Pedestrian and Vehicle Detection for Advanced Driver Assistance Systems

Lavanya P*, Harshith G**, Chiraag*** and Shylaja S S****
* Student, PESIT
lava281995@gmail.com
**Student, PESIT
harshit.gowda78@gmail.com
***Student, PESIT
chiraag7@gmail.com
****Professor, PESIT
shylaja.sharath@pes.edu

Abstract: Application of image processing to real-time systems is growing at a rapid speed and has great potential for exploitation. Human errors are inevitable and a system designed with predictable response time and a deadline, aid in overcoming these shortcomings. This paper implements a pedestrian and vehicle detection system that combines Histogram of Gradients (HOG) and Support Vector Machine (SVM) with a Cascade Classifier and Haar. The implementation described processes every 10th frame of the video and is tested on standard datasets. Error percentage was calculated using Weak law of large numbers for the algorithms and the improvised result after preprocessing the image.
Our approach extends the functionality of the inbuilt people detector and trains a classifier to detect 4 wheelers, it minimizes processing delay and errors by tweaking the region of interest as per the requirement of a real-time application.

Keywords: HOG, SVM, Cascade Classifier, HAAR, Weak law.

Introduction

Urbanization has led to the rapid increase in the number of vehicles on the road, thereby increasing the need to control congestion due to vehicular traffic and improvise safety measures for pedestrians. Amongst the top causes for road accidents the most prominent ones are due to distracted driving, recklessness and weather adversities. Traffic monitoring and surveillance are important for road safety and management. Advanced driver system assistance has played an integral role in development of Intelligent Transportation System. Looking at a statistical projection of traffic fatalities, the death toll due to road accidents is about 1 million per annum, urging the need to improvise safety measures overcoming the ignorance and carelessness of humans. Our implementation of an advanced driver system aims to do so by integrating a pedestrian and vehicle detection system under adverse weather conditions like fog.

Related Work

There is an extensive literature on object detection and here we mention a few relevant examples that helped us decide implementation specifics for vehicle and pedestrian detection.
Corner detection was tested; initially the drawback was low accuracy and false detection. Sift and Surf was not accurate enough for pedestrians. The trade-off between accuracy and output latency time was solved by choosing HOG and SVM which were comparatively slow but accurate for pedestrians and Haar Feature-based Cascade Classifiers for vehicles which would show much lesser variations in comparison to pedestrians.
Histograms of Oriented Gradients for Human Detection [1] combined with ideas from Chris McCormick [3] tutorial implements a Histogram normalization and block normalization technique that make it invariant to illumination and to an extent posture changes as well. Hence OpenCV, comprising of a class to run the HOG person detector was used for implementing the pedestrian detection. Haar Feature-based Cascade Classifiers [2] implemented for face detection was extended and applied using a machine learning technique where a cascade function is trained from a set of positives and negative images, which is then used to detect cars in test images/frames. Our implementation combines both techniques to extract the best features in each and tries to minimize existing flaws and response time.

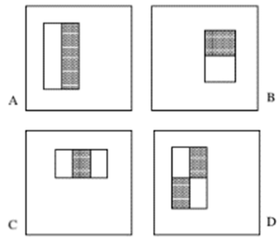

Figure 1. Haar rectangle features [4]

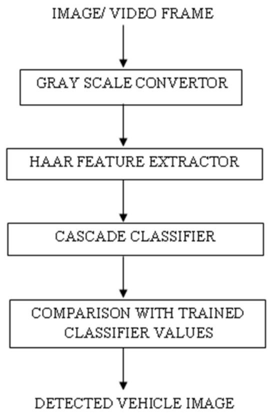

Figure 2. Data flow diagram for vehicle detection

Constraints on a Real Time System

Continuous and interactive processing: Must process high frame rate video input to prevent missing detail under fixed time constraints.
Fixed Deadlines: Missing a deadline implies a system is malfunctioning at its peak.
Predictable response time: In order to alert a driver, the response time should be within a specific range.
Efficient functioning under adverse weather conditions: Detection must also be successful under fog which produces images with high noise.

Proposed Method

Vehicle Detection

To identify vehicles, the cascade classifier (Haar) was trained to recognize objects within the image; this required a set of positive images that contain the object we are trying to detect and negative images. This completed the phase of asset preparation. [4]. A metadata file is then generated, which contains the file name, number of objects, position and dimensions of objects. This metadata file is then used to create a vector file that will be used by the classifier containing all positive images within a single file. The script available in OpenCV library is then invoked to train the classifier. A Python script using the classifier to detect vehicles and to highlight each detected vehicle was implemented.

Pedestrian Detection

HOG was chosen as a feature descriptor [3] as it provides the functionality of reproducing the same or similar feature descriptor when an object is viewed under noise conditions. The HOG person detector is used that implements a sliding window detector technique which traverses the entire image that in turn influences the processing delay. For every position of the window, a HOG feature descriptor is generated which is then fed into a trained SVM, which classifies it as either "pedestrian" or "not a pedestrian". The implementation of the HOG person detector used here takes reference from Chris Mccormick's work [3] that uses a detection window of 64X128 pixels confining it to a 9-bit histogram ranging from 0 to 180 degrees. To make the process invariant to changes in illumination or noise, the cells are first grouped into blocks prior to normalization (Fig. 3). This implementation of HOG was used from the OpenCV documentation to detect pedestrians.

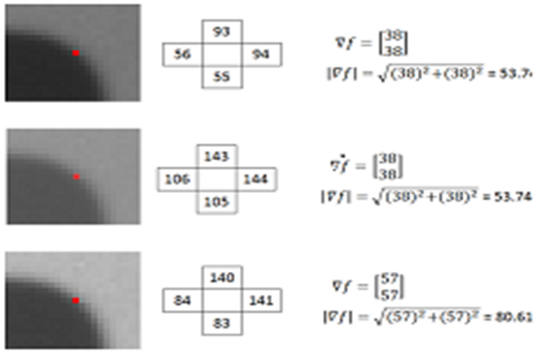

Figure 3. Illumination and pixel invariance [3]

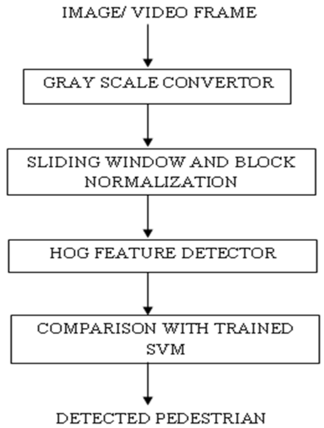

Figure 4 .Data flow diagram for pedestrian detection

Video Processing

The trade-off between output latency and accuracy of detection was solved by processing every 10th frame of the video.
The region of interest is chosen so that the unnecessary parts of the image are not considered for processing. Both pedestrian and vehicle detection are integrated to get a combined result.

Error Calculation

A mathematical model for the determination of error in the inbuilt HOG and HAAR was built using Weak Law of Large Numbers.

Error Calculation for pedestrians

To provide a mathematical model for percentage of error in the feature descriptor provided by HOG for person detection,
we used weak law of large numbers [8] to state the error percentage with an accuracy meter and confidence meter. Various symbols used are explained in Table1.

Applying Pollsters algorithm,

$$P(|\mu - f| \geq \delta) \leq C$$

Applying Chebyshev Inequality states,

$$P(|\mu\text{-f}| \geq C) \leq \frac{\sigma 2}{c2}$$

Combining the two algorithms, we establish a result for n=504, the probability of detection of the pedestrian was found to be 58± 9% with a confidence of 95%.

Table 1. Symbols and their meaning

Symbol	Quantity
F	Probability of detection
μ	Sample probability of the sample space
X	1 for successful and 0 for unsuccessful detection
Δ	Error
C	1-{confidence of μ lying in (f ± δ) }
σ^2	Variance
σ_{mn}	σ^2/total number of objects

Figure 5. Processing stages for pedestrian detection

Figure 6. Detection under Fog condition

Figure 7. Processing stages for vehicle detection

Error calculation for vehicles

To minimize the detection of extra objects, we used a preprocessing technique that changed the region of interest on which the algorithm is implemented. The upper 33% of the image is cropped to remove extra objects detected in that horizon and above, also significantly reducing the processing delay. The region of interest is confined to the road and eliminates detection on the side pavements by mathematical manipulations, increasing the accuracy factor by 73%.

Result

We have implemented a detection system for vehicles using a Haar Feature-Based Cascade classifier, combined with HOG and SVM for pedestrian detection that provides low latency outputs using preprocessing techniques.
The algorithm was tested on the following datasets:
Frames of video [5]
A mathematically modeled proof for improvement in accuracy rate was determined using weak law of large numbers, pedestrian detection was found to have a success rate of 58%±9% (error rate) with a confidence of 95% in the video frames. The modification of region of interest improved accuracy of detection of vehicles in the video by 73%.
Pedestrian dataset from MIT [6]
Using weak law of large numbers, pedestrian detection was found to have a success rate of 84%±5% (error rate) with a confidence of 90% in the training data.
The algorithm successfully detects pedestrians under fog conditions (Fig. 6).

Conclusion

A detection system using a feature descriptor was implemented that detects both pedestrians and vehicles, decreasing latency time of the current algorithm. It currently employs a preprocessing technique to minimize false positives, although there is room for optimization using algorithmic techniques. The code detects pedestrians when full body is visible, expansion of this to two wheelers by increasing the size of dataset has potential scope.

Acknowledgement

We wish to express our sincere gratitude to our college PES Institute of Technology, Bengaluru for providing us the opportunity to implement this project and for their support.

References

[1] NavneetDalal and Bill Triggs, "Histograms of Oriented Gradients for Human Detection", CVPR Conference.
[2] http://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html#gsc.tab=0
[3] https://chrisjmccormick.wordpress.com/2013/05/09/hog-person-detector-tutorial/
[4] P Viola and M Jones. "Rapid Object detection using a boosted cascade of simple features", CVPR Conference.
[5] https://www.youtube.com/watch?v=98t-yh4MC6Y
[6] http://cbcl.mit.edu/software-datasets/PedestrianData.html
[7] http://cogcomp.cs.illinois.edu/Data/Car/
[8] https://www.youtube.com/watch?v=j9WZyLZCBzs&list=PLu14u3cNGP61 MdtwGTqZA0MreSaDybji8