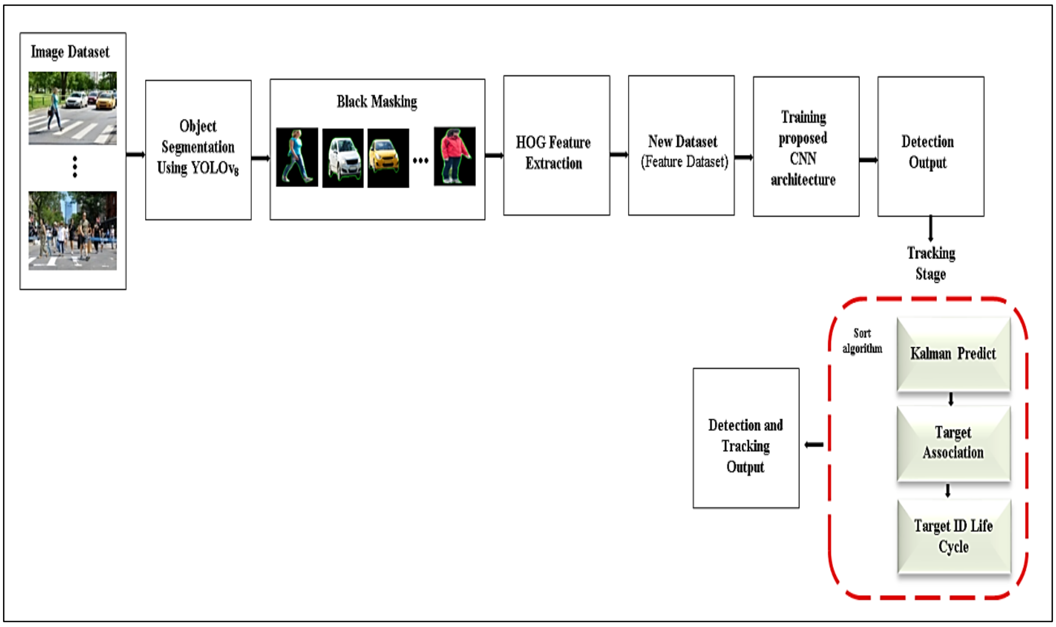


Figure 1: The block diagram of the proposed model

4. Segmentation

Image segmentation refers to the process of dividing an image into regions based on certain similarities, such as color, texture, or shape. It can also involve identifying semantically meaningful parts within an image. Although various algorithms exist for image segmentation, no single algorithm can universally perform well on all types of images, and the effectiveness of an algorithm can vary depending on the specific image being processed. Therefore, image segmentation remains a challenging task in the fields of image processing and computer vision and continues to be an unresolved problem [5].

In this presented model, YOLOv8 is utilized for object segmentation. YOLOv8 is a single neural network that has gained significant popularity in computer vision due to its impressive speed and accuracy [6]. The output of the segmentation model consists of masks, or contours, outlining each object in the image. Each object class is assigned a specific label; for example, label 0 is assigned to the person class, while label 1 is assigned to the car class [7]. Figure 2 demonstrates an instance of utilizing YOLOv8 for object segmentation in the suggested model. The procedure of object segmentation is summarized in the following points:

i. Installing libraries for the YOLOv8 segmentation algorithm.

ii. Reading the dataset images used in the proposed model .

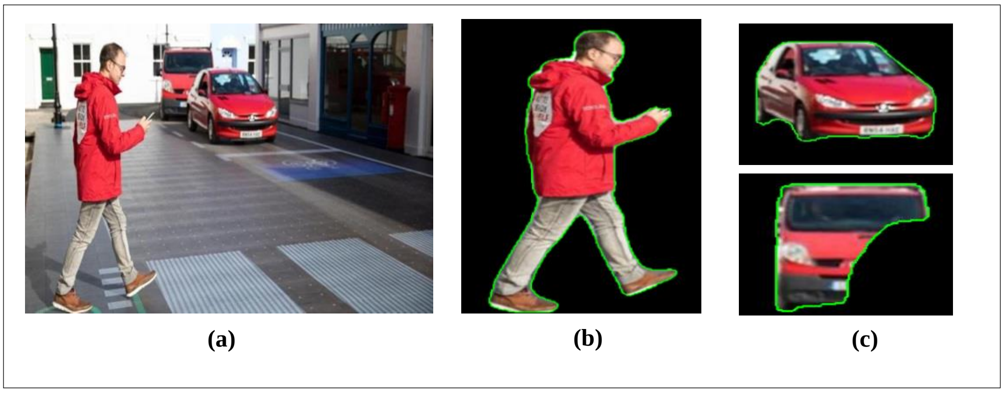
iii. For each image, the YOLOv8 method returns two parameters:

  a. Class_id :Indicates the class or label of the segmented object.

  b. Segmentation: contains the contour information of the segmented object.

iv. Applying a black mask to isolate the ROI within the image by placing 0 (black color) on every pixel that does not contain an object based on the information stored in the segmentation parameter.

After object segmentation, the segmented objects undergo preprocessing steps, including RGB to grayscale conversion, to reduce computational complexity due to the fact that a grayscale image uses 8 bits to describe each pixel while an RGB color uses 24 bits [10]. After that, resize it to ensure a uniform size for the next steps.



(a) **(b)** **(c)**

Figure 2: Segmentation results (a) input image (b) positive objects (pedestrians) (c) negative objects (non-pedestrians)

5. Feature extraction

A feature descriptor refers to an algorithm that produces a feature array or vector from an input image [11]. The primary objective of a feature descriptor is to simplify the image into a numerical representation by extracting relevant information and disregarding irrelevant data [12]. Thus, the feature descriptor can differentiate between different objects according to their unique properties. Types of feature descriptors include SIFT, BRIEF, FAST, ORB, HOG, and so on [13].

The HOG descriptor, where employed in the proposed model, is a popular and extensively used technique in computer vision and image processing applications, particularly for object detection [9]. Algorithm 1 presents the primary steps of the HOG technique, as shown in Figure 3. The procedure for the feature extraction phase can be summarized as follows:

i. Resizing the segmented object to (64×128) pixels in order to match the origin HOG algorithm.

ii. Convert the resized image into gray scale image.

iii. Computing HOG features, the output is a feature vector of shape (1×3780) for each image.

iv. Storing the features according to class_id as follow:

a. If the Class_id value is 0, it means that the image contains pedestrians, and the feature is stored as follow:

- Positive_features :Stores features of pedestrian objects.
- Positive_Labels :Store label "Pedestrian".

b. Otherwise, the image has no pedestrians and the feature is stored as follow:

- Negative_features:Stores features for non-pedestrian objects.
- Negative_Labels :Stores label "Non-Pedestrian".

Following the procedure mentioned above, the positive and negative features, as well as their related labels, are saved as an array ($\text{DB}_{\text{Features}}$), which will later be utilized to train the classifier. Figure 4 describes the results obtained after implementing the HOG algorithm.

Algorithm 1 HOG Feature Extraction

```
Input: Segmented object image
Output: Feature vector array representing the object

Procedure
    ▪ Divide the image into cells and blocks (2×2) cells, with each cell is (8×8) pixels.
    ▪ For each pixel, compute the horizontal ($G_X$) and vertical ($G_y$) gradients using ($D_x$) and ( $D_y$)
      filter mask to approximate image intensity changes, as presented in following equation :
        $G_X = D_X * I$ ……. (1)
        $G_y = D_y * I$ ……. (2)
        $D_x = [-1 \ 0 \ 1] , D_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$ ……. (3)
    ▪ Calculate the gradient magnitude ($|G|$)and orientation ($\theta$) for each pixel:
        $|G| = \sqrt{{G_X}^2 + {G_y}^2}$ ……. (4)
        $\theta = tan^{-1} \frac{G_y}{G_X}$ ……. (5)
    ▪ For each cell, accumulate gradient orientations into a histogram of orientations.
    ▪ Normalize histograms within each block to enhance robustness to lighting variations and
      contrast changes.
    ▪ Concatenate the normalized block histograms to form the final feature vector.
End Procedure
```

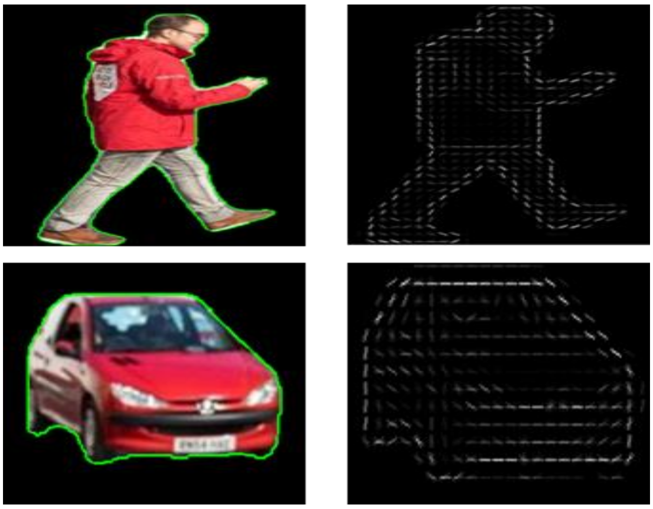


Figure 3: The main steps of the HOG algorithm

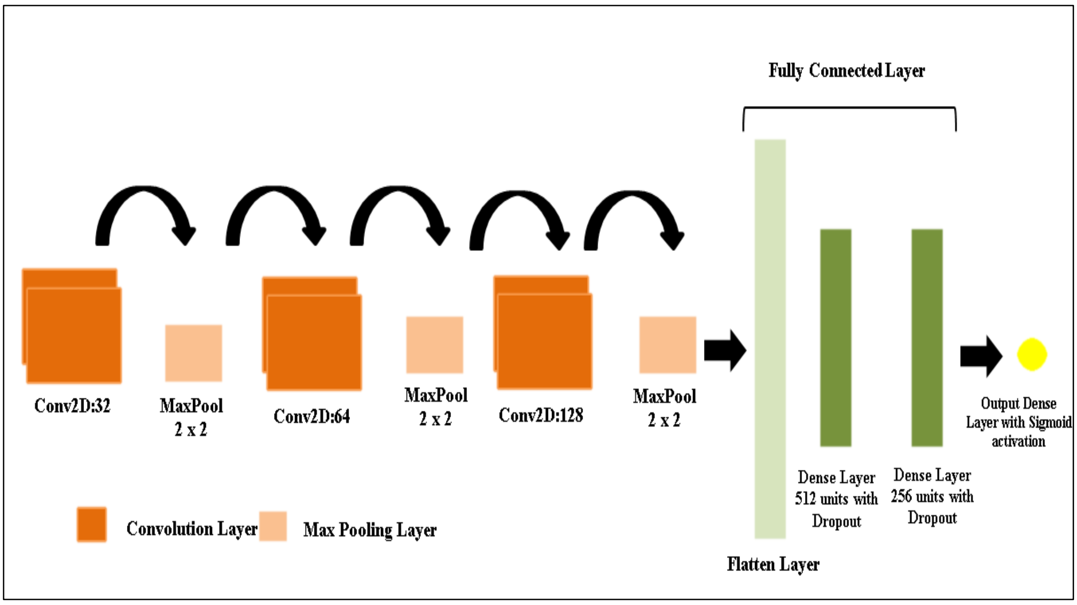


Figure 5: Proposed CNN Architecture

6. Classification

Image classification refers to the process of assigning class labels to pixels or regions within a digital image. Its objective is to determine the category to which an image belongs and identify the presence or absence of specific objects. Pedestrian detection, for example, is a binary image (indicating the presence of a pedestrian, positives) or (0 or -1) (indicating the absence of a pedestrian, negatives) [14]. In this study, CNN was employed for the classification and detection of pedestrians. CNN is a type of DL architecture specifically designed for analyzing visual data. CNNs operate similarly regardless of whether they are 1D, 2D, or 3D, with a difference in the input data structure and the movement of the feature detector or convolutional kernel over the data [15]. One advantage of CNNs is their ability to automatically learn and extract relevant features from the data, which often reduces the need for extensive pre-processing compared to traditional image classification algorithms [13]. Figure 5 illustrates the proposed architecture of CNN. The input is subjected to three sets of convolutional operations, accompanied by Rectified Linear Unit (ReLU) activation functions and max-pooling layers to enhance classification accuracy. After that, the output is subjected to non-maximum suppression (NMS) algorithms, which aid in selecting the most precise boundary boxes. During the training process of a CNN, the error or loss is computed based on the difference between the predicted output and the actual output. This error is used to update and optimize the trainable parameters of the network, which include filter values and neuron weights.

The procedure for using CNN as a classifier in the proposed model is explained as follows:

i. Installing the libraries required to build and train the CNN model.

ii. Load $DB_{Features}$ and their corresponding labels.

iii. Dividing $DB_{Features}$ into subsets for training and testing using a ratio of 80% and 20%, respectively.

iv. In order to ensure compatibility with the accepted input format for the CNN, the training and testing features were transformed into a 2D array.

v. Encoding the labels, which are converted to numeric values, where 0 represents the pedestrian label and 1 represents the non-pedestrian label. This conversion is making the model easier to understand.

vi. Prior to model training, key parameters such as the optimizer (Adam), loss function (binary cross-entropy function), learning rate (0.0001), number of epochs (100), and batch size (32) were selected after performing several experiments. Next, the training process started, with a total of (2182) iterations. Figure 6 depicts the training process, including loss and accuracy. It is clear from the figure that the model is operating effectively throughout training. The error lowers steadily as the number of epochs increases, indicating that the model is improving and making accurate predictions.

vii. After completing the training process, the model was saved for subsequent use during the testing stage.

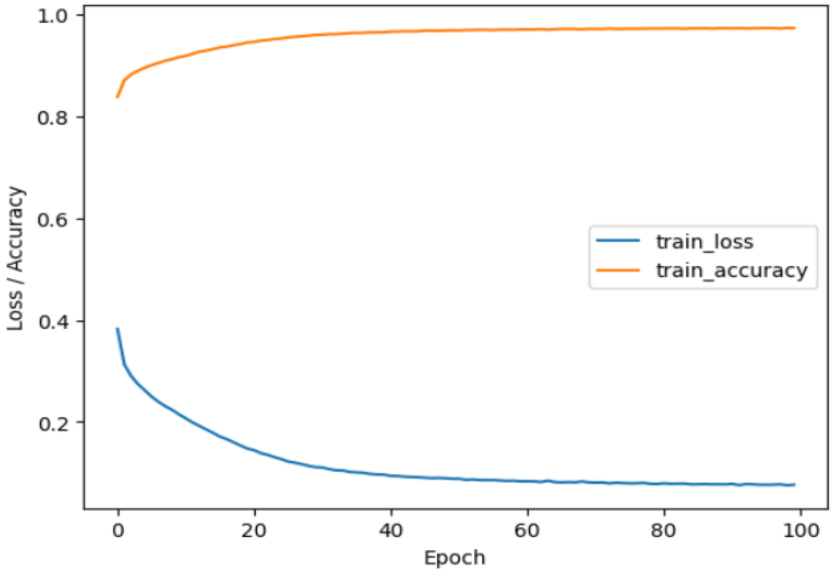


Figure 6: Graph of training accuracy and loss of the model

7. Tracking

After the detection process, the output consists of bounding boxes that enclose the detected pedestrians. The tracking algorithm utilizes the information from the bounding boxes to track the movement and trajectory of the pedestrians over time. By using the bounding boxes as references, the tracking algorithm can associate the detected pedestrians across consecutive frames and maintain their identities throughout the video or image sequence [2]. The proposed model utilized the SORT algorithm [16], which is a well-known algorithm used for online pedestrian tracking [17]. It comprises two key components: a KF for estimating object states and the Hungarian algorithm for associating the predictions from the KF with newly detected objects [18]. Figure 7 illustrates the fundamental structure of the SORT algorithm.

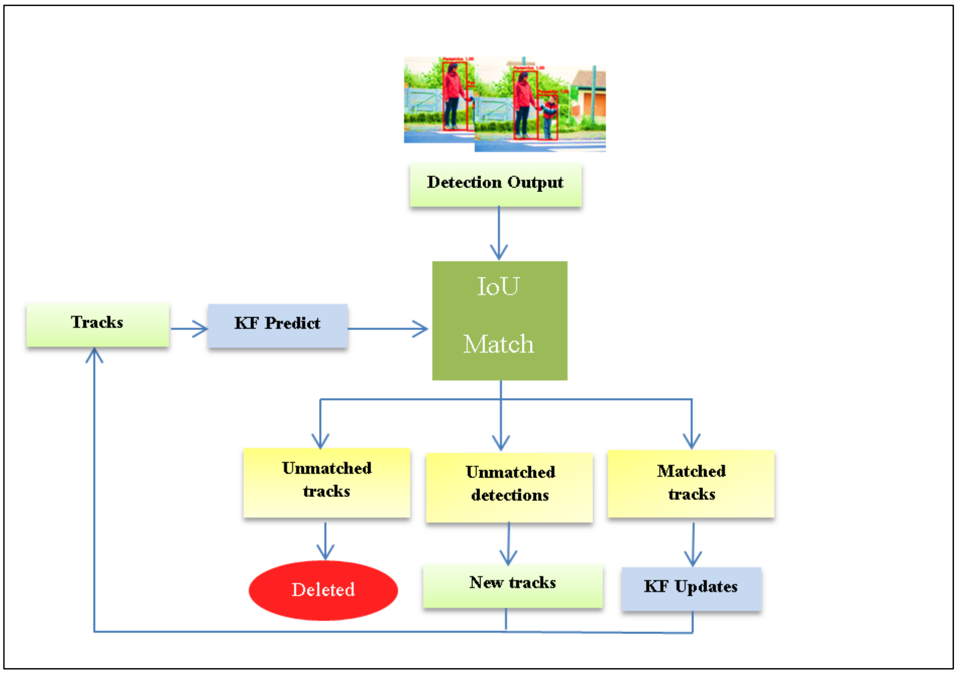


Figure 7: The structure of the SORT algorithm [19]

The Data Association module is responsible for matching predicted KF bounding boxes with measured object detector-bounding boxes [19]. This is achieved by formulating a linear assignment problem and computing a cost matrix using the Intersection over Union (IOU) as a metric, Eq. (6). The Hungarian algorithm is applied to associate the bounding boxes based on the cost matrix. Any associations with an IOU lower than a specified threshold are discarded [18].

$$iou(D_i\,,P_i) = \frac{D_i\ \cap\ P_i}{D_i\ \cup\ P_i}$$

Here, IoU $(D_i\,,P_i)$ represents the intersection over union between a detected bounding box $(D_i\,)$ and a predicted bounding box $(\,P_i\,)$ [18].
Track creation and deletion are handled by the Track Management module. It involves three key steps. Firstly, the track's state is updated with the corresponding detection when a successful match occurs. Secondly, when detections do not overlap with existing tracks below a minimal IOU threshold, new tracks are formed. Lastly, tracks that do not receive updates or associations are removed in order to prevent maintaining a large number of tracks for false positives or objects that have left the place [20]. The procedure of the SORT algorithm is summarized as follows:

i. Importing the libraries necessary for the SORT algorithm.

ii. The input of the SORT algorithm is a list of boundary boxes representing the detected pedestrians, this boundary box includes information about the position and size of the pedestrian.

iii. Each frame fed into the SORT algorithm is checked to see whether it contains bounding boxes or not. It is ignored and moved to the second frame if it has no bounding boxes.

iv. The initial frame is called a “track” frame and represents the pedestrian’s path over time. It stores information about the pedestrian’s current state, predicted state, and unique ID. Therefore, it is used to determine the starting state of tracks. As the pedestrian moves and the video progresses, the “track frame” information is updated depending on the information of the most recent frame.

v. Based on the position and velocity information of the track frame, the SORT algorithm predicts where pedestrians will be in the second frame.

vi. Data association for matching the prediction position of the current frame P_i, newly detected boundary box D_i is applied using the Hungarian algorithm, which uses the IOU threshold to perform this association. In this work, the threshold values were determined in the experiment as follows:

  a. If IOU (P_i , D_i) is greater than (0.5), update the Track frame information with the new measurement.

  b. If IOU (P_i , D_i) falls within (0.3 - 0.5), create a new track using the detected pedestrian information.

  c. If IOU (P_i , D_i) is less than (0.3), remove the tracks.

vii. The tracking result is displayed for each frame by drawing a boundary box for the pedestrian with a unique ID.

viii. Continue repeating this process for each frame until the end of the video.

8. Dataset

The experiments are implemented using the EPFL dataset[1] which includes a set of videos with diverse poses and illumination as well as indoor and outdoor sequences. Therefore, this dataset is a good selection for evaluating the performance of the suggested model. The EPFL dataset comprises outdoor sequences with six videos for the campus scene and four videos for the passageway scene. Initially, the videos were converted into individual frames, resulting in a total of 50,000 images. Hence, these images were processed using YOLOv8 for object segmentation. Each segmented object was further analyzed using the HOG descriptor, generating a new feature dataset consisting of 87,260 object features. This dataset was separated into 56,710 positive features and 30,550 negative features, each of size 64 x 128. The generated dataset is grouped for training and testing at 80% and 20%, respectively. The training dataset contained 69,808 data samples, while the testing dataset contained 17,452 data samples. The summary of the EPFL dataset is shown in Table 1. Figure 8 represents images of samples of the EPFL dataset.

Table 1: The EPFL dataset

Total Sample	Positive sample	Negative Sample	Training Sample	Testing Sample
87,260	56,710	30,550	69,808	17,452

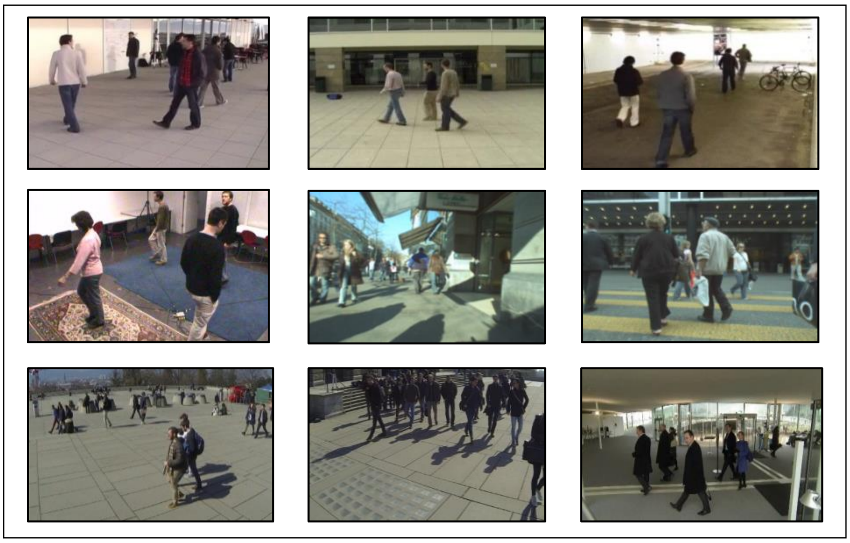


Figure 8: Samples of EPFL pedestrian dataset

9. Experimental results

The experimental environment of the simulation for this system is done using the Python language in the Google Colab environment, an online cloud platform designed for high-performance computation and training DL models.

Different metrics are employed to assess the performance of the detection and tracking models**.** In the detection task, the performance of the proposed model is evaluated using different performance measures: precision, recall, F1 score, and accuracy. These measures are calculated based on the following equations [21]:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1 = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} \tag{9}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

Where TP represents true positives, FP represents false positives, FN represents false negatives, and TN represents true negatives [22], in the tracking task, the performance model is assessed using two metrics: multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP). The calculation formula is as follows:

$$MOTA = 1 - \left[\frac{\sum_t (FN_t + FP_t + IDs_{t)}}{\sum_t GT_t}\right] \tag{11}$$

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t C_t} \tag{12}$$

Here, IDs_t represents the number of times the object ID has changed during tracking. GT is the ground truth, which includes the correct object locations. d_t^i denotes the distance between the predicted object location and the corresponding ith ground truth object location in t frame.C_t is the total number of matches in t frames between the ground truth and the detection out

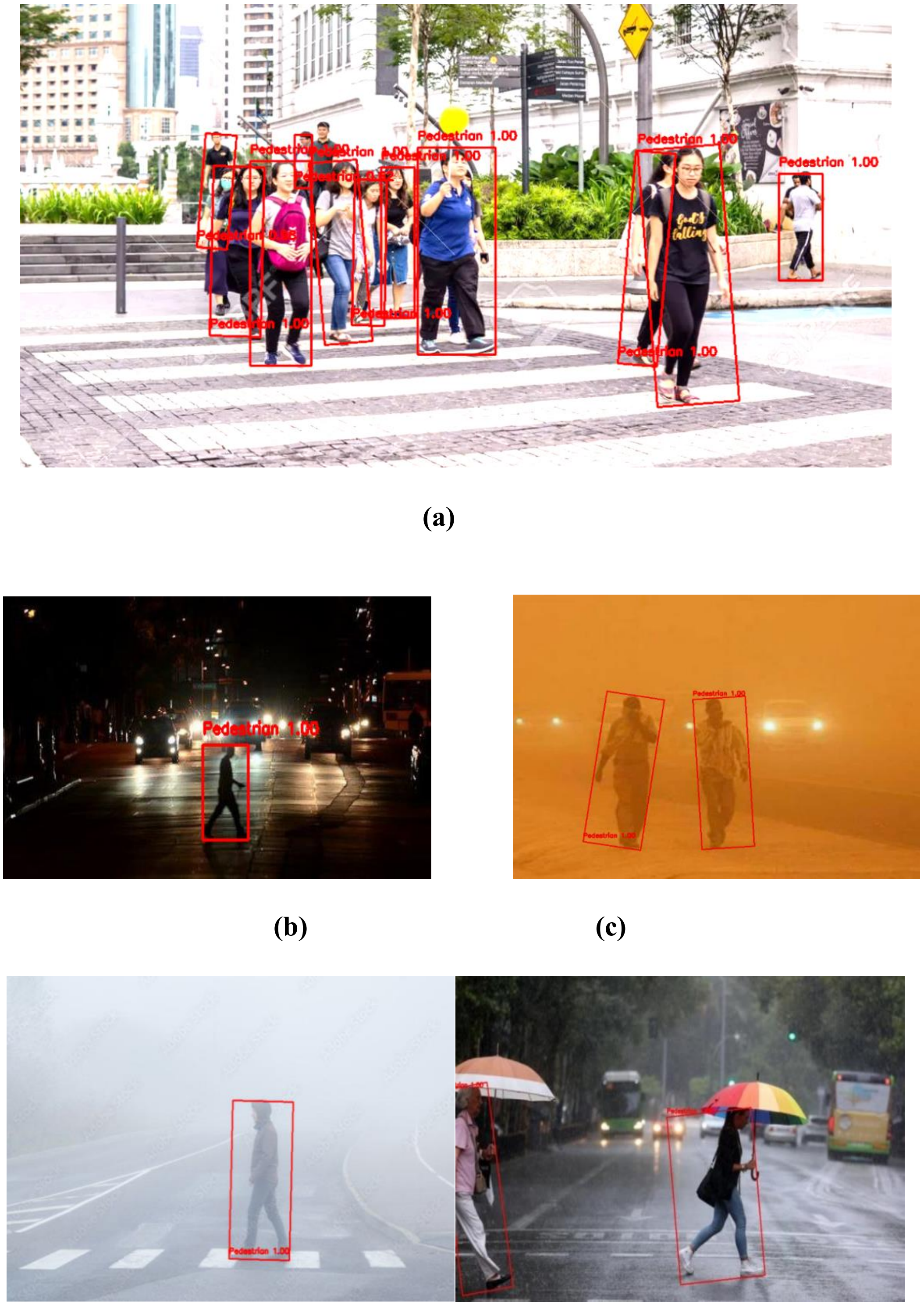
put [4].

The outcomes of the developed system are reported in Table 2. In Table 2, the symbol ↑ means that a higher value is preferable, whereas the symbol ↓ indicates that a lower value is desirable.

Table 2: Result of the proposed pedestrian detection and tracking model

Precision ↑	Recall ↑	F1-score ↑	Accuracy ↑	MOTA ↑	MOTP ↑	Detection Speed(s) ↓
93.97	95.97	94.32	93.34	88.96	89.77	0.42

Figure 9 presents the experimental results of the proposed model conducted under various weather conditions. The results demonstrate the effectiveness of the suggested model in achieving good performance across different environmental circumstances.



(a)

(b) **(c)**

(d) **(e)**

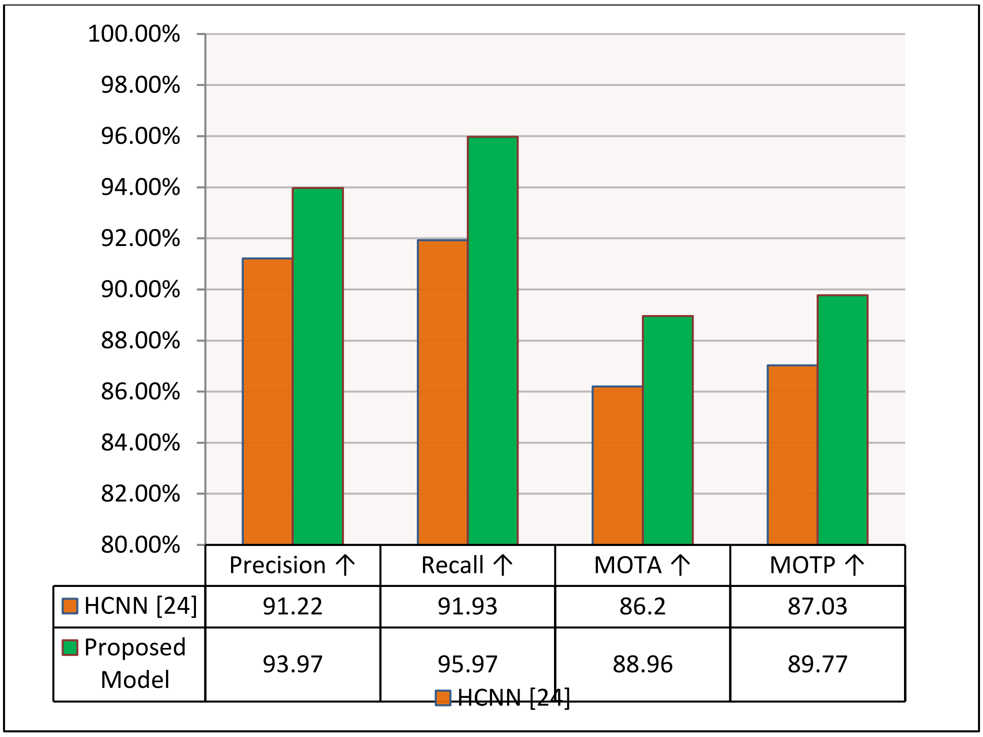
Figure 9: Results of the proposed model in different weather conditions: (a) during the day; (b) at night; (c) dusty; (d) rainy; and (e) foggy

10. Discussion

Careful consideration should be given when comparing the performance of different methods. Several parameters can differ between experiments and have a significant impact on performance, including the selection of the learning dataset, the resolution of the input images, and the threshold value [23]. Taking these variables into account is critical to ensuring fair and accurate comparisons between different methods.

In order to evaluate the performance of the proposed model, we performed a comparison with the method described in [24] using different metrics, including precision, recall, MOTA, and MOTP. The evaluation results are shown in Figure 10.

The F1-score is defined as the harmonic mean of precision and recall, providing a more comprehensive assessment of model performance because precision and recall are inversely related, meaning that increased precision often leads to decreased recall and vice versa. Table 3 summarizes the comparisons between the proposed model with Faster RCNN [25], YOLOv3 [26], and Efficient Detection [27], re-training on the EPFL dataset [28] based on the F1-score value.



	Precision ↑	Recall ↑	MOTA ↑	MOTP ↑
🟧 HCNN [24]	91.22	91.93	86.2	87.03
🟩 Proposed Model	93.97	95.97	88.96	89.77

Figure 10: Comparison results of the suggested model with the existing method

Table 3: Comparison of state-of-the-art methods based on the F1-Score value Using the EPFL dataset

Methods	F1-Score %
Faster RCNN [25]	90.0
YOLOv3 [26]	89.0
EfficientDet [27]	89.0
HCNN [24]	91.8
Proposed model	**94.32**

The results demonstrate that the proposed method outperformed other approaches in terms of performance. It has been observed that the model has a higher precision value, meaning that the proposed model is effective in correctly identifying pedestrians with fewer incorrect classifications. Moreover, the suggested model has a high recall value, which means it is capable of correctly identifying a large portion of relevant instances (TP) from the total number of actual positive instances (TP + FN). In other words, the high recall value indicates that the model has a low tendency to miss positive instances (low FN). Furthermore, the proposed model outperforms comparative methods based on the F1 score. The F1 score value of the proposed model indicates well-balanced model performance, considering precision and recall. Additionally, the tracking results indicate that the proposed model can track pedestrians accurately compared to [24]. The outstanding superiority of the proposed model can be attributed to the hybrid approaches that mix traditional techniques and DL methods. While the proposed model achieves satisfactory results, there are some challenges that our proposed pedestrian detection and tracking model needs to address, such as performing real-time processing in resource-limited contexts. Dealing with complex situations with frequent occlusions remains difficult. Occlusions can arise in crowded urban areas, and resolving this limitation is the subject of ongoing research.

11. Conclusion

This study presents a robust algorithm for pedestrian detection and tracking by integrating YOLOv8 for object segmentation, HOG for feature extraction, and CNN for classification and detection. For tracking purposes, the SORT algorithm was employed to keep track of pedestrians in successive frames. Extensive experiments were conducted using the EPFL dataset. The results highlight the model's remarkable capability to detect and track pedestrians, achieving an accuracy of 93.34%, even in challenging weather conditions. In future work, it is recommended to improve the accuracy of the proposed system by incorporating more features, such as local binary patterns (LBP) and Haar features, and training the model on larger datasets, including noisy data. Moreover, computational speed can be enhanced by using multi-core parallel environments. As well as developing a mechanism to calculate the distance between pedestrians and the camera, this leads to a more effective detection and tracking system.

12. Acknowledgments

The authors wish to thank the University of Baghdad, College of Engineering, Department of Computer Engineering, for their kind assistance and support.

13. Author’s Declaration

- Conflicts of Interest: None.
- We hereby confirm that all the figures and tables in the manuscript are mine. Besides, the figures and images, which are not mine, have been given permission for re-publication attached to the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at the University of Baghdad.

References

[1] P. Chong and Y. H. Tay, “A novel pedestrian detection and tracking with boosted HOG classifiers and Kalman filter,” *Proc. - 14th IEEE Student Conf. Res. Dev. Adv. Technol. Humanit. SCOReD 2016*, pp.1-5, Dec. 2016 . Doi: 10.1109/SCORED.2016.7810052.

[2] K. Seemanthini and S. S. Manjunath, “Human Detection and Tracking using HOG for Action Recognition,” *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1317–1326, 2018, Doi: 10.1016/j.procs.2018.05.048.

[3] A. Mateus, D. Ribeiro, P. Miraldo, and J. C. Nascimento, “Efficient and robust Pedestrian Detection using Deep Learning for Human-Aware Navigation,” *Rob. Auton. Syst.*, vol. 113, pp. 23–37, 2019, Doi: 10.1016/j.robot.2018.12.007.

[4] X. Chen, Y. Jia, X. Tong, and Z. Li, “Research on Pedestrian Detection and DeepSort Tracking in Front of Intelligent Vehicle Based on Deep Learning,” *Sustain.*, vol. 14, no. 15, p. 9281, July.2022, Doi: 10.3390/su14159281.

[5] A. Amir, A. Karim, and R. A. Sameer, “Comparing the Main Approaches of Image Segmentation,” *Iraqi J. Sci.*, vol. 58, no. 4B, pp. 2211–2221, 2017. Doi: 10.24996/ijs.2017.58.4b.24.

[6] S. M. Alkentar, B. Alsahwa, A. Assalem, and D. Karakolla, “Practical comparation of the accuracy and speed of YOLO, SSD and Faster RCNN for drone detection,” *J. Eng.*, vol. 27, no. 8, pp. 19–31, 2021, Doi: 10.31026/j.eng.2021.08.02.

[7] S. Temitope Yekeen, A. L. Balogun, and K. B. Wan Yusof, “A novel deep learning instance segmentation model for automated marine oil spill detection,” *ISPRS J. Photogramm. Remote Sens.*, vol. 167,pp. 190–200, April. 2020. Doi: 10.1016/j.isprsjprs.2020.07.011.

[8] A. Ben Khalifa, I. Alouani, M. A. Mahjoub, and N. E. Ben Amara, “Pedestrian detection using a moving camera: A novel framework for foreground detection,” *Cogn. Syst. Res.*, vol. 60, pp. 77–96, 2020, doi.org/10.1016/j.cogsys.2019.12.003.

[9] M. F. Aslan, A. Durdu, K. Sabanci, and M. A. Mutluer, “CNN and HOG based comparison study for complete occlusion handling in human tracking,” *Meas. J. Int. Meas. Confed.*, vol. 158, p. 107704, March.2020. Doi: 10.1016/j.measurement.2020.107704.

[10] B. Ali Hussain and M. S. Hathal, “Development of Iraqi License Plate Recognition System based on Canny Edge Detection Method,” *J. Eng.*, vol. 26, no. 7, pp. 115–126, July.2020. Doi: 10.31026/j.eng.2020.07.08.

[11] A. Mohsin and M. Sadoon, “Developing an Arabic Handwritten Recognition System by Means of Artificial Neural Network,” *J. Eng. Appl. Sci.*, vol. 15, no. 1, pp. 1–3, 2020. Doi: 10.36478/jeasci.2020.1.3.

[12] I. M. Hameed, S. H. Abdulhussain, and B. M. Mahmmod, “Content-based image retrieval: A review of recent trends,” *Cogent Eng.*, vol. 8, no. 1, p. 1927469, Jun.2021. Doi: 10.1080/23311916.2021.1927469.

[13] M. M. Rahman, S. Nooruddin, K. M. A. Hasan, and N. K. Dey, “HOG + CNN Net: Diagnosing COVID-19 and Pneumonia by Deep Neural Network from Chest X-Ray Images,” *SN Comput. Sci.*, vol. 2, no. 5, pp. 1–15, 2021, Doi: 10.1007/s42979-021-00762-x.

[14] S. Amraee, M. Chinipardaz, and M. Charoosaei, “Analytical study of two feature extraction methods in comparison with deep learning methods for classification of small metal objects,” *Vis. Comput. Ind. Biomed. Art*, vol. 5, p. 13, 2022, Doi: 10.1186/s42492-022-00111-6.

[15] R. . D. Haameid, B. Q. Al-Abudi, and R. N. Hassan, “Automatic Object Detection, Labelling, and Localization by Camera’s Drone System,” *Iraqi Journal of Science*, vol. 62, no. 12, pp. 5008–5023, Dec. 2021. Doi: 10.24996/ijs.2021.62.12.37.

[16] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and real-time tracking,” *Proc. - Int. Conf. Image Process. ICIP*, vol. 2016-Augus, pp. 3464–3468, 2016, Doi: 10.1109/ICIP.2016.7533003.

[17] F. J. Ansari and A. Ali, “Comparison and study of Pedestrian Tracking using Deep SORT and state of the art detectors,” *Ilkogretim Online - Elementary Education Online*, vol. 20, no. 5, pp. 7848–7859, 2021, Doi: 10.17051/ilkonline.2021.05.889.

[18] R. Pereira, G. Carvalho, L. Garrote, and U. J. Nunes, “Sort and Deep-SORT Based Multi-Object Tracking for Mobile Robotics: Evaluation with New Data Association Metrics,” *Appl. Sci.*, vol. 12, no. 3, p. 1319, Jan.2022. Doi: 10.3390/app12031319.

[19] T. Hong *et al.*, “Multitarget Real-Time Tracking Algorithm for UAV IoT,” *Wirel. Commun. Mob. Comput.*, vol. 2021, p. 9999596 , Aug.2021. Doi: 10.1155/2021/9999596.

[20] Z. Sun, J. Chen, L. Chao, W. Ruan, and M. Mukherjee, “A Survey of Multiple Pedestrian Tracking Based on Tracking-by-Detection Framework,” *IEEE Trans. Circuits Syst. Video Technol.*, IEEE, vol. 31, no. 5, pp. 1819–1833, May.2021.Doi: 10.1109/TCSVT.2020.3009717.

[21] B. Kim, N. Yuvaraj, K. R. Sri Preethaa, R. Santhosh, and A. Sabari, “Enhanced pedestrian detection using optimized deep convolution neural network for smart building surveillance,” *Soft Comput.*, vol. 24, no. 22, pp. 17081–17092, 2020, Doi: 10.1007/s00500-020-04999-1.

[22] Ahmed Faiq Al-Alawy, E. E. Al-Abod, and Raya Mohammed Kadhim, “Proposed Face Detection Classification Model Based on Amazon Web Services Cloud (AWS),” *J. Eng.*, vol. 29, no. 4, pp. 176-206, April.2023. Doi: 10.31026/j.eng.2023.04.12.

[23] Z. Chen, R. Khemmar, B. Decoux, A. Atahouet, and J. Y. Ertaud, “Real time object detection, tracking, and distance and motion estimation based on deep learning: Application to smart

mobility," *2019 8th Int. Conf. Emerg. Secur. Technol. EST 2019*, pp. 1–6, 2019, Doi: 10.1109/EST.2019.8806222.

[24] L. Kalake, Y. Dong, W. Wan, and L. Hou, "Enhancing Detection Quality Rate with a Combined HOG and CNN for Real-Time Multiple Object Tracking across Non-Overlapping Multiple Cameras," *Sensors*, vol. 22, no. 6, p. 2123 , March.2022. Doi: 10.3390/s22062123.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, Doi: 10.1109/TPAMI.2016.2577031.

[26] J. Redmon and A. Farhadi, "YOLO v.3," *Tech Rep.*, pp. 1–6, 2018, [Online]. Available: https://pjreddie.com/media/files/papers/YOLOv3.pdf%0Ahttps://pjreddie.com/yolo/.

[27] M. Tan, R. Pang, and Q. V. Le, "EfficientDet," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 10778–10787, 2020.

[28] A. López-Cifuentes, M. Escudero-Viñolo, J. Bescós, and P. Carballeira, "Semantic-driven multi-camera pedestrian detection," *Knowl. Inf. Syst.*, vol. 64, no. 5, pp. 1211–1237, 2022, Doi: 10.1007/s10115-022-01673-w.