IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 29, 2020, accepted April 10, 2020, date of publication April 13, 2020, date of current version April 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2987777

Predicting Pedestrian Intention to Cross the Road

KARAM M. ABUGHALIEH AND SHADI G. ALAWNEH, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, Oakland University, Rochester, MI 48309, USA
Corresponding author: Shadi G. Alawneh (shadialawneh@oakland.edu)

ABSTRACT The goal of this research is the development of a driver assistant feature, which can warn the driver in case a pedestrian is in a potential risk due to sudden intention to cross the road. The process of crossing pedestrian is defined as the changing of pedestrian orientation on the curb toward the road. We built a Convolutional Neural Network (CNN) model combined with depth sensing camera to estimate the pedestrian orientation and distance from the vehicle. The model detects the higher human body keypoints in 2D space while the depth info make it possible to translate the points into a 3D space. These info are tracked per pedestrian and any change in the pedestrian moving pattern toward the road is translated to a warning for the driver. The CNN model is end-end trained using different datasets presenting pedestrian in different configurations and scenes.

INDEX TERMS ADAS, GPU, CNN.

I. INTRODUCTION

One of the main tasks for assistive and autonomous driving systems is to assure traffic safety for drivers and pedestrians by reducing the human error leading to crashes with other vehicles, road infrastructure and pedestrians. Pedestrian injuries in traffic accidents have high lethality due to the vulnerability of the pedestrians. According to Governors Highway Safety Association (GHSA) preliminary report for 2019 [1], 6590 pedestrian were killed in motor vehicle accidents, with an increase of almost 300 deaths from the reported number in 2018.

Governments spared no effort to make roads a safer place to use. By crafting better road regulations and construct roads infrastructures. On the other hand, tech companies and researchers are trying very hard to make vehicles safer for both pedestrians and drivers by using advanced technologies helping the driver to avoid crashes and even if happened to reduce the impact of it.

Autonomous cars on various levels, including the fully automated or the ones that equipped with advanced driver assist systems (ADAS) should have robust and efficient algorithms to avoid vehicle-pedestrian crashes as much as possible either by initiating the required driving actions or by giving the drivers extra information to be aware of their surroundings. Both autonomous and connected vehicle technologies should have the abilities to determine if the pedestrian is crossing the road in the path of the vehicle, in order to have an enough response time to issue the required alerts for the driver or trigger safety breaking action.

The presence of communication channels in connected vehicles technology either between the vehicle and it's surrounding infrastructure or other nearby vehicles can enhance the process of sensing the pedestrians. This can be achieved by providing the sensing information as a service to other vehicles [2]. The pedestrian data can be detected and shared from a leading vehicle to the vehicle in the back which will reduce the processing time in these vehicles and therefor more time for reaction.

Vision based pedestrians detection field has been very active and is rich with methods and algorithms [3]. During the last decade, deep learning techniques witnessed breakthrough in the applications and performance. Graphics Processing Units (GPUs) played a significant role in this breakthrough by enabling fast processing of big data and training large CNN models. Deep learning based pedestrian detectors provides accurate pedestrians detection even with the large amount of variation in human look caused by clothes and body shapes. Even with such advances in pedestrian detectors, avoiding vehicle to pedestrian crashes still a challenging task such in cases where a pedestrian decides to cross the road suddenly. In such cases the human driver and the autonomous driver have shorter time to initiate the required response.

Pedestrian detection is a critical step in any pedestrian-safety algorithm but it's only the first step for a safer pedestrian-vehicle interaction. The vehicles should have the ability to analyze and track the activities of pedestrian along video frames in order to determine the required actions

The associate editor coordinating the review of this manuscript and approving it for publication was Moayad Aloqaily.

to reduce the risk of crashing. Providing the driver or the auto-driver with information related to pedestrians behavior on the road can significantly increase the pedestrian safety. Activities like intention to cross or detecting pedestrian awareness can be a part of the decision making inputs to perform smooth maneuver to prevent accident or reduce the impact of it. Prediction of pedestrian crossing the road one second prior to the actual action can provide extra distance for a vehicle automatic response or a driver response to take action. A couple of seconds of prediction for a pedestrian intention could be critical to avoid crashes or reducing the chance of injury requiring hospitalization.

Interpretation of pedestrian actions and movements on the curb could detect the pedestrian intention to cross the street or not. Actions like bending the higher part of the body, heading toward the street or making eye contact with the driver could give a higher indication for the intention of the pedestrian to cross the road. All these signs can be essential parts in the process of designing the assistive and autonomous driving systems to become more suitable for urban environments. A Proper estimation for the pedestrian path depending on the pedestrian pose and speed provides the vehicle with accurate estimation of probability of crash with the pedestrian. Another source of info that is significant for the process of prediction is the environment around the vehicle like the distance between the pedestrian and vehicle, crossing sign and pedestrian crossing walk existence.

The proposed approach in this work builds on the ideas from a previous work [4] presenting an enhanced CNN model for body landmarks detection in addition to a pedestrian intention to cross the street detector, the detector is based on detecting sudden changes of pedestrian orientation toward the street. The novel contributions of this research work can be summarized in the following:

- Developing a CNN model for detecting human body landmarks with a higher accuracy than our previous work.
- Increasing the dataset size for labeled pedestrians for the landmarks of shoulder, neck and nose proposed in our previous work.
- Developing a street crossing intention based on detecting sudden pedestrian orientation change toward the road. The orientation detection is based on our previous work of depth module that translates the detected landmarks into a 3D space where the body orientation is estimated.

The rest of the paper is organized as follows. Section II presents the related work, then the system overview is described in details in Section III. Section IV describes the obtained result with some discussion and analysis. Finally, Sections V and VI concludes with the ongoing research and future plans.

II. RELATED WORK

Pedestrian behavior analysis includes detecting one or more sign like pedestrian body orientation, head orientation, pedestrian focus. Body orientation is taken in a certain reference; mostly the camera. In addition to autonomous vehicles pedestrian safety functionality, social robots are one of the main applications requiring this kind of information in order to build advanced path planing algorithms, other fields that can make use of such information is surveillance for behavior and interaction analysis.

Many techniques have been widely used for understanding pedestrians behaviors in the road, either by understanding the pedestrian motion or analyzing the pedestrian behaviors and intentions. Using on body sensors is one of the methods used to capture the pedestrian orientation, Peng and Qian [5] used motion capture devices to estimate the human body orientation. The work in [6] also used external magnetic sensors to estimate the orientation. Such methods work on controlled environments but not suitable for on road pedestrians.

The head pose provides good clues about the pedestrian focus and can be used in overall body orientation estimation, Chen *et al.* [7] proposed an approach that jointly estimates body pose and head pose from surveillance video, taking advantage of the soft couplings between body position (movement direction), body pose, and head pose. The authors in [8] focused on estimating the human head orientation from extremely low-resolution RGB images using a non-linear regression and used Support Vector Regression (SVR).

Utilizing deep learning methods to estimate the body orientation, Choi [9], used a convolutional neural networks for estimating human body orientation. The model classifies the input image into one of eight classes covering the 360 degrees. In a previous work [10] we used a combination of OpenPose implementation [11] and lifting from the Deep Learning implementation [12] to estimate a human body orientation. OpenPose was used to detect the human body landmarks defines in the COCO dataset [13] which were 17 points, then these points were passed to the other algorithm to produce the 3D space translation. We were able to estimate the body orientation using these points by building vectors using the shoulders points, and another vector using hips points. In another work [4], a CNN model was developed to detect the body landmarks of the shoulders, neck and face only, this time the points were translated to a 3D space using a depth camera. Then the same concept of using vectors to compute the orientation is applied successfully.

In the area of understanding the pedestrian behaviors and intention. Pedestrian's intention can be analyzed by tracking their current status and previous one, the status might include walking directions, motion speed, position, head orientation and awareness, awareness is highly related to head orientation, eyes direction and being busy using mobile phone for example. The head orientation is a very important indication for the pedestrian behavior, [14] utilizes human body language to predict behaviors based on head orientation. A stereo camera vision was used for human detection and head pose estimation using a Latent-Dynamic Conditional Random Field mode. More research examples for pedestrian intention based on head orientation estimation can be found

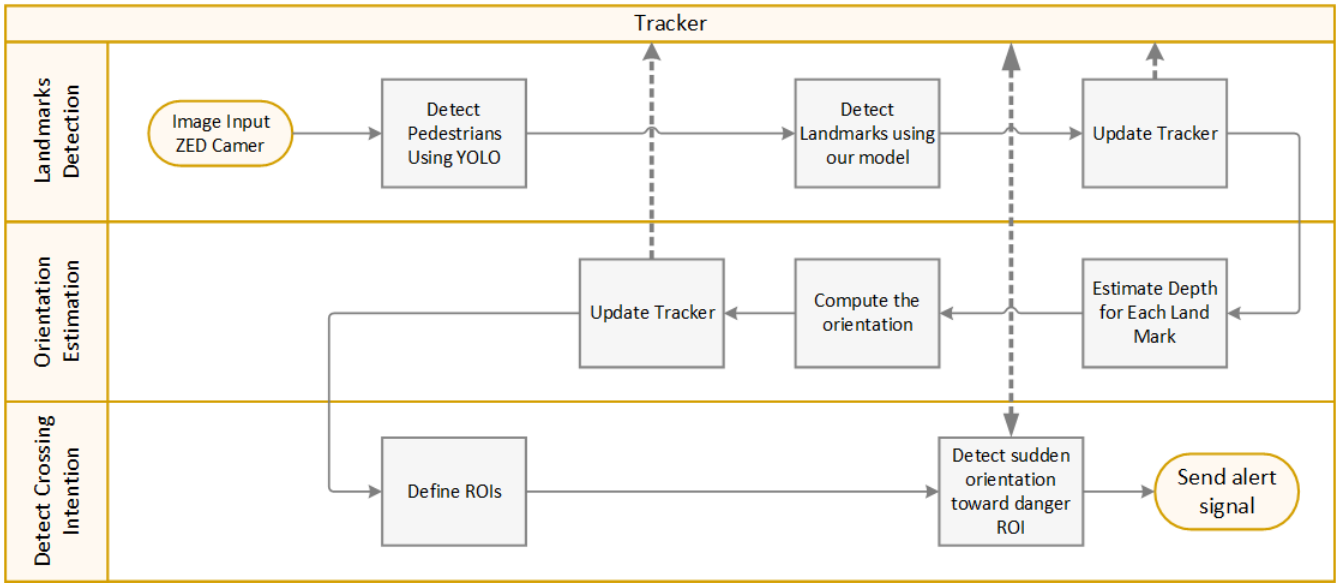


FIGURE 1. System pipeline overview, showing the different modules in the system in addition to the flow of the process and tracker updating phases.

in [15], [16], these approaches present methods based on monocular and stereo cameras.

In [17], the authors also used the body language in 3D to perform pedestrian activity and path prediction based on pose estimation. The system uses a LIDAR and stereo vision camera equipped on a moving vehicle. Kataka *et al.* [18] used body pose and gait analysis to recognize pedestrians activities. The authors localized pedestrian using extended CoHOG + AdaBoost while dense trajectories are used for activity analysis. The classification has four classes: crossing, walking, standing and riding a bicycle. In [19], Kooij *et al.* used a stereo vision system to extract the context information as the head orientation, the vehicle-pedestrian distance and spatial layout by the distance of the pedestrian to the curbside on top of a Switching Linear Dynamical System to predict more accurate path and action for a horizon of one second.

In [20], pedestrian intent prediction is used for risk estimation using clues of pedestrian dynamics, and map information based on GPS location. The system is monocular based and use the vision info for trajectory tracking and predictions in the near future to issue risk alert. The pedestrian annotation is done manually as the process of detection is out of the scope of this work.

A decent amount of effort has been placed on the task of pedestrians detection and their behavior estimation. Detection models varies between hand-crafted features and deep learned ones using different datasets for the different tasks. Building a system that estimates the risk on pedestrians based on their behavior in the roads requires combining the tasks of human (pedestrian) detection and walking orientation estimation while keep performing in real-time.

III. METHODOLOGY AND SYSTEM OVERVIEW

The main concept in this approach is to construct a 3D visualization of the human body that gives a clear clue for the body orientation. In order to achieve that, this method focuses on important landmarks on the pedestrian. These landmarks are the shoulders, the neck and the face. These body landmarks are chosen because they are highly related to the body orientation. The relation between these landmarks and the orientation is more obvious when inserting how far each point is from the observer plane; the camera in this case. Imagining a line connecting the two shoulders points that is centered on neck point can give a better picture for the concept. Finding the normal vector for this line gives the body orientation.

The described methodology till now estimates the pedestrian orientation, detecting crossing the street intention requires a tracker that keeps tracking each pedestrian detected orientation. Having this tracker makes it possible to detect changes of the orientation toward the road for each pedestrian. This change can be understood as an intention to cross the road, based on this intention initial actions like slowing down can be taken by the driver or the auto-driver. Slowing down will provide a longer reaction time in case the pedestrian continued crossing the road.

The designed system that translates this methodology consists of different modules performing different tasks in order to detect the pedestrian intention to cross the road, Fig. 1 shows the system pipeline. The modules of the system perform the tasks of pedestrian detection, body landmarks detection, depth sensing and orientation estimation, the system also has a pedestrian tracker that keeps track of all gathered information for each pedestrian from the other modules. The following subsections explain each module in more details.

A. BODY LANDMARKS ESTIMATION

This module consists of two sub-modules, the first module is a pedestrian detector, while the second module is our trained CNN body landmarks estimator module. The two modules

work together in sequence to extract the pedestrians body landmarks; the extracted pedestrians from the input image by the first module are passed to the landmarks estimator module where it only works inside pedestrian detection regions.

1) PEDESTRIANS DETECTOR

Pedestrian detection is the first essential task in the system pipeline. Given a frame input from the camera this module task is to detect and localize every pedestrian in the scene. So each pedestrian is bounded by a boundary box that will be registered or updated in the pedestrians tracking module; explained in the following section. Any pedestrian detector could be used here, but considering detection accuracy and robustness, YOLO [21] detectors are used. YOLOv3 [22] and TinyYolov3 were tested for resources and processing speed testing. TinyYolov3 is a tiny version of YOLOv3, and is much faster but less accurate.

YOLO (You Only Look Once), is a single-stage neural network for object detection, a boundary box and class prediction are generated as the output of processing the input image. Previous methods for object detection, like R-CNN [23] and its variations perform object detection in multiple steps; extract 2000 regions from the image in a process called region proposals then classify these regions. This can be slow to run and also hard to optimize, because each individual component must be trained separately. On the other side YOLO, performs the detection with a single neural network. Single stage methods like YOLO [21], [22], [24], and SSD [25] achieve high performance speed but YOLO outperforms the others as shown in Table 1.

TABLE 1. Performance comparison for neural network algorithms done by [24].

Detection Frameworks	MeanAverage Precision (mAP)	FPS
Fast R-CNN	70.0	0.5
Faster R-CNN VGG-16	73.2	7
YOLO	63.4	45
SSD300	76.8	46
YOLOv2 544 × 544	78.6	40

2) BODY LANDMARKS ESTIMATION MODEL

The proposed neural network uses a CNN model that performs human landmarks estimation for the upper body part, this module works only inside the regions of pedestrians detected by the pedestrian detection module. As mentioned before, the points of interest in this work are the two shoulders, the neck and the face. In our context the face keypoints is the same as presented by the nose point in COCO [13] and MPII [26] parts mapping. The human body orientation is highly related to the position of these points, check Figs. 2 and 3 below.

All the images of the dataset are resized to match the CNN model size input which is 75 × 75 pixels, then provided as labeled examples with their respective keypoints for the model to perform the training. Such type of training is called

FIGURE 2. Example of the CNN model output, the body landmarks are detected if visible.

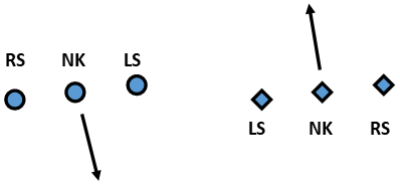


FIGURE 3. The relation between the selected body landmarks and the body orientations, an observer can easily conclude the body orientation given a top view of the detected landmarks.

a supervised learning. The CNN model outputs a vector with containing eight values representing the x and y coordinates for the four landmarks. The resulted model output is validated on another test dataset to evaluate the training process, this set is called validation or testing data.

The CNN architecture consists of a sequential structure of different types of layers. Neural network layers learn to extract features by activating certain nodes if a desired feature is found in the layer input, this is achieved by adjusting the layer parameters in the training process using the labeled examples. The model has input layers, hidden layers and output layer, the input layer is directly connected to the input image, while the hidden layers input comes from the input layer or another hidden layer output until the output layer is reached. As the name implies, the main layers were used in the building blocks of the model are of convolutional type for feature extraction followed by max pooling for down sampling the feature maps size for faster performance and to keep dominant features only by filtering out the weak features.

Another layers of dropout were also implemented for removing redundant nodes. The final output is flattened and a fully connected layers are used to extract the final eight values. The activation functions used in the layers is rectified linear unit (ReLU), which is faster than segmoid in the training process. The model consists of 6 building blocks, five

convolutional layers followed by max pooling layers, then a fully connected layer followed by drop out and finally the output layer. The convolution layers use the same filter size of (3×3) but with different counts; 32, 32, 64, 128 and 256. While the fully connected layer used is 512 nodes. The total number of trainable parameters in the network is 925992. Fig. 4 shows the full architecture.

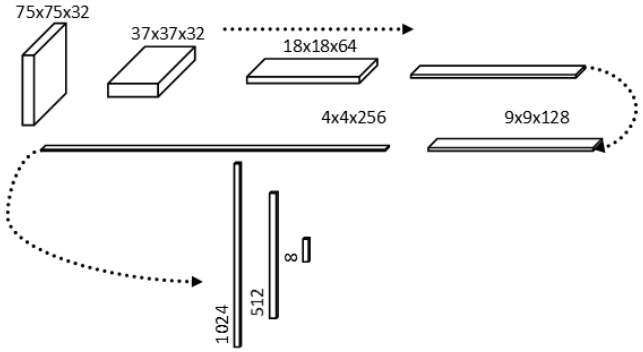


FIGURE 4. The architecture of the CNN model.

An important activity of building the body landmarks estimation model is preparing the training and the validation dataset. The collected dataset is a collection from different datasets available publicly online for different tasks like pedestrian detection images [27] and walking direction detection [28] in addition to self-collected pedestrian images using the ZED camera [29]. The images contains the pedestrian only with no other objects and minimal background as possible. For the self-collected images by ZED camera, a python script is written to automate the process of extracting pedestrians images and save them as separate files. The process includes pedestrian detection using YOLOv3, cropping the detection and save it as a new image file.

In order to get dataset ready for the training process the labeling information is required. The dataset was labeled manually using labelbox online tool [30]. The labeling object are defined as the four desired points, Right Shoulder(RS), Left Shoulder (LS), Neck (NK) and Nose (NS). The tool records the x and y coordinates for each selected point. The total number of collected sampled images is 20000 while the total number of labeled samples in 6000 images. In deep learning a larger dataset is always better to get higher accuracy. In order to achieve that with the small labeled dataset a well known practice called image augmentation was used. Image augmentation increases the number of training and validation samples by generating new samples. The process is done by applying different operations like horizontal and vertical flips, rotations, brightness change, adding noise, horizontal and vertical shifts. Tools like Keras [31] can do image augmentation automatically.

In this work the automated tools are not suitable for image augmentation, since an extra caution is required to fix the labels when a vertical flip is done, considering flipping the images vertically; the images of pedestrian facing the camera or facing the other direction, here the pedestrian left side and right side should stay the same from the camera view, while pedestrians showing one side of their bodies as in walking from side to side in front of the camera will have inverted left and right sides. So to over come these issue a scripts that perform image resizing, vertical and horizontal shifting is implemented to create more image variations, in addition to manual labeling for horizontally flipped images.

In this work also rotational and horizontal flip augmentation methods were not needed as they do not cover real examples. The total number of variation made for each image is almost 100 generating 600000 total samples that divided into training (80%) and validation (20%) sets. Fig. 5 illustrates some of the applied variations.

FIGURE 5. Sample of image variations applied on the base image on the top left corner.

B. PEDESTRIANS TRACKING MODULE

The tracker module works along all other modules. Each pedestrian is registered as a track-able object, where the tracker keeps all pedestrian obtained information saved as long as the pedestrian appears in the scene for a certain number of frames. The tracker does not perform the task of object detection but assumes that the tracked object is provided manually or as in this work automatically by a TinyYOLOv3 object detector. The tracker starts working when receiving the detected pedestrian from the pedestrian detector module. After that the YOLO detector will not be activated to update the detections for 25 frames to avoid unnecessary computations and achieve faster performance. At this stage the tracker can keep tracking the pedestrians in the upcoming frames and update the coordinates boundary boxes that was initially provided by YOLO for each pedestrian.

In order to build a tracker that assigns the right label and information to the same pedestrian in every frame the designed tracker consists of two components. a centroid tracker and Kernelized Correlation Filters (KCF) tracker [32].

The KCF is a variant of correlation filters. Correlation based filters consider a samples match if the samples have a high correlation value. KCF uses this idea for object tracking. KCF finds the correlation between the tracked object in the current frame and other patches in the next frame. The highest value correlation indicates in which direction the tracked object has moved. KCF tracker is not robust enough to significant change in object appearance. OpenCV [33] implementation is used for KCF tracker.

To keep a correct labeling and pedestrian information assignments the centroid tracker is implemented. The centroid tracker inputs are the tracked pedestrian objects from the KCF tracker, where the boundary boxes of each detected pedestrian is updated in every frame. As mentioned before each detected pedestrian is registered as a track-able object and given a unique ID with all other information and maintained by the centroid tracker. The centroid tracking approach as shown in Fig. 6 uses the Euclidean distance between the already registered tracked objects centroids and new objects centroids in a subsequent frame in the video.

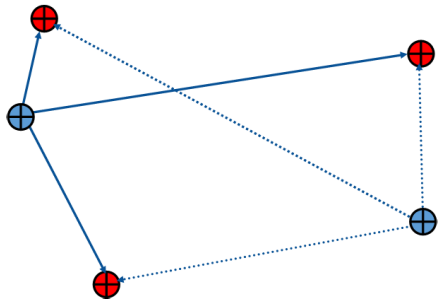


FIGURE 6. Blue point represent the centroid object in the previous frame while the red points represent the centroids of the detected objects in the current frame. The euclidean distance is measured for each centroid and the closest centroids in the new frame is given the same ID for the object in the previous frame.

Euclidean distance is computed in every frame between previously registered object and the newly updated centroids location. Based on the Euclidean distance analysis, the objects IDs will be updated by either assigning the same ID to the nearest centroid or giving a new ID if the a new object appeared and not registered previously or dropping the tracked object ID if the object is absent from the scene for certain number of frames.

C. DEPTH SENSING MODULE

At this point of the system pipeline, the pedestrian are detected with their body landmarks. one more thing to obtain in this module is how far these body landmark are from the camera plane, this distance is referred to as the depth. To obtain the depth measurements, a stereo vision [34] system is utilized. The depth information is measured for each pedestrian detected body landmarks to construct a 3D space presentation for the points. These points in the 3D world will be used to construct the vectors required for the orientation computation. The vectors creation depends on points visibility, we are assuming that the neck is always visible while one of the other points might not. The vector are constructed always from the left shoulder to the right shoulder through the neck point. The normal of the vector in the direction of face point is assumed to be the human body orientation.

Stereo vision setup can estimate the distance of a certain point using the two images taken by the two cameras. The cameras are separated by a known distance (b) known as the baseline. The difference in the viewpoints for the same scene from the two cameras provides extra information enabling the process of generating a depth map. The depth map is usually in a gray scale format and shows the distance between the camera and the objects in the scene. The extra information is the what called disparity, disparity is the horizontal shift that can be observed between the left camera image and the right camera image and it can be found on the pixel level, check Fig. 7. For the vertical shift to be valid a perfect alignment for cameras is assumed to be present in order match each pixel row in both images, this alignment is guaranteed by the mounting of separate cameras or the packaging of stereo camera manufacturer, otherwise an alignment pre-processing is required.

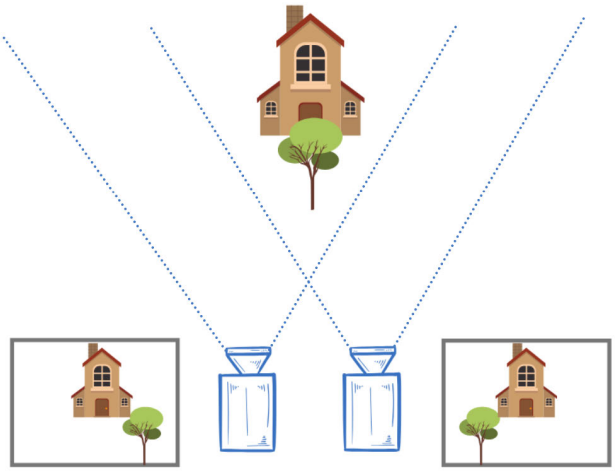


FIGURE 7. Disparity in stereo images.

The following set of equations describe the math concepts behind the stereo vision model in Fig. 8, given the disparity (d), the focal length of the two identical cameras (f) and the baseline distance separating the cameras (b), then the depth (z) can be defined as following based on the simple model of the pinhole camera. note that (z) is a plane to plane distance.

$$\frac{x_1 - x_2}{b} = \frac{f}{Z} \tag{1}$$

$$Z = \frac{bf}{x_1 - x_2} \tag{2}$$

Before obtaining the disparity a stereo matching should be achieved first. Assuming aligned images where each row

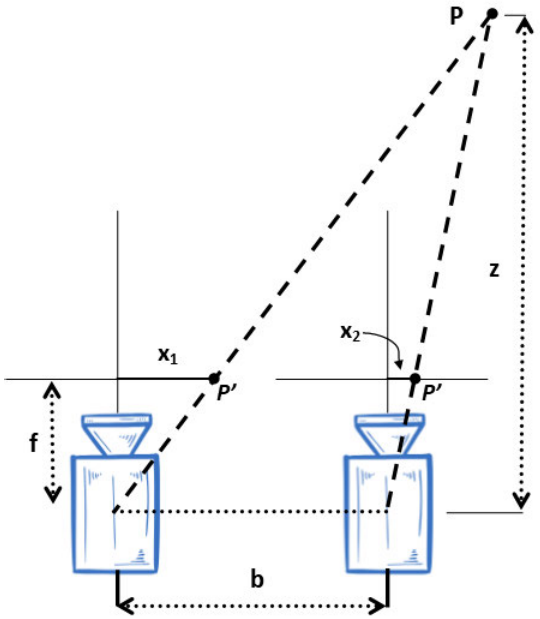


FIGURE 8. Stereo vision model.

in the left image is aligned to the one in right image the stereo matching is the process of finding corresponding pixels in stereo pair of images as shown in Fig. 9. After that the displacement of the pixels in reference to the left image for example is found and the disparity map is obtained. Those values can then be used to compute the depth as in shown in Fig. 8. The matching process is done row by row, as a starting point reference the same pixel column can be the starting point of search. The matching process is based similarity measure that defines the closest candidate to the target pixel.

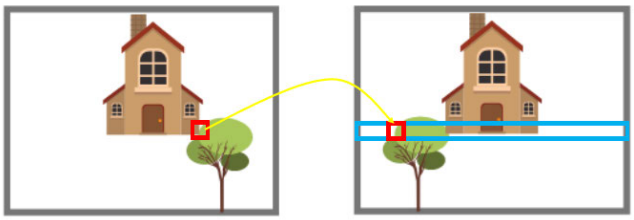


FIGURE 9. Pixel matching in stereo images view.

For this work the ZED camera by Stereolabs is used. The ZED camera is a stereo vision system that can be used to provide a 3D perception of the world. Providing a long range depth perception up to 20m [29]. It is suitable for many applications as in robot navigation, virtual reality, tracking, motion tracking and so on. The depth computation through the stereo vision in ZED camera is accelerated using CUDA [35] GPU computations.

D. ORIENTATION ESTIMATION MODULE

At this stage, the depth information collected by the previous depth sensing module are used to estimate the orientation for each pedestrian and update the tracker. One way to compute the orientation is by estimating the orientation in the 3D space using 3D vectors, another smarter and much simpler approach is to convert the 3D space into a 2D space by eliminating the height component of the pedestrian, in another words the 3D points are projected on the floor plane. The resulted 2D plane now contains the depth info on the y-axis and the location of the point at the x-axis, the same concept illustrated in Fig. 3 before by observing the 3D space from a top view, the concept is illustrated with two pedestrians example in Fig. 10. The new top view provides the algorithm with a clear conclusion about the pedestrian orientation.

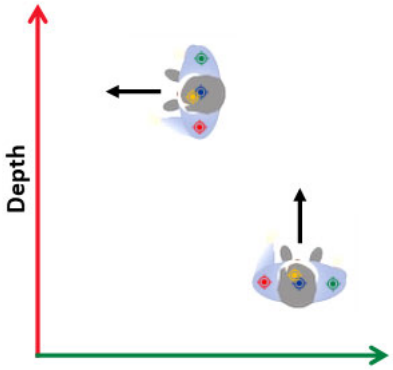


FIGURE 10. Illustration of the idea of converting the 3D space info into 2D space, the top view result is enough to get the pedestrian orientation.

The orientation is computed based on the 2D vectors constructed by connecting the available detected points as follows (LS-RS, LS-NK-RS, LS-NK, RS-NK, LS-NS, RS-NS), as noted the direction is always from the left side to the right side. Then using tan inverse function the orientation angle is known, the final angle is adjusted to be in reference to the camera view, so a pedestrian facing the camera is having a 0 degree orientation while pedestrian orientation toward right has 90 degrees orientation, to over come the limitation of tan inverse function in dealing with the whole range of angles atan2 is used in this work which is defined as follows:

$$\theta = arcTan2\left(\frac{y}{x}\right)$$

$$\text{arcTan2 (y, x)} = \begin{cases} arctan\,(y,x)\,, & x > 0 \\ \frac{\pi}{2}\arctan\left(\frac{x}{y}\right), & y > 0 \\ -\frac{\pi}{2} - \arctan\left(\frac{x}{y}\right), & y < 0 \\ \arctan\left(\frac{y}{x}\right) \pm \pi, & x < 0 \\ undefined, & x, y = 0 \end{cases} \quad (3)$$

E. CROSSING INTENTION DETECTION MODULE

This module builds on all the information gathered from the previous modules for each detected pedestrian. The module utilize the collected data to detect pedestrian intention of crossing the road in order to improve the situation awareness for the driver or the auto driver in urban environments. The driver view is categorized into two regions, the car path

region and the curb region or everywhere else. Based on the regions the detected pedestrians cases are categorized into three pedestrians defining the required level of driver awareness:

- Safe: A pedestrian walking on the curb, with an orientation parallel to the car path, no signs of crossing intention.
- Watch: A pedestrian walking on the curb but changed their walking orientation toward the car path, a sign of crossing intention.
- Risk: A pedestrian detected in the car path region, the pedestrian already crossing.

This task is actually a part of the tracker module which have an overview and a short history of information for each active track-able pedestrian object. Since the tracker keeps a record for each pedestrian orientation, it is possible to detect any changes in the orientation pattern for those who are walking on the curb. Once a change in orientation is detected, the tracker decides whether this change is in the direction of the car path or not, the tracker decides that based on the pedestrian detection location relative to the car path. This sudden change of orientation labels the pedestrian for extra attention from the driver.

IV. RESULTS

The following section will discuss the output result for each module then analyze the overall system performance.

A. BODY LANDMARKS ESTIMATION PERFORMANCE

The CNN model is trained using 80% of the examples in the labeled dataset previously prepared for this task, the other %20 is left for validation. In this work Keras library was used to implement the model and perform the training process. The training is conducted on Intel(R) Core(TM) i7-7700HQ CPU at 2.80GHz with a 16 GB of RAM and equipped with NVIDIA GeForce GTX 1060. The used loss function is Mean Square Error (MSE) with ADAM optimizer and 0.0001 learning rate. The training is performed for 25 epoch with 128 batch size. The model reached an accuracy of %94 on validation. Fig. 11 shows the training accuracy.

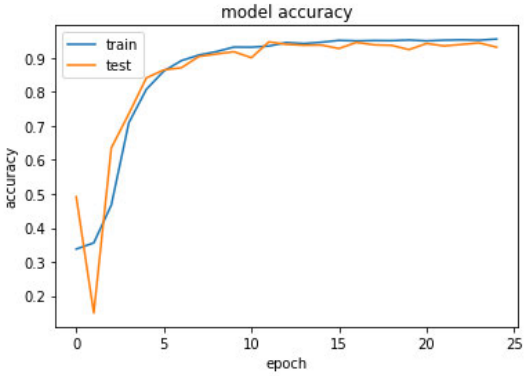
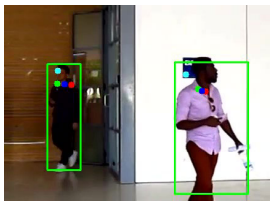


FIGURE 11. Model validation accuracy.

(a) Example of correct landmarks detection, missing point is set to (0,0) of the boundary boxy.

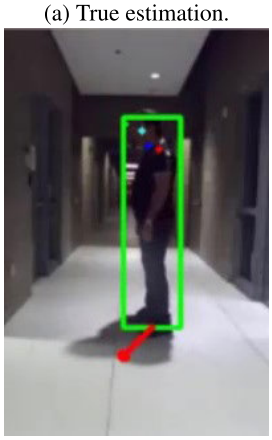


(b) Out of correct and wrong estimation.

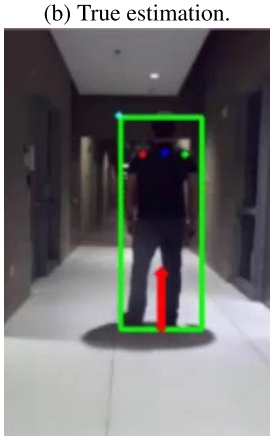
FIGURE 12. Examples of model output.

(a) True estimation.

(b) True estimation.



(c) False estimation.

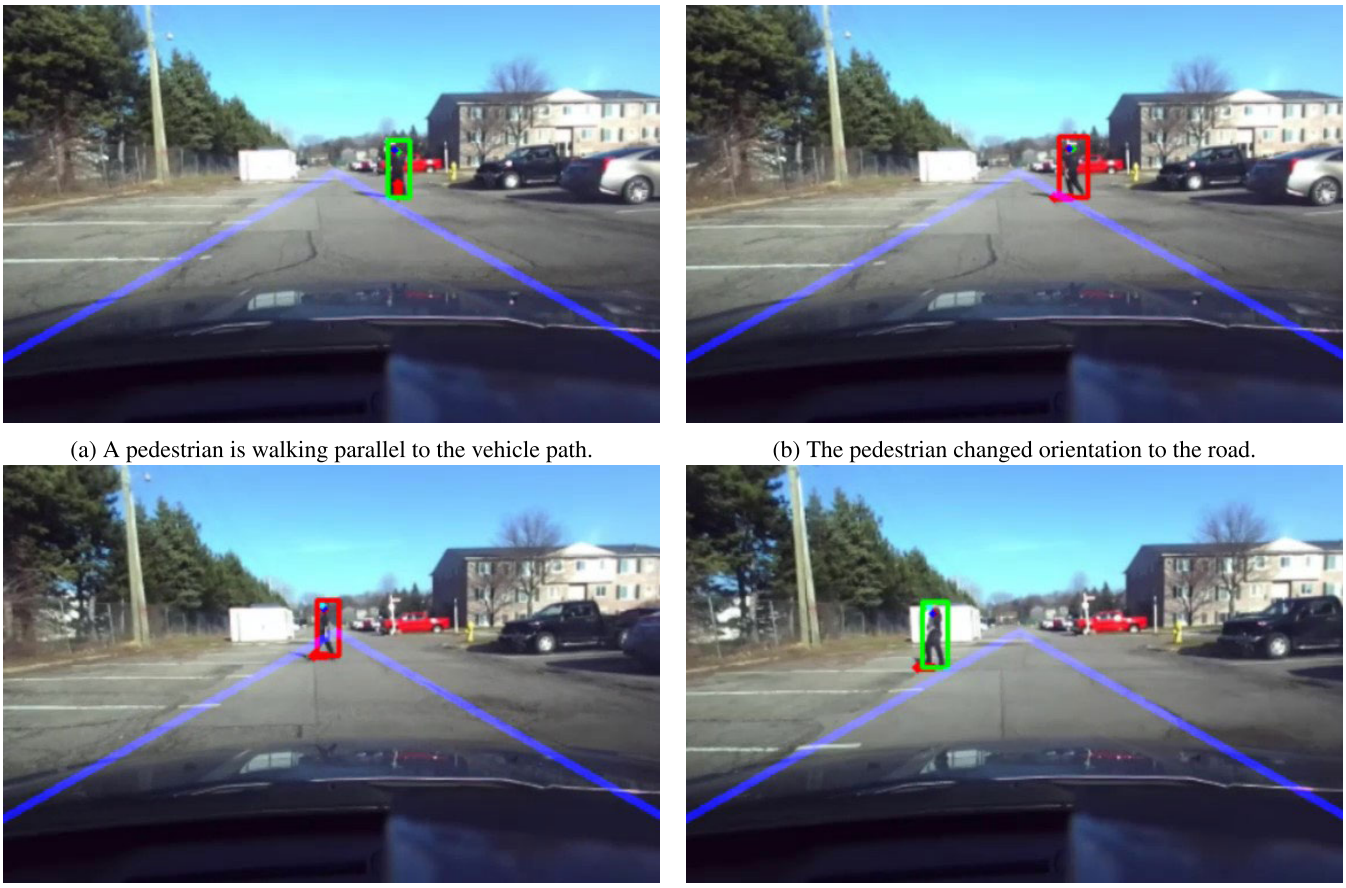


(d) True estimation.

FIGURE 13. Examples of orientation module output, the green box is YOLOv3 pedestrian detection output, the points represents the body landmarks while the red arrow points to the estimated orientation.

An important factor to get high accuracy body landmarks points detection is to provide the model with cropped pedestrian detection that is close to the pattern provided in the training, no big areas for the background. This issue has been taken care of in the detector module, but still exists if the pedestrian spreading arms which is not common but worth mentioning, in this case the boundary box becomes larger including a lot of background and result in wrong landmarks points estimation.

Example of model output, are shown in Fig. 12, more training examples are required to get more generalized prediction in order to avoid miss-prediction as shown in Fig. 12b.



(a) A pedestrian is walking parallel to the vehicle path.

(b) The pedestrian changed orientation to the road.

(c) The pedestrian is crossing and inside the vehicle path.

(d) Pedestrian finished crossing.

FIGURE 14. Examples of tracking a pedestrian walking parallel to the car path then crossing the road in front of a vehicle, green boundary box indicates a safe case while red indicates driver attention or action is required.

B. DEPTH SENSING PERFORMANCE

As previously mentioned, the ZED camera computes the depth information using re-projection from the model as shown in Fig. 8. The ZED camera SDK provides a depth map and a point cloud for four resolutions setup VGA, HD720, HD1080 and HD2K. Point cloud is computationally more expensive than the depth map and thats why this work depends only on the depth map, in average the processing depth map is 20%-30% faster than computing the point cloud, where for example the time required to compute the point cloud for a frame on HD720 is 1.9ms while the 1.7ms is required to compute the depth map.

Testing the depth measuring accuracy shows that higher resolution provides a higher depth accuracy but more computations are required. The best accuracy for HD2K resolution with error in measured distance 20cm while for VGA resolution there might reach 75cm. As a trade of between accuracy and computation time the resolution of HD720 is used, which can have errors up to 35cm.Regarding the proposed method for orientation estimation, the error in measurement will not critically affect the orientation estimation unless the measured region have different error values. The authors in [36] made more detailed study on modeling ZED camera error on TK1 Nvidia development board they mainly referred the error to the hardware and the algorithm.

TABLE 2. Orientation estimation confusion matrix.

		Actual							
		0	45	90	135	180	225	270	315
Predicted	0	42	5	0	0	0	0	0	5
	45	2	38	4	1	0	0	0	0
	90	0	3	44	6	0	1	0	0
	135	1	2	1	37	3	1	0	0
	180	0	0	0	3	45	4	0	2
	225	0	2	0	1	2	37	3	1
	270	1	0	1	1	0	4	46	4
	315	4	0	0	1	0	3	1	38
	Accuracy	84%	76%	88%	74%	90%	74%	92%	76%

C. ORIENTATION ESTIMATION PERFORMANCE

To evaluate the orientation module, the problem is converted to classification problem by categorizing the angle into 8 categories covering the angles 0-360. The evaluation is made by testing 50 test examples for each category and monitoring the model output. Angles are categories in intervals of 45 degrees to reduce the size of the confusion matrix and the evaluation process. Table 2 shows the obtained matrix with accuracy average of 81.75% compared to 82.5% accuracy in [37] using

TABLE 3. Orientation estimation classes precision.

Class	0	45	90	135	180	225	270	315
Precision	80.77%	84.44%	81.48%	82.22%	83.33%	80.43%	80.70%	80.85%

TABLE 4. Pedestrian actions confusion matrix.

		Actual Intention			
		Walking	Crossing		Precision
Predicted	Walking	35	4	1	87.5%
	Intention	5	16	1	72.72%
	Intention	0	0	23	100%
	Accuracy	87.5%	80%	92%	

CNN model with single pedestrian images input and 70.6% compared to [38], Table 3 shows the precision for each class with an average of 81.76%. Fig. 13 shows examples of the module output.

D. OVERALL PERFORMANCE

The purpose of this system is to detect pedestrian intention to cross the road in front of a vehicle, Fig. 14 illustrates an example. The system evaluation is done on 20 video sequences. The videos are filmed by the ZED camera outdoors in sunny and cloudy weathers and indoors with proper lighting. The evaluated events are manually extracted for testing to sequences of 5-10 seconds include pedestrian walking on the road side (40), pedestrian crossing the road (25) and pedestrians walking then crossing the road (20). Table 4 shows the confusion matrix for the classification with the accuracy average of 87% and precision average of 86.74% for the classes. Different approaches in the literature adopt different classification and result verification methods, the work in [39] classify crossing vs not crossing pedestrian actions reached 70% accuracy using CNN extracted features, but this accuracy was increased to 88% using OpenPose extracted features and a SVM/Random Forest classifier. Looking at the [40], the classified actions describing the pedestrian actions are: standing, starting, stopping and walking. The work achieved overall accuracy of 85%.

V. CONCLUSION

Despite the huge efforts done by government and vehicles manufactures to increase vehicles safety U.S. pedestrian fatalities have increased in the last few years. Vehicles are equipped with more advanced safety modules, crash avoidance technologies, pedestrian detection systems and even more equipped with systems to minimize the effect of crashes and reduce injuries such as active hoods and windshield airbags. Many experts are optimistic about the advances in the world of autonomous vehicles; they count on it to reduce pedestrian fatalities through eliminating the human drivers errors.

Even with the advances in computer vision algorithms especially with the great performance of deep learning breakthrough and the incredible result of pedestrian detectors. Pedestrian detection still a challenging task and advanced detector might fail to detect pedestrians in some situations. Pedestrians have variety of physical shapes, heights, widths and clothes, they appear in different environments, backgrounds and weather conditions. This makes the task of predicting human behavior such as the intention to cross the road a more complex and challenging task but at the same time it's a very promising technique in avoiding crashes and reducing pedestrian fatalities.

Pedestrians who intends to cross the road and getting in the path of the vehicle are more critical to the driver than those who walk on the curb and not intending to cross the road. One second prediction for a pedestrian crossing the road ahead of a car driving at typical urban speed of 50 Km/h can provide a distance of 13.8 meters for a vehicle automatic response or a driver response, this time could be even longer if slowing down action is considered before the pedestrian start crossing the road. A couple of seconds of prediction for a pedestrian intention could be critical to avoid crashes or reducing the chance of injury requiring hospitalization. Pedestrians build their decision of crossing the road based on how fast and how far is the coming vehicles, but these decisions might be wrong due to wrong estimation and here comes the driver and auto driver roles.

Recognizing pedestrian behavior from a driver view can be perceived by different actions and signs from the pedestrian. Such actions and signs can be related to head movement when looking at road sides as a sign of waiting for the right moment of crossing, other signs are related to legs movements and body bending toward the street which are a clear indication for starting a walking action. Other signs like low traffic density on the opposite lane can encourage pedestrian to cross the road. Looking at these signs not all of them are easily implemented into a computerized algorithms. This work focuses on implementing the behavior of bending toward the road as computer vision technique.

This paper has described a vision based approach for detecting pedestrians crossing intention to cross the road. The approach uses a combination of deep learning techniques and depth sensing to build a 3D understanding of the pedestrian orientation in reference to the camera view. A very important assumption that held here is the walking orientation is assumed to be the same as the body orientation.

Deep learning models were used to extract important body landmarks that highly related to the body orientation. The method is based on two deep learning components; publicly available CNN pedestrian detector model as YOLO and another CNN model developed and trained by the authors. The training process included dataset collecting, dataset labeling and image augmentation to increase the number of labeled training examples.

The CNN model achieved high validation accuracy of 94% in estimating the body landmarks for the detected pedestrian by YOLO. Moving to the orientation performance the model achieves high accuracy for the main orientations (0,90,180 and 270) degrees but lower accuracy for other orientations. The lower accuracy and miss predictions for

the orientation might come from different sources like the model itself or the ZED camera. The model accuracy can be enhanced using richer dataset of pedestrian to achieve more generalized model while the depth sensing might be enhanced using a higher resolution video format, more processing power or even replace the sensor with better depth sensor. Other solution might include LIDARs and sensor fusion techniques.

Checking the final system output; classifying pedestrians into walking, crossing and intention to cross. The system achieves high accuracy for detecting the crossing pedestrian since it directly depends on the pedestrian detector and the predefined region of interest. Walking pedestrian class also achieves high accuracy but the errors are coming from the orientation estimation module might produce false positive classification into the third class of crossing intention. The overall system performance might be affected by the accuracy of the depth information provided by the ZED camera.

As a summary this system addresses traffic safety, the pedestrians safety in particular in the following manner:

- Providing the vehicle driver with extra awareness of the pedestrian behavior in front of the vehicle.
- Pedestrians who suddenly appear in front of the vehicle are harder to avoid due to the short reaction time window, this system can help the driver and the auto-driver to react faster by slowing down once crossing intention is detected.
- The system also does the task of pedestrians detection in the vehicle path allowing for higher chance of crash avoidance.

VI. FUTURE WORK

This research work opens the door for a lot of ideas and enhancements such as enhancing the system by including advanced depth sensor. Using advanced depth sensor with higher resolution cameras will require more processing power which can be resolved by adding multiple GPUs to distribute the heavy computations required by the CNN models and the stereo vision algorithm. Another idea worth investigated here is using the same pipeline with LIDAR depth map instead of a stereo vision camera to avoid the stereo vision computations. The issue here is the resolution provided by the LIDAR which is lower than the ZED camera.

REFERENCES

[1] R. Retting, *Pedestrian Traffic Fatalities by State: 2019 Preliminary Data*. Washington, DC, USA: Governors Highway Safety Association, 2020.
[2] A. A. Alkheir, M. Aloqaily, and H. T. Mouftah, "Connected and autonomous electric vehicles (CAEVs)," *IT Prof.*, vol. 20, no. 6, pp. 54–61, Nov. 2018.
[3] D. Gerónimo and A. M. López, *Vision-Based Pedestrian Protection Systems for Intelligent Vehicles*. Springer, 2014.
[4] K. Abughalieh and S. Alawneh, "Pedestrian orientation estimation using CNN and depth camera," in *Proc. SAE Tech. Paper Ser.*, Apr. 2020, pp. 1–9.
[5] B. Peng and G. Qian, "Binocular dance pose recognition and body orientation estimation via multilinear analysis," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2008, pp. 1–8.
[6] A. M. Sabatini, "Estimating three-dimensional orientation of human body parts by Inertial/Magnetic sensing," *Sensors*, vol. 11, no. 2, pp. 1489–1525, 2011.
[7] C. Chen, A. Heili, and J.-M. Odobez, "A joint estimation of head and body orientation cues in surveillance video," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 860–867.
[8] J. Chen, J. Wu, K. Richter, J. Konrad, and P. Ishwar, "Estimating head pose orientation using extremely low resolution images," in *Proc. IEEE Southwest Symp. Image Anal. Interpretation (SSIAI)*, Mar. 2016, pp. 65–68.
[9] J. Choi, B.-J. Lee, and B.-T. Zhang, "Human body orientation estimation using convolutional neural network," 2016, *arXiv:1609.01984*. [Online]. Available: http://arxiv.org/abs/1609.01984
[10] K. Abughalieh and S. Alawneh, "Real time 2D pose estimation for pedestrian path estimation using GPU computing," in *Proc. SAE Tech. Paper Ser.*, Apr. 2019, pp. 1–5.
[11] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," 2018, *arXiv:1812.08008*. [Online]. Available: http://arxiv.org/abs/1812.08008
[12] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3D pose estimation from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2500–2509.
[13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Zürich, Switzerland: Springer, 2014, pp. 740–755.
[14] A. T. Schulz and R. Stiefelhagen, "Pedestrian intention recognition using latent-dynamic conditional random fields," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 622–627.
[15] F. Flohr, M. Dumitru-Guzu, J. F. P. Kooij, and D. M. Gavrila, "A probabilistic framework for joint pedestrian head and body orientation estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1872–1882, Aug. 2015.
[16] E. Rehder, H. Kloeden, and C. Stiller, "Head detection and orientation estimation for pedestrian safety," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 2292–2297.
[17] R. Quintero, I. Parra, D. F. Llorca, and M. A. Sotelo, "Pedestrian path prediction based on body language and action classification," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 679–684.
[18] H. Kataoka, Y. Aoki, Y. Satoh, S. Oikawa, and Y. Matsui, "Fine-grained walking activity recognition via driving recorder dataset," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 620–625.
[19] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, "Context-based pedestrian path prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Zürich, Switzerland: Springer, 2014, pp. 618–633.
[20] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Trajectory analysis and prediction for improved pedestrian safety: Integrated framework and evaluations," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 330–335.
[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[22] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767
[23] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[24] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.
[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Amsterdam, The Netherlands: Springer, 2016, pp. 21–37.
[26] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, "A dataset for movie description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3202–3212.
[27] Y. Deng, P. Luo, C. C. Loy, and X. Tang, "Pedestrian attribute recognition at far distance," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 789–792.
[28] A. Dominguez-Sanchez, M. Cazorla, and S. Orts-Escolano, "Pedestrian movement direction recognition using convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3540–3548, Dec. 2017.
[29] Stereolabs Inc. (Mar. 2020). *Stereolabs Zed Camera*. [Online]. Available: https://www.stereolabs.com/zed/

[30] (Mar. 2020). *Labelbox*. [Online]. Available: https://labelbox.com
[31] F. Chollet. (Mar. 2020). *Keras*. [Online]. Available: https://keras.io
[32] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
[33] G. Bradski. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools. [Online]. Available: https://www.drdobbs.com/open-source/the-opencv-library/184404319
[34] L. Matthies, "Dynamic stereo vision," Ph.D. dissertation, Dept. Comput. Sci., Carnegie Mellon Univ. Pittsburgh, PA, USA, 1989.
[35] C. Nvidia, "Nvidia cuda c programming guide," *Nvidia Corp.*, vol. 120, no. 18, p. 8, 2011.
[36] L. E. Ortiz, V. E. Cabrera, and L. M. G. Goncalves, "Depth data error modeling of the ZED 3D vision sensor from stereolabs," *ELCVIA Electron. Lett. Comput. Vis. Image Anal.*, vol. 17, no. 1, p. 1, 2018.
[37] K. Kumamoto and K. Yamada, "CNN-based pedestrian orientation estimation from a single image," in *Proc. 4th IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2017, pp. 13–18.
[38] K. Hara, R. Vemulapalli, and R. Chellappa, "Designing deep convolutional neural networks for continuous object orientation estimation," 2017, *arXiv:1702.01499*. [Online]. Available: http://arxiv.org/abs/1702.01499
[39] Z. Fang and A. M. Lopez, "Is the pedestrian going to cross? Answering by 2D pose estimation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1271–1276.
[40] R. Quintero, I. Parra, D. F. Llorca, and M. A. Sotelo, "Pedestrian intention and pose prediction through dynamical models and behaviour classification," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 83–88.

KARAM M. ABUGHALIEH received the M.Sc. degree in electrical engineering from Princess Sumaya University for Technology (PSUT), Amman, Jordan, in February 2011. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Oakland University. He also worked on object detection and tracking in the master's degree thesis for UAV applications. He also works as a Teaching and Research Assistant with Oakland University. He has obtained a very good experience in embedded systems design.

SHADI G. ALAWNEH (Senior Member, IEEE) received the B.Eng. degree in computer engineering from the Jordan University of Science and Technology, Irbid, Jordan, in 2008, and the M.Eng. and Ph.D. degrees in computer engineering from the Memorial University of Newfoundland, St. John's, NL, Canada, in 2010 and 2014, respectively. Then, he was a Staff Software Developer with the Hardware Acceleration Lab, IBM, Canada, from May 2014 to August 2014. After that, he was a Research Engineer with C-CORE, from 2014 to 2016, and became an Adjunct Professor with the Department of Electrical and Computer Engineering, Memorial University of Newfoundland, in 2016. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Oakland University. He has authored or coauthored scientific publications, including international peer-reviewed journals and conference papers. His research interests include parallel and distributed computing, general purpose GPU computing, parallel processing architecture and its applications, autonomous driving, numerical simulation and modeling, and software design and optimization. He is a Senior Member of the IEEE Computer Society.