Design of a real-time computer-vision and gaze-based pedestrian warning system

By

Lokkeshver Thirunavukkarasu Kumaaravelu

in partial fulfilment of the requirements for the degree of

Master of Science
in Mechanical Engineering

at the Delft University of Technology,
to be defended publicly on Tuesday May 31, 2022 at 15:00

Student Number:	4899490
Supervisors:	Dr.ir. Joost de Winter, TU Delft Dr. Dimitra Dodou, TU Delft Dr. Pavlo Bazilinskyy, TU Delft Ir. Vishal Onkhar, TU Delft Wilbert Tabone M.Sc., TU Delft
External committee:	Dr.ir. Yke Bauke Eisma, TU Delft

TU Delft

Abstract

Vulnerable road users account for more than 50% of traffic fatalities, and among these, pedestrians are the most susceptible to fatalities due to their distraction and misperception of other road users. To mitigate their plight, systems that warn drivers and pedestrians in case of a possible collision have been developed. Among systems that focus on the pedestrian's perspective, existing concepts are capable of predicting collisions but lack elements that monitor the visual attention of pedestrians. We address this gap by developing a gaze-based pedestrian warning system based on the Tobii Pro Glasses 2, a head-mounted eye-tracker. The system consists of: (1) a custom trained fast neural network (YOLO v4) on the KITTI object detection dataset that processes the video feed of the eye-tracker to detect approaching vehicles and (2) a module that uses the pedestrian's gaze to identify whether their attention falls on the closest moving vehicle that is approaching the pedestrian, both in real-time. If the pedestrian does not look at the approaching vehicle, they are given an auditory alert that warns them of a possible collision. In a pilot study conducted on a busy road in an urban environment, the system was evaluated under different pedestrian walking speeds and gaze behaviours to test the algorithm's robustness. The pilot study revealed that our system alerted the inattentive pedestrian with an accuracy of 67%. The mean vehicle detection accuracy and a mean moving vehicle identification accuracy from the pilot were 93% and 60%, respectively, a promising result given the use of only a mono camera. Despite the use of computer vision techniques, the system worked at an inference speed of 50 FPS due to the multi-processing capabilities of our algorithm. Our efforts are a first step in developing pedestrian warning systems based on eye-tracking technology to improve road safety in the future. The algorithm (Python-based) code used for this work has been made publicly available.

1 Introduction

Road traffic accidents result in 1.35 million fatalities each year worldwide, and 54% of those deaths are vulnerable road users such as pedestrians, cyclists, and motorcyclists (World Health Organisation, 2020). Among these, the rate of pedestrian fatalities is higher than that of cyclists, making them the most vulnerable group of road users (European Commission, 2021). Most pedestrian fatalities occur due to frontal impacts with vehicles in urban environments and on non-intersecting roads (NHTSA, 2019).

A possible cause for pedestrian deaths is their inattentiveness (Mwakalonge et al., 2015) and reduced situational awareness due to distracted walking, mobile phone usage, and rubbernecking, thereby failing to utilise the relevant information on the road (World Health Organisation, 2013; Otte et al., 2012). Thompson et al. (2012) studied pedestrians' sociological and technological distractions on crosswalks via recorded videos and found that pedestrians using mobile phones did not look at either side of the road before crossing and had slower crossing times. Similarly, Mwakalonge et al. (2015) also found that pedestrians who text while walking are 3.9 times more prone to unsafe crossing behaviour than a visually attentive pedestrian.

Eye-tracking involves analysing eye movements and positions and measures the point of gaze or eye motion relative to the head. Leveque et al. (2020) reviewed pedestrian eye-tracking studies and stated that eye tracking of pedestrians helps in understanding their perception and cognition of the environment. Dey et al. (2019) studied pedestrians' willingness to cross the road based on their gaze patterns and found that pedestrians anticipated a vehicle's motion by looking at different parts of it and the surrounding environment before making a crossing decision. Trefzegar et al. (2018) studied the gaze behaviour of pedestrians and cyclists using eye-trackers and reported that pedestrians look at either side of the road and their crossing trajectory before crossing the road. Furthermore, De Winter et al. (2020), in their eye-tracking study in a parking garage, found that pedestrians look for visual cues such as wheel movement or taillights that turn on or off to detect vehicle motion, which contributes to their safety while walking.

Pedestrian warning systems aim to improve pedestrian safety by increasing their cognitive ability through assistive technologies and sensors (Hasan et al., 2022). Pedestrian warning systems could be found in smart vehicles (ADAS equipped), pedestrian-held devices and infrastructure. We focus on systems that use pedestrian-held devices (smartphones, wearables) to warn the pedestrian in case of a possible collision with an oncoming vehicle. The warning modalities include auditory, vibratory, visual and multimodal alerts. The system uses multiple in-built sensors of pedestrian-held devices (smartphones, wearables) to sense contextual data, pedestrian's location, motion and surrounding objects. These systems also receive alerts to the pedestrian's smartphone (via Wi-Fi/LTE) about the approaching vehicle through vehicle-to-pedestrian (V2P) communication and also provide information about the pedestrian (location, activity) to the driver. The system computes the possibility of a collision based on the sensor data from the pedestrian-held device and triggers an alert to the pedestrian. The systems based on pedestrian-held devices are elucidated below.

Smartphone's camera (front/rear) of the pedestrian is used to detect possible collisions with oncoming vehicles and to detect the pedestrian's attention. Wang et al. (2012) developed *Walksafe*, an android application that detects vehicles from the smartphone's rear camera during an active call. The vehicles are detected by an Adaboost (ML) algorithm trained on front and rear images of vehicles. The application could detect vehicles up to a maximum distance of 50 m, and the detection speed of the algorithm was 8 FPS.

Li et al. (2018) created *Safe Walking*, an android application that uses the front camera, accelerometer, and gyroscope to ensure pedestrians pay attention to the road ahead. The algorithm checks if the pedestrian has been staring at their phone screen using the front camera and detects the pedestrian's walking speed using an accelerometer and gyroscope. The algorithm triggers a vibratory alert if the pedestrian's walking speed is above 1.2 mph (1.9 km/h) and if they look at the screen for more than 6 seconds continuously.

Location-based systems use GPS coordinates from pedestrians' and drivers' smartphones to detect possible collisions and warn them. These systems also check if the pedestrian is distracted from the smartphone's screen activity. Lin et al. (2016) devised *pSafety*, an Android application that sends visual alerts to pedestrians and drivers. The positioning error of GPS is reduced by the Sector Overlap Detection algorithm, which uses sectors to estimate the overlap of pedestrian and vehicle. The pedestrian and driver with the highest duration of sector overlap are warned.

Won et al. (2020) devised *SaferCross*, a smartphone application that warns the driver when the pedestrian ignores the alert. The application predicts the pedestrian's intention to cross based on their location coordinates (nearing a pedestrian crossing) and then checks the pedestrian's distraction by phone-viewing activities based on their screen status. The collision is predicted by obtaining the vehicle's location and its speed from the application in the driver's smartphone (when in the pedestrian's vicinity).

Wireless communication-based systems rely on Wi-Fi communication to predict collisions between pedestrians and vehicles. Hwang et al. (2014) proposed *the Safety-Aware Navigation App (SANA), a cloud-based smartphone application that warns* drivers and pedestrians. The smartphone application transfers pedestrians and vehicles' location, direction, and speed to a cloud-based node using Wi-Fi/LTE that calculates the collision probability. An auditory or vibratory alert message is sent by the smartphone of the pedestrian and driver when a high probability collision is detected.

Watanabe et al. (2019) proposed a system based on the neighbour discovery protocol (NDP) to alert the pedestrian about an oncoming vehicle. The NDP is a protocol used for IPv6 traffic that allows different nodes on the same link to advertise their existence to their neighbours and learn about the existence of their neighbours. The system consists of a pedestrian-held device and a device on the vehicle that both transmit and receive beacon signals (based on IEEE 802.15.4g standard) to identify the presence of an oncoming vehicle. An auditory warning is issued when the pedestrian's device receives a response beacon sent by the

vehicle (when the vehicle's speed is above a predefined threshold). The vehicles which are not in the field of view of pedestrians (e.g., vehicles around a corner) can also be detected by this system as beacons could travel through surfaces.

Hasan et al. (2021) developed *StreetBit*, to warn a distracted pedestrian through audio or visual alerts using Bluetooth beacons installed at intersections. The signal strength is used to estimate the distance to the intersection and estimate the pedestrian's location. Pedestrian movement and phone status (screen activity) data are acquired to monitor the pedestrian for distractions while walking. The system activates when the pedestrian is within a 20 m radius from the intersection, and the warning is only triggered when the pedestrian is distracted (i.e., the mobile phone screen is active) and within an 8 m radius.

Sound-based systems use microphones to detect vehicle sounds and alert the pedestrian. De Godoy et al. (2018) devised *Pedestrian Audio Wearable System* (PAWS), which uses multi-channel audio sensors to detect cars based on their honking, engine noise, and tire noise. The system detects the presence, distance, and direction of an oncoming vehicle every 91 ms through ML algorithms. The pedestrian receives auditory and vibratory warnings on their smartphone in real-time. PAWS' performance in detecting oncoming vehicles was evaluated in a realistic environment, and a distracted pedestrian's ability to detect oncoming vehicles was simulated. The distracted participant missed 36%, 32%, and 18% of the vehicles on a campus street, highway, and metropolitan area, respectively, whereas the system missed only 1%, 4%, and 3% of the vehicles in the respective environments.

Similarly, Lee et al. (2018) devised a smartphone application that predicts the oncoming vehicle via a microphone connected to the smartphone and warns the pedestrian. The audio input is classified using machine learning classifiers, Knn, decision tree, random forest and multi-layer perceptron. The application could detect vehicles moving at a maximum speed of 50 km/h and at a background noise of 50 dB. The application could identify vehicles at a rate of 97%. The authors demonstrated the capabilities of the system and suggested real-time usage.

Additionally, Vehicle-to-Pedestrian (V2P) communication systems compute the possibility of a collision with the vehicle and communicate this to pedestrians using safety messages sent via WLAN or infrastructure-based communication due to the low latency of WLAN communication (Engel et al., 2013; Sewalkar et al., 2019). V2P systems rely on a vehicle-based device to compute the possibility of a collision and warn the driver, and the pedestrian. These systems commonly use GPS coordinates to calculate collision probability and create auditory or visual alerts for both parties (Wu et al., 2014). Furthermore, although systems work efficiently in Non-Line-of-Sight (NLoS) and Line-of-Sight (LoS) scenarios, the blockage of the Wi-Fi signal by the human body and other environmental factors restrict the range of WLAN, hindering the performance of these systems (Anaya et al., 2014).

From the above, it can be concluded that the majority of pedestrian-based warning systems use the pedestrian's hand-held device (such as a smartphone) to predict and warn the pedestrian about a possible collision with an oncoming vehicle. The warning modality can be visual, auditory, haptic or multimodal, and the systems work effectively in NLoS and LoS scenarios. However, the situational awareness of the pedestrian is rarely monitored. In their study examining eye movements for measuring situational awareness, De Winter et al. (2018) found that eye movements could be used in a real-time assessment of situational awareness. Likewise, in their real-time eye-tracking study to predict situational awareness, Zhou et al. (2022) found that eye-tracking could measure visual attention and serve as a direct measure of situational awareness.

The existing pedestrian warning systems do not monitor the visual attention of the pedestrian on the approaching vehicles. This study aims to develop a gaze-based warning system for pedestrians that uses an

eye-tracker to monitor a pedestrian's situational awareness (via measuring visual attention) in real-time through their gaze behaviour. The system aims to alert a distracted pedestrian in case of a possible collision with an oncoming vehicle. The system uses computer vision techniques to identify vehicles of interest and is hindered by the limited field of view of the eye-tracker camera. The system rests on the assumption that the pedestrian would turn his head to compensate for the limited field of view of the eye-tracker. The study also aims to replicate the effect of a stereo camera on the monocular camera of the eye-tracker using deep learning networks to distinguish vehicles that are in motion. The study aims to develop computationally efficient computer vision algorithms for the system to work in real-time in an outdoor environment. The system also assumes that the pedestrian notices the vehicle upon giving an alert. In the following sections, the system design is explained, and its working is demonstrated in real-time in naturalistic use-case scenarios. The system behaviour is observed, and its accuracy is evaluated. Finally, improvements that could be made to make the system commercially available are listed.

2 System design

2.1 Overview

The system uses a head-mounted eye-tracker (Tobii Pro Glasses 2) equipped with a scene camera and infrared (IR) eye cameras. The eye-tracker streams a video from the pedestrian's point of view and the pedestrian's gaze point in 2D pixel coordinates relative to the video feed. The recording unit of the eye-tracker transmits both data streams via Wi-Fi to a laptop for real-time processing.

We developed a custom Python-based algorithm to identify vehicles in every frame that pose a collision risk to the pedestrian wearing the eye-tracker as they are about to cross the road. This identification happens in every video frame as follows: the image recognition algorithm detects vehicles in the video frame and identifies the closest moving vehicle in each frame, and the system checks the duration of the pedestrian's gaze fixation on that vehicle and triggers an auditory alert to the pedestrian if the gaze fixation duration is lower than 300 ms. We set this value by trial and error within a range of typical fixation durations identified by literature (Negi et al., 2020; Galley et al., 2015). The assumption is that below this threshold, no visual information is acquired by the pedestrian; i.e., they have not seen the approaching vehicle, and therefore, crossing is unsafe. The auditory warning is played on a Bluetooth speaker attached to the belt of the pedestrian. The system works in real-time with a speed of 50 FPS with a latency of 25 ms through the use of multi-processing and multi-threading techniques in Python. The system's use case is illustrated in Figure 1.

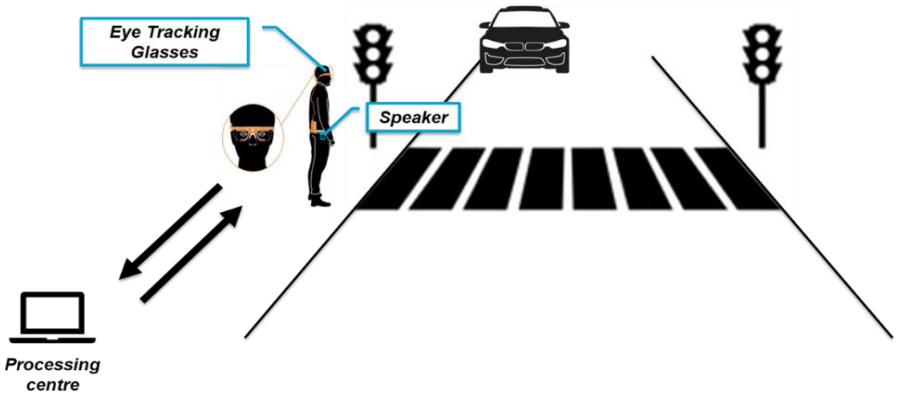


Figure 1. Illustration of the system's use case. The image depicts a pedestrian equipped with our system and with the intention to cross the road.

The system contains six modules: an initialisation module, a live data module, an object detection module, a moving object classification module, an awareness monitoring module and an output module. The workflow of the system is depicted in Figure 2.

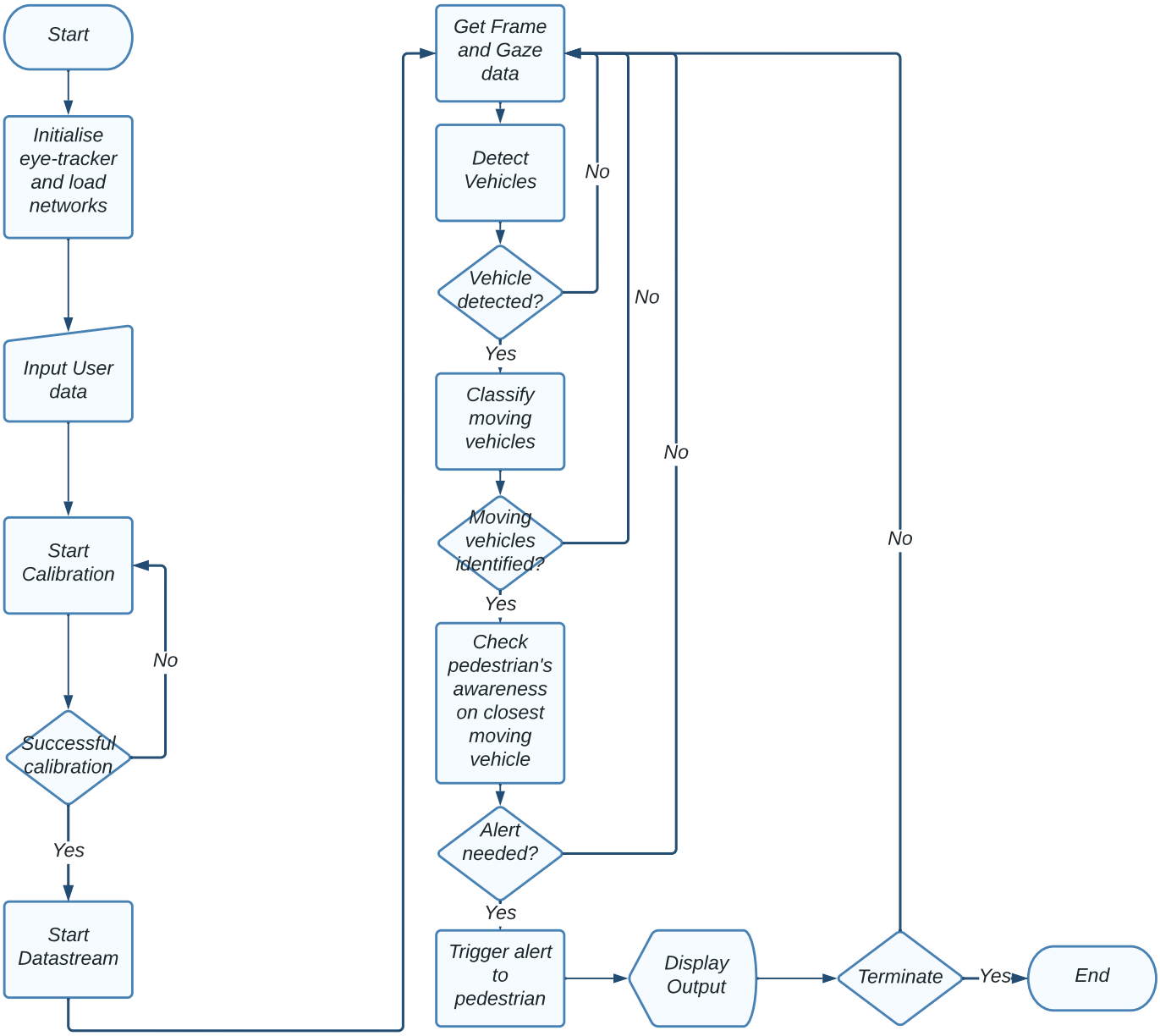


Figure 2. The figure illustrates the workflow of the system with different modules.

2.2 Initialisation module

The initialisation module imports dependencies and initialises the Tobii Pro Glasses 2 Controller, the YOLOv4 network (Bochkovskiy et al., 2020), and the monodepth2 network (Godard et al., 2019) with their trained weights for object detection and depth estimation, respectively, through function call statements (see Sections 2.4 and 2.5.3 for estimating the training weights). The Tobii Pro Glasses 2 are controlled through a Python-based controller (De Tommaso et al., 2019) that consists of functions to access the Tobii Pro Glasses 2 API (v.1.3) (Tobii Pro AB., 2015). The eye-tracker streams data to the laptop via the Real Time Streaming Protocol (RTSP). The initialisation module requests the pedestrian to perform a gaze calibration, which is carried out by the standard calibration method of the Tobii glasses (i.e., using a printed bull's eye held at arm's length). If the calibration is successful, an audible confirmation is issued. In case of an unsuccessful calibration, the module prompts the experimenter for a recalibration. The initialisation module waits for confirmation from the experimenter to start the data streaming process to acquire eye-tracker data in real-time.

2.3 Live data module

2.3.1 Gaze Synchronisation

The algorithm receives the video and gaze data from the eye-tracker in real-time. The monocular camera of the eye-tracker samples videos at 50 Hz, and the IR camera samples gaze at 50 Hz. The latency of the eye tracker in transmitting the video and gaze sample is 5 ms. The video is read using OpenCV, and the video frames are used as inputs for image recognition algorithms (YOLOv4 and monodepth2). The video stream contains video frames and their timestamps (in milliseconds). The gaze data stream contains gaze data, timestamp (in microseconds), status indicator, gaze index and latency.

The gaze timestamp is based on the internal clock of the recording unit in the eye-tracker, whereas the video timestamp is that of the video stream read by the algorithm. The video stream is being read by the algorithm first, and the gaze data is buffered next. Also, the timestamps do not match as they have different base clocks. Hence the right gaze data should be matched with its corresponding video frame. According to the Tobii Pro Glasses 2 API guide, the offset between the gaze presentation timestamp and the presentation timestamp of the video frame is used to match the gaze data with its corresponding video frame (Tobii Pro AB., 2018). The presentation timestamp (PTS) is the timestamp metadata field in a MPEG transport stream that indicates the time of presentation (Yuste et al., 2015).

The gaze presentation timestamp is obtained from the PTS Sync Package of the Tobii Pro Glasses 2 API. Because the video stream does not contain a PTS, the PTS is calculated as follows (Ng., 2011):

$$Video\ PTS = \frac{1}{FPS} \times Frame\ Number \times Resolution\ of\ PTS$$
$$Video\ PTS = Video\ Timestamp \times Resolution\ of\ PTS$$

The offset in the PTS values of the gaze and video is 57566 between any two successive keyframes (this difference was estimated based on the observation of the live data feed). Therefore, the offset between the PTS values of gaze and video frames should be within 57566 for the gaze marker to be matched with its corresponding video frame. The condition for matching the gaze data and video frame is as follows:

$$|PTS\ of\ Gaze - PTS\ of\ Video| < 57566$$

2.3.2 Real-time processing

Real-time data processing is achieved through multi-processing and multi-threading techniques in Python. The algorithm depends on external devices for input (I) (eye-tracker) and output (O) (auditory alert and display of video stream) processes and the CPU for computation. Because the Python interpreter allows only one process to execute at a time, and to overcome this limitation, multi-processing is used for I/O processes, and multi-threading is used for computations. The live data module and the output module are initialised as separate Python processes using multi-processing for faster processing as these processes are bounded by Input/Output. The object detection, moving/non-moving classification, and awareness detection modules are initialised together as a single Python thread (runs on the main thread) using multi-threading to do the processing sequentially. The data from each process is stored in a FIFO queue of maximum size 'one', and the data is acquired from the queue sequentially. The system runs all the modules in coherence using this technique and helps achieve real-time working speeds. The latency in processing each frame by the system is depicted in Figure 3. The eye-tracker has a latency of 5ms, and the algorithm has a processing time of 20 ms resulting in a net latency of 25 ms.

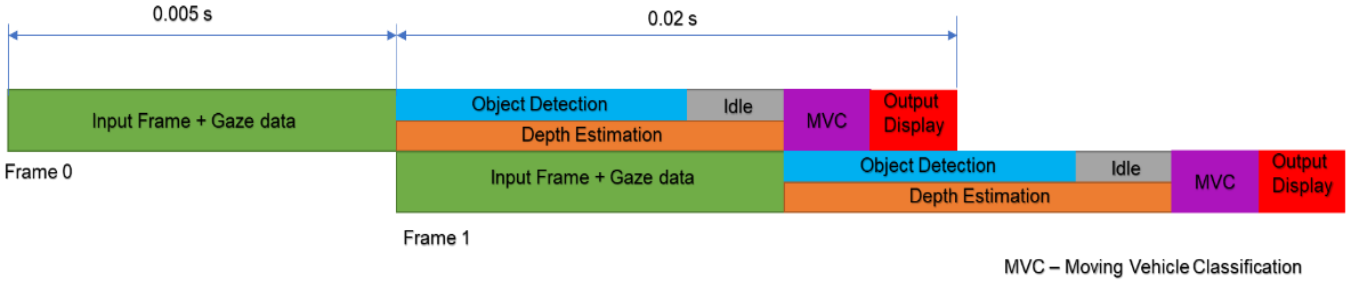


Figure 3. The latency involved in the system while processing the input data from the eye tracker.

2.4 Object detection module

An object detection algorithm should have a high detection speed to perform real-time detections. Object detection algorithms are categorised into single-stage and two-stage based on the number of networks used in the pipeline for region proposals, object classification and localisation (Wang et al., 2019). Single-stage object detection algorithms outperform two-stage object detection algorithms in detection speed but have less accuracy (Jiao et al., 2019). YOLOv4 works at real-time inference speeds (with minimum system requirements) and has an accuracy higher than SSD and RetinaNet in detecting vehicles in frames (Qiao et al., 2020, Kim et al., 2020). YOLOv4 (Bochkovskiy et al., 2020) consists of a Spatial Pyramid Pooling (SPP) block over a CSPDarknet53 backbone, a PANet path aggregation neck, and only the dense prediction head is derived from YOLOv3. Compared to YOLOv3, YOLOv4 has a 10% higher average precision and 12% higher inference speed.

Because the algorithm has to detect different types of vehicles a pedestrian might encounter in an urban environment, the KITTI 2D Object Evaluation Dataset (Geiger et al., 2012) was chosen to train the YOLOv4 algorithm. The dataset consists of 14970 images with objects labelled 'Car', 'Van', 'Truck', 'Pedestrian', 'Cyclist', Person_sitting', 'Tram', and 'Misc'. The KITTI training labels have 14 values for each object by default. Only the object class name and its bounding box coordinates were kept from these values since YOLOv4 only requires these two values. We trained the algorithm on KITTI dataset images because there were no pre-trained weights suitable for the algorithm. An 832 x 832 network size with a batch size of 64 for 16000 (number of classes x 2000) iterations as high network size results in high precision. The trained model is available in the GitHub repository (Thirunavukkarasu Kumaaravelu, 2022). Table 1 shows the resulted average precision (area under the precision-recall curve) of the trained classes for each object used in the system.

Table 1. Average precision of trained classes at a confidence threshold of 0.5

Object class	**Average precision @ conf = 0.5 (%)**
Car	95.52
Van	94.60
Truck	98.78
Pedestrian	72.21
Person sitting	59.52
Cyclist	83.65
Tram	93.63
Misc	89.81
Mean average precision @ conf = 0.5: 85.79	

For object detection, the image frame from the scene camera is resized to a network size of 416 x 416 px to increase detection speed as smaller network sizes have higher detection speeds. Since YOLOv4 is a fully convolutional network, image resizing does not affect its precision. By default, the algorithm returns the labels,

confidence percentages, and coordinates of vehicles. The output from the object detection module is visualised in Figure 4.

Figure 4. Object detection in a video frame. The object detection algorithm detects the vehicles in the frame and is highlighted by bounding boxes with the object class and confidence. The red marker signifies the gaze of the pedestrian in that frame.

2.5 Moving object classification module

The moving and non-moving objects between sequential frames are distinguished by the following three steps: (1) estimation of the camera pose in the environment, (2) matching the same objects between the two frames, and (3) geometric transformation of the matched objects.

2.5.1. Estimation of the camera pose in the environment

The camera pose is the position and orientation of the camera with reference to the world coordinate system and is given by the rotation and translational matrices. The camera pose is estimated using the change in pixels between two subsequent frames. The change in pixels is estimated by first identifying features (i.e., parts or patterns in an object) in the frame by Oriented FAST and Rotated BRIEF (ORB) detector (Rublee et al., 2011) and then matching the features between the two frames by the Brute Force (BF) Matcher (OpenCV: Feature Matching, n.d.) The BF Matcher returns the corresponding points in the two frames in pixel coordinates. These points are used as input together with the camera parameters to calculate the Essential matrix, a 3 x 3 matrix containing geometric relations between the two frames. The camera parameters are obtained from the manufacturer. Using the Singular Value Decomposition (SVD) of the Essential matrix, the Rotation matrix *R (3x3)* and Translation matrices *t (3x1)* are calculated, which define the camera pose with respect to the two subsequent frames.

$$Pose\ (P) = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

2.5.2 Matching the same objects between the two frames

YOLOv4 detects multiple objects in the frame and their locations (bounding box coordinates), but it does not uniquely identify the objects in the frame. Hence, these objects needed to be matched between two subsequent frames. For matching the same objects between two subsequent frames, the coordinates of the detected objects of the same label (e.g., 'Car') are compared between the two frames. The coordinates with the smallest distance between their origins are then attributed to the same object. In case of an unequal number of detections between two subsequent frames, the ones with the lowest confidence threshold in the previous/current frame are discarded.

2.5.3. Geometrical transformation of the matched objects

The algorithm loops for each vehicle in the frame to check if it is a moving or non-moving vehicle. The detected objects with labels' Car', 'Van', and 'Truck' are checked for movement between the previous and current frames for every frame in the video stream. The vehicle is masked (so that only the vehicle is visible in the frame for feature matching) in both frames based on their respective bounding box coordinates, and corresponding points are identified using the ORB detector and BF Matcher. The corresponding points (points in the detected vehicle in pixel coordinates as seen in Fig. 5) represent the object in image coordinates. To transform these points into camera coordinates, they are subtracted by the optical centres of the camera (Szeliski, 2010).

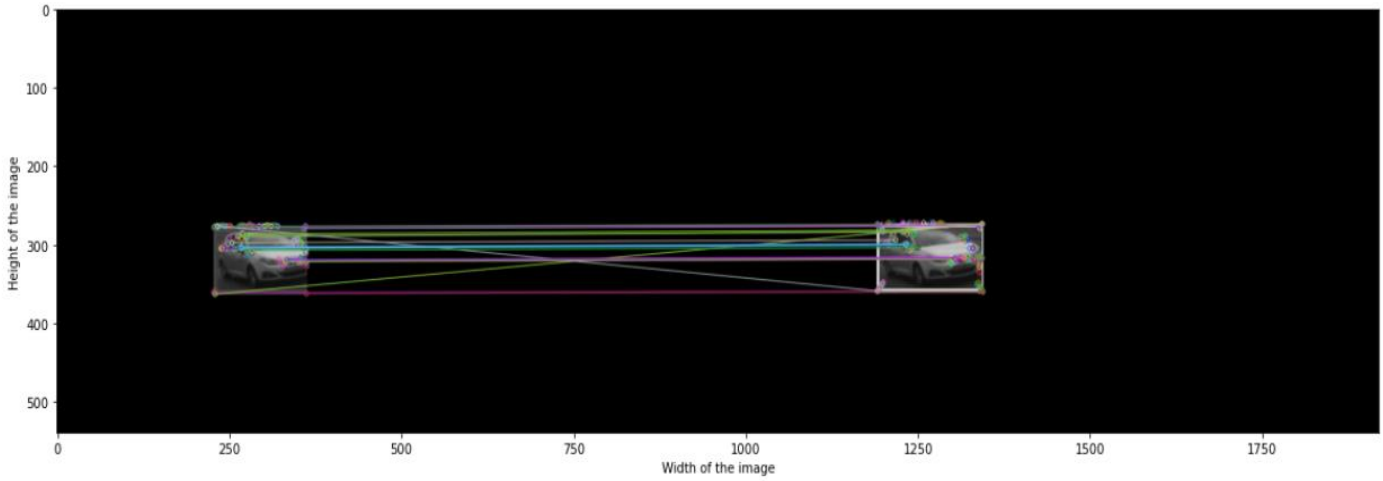


Figure 5. The figure illustrates the corresponding points matched between the previous and the current frames by the BF matcher.

The eye-tracker houses a monocular scene camera, and the camera coordinate in the 'z-axis' (depth) is estimated through a monocular depth estimation network "monodepth2" (Godard et al., 2019). The depth value for each pixel in the output video frame from the monocular scene camera is estimated by the monodepth2 network (see Figure 6) and is used in this calculation. This monocular depth estimation network predicts the depth of the input image up to 80 m (Godard et al., 2019). The network is trained to predict the depth of the input image from the outlook of a trained image. The weights trained by Godard et al. from the KITTI Benchmark suite are used for predicting the depth.

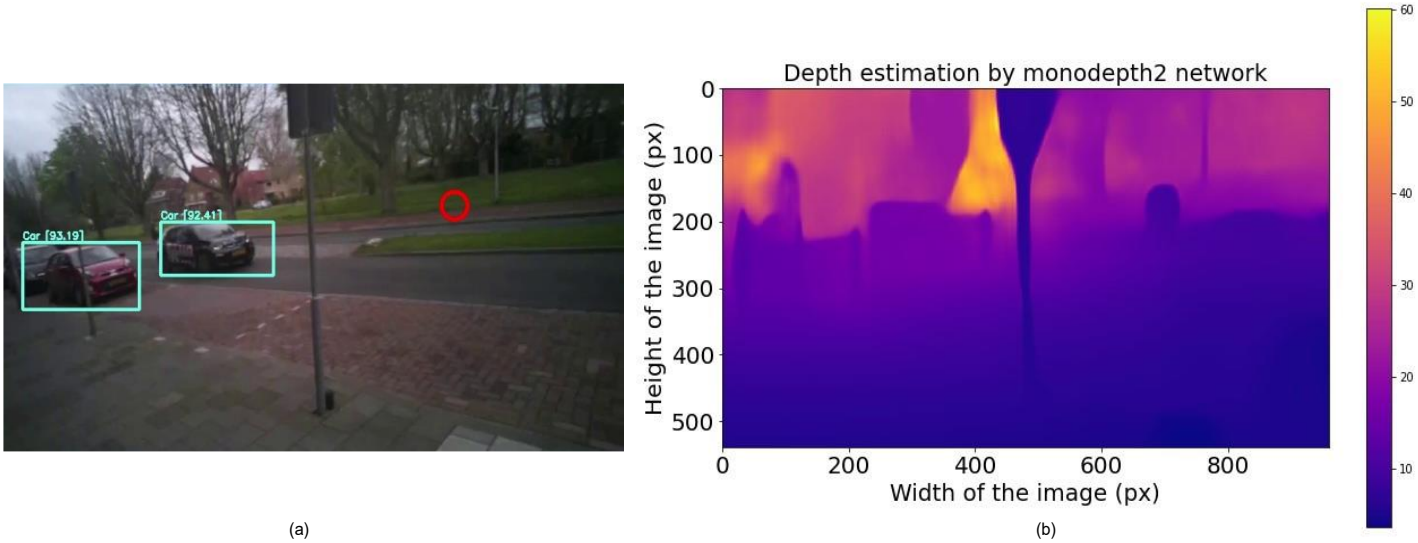


Figure 6. Depth estimation map for the image on the left by the monodepth2 network. The distance in metres for each pixel in the image frame is shown in the colour bar in Figure 6 (b).

The corresponding points of the previous frame in camera coordinates are multiplied with the Rotation and Translation matrices (camera pose) to estimate the pose of the image in the current frame. The next pose is then multiplied with the Intrinsic matrix (a 3x3 matrix with the camera's focal lengths and optical centres) to reproject the points in camera coordinates into image coordinates. The estimated points in image coordinates are compared with the points in the current frame to check the translational distance (in pixels) moved by the object. The translational distance moved (in px) by the object is calculated by the difference in the pixel coordinate of the current frame to the estimated pixel coordinate.

If the translational distance moved (in px) is above the threshold value, the object is classified as a moving object. The translational distance threshold for the object in the image was set to 1000 pixels. This threshold was determined by carrying out the steps mentioned above on subsequent frames in which a cyclist travelled at a constant speed (~15 km/h) for a set distance (7 m). The cyclist performed ten trials, and the values of pixels shifted between two consecutive frames were noted against the constant speed during the trial. Thus, the number of pixels shifted is computed, and the pixel value is extrapolated to the speed limits of the urban environment. Any vehicle that exceeds this set threshold is termed a moving vehicle.

2.6 Awareness monitoring module

The awareness module checks for the pedestrian's awareness at 50 Hz. The algorithm checks if the pedestrian's gaze is within the bounding box of the closest moving vehicle, which is identified by the distance between the vehicle's bounding box origin and the centre of the frame. This module relies on gaze data from every frame to visualise where the pedestrian looks in the environment at any instant. When the pedestrian fails to notice the vehicle (within the field of view), i.e., gaze duration within the bounding box of the closest moving vehicle is less than 300 ms, the module triggers an auditory alert transmitted via the Bluetooth speaker to the pedestrian. The module does not track the vehicles in the frame, and hence when a pedestrian rotates his head in a left/right/left manner, the pedestrian may encounter more than one warning when he does not notice the exact vehicle in his field of view. Figure 7 depicts an instance where the pedestrian fails to notice an oncoming car.

Figure 7. A moving car and the pedestrian's gaze at an instant in the video. The image illustrates the gaze of the pedestrian in the environment.

2.7 Output Module

The output module displays the pedestrian's field of view to the experimenter via his laptop display. The display output contains the pedestrian's gaze marker and the detected vehicles with bounding boxes for each frame. The text output containing the object class name, confidence and bounding box coordinates, moving

status, pedestrian awareness status, and gaze marker coordinates are displayed for each frame in the terminal output window (for Windows users - Command Prompt). This module helps visualise the system's working to the experimenter, as seen in Figure 8.

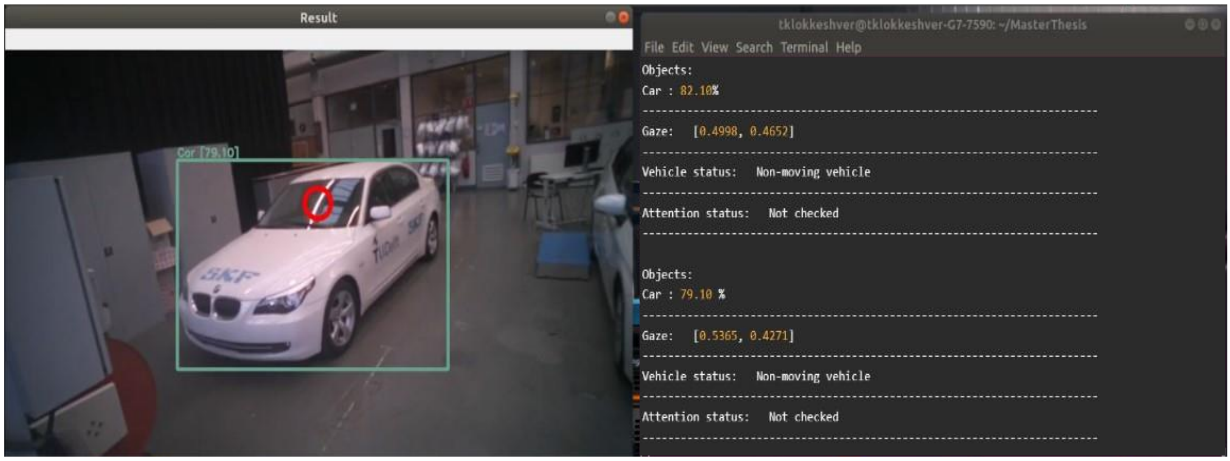


Figure 8. The figure describes the experimenter's point of view when a pedestrian is using the system. The output video and text in the terminal window could be seen in the right and left portions of the figure.

3 Validation

3.1 Methods

3.1.1 Participants

One experimenter played the role of a pedestrian who was about to cross the street, while the other experimenter launched and monitored the pedestrian warning system via a laptop.

3.1.2 Equipment

A head-mounted eye-tracker Tobii Pro Glasses 2 was used to stream video of the pedestrian's point of view, and their gaze point in 2D pixel coordinates relative to the video feed through the scene camera and infrared (IR) eye cameras, respectively. The scene camera captures frames at 50 Hz and a resolution of 960 x 540 px with a 90º diagonal field of view. Gaze data is also recorded at 50 Hz. The recording unit of the eye-tracker transmits both data streams via Wi-Fi to a laptop for real-time processing. The laptop uses a 9th generation Intel i7 processor with 16GB of RAM and 6GB of NVIDIA RTX2060 graphics. JBL clip4 is the Bluetooth speaker used to provide auditory warnings to the participant, and it was connected to the laptop. Another Bluetooth speaker Bose Revolve was connected with the iPad Air 4 to play the voice instructions for the pedestrian during the study.

3.1.3 Pilot study

A pilot study was conducted to demonstrate and evaluate the system's working in real-time. The study was conducted outdoors on a pavement next to a busy street in an urban area, and the experimenters ensured that there were no passers-by in the pedestrian's way during the trials and also ensured that moving traffic was in view. This experimental setting aimed to recreate a realistic scenario in which a pedestrian was about to cross a street at an unmarked crossing. The location was well-lit by natural light, and tinted lenses were used for the eye-tracker to reduce infrared interference from sunlight. Locations on the pavement were designated to indicate start and endpoints for the pedestrian to walk, as seen in Figure 9 (a).

Figure 9. Location of the pilot experiment. Figure (a) shows the start and endpoint for the pedestrian to walk toward the road as if they were to cross. The distance between these points is 5 m. Figure (b) illustrates the pedestrian's Field of View (FOV) when he is about to start walking.

Twelve different trials were performed, each with a specific action and gaze targets, as seen in Table 2. Each trial was a combination of one of three actions (standing, casual walking, or fast walking) and one of four types of gaze targets (stationary objects such as trees, buildings, and houses, any parked cars on the left and right, closest moving cars on left and right, or their mobile phone; Figures 10-12). The pedestrian actions were chosen in order to test the robustness of the moving vehicle classification module as it relies on camera pose estimation. The gaze targets are chosen to test the robustness of the alertness module of the system. In the standing trials, the pedestrian stood at the edge of the pavement, whereas in the walking trials, he walked either at a casual pace (~1.1 m/s) or a fast pace (~1.5 m/s) from the start to the endpoint on the pavement.

Table 2. List of trials

Trial Number	Pedestrian's action	Gaze target(s)	Abbreviation	Duration (seconds)
1	Standing	Stationary Objects	S-SO	9
2		Parked Cars	S-PC	
3		Moving Cars	S-MC	
4		Mobile Phone	S-MP	
5	Casual Walking	Stationary Objects	C-SO	7
6		Parked Cars	C-PC	
7		Moving Cars	C-MC	
8		Mobile Phone	C-MP	
9	Fast Walking	Stationary Objects	F-SO	5
10		Parked Cars	F-PC	
11		Moving Cars	F-MC	
12		Mobile Phone	F-MP	

(a) (b) (c)

Figure 10. Stationary objects trials. Figure (a) shows the first stationary object (a tree). Figure (b) shows the second stationary object (a wall). Figure (c) shows the third stationary object (a house in the background). These objects are chosen such that the pedestrian looks in a left-right-left manner.

(a) (b)

Figure 11. Parked cars trials. Figure (a) depicts the car parked on the left side of the street. Figure (b) shows the car parked on the right side of the street. The pedestrian looks at these cars in a left-right-left manner.

(a) (b)

Figure 12. Moving car and mobile phone trials. Figure (a) illustrates the moving car trials where pedestrians look at the cars moving on the street. The red bounding box on the car shows that the car is moving. Figure(b) shows the mobile phone trials where the pedestrian constantly looks at his mobile phone. The red border on the image signifies an alert to the pedestrian about the approaching car.

Each trial's duration was bounded by an audio file of a computer-generated voice saying "Start" and "End of trial", which also served as instructions for the pedestrian to start walking (if applicable) and gazing. The second experimenter played the audio instruction to start the trial when a moving car approached the pedestrian. The duration of each trial varied between five and nine seconds based on its type. For safety reasons, the pedestrian did not step onto the street in any trial but only walked till / stood at the edge of the pavement.

Before starting the trials, the pedestrian wore the Tobii Pro Glasses 2 and fastened the Bluetooth speaker to his trousers. The second experimenter booted up the system, entered the relevant pedestrian details described in the initialisation module section, and oversaw the pedestrian's calibration of the Tobii Pro Glasses 2. The pedestrian stood at the start point and was ready to perform the trials in a random order dictated by the experimenter. Apart from the mobile phone trials (in which the pedestrian was not to rotate his head), objects in all other trials were to be looked at by the pedestrian in a left-right-left manner by turning his head while walking/standing. The head movement was done to mimic the typical scanning behaviour of pedestrians before they cross a street (Trefzegar et al. (2018). At the end of each trial, the participant had to look straight ahead at the other side of the road. When moving cars were in the vicinity, the second experimenter played the correct audio file (depending on the trial), and the pedestrian performed the required actions and gaze behaviours. The experimenter also ensured that there was a moving car in every trial. Tobii recorded the pedestrian's view of the scene and his gaze data during the trials. The system streamed these to a laptop, detected oncoming cars, warned the pedestrian whenever appropriate, and saved outputs for further analysis.

3.2 Data analysis

The data from the pilot study were analysed to evaluate and understand the system's behaviour. The vehicle of interest (VOI) is the closest moving vehicle (if present) or the closest vehicle in the frame. The ground truth and detections were always evaluated for the vehicle of interest. The output video from the system, vehicle detections, pedestrian gaze points, moving statuses of vehicles, and the pedestrian's awareness (gaze marker falls within the bounding box of the VOI) for each frame were analysed using Python scripts. The output video was converted into a set of individual frames for ground truth annotation.

Definition of ground truth. Ground truth was defined in a three-step visual inspection of each video frame. First, it was inspected if any vehicles were present (Yes = 1) or absent (No = 0), independently of whether there were non-moving or moving. Second, for the frames with vehicles present, it was inspected whether VOI was moving (Yes = 1) or not (No = 0). Third, for the frames with moving VOI, it was inspected whether the pedestrian looked at the VOI (Yes = 1) or not (No = 0). The ground truth for the alert is assigned by checking each frame in the video feed and identifying situations wherein the pedestrian is at the edge of the pavement with an approaching car in his vicinity.

Correctness of system detection. Detection output from the system was evaluated for each video frame. First, vehicle detection output was checked if VOI was detected (Yes = 1) or not (No = 0). Second, for the detected VOI, it was checked for its classification, moving (Yes = 1) or non-moving (No = 0). Third, if the VOI is moving, it was checked if an alert is given (Yes = 1) or not (No = 0) based on the pedestrian's awareness of the VOI. If there were no vehicles present in the frame, a redundant value of '-1' is assigned for the second and the third.

3.2.1 Accuracy calculation

Accuracy is defined as the ratio of correct predictions to the total number of cases examined. It is computed using:

Accuracy = (True Positives + True Negatives) / (Positives + Negatives)

A true positive is when the ground truth and detection values are '1'. A true negative is when the ground truth and detection value are '0'. A false positive is when the ground truth is '0', and the detection value is '1'. A false negative is when the ground truth is '1' and the detection value is '0'.

The metrics mentioned above are calculated for every frame of the trials. Accuracies are calculated for (1) detection of a vehicle in the frame, (2) identification of the moving vehicle, and (3) alerts issued to the pedestrian for each trial. The accuracies of each trial are averaged to obtain the mean vehicle detection accuracy and mean moving vehicle identification accuracy for different pedestrian actions and gaze targets.

3.2.2 Estimating bounding box area and warning instant

Vehicle detection by YOLOv4 returns the object label, confidence, and bounding box coordinates. The bounding box coordinates contain the centre point (x, y), width, and height of the bounding box. The bounding box area is given by the product of its width and height. The bounding box area is plotted against trial time to analyse the vehicle's behaviour as it moves through the pedestrian's field of view. The timestamps with an alert ground truth of '1' are plotted by a series of parallel lines in orange (looks like a section), and the orange section is labelled as an alert zone. Black lines are plotted at an interval of 300 ms in the alert zone. The blue dashed line signifies the instant when an alert is given to the pedestrian. The green dash-dot line signifies the instant when the pedestrian reaches the pavement.

3.3 Results

3.3.1 Vehicle detection accuracy

Figure 13 depicts vehicle detection accuracy for the twelve trials in the study. Accuracies varied between 84% and 98%. It can be seen that the trials in which the pedestrian looked at his mobile phone had the highest vehicle detection accuracy. On the flip side, the trials that involved looking at moving cars had the lowest vehicle detection accuracy for standing and casual walking trials but had the second-highest accuracy in the fast walking trial. Furthermore, detection accuracy in the trials involving looking at stationary objects and parked cars were between the mobile phone and moving car trials.

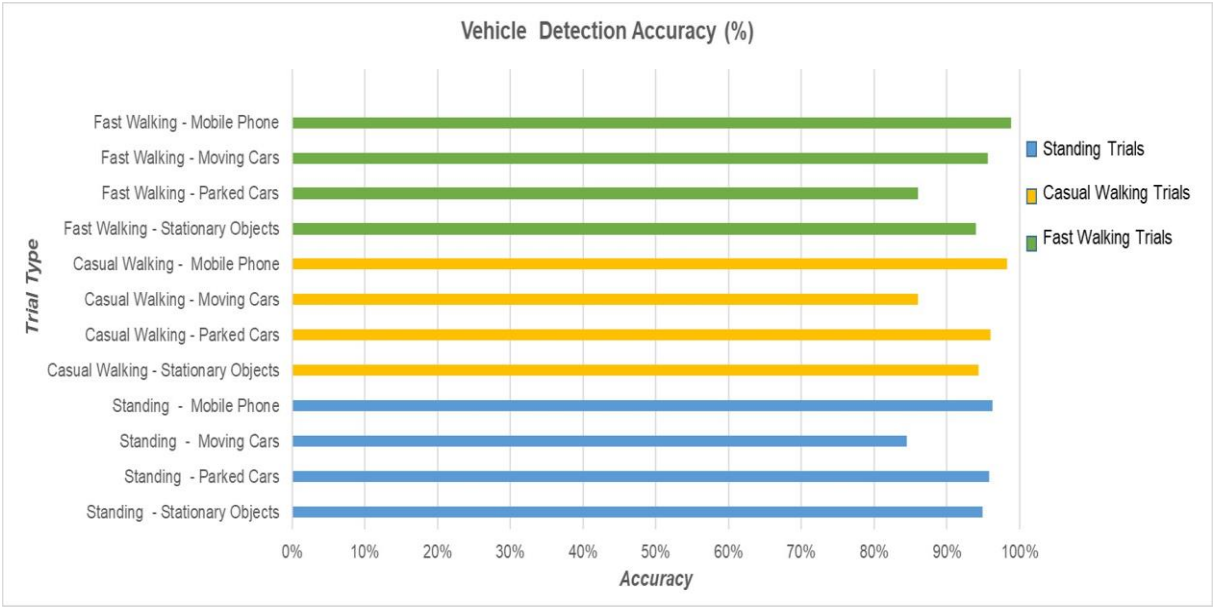


Figure 13. Vehicle detection accuracy per trial type.

Table 3. Mean vehicle detection accuracy for different gaze targets

No	Gaze target(s)	Mean vehicle detection accuracy (3 trials per gaze target) (%)
1	Stationary Objects	94.41
2	Parked Cars	92.62
3	Moving Cars	88.72
4	Mobile Phone	97.78

Table 4. Mean vehicle detection accuracy for different pedestrian actions

No	Pedestrian's action	Mean vehicle detection accuracy (4 trials per pedestrian's action) (%)
1	Standing	92.85
2	Casual Walking	93.67
3	Fast Walking	93.63

Tables 3 and 4 depict the mean vehicle detection accuracies for the different gaze targets and pedestrian actions, respectively. The mean vehicle detection accuracies were similar for the different pedestrian's actions.

3.3.2 Moving vehicle classification accuracy

Figure 14 shows the moving vehicle classification accuracy. The accuracies ranged between 32% and 88%. Similar to Fig.13, mobile phone trials had the highest accuracy among the trials. The stationary object trial with casual walking had the lowest moving vehicle classification accuracy among the trials. Overall, all the accuracies were lower than that of Fig.13.

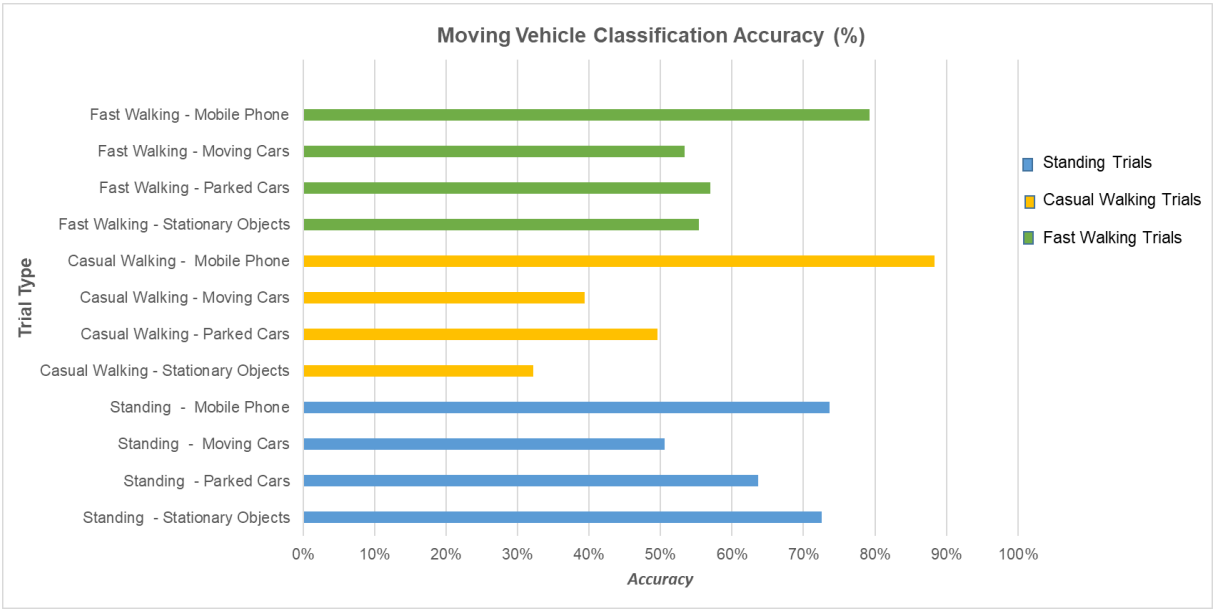


Figure 14. Moving vehicle classification accuracy vs trial type.

Table 5. Mean moving vehicle identification accuracy for different trial types

No	Gaze target(s)	Mean moving vehicle identification accuracy (3 trials per gaze target) (%)
1	Stationary Objects	53.36
2	Parked Cars	56.73
3	Moving Cars	47.77
4	Mobile Phone	80.41

Table 6. Mean moving vehicle identification accuracy for different walking speeds

No	Pedestrian's action	Mean moving vehicle identification accuracy (4 trials per pedestrian's action) (%)
1	Standing	65.08
2	Casual Walking	52.36
3	Fast Walking	61.25

Tables 5 and 6 show the mean moving vehicle identification accuracies for different gaze targets and pedestrian actions. Similar to Table 3, moving cars had the lowest accuracy among all the gaze targets.

3.3.3 Alert accuracy

From Table 7, the system's accuracy in alerting an inattentive pedestrian of a possible collision is 66.7%. In other words, for two out of every three possible collisions, the system performs its function. Despite the alerts given, 25 % of the alerts were provided with an average delay of 2 ms.

Table 7. Alert ground truth and alert given for each trial

Trial number	Trial type	Alert ground truth (1 - Alert needed; 0 - No alert needed)	Alert (1 - Alert given; 0 - No alert)
1	S - SO	1	1
2	S - PC	0	0
3	S - MC	1	0
4	S - MP	1	1
5	C - SO	1	1
6	C - PC	1	0
7	C - MC	0	0
8	C - MP	1	0
9	F - SO	1	1
10	F - PC	1	0
11	F - MC	1	1
12	F - MP	1	1
Accuracy		***66.67 %***	

3.3.4 Bounding box area vs. alert

Figure 15 shows the bounding box area of the VOI plotted against trial time for the stationary objects trial. The constant areas between timestamps zero and one, three and four in Figure 15 (b) illustrate the parked car, whereas the increase in area between timestamps five and six signifies that of the moving car. In Figure 15 (a), the dashed blue line is behind the alert zone (orange section), signifying a delayed alert to the pedestrian. The sharp rise in the area in Figure 15 (a) shows that the car entered the field of view as the pedestrian rotated his head. From Figure 15 (c), the alert zone for a shorter duration is depicted with an alert being triggered at the end as the pedestrian did not notice the oncoming vehicle in the farthest lane. The change in the area at the peaks between timestamps one and three in Figure 15 (c) depicts the head rotation of the pedestrian as he walks toward the pavement.

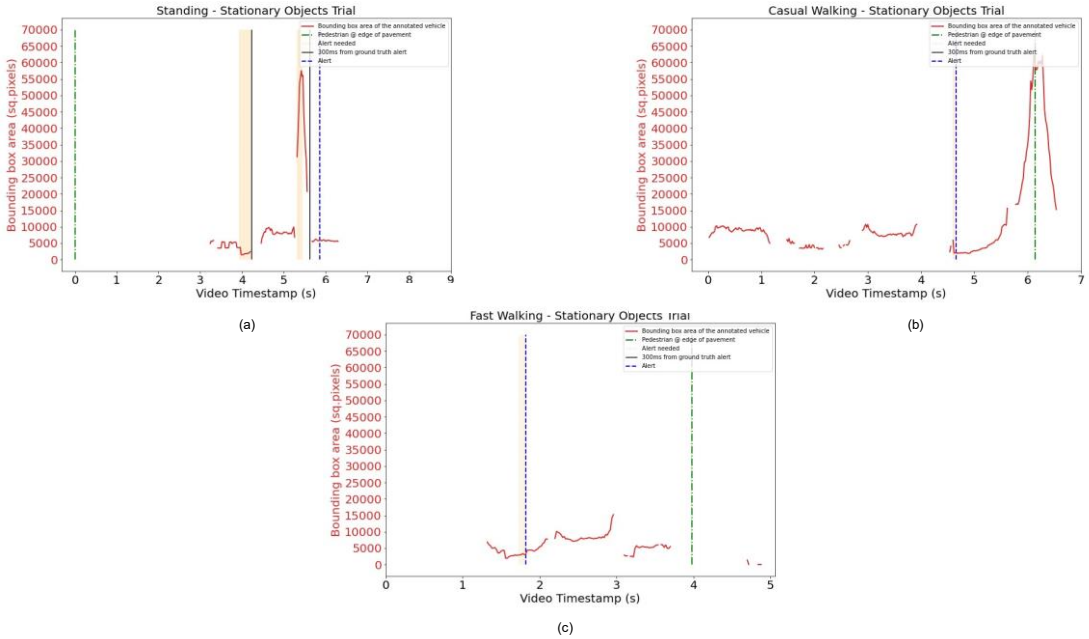


Figure 15. The bounding box areas of VOI against the trial time for different pedestrian actions (walking, casual walking, and fast walking, respectively). The orange section is the alert zone within which an alert is given to the pedestrian as they had not looked at the car (from ground truth). The black line marks the instant, which is 300 ms from the start of the alert zone. The blue dashed and green dash-dot line signifies the instant of an alert and the pedestrian reaching the pavement.

Figure 16 illustrates the bounding box area of VOI plotted against time for different walking speeds during the parked car trials. Figure 16 (a) shows the absence of moving cars in the trial, which resulted in no alerts and alert zones (orange sections). Figure 16 (b) shows that there was no alert, although the pedestrian reached the pavement (green dash-dot line) and needed an alert (orange section) as he had not looked at the car. The value of the bounding box area in Figure 16 (b) depicts that the car was closer to the pedestrian. Figure 16 (c) illustrates the same situation as Figure 16 (b), but the bounding box area of the vehicle in the alert zone is less when compared to that of Figure 16 (b), illustrating that the car was in the farthest lane to the pedestrian.

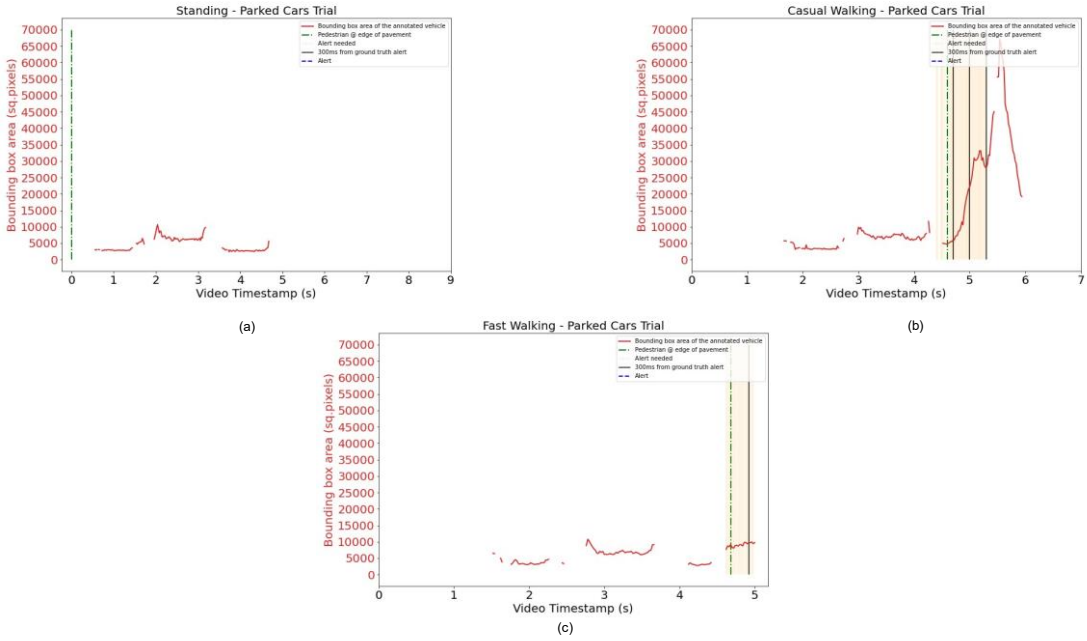


Figure 16. The plots depict the bounding box areas of VOI against time during the parked car trials for different pedestrian actions. The plots have no blue dashed lines signifying that the system did not provide alerts to the pedestrian during these trials.

Figure 17 depicts the bounding box area of VOI plotted against time for moving car trials with different pedestrian actions. The change in area in Figure 17 (a) between timestamps eight and nine illustrates the car's gradual movement from left to right within the field of view of the pedestrian. The pedestrian did not notice the car as seen from the alert zone depicted. Figure 17 (b) shows no alert zones (orange section) as the pedestrian looks at the moving cars during the trials, and the system did not trigger any alerts. In the same figure, the area reduces to zero when the VOI is not detected. Figure 17 (c) shows an alert as the pedestrian failed to look at the moving car, which is illustrated by the blue dashed line and orange section.

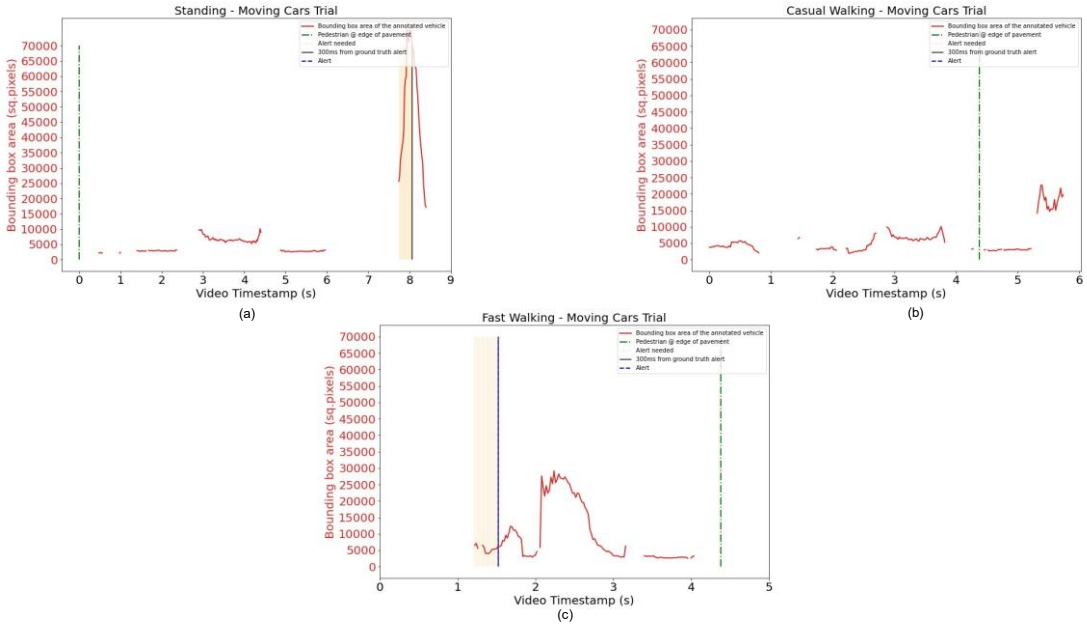


Figure 17. The figure illustrates the bounding box area of the VOI against time for moving car trials with different pedestrian actions. The pedestrian looks at the moving cars during the trials, as seen from the plot (b), which has no alert zones (orange sections) over the trial.

Figure 18 illustrates the bounding box area of the VOI for the mobile phone trials with different walking speeds. From Figures 18 (a) and (c), the system has alerted the pedestrian when they walk while using their mobile phone within the alert zone, and the alerts were delayed. The failure to alert the pedestrian even though with only one vehicle in the trial is depicted in Figure 18 (b).

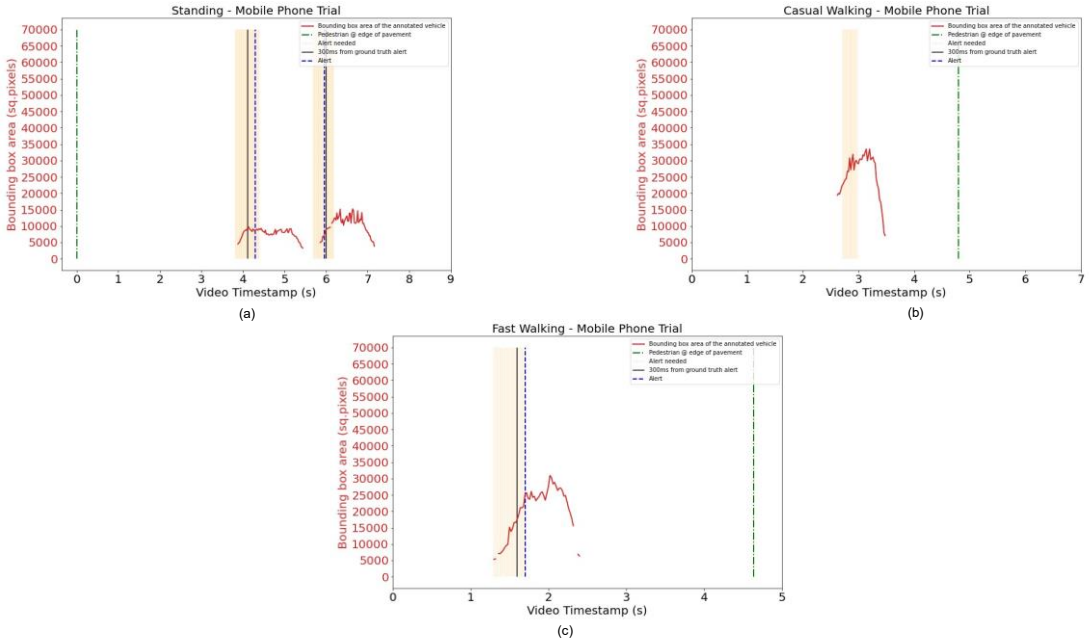


Figure 18. The plot illustrates the bounding box areas of VOI for the mobile phone trials with different walking speeds. The plots have only one or two vehicles moving within the trial duration, and it depicts the alerts provided for the respective vehicles.

In Figure 18, the bounding box area of VOI has a similar pattern as the car moves through the frame. The bounding box area gradually increases and peaks when the car is perpendicular to the pedestrian and then gradually decreases as the car moves out of the FOV of the pedestrian. The pattern in the bounding box area of the VOI illustrates that the pedestrian had not gazed at his surroundings and was constantly looking straight ahead (see Fig.19).

Figure 19. Illustrates the casual walking mobile phone trials. It is seen that the pedestrian looks straight ahead with his gaze on the mobile phone without heeding the vehicles in the environment.

4 Discussion

4.1 Main findings

The study aimed to develop a warning system that monitors the visual attention of pedestrians constantly and alerts them when they do not notice an approaching vehicle prior to crossing the road. We use the monocular scene camera and infrared (IR) eye cameras in the head-mounted eye tracker to estimate the closest moving vehicle to the pedestrian and monitor their awareness of that vehicle, respectively. The algorithm in the warning system uses YOLOv4, monodepth2 and camera pose to identify moving vehicles in the frame in real-time. The pedestrian is considered to be aware of the approaching vehicle when their gaze fixation lasts for at least 300 ms in the area within the bounding box of that vehicle. The system triggers an auditory alert to the pedestrian when they have not noticed the closest moving vehicle.

The novelty of our study is that we used a portable head-mounted eye-tracker to monitor the pedestrian's gaze and alert the distracted pedestrian in real-time while on the road. We used a combination of computer vision algorithms to replicate the effect of the stereo camera from the input of the monocular camera in the eye-tracker and detect moving cars toward the pedestrian. Compared to the existing pedestrian-based warning systems that detect vehicles in the vicinity of the pedestrian, our system monitors the awareness of the pedestrian on the approaching vehicle in real-time (25 ms latency) through their gaze behaviour and warns them if they do not notice it. Our system incorporates multi-threading and multi-processing techniques in Python to execute the different modules in coherence and achieve real-time inference speeds with a low latency of 25 ms compared to 55 ms in PAWS (De Godoy et al., 2018). In addition, our system works at a real-time inference speed of 50 FPS. Our system may find use in the upcoming smart-glasses and V2P communication systems in the future.

Our system was demonstrated in real-time in an outdoor environment through a pilot. We found that the vehicle detection accuracy was high in trials where the pedestrian looked at his mobile phone. A possible explanation for this result is that the pedestrian did not rotate his head to look around. Consequently, the video image was clear and had well-defined objects, which resulted in the highest vehicle detection accuracy. The accuracy was the lowest in the trials in which the participant was asked to look at the moving cars. A possible explanation for this result is that the parked cars in the street occluded the moving cars for a part of the trial, thereby hindering their detection by the algorithm.

The results also showed that the accuracy was lower for identifying moving vehicles compared to the detection of any vehicle. Camera shake due to the head movements produces blurry frames and affects feature matching, resulting in computational errors in estimating camera pose and affecting the identification of the moving vehicles. Similar to the results regarding vehicle detection, the mobile phone trials resulted in the highest moving vehicle identification accuracy because of limited camera motion. Likewise, standing trials had the highest mean moving vehicle identification accuracy among gaze targets.

The values of the bounding box area of vehicles in the results illustrate the type and location of the vehicle on the road. The parked vehicles had constant areas over time, whereas the area of moving vehicles increased and decreased gradually over time. The vehicle in the closest lane to the pedestrian had the highest bounding box area as it occupied the majority of the frame and the vehicle in the farthest lane had the bounding box area lower than that of the vehicle in the closest lane. The sudden increase in area signifies that the vehicle entered the camera's field of view as the pedestrian rotated his head, and the decrease in the area to zero signifies a missing detection of the vehicle in the frame. The bounding box area serves as a measure to identify the vehicle's behaviour in the video frame.

Our system's working was inferred from the plots containing the bounding box area of vehicles plotted over time. We found that the system triggered the alert to the pedestrian about the closest moving vehicle as seen from the bounding box area. Likewise, the system did not trigger alerts to the pedestrian when there were only parked vehicles in the video frame. The alerts sent to the pedestrian relied on the moving vehicle classification and the pedestrian's gaze in each frame. The mobile phone trials had the highest moving vehicle classification accuracy, resulting in triggering alerts within the alert zone (no delay). Likewise, the absence and delay of alerts to the pedestrian could be reasoned out due to the misclassification of the moving vehicles even though the pedestrian did not notice the moving vehicle.

The effect of the limited field of view of the camera in our system on triggering alerts could be inferred from the bounding box area plots. The system encounters a delay in alerting the pedestrian when a moving vehicle suddenly enters the pedestrian's field of view as he rotates his head in a left/right/left manner, and this could give warnings for the same car twice whenever it enters the field of view of the pedestrian as the vehicle is not tracked. Likewise, the system could detect only the vehicles within the frame when the pedestrian had not rotated his head, limiting the system to detect and trigger alerts to the pedestrian for the vehicles approaching on either side of the road.

Compared to the existing pedestrian warning systems in the literature (Wang et al. (2012); Li et al. (2018); De Godoy et al. (2018)), our system works at a higher inference speed of 50 FPS, despite the use of deep learning networks in real-time. Our system could identify vehicles with an accuracy of 93% and classify them as moving/non-moving with an accuracy of 60% compared to WalkSafe (Wang et al. 2012), which identified vehicles with an accuracy of 77%. Although SafeWalking (Li et al. 2018) had an accuracy of 91%, it only warned pedestrians to look at the road when they were constantly looking at their phones. On the other hand, our system warns the distracted pedestrian about the approaching car when they do not look at it with an accuracy of 67%. Our system improves the existing vision-based systems in the literature and presents a more usable system in realistic outdoor environments.

4.2 Limitations

Although the system works effectively in real-time, there are some limitations. First, the system is an experimental prototype that runs on the computer's graphic processor with the eye-tracker connected via Wi-Fi. It increases the system's dependency on a Wi-Fi range for fast data transmission. The latency increases above a distance of 10m between the computer and the recording unit of the eye-tracker. Moreover, because the recording unit was fastened to the front pockets of the trousers, the Wi-Fi signal might have been blocked by the human body (Anaya et al., 2014).

Second, the video output from the monocular scene camera at 50Hz had a lower resolution (960 x 540 px) compared to today's standards, and the output frames became blurred when subjected to a faster head movement. The camera suffered from changes in light exposure during the day and different weather conditions. Also, the diagonal Field of View of the scene camera was 90º, limiting the visual field where the algorithm could search for approaching vehicles towards the pedestrian. The camera hardware could be improved in future for better processing capabilities.

Third, the algorithm uses monocular depth networks to compute stereo depth on the input frame from the monocular scene camera, but these are not as accurate as the depth from a stereo camera. The algorithm may face computational errors in feature matching and camera pose calculation depending on the image frame quality received from the monocular scene camera. In addition, the algorithm entirely relies on the vehicle detections to check for pedestrians' awareness of that vehicle and at all times needs two consecutive frames with detections to compute the moving vehicles in the frame.

Finally, the evaluation of the system was conducted with one participant, and further testing with more participants and different scenarios would be needed to establish the performance of the warning system. Although the pilot experiment had controlled variables, the moving cars on the street were not controlled, resulting in cars moving at different speeds in the trials. The moving cars could also be controlled to better correlate the bounding box area of vehicles over time among the trials.

4.3 Conclusion

The study developed a novel pedestrian-based warning system using eye-tracking technology that works in real-time and warns the distracted pedestrian of a possible collision with an approaching vehicle. The system uses a wearable eye-tracker to monitor the environment and visual attention of the pedestrian to provide alerts in case of a probable collision. The algorithm in use mimics the effect of a stereo camera from the input of the monocular scene camera in the eye-tracker to identify the moving vehicles on the road. The system considers the pedestrian to be aware of the approaching vehicle when they gaze at the approaching vehicle continuously for at least 300 ms (gaze fixation duration). Results from the pilot study suggest that the system alerted the inattentive pedestrian with an accuracy of 66.7% at an inference speed of 50 FPS (real-time).

As pointed out above, the system has the highest vehicle detection and moving vehicle identification accuracy in the mobile phone trials because of the limited head movement. The vehicle detection accuracy and moving vehicle identification accuracy reduce due to the camera's motion and head movement of the pedestrian. The bounding box area over time plots depicts the system's working with respect to the movement and position of the vehicle at any instant in the video frame. Extracting bounding box areas to detect the vehicle's moving status from a monocular camera could be explored in the future.

The system has potential for further research and applications as smart glasses and affordable eye-trackers are on the horizon. The topic could be of interest as it encapsulates a warning system based on visual attention that could be applied to pedestrians and drivers to improve road safety. It also stimulates further research of behavioural studies to analyse and understand the pedestrian's behaviour when subjected to an alert in a naturalistic scenario. The system could be further amplified with better hardware, user interface, and portability. The system could serve in smart glasses among various functionalities, enabling its user to be vigilant. For the time being, the developed system could serve as a base to build upon and realise the use of warning systems for pedestrians to mitigate their vulnerability.

5 Supplementary materials

Supplementary material including pilot data, analysis scripts, and plots is available at https://www.dropbox.com/sh/8jv4jzo3qwebr2o/AAAL3R_SjI-ywIfr9QdnCaL7a and maintained version of the code is available at https://github.com/lokkeshvertk/pedestrian-gaze.

6 References

Anaya, J.J., Merdrignac, P., Shagdar, O., Nashashibi, F., Naranjo, J.E. (2014). Vehicle to Pedestrian Communications for Protection of Vulnerable Road Users. *IEEE Intelligent Vehicles Symposium,* 1-6. Michigan, United States. https://hal.archives-ouvertes.fr/hal-00992759

de Godoy, D., Islam, B., Xia, S., Islam, M.T., Chandrasekaran, R., Chen, Y.C., Nirjion, S., Kinget, P.R., Jiang, X. (2018). PAWS: A Wearable Acoustic System for Pedestrian Safety. *IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation*, 237-248. https://doi.org/10.1109/IoTDI.2018.00031

de Tommaso, D., & Wykowska, A. (2019). TobiiGlassesPySuite. *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. https://doi.org/10.1145/3314111.3319828

de Winter, J.C.F., Bazilinskyy, P., Wesdorp, D., De Vlam, V., Hopmans, B., Visscher, J., Dodou, D. (2021). How do pedestrians distribute their visual attention when walking through a parking garage? An eye-tracking study. *Ergonomics.* https://doi.org/10.1080/00140139.2020.1862310

de Winter, J. C. F., Eisma, Y. B., Cabrall, C. D. D., Hancock, P. A., & Stanton, N. A. (2018). Situation awareness based on eye movements in relation to the task environment. Cognition, Technology & Work, 21(1), 99–111. https://doi.org/10.1007/s10111-018-0527-6

Dey, D., Walker, F., Martens, M., Terken, J. (2019). Gaze Patterns in Pedestrian Interaction with Vehicles: Towards Effective Design of External Human-Machine Interfaces for Automated Vehicles. *AutomotiveUI '19: Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications,* 369–378. https://doi.org/10.1145/3342197.3344523

Engel, S., Kratzsch, C., David, K. (2013). Car2Pedestrian-Communication: Protection of Vulnerable Road Users Using Smartphones. *In: Fischer-Wolfarth J., Meyer G. (eds) Advanced Microsystems for Automotive Applications 2013. Lecture Notes in Mobility. Springer,* Heidelberg, Germany. https://doi.org/10.1007/978-3-319-00476-1_4

European Commission. (2021). Pedestrians and Cyclists. *Mobility and Transport - European Commission*. https://ec.europa.eu/transport/road_safety/specialist/knowledge/pedestrians_en

Galley, N., & Biniossek, D. (2015). Fixation durations - why are they so highly variable. https://doi.org/10.13140/RG.2.1.3128.1769

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2012.6248074

Godard, C., Mac Aodha, O., Firman, M., & Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 3828-3838). https://doi.org/10.48550/arXiv.1806.01260

Hasan, R., Hoque, M. A., Karim, Y., Griffin, R., Schwebel, D., & Hasan, R. (2021). StreetBit: A Bluetooth Beacon-based Personal Safety Application for Distracted Pedestrians. *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. https://doi.org/10.1109/ccnc49032.2021.9369650

Hasan, R., & Hasan, R. (2022). Pedestrian safety using the Internet of Things and sensors: Issues, challenges, and open problems. *Future Generation Computer Systems*, 134, 187–203. https://doi.org/10.1016/j.future.2022.03.036

Hwang, T., Jeong, J. P., & Lee, E. (2014). SANA: Safety-Aware Navigation App for pedestrian protection in vehicular networks. *2014 International Conference on Information and Communication Technology Convergence (ICTC)*. https://doi.org/10.1109/ictc.2014.6983341

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A survey of deep learning-based object detection. *IEEE access*, *7*, 128837-128868. https://doi.org/10.48550/arXiv.1907.09408

Kim, J., Sung, J.-Y., & Park, S. (2020). Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition. *2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, 1–4. https://doi.org/10.1109/ICCE-Asia49877.2020.9277040

Lévêque, L., Ranchet, M., Deniel, J., Bornard, J.-C., Bellet, T. (2020). Where Do Pedestrians Look When Crossing? A State of the Art of the Eye-Tracking Studies, *IEEE Access*, vol. 8, 164833-164843. https://doi.org/10.1109/ACCESS.2020.3021208

Lee, C. J., Tseng, Y. H., & Chang, P. C. (2018, September). Audio-based early warning system of vehicle approaching event for improving pedestrian's safety. In 2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin) (pp. 1-3). IEEE. https://doi.org/10.1109/ICCE-Berlin.2018.8576220

Li, Y., Xue, F., Fan, X., Qu, Z., Zhou, G. (2018). Pedestrian walking safety system based on smartphone built-in sensors. *The Institution of Engineering and Technology*, Vol.12, 751-758. https://doi.org/10.1049/iet-com.2017.0502

Lin, C. H., Chen, Y. T., Chen, J. J., Shih, W. C., & Chen, W. T. (2016). pSafety: A Collision Prevention System for Pedestrians Using Smartphone. *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. https://doi.org/10.1109/vtcfall.2016.7881183

Mwakalonge, J., Siuhi, S., White, J. (2015). Distracted walking: Examining the extent to pedestrian safety problems. *Journal of Traffic and Transportation Engineering*, 2, 327-337. https://doi.org/10.1016/j.jtte.2015.08.004

National Highway Traffic Safety Administration (NHSTA). (2021, February 5). Data Visualisation - Fatality Analysis Reporting System (FARS): Pedestrians. https://explore.dot.gov/views/DV_FARS_PD/Home?%3Aiid=1&%3AisGuestRedirectFromVizportal=y&%3Aembed=y

Negi, S., & Mitra, R. (2020). Fixation duration and the learning process: an eye tracking study with subtitled videos. *Journal of Eye Movement Research*, *13*(6). https://doi.org/10.16910/jemr.13.6.1

Ng, T. (2011). *FFMPEG - av_interleaved_write_frame return error code -22 when passing in a H.264 frame*. Thompson's Technological Insight. http://thompsonng.blogspot.com/2011/09/ffmpeg-avinterleavedwriteframe-return.html

OpenCV: Feature Matching. (n.d.). OpenCV. https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html#:%7E:text=Brute%2DForce%20matcher%20is%20simple,the%20BFMatcher%20object%20using%20cv.

Otte, D., Jänsch, M., Haasper, C. (2012). Injury protection and accident causation parameters for vulnerable road users based on German In-Depth Accident Study GIDAS. *Accident Analysis and Prevention 44*, 149–153. https://doi.org/10.1016/j.aap.2010.12.006

Qiao, D., & Zulkernine, F. (2020). Vision-based Vehicle Detection and Distance Estimation. *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. Published. https://doi.org/10.1109/ssci47803.2020.9308364

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G.R. (2011). ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision*, 2564-2571. https://doi.org/10.1109/ICCV.2011.6126544

Sewalkar, P., & Seitz, J. (2019). Vehicle-to-Pedestrian Communication for Vulnerable Road Users: Survey, Design Considerations, and Challenges. *Sensors* (Basel, Switzerland), *19*(2), 358. https://doi.org/10.3390/s19020358

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications (Texts in Computer Science)* (2011th ed.). Springer. (pp.29-52). http://szeliski.org/Book/

Thirunavukkarasu Kumaaravelu, L. *YOLOv4 trained on KITTI Dataset* [Computer software]. https://github.com/lokkeshvertk/darknet

Thompson, L.L., Rivara, F.P., Ayyagari, R.C., Ebel, B.E. (2012). Impact of social and technological distraction on pedestrian crossing behaviour: an observational study. *Injury Prevention*, vol.19, 232-237. http://dx.doi.org/10.1136/injuryprev-2012-040601

Tobii Pro AB. (2015). *Free application programming interface |Tobii Pro Glasses 2 API*. Tobii Pro. https://www.tobiipro.com/product-listing/tobii-pro-glasses-2-sdk/

Tobii Pro AB. (2018). *Tobii Pro Glasses 2 API Developer's Guide* (V1.3 ed.). Tobii AB.https://www.tobiipro.com/pop-ups/glasses-2-api-form/?v=1.3

Trefzger, M., Blascheck, T., Raschke, M., Hausmann, S., Schlegel, T. (2018). A Visual Comparison of Gaze Behavior from Pedestrians and Cyclists. *In Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications (ETRA' 18),* Article 104, 1–5*.* https://doi.org/10.1145/3204493.3204553

Yuste, L. B., Boronat, F., Montagud, M., & Melvin, H. (2015). Understanding Timelines Within MPEG Standards. *IEEE Journals & Magazine | IEEE Xplore*. https://ieeexplore.ieee.org/document/7293587/

Wang, T., Cardone, G., Corradi, A., Torresani, L., Campbell, A.T. (2012). WalkSafe: a pedestrian safety app for mobile phone users who walk and talk while crossing roads. *In Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications (HotMobile' 12)*, Article 5, 1–6. https://doi.org/10.1145/2162081.2162089

Wang, H., Yu, Y., Cai, Y., Chen, X., Chen, L., Liu, Q. (2019). A Comparative Study of State-of-the-Art Deep Learning Algorithms for Vehicle Detection. *IEEE Intelligent Transportation Systems Magazine*, vol. 11, 82-95. https://doi.org/10.1109/MITS.2019.2903518

Watanabe, Y., Shoji, Y. (2019). A Vehicle-Approach Alert System Based on the Neighbor Discovery Protocol for Pedestrian Safety, *2019 Global IoT Summit (GIoTS)*, 1-6. https://doi.org/10.1109/GIOTS.2019.8766405

Won, M., Shrestha, A., Park, K.-J., Eun, Y. (2020). SaferCross: Enhancing Pedestrian Safety Using Embedded Sensors of Smartphone, *IEEE Access*, vol. 8, 49657-49670. https://doi.org/10.1109/ACCESS.2020.2980085

World Health Organisation. (2013). Pedestrian safety: a road safety manual for decision-makers and practitioners. *World Health Organisation*. https://www.who.int/publications/i/item/pedestrian-safety-a-road-safety-manual-for-decision-makers-and-practitioners

World Health Organisation. (2020, February 7). Road traffic injuries. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

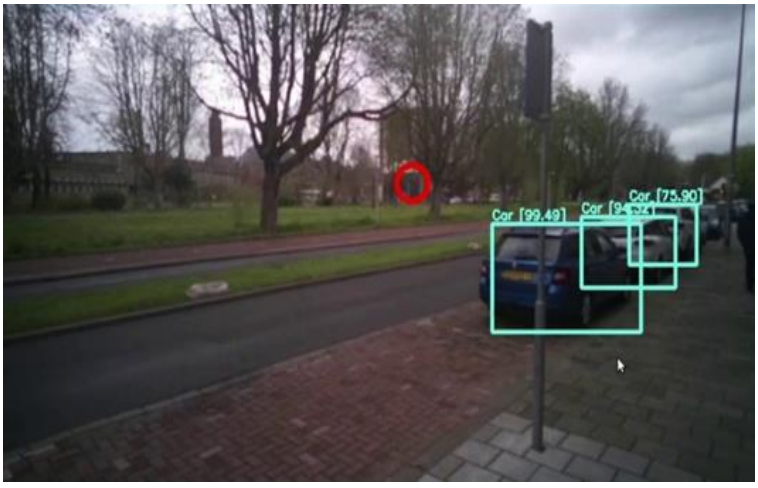
Wu, X., Miucic, R., Yang, S., Al-Stouhi, S., Misener, J., Bai, S., & Chan, W. (2014). Cars Talk to Phones: A DSRC Based Vehicle-Pedestrian Safety System. *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, 1-7. https://doi.org/10.1109/VTCFall.2014.6965898

Zhou, F., Yang, X. J., & de Winter, J. C. F. (2022). Using Eye-Tracking Data to Predict Situation Awareness in Real Time During Takeover Transitions in Conditionally Automated Driving. *IEEE Transactions on Intelligent Transportation Systems*, *23*(3), 2284–2295. https://doi.org/10.1109/tits.2021.3069776

7 APPENDIX

7.1 Ground truth annotation

Do I see the car?	1
Is the car moving?	1
Does the pedestrian need an alert?	1



Do I see the car?	1
Is the car moving?	0
Does the pedestrian need an alert?	0

Do I see the car?	1
Is the car moving?	1
Does the pedestrian need an alert?	1

Do I see the car?	1
Is the car moving?	1
Does the pedestrian need an alert?	1

Do I see the car?	1
Is the car moving?	1
Does the pedestrian need an alert?	0

Do I see the car?	0
Is the car moving?	0
Does the pedestrian need an alert?	0

Do I see the car?	1
Is the car moving?	1
Does the pedestrian need an alert?	0

Do I see the car?	1
Is the car moving?	1
Does the pedestrian need an alert?	0

Do I see the car?	1
Is the car moving?	0
Does the pedestrian need an alert?	0

7.2 Task instructions to pedestrian

TASK INSTRUCTION

Before the experiment

You will be asked to put on the eye-tracking glasses and fasten the recording unit and the Bluetooth speaker to your belt.

Check if the glasses are comfortable. If necessary, the experimenter can replace the nose pads. After this, the experimenter will ask you to look at a card to calibrate the glasses.

During the experiment

You will be asked to either stand or walk on the pavement a number of times, while following recorded instructions about where to look. Specifically, the following instructions will be given:

1. Stand on the marked spot on the curb
2. Walk casually on the pavement towards its edge
3. Walk fast on the pavement towards its edge

DO NOT STEP ONTO THE STREET.

At the beginning of each trial, you will hear instructions about the *type of objects* you should look at during the trial:

1. Look at stationary objects
2. Look at parked cars
3. Look at moving cars

4. Look at a mobile phone

During the trial, you will hear instructions about *the direction* where you should look at that specific moment:

1. Look left
2. Look right
3. Look straight

During the trials, you might hear a beeping sound; this is a warning that there is a car approaching that you have not noticed. When you hear this warning, abandon the gaze instruction you were following and check the road for approaching traffic.

After the experiment

You are free to go!!