2015 International Conference on Computational Intelligence and Communication Networks

LBP-Haar Multi-Feature Pedestrian Detection For Auto-Braking and Steering Control System

Bharadwaj Vishanth Thiyagarajan[1], Andhare Mayur[2], Bothara Ravina[3], Govindwad Akilesh[4].

COMPUTER ENGINEERING.
P.I.C.T (University of Pune)
PUNE, INDIA

Abstract: Fatality due to road accidents are increasing with the increase in population and number of vehicles. Intelligent systems are developed to counter act the loss due to road accidents. The paper proposes one such method to counter the accidents by the implementation of pedestrian detection by the use of LBP histogram and HAAR-like features. LBP histogram are used for cross checking the HAAR-like features where the upper body, lower body, face are detected using the Haar like features and LBP classifiers. The twin stage algorithm is used because LBP classifiers are 20% percent faster than Haar based algorithm and Haar features are more accurate than LBP classifiers. Thus this can be considered as a 4 layer cross check using two different algorithms in a cost efficient way to improve the accuracy. The use of BeagleBone Black(cortex A8 sitara) is made here for the simulation of vehicle control which includes auto-braking and steering locking. The implementation is a 2-phase system, which includes city-drive and high-way drive for speed control.

***Key Words* - Object Detection, face detection, Vehicle Control, Real-time, Ada-Boost, LBP classifiers, Haar classifiers .**

I. INTRODUCTION

An attempt is made in this paper to improve the pedestrian detection system by the use of LBP classifiers and Haar-like features where these features can be used to cross check and provide more accurate results.

Pedestrian Detection is currently a growing field to reduce the casualties due to car accidents. It is used in ADAS (Advance Driver Assistance System). The scope of these algorithms can be increased where they can be used to detect animals also, which can decrease the number of casualties. The ADAS must be robust and is classified as a hard real-time system. Hence the system so created must function accurately without any delay. Hence the use of Beagle-Bone Black is used. Since it is an embedded technology, it can be used to perform replace spinlocks and preemptable-mutexes by installing the real-time kernel patches, which can perform task per-emption to suffice the real-time needs of ADAS.

The fields of computer vision is also used in many other Hard-real time applications like endoscopy, brain surgery, space travels for detecting meteors and also detecting anomalies in the atmosphere with the help of satellites. The beagle-bone is used for simulation purpose where the LEDs were attached and used in the place of accelerations and brakes accordingly for testing purpose.

II. RELATED WORKS

This paper contains a 4-stage mechanism. Object tracking, LBP based detection, Haar-like feature detection, Action performed.

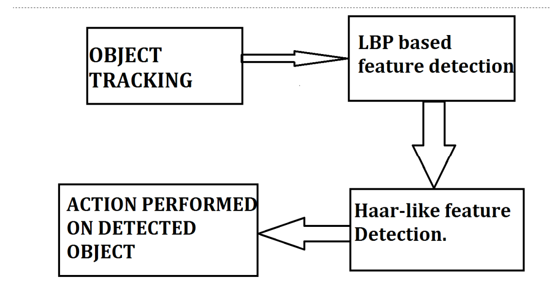

Fig.1. Flow Diagram of the Component.

1. OBJECT-TRACKING algorithm:
   a. Mean -Shift.
   b. Histogram back-projection.
   c. Continues adaptive mean (CAM) – shift.
2. LBP based feature detection.
   a. Local Binary Pattern.
3. Haar-like feature detection.
   a. HAAR-features
   b. Integral Image
   c. ADA-BOOST algorithm.
   d. Cascading the classifiers.
4. Action Performed on detected object.

III. OBJECT TRACKING ALGORITHM

A. *Mean Shift Algorithm:*

1.) Image conversion to HSV.
2.) Image selected by variable H (hue).
3.) Choose the search window size.
4.) Primary position of the search window is chosen.
5.) Examine for the centre of mass of the object.
6.) Examine the centre of mass of the search window.
7.) Examine the parameters for the above task.

B. Histogram Back Projection:

1.) Choose a window to track the face.
2.) Extract the hue values and form a histogram.
3.) Convert the pixel of captured frame to Histogram pixel.
4.) Repeat step 1

978-1-5090-0076-0/15 $31.00 © 2015 IEEE
DOI 10.1109/CICN.2015.340

Fig. 2. Back Projection

C. CAM Shift:

1.) Algorithm adapts the screen size of the object being tracked.

2.) Window width = S.

3.) Window Length= 1.2S.

4.) M00 = Moment of order 0.

$$S = 2 * \sqrt{M_{00}} / 256 \quad (1)$$

IV. HAAR FEATURES

A. Haar Wavelets:

Detecting object classifier which is based on the value of simple features; instead of going through pixel directly there are many motivations to go through sample features. There are some features, which is shared by all human faces. For object detection there are several steps, it's need to follow: very first step is edge detection, in edge detection set of mathematical methods which aim to identifying points in a digital image at which image brightness change sharply are organized into set of curved line segments termed as edges.

Fig. 4. Edge Detection

In the above image. The output image contains some high(white) intensity pixel which perfectly matches the convolution kernel pixels, which are derived from the image pixel. Viola-Jones algorithm based on the same principle explained above. Here any object detection performance includes scanning of the object and filling it in the form of black and white pixels. Then the calculation is made for every feature where the sums of the black pixels are calculated and the white pixels are calculated. Then the subtraction is done where the white pixels sum is subtracted from the black pixels sum. There are 5 types of Haar features. which is shown below.

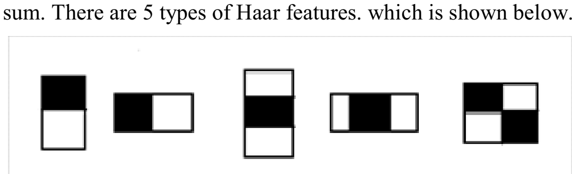

Fig. 5. Five basic types of Haar Features

A 24x24 window is used for training purpose and includes the features like position, scale and type so the total features go beyond 160,000+ features. The regions have the same size and shape and horizontally and vertically adjacent

V. LBP

LBP features are very much helpful in performing various applications ,including texture classification and segmentation, image retrieval and surface inspection. The face image is divided into several regions from which the LBP feature distributions are extracted and concatenated into an enhanced feature vector to be used as a face descriptor. The LBP operator was originally designed for texture description. The operator assigns a label to every pixel of an image by thresholding the 3x3 neighborhood of each pixel with the center pixel value and considering the result as a binary number. Then the histogram of the labels can be used as a texture descriptor.

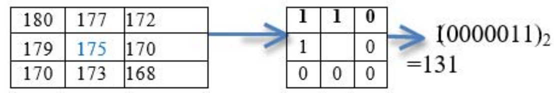

Fig.6 256-bin Histogram of the labels.

Different texture primitives are Spot, Flat, Line end, Edge, Corner. Different neighbor sizes used by LBP operator. The operator $LBP_{4,1}$ uses 4 neighbors while $LBP_{16,2}$ uses 16 neighbors with radius of 2.

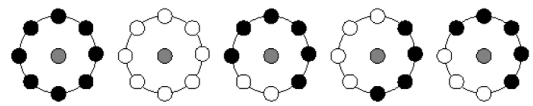

Fig.7.Examples of texture primitives in LBP

The notation for operator $LBP_{P,R}$ refers to a neighborhood size of P equally spaced pixels on a circle of radius R that form a circularly symmetric neighbor set. It produce 2^P different output values, corresponding to the 2^P different binary patterns that can be formed by the P pixels in the neighbor set. Therefore, it is possible to use only a subset of the 2^P LBPs to describe textured images. These fundamental patterns are those with a small number of bitwise transitions from 0 to 1 and vice-versa. For example 00000000 (0 transition)11111111 (0 transition) while 00000110 (2 transitions) and 01111110 (2 transitions) are uniform LBP whereas the patterns 11001001 (4 transitions) and 01010011 (6 transitions) are not. Accumulating the patterns, which have more than 2 transitions into a single bin, yields an LBP descriptor.

A. LBP Based Facial Representation:

Each face image can be considered as a composition of micro- patterns, which can be effectively detected by the LBP operator. Ahonen et al. introduced a LBP based face representation for face recognition. To consider the shape information of faces, they divided face images into M small non-overlapping regions R0, R1, ..., R (as shown in Figure). The LBP histograms extracted from each sub-region are then concatenated into a single, spatially enhanced feature histogram defined as:

$$H_{i,j} = \sum_{x,y} I(f_1(x,y) = i) I((x,y) \in R_j) \quad \ldots(3)$$

Where i = 0,..., L-1, j = 0, ..., M-1. The extracted feature histogram describes the local texture and global shape of face images.

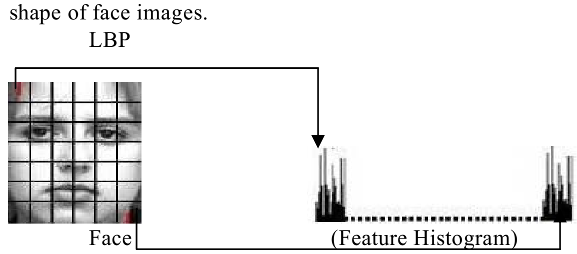

Fig.8.LBP based facial representation.

VI. ADABOOST:

Adaptive Boosting is a Machine Learning algorithm which can be conjunction with many other types of learning algorithm to improve their performance. It is used to create a strong classifiers by conjuncture of many weak classifiers. A weak classifier is simply a classifier that performs poorly, but performs better than random guessing. There is always a chance that the weak classifiers can detect more than 50% faces in the given set of images. The remaining classifiers carry this chain where each classifier can detect more than 50% of the remaining faces. All these detected features are combined to get a strong classifier, which can detect the faces. This concept is applicable not only to faces but also to many other objects. Adaboost helps you choose the training set for each new classifier that you train based on the results of the previous classifier and also determines how much weight should be given to each classifier's proposed answer when combining the results.

A simple formula is demonstrated below:

$$F(x)=\alpha 1f1(x) + \alpha 2f2(x) + \alpha 3f3(x) +\alpha 4f4(x) \quad ...(2)$$

Į = assigned weight.

$f_N(x)$= weak classifier (features tracked).

F(x) = strong classifier generated from the tracked features.

Every weak classifier can have two outputs namely '1' or '0'. The output '1' is generated when a successful detection is done and the output '0' is generated when detection fails.

A. *Training Set Selection:*

Each weak classifier should trained on a random subset of the total training set AdaBoost assigns a "weight" to each training example, which determines the probability that each example should appear in the training set. Examples with higher weights are more likely to be included in the training set, and vice versa. After training a classifier, AdaBoost increases the weight on the misclassified examples so that these examples will make up a larger part of the next classifiers training set, and hopefully the next classifier trained will perform better on them. After each classifier is trained, the classifier's weight is calculated based on its accuracy. More accurate classifiers are given more weight. A classifier with 50% accuracy is given a weight of zero, and a classifier with less than 50% accuracy is given negative weight.

One of the biggest applications of AdaBoost is the Viola-Jones face detector, which is for detecting faces in an image. The Viola-Jones face detector uses a "rejection cascade" consisting of many layers of classifiers. If at any layer the detection window is *not* recognized as a face, it's rejected and move on to the next window. The first classifier in the cascade is designed to discard as many negative windows as possible with minimal computational cost.

Each layer of the cascade is a strong classifier built out of a combination of weaker classifiers. However, the principles of AdaBoost are also used to find the best features to use in each layer of the cascade. The following figures describes the methodology in a flow manner. Part A of each figure is the pure form and part B of each figure is the form created by the weight increase.

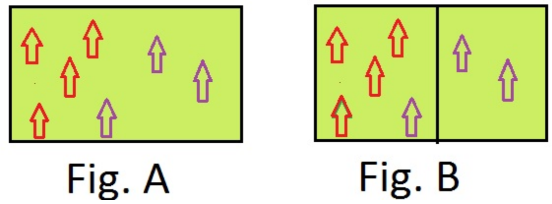

Fig. 9. Adaboost containing images for Positive and Negative images

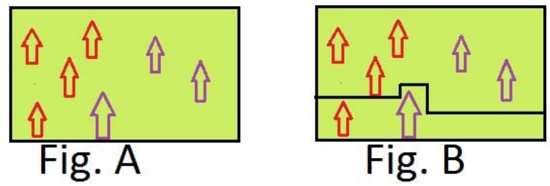

Fig. 10. Adaboost containing images for separation Positive and Negative images

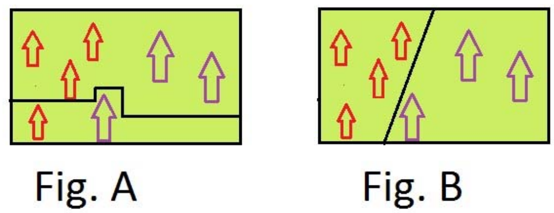

Fig. 11. Adaboost containing images for weighted Positive and Negative images

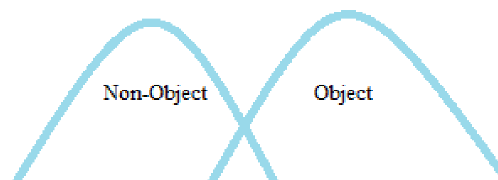

Fig. 12. Graph of Gaussian weak classifier

B. *Adaboost Algorithm:*

For n rounds we consider:

1.) Examine weighted error and choose the best value
2.) Re-weight the example taken :
Incorrect examples- higher weight
Correct example- lower weight

A combination of weak classifier generate a strong classifier.

VII. CASCADING CLASSIFIER

A. *The features of classifiers:*

Every time the algorithm has to scan the same image with new windows size and this becomes prohibitive because of the time and energy cost that will incur. Even if an image, consider that it will have n faces. But there may be negative images >= n. Hence scanning these negative images puts an extra burden on the processor costing us energy and time. Hence the algorithm must be so effective that it quickly rejects the negative images (non-faces in this case) and spend more time on analyzing the positive images (faces in this case). Hence the use of cascade of classifiers came to existence because forming a classifier out of linear combination of all these classifiers becomes quite tedious. Single classifier isn't accurate enough. It is called a weak classifier. Cascading consist of series of weak classifier that are barely better than 50% correct. If the object/area/face passes a single classifier, go to the next classifier; otherwise, it doesn't match.

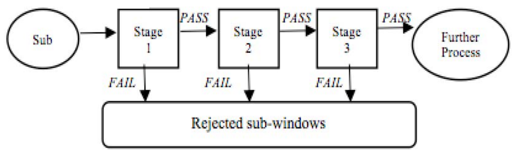

Fig. 13. Cascading and Discarding sub-window

There are three main steps in designing a classifier:

1.) Total number of stages
2.) Total number of features
3.) Threshold of each strong classifier

VIII. IMPLEMETATION

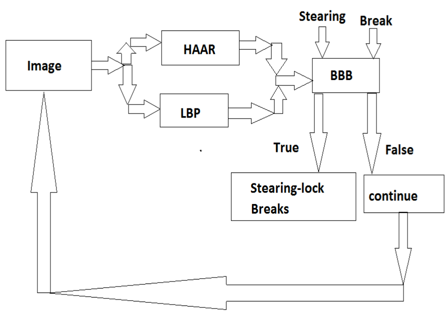

Fig.14.Flow char of the Work Flow

Terms:
BBB = Beagle Bone Black (Cortex A8 Sitara).
LBP = Local Binary Pattern.

The implementation was performed on ALTO 4 wheeler vehicle with the help of Beagle Bone Black where it was observed that LBP cascades were 20% faster than the Haar cascades and the accuracy of Haar files were 20% more than that of the LBP cascades. Hence the BBB was attached the LEDs named G and R. G denotes the steering Lock and R denotes the Auto Braking system. Whenever the object appeared in between, programming was done such that the steering lock and the Auto Braking Led glow with minimal time difference. Hence the vehicle can stop instantly with steering lock with the maximum safety of the drivers and passengers. The range of identification for the algorithms varies for every vehicle and depends on their range of their headlight. The following figures show the implementation in the normal room light for better understanding.

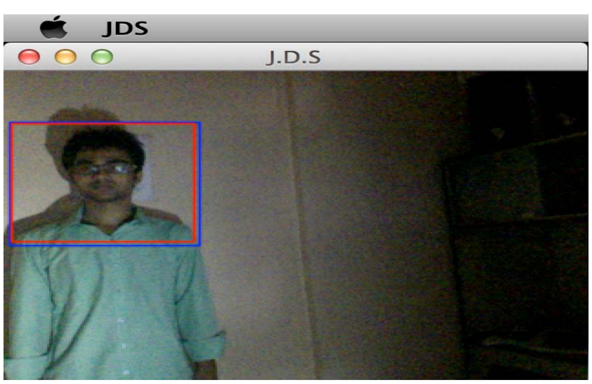

Fig.15. Implementation for Face Detection

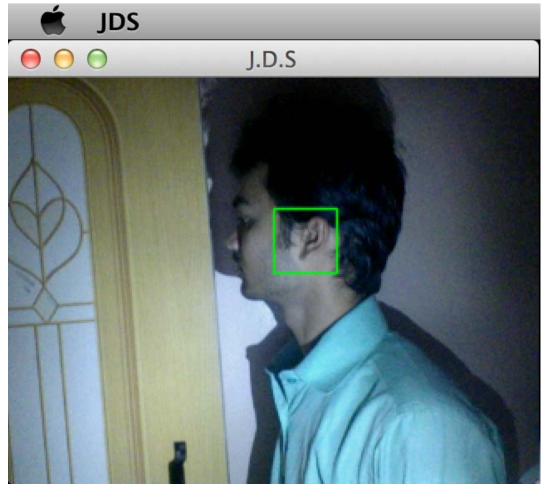

Fig.16.Implementationfor Ear detection for sideways crossing using LBP

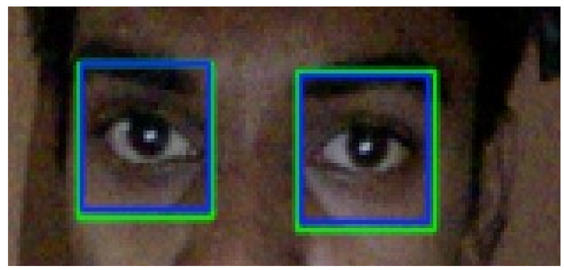

Fig.17.Testing the accuracy of LBP and Haar in eye detection

Figure 15 shows that LBP and Haar features are combined to detect Head/Upper Body for the person who is exactly in front of the camera. (blue rectangle – Haar cascade; Red rectangle – LBP cascade) Figure 17 shows a single yellow rectangle, which was scanned by the LBP cascade which is 20% faster than the HAAR cascade. Figure 18 shows an explanation of the LBP-HAAR combo where the blue represents the HAAR and the yellow represents the LBP cascade. Here the yellow rectangle appeared before the blue rectangle, which shows the speed of the LBP cascade over Haar Cascade. Since the accuracy of LBP cascade not as good as Haar cascade, The haar features are also stitched with the LBP for having both speed and accuracy.

IX. CONCLUSION

Here the every head-lights have a different range and functionalities, Hence the combination of the Haar And LBP algorithms can be used in order to improve the accuracy and speed due to their respective properties in order to increase the driver and passenger safety. The use of BBB sitara as a hardware for communication with Brakes and Steering which supports the hardware with the ease of programming in the pin slots and also ensures high reaction time with the speed of 30Mhz. Hence it becomes highly convenient and efficient hardware to handle such hard real time situations. It can also provide pre-emptive mutex locks, which can perform in extreme real time situations. It was found that the speed of the LBP cascades were 20% faster. This can be combined with the HOG for better accuracy. Hence with a better accuracy can be obtained at a high speed by passing the LBP bound figures to the HAAR wavelets for improvised tracking.

X. ACKNOWLEDGEMENT:

We would like to thanks our mentors Ananthi,Vijaya, Rotini, Sangita for supporting and guiding us through our difficulties.

XI. REFERENCES:

1) B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” Proceedings of the DARPA imaging understanding workshop, pp. 121–130, 1981.

2) J.Y. Bouguet, “Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm,” Intel Corporation, Microprocessor Research Labs, pp. 1- 9, 2000.

3) M. H Yang, “Detecting faces images, A survey,” IEEE Transations on Pattern Analysis and Machine Inteligence vol. 24, no. 1, pp. 34–58, 2002.

4) L. Stan and Z. Zhang, “FloatBoost learning and statistical face detection”, IEEE Trans. On Pattern Analysis and Machine Intelligence. vol. 26, no. 9, pp. 1112-1123, 2004.

5) M. Lades, J.C. Vorbrüggen, J. Buhmann , J. Lange, C. Malsburg, R. Würtz., and W. Konen, “Distortion invariant object recognition in the dynamic link architecture,” IEEE Trans. Computer, vol. 42, no. 3, pp. 300-310, 1993.

6) K. Sung and T. Poggio, “Example-based learning for view-based face detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 20, pp. 39– 51, 1998.

7) J. Shi and C. Tomasi, “Good Features to Track,” IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600, 1994.

8) Paul Viola, Micheal Jones, “Rapid Object detection using a boosted cascade of simple features”, Computer Vision and pattern recognition conference, 2001.

9) Bharadwaj Vishanth Thiyagarajan, “Facial Detection and facial feature points’ detection with the help of KLT algorithm”, IJARCSMS, vol. 2,issue 8, august 2014.

10) Bharadwaj Vishanth Thiyagarajan, “Revelution in Teaching/Moocs by the use of real-time face detection”, IEEE ICECCT Conference, pp. 503-510, March, 2015.

11) W Yun-Long, “ Face location with an extended set of LBP”, IEEE ICCSN Conference, 2011, page 325-328.

12) Yunlong Wei, “Face Location with LBP scale transform”, IEEE ICCCAS Conference, 2010, page 347-350.