International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Predicting Pedestrian Behavior at Zebra Crossings using Bottom-up Pose Estimation and Deep Learning

Pannalal Boda[1], **Y. Ramadevi** [2]

Submitted: 09/09/2023 **Revised**: 21/10/2023 **Accepted**: 06/11/2023

Abstract: Anticipating pedestrian behavior is critical for traffic management, developing Advanced Driver Assistance Systems (ADAS), and creating autonomous vehicles. However, the unpredictability of pedestrians at zebra crossings poses a significant challenge in designing systems that can aid drivers or enable self-driving. Existing studies often overlook pedestrian behavior and intentions when predicting motion, and there is no integrated system that connects perception and decision-making tasks. To address these challenges, we propose a new bottom-up pedestrian Pose Estimation model based on a CNN network that is trained with the deep learning VGG-19 Pretrained model. This model allows for the analysis of videos captured at zebra crossings and enables the detection and classification of pedestrian poses and movements such as walking, standing, hand signals, crossing, and not crossing. We train and evaluate our models on the pedestrian intention estimation (PIE) dataset using the COCO-18 key point model. Our approach provides a comprehensive solution for predicting pedestrian behavior at zebra crossings. Machine learning-based classifiers are used to compare classification performance across different prediction horizon values, resulting in improved accuracy and efficiency. Our proposed solution has significant implications for traffic management, ADAS, and autonomous vehicles, as it enables them to better anticipate and respond to pedestrian actions. Overall, this study highlights the importance of integrating perception and decision-making tasks in predicting pedestrian behavior and provides a promising solution for addressing this critical problem.

Keywords: *Pedestrians' pose estimation, behavioral analysis, Advanced Driver Assistance Systems, autonomous vehicles, Pedestrian Behavior classification..*

1. Introduction

According to the World Health Organization, road traffic accidents cause 1.35 million deaths each year, with vulnerable road users, including pedestrians [1], accounting for more than half of those killed. With the increasing prevalence of connected autonomous vehicles (CAVs) [2], protecting pedestrians has become even more critical. Predicting the behavior of pedestrians in zebra crossing zones is essential for autonomous vehicle navigation, but it is challenging because pedestrians do not always follow the rules, and their behavior is influenced by other road users. Therefore, accurately categorizing pedestrian behavior is essential for the safety of all road users.

This paper proposes a method for accurately categorizing pedestrian behavior at zebra crossings using computer vision and machine learning techniques. Human activity recognition has various uses in artificial intelligence, including robotics, autonomous cars, surveillance, and help systems [3]. However, to be useful, computers must understand what people are doing. In this context, accuracy is crucial for computer vision systems meant to recognize pedestrians since this will help lower the number of times people are overlooked. Pedestrian detection is an essential function of driver-aid systems, and developing the ability to recognize pedestrian activities is equally crucial.

To address this challenge, the study uses the PIE dataset videos and posture estimation method to identify important landmarks on the bodies of pedestrians. The study [4] uses four machine learning models, namely Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting (GBM), and Extreme Gradient Boosting (XGB), to analyze the data. The study categorizes pedestrian actions, including walking, standing, hand signal, crossing, and not crossing, using improved sparse optical flow to identify moving objects. The optimal model has an AUC of 0.922, and multiple timescales are used to make predictions [5].

Autonomous vehicles are built with safety as an integral design principle to ensure the safety of all road users, including pedestrians and drivers of other cars. The proposed technique can be implemented in autonomous driving systems to recognize pedestrian crossings and categorize pedestrian actions. By giving drivers more information about how pedestrians act at zebra crossings, this study aims to help them avoid accidents. Improved

1Research Scholar, CSE Department
Osmania, University,Hyderabad, Telangana, India.
E-mail: bpannalal555@gmail.com
2Professor, CSE Dept, Chaitanya Bharathi Institute of Technology, Osmania University,
Hyderabad, Telangana, India.
E-mail: yramadevi_cse@cbit.ac.in

pedestrian activity recognition systems can significantly reduce the risk of accidents involving pedestrians and other road users.

The primary contribution of this research paper can be summarized as follows:

1. A method for detecting and tracking pedestrians in a video surveillance system using a bounding box approach.
2. A deep learning-based bottom-up approach is employed to detect pedestrian poses, utilizing Skeleton key point extraction.
3. To enhance behavioral analysis, a combination of OpenPose pose estimation model, CNN architecture, and pre-trained VGG-19 model is utilized.
4. Machine learning-based classifiers are utilized to compare the effectiveness of classification by varying prediction horizon values.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 discusses the methodology and dataset, while Section 4 describes the proposed research work in terms of functional blocks. Section 5 presents the findings and analyses. Section 6 describes the work's outcomes and future scope.

2. Related Work

2.1Identification of pedestrian at crossing

Detecting pedestrian crossing intent is crucial for the successful implementation of fully autonomous cars in urban areas. The research in this field mainly focuses on detecting pedestrian intentions through detection and tracking techniques. In one study [6], the authors developed a pedestrian crossing intention model and compared it with the standard SVM approach. Both models showed good identification accuracy, but the AT-LSTM model outperformed with an accuracy of over 96.15% in predicting pedestrian intentions at crosswalks. Another research article [7] proposed a real-time pedestrian identification system that included a deep learning classifier, methods to recognize zebra crossings, and a dual camera mechanism, which worked effectively. The HOG+SVM method suggested in reference [8] is widely recognized as a powerful characteristic for pedestrian detection due to its ability to accurately identify the human-specific, stable head-shoulder and inconsistent lower body look. Another approach [9] proposed a fast object detector that relied on boosted cascades of basic features. These two approaches have laid the foundation for future advancements in pedestrian detection methods that can achieve "real-time" detection. However, recent studies have shown that deep learning-based algorithms can identify pedestrians, but with lower accuracy.

2.2 Pedestrian Pose Estimation

The recognition of human postures is possible through 2D and 3D pose estimation, as discussed in previous research [10]. 2D posture estimation involves calculating the X and Y coordinates of body joints in 2D space based on input data, such as an image or video frame. On the other hand, 3D pose estimation adds a Z dimension to the 2D image, allowing for the prediction of an object's exact spatial location. While deep neural networks (DNNs) are effective at predicting single-person poses, they struggle with multiple-person poses, which can significantly increase computational complexity and real-time inference time. To address this issue, the researchers proposed two techniques: top-down and bottom-up. Traditional 2D human pose estimation methods rely on hand-crafted feature extraction methods for each body component. In contrast, early computer vision methods used a stick figure description of the human body to derive global pose structures. However, recent deep learning-based methods, such as OpenPose [11], Cascaded Pyramid Networks (CPN) [12], AlphaPose [13], and HRNet [14], have made substantial progress in solving this problem, leading to significant improvements in performance for both single- and group-based pose estimation. In the field of behavior analysis, observing and understanding a scenario are crucial for predicting the actions of pedestrians in traffic incidents. Traditionally, predicting a behavior's trajectory, velocity, or other characteristics, and learning its underlying mechanics, such as how individuals walk, run, stand, etc., have dominated this field.

2.2.1 Pose estimation with Deep learning

In recent times, deep learning has been proven to be more effective than traditional computer vision methods for various tasks, such as object detection and image segmentation. The use of deep learning techniques has significantly enhanced posture estimation performance. One popular approach to posture estimation is the regional multi-person pose estimate (AlphaPose), which predicts human postures from bounding boxes. This approach can be applied to both still images and videos and is especially useful when bounding boxes are inaccurate.

The regional multi-person pose estimation (RMPE) framework was developed by the author of to estimate postures using imprecise bounding boxes [15]. The suggested architecture includes symmetric spatial transformer network (SSTN) [16], parametric poses non-maximum suppression (NMS) [17], and pose-guided proposal generator (PGPG), which handle incorrect bounding boxes and duplicate detections, resulting in a 17% improvement over the MPII dataset. Another model uses balanced Gaussian process dynamical models (B-GPDM) and Naive Bayes classifiers to predict pedestrian location and poses [18]. A hidden Markov model recognizes intentions based on the 3D positions and displacements of 11 joints along the pedestrian's body [19].

In reference the author employs convolutional neural networks for estimating multi-person human positions from videos with high precision and low background [20]. Similarly, in reference the author demonstrates how monocular vision-based human position estimation with deep CNNs can be used to identify vulnerable road users (VRUs) by their intentions [21].

Pose estimation can have a positive impact on pedestrian wellbeing, as it can be used to identify vital anatomical landmarks [22]. Convolutional neural networks can also be used to identify important frames from videos, recognize activity, and identify distracted driving through the detection of hand, face, and upper-body poses [23-25]. Pose estimation has also been successful in predicting pedestrians' crossing intentions.

2.2.2 Pedestrians' Crossing Intention Prediction

Traditionally, predicting a pedestrian's desire to cross a street has been considered as part of "trajectory prediction." For interior localization, Wi-Fi and Bluetooth have been commonly used, while cameras and LiDAR are frequently used on the road [26]. Table I summarizes the different research comparisons. Various studies have employed cameras to predict pedestrians' crossing intentions or trajectories. Machine learning models such as Support Vector Machines (SVMs) and deep learning models like Long Short-Term Memory (LSTM) are widely used in addition to the Kalman Filter (KF) and Gaussian Process Dynamical Models (GPDMs) [27-29]. According to LSTM was utilized to predict when pedestrians and cyclists would cross the street [30]. In the authors proposed a different approach to detecting pedestrian intent by forecasting pedestrian behavior at a crosswalk using single-view photos [31]. They utilized a convolutional neural network (CNN) for identification, tracking, and position estimation. Authors also utilized the angle between joints to determine the horizontal direction, similar to how posture estimation is performed. In contrast, deep neural networks were utilized for both independent and joint behavior analysis, forecasting five patterns of conduct using a Bayesian inference system (crossing, stopping, starting, etc.) [32]. The authors determined the dimensions of their joints using their postures to determine their lengths, angles, rotational speeds, and linear velocities. The kinematic variables of pedestrians were found to be more accurate and reliable than inertial measuring units (IMU).

The Theory of Behavior (TPB) model and other behavioral and statistical models were used to examine pedestrian crossing intentions [33]. Demographics such as age and gender, as well as socioeconomic characteristics like foot traffic and pedestrians' understanding of the risks they face, are critical aspects of pedestrians' lives. It is important to consider pedestrians' unique characteristics in the evaluation. Furthermore, the final few seconds before crossing the road are when pedestrians are most likely to change their minds and alter their crossing intention [34].

2.3 Pedestrian Behavior Classification at crossing

The behavior of pedestrians is influenced by a multitude of variables, making it highly variable. To investigate the impact of demographics, mobile phone usage, and walking speed on pedestrian behavior, it is necessary to classify it. By doing so, we gain insights into how people react in various situations, enabling us to comprehend their movements better. This approach provides us with a more intricate and nuanced perspective of pedestrian behavior, allowing us to develop a deeper understanding of the factors that influence it.

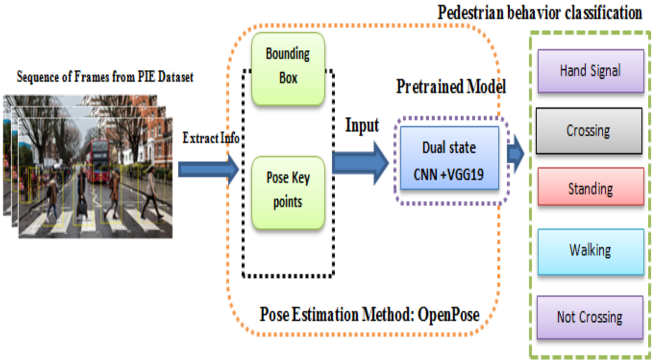

Fig 1 Understanding Outline of the Proposed Model

3. Methodology

In order to detect pedestrian activity at Zebra crossings, a series of steps are required. Initially, the input video is divided into individual frames, which are then processed by a pedestrian detection module. This module utilizes bounding box detection techniques to isolate pedestrians in each frame. The resulting output from the pedestrian detection module is then forwarded to a posture estimation module. This module employs a deep learning-based model known as OpenPose to estimate the positions of pedestrians' bodies. Finally, machine learning classifiers are utilized to determine whether a pedestrian intend to walk, stands, or signal using hand gestures.

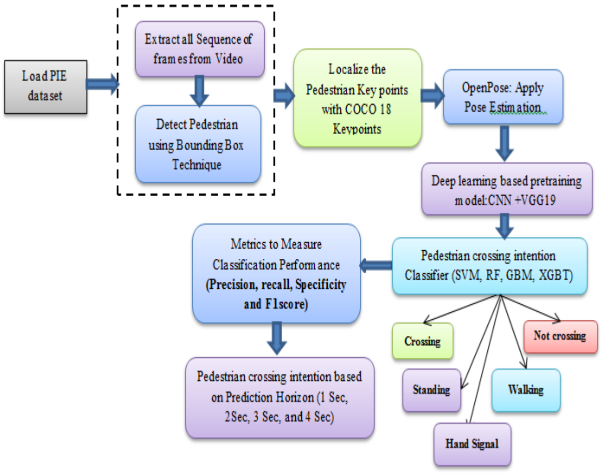
Load PIE dataset
Extract all Sequence of frames from Video
Detect Pedestrian using Bounding Box Technique
Localize the Pedestrian Key points with COCO 18 Keypoints
OpenPose: Apply Pose Estimation
Deep learning based pretraining model:CNN +VGG19
Pedestrian crossing intention Classifier (SVM, RF, GBM, XGBT)
Metrics to Measure Classification Performance (Precision, recall, Specificity and F1score)
Pedestrian crossing intention based on Prediction Horizon (1 Sec, 2Sec, 3 Sec, and 4 Sec)
Crossing
Not crossing
Standing
Walking
Hand Signal

Fig 2 Block diagram of the Proposed Work

3.1 Data Collection

Pedestrian Intention Estimation (PIE) It's a brand new dataset focused on pedestrian traffic that can actually estimate intent and predict trajectory. PIE consists of over 6 hours of onboard camera traffic footage, along with synchronized OBD vehicle data such as speed, heading direction, and GPS coordinates. The dataset provides rich spatial and behavioral annotations, including traffic lights, signs, and zebra crossings, to characterize interactions between pedestrians and vehicles. With over 300,000 labeled video frames of pedestrians, PIE is one of the largest datasets available for studying pedestrian behavior. In Table 1, the dataset reveals that out of 894 people, 519 pedestrians and 429 pedestrians have no intention of crossing the street, respectively.

Table 1. Quantitative analysis of PIE behavioral annotations

Compilation of all frames	909,480
Sum of all frames with annotations	293,437
Number of persons whose actions have been annotated	1842
Bounding-box count for pedestrians	738,970
Average length of pedestrian track	401 frames
Pedestrian counts	
Number of pedestrians who intend to but do not cross	894
Number of pedestrians with no crossing intention	429
Number of pedestrians to cross who eventually cross in front of the vehicle	519

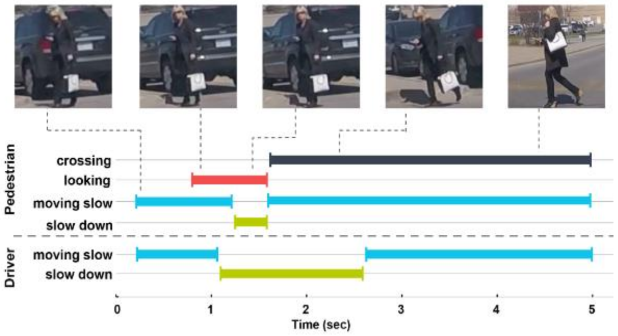

Fig 3 Pedestrian into their relevant classes for each frame

In this research paper, the methodology involved assigning each cropped pedestrian to a corresponding frame class. This process was demonstrated in Fig. 3, where the start and end frames were identified for each behavior. Some behaviors shared the same class, and in such cases, instead of attempting to determine the class, multiple copies of the image were created. This approach was used for behaviors such as walking, standing, displaying a hand signal, crossing, and not crossing.

3.2 Annotations

The methodology of this work involves the analysis of pedestrian intention estimation at zebra crossings. Table 2 is used to present spatial annotations that denote the behavioral classes and associated annotations. These annotations are used to estimate pedestrian intentions accurately.

Table 2. PIE dataset features with class labels

Type of Object	**Spatial Annotations**	**Annotations**	**Behavioral Class (“0”) or (“1”)**
Pedestrian	zebra crossings	Action	Walking (‘0”)
			Standing (“1”)
		Gesture	Hand signal (“0”)
			No hand Signal (“1”)
		Look	Looking (‘0”)
			Not Looking (“1”)
		Cross	Crossing (‘0”)
			Not Crossing (“1”)

Table 3. Observed Behaviour classes upon Pedestrian Count

Behavioral Class	**Pedestrian Count**
Walking	150
Standing	279
Hand signal	60
Crossing	519
Not Crossing	834

3.3 Extraction of Bounding Box Information

Let each provided example be denoted by e_i, where i is the index of the example. Each example e_i contains a top-left bounding box coordinate $(x1_i, y1_i)$ and a bottom-right bounding box coordinate $(x2_i, y2_i)$. These coordinates can be represented as a 4-dimensional vector $b_i = [x1_i, y1_i, x2_i, y2_i]$.

Furthermore, each bounding box b_i is associated with an occlusion tag o_i, which can take on one of three values: 0 (no occlusion), 1 (occlusion of at least 25%), or 2 (complete occlusion of at least 75%). Therefore, the occlusion tag for example e_i can be represented as a scalar value o_i. To extract meaningful information from the data, we consider the context of the video, including the day, time, and place denoted by c_i. Additionally, we consider the pedestrian's activities a_i (e.g., walking, looking) and physical characteristics p_i (e.g., posture, clothing, and accessories). Each example e_i has a single label annotation that includes these attributes:

$$l_i = (c_i, b_i, o_i, a_i, p_i) \quad (1)$$

The dataset also includes frame-specific traffic data, such as road signs and light timings. These annotations are single-frame annotations that capture the movements and accelerations of vehicles as seen in each individual frame.

$$t_i = (d_i, s_i, r_i, l_i) \quad (2)$$

where d_i represents the time and date of the frame, s_i represents the speed of the vehicles in the frame, r_i represents the road conditions, and l_i represents the label annotation for the pedestrians in the frame.

Algorithm -1 Extracting bounding box information

Input: Examples $E = \{e_1, e_2, \dots, e_n\}$ containing top-left and bottom-right bounding box coordinates and occlusion tags.

Output: Extracted bounding box information B = $\{b_1, b_2, \dots, b_n\}$.

```
1. Initialize an empty list $B$.
2. For each example $e_i$in $E$:
    2.1 Extract the top-left bounding box coordinate $(x1_i, y1_i)$and the bottom-right bounding box coordinate$(x2_i, y2_i)$.
    2.2 Store the coordinates as a 4-dimensional vector $b_i = [x1_i, y1_i, x2_i, y2_i]$.
    2.3 Extract the occlusion tag $o_i$for the bounding box.
    2.4 Store the bounding box $b_i$and its associated occlusion tag $o_i$as a tuple $(b_i, o_i)$.
    2.5 Append the tuple to the list $B$.
3. Return the extracted bounding box information $B$.
```

3.3.1 Pedestrian Detection and Tracking using Bounding Box Approach

The method involves using computer vision algorithms to detect and track pedestrians in a video surveillance system. The approach is based on creating bounding boxes around the pedestrians and then tracking those boxes over time.

Mathematical Model:

Let us assume that we have a video sequence consisting of N frames, where each frame is represented as $I(n)$, where n is the frame index. Each frame $I(n)$ has a set of bounding boxes $B(n) = \{B1(n), B2(n), \dots, BM(n)\}$ where M is the number of pedestrians detected in the frame. Each bounding box is defined by its top-left corner (x, y), its width w and its height h.

Each bounding box $B(n)$ is represented by a 4-dimensional vector $b(n, m) = [x, y, w, h]T$, where T denotes the transpose of the vector. The tracking algorithm attempts to find the correspondence between the bounding boxes in adjacent frames, so that we can track the same pedestrians over time.

Let us define the correspondence matrix C, where $C(i, j) = 1$ if the bounding box $Bi(n)$ in frame n corresponds to bounding box $Bj(n + 1)$ in frame $n + 1$, and $C(i, j) = 0$ otherwise. We want to find the optimal correspondence matrix C that minimizes the total cost of tracking over all frames.

The cost of tracking is defined as the sum of the localization cost and the temporal cost. The localization cost is the distance between the predicted location of a bounding box in the next frame and the actual location of the bounding box. The temporal cost penalizes large changes in the bounding box size or location over time.

The total cost of tracking is defined as:

$$E(C) = \alpha * E_{loc(C)} + \beta * E_{temp(C)} \quad (3)$$

where α and β are weighting factors that control the relative importance of the localization and temporal costs, respectively.

The localization cost $E_{loc(C)}$ is defined as:

$$E_{loc(C)} = \sum i,j\, c(i,j) * D\big(Bi(n), Bj(n+1)\big) \qquad (4)$$

where $D\big(Bi(n), Bj(n+1)\big)$ is the Euclidean distance between the center of the bounding box $Bi(n)$ in frame n and the predicted location of the bounding box $Bj(n+1)$ in frame $n+1$.

The temporal cost $E_{temp(C)}$ is defined as:

$$E_{temp(C)} = \sum i,j\, \big(1 - c(i,j)\big) * S\big(Bi(n), Bj(n+1)\big)$$
(5)

where $S\big(Bi(n), Bj(n+1)\big)$ is a similarity function that measures how similar the bounding boxes $Bi(n)$ and $Bj(n+1)$ are in terms of their size and location.

The optimization problem can be solved using the Hungarian algorithm, which finds the optimal assignment of the bounding boxes in adjacent frames that minimizes the total cost of tracking. In summary, the method for detecting and tracking pedestrians in a video surveillance system using a bounding box approach involves creating bounding boxes around pedestrians in each frame, and then using a tracking algorithm to find the optimal correspondence between the bounding boxes in adjacent frames. The correspondence is found by minimizing the total cost of tracking, which is a combination of the localization and temporal costs. The optimization problem can be solved using the Hungarian algorithm.

3.4 A Deep Learning-based bottom-up approach: Skeleton key point extraction and pedestrian poses detection

3.4.1 Skeleton Data Extraction and Pose Detection

In this study, we utilized OpenPose to estimate the poses of pedestrians in sequences. OpenPose is capable of simultaneously estimating 15-18, or 25 key points in the body and feet in real-time. The 18-key point skeleton model consists of actual human body joints as shown in figure 4, including a nose (0), a neck (1), right and left shoulders (2,5), right and left elbows (3,6), right and left wrists (4,7), right and left hips (8,11), right and left knees (9,12), right and left ankles (10,13), and both eyes (14,15). Although variations in foot, ear, and eye movement can impact the evaluation of action quality, we limited our analysis to only 14 different joint movements. The absence of analysis-related senses (N = 0) was also noted. To isolate the target performer's skeleton data from background noise, we employed scale computation and a key point confidence comparison after OpenPose provided multi-person skeleton identification findings. If the body failed to move due to occlusion or self-occlusion, joint coordinates were reset to zero, and linear interpolation was applied to fill in the gaps between frames to capture absent skeletal data. We found that deep learning improved pedestrian posture detection and was a promising option for understanding pedestrian behavior. OpenPose was selected as the preferred posture estimator due to its fast and accurate multi-person posture estimation capability, providing confidence scores for each key point. We trained our model on a dataset of 1,842 pedestrian samples, divided into training, test, and validation sets with 50%, 40%, and 10% respectively.

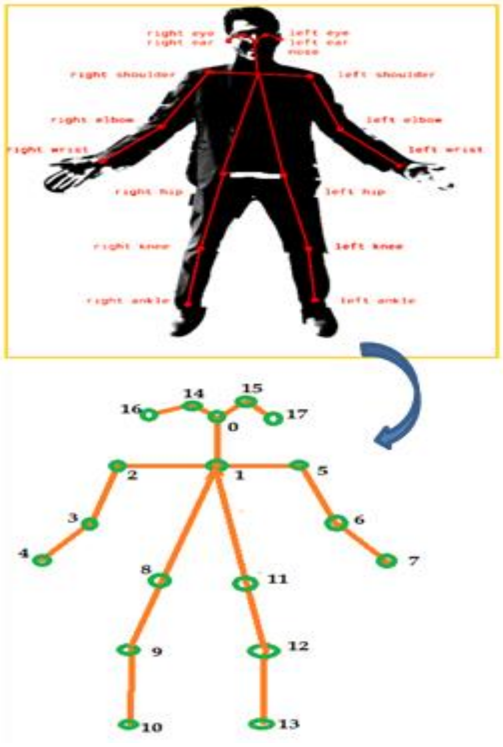

Fig 4 Pedestrian Key point detection and Transformation

Algorithm 2: Estimated poses of pedestrians in the video sequences

Input: Video sequences of pedestrians in motion

Output: Estimated poses of pedestrians in the video sequences

1. Initialize OpenPose for simultaneous estimation of K key points in the body and feet of pedestrians in real-time.

2. For each frame in the video sequences:

a. Apply OpenPose to obtain multi-person skeleton identification findings, resulting in an array of size (N x K x 3), where N is the number of detected individuals, K is the number of key points, and 3 represents the (x, y, confidence) values for each key point.

b. For each individual in the array, employ scale computation and key point confidence comparison to isolate the target performer's skeleton data from background noise, resulting in an array of size (1 x K x 3).

c. If a key point is missing, reset the corresponding (x, y) values to (0, 0), and apply linear interpolation to fill in the gaps between frames, resulting in an array of size (1 x K x 3).

d. Use the resulting skeletal data to estimate the poses of pedestrians in the video sequences, resulting in an array of size (N x 14), where 14 represents the selected joint movements used for analysis.

3. Train a deep learning model using the dataset of 1,842 pedestrian samples, divided into training, test, and validation sets.

4. For each new video sequence:

a. Apply the trained model to estimate the poses of pedestrians in the video sequence, resulting in an array of size (N x 14).

b. Analyze the estimated poses to understand pedestrian behavior.

The bottom-up approach used in pedestrian pose detection involves detecting body parts in an image and then grouping them together to form full-body poses. Let's denote an input image as I with dimensions$W \ x \ H$, where W is the width of the image and H is the height of the image. The goal is to detect the locations of body parts, which include the head, shoulders, elbows, wrists, hips, knees, and ankles, in the image.

The bottom-up approach starts by detecting these individual body parts using convolutional neural network (CNN) architecture. Let's denote the set of detected body parts as B. Each body part b in B is represented as a $2D$ coordinate (x_b, y_b), where x_b and y_b are the pixel coordinates of the body part in the image. Thus, the set of detected body parts B can be represented as:

$$B \ = \ \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \qquad (6)$$

where n is the number of detected body parts.

Once the body parts are detected, the next step is to group them together to form full-body poses. This is done using a "part affinity field" (PAF) network, which predicts the likelihood that two body parts belong to the same limb. Let's denote the set of detected limbs as L. Each limb l in L is represented as a pair of body part indices (i, j), where i and j are indices of the two body parts that form the limb. Thus, the set of detected limbs L can be represented as:

$$L \ = \ \{(i_1, j_1), (i_2, j_2), \dots, (i_m, j_m)\} \qquad (7)$$

where m is the number of detected limbs.

The PAF network predicts two maps for each limb: an x-direction PAF and a y-direction PAF. Let's denote the set of x-direction PAFs as X and the set of y-direction PAFs as Y. Each PAF map p in X or Y is represented as a 2D vector field, where each vector indicates the direction and magnitude of the limb at each pixel in the image. Thus, the sets of x-direction and y-direction PAFs can be represented as:

$$X \ = \ \{p_1, p_2, \dots, p_m\} Y \ = \ \{q_1, q_2, \dots, q_m\} \qquad (8)$$

where each p_i and q_i is a $W \ x \ H$ vector field.

To group the body parts together to form full-body poses, the PAF maps are used to create a "confidence map" for each body part, which indicates the likelihood that the body part is part of a limb. Let's denote the set of confidence maps as C. Each confidence map c in C is represented as a 2D heat map, where each pixel indicates the likelihood that the body part is part of a limb. Thus, the set of confidence maps C can be represented as:

$$C \ = \ \{c_1, c_2, \dots, c_n\} \qquad (9)$$

where each c_i is a $W \ x \ H$ heat map.

Finally, the body parts are grouped together to form full-body poses using a greedy algorithm that iteratively adds body parts to the pose based on the highest confidence score. The resulting set of poses can be denoted as P:

$$P \ = \ \{(x_1, y_1, \dots, x_k, y_k), (x_1', y_1', \dots, x_k', y_k'), \dots, \left(x_p, y_p, \dots, x_k^p, y_k^p\right)\}$$
(10)

where each pose is a set of body parts

3.5 Understanding the Pedestrian Behaviour

Let V be the video data and P be the set of pedestrian behavior tracks in V. We classify each behavior track p in P into one of the following categories: walking, standing, hand signaling, crossing, and not crossing. We use OpenPose to extract a set of key points for each pedestrian behavior track p. Let K be the set of key points extracted from P.

We obtain a set of joint trajectories for each pedestrian behavior track p by applying video skeleton detection to K. Let $J(p)$ be the set of joint trajectories obtained for p. Each joint trajectory j in $J(p)$ is represented as a set of joint positions in a discrete 3D space, denoted by $j \ = \ \{j_t | \ t = 1, \dots, T\}$, where T is the number of frames in the video V and $j_t$represents the joint position at time t.

To capture the spatial context of joints, we take a localized $3D$ patch at the location of each joint. Let V_j^kbe the joint motion volume for joint k in trajectory j of pedestrian behavior track p. $V_j^k$is created by combining the key point corresponding to joint k and the time parts of the video in which it appears. We apply $2D$ Gaussian smoothing to $V_j^k$to remove noise data caused by joint position failure or false detection, resulting in a smoothed joint motion volume v_j^k.

We calculate the central moment features for each super pixel in the smoothed joint motion volume v_j^k. Let $F\left(v_j^k\right)$be the set of central moment features obtained for v_j^k.

We combine the features of all super pixels in the smoothed joint motion volume $v_j^k$to create the motion feature for joint k in trajectory j of pedestrian behavior track p, denoted by $F_k^m, j(p)$. We then combine the motion features for all joints in $J(p)$ to create the spatiotemporal feature description of the action instance in the video V, denoted by $F^{s(p)}$.

We divide the sequences into different lengths, with most experiments focusing on three-frame sequences. Arm and leg bones are described by segments between key points 5-17, and angles formed by these lines and the horizontal are used as input features. The angles between the bones and the horizontal line are also included. Two more angles are calculated between keypoints 11, 13, and 15, and keypoints 12, 14, and 16 to account for the right and left angles at the knee.

The above mathematical model represents the methodology used to extract and process pedestrian behavior data from video sequences, with a focus on posture estimation and joint trajectory analysis using OpenPose and video skeleton detection techniques. The resulting spatiotemporal feature descriptions of pedestrian actions can be used for machine learning applications in the field of computer vision.

3.6 Behavioral Analysis with OpenPose, CNN, and VGG-19

Pedestrian behavioral analysis at zebra crossings is a challenging task due to the unpredictability of pedestrians. To address this, we can use a combination of computer vision and deep learning techniques to detect and classify pedestrian poses and movements as shown in Figure 5.

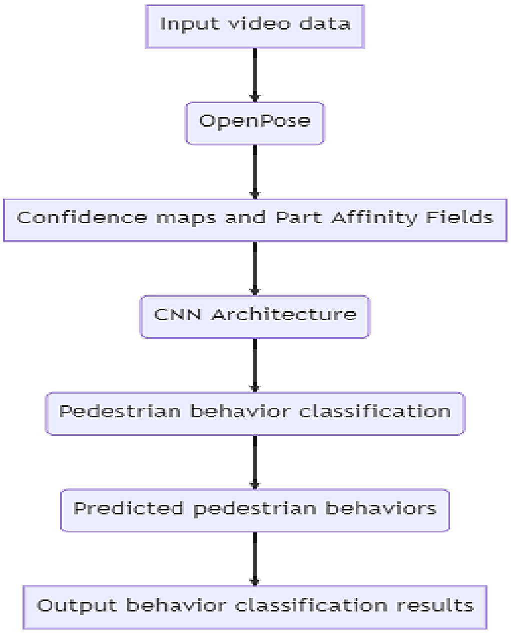
Input video data
OpenPose
Confidence maps and Part Affinity Fields
CNN Architecture
Pedestrian behavior classification
Predicted pedestrian behaviors
Output behavior classification results

Fig 5 Flow model of Behavioral Analysis with OpenPose, CNN, and VGG-19

3.6.1 OpenPose

OpenPose is a popular computer vision technique for human pose estimation that can be used to detect and locate the body parts of pedestrians in video data. The output of OpenPose is a set of confidence maps and Part Affinity Fields (PAFs) that represent the location of each body part and the association between them. These confidence maps and PAFs can be used as inputs to CNN architecture for pedestrian behavior classification.

Let I be an input image or frame of size$W\ x\ H\ x\ 3$, where W and H are the width and height of the image, and 3 represents the three color channels (R, G, B).

The VGG-19 convolutional network is used to extract features from the input image I, and its output is passed to two branches:

1. *Confidence Maps:* The first branch predicts a set of confidence maps C of size $W\ x\ H\ x\ N$, where N is the number of body parts to be detected. Each confidence map c_i, for $i\ =\ 1$ to N, represents the probability of finding the $i-th$ body part at each pixel location (x, y) in the image.

2. ***Part Affinity Fields****:* The second branch predicts a set of Part Affinity Fields P of size $\ x\ H\ x\ 2M$, where M is the number of pairs of body parts to be associated. Each Part Affinity Fieldp_j, for $j\ =\ 1$ to $2M$, represents the likelihood of two body parts being connected. Specifically, for each pixel location (x, y), the Part Affinity Field $p_{j(x,y)}$ is a two-dimensional vector that indicates the direction and strength of the connection between the two body parts.

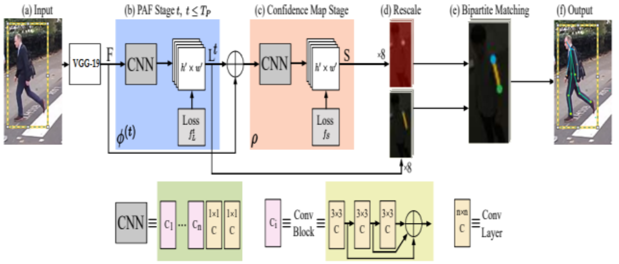

Fig 6 OpenPose network Architecture

The OpenPose network as shown in figure 6 is designed to estimate the location of body parts in an image by creating a set of detection confidence maps and a set of part affinity fields. The network consists of multiple stages, with each stage refining the results of the previous stage. The confidence maps correspond to each joint and have the same size as the input image. The authors of the paper use two loss functions to train the network, one for each branch. They employ a regular L2 loss to compare the estimated predictions to the ground truth maps and fields. OpenPose has 19 body parts and 19 "limbs" or body-to-body connections. The weight function W(p) represents the mask that protects true positive predictions during training. Finally, the overall objective is obtained by combining the two loss functions.

Mathematical model for applying OpenPose to each frame of the video data:

1. Define the OpenPose model architecture as a CNN with input tensor X and output tensors C and F, representing the confidence maps and PAFs, respectively: $C, F = OpenPose(X)$
2. Load the pre-trained OpenPose model into memory.
3. For each frame of the video data, read in the frame and convert it to a tensor X suitable for the OpenPose model.
4. Pass the tensor X through the OpenPose model to obtain the confidence maps C and $PAFs\ F: C, F = OpenPose(X)$
5. Use the confidence maps C and PAFs F to estimate the location of body parts in the image or video frame.
6. Repeat steps 3-5 for each frame of the video data.

3.6.2 CNN Architecture with VGG-19

The CNN architecture can be trained to classify different pedestrian poses and movements, such as walking, standing, hand signals, crossing, and not crossing. The VGG-19 pre-trained model can be used as a feature extractor to extract high-level features from the input image or frame. The output of the VGG-19 model can be concatenated with the confidence maps and PAFs from OpenPose to form a comprehensive feature vector for pedestrian behavior classification.

The CNN architecture consists of several layers, including the convolutional layer that generates a feature map by filtering the image several pixels at a time, the pooling layer that minimizes the size of the data produced by the convolutional layer for efficient storage, and fully connected input, layer, fully connected layer, and fully connected output layer, which incorporate scores into the feature analysis's outputs and yield conclusive probabilities for labeling the image's category. To ensure robust and accurate predictions, a network uses a combination of forward and backward propagation to iteratively examine all of the training samples in the network and determine the optimal weights.

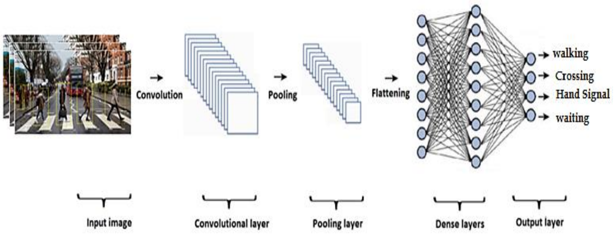

Fig7: CNN Architecture

To ensure that only the most robust and accurate neurons are activated when making a prediction, a network uses a combination of forward and backward propagation to iteratively examine all of the training samples in the network and determine the optimal weights.

The CNN architecture can be represented as a set of layers, including convolutional layers, pooling layers, and fully connected layers. Let W be the set of weights and b be the set of biases for each layer in the CNN architecture. The output of the CNN architecture for an input feature vector X can be represented as

$$Y = CNN(X; W, b). \qquad (11)$$

The overall pedestrian behavior classification model can be represented as a function $f(I)$ that takes an input image or video frame I and outputs a set of predicted pedestrian behaviors $B = \{b1, b2, \ldots, bn\}$. The predicted pedestrian behaviors can be represented as binary labels or probabilities for each behavior class.

To train the pedestrian behavior classification model, we can use a labeled dataset of human behaviors. Let $D = \{(I1, B1), (I2, B2), \ldots, (IN, BN)\}$ be a labeled dataset of N image or video frame samples, where each sample has an associated set of ground-truth pedestrian behaviors.

The objective of training the model is to minimize the classification error between the predicted behaviors and the ground-truth behaviors. This can be achieved by minimizing the cross-entropy loss function:

$$L = -\frac{1}{N}\sum\left(bi \log\log\left(f(Ii)\right) + (1 - bi) \log\log\left(1 - f(Ii)\right)\right) \qquad (12)$$

where bi is the ground-truth label for the $i - th$ image or video frame sample, and $f(Ii)$ is the predicted probability or label for the i-th sample.

The model can be trained using stochastic gradient descent (SGD) or a similar optimization algorithm to update the weights and biases of the CNN architecture. The performance of the model can be evaluated using metrics such as accuracy, precision, and recall on a separate validation set. In summary, integrating OpenPose, CNN architecture, and pre-trained VGG-19 model can enhance the pedestrian behavioral analysis

Algorithm -3: Pedestrian Behavior Prediction Algorithm using OpenPose, VGG-19, and CNN.

Input: Labeled dataset of human behaviors, video data

Output: Predicted pedestrian behaviors for each frame of the video data

1. Apply OpenPose to each frame of the video data to estimate the location of body parts:

   a. Let I be an input image or video frame.

   b. Use OpenPose to generate confidence maps C and Part Affinity Fields (PAFs) F for each body part in the image or frame.

   c. Output the set $S = \{C, F\}$ for each frame of the video data.

2. Use a pre-trained VGG-19 model to extract high-level features from the input image or frame:

   a. Let I be an input image or video frame.

   b. Apply the pre-trained VGG-19 model to the input image or frame to extract high-level features VGG-19(I).

   c. Output VGG-19(I) for each frame of the video data.

3. Train a CNN to classify pedestrian poses and movements:

   a. Collect a labeled dataset of human behaviors.

   b. Split the dataset into a training set and a validation set.

   c. Train the CNN on the training set using backpropagation and gradient descent, with the input features $X = \{S, VGG - 19(I)\}$ and the ground-truth pedestrian behaviors as the output labels Y.

   d. Evaluate the performance of the trained CNN on the validation set.

4. Use the trained CNN to predict pedestrian behaviors for each frame of the video data:

   a. For each frame of the video data, apply OpenPose to estimate the location of body parts and use the pre-trained VGG-19 model to extract high-level features.

   b. Use the trained CNN to predict the pedestrian behaviors for each frame of the video data.

5. Evaluate the performance of the pedestrian behavior prediction algorithm:

   a. Apply the algorithm to the validation dataset to predict the pedestrian behaviors.

   b. Evaluate the accuracy and performance of the algorithm on the validation dataset.

Note: The following notations are used in the algorithm:

- I: input image or video frame
- C: confidence maps generated by OpenPose
- F: Part Affinity Fields generated by OpenPose
- S: set of confidence maps and Part Affinity Fields, i.e., $S = \{C, F\}$
- VGG-19(I): high-level features extracted from the input image or frame using a pre-trained VGG-19 model
- X: input features for the CNN, i.e., $X = \{S, VGG - 19(I)\}$
- Y: ground-truth pedestrian behaviors
- CNN: Convolutional Neural Network used for pedestrian behavior classification

Pseudocode: Pedestrian Behavior Prediction Algorithm using OpenPose, VGG-19, and CNN.

Input: Input image or video frame

Output: Predicted behaviors: predicted pedestrian behaviors for each frame of the video data

```
Function OpenPose(Frame):
Let C, F be confidence maps and Part Affinity Fields
        generated by OpenPose for Frame
Return S = {C, F}
End Function
Function VGG19Features(Frame):
Let Features be high −
level features extracted from Frame using a pre −
trained VGG − 19 model
Return Features
End Function
Function TrainCNN(TrainingSet):
Split TrainingSet into a training set and a validation set
```

```
$Let\ X = \{S, VGG19Features(I)\} be\ the\ input\ features\ for\ the\ CNN$
$Let\ Y\ be\ the\ ground - truth\ pedestrian\ behaviors$
$Train\ the\ CNN\ on\ the\ training\ set\ using\ back\ propagation\ and\ gradient\ descent\ with\ X\ and\ Y\ as\ input\ and\ output, respectively$
$Evaluate\ the\ performance\ of\ the\ trained\ CNN\ on\ the\ validation\ set$
$End\ Function$
$Function\ PredictBehaviors(VideoData):$
    $Let\ PredictedBehaviors\ be\ an\ empty\ list$
    $For\ each\ frame\ of\ VideoData\ do$
        $Let\ S = OpenPose(Frame)$
        $Let\ Features = VGG19Features(Frame)$
        $Let\ InputFeatures = \{S, Features\}$
        $Let\ Prediction = CNN(InputFeatures)$
        $Add\ Prediction\ to\ PredictedBehaviors$
    $End\ For$
    $Return\ PredictedBehaviors$
$End\ Function$
$Function\ EvaluatePerformance\left(\begin{matrix}Algorithm,\\ ValidationSet\end{matrix}\right):$
    $Let\ PredictedBehaviors = Algorithm(ValidationSet)$
    $Evaluate\ the\ accuracy\ and\ performance\ of\ the\ algorithm\ on\ the\ validation\ dataset$
$End\ Function$
```

Mathematical Model for above algorithm: 3

Step 1: Mathematical model for applying OpenPose to each frame of the video data

1. Define the OpenPose model architecture as a CNN with input tensor X and output tensors C and F, representing the confidence maps and PAFs, respectively: C, F = OpenPose(X)
2. Load the pre-trained OpenPose model into memory.
3. For each frame of the video data, read in the frame and convert it to a tensor X suitable for the OpenPose model.
4. Pass the tensor X through the OpenPose model to obtain the confidence maps C and PAFs F: C, F = OpenPose(X)
5. Use the confidence maps C and PAFs F to estimate the location of body parts in the image or video frame.
6. Repeat steps 3-5 for each frame of the video data.

Step 2: Use a pre-trained VGG-19 model to extract high-level features from the input image or frame

1. Let I be the input image or frame from the pedestrian intention estimation dataset.
2. Download and load the pre-trained VGG-19 model into memory.
3. Convert the input image or frame to a suitable input format for the VGG-19 model, represented as a tensor X.
4. Pass the tensor X through the VGG-19 model to obtain the output tensor Y: $Y = VGG - 19(X)$
5. Extract the high-level features from the output tensor Y.

Step 3: Train a CNN to classify pedestrian poses and movements

1. Define the CNN model architecture as a function f with input X and output Y': $Y' = f(X)$
2. Define the loss function as $L(Y', Y) = -sum(Y * log\ log\ (Y')\ )$.
3. Define the optimization algorithm as stochastic gradient descent (SGD) with learning rate alpha and weight update rule $w = w - alpha * dw$.
4. Split the dataset into a training set $D_{train}$and a validation set D_{val}.
5. Initialize the model parameters with random values.
6. For each epoch in a fixed number of epochs or until convergence:

   a. Iterate over the training set $D_{train}$and compute the output Y' of the CNN model for each input X: $Y' = f(X)$

   b. Compute the loss $\mathrm{L(Y', Y)}$between the predicted output $\mathrm{Y'}$ and the ground-truth output Y.

   c. Compute the gradient of the loss function with respect to the model parameters using backpropagation: $\mathrm{dw} = \frac{\mathrm{dL(Y',Y)}}{\mathrm{dw}}$

   d. Update the model parameters using SGD: $\mathrm{w = w - alpha * dw}$

   e. Evaluate the performance of the CNN model on the validation set $\mathrm{D_{val}}$.
7. Select the CNN model with the best performance on the validation set as the final model.

4. Results And Analysis

4.1 Experimental Setup

The experiment was conducted to understand pedestrian behavior at zebra crossings using a Pedestrian Pose Estimation model. The COCO-18 keypoint dataset was used, which contained 1842 samples of pedestrian data of

PIE dataset [35] as explain in section 3.1 . The data was split into train, test, and validation sets at the ratios of 50%, 40%, and 10%, respectively. Only tracks longer than 2 seconds (observation + prediction) were kept for behavior prediction. The Pedestrian Pose Estimation model was trained using CNN architecture with a VGG-19 Pretrained model to train the system accurately with low training loss and high training accuracy. The Python 3.6.7, Anaconda distribution 2018.12, and Spyder 3.3.2 environments were used for loading and training the chosen network architectures. The PyTorch 1.0.1, CudaToolKit 9.0, Torch vision 0.2.2, Numpy 1.16.2, and Pandas 0.24.2 software packages were used in this process.

The training of the intent model was done for 300 iterations at a batch size of 128 with L2 regularization set to 0.001. The trajectory model was trained using 50 iterations with a batch size of 64 and an L2 regularization of 0.0001. The evaluating metrics used in the study included precision, recall, F1 score, accuracy, and AUC. SVM, RF, GBM, and XGBoost models were used to predict pedestrian crossing intentions, and the model performance was improved by fine-tuning hyperparameters. The classification results were presented, and the Random Forest Classifier was found to perform better in classification, with a recall value of 0.949 in the walking class and an AUC value of 0.922.

4.2 Evaluating Metrics

The evaluation measures are listed below, including precision, recall, F1 score, accuracy, and AUC.

$$\text{Precision } = \frac{\text{TP}}{(\text{TP} + \text{FN})} \qquad (13)$$

$$\text{Recall } = \frac{\text{TP}}{(\text{TP} + \text{FN})} \qquad (14)$$

$$\text{F1 score } = 2 * \left(\frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}\right) \qquad (15)$$

$$\text{Accuracy } = \frac{(\text{TP} + \text{TN})}{\text{TP}} \qquad (16)$$

AUC: AUC stands for Area Under the ROC Curve, which is a plot of the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. It represents the ability of the model to distinguish between positive and negative classes.

The formulas for calculating TPR and FPR are as follows:

$$\text{TPR } = \frac{\text{TP}}{(\text{TP} + \text{FN})} \qquad (17)$$

$$\text{FPR } = \frac{\text{FP}}{(\text{FP} + \text{TN})} \qquad (18)$$

The AUC represents the area under the ROC curve, and its value ranges from 0 to 1, where a value of 0.5 indicates a random classifier, and a value of 1 indicates a perfect classifier

In an experiment, SVM, RF, GBM, and XGBoost are used to predict pedestrian crossing intentions. Model performance is improved by fine-tuning hyper parameters.

4.3 Experiment Results

4.3.1 Classification: The Classification results in the paper are presented in Table 4 and Table 7. These tables show the precision, recall, F1 score, accuracy, and AUC values for different models used in the study to predict pedestrian crossing intentions [36].

Table 4. Shows the classification results for four models with a forecast horizon of one second (Test Data)

Model	Class	Precision (%)	Recall (%)	F1Score (%)	Accuracy (%)	AUC
SVM	Walking	73.15	92.29	80.51	82.23	0.8
	Macro Average	72.32	91.11	77.56		34
RF	**Walking**	**89.28**	**94.93**	**87.51**	**88.57**	**0.9**
	Macro Average	**88.98**	**88.12**	**84.92**		**10**
GBM	Walking	69.10	75.58	86.80	80.98	0.8
	Macro Average	72.93	88.44	81.21		21
XGBT	Walking	88.13	82.57	80.12	85.35	0.8
	Macro Average	79.11	87.48	76.60		93

* Bold indicates the best metric-based model.

Table 4 presents the classification results for four models - SVM, RF, GBM, and XGBoost, with a forecast horizon of one second on the test dataset. The table shows the precision, recall, F1 score, accuracy, and AUC values for each of the five categories - walking, standing, hand signaling, crossing, and not crossing. The table also shows the macro-average value, which is the average of the values from all five categories. The best performing model is marked in bold. In this case, the Random Forest Classifier is the best model, with an AUC value of 0.910.

Table 5. Confusion matrix from the RF model (test dataset)

	Predicted Class					
Actual Class	**Class**	**Walking**	**Standing**	**Hand Signal**	**Crossing**	**Not Crossing**
	Walking	**150**	7	2	6	3
	Standing	0	**111**	11	7	7
	Hand Signal	6	5	**180**	0	15
	Crossing	0	2	3	**110**	3
	Not Crossing	2	0	4	1	**100**

*Marked in Bold number of Correctly Classified Samples

Table 5 shows the confusion matrix for the validation dataset used by the RF model. The confusion matrix shows the number of correctly and incorrectly classified samples for each of the five categories.

Table 6. A classification report with different values of prediction

Prediction Horizon	Class	Precision (%)	Recall (%)	F1Score (%)	Accuracy (%)	AUC
1 sec	Walking Macro Average	89.28 88.98	94.93 88.12	87.51 84.92	88.57	0.91 0
2 sec	Walking Macro Average	73.10 75.81	85.32 83.44	81.81 79.21	87.98	0.88 1
3 sec	Walking Macro Average	65.26 76.12	75.58 82.23	86.80 87.12	80.98	0.84 8
4 sec	Walking Macro Average	84.21 72.28	82.57 87.48	80.12 76.60	85.35	0.79 3

Table 6 shows the classification report for different prediction horizons - 1 sec, 2 sec, 3 sec, and 4 sec, on the test dataset. The table shows the precision, recall, F1 score, accuracy, and AUC values for each of the five categories. The macro-average value is also presented for each prediction horizon. The AUC value decreases steadily as the length of the prediction horizon is extended.

Table 7. Classification report for General Case (Test Dataset)

Model	Class	Precision (%)	Recall (%)	F1Score (%)	Accuracy (%)	AUC
SVM	Walking	72.89	91.21	82.63	80.45	0.826
	Macro Average	69.33	89.22	77.56		
RF	Walking	91.88	94.93	87.51	89.87	0.922
	Macro Average	86.11	89.37	82.89		
GBM	Walking	72.67	72.12	82.22	81.31	0.831
	Macro Average	72.93	88.44	81.21		
XGBT	Walking	82.21	79.21	79.14	84.16	0.864
	Macro Average	73.66	85.21	77.20		

Table 7 presents the classification report for the general case on the test dataset. The table shows the precision, recall, F1 score, accuracy, and AUC values for each of the four models - SVM, RF, GBM, and XGBoost. The table also shows the macro-average value, which is the average of the values from all five categories. The best performing model is marked in bold, which is the Random Forest Classifier, with an AUC value of 0.922.

4.3.2 Comparison Study

In this section a comparative study was conducted to evaluate the proposed method for understanding pedestrian behavior at zebra crossings. A reference model was created to use as a benchmark, and the results of experiments performed on the test dataset were compared for different models.

Table 7 presents the classification report for the general case on the test dataset for four models - SVM, RF, GBM, and XGBoost. The table shows the precision, recall, F1 score, accuracy, and AUC values for each of the models. The table also shows the macro-average value, which is the average of the values from all five categories. The best performing model is marked in bold, which is the Random Forest Classifier, with an AUC value of 0.922.

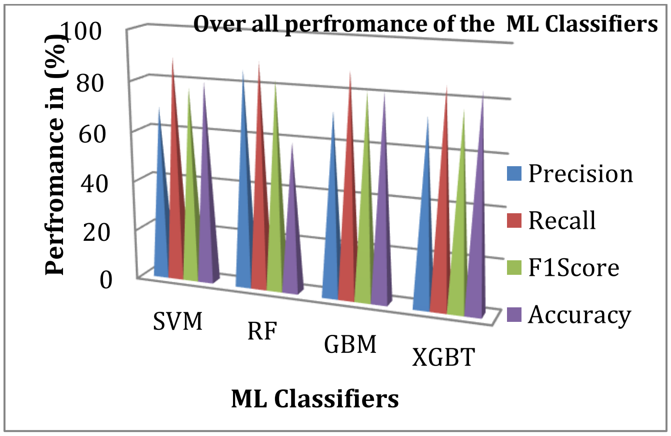

Fig 8. Overall Classification performance report for General Case (Test Dataset)

AUC curve

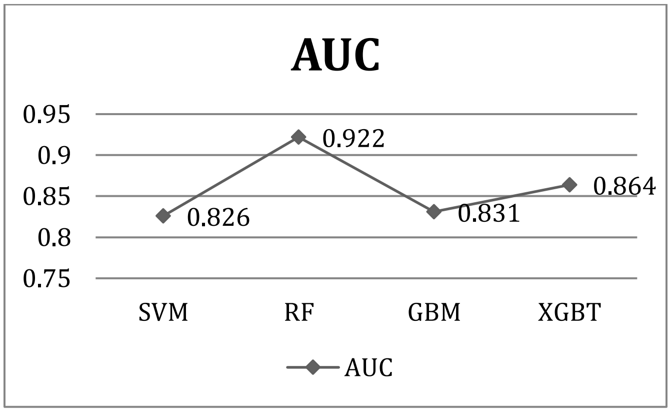

Fig 9. AUC Curve for proposed work

Figures 8 and 9 demonstrate the overall classification performance report and AUC curve, respectively, for the proposed work. The AUC curve shows the ability of the proposed method to distinguish between positive and negative classes.

The comparative study shows that the proposed method using a Multi-Person Pose Estimation model is effective in understanding pedestrian behavior at zebra crossings, and the Random Forest Classifier is the best performing model. The study provides a baseline for future research and suggests that transfer learning with pre-trained models and deep level pose estimation using a larger number of COCO key points can lead to more accurate training and improved results.

Overall, the Random Forest Classifier is found to be the best performing model in the study, with the highest recall value of 0.949 in the walking class and an AUC value of 0.922. The classification results demonstrate that the proposed method is effective in predicting pedestrian crossing intentions at zebra crossings.

5. Conclusion

In conclusion, the proposed method of predicting pedestrian behavior at zebra crossings using bottom-up pose estimation and deep learning-based classifiers is effective and has significant implications for improving pedestrian safety. The proposed solution provides a comprehensive solution for detecting and classifying pedestrian poses and movements such as walking, standing, hand signals, crossing, and not crossing at zebra crossings. The research paper addresses the challenges of predicting pedestrian behavior and intentions, which are essential for traffic management, developing Advanced Driver Assistance Systems (ADAS), and creating autonomous vehicles. The proposed method allows for the analysis of videos captured at zebra crossings and enables better anticipation and response to pedestrian actions. The study highlights the importance of integrating perception and decision-making tasks in predicting pedestrian behavior and provides a promising solution for addressing this critical problem. The Random Forest Classifier performs the best among SVM, RF, GBM, and XGBoost in classifying pedestrian behavior, achieving a recall value of 0.949 in the walking class and an AUC value of 0.922.

Future research can focus on improving the accuracy of the proposed method by using transfer learning with pre-trained models and deep level pose estimation using a larger number of COCO key points. The proposed method can also be extended to real-time applications, and the system can be integrated into existing traffic infrastructure to improve pedestrian safety.

Statements and Declarations

Acknowledgment: The authors would like thank the anonymous reviewers.

Competing Interests: The authors declare that they have no financial or non-financial interests that are directly or indirectly related to the work submitted for publication.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest: Authors declare no conflict of Interest.

Ethics Approval: The authors confirm that any human subjects involved in the study provided informed consent, and that the study was approved by the relevant institutional ethics committee.

Data Availability: We regret that due to the nature of this research and potential privacy concerns, we are unable to publicly share the code and data used to generate the results presented in this paper. However, we are happy to work with interested parties on a case-by-case basis to share as much information as possible within the bounds of privacy considerations.

Consent for Publication: All authors give the permission to the Journal to publish this Research paper

References

[1] "Road Traffic Injuries." https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, www.who.int/news-room/fact-sheets/detail/road-traffic-injuries. Accessed 12 Nov. 2022.

[2] Veena Hosamani, & Dr. H S Vimala. (2018). Data Science: Prediction and Analysis of Data using Multiple Classifier System. *International Journal of Computer Engineering in Research Trends*, *5*(12), 216–222.

[3] Sethuraman, R., Sellappan, S., Shunmugiah, J., Subbiah, N., Govindarajan, V., & Neelagandan, S.An Optimized AdaBoost Multi-class Support Vector Machine for Driver Behavior Monitoring in the Advanced Driver Assistance Systems. *Expert Systems with Applications*, 118618 (2022).

https://doi.org/10.1016/j.eswa.2022.118618

[4] Schwalb, E. Analysis of Hazards for Autonomous Driving. *Journal of Autonomous Vehicles and Systems*, *1*(2).(2021) https://doi.org/10.1115/1.4049922

[5] A, B. Improvised Autonomous Car for a Safe Travel. *Current Trends in Biomedical Engineering &amp; Biosciences*, *6* (2017).

https://doi.org/10.19080/ctbeb.2017.06.555678

[6] Zhang, H., Liu, Y., Wang, C., Fu, R., Sun, Q., & Li, Z. Research on a Pedestrian Crossing Intention Recognition Model Based on Natural Observation Data. *Sensors*, *20*, 1776(2020) https://doi.org/10.3390/s20061776

[7] S D, V. S., & C J, P. (2023). A Study on Vision Based Lane Detection Methods for Advanced Driver Assistance Systems. *International Journal of Computer Engineering in Research Trends*, *10*(8), 1–10.

https://doi.org/10.22362/ijcert.v10i8.750

[8] Zhang, S., Klein, D. A., Bauckhage, C., & Cremers, A. B. *Fast moving pedestrian detection based on motion segmentation and new motion features. Multimedia Tools and Applications, 75, 6263–6282(2015).*

doi:10.1007/s11042-015-2571-z

[9] D.Sreedevi, Prof.K.Samatha, & Prof.M.P.Rao. (2019). A Review on Typical and Modern Brain MRI Image Segmentation Methods and Challenges. *International Journal of Computer Engineering in Research Trends*, *6*(5), 322–329.

[10] M Bhavsingh, & S.Jan Reddy. (2023). Enhancing Safety and Security: Real-Time Weapon Detection in CCTV Footage Using YOLOv7. *International Journal of Computer Engineering in Research Trends*, *10*(6), 1–8.

https://doi.org/10.22362/ijcert.v10i6.855

[11] Chen, W., Jiang, Z., Guo, H., & Ni, X. Fall Detection Based on Key Points of Human-Skeleton Using OpenPose. *Symmetry*, *12*, 744(2020).

https://doi.org/10.3390/sym12050744

[12] Wu, Y., Zhang, X., & Fang, F. Automatic Fabric Defect Detection Using Cascaded Mixed Feature Pyramid with Guided Localization. *Sensors*, *20*, 871(2020). https://doi.org/10.3390/s20030871

[13] Zheng, H., & Liu, Y. Lightweight Fall Detection Algorithm Based on AlphaPose Optimization Model and ST-GCN. *Mathematical Problems in Engineering*, 1–15(2022). https://doi.org/10.1155/2022/9962666

[14] S, N., N, P., & P, N. (2023). A Study on Flower Classification Using Deep Learning Techniques. *International Journal of Computer Engineering in Research Trends*, *10*(4), 161–166.

[15] Li, B., Ji, Y., Li, Y., Xu, Y., & Liu, C.Pose Knowledge Transfer for multi-person pose estimation. *Signal, Image and Video Processing*, *16*, 321–328(2021). https://doi.org/10.1007/s11760-021-01922-5

[16] Han, Y., & Wang, G. Skeletal bone age prediction based on a deep residual network with spatial transformer. *Computer Methods and Programs in Biomedicine*, *197*, 105754(2020).

https://doi.org/10.1016/j.cmpb.2020.105754

[17] Angadi, S., & Nandyal, S. Human Identification Using Histogram Of Oriented Gradients (Hog) And Non-Maximum Suppression (Nms) For Atm Video Surveillance. International Journal of Innovative Research in Computer Science and; Technology, 9,(2021)https://doi.org/10.21276/ijircst.2021.9.3.1

[18] M Bhavsingh, B.Pannalal, & K Samunnisa. (2022). Review: Pedestrian Behavior Analysis and Trajectory Prediction with Deep Learning. *International Journal of Computer Engineering in Research Trends*, *9*(12), 263–268

[19] Zhang, Y., Liu, W., Yang, X., & Xing, S. Hidden Markov Model-based Pedestrian Navigation System using MEMS Inertial Sensors. Measurement Science Review, 15,35–43(2015). https://doi.org/10.1515/msr-2015-0006

[20] Martinez-Gonzalez, A. N., Villamizar, M., Canevet, O., & Odobez, J. M. Efficient Convolutional Neural Networks for Depth-Based Multi-Person Pose Estimation. *IEEE* Transactions on Circuits and Systems for Video Technology, *30*,4207–4221(2020).

https://doi.org/10.1109/tcsvt.2019.2952779

[21] Fang, Z., & Lopez, A. M. Intention Recognition of Pedestrians and Cyclists by 2D Pose Estimation. IEEE Transactions on Intelligent Transportation Systems, 1–11(2019).

[22] Munea, T. L., Jembre, Y. Z., Weldegebriel, H. T., Chen, L., Huang, C., & Yang, C. The Progress of Human Pose Estimation: A Survey and Taxonomy of Models Applied in 2D Human Pose Estimation. *IEEE Access*, *8*, 133330–133348(2020). https://doi.org/10.1109/access.2020.3010248

[23] Cheng, Y., Yang, B., Wang, B., & Tan, R. T. 3D Human Pose Estimation Using Spatio-Temporal Networks with Explicit Occlusion Training. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*, 10631–10638 (2020). https://doi.org/10.1609/aaai.v34i07.6689

[24] Liu, W., & Mei, T. Recent Advances of Monocular 2D and 3D Human Pose Estimation: A Deep Learning Perspective. *ACM Computing Surveys*.(2022) https://doi.org/10.1145/3524497

[25] Soccolich, S. A., Fitch, G. M., Perez, M. A., & Hanowski, R. J. Comparing Handheld and Hands-free Cell Phone Usage Behaviors While Driving. *Traffic Injury Prevention*, *15*(sup1), S21–S26(2014). https://doi.org/10.1080/15389588.2014.934958

[26] Zhang, S., Abdel-Aty, M., Wu, Y., & Zheng, O. Pedestrian Crossing Intention Prediction at Red-Light Using Pose Estimation. *IEEE Transactions on Intelligent Transportation Systems*, *23*, 2331–2339(2022). https://doi.org/10.1109/tits.2021.3074829

[27] Ju, H., & Park, C. G. A pedestrian dead reckoning system using a foot kinematic constraint and shoe modeling for various motions. *Sensors and Actuators A: Physical*, *284*, 135–144(2018). https://doi.org/10.1016/j.sna.2018.09.043

[28] Ahmed, T., Moeinaddini, M., Almoshaogeh, M., Jamal, A., Nawaz, I., & Alharbi, F. A New Pedestrian Crossing Level of Service (PCLOS) Method for Promoting Safe Pedestrian Crossing in Urban Areas. *International Journal of Environmental Research and Public Health*, *18*, 8813(2021). https://doi.org/10.3390/ijerph18168813

[29] A., S. R., S., A., R., H., & S., S. K. Stacking Deep learning and Machine learning models for short-term energy consumption forecasting. *Advanced Engineering Informatics*, *52*,101542(2022). https://doi.org/10.1016/j.aei.2022.101542

[30] Hartwig, L., Kaufmann, C., Risser, R., Erbsmehl, C., Landgraf, T., Urban, M., & Schreiber, D. Evaluating Pedestrian and Cyclist Behaviour at a Level Crossing. *Transactions on Transport Sciences*, *11*, 41–54(2020). https://doi.org/10.5507/tots.2019.009

[31] Hariyono, J., & Jo, K. H. . Detection of pedestrian crossing road: A study on pedestrian pose recognition. *Neurocomputing*, *234*, 144–153(2017). https://doi.org/10.1016/j.neucom.2016.12.050

[32] LI, W., ZHAO, X., & SUN, D. Prediction of trajectory based on modified Bayesian inference. *Journal of Computer Applications*, *33*, 1960–1963(2013). https://doi.org/10.3724/sp.j.1087.2013.01960

[33] Min, J., Lim, E., & Son, S. The effect of airline kiosk service on behavior intention : Apply the theory of planned behavior(TPB). *International Journal of Tourism and Hospitality Research*, *36*, 187–197(2022). https://doi.org/10.21298/ijthr.2022.1.36.1.187

[34] Kalatian, A., Sobhani, A., & Farooq, B. Analysis of distracted pedestrians' waiting time: Head-Mounted Immersive Virtual Reality application. *Collective Dynamics*, *5*.(2020) https://doi.org/10.17815/cd.2020.32

[35] Rasouli, Amir & Kotseruba, Iuliia & Kunic, Toni & Tsotsos, John. PIE: A Large-Scale Dataset and Models for Pedestrian Intention Estimation and Trajectory Prediction. 6261-6270(2019). 10.1109/ICCV.2019.00636.

[36] Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., & Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*, 172–186(2021). https://doi.org/10.1109/tpami.2019.2929257.

[37[Diniesh, V. C. ., Prasad, L. V. R. C. ., Bharathi , R. J. ., Selvarani, A., Theresa, W. G. ., Sumathi, R. ., & Dhanalakshmi, G. . (2023). Performance Evaluation of Energy Efficient Optimized Routing Protocol for WBANs Using PSO Protocol. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 116–121. https://doi.org/10.17762/ijritcc.v11i4s.6314

[38] Mr. A. Kingsly Jabakumar. (2019). Enhanced QoS and QoE Support through Energy Efficient Handover Algorithm for UMTS Architectures. International Journal of New Practices in Management and Engineering, 8(01), 01 - 07. https://doi.org/10.17762/ijnpme.v8i01.73

About Authors

PANNALAL BODA received the B.Tech. degree in computer science engineering from the Jawaharlal Nehru Technological University, Hyderabad, Telangana, India, in 2004, and the M.Tech. degree in computer science engineering from the Jawaharlal Nehru Technological University, Anantapur, Andhra Pradesh, India, in 2009. He is currently pursuing the Ph.D. degree in computer science engineering with Osmania University. His research interests include the pedestrian behaviour prediction for intelligent vehicles, pedestrian modelling and simulation, decision making, local path planning, and intelligent decision technology, machine learning, and deep learning.

Dr. Y. RAMADEVI has completed has Ph.D. from Hyderabad Central University, Hyderabad. She has around 30 years of Teaching and 20 years of Research Experience. She is currently professor Department of CSE_AIML Chaitanya Bharathi Institute of Technology, INDIA. Her Research areas include Artificial Intelligence, Machine Learning, Internet of Things, Bioinformatics etc. She has more than 100 publications in reputed Journals, delivered Keynote speeches at various International conferences and workshops. She has patents and Copyrights. She has Funded Research projects from AICTE. She is research supervisor for Osmania University, JNTU: and has successfully guided Eleven Ph.D. and presently eight are Pursuing under her.