**Jaywalking detection - autonomous vehicle:** a survey

**--Harsh Nishad**

**Abstract** Jaywalk is the most common urban traffic violation, more than 270 000 pedestrians lose their lives on the world’s roads each year accounting for 22% of the total 1.24 million road traffic deaths. However, there are cities that are heavily centered around road traffic and which do little to provide pedestrians with means for safe and frequent road crossing. So In this paper, we attempt to survey different proposed methods from the enormous amount of online information present to detect Jaywalking. We have analyzed the various methods depending on their ability to successfully detect (various benchmarks) whether the pedestrian is Jaywalking or not.

**Keywords** Jay walking detection· Online survey · Autonomous driving

**1 Introduction**

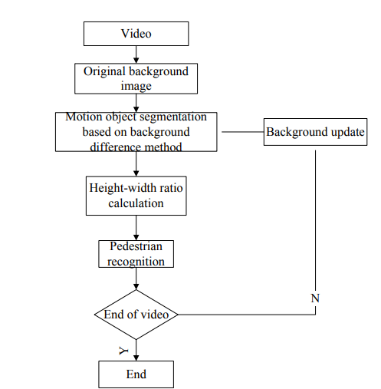
In India, jaywalking is not explicitly included in the law as an offense but is covered under the broader term 'obstruction of traffic' in state and metropolitan laws. Examples include section 28B of the Delhi Police Act, 33B of the Bombay Police Act, and 92G of the Karnataka Police Act. Many places in India lack pavements, and also pedestrian crossings and footpaths, ignorance of safety rules, and the poor regulation of related laws by authorities. Drives against jaywalking are conducted by the police departments from time to time and offenders are given fines of 100 to 500 Indian rupees, depending upon the jurisdiction. Drivers must yield the right of way for pedestrians at unsignalled crossings and marked pedestrian crossings.

**2. Motion path analysis**

**2.1 Methodology: Pedestrian recognition and tracking**

The motion objects which are extracted by the background difference method include pedestrian and vehicle, so the pedestrian recognition is necessary. By contrasting the pedestrian with the vehicle, it is obvious that the height-width ratio of the pedestrian is different from vehicle. The object’s height-width ratio ρ (ρ = H W/) is defined by the height-width ratio of the objects outside rectangular, and calculated every five frames.

Figure displays how height-width ratio will be calculated and recognition is being performed



**2.2 Pedestrian tracking**

**To obtain pedestrian motion path, it requires to track pedestrian which includes 3 steps**

**2.2.1 A predictive algorithm (linear)**

Predictive algorithm is used to predict the pedestrian position. According to the continuity of the pedestrian’s motion in time and space and the pedestrian’s low speed in the detection area, the linear predictive algorithm is suitable to predict the pedestrian position**.**

**2.2.2 Cost function and Object position**

The cost function is calculated for every tracked and candidate object, the less the value is the more the similar two objects are.

The object, which makes the cost function has the minimum value in all candidate objects, is considered as the same object, and will be recorded to update the pedestrian’s motion path.

**2.3 Results and setbacks**

The detection ratio is equal to the number of detected jaywalkers divided by the real number of jaywalkers in video. it is up to 90%

The main set back of this model is that if pedestrian’s clothes are similar to the color of road, the pedestrian segmentation is fragmented and the pedestrian recognition is unsuccessful.

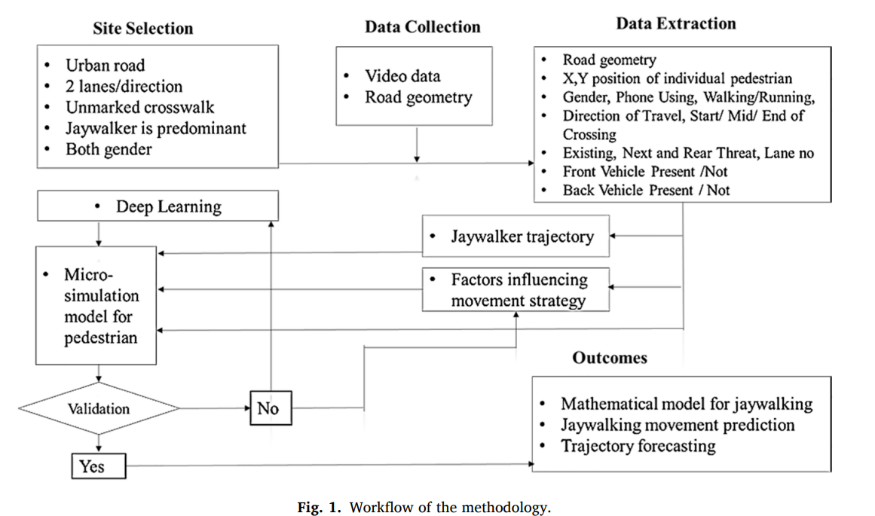
The pedestrian tracking is unsuccessful when the pedestrian suddenly speeds up from inactivity

**3 Jaywalking detection using artificial neural network**

This paper apart from detecting jaywalking also collects data related to jaywalkers such as gender, direction of crossing, walking or running, cell phone use, roadway lane number, etc. This includes a dataset comprises of 2504 samples which is collected under non-lane based heterogeneous traffic conditions.

**3.1 Methodology**

This model includes four sections: data collection, extraction and processing, analysis and prediction.



The video data is collected using a high-definition video recording device. To extract the coordinates of jaywalkers who might be jaywalking as well as other input variables from the video, Traffic Data Extractor software is used.

The packages that are used in Python are ‘NumPy’ for scientific computing, ‘pandas’ for data frame handling, ‘TensorFlow’, sci-kit learn and ‘karas’ for Artificial neural network algorithms.

**3.2 Results and setbacks**

One-way ANOVA analysis is conducted to examine whether there are significant differences in running, phone using and risk-taking behavior across male and female jaywalkers.

Phone using (pvalue: 0.185)

Running behavior (p-value: 0.417)

is found to be not statistically significant at a 95% confidence interval

Two-way ANOVA analysis to examine effects of jaywalking-related factors on crossing speed of jaywalkers.

Difference in speed in different lanes came out as statistically significant (p-value 0.026)

Interaction effects of the lane and walking/running variable on speed came out as not significant (p-value 0.237)

**4 Thermal camera-based Jaywalking estimation using deep learning**

A two-step hierarchical deep learning methodology using visible and thermal camera is planned to solve these challenges.

The two steps are:

**1**.Deep learning-based scene classification

**2**.Scene specific segmentation framework

The proposed framework is validated on the FLIR public dataset and compared with baseline algorithms

**4.1 Methodology: hierarchical framework**

This inlcudes of a classification step and a semantic segmentation step. The classification step is formulated using a single deep learning-based scene classifier which classifies the driving scene into a legal cross work.

The hierarchical framework is developed to reduce the following pedestrian behavior estimation errors:

**4.2 Algorithm**

**4.2.1 Classification step: Architecture**

The design comprises of feature extraction, classification and output layers. The two feature extraction branches contain five blocks. The first four blocks each contain a 2D convolutional layer with batch-normalization followed by a max-pooling layer. The fifth block contains a 2-dilated convolutional layer with batch-normalization.

**4.2.2 Semantic Segmentation Step: Architecture**

This contains two multi-class semantic segmentation networks. First is trained on the legal cross walk scenes, second is trained in illegal pedestrian crossing.

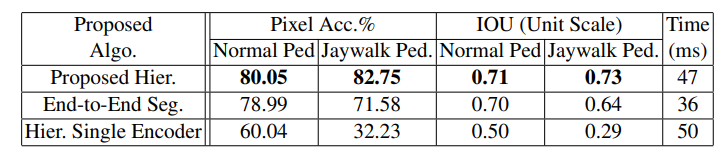
Deep learning-based encoder-decoder architecture is used for thermal fusion segmentation. encode layer is for feature extraction layer. decoder layer contains four transpose convolution layer and 5 convolutional layers.

The output of the last decoder layer is fed into the output layer which performs the multiclass segmentation

**4.3 Results and setbacks**

As mentioned in table below the accuracy of the proposed model is more as compared to the accuracy of the end -end segmentation or single encoder model.

This hierarchical model reduced the errors faced by the baseline model that is segmentation of individual having similarities between jaywalking pedestrians and pedestrians in legal crossing scenes.



**5.data generation and classification using convolution neural network**

The proposed system automatically generates a jaywalker based on the existing pedestrian objects in the image provided.

First train the existing network with the image dataset, second generator synthesizes jaywalker randomly, third the CNN classifier is used to detect the jaywalkers in the real time.

**5.1 Methodology: proposed system**

First, the drivable area is detected in the base road image by Mask RCNN, learned from the black box dataset. This detector is called a drivable area detector.

Second, Mask R-CNN pretrained with MS-COCO2014 detects and segments person objects.

Third, the segmented mask of an individual is copied, transformed, and pasted into the inner area of the mask.

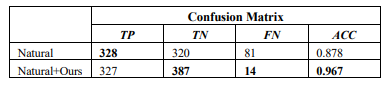
**5.1.1Dataset generation:**

From the Mask R-CNN trained with the black box dataset, we only segment the mask of drivable areas called D. Vehicles and peoples are excluded from D.

The location of the mask of the person is selected randomly.

**5.1.2 Jaywalker classifier**

This classifier contains the architecture structure trained on ImageNet data. And we retrained this classifier with a different top layer that can recognize jaywalk classes of images. Finally, this classifier determines whether each subject image contains a jaywalking event.



**5.2 Results and setbacks**

The accuracy of the trained model is higher as compared to the model that contains the natural dataset.

The accuracy of the model is calculated using the formula

𝐴𝐶𝐶(𝐴𝑐𝑐𝑢𝑟𝑎𝑐𝑦) = (𝑇𝑟𝑢𝑒 𝑃𝑜𝑠𝑖𝑡𝑖𝑣𝑒+𝑇𝑟𝑢𝑒 𝑁𝑒𝑔𝑎𝑡𝑖𝑣𝑒)/ 𝑃𝑜𝑠𝑖𝑡𝑖𝑣𝑒+𝑁𝑒𝑔𝑎𝑡𝑖𝑣e

And the better results are shown when using both the untouched dataset and the generated synthetic dataset, having an 8% improved accuracy.

**6 Early detection of obstacle/pedestrian while driving**

They have developed near range surface obstacle sensing system using vision sensor, which can alert the driver, if any obstacle is entering the drive region or in the drive region within 15 meters.

To improve real time performance of deep learning model, proposed model first predicts the drive region on road using monocular camera based on blind point (BP).

You Look at Once darknet v3 detector is integrated with tracking algorithm Deep simple online and real time tracking and multilevel perceptron neural network does early predication to alert the driver based on spatial information of detected obstacle.

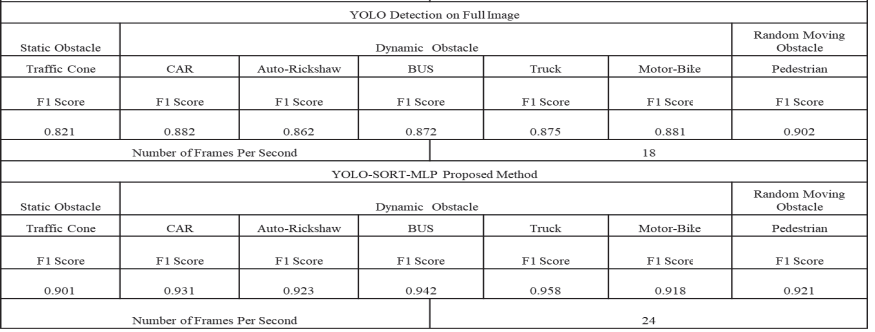
**6.2 Methodology: Camera calibration**

They are adjusting parameters of a camera lens to store images of the unstructured objects on road, which guarantees that in-car driver assistance system receive accurate and correct information. With the help of Open CV

**6.2.1 Unstructured Object Detection**

YOLOv3 variant of darknet architecture which has 53 layers trained on ImageNet dataset is used as pre-trained model. YOLOv3 contain 106 layers which include skip connections and sampling

**6.2.2 Driver Alert Prediction**

A multilayer neural network makes accurate decision to alert the driver during navigation based on YOLO ROI input variables and distance of the object from the vehicle.

**6.3 Results and setback**.

There is a comparison between YOLO detection on full image and YOLO-SORT-MLP proposed model, the F1 score in the proposed model is higher as compared to the model without sorting.

The proposed system only gives alert to the driver, which might get ignored and may lead to an accident. To overcome this drawback, the visual alert system can be merged with a Controller Area Network (CAN) signal to take necessary actions.

Jaywalking detection

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Model /Algorithm | Dataset | Accuracy |
|  |  |  |
|  |
| Motion Path analysis | Linear predictive and linear fitting algorithm | Video set from Guangzhou(place) | 90% |
| Artificial Neural networks | ANN | Video set from Dhaka, Bangladesh. | P=0.185, p=0.417 (at 95% confidence level) |
| Thermal and Camera, Deep learning | Hierarchical Framework: Classification, segmentation | Video set from Thermal camera | 96% |
| Convolution Neural Network  And Date generation | Mask RCNN-GoogLeNet classifier | Generated using SCOCO dataset | 96.7% |
| Obstacle Detection | YOLO v3 and tracking algorithm | Images extracted from CCTV | 92.1% |
|  |  |  |  |
|  |  |  |  |

Source: Reference

**7 Pedestrian: Jay walking**

**7.1 Who is Jay walking**

As mentioned earlier in the introduction that the pedestrian is jaywalking, but in the Indian scenario we cannot only aspect people to be on the road but also animals, object etc. which is a concerning topic for the autonomous cars.

we can do one thing that is to perform pedestrian analysis or pedestrian detection, for the autonomous car beforehand so that the action required to deal with the jaywalker/object/animal can be taken at a faster rate.

**8. Particle-based tracking model for automatic anomaly detection.**

The proposed method is based on particle-based trajectories, analyzed through a cascade of HMM and HDP-HMM models

In this the model will automatically discover recurrent activities occurring in a video scene, and to identify the temporal relations between these activities, e.g., to discover the different flows of cars at a road intersection, and to identify the traffic light sequence that governs these flows.

6.1 **Methodology: proposed system**

**6.1.1 Particle based tracking**

The principle of the proposed tracking is to quickly locate moving objects in the scene using randomly distributed particles in the image

Particle's trajectories are finally analyzed and filtered by computing a set of various features (e.g., linearity of track, track length, track duration, track direction, start/stop particle location, etc.) to recycle useless ones

**6.1.2 HMM/HDP-HMM cascade**

This comprises of two stages:

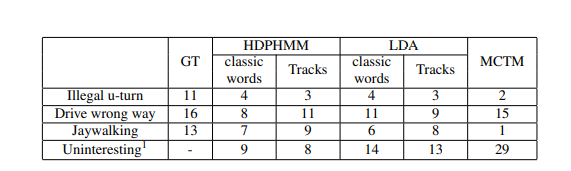
In first HMM stage to classify the particle-based trajectories, and a second HDP-HMM stage to identify the co-occurring trajectories and the temporal relation between them.

For the second stage, we adopt a HDP-HMM [12], that automatically determines the different activities in the scene and the temporal relation between them.

**6.2 Dataset utilized**

This dataset contains 50 minutes of one CCTV footage at a resolution of 360x288 (30 fps). 5 minutes is used for training and the remaining data for testing purpose.

**6.3 Results and Setbacks**

Below table shows the possible situation a pedestrian can behave that is, interpretation of the clip discovered, especially for the jaywalking category, the precision and recall rate were respectively 38% and 45% of recall for MCTM, while for the same recall rate, we reach a precision rate of 72%

The use of features not directly related to object motion (e.g., background subtraction outputs) would also be interesting to cope with temporal behaviors including static stage (e.g., cars stopped at traffic light, or people standing by) or features like speed would help to distinguish pedestrians and vehicles activities.

**7 Active Online Anomaly Detection using Dirichlet Process Mixture Model and Gaussian Process Classification**

In this we have two models:

i) Dirichlet process mixture models (DPMM) based modeling of object motion and directions in each cell.

ii) Gaussian process based active learning paradigm involving labeling by a domain expert.

7.**1 Methodology: proposed system**

Dirichlet process mixture model (DPMM)-based modeling of object motion and directions within each cell of pixels to generate a fine-grained representation of scene activities.

For active anomaly detection they adapted a Gaussian Process framework to process incoming samples sequentially, seek labels for confusing or informative samples and update the AD model.

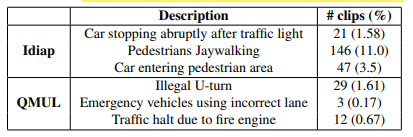
**7.2 Dataset utilized**

TJ and QMUL datasets

**7.3 Results and setbacks**

The Idiap Junction data [31] (Fig.2(a)), is a video from a busy road junction. The video is 44 minutes long, and recorded at 25 fps with a frame size of 360 × 288.

The QMUL Junction data [14] (Fig.2(b)) is filmed at a four-road junction. The video is 1 hour (90000 frames) long, recorded at 25 fps at 360 × 288 resolution



**8 A Machine Learning Approach to Pedestrian Detection for Autonomous Vehicles Using High-Definition 3D Range Data**

In this they have used automated sensor-based system to detect pedestrians in an autonomous vehicle application

**8.2 Methodology: proposed system**

**8.2.1 CIC Autonomous Vehicle**

Structure of any autonomous vehicle

**8.2.2 Sensor System**

The objective of a sensor system is to gather data from the surrounding environment of the AV and feed that data to the control system, where it would be processed and fused to decide what the AV should do next.

**8.2.3 Pedestrian Detection Algorithm**

This algorithm revolves around the processing of clusters of points contained inside a cube of dimensions 100 × 100 × 200 cm.

1. Select those cubes that contain a certain number of points, inside a threshold. This allows to eliminate objects with a reduced number of points which could produce false positives. Also, it reduces the computing time.

2. Generate XY, YZ and XZ axonometric projections of the points contained inside the cube.

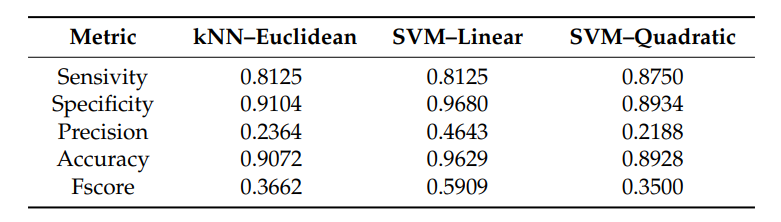
3. Generate binary images of each projection, and then pre-process them.

4. Extract features from each axonometric projection and 3D LIDAR raw data. 5. Send the feature vector to a machine learning algorithm to decide whether it is a pedestrian or not.

**8.2.4 Machine Learning Algorithm**

Tested the performance of three ML Algorithms (MLA) to detect pedestrians: k-Nearest Neighbors (kNN), Naïve Bayes Classifier (NBC), and Support Vector Machine (SVM). These algorithms were used to implement binary classifiers to detect two kinds of objects, pedestrian and no pedestrian, and they were tested with different configuration parameters and kernel functions

**8.3 Results**

Pedestrian detection has traditionally been performed using machine vision and cameras, but these techniques are affected by changing lighting conditions. 3D LIDAR technology provides more accurate data (more than 1 million points per revolution), which can be successfully used to detect pedestrians in any kind of lighting conditions

The high success rate and scalability of machine learning algorithms will enable the detection of different objects during vehicle navigation.

**9 Real-Time Pedestrian Detection on a Xilinx Zynq using the HOG Algorithm**

Advanced driver assistance systems (ADAS) are the key to enable autonomous cars in the near future. One important task for autonomous cars is to detect pedestrians reliably in real-time.

**9.1 Methodology: proposed system**

**9.1.1 HOG ALGORITHM IMPLEMENTED IN HARDWARE**

The first step of the algorithm calculates the luminance value for each pixel in case of a colored image.

the HOG descriptor is computed by dividing the image in 8x8 pixels that are called cells. In each cell, the two gradient components in the x- and y- directions are determined.

**9.1.2. Classifier**

It is based on the idea of creating a strong and accurate classifier by combining together several weak and inaccurate classifiers.

AdaBoost classifiers are most suitable when dealing with dense features sets which is the case in the HOG algorithm.

**9.1.3 HOG ALGORITHM IMPLEMENTED IN SOFTWARE**

In addition to the hardware implementation of the HOG algorithm, a software approach using OpenCV 2.4.1 has been implemented.

The OpenCV library offers an implementation of the Dalal and Triggs HOG algorithm along with a pre-trained SVM classifier for human detection. This function is used to detect pedestrians in 1920x1080 input frames.

**9.3 Results and setbacks**

The hardware implementation has a reliable detection rate of 90.2% using a classifier trained by an AdaBoost algorithm and minor false positive rate of 4%.

Future work is to improve further the frame processing time of the software implementation by exploiting NEON cores from the ARM processor.

**10. Design of a Real-time Pedestrian Detection System for Autonomous** Vehicles.

In this they have designed a real-time pedestrian detection system for autonomous vehicles is proposed and its performance is evaluated using images from standard datasets as well as Realtime video input.

**10.1 Methodology: proposed system**

The input frames captured from camera are first processed to obtain Histograms of Oriented Gradients (HOG) features.

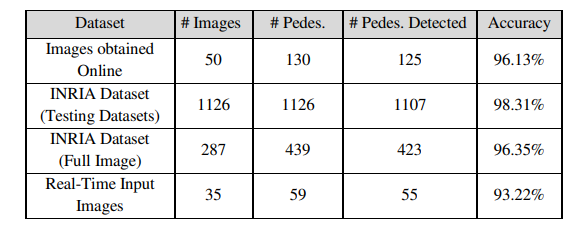
The input image is divided into 256 cells with a cell size of 16 x 16 pixels and each cell is divided into four sub-cells with a sub-cell size of 8 x 8 pixels.

**10.1.1 SVM Classifier**

SVM is considered to be the simplest and fastest classifier for both linear as well as non-linear classification problems. SVM learning aims at finding a good hyperplane in a higher dimensional feature space, that best separates two classes.

**10.2 Results**

The system is capable of detecting pedestrians with an accuracy of 98.31% and it is observed that the non-detected pedestrians are also detected, once they come closer to the camera, thus achieving 100% recognition accuracy.



**11 Multimodal Multi-Pedestrian Path Prediction for Autonomous Cars**

**11.1 Methodology: proposed system**

future position of pedestrians in traffic scenarios is required for safe navigation of an autonomous vehicle but remains a challenge named M2P3, which combines a conditional variational autoencoder with recurrent neural network encoder-decoder architecture in order to predict a set of possible future locations of each pedestrian in a traffic scene

a pedestrian heading towards a t-intersection, has an equal probability of going either left or right.

The pedestrian trajectory prediction problem is modeled with a generative model, a CVAE, where the posterior distribution P (Y | X) is learned with the help of a latent variable Z.

The M2P3 gets as an input the trajectory and scale (represented as X) of each detected pedestrian in ego-view and predicts for each input for about two third of one second (n = 10 frames) the three most likely future trajectories Yˆ for one second into the future (m = 15 frames).

**11.2 dataset**

Annotated collection of short video clips, capturing typical urban traffic scenarios in various weather conditions. The clips are taken from a single RGB camera, mounted behind the windshield of a moving car. All pedestrians are manually annotated with bounding boxes and unique tracking identifier.

The ratio between training and validation videos is 80% to 20% for fine-tuning the hyper-parameters of the implemented M2P3 model.

**11.3 Results**

The agent trajectories in the SSD dataset are more difficult to predict than the ones in the ETH/UCY dataset because there are 60 different scenes, each having its own layout and agents interacting between each other.

**12 Pedestrian Detection Based on Deep Learning**

In this they have integrated deep learning and combining a new type of local pattern with the RGB raw image as input, instead of using just the RGB image as input. Uses a new type of local pattern called Triangular Patterns.

**12.1 Methodology: proposed system**

tiled convolutional neural networks for object detection. Based on color images in RGB, the paper uses Tile-Convolutional Neural Networks (T-CNN) for object detection in images.

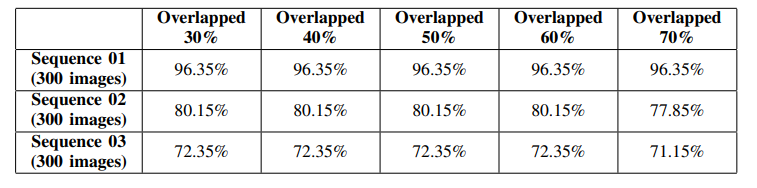
System is composed of two big steps: Obtaining the local patterns of the original image and using both the local patterns and the original RGB raw image for input into either T-CNN or Faster R-CNN to detect pedestrians.

**12.1.2 Input of Triangular Patterns plus RGB images in T-CNN**

T-CNN can take two types of data at once for input. Here, Triangular Patterns will also be used as input along with the original RGB image instead of using only one type of image as input or combining the two to make one image

**12.2 Results**

Using Triangular Patterns with the RGB raw image for input is effective in increasing the accuracy of pedestrian detection.



Future plane can include to deploy the model on Nvidias Drive PX2 hardware, which has a different GPU architecture with a normal computer Drive PX2 has a computation power about that of a desktop computer equipped with a Nvidia Geforce GTX TITAN X graphics card and is used for various projects worldwide including projects for autonomous driving.

Pedestrian detection, involving Jaywalking

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Model/Algorithm | Dataset | Accuracy  For jaywalking among others |
|  |  |  |  |
| Particle-based tracking | HMM/HDP-HMM cascade | CCTV footage | 35% |
| Anomaly detection | Dirichlet Process (DPMM) | TJ-QMUL | (Tj) 11% |
| Pedestrian Detection | kNN,NBC,SVM | 3D High range Data | 90.72% |
| Detection on Xilinx Zynq | HOG algorithm,AdaBoost classifier | Raw images from public sources size(1920x1080) | 90.2% |
| Real time Pedestrian | HOG Algorithms classifier | INIRA | 98.31% |
|  |  |  |  |
| Path Prediction | M2P3 | Video from SSD | 80% |
| Detection using Deep Learning | T-CNN, triangular Method | RGB images | 96% |
|  |  |  |  |

Source: Reference

**Conclusion:**

Increasing vehicle density on roads and unethical driving is a major cause for unstructured traffic environment, and it is leading to lot of road accidents. Jaywalking is one of the incidents that can be a major problem for the autonomous vehicles, there comes the requirement of jaywalking classifier trained with generated dataset which can classify if the pedestrian is jaywalking or not, so that it alerts the autonomous car to find an alternate route, in this paper we have considered few different methods that are present online, and analyzed them depending upon their accuracy and F1 score. In future, we can investigate the framework with a much larger dataset in varying Indian scenarios.

**References**

[1] https://ascelibrary.org/doi/10.1061/41184%28419%29484

[2] <https://trea.com/information/method-and-system-for-detecting->jaywalking-of-vulnerable-road-users/patentapplication/134be2be-e36c-435c-a60e-390522d00f4c

[3] https://www.sciencedirect.com/science/article/abs/pii/S0925753521002927

[4]https://openaccess.thecvf.com/content/ACCV2020W/MMHAU/papers/John\_Visible\_and\_Thermal\_Camera-based\_Jaywalking\_Estimation\_using\_a\_Hierarchical\_Deep\_ACCVW\_2020\_paper.pdf

[5] https://ieeexplore.ieee.org/document/9078526

[6] https://ieeexplore.ieee.org/document/9002910

[7] https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8954864

[8] <https://www.mdpi.com/1424-8220/17/1/18>

[9]https://www.researchgate.net/publication/355852022\_Pedestrian\_and\_Vehicle\_Detection\_in\_Autonomous\_Vehicle\_Perception\_Systems-A\_Review

[10] <https://ieeexplore.ieee.org/document/6116394>

[11] <https://www.sciencedirect.com/science/article/abs/pii/S0001457520302475>

[12] <https://ieeexplore.ieee.org/abstract/document/7926657>

[13] <https://www.mdpi.com/1424-8220/17/1/18>

[14] <https://www.mdpi.com/1424-8220/17/1/18>

[15] <https://ieeexplore.ieee.org/abstract/document/7393339>

[16] https://ieeexplore.ieee.org/abstract/document/7393339

[17] <https://dl.acm.org/doi/abs/10.1145/3447332.3447333>

[18] <https://ieeexplore.ieee.org/abstract/document/6521941>

[19] <https://www.sciencedirect.com/science/article/abs/pii/S0031320321000339>

[20] <https://www.mdpi.com/1424-8220/22/3/1082>

[21] <https://www.sciencedirect.com/science/article/abs/pii/S0097849319301323>

[22] https://ieeexplore.ieee.org/abstract/document/9440863