

Computer vision-based framework for pedestrian movement direction recognition

Monika*, Pardeep Singh and Satish Chand

School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India

Abstract. Pedestrians are the most critical and vulnerable moving objects on roads and public areas. Learning pedestrian movement in these areas can be helpful for their safety. To improve pedestrian safety and enable driver assistance in autonomous driver assistance systems, recognition of the pedestrian direction of motion plays an important role. Pedestrian movement direction recognition in real world monitoring and ADAS systems are challenging due to the unavailability of large annotated data. Even if labeled data is available, partial occlusion, body pose, illumination and the untrimmed nature of videos poses another problem. In this paper, we propose a framework that considers the origin and end point of the pedestrian trajectory named origin-end-point incremental clustering (OEIC). The proposed framework searches for strong spatial linkage by finding neighboring lines for every OE (origin-end) lines around the circular area of the end points. It adopts entropy and $Q_{measure}$ for parameter selection of radius and minimum lines for clustering. To obtain origin and end point coordinates, we perform pedestrian detection using the deep learning technique YOLOv5, followed by tracking the detected pedestrian across the frame using our proposed pedestrian tracking algorithm. We test our framework on the publicly available pedestrian movement direction recognition dataset and compare it with DBSCAN and Trajectory clustering model for its efficacy. The results show that the OEIC framework provides efficient clusters with optimal radius and minlines.

Keywords: Unsupervised learning, line clustering method, origin-end point, pedestrian direction of motion, YOLOv5

1. Introduction

Pedestrian movement direction recognition is important in autonomous driver assistance and security surveillance systems. In real-time pedestrian detection and predicting their behavior have been extensively explored in the field of computer vision [1]. In recent years, various deep learning and machine learning techniques have been developed to learn pedestrian dynamic motion based on trajectory, motion, gait etc. [2, 3]. But these techniques need large labeled data and high computational cost to perform the task. The various factors that impact pedestrian movement detection are: occlusion, illumination and body pose [4–6]. To solve these challenges, we proposed a simple and intuitive origin-end point incremental clustering (OEIC) framework for pedes-

trian movement direction recognition by extracting spatial linkage information using OE line data. The proposed framework learns pedestrian movement dynamics in road scenes. The frames are extracted from the input videos and further processed for pedestrian detection and tracking. Similar pedestrian tracks are clustered together that capture the pedestrians motion activities in the surrounding environment using origin-end point incremental clustering (OEIC). Fig.1. shows the basic pipeline of our work.

1.1. Motivation

From related work, as discussed in Section 2, we observe the pedestrian direction of motion recognition have following three research gaps. First, the authors used machine learning or deep learning classifiers based on spatial data, i.e. body pose, head orientation, etc., to classify the direction of motion. These methods may fail due to the impossibility of

*Corresponding author. Monika, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India. E-mail: monikasingh314@gmail.com.



Fig. 1. Basic flowchart of our proposed work.

properly handling and adapting to changes in pedestrian movements [7]. Another is limited pedestrian datasets are available for training supervised classifiers. Second, pedestrian trajectory analysis focusing on the visual representation of temporal movements. These methods take high computational costs. Third, it focused on pedestrian motion recognition using sensor data. These approaches are very intrusive from the pedestrian viewpoint and need to provide more information to distinguish between different pedestrian actions. These research gaps motivate us to develop a computationally efficient framework for the pedestrian direction of motion recognition without labeled data and has applications in finding mobility patterns of pedestrians in critical areas like shopping malls, square, zebra crossings, stations and public gatherings.

The OEIC framework starts with the idea that the origin and end (OE) line of pedestrians with similar flows or trajectory are in the same region within a certain radius (D_r). These OE flows allow us to search pedestrian direction of motion such as front, right and left movements. To accomplish this task, we need to first detect the pedestrian using YOLOv5 algorithm and then track multi-pedestrians across the frame using centroid coordinates from the output of the detected bounding box and then apply the OEIC framework. We evaluate our proposed work using $Q_{measure}$ and entropy to showcase dominant mobility patterns from a large number of flow data effectively.

We summarize our contributions as follows:

- We use a state-of-the-art YOLOv5 object detection deep learning model for pedestrian detection.
- After detecting the pedestrian, their trajectories are estimated across the video frames.
- A novel clustering method is proposed to cluster origin-end point trajectory for pedestrian movement direction recognition named Origin-End point Incremental Clustering (OEIC).
- The cluster quality is estimated using Entropy (H) and Quality measure ($Q_{measure}$).
- The proposed method is evaluated on a publicly available dataset named- Pedestrian movement direction recognition for its efficacy.
- OEIC framework performed better than other State-of-the-art methods.

The remainder of the paper is organized as follows: The overview of previous work is presented in Section 2. Section 3 discusses pedestrian detection, tracking and Origin-End incremental Clustering (OEIC) framework. The experimental results are presented in Section 4. Finally, in Section 5, we conclude the paper.

2. Related work

Pedestrian detection and tracking are the basis for pedestrian safety in ADAS and system monitoring. Various techniques have been discussed in recent years for pedestrian detection and tracking [8–12]. However, occlusion, low resolution, illumination, weather conditions and day timing are challenging tasks for video pedestrian detection.

In recent years classic as well as machine learning (ML) techniques have been used for pedestrian detection [13–17]. Authors in [13, 16, 17] discussed hybrid techniques for pedestrian detection. The major concern with these techniques is high computational complexity. Authors in [18] discuss sixteen pedestrian detectors and test them over several public datasets. The obtained results do not handle pedestrians occlusion and high dimensional variations.

The hybrid machine learning approach incorporating a Support Vector Machine (SVM) with Adaboost has been discussed in a complex video scene covering roads and surrounding objects [19]. Authors in [20, 21] discussed person detection methods using Kalman Filter(KF) over publicly available video dataset capturing dash cam videos and wearable camera. The optical flow feature has been used to detect humans and vehicles in traffic scenes [22]. This method resolves illumination and occlusion problems.

Another study to handle occlusion and visibility issues for pedestrian detection in videos [1] uses Restricted Boltzmann Machine (RBM). In [23], the authors used Deep Convolutional Neural Networks (DCNN) for detection. However, the discussed techniques for detection fails for low illumination data. FlowNet2 CNN with optical flow technique is used for person abnormal behavior detection in the crowded scene [22]. It overcomes various challenges for person detection. YOLOv5 [24] state-of-the-art object detection model is used to detect various objects with over 80 classes trained over the Common Objects in Context (COCO) dataset. It is a reliable and efficient method for pedestrian detection.

Pedestrian tracking is the next crucial step for understanding the intent of pedestrians. Authors in [25] discussed the heuristic-based technique for tracking human gait. Traditionally KF, histogram of Oriented Gradients (HOG), intensity, optical flow, Scale-Invariant Feature Transform (SIFT), Local Binary Pattern (LBP) etc. have been used for estimating the pedestrian trajectory [26, 27]. The approaches discussed above focus on detection but cannot handle pedestrian movement which plays a vital role in pedestrian safety.

In recent years, machine learning and Deep Learning have been discussed for pedestrian tracking. ANN-based tracking used in [28] for pedestrian tracking and intent prediction. They used person head coordinates for tracking the person across the frames in videos. In paper [3] person's neck position have been used for tracking on traffic roads. In paper [29] dense optical flow features have been discussed for pedestrian tracking. In paper [30] authors discussed combined CNN-based pedestrian detection, tracking and pose estimation to predict the crossing action from dash-cam based videos. Paper [31] discusses an IMU-based pedestrian positioning tracking technique that assists bedridden patients by tracking the patient indoor environment. A generic deep learning and color intensity-based technique is discussed for person tracking in [32]. It overcomes the challenges i.e. failing at night and low illumination weather conditions. A multi-object real time tracking technique is discussed in the field of computer vision application in [33]. The authors used MobileNet based custom CNN techniques for feature extraction and a set of algorithms to generate associations between frames and tested them on the publicly available Towncentre dataset.

Pedestrian dynamic motion recognition is an important task in ADAS and pedestrian safety. In paper [5] author discussed three CNN-based models (AlexNet, GoogleNet and ResNet) for pedestrian direction. The results show that ResNet gives 79% accuracy in comparison to other models. Authors in [34] discussed Capsule Network for classifying pedestrian walking direction in order to alert the driver. Pedestrian body pose feature were used to predict pedestrian motion on road scene in [35]. In paper [36] LSTM encoder-decoder is discussed to learn abnormal events in a complex environment. The hybrid method failed to differentiate between two different actions. To study abnormal behavior detection on video surveillance, incremental semi-supervised learning has been discussed [4]. Using the trajectory

data feature, the classifier distinguishes between normal and abnormal behavior patterns. Recurrent neural network based technique have been discussed by [37] for embedding pedestrian dynamics in multiple moving agents. It captures the pedestrian's motion using Long-Short-Term Memory networks. Results were evaluated on publicly available Stanford Drone and SAIVT Multi-Spectral Trajectory datasets. Authors in [38] discussed Micro-Doppler (MD)-based classification technique for the accurate direction of motion estimation. They used support vector regression and multilayer perceptron-based regression algorithms.

The primary challenge in the supervised learning technique is that we need labeled data for training the model. Therefore, we study unsupervised learning techniques that clusters similar activities in one cluster based on their similarity. This results in finding clusters grouped together having similar features. In [39], authors discussed an unsupervised learning algorithm that can form clusters of human motion for road surveillance data. Authors in [40] discussed an extended version of the Variational Gaussian Mixture model (VGMM) based probabilistic trajectory prediction technique. It subclassifies pedestrian trajectories in an unsupervised manner based on their estimated sources and destinations and trains a separate VGMM for each sub-category. K-mean algorithm has been discussed in [41] for activity recognition using static frames features.

In [42] authors discussed hierarchical clustering technique to detect small groups of people traveling together that can be used in evacuation planning and handling real-time public disturbances. An unsupervised mobile robot have been developed to learn human motion patterns in a real-world environment to predict future behaviors [43]. In [44] authors discussed group sparse topical coding (GSTC) technique to study motion patterns. In [45] used trajectory clustering based on low level trajectory patterns and high level discrete transitions for predicting anomalous pedestrian detection. All the methods discussed above predict pedestrian motion based on trajectory data or pedestrian captured from top view.

To the best of our knowledge, vision base unsupervised learning for pedestrians direction of motion recognition has not been studied earlier. We propose an unsupervised learning framework that clusters the direction of pedestrian motion provided an initial and end point of pedestrian dynamics so that it can be used in ADAS to alert the driver and raise risk alarm.

3. Methodology

The objective of our work is to develop a fast and efficient unsupervised framework for pedestrian direction of motion recognition. We first detect a pedestrian in the scene using the YOLOv5 model. Then we extract the origin and end point coordinates by tracking the detected pedestrian across the frames. Further, we apply unsupervised incremental clustering to capture possible pedestrian dynamics in the given scene. In Fig. 2, we explain the architecture of our proposed framework in detail.

3.1. Pedestrian detection

YOLO is the first object recognition model that integrates object detection and bounding box into a single end-to-end differential network. All the variants of YOLO are developed and maintained in darknet environment and are based on CNN. The YOLO model divides the image into regions in the form of a bounding box and calculates each region's probability. These bounding boxes are weighted using the predicted probabilities. We "only look once" at the image since the technique only requires one forward propagation run through the neural network to produce predictions. After a non-max suppression, it produces known items along with the bounding boxes (which ensures that the object detection algorithm only recognizes each object once). Due to its real-time adequate pedestrian and road surrounding object recognition, we used the recently released architecture YOLOv5.

The YOLOv5 is an object detection model specifically pre-trained over the Common Objects in Context (COCO) dataset. The COCO dataset is primarily utilized for various tasks like recognition of objects, segmentation of objects and labeling tasks. It contains approx. 2,00,000 images and have 80 different classes. In our study, we run the YOLOv5 algorithm related to person classes. The detected output of YOLOv5 [46, 47] gives bounding box coordinate file of each object detected on each frame in the video. These coordinates are further used in the tracking of the person's class.

3.2. Pedestrian tracking

Tracking sequences of objects in video is an essential task for the pedestrian direction of motion recognition. We use spatial information of the frames across the video. Our approach for tracking is to track

the detected person across the frames using output obtained from YOLOv5. The output of YOLOv5 is in the form of bounding box coordinates signified by (x_i, y_i, h_i, w_i) and class-id (c_i). Where, (x_i, y_i) are the top left coordinate of the detected bounding box and h_i, w_i are the height and width of the bounding box. Corresponding to person class, we track each person based on its nearest centroid position (x_i, y_i) across the frames which form sequence of paths. This centroid is calculated from detected points (x_i, y_i, h_i, w_i) of person class (c_i). If the position of point is far from the sequence, it's not considered part of that path. We use euclidean distance to calculate the nearest centroid position discussed in equation (1) below. This results, in obtaining sequence of paths $T = \{t_1, t_2, \dots, t_n\}$, where each $t_i = \{(x_1, y_1), (x_2, y_2), \dots, (x_p, y_p)\}$ for each person at the end of the video. To better understand our work, we discuss our approach in algorithm 1 shown below:

Algorithm 1 Pedestrian Tracking algorithm

Input: Class-id (c_j) and (x_i, y_i) centroids of person detected per frame

Output $T = \{t_1, t_2, \dots, t_n\}$, where n= Number of person tracks

Initialize: $T = \emptyset$

```

1: for each centroid  $(x_t, y_t)$  in frame  $f_t$  at time  $t$ ,
    $f_t \in \{f_1, f_2, \dots, f_n\}$  do
2:   if  $t == 1$  then
3:     assign unique_id to  $t_k$  and add centroid
        $(x_t, y_t)$  to  $t_k$ 
4:     add  $t_k$  to  $T$ 
5:   end if
6:   calculate euclidean distance ( $dist$ ) between
       centroids  $(x_t, y_t)$  detected in frame  $f_{t-1}$  and
        $f_t$  using equation (1)
7:   if distance  $dist((x_t, y_t), (x_{t-1}, y_{t-1})) <$ 
       threshold (set threshold=20) then
8:     add  $(x_t, y_t)$  to  $t_k$  of  $(x_{t-1}, y_{t-1})$ 
9:   else
10:    assign new unique_id of  $t_{k+1}$ 
11:    add  $(x_t, y_t)$  to  $t_{k+1}$ 
12:   end if
13:   add  $t_k$  to  $T$ 
14: end for
15: return  $T$ 

```

To calculate the euclidean distance ($dist$) between two points (x_i, y_i) and (x_j, y_j) following equation is used.

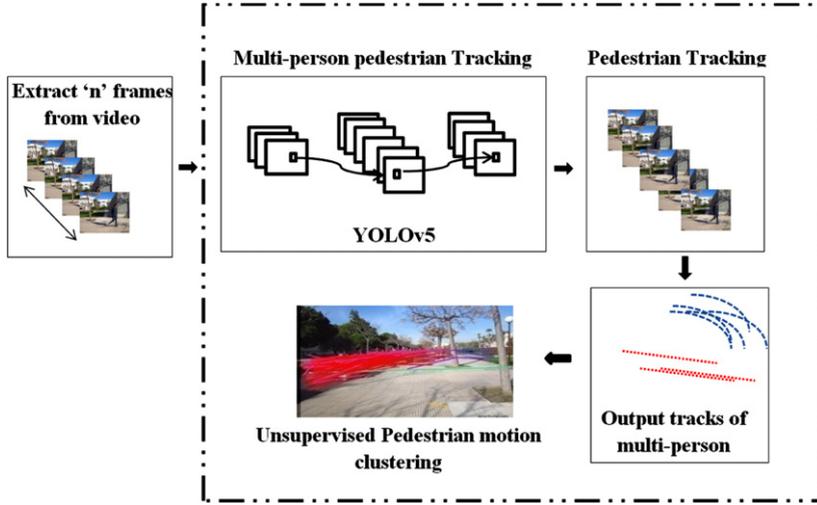


Fig. 2. Proposed framework for pedestrian direction of motion recognition. In step 1, we take video as input and then perform a deep learning model YOLOv5 for pedestrian detection. In step 2, we track pedestrians across the frame. In step 3, we perform unsupervised incremental clustering of the pedestrian direction of motion recognition.

$$\text{dist}((x_i, y_i), (x_j, y_j)) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

3.3. Problem formulation

The main goal of our study is to cluster similar pedestrian motion trajectories in one cluster based on pedestrian paths. The problem is formulated as:

Let $L = \{L_1, L_2, \dots, L_n\}$ be set of lines and each L_i is a directed line connecting origin-end point coordinates i.e. (x_o, y_o) , (x_e, y_e) and $n = |L|$, represents the number of lines. We cluster the set of lines, L using proposed OEIC framework to form cluster of similar flow.

3.4. Origin-end point incremental clustering(OEIC) framework for pedestrian movement direction recognition

Origin-End (OE) data are a special case of trajectory data that focuses on the origin and end coordinates of the trajectory. Whereas in pedestrian trajectory and intent prediction, the actual trajectory information is important. Therefore, it is essential to monitor the location of each movement actively. In the pedestrian direction of motion, the origin and end points are enough to analyze the connections between locations and their spatial characteristics for detecting their dynamic motion. The key idea is to cluster similar OE lines with sufficient neighbor lines. To

better understand the OEIC framework, we discuss our approach in algorithm 2.

We now discuss the parameters and their definitions used in our framework:

Centerline L_i and its neighbouring lines (Nls) are defined as:

Let O and E be the origin and end points and L_j is line joining j^{th} point of O and E . If line L_j lies within searching radius of O_i and E_i , then L_j is defined as neighboring line of centerline L_i as shown in Fig.3. A centerline may have many neighbouring lines; the number of these neighbouring lines $Nls(L_i)$ is represented as:

$$\begin{aligned} Nls(L_i) &= \{L_j \in L \mid \text{dist}(O_i, O_j) \\ &\leq D_r \cap \text{dist}(E_i, E_j) \leq D_r\} \end{aligned} \quad (2)$$

where D_r is search radius and $\text{dist}(O_i, O_j)$ and $\text{dist}(E_i, E_j)$ are euclidean distance between two endpoints. L is set of OE lines i.e. $L = \{L_1, L_2, \dots, L_n\}$.

Parameters 1: Search Radius D_r is the first parameter in OEIC framework. To find cluster of lines, we need to cluster similar lines from OE lines data in a given radius. Size of search radius is directly proportional in determining the shape, size and quality of the clustering results. Generally greater the search radius, many neighboring lines can be detected. However, a particular scenario that needs to be considered, i.e., when line L_j is short (e.g. $L_j < 2 * D_r$, then it could not assure spatial resemblance because L_j can be in different or opposite direction of L_i (i.e., green line in

Algorithm 2 Origin-End point Incremental Clustering (OEIC)

Input: OE lines L , D_r and $Minlines$
Output $C = \{L_1, L_2, \dots, L_k\}$, where k = Number of cluster and $Nls(L_i)$ contains all the neighbouring OE lines of L_i
Initialize: $C = \emptyset$

```

1: for each  $L_j \in L$  do
2:   if  $j == 1$  then
3:     add  $L_j$  to  $C$ 
4:     add  $L_j$  to  $Nls(L_i)$ 
5:   end if
6:   calculate length of  $L_j$  using equation
   euclidean distance formula from equation (1)
7:   if  $length(L_j) > 0.83 * D_r$  then
8:     if O and E points of  $L_j$  are within the
       radius  $D_r$  of O and E point of line  $L_i$  then
9:       Select  $L_i$  which is nearest to  $L_j$  and
       assign  $L_j$  to  $Nls(L_i)$ 
10:    else
11:      Add  $L_j$  to  $C$ 
12:    end if
13:  else
14:    ignore the line  $L_j$ 
15:  end if
16:
17: end for
18: if  $Nls(L_i) > Minlines$  then
19:    $L_i$  is valid cluster
20: end if
21: return  $C, Nls$ 

```

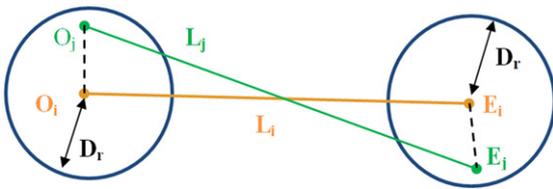


Fig. 3. Illustrating the neighboring line of a centerline.

Fig.4). This results in line clustering with wrong information.

To ensure that our framework should cluster all the lines that are geographically close to each other and in the same direction. We consider only those line L_i that are greater than $2.83 D_r$ (calculated and explain in fig 5 i.e. $L_i \geq D_r / \sin 45^\circ$). Hence, in our framework, only those OE lines can participate in the clustering whose length is greater than $0.83 * D_r$. This threshold ensures that the angle between the centerline and its

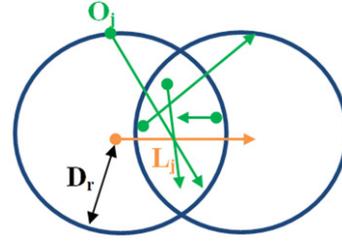


Fig. 4. Particular cases of search radius (D_r) (Origin and End points are represented as 'o' and arrow).

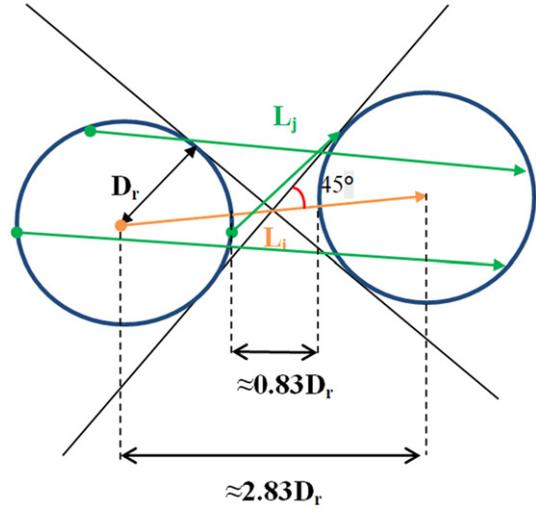


Fig. 5. The length constraint of L_i (Origin and End points are represented as 'o' and arrow).

neighboring line will always be less than 45° , and the length change won't differ by more than twice D_r .

Parameter 2: Selecting a minimum number of neighboring lines $Minlines$ in a cluster: In general, a cluster is said to be informative if it contains a sufficient number of elements in that cluster. A centerline is said to be informative if it has more neighboring lines. The result may exclude all the centerlines with a number of neighboring lines less than the threshold $Minlines$ in the cluster analysis.

Determining the values of the two parameters discussed above, we propose the following strategy.

Determining the Parameters (D_r and $Minlines$) With the help of field expert knowledge, we could determine D_r and $Minlines$ directly. For example, if we want to find the strongest spatial connection in motion recognition with OE data, we can set the D_r as 80 and the $Minlines$ to be 10% of set of lines. Experience-based parameter estimation may not always be optimal. Hence, in the absence of prior knowledge, we use statistical clustering measures i.e.,

Entropy and $Q_{measure}$ to validate the clustering quality and parameter selection [48].

Entropy: In information theory, entropy measures how uncertain an event is in a given specific probability distribution [49]. If all the outcomes are equally likely, then the entropy (H) value is maximum. In the worst-case clustering scenario, Nls tends to be uniform when the value of D_r is too small, and Nls becomes 1 for almost all lines i.e., each line is itself acting as a cluster and for D_r that is too large, all the lines L_i get in one cluster. Thus, the entropy becomes maximized. We calculate the entropy as per equations (3) and (4) to get the optimal value of D_r that minimizes H(L).

$$H(L) = - \sum_{i=1}^n p(L_i) * \log_2 p(L_i) \quad (3)$$

$$p(L_i) = Nls(L_i) / \sum_{i=1}^n Nls(L_i) \quad (4)$$

We initialize D_r with a small value and increase it gradually. Simultaneously at each D_r , we also calculate entropy for all Nls . This heuristic method provides a reasonable range where the optimal value is likely to reside. This method gives a good estimation of the number of clusters and the value of search radius D_r [48].

Minimum lines: Another parameter, *Minlines* discussed above, we follow a simple quality measure for a ballpark analysis to determine a minimum number of lines [50]. It not only considers Sum of Squared Error (SSE) [48] but also considers noise penalty to penalize incorrectly classified noises. The noise penalty becomes larger if we select too small D_r or too large *Minlines* parameter. Consequently, our quality measure is the sum of the total SSE and the noise penalty as discussed in equations (5-7) below. Here, \mathcal{N} denotes the set of all noise OE lines.

$$Q_{measure} = \text{Total SSE} + \text{Noise Penalty} \quad (5)$$

$$\text{Total SSE} = \sum_{i=1}^{num_{clus}} \left(\frac{1}{2|C_i|} \sum_{x \in C_i} \sum_{y \in C_i} \text{dist}(x, y)^2 \right) \quad (6)$$

$$\text{Noise Penalty} = \frac{1}{2|\mathcal{N}|} \sum_{w \in \mathcal{N}} \sum_{z \in \mathcal{N}} \text{dist}(w, z)^2 \quad (7)$$

As trajectories are closely related to time series, Euclidean distance is also adopted in measuring tra-

jectory distance [50]. For two line L_i and L_j with the same size n , the Euclidean distance $\text{dist}(L_i, L_j)$ is defined as follows:

$$\text{dist}(L_i, L_j) = \frac{(\text{dist}(O_i, O_j) + \text{dist}(E_i, E_j))}{2} \quad (8)$$

where O_i, O_j are the origin and E_i, E_j are the end point of L_i and L_j , respectively. Euclidean distance is measured using equation 1. The notations used in our framework are summarized in Table 1.

4. Experimental result

4.1. Dataset

To test our proposed framework's performance, we need a specific dataset of pedestrians moving in different directions and covering various road scenarios like a zebra crossing, signal area, sidewalk etc. We use the publicly available dataset Pedestrian direction recognition [5]. It's the only dataset that captures the pedestrian movement direction. This dataset consists of several hours of videos with 10800 video frames recorded in five different locations captured at 30 fps with 640 x 480 resolution. The hand-held tripod is used to capture pedestrian movement statically and dynamically.

Other datasets related to pedestrians like PEI, INRIA, KITTI, CityPersons, TUD, and EuroCity etc, have been discussed in the literature [9]. These datasets focus on detection and their action/activity, but no information about their direction and intention is provided. The Daimler dataset is the only dataset that provides ground truth information about detection and their intention. It has short video clips with one of the four labeled classes labeled i.e. crossing, walking, standing and bending. So, we used the pedestrian direction recognition dataset developed specifically for the direction of motion recognition. Fig. 6 shows the sample frames of the pedestrian direction recognition dataset.

4.2. Results and discussion

The experimental result of the proposed framework is discussed in this section.

4.2.1. Detection and tracking

We prefer the YOLOv5 deep learning method over other methods due to its efficient and fast pedestrian

Table 1
Major notations used in OEIC framework

S. No.	Notations	Description
1	OE	Origin end point
2	O_i	i^{th} origin point of the line L_i
3	E_i	i^{th} end point of the line L_i
4	L_i	Centerline of each cluster
5	L_j	Neighboring line of the centerline L_i
6	Nls	The number of neighbouring lines of centerline L_i are stored in Nls .
7	D_r	Search Radius
8	$Minlines$	Minimum number of neighboring lines in a cluster.
9	L	Set of origin end-point(OE) lines i.e. $L = \{L_1, L_2, \dots, L_n\}$
10	$H(L)$	Entropy
11	SSE	Square Sum Error
12	$Q_{measure}$	Quality Measure.
13	\mathcal{N}	Set of all noise OE lines.
14	$dist(x, y)$	Calculates the euclidean distance between x and y.
15	$NoisePenalty$	Penalty to penalize incorrectly classified noises.
16	$TotalSSE$	Sum of Squared Error(SSE).
17	OE line	Line formed by joining origin and end (x_o, y_o) , (x_e, y_e) coordinates.
18	C_i	i^{th} cluster of OEIC framework.



Fig. 6. Dataset images: Left, front and right moving frames from videos.

detection response in road scenarios [46, 47]. The results are shown in Fig. 7.

After detection, we get a bounding box of the pedestrian detected in each video frame. Corresponding to the pedestrian class, we perform pedestrian tracking across frames to get the tracks of each pedestrian. Fig. 8 shows the result of tracking pedestrians in red color.

4.2.2. Origin-end point incremental clustering (OEIC) framework

To understand spatial variability of the pedestrian direction of motion, we use OE data lines. To obtain optimal parameters of D_r and $Minlines$, we initialize D_r with a small parameter value of 10 and iterate it over an equal time interval. The analysis progress and results are shown in the following section.

With various D_r values, we determined the entropy of each Nls value. To determine the optimal number of clusters, we have to select the value of D_r

at the "elbow," i.e. the point after which the distortion becomes steady. We noted from Fig. 9 that with an increase in D_r , the entropy of Nls decreases and after $D_r=80$, entropy becomes steady. Together with entropy, we calculated $Q_{measure}$ to determine the minimum number of OE lines for stating as a right cluster. Fig.10 shows the quality measure $Q_{measure}$ as D_r and $Minlines$ are varied. We observe that our $Q_{measure}$ becomes nearly minimal when the optimal parameter values are used. The minimum $H(L)$ is achieved at $D_r = 80$ and $Q_{measure}$ possess minimum error value at $Minlines \geq 27$.

Table 2 shows the outline information of the OEIC clustering result. Figure 11 and 12 show the clustering results in 3D and image frame for better visualization of the proposed framework using the optimal parameter values of D_r and $Minlines$. We observe that three clusters are discovered in most of the dense regions. A dot (.) in the graph represents the origin point and an arrow/star (<, >, *) represents the end point of pedestrians movement. Fig. 11 and 12 show three clusters



Fig. 7. Results of pedestrian detection using YOLOv5 on three frames taken from the video dataset.

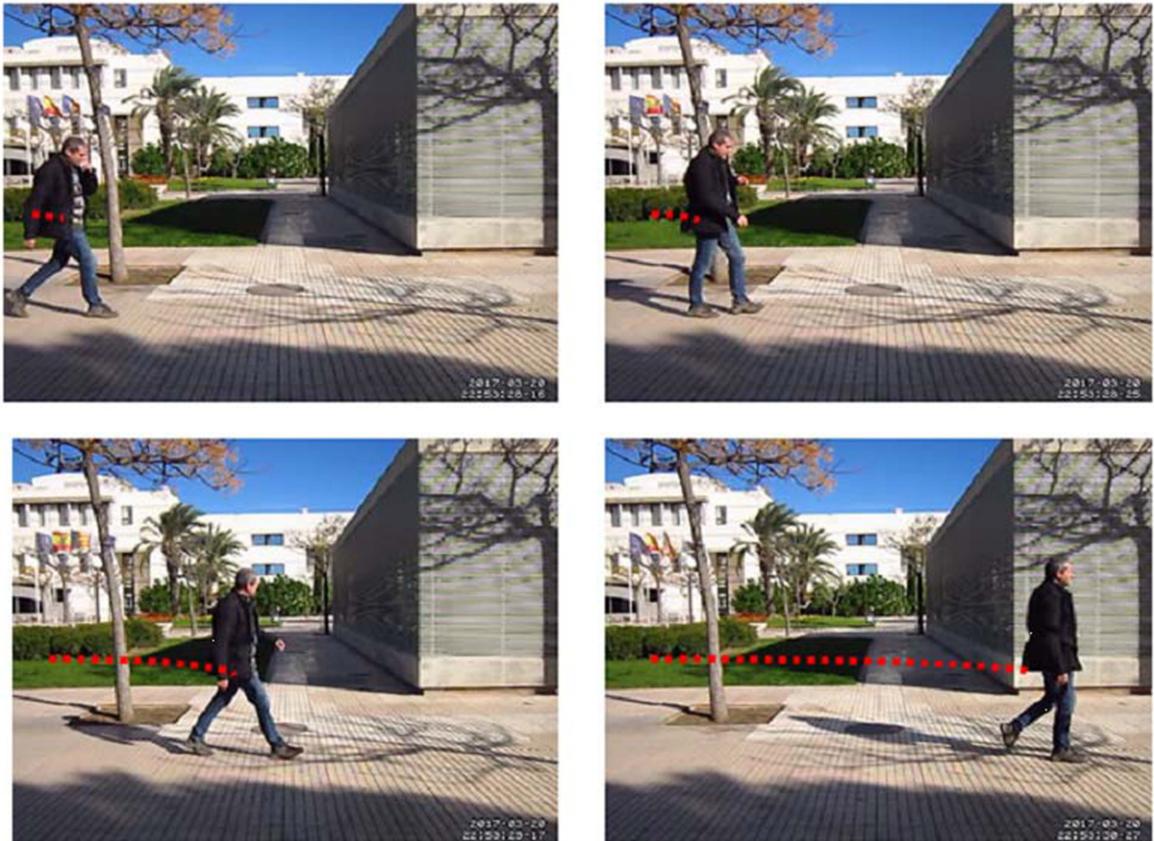


Fig. 8. Result of single-pedestrian tracking on a video frame.

in the region, verified from the pedestrian direction recognition dataset.

Table 2 shows the outline information of the OEIC clustering result. Fig. 11 and 12 show 3D and 2D views of three clusters verified from the pedestrian direction recognition dataset. We are interested in the computational efficiency of the proposed framework. Therefore, different experiments were carried out to measure the execution time for each stage of the pipeline i.e. from pedestrian detection to clustering as shown in Table 3. We also run two other well-known

clustering algorithms: DBSCAN and Trajectory clustering for performance comparison purposes. It is clear from the results that the proposed methods give a better value of $Q_{measure}$ than the other two clustering methods. Our model takes less computational time than Trajectory Clustering but a little more than DBSCAN as shown in Table 4.

To the best of our knowledge and exploration, this is the first work that uses unsupervised learning with origin-end clustering methods for pedestrian direction of motion. The authors of the dataset we

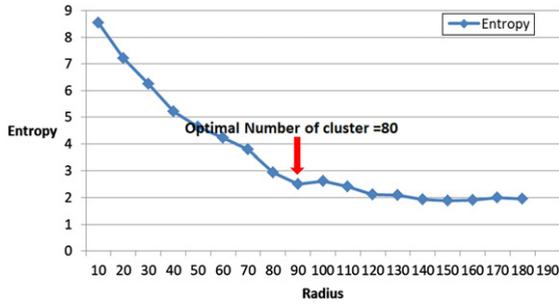


Fig. 9. Entropy of all NIs values with different radius from 10 to 180.

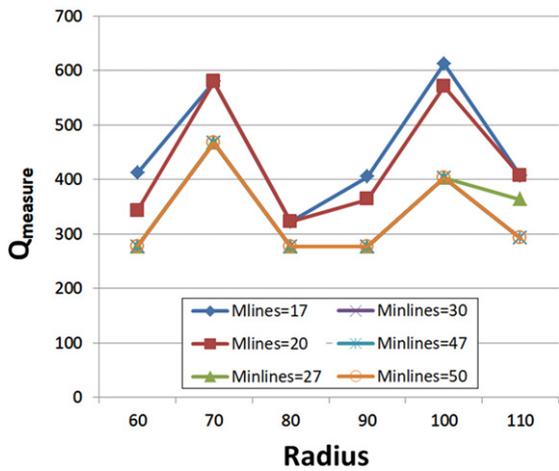


Fig. 10. Quality measure ($Q_{measure}$) with different values of Radius (D_r) and $Minlines$.

used in our experiment have applied the CNN model for pedestrian direction of motion classification [5]. Since we are performing clustering on this data, the OEIC framework is revalidated by the label vector that all the pedestrians walking toward left, right and straight on the road are differentiated with an accuracy of 85%. We also find studies in the literature that use supervised and unsupervised methods related to our work using different datasets. The comparative results are shown in Table 6.

4.3. Effects of parameter values

We have tested the effects of D_r and $Minlines$ values on the clustering result. If we use a smaller D_r or a larger $Minlines$ compared with the optimal ones, our algorithm discovers a larger number of smaller clusters (i.e., having fewer line segments). In contrast, if we use a larger D_r or a smaller $Minlines$, our algorithm discovers a smaller number of larger clusters.

Table 2
Features of the clusters with different fixed radius

Radius	Number of Clusters	Average number of cluster lines	Length of center lines
10	467	2	28.3
20	242	3	56.6
30	137	5	84.9
40	85	8	113.2
50	55	11	141.5
60	37	16	169.8
70	29	20	198.1
80	21	26	226.4
90	18	30	254.7
100	11	47	283
110	10	50	311.3
120	8	77	339.6
130	8	74	367.9
140	6	73	396.2
150	6	71	424.5

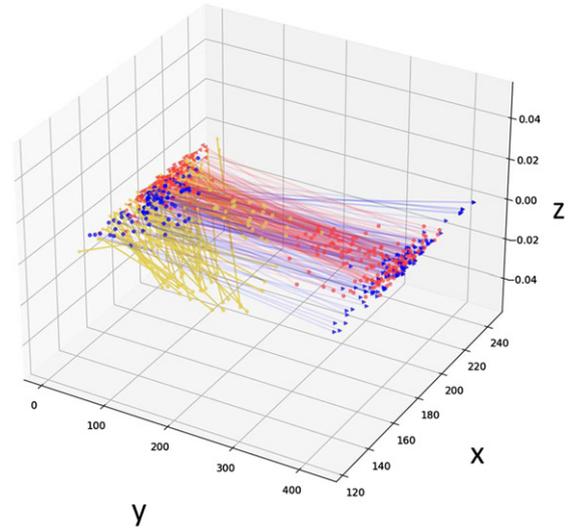


Fig. 11. 3D representation of clustering result. Yellow, red and blue colored lines represent OE lines cluster results. Here three clusters are formed using the OEIC framework.

For example, when $D_r = 10$, then 467 clusters are found, and each cluster contains 1-2 line segments on average; in contrast, when $D_r = 80$, three clusters are discovered, and each cluster includes 30 lines on average.

4.4. Error analysis

For pedestrian dynamics learning, our proposed framework performs well. But to better understand the errors made by the OEIC framework, we investigate and discuss below:

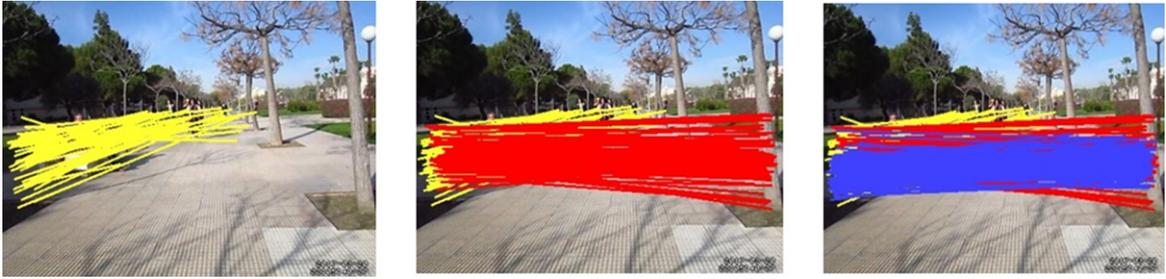


Fig. 12. Clustering results of OEIC framework on pedestrian direction recognition dataset.

Table 3
Runtime of each stage of the framework

Stages	Runtime (sec)
Person detection	1.30
Multi-Pedestrian Tracking	2.10
OEIC	15.10
Total	18.50

Table 5
Result comparison of our framework with other clustering methods in terms of $Q_{measure}$ on pedestrian direction recognition dataset

Clustering Algorithm	$Q_{measure}$
DBSCAN	456
Trajectory Clustering	304
OEIC(our)	286

Table 4

Comparison w.r.t runtime of our framework with other clustering algorithms on pedestrian direction recognition dataset

Clustering Algorithm	Time(sec)
DBSCAN	16.20
Trajectory Clustering	85.00
OEIC(our)	18.50

- $Nl_s(L_i)$: If the number of OE lines in a cluster is less than the threshold value i.e minlines then we may loss some informative clusters.
- Same cluster predicted as two clusters due to overlapping regions. This may be due to partial difference in their spatial movement.

5. Conclusion and future work

This paper discusses a simple and intuitive Origin-End point Incremental Clustering (OEIC) framework for pedestrian movement direction recognition by extracting spatial linkage information from an origin-end (OE) line data. It identifies neighboring lines by searching the endpoints of OE lines within a circu-

lar region (D_r) of the centerline (L_i). Entropy and $Q_{measure}$ have been used to determine the optimum value that helps to select quality clusters. To demonstrate the effectiveness of the proposed framework, we performed experiments on a publicly available dataset-pedestrians movement direction recognition, which contains videos capturing pedestrian movement in different directions on road scenes. The results show that the proposed framework performs better than the DBSCAN and Trajectory Clustering algorithms. The advantage of our framework is that it is simple and backed by extensive experimentation. It can be used in various applications especially in pedestrian safety and detecting anomalous trajectories and understanding pedestrian movement behavior in monitoring and ADAS systems.

This work is a base for analyzing pedestrian motion direction recognition. We would develop a hybrid-clustering technique to obtain new insights into pedestrian dynamics using other spatio-temporal features.

Table 6
Result comparison of our framework with state-of-the-art methods

Method	Model	Dataset	Accuracy(%)
Supervised [5]	CNN (ResNet)	pedestrians movement direction	79
Supervised [34]	Capsule Network	SAFEROAD	79
Supervised [35]	multi-Bayesian model	dashcam video	64
Unsupervised	OEIC	pedestrians movement direction	85

Declaration of competing interest

The authors have no competing interest.

Acknowledgment

The authors thank the anonymous reviewers for their insightful comments and suggestions.

References

- [1] Wanli Ouyang, Xingyu Zeng and Xiaogang Wang, Modeling mutual visibility relationship in pedestrian detection, In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3222–3229, 2013.
- [2] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich, Going deeper with convolutions, In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [3] Amar A. El-Sallam and Ajmal S. Mian, Human body pose estimation from still images and video frames, In *International Conference Image Analysis and Recognition*, pages 176–188. Springer, 2010.
- [4] Rowland R. Sillito and Robert B. Fisher, Semi-supervised learning for anomalous trajectory detection, In *BMVC*, volume 1, pages 035–1. Citeseer, 2008.
- [5] Alex Dominguez-Sanchez, Miguel Cazorla and Sergio Orts-Escolano, Pedestrian movement direction recognition using convolutional neural networks, *IEEE Transactions on Intelligent Transportation Systems* **18**(12) (2017), 3540–3548.
- [6] Xiaomeng Shi, Zhirui Ye, Nirajan Shiwakoti and Zheng Li, A review of experimental studies on complex pedestrian movement behaviors, In *Proc. of 15th COTA Int. Conf. of Transp.*, pages 1081–1096, 2015.
- [7] Nicolas Schneider and Dariu M. Gavrilă, Pedestrian path prediction with recursive bayesian filters: A comparative study, In *german conference on pattern recognition*, pages 174–183. Springer, 2013.
- [8] Amin Ullah, Khan Muhammad, Javier Del Ser, Sung Wook Baik and Victor Hugo C. de Albuquerque, Activity recognition using temporal optical flow convolutional features and multilayer lstm, *IEEE Transactions on Industrial Electronics* **66**(12) (2018), 9692–9702.
- [9] Di Tian, Yi Han, Biyao Wang, Tian Guan and Wei Wei, A review of intelligent driving pedestrian detection based on deep learning, *Computational Intelligence and Neuroscience* **2021** (2021).
- [10] Yanqiu Xiao, Kun Zhou, Guangzhen Cui, Lianhui Jia, Zhanpeng Fang, Xianchao Yang and Qiongpei Xia, Deep learning for occluded and multi-scale pedestrian detection: A review, *IET Image Processing* **15**(2) (2021), 286–301.
- [11] Akshay D. Deshmukh, Sanchit Gupta, Ambalika Donge, Piyush Katolkar, Abhishek Tipre and Rakesh A. Bairagi, Review on cars and pedestrian detection, *International Journal of Recent Advances in Multidisciplinary Topics* **2**(6) (2021), 297–300.
- [12] Kamil Roszyk, Michał R. Nowicki and Piotr Skrzypczyński, Adopting the yolov4 architecture for low-latency multispectral pedestrian detection in autonomous driving, *Sensors* **22**(3) (2022), 1082.
- [13] Misbah Ahmad, Imran Ahmed, Kaleem Ullah, Ayesha Khattak, Awais Adnan, et al., Person detection from overhead view: a survey, *International Journal of Advanced Computer Science and Applications* **10**(4) (2019).
- [14] Roshni M. Raval, Harshad B. Prajapati and Vipul K. Dabhi, Survey and analysis of human activity recognition in surveillance videos, *Intelligent Decision Technologies* **13**(2) (2019), 271–294.
- [15] M.V. Athira and Diliya M. Khan, Recent trends on object detection and image classification: A review, In *2020 International Conference on Computational Performance Evaluation (ComPE)*, pages 427–435. IEEE, 2020.
- [16] Naga Venkata Sai Prakash Nagulapati, Sudharsan Reddy Venati, Vishal Chandran and R. Subramani, Pedestrian detection and tracking through kalman filtering, In *2022 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pages 1–6. IEEE, 2022.
- [17] Shreya Bhat, Sphoorti S. Kunthe, Vinuta Kadiwal, Nalini C. Iyer and Shruti Maralappanavar, Kalman filter based motion estimation for adas applications, In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 739–743. IEEE, 2018.
- [18] Mohammed F. Alrifai, Omar Ayad Ismael, Asaad Shakir Hameed and Mustafa B. Mahmood, Pedestrian and objects detection by using learning complexity-aware cascades, In *2021 2nd Information Technology To Enhance e-learning and Other Application (IT-ELA)*, pages 12–17. IEEE, 2021.
- [19] Lie Guo, Ping-Shu Ge, Ming-Heng Zhang, Lin-Hui Li and Yi-Bing Zhao, Pedestrian detection for intelligent transportation systems combining adaboost algorithm and support vector machine, *Expert Systems with Applications* **39**(4) (2012), 4274–4286.
- [20] Vannat Rin and Chaiwat Nuthong, Front moving vehicle detection and tracking with kalman filter, In *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, pages 304–310. IEEE, 2019.
- [21] Xiaogang Chen, Pengxu Wei, Wei Ke, Qixiang Ye and Jianbin Jiao, Pedestrian detection with deep convolutional neural network, In *Asian Conference on Computer Vision*, pages 354–365. Springer, 2014.
- [22] Simon Denman, Vinod Chandran and Sridha Sridharan, An adaptive optical flow technique for person tracking systems, *Pattern Recognition Letters* **28**(10) (2007), 1232–1239.
- [23] Hailong Li, Zhendong Wu and Jianwu Zhang, Pedestrian detection based on deep learning model, In *2016 9th international congress on image and signal processing, biomedical engineering and informatics (cisp-bmei)*, pages 796–800. IEEE, 2016.
- [24] Aysegul Dundar, Jonghoon Jin, Berin Martini and Eugenio Culurciello, Embedded streaming deep neural networks accelerator with applications, *IEEE Transactions on Neural Networks and Learning Systems* **28**(7) (2016), 1572–1583.
- [25] Yonglong Tian, Ping Luo, Xiaogang Wang and Xiaoou Tang, Deep learning strong parts for pedestrian detection, In *Proceedings of the IEEE international conference on computer vision*, pages 1904–1912, 2015.
- [26] Ujwalla Gawande, Kamal Hajari and Yogesh Golhar, Pedestrian detection and tracking in video surveillance system: issues, comprehensive review, and challenges, *Recent Trends in Computational Intelligence*, pages 1–24, 2020.
- [27] Daniela Ridet, Eike Rehder, Martin Lauer, Christoph Stiller and Denis Wolf, A literature review on the prediction of pedestrian behavior in urban scenarios, In *2018 21st Inter-*

- national Conference on Intelligent Transportation Systems (ITSC), pages 3105–3112. IEEE, 2018.
- [28] Michael Goldhammer, Andreas Hubert, Sebastian Koehler, Klaus Zindler, Ulrich Brunsmann, Konrad Doll and Bernhard Sick, Analysis on termination of pedestrians' gait at urban intersections, In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1758–1763. IEEE, 2014.
- [29] Sebastian Koehler, Michael Goldhammer, Sebastian Bauer, Stephan Zecha, Konrad Doll, Ulrich Brunsmann and Klaus Dietmayer, Stationary detection of the pedestrian? s intention at intersections, *IEEE Intelligent Transportation Systems Magazine* **5**(4) (2013), 87–99.
- [30] Christoph G. Keller and Dariu M. Gavrilă, Will the pedestrian cross? a study on pedestrian path prediction, *IEEE Transactions on Intelligent Transportation Systems* **15**(2) (2013), 494–506.
- [31] Nan Bai, Yuan Tian, Ye Liu, Zhengxi Yuan, Zhuoling Xiao and Jun Zhou, A high-precision and low-cost imu-based indoor pedestrian positioning technique, *IEEE Sensors Journal* **20**(12) (2020), 6716–6726.
- [32] Mai Thanh Nhat Truong and Sanghoon Kim, A tracking-by-detection system for pedestrian tracking using deep learning technique and color information, *Journal of Information Processing Systems* **15**(4) (2019), 1017–1028.
- [33] Geetanjali Bhola, Akhil Kathuria, Deepak Kumar and Chandan Das, Real-time pedestrian tracking based on deep features, In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 1101–1106. IEEE, 2020.
- [34] Safaa Dafrallah, Zakaria Sabir, Aouatif Amine, Stephane Mousset and Abdelaziz Bensrhair, Pedestrian walking direction classification for moroccan road safety, In *Proc Int Conf Ind Eng Oper Manage*, pages 1–8, 2020.
- [35] Guangzhe Zhao, Mrutani Takafumi, Kajita Shoji and Mase Kenji, Video based estimation of pedestrian walking direction for pedestrian protection system, *Journal of Electronics (China)* **29**(1) (2012), 72–81.
- [36] Fuqiang Zhou, Lin Wang, Zuoxin Li, Wangxia Zuo and Haishu Tan, Unsupervised learning approach for abnormal event detection in surveillance video by hybrid autoencoder, *Neural Processing Letters* **52**(2) (2020), 961–975.
- [37] Tharindu Fernando, Simon Denman, Sridha Sridharan and Clinton Fookes, Neighbourhood context embeddings in deep inverse reinforcement learning for predicting pedestrian motion over long time horizons, In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.
- [38] Peter Khomchuk, Inna Stainvas and Igal Bilik, Pedestrian motion direction estimation using simulated automotive mimo radar, *IEEE Transactions on Aerospace and Electronic Systems* **52**(3) (2016), 1132–1145.
- [39] Yang Song, Luis Goncalves and Pietro Perona, Unsupervised learning of human motion, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(7) (2003), 814–827.
- [40] Nachiket Deo and Mohan M. Trivedi, Learning and predicting on-road pedestrian behavior around vehicles, In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.
- [41] Yunpeng Chang, Zhigang Tu, Wei Xie and Junsong Yuan, Clustering driven deep autoencoder for video anomaly detection, In *European Conference on Computer Vision*, pages 329–345. Springer, 2020.
- [42] Weina Ge, Robert T. Collins and R. Barry Ruback, Vision-based analysis of small groups in pedestrian crowds, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(5) (2012), 1003–1016.
- [43] Paul Duckworth, Yiannis Gatsoulis, Ferdian Jovan, Nick Hawes, David C. Hogg and Anthony G. Cohn, Unsupervised learning of qualitative motion behaviours by a mobile robot, In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 1043–1051, 2016.
- [44] Amin Moradi, Asadollah Shahbahrami and Alireza Akoushdeh, An unsupervised approach for traffic motion patterns extraction, *IET Image Processing* **15**(2) (2021), 428–442.
- [45] Yutao Han, Rina Tse and Mark Campbell, Pedestrian motion model using non-parametric trajectory clustering and discrete transition points, *IEEE Robotics and Automation Letters* **4**(3) (2019), 2614–2621.
- [46] Imam Husni Al Amin, Falah Hikamudin Arby, et al., Implementation of yolo-v5 for a real time social distancing detection, *Journal of Applied Informatics and Computing* **6**(1) (2022), 01–06.
- [47] Margrit Kasper-Eulaers, Nico Hahn, Stian Berger, Tom Sebulonsen, Oystein Myrland and Per Egil Kummervold, Detecting heavy goods vehicles in rest areas in winter conditions using yolov5, *Algorithms* **14**(4) (2021), 114.
- [48] Jae-Gil Lee, Jiawei Han and Kyu-Young Whang, Trajectory clustering: a partition-and-group framework, In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604, 2007.
- [49] Claude Elwood Shannon, A mathematical theory of communication, *The Bell System Technical Journal* **27**(3) (1948), 379–423.
- [50] Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou and Kai Zheng, A survey of trajectory distance measures and performance evaluation, *The VLDB Journal* **29**(1) (2020), 3–32.