Human detections using Beagle board-XM

CHANDAN KUMAR[1] V. AJAY KUMAR[2] R. MURALI[3]

[1](M. TECH STUDENT, EMBEDDED SYSTEMS, DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING,
VIJAYA KRISHNA INSTITUTE OF TECHNOLOGY AND SCIENCES

[2](GUIDE , ASSOCIATE PROFESSOR, DEPT OF ELECTRONICS & COMMUNICATION ENGINEERING, VIJAYA KRISHNA INSTITUTE OF TECHNOLOGY AND SCIENCES

[3](HOD OF ECE , ASSOCIATE PROFESSOR, DEPT OF ELECTRONICS & COMMUNICATION ENGINEERING, VIJAYA KRISHNA INSTITUTE OF TECHNOLOGY AND SCIENCES

Abstract—

In this paper, we describe implementation of human detection system that detects the presence of humans in the static images on DM3730 processor to optimize the algorithms for high performance. The purpose of such a model is to monitor surveillance cameras continuously where it is difficult for human operators. Human bodies are non-rigid and highly articulated hence detecting human bodies based on appearance is more difficult than detecting other objects. Human detector usually includes learning phase and detection phase. In the learning phase, Support Vector Machine (SVM) learning will be done using Histogram of Oriented Gradients (HOG) feature vectors of training data set. Training data set consists of positive (human) and negative (non-human) images. In the detection phase, Human/non-human classification will be done on the test image based on SVM classifier. The algorithms are benchmarked on the BeagleBoard xM based on low-power Texas Instruments (TI) DM3730 ARM Cortex-A8 processor. Functions and library in OpenCV which developed by Intel Corporation was utilized for building the human tracking algorithms.

Keywords: Histogram of Oriented Gradients, Support Vector Machine, Beagleboard-xM, DM3730, OpenCV

.

1. INTRODUCTION:

In the images we have different no. of objects like buildings, vehicles, animals and humans etc. but detection of humans in the images is a primary step in lot of applications like Automatic digital content management, Intelligent transportation systems, Human computer interaction and video surveillance etc. In this paper, we consider a sub-problem of object recognition: human detection. Human detection is a challenging problem due to a lot of factors. First, the within-class variations are very large. Second, background clutter is common and varies from image to image. Third, partial occlusions create further difficulties because only part of the object is visible for processing. A robust human detector must deal with the change of viewpoint, pose, clothing, illumination, and so forth. The detector must be capable of distinguishing the object from complex background regions.

We have developed a low cost, low power, portable embedded system which detects the presence of humans and count the no. of humans in the static images using Pre trained SVM (Support Vector machine) with HOG (Histogram of Oriented Gradients) features on Beagleboard-xM (Open source embedded hardware).

This paper is organized as follows: in Section 2 the related work about the detection of humans is presented. Section 3 and 4 describes the proposed method. Experimental results are shown on section 5 and finally section 6 presents the conclusion and future scope.

2. Previous work

Some research works have already been performed in order to detect the presence of humans. Previous methods on human detection differ in two perspectives: first, they may use different features such as edge features, Haar-like

features [2], and gradient orientation features. Second, they may use different classifiers such as neutral network (NN), support vector machine (SVM) and cascaded AdaBoost. Edge features have been used in earlier works for object detection. Gavrila and Philomin (1999) use edge template as the feature and compare edge images to a template dataset using the chamfer distance. The Haar-like features are first proposed by Oren,et al.(1997) as complete Haar wavelets for pedestrian detection. Later, Papageorgiou & Poggio(2000) make a thorough study of the complete Haar wavelets for the detection of face, car, and pedestrian. AdaBoost (Y.Freund and R.E.Schapire [1995]) has been applied as automatic feature selection to carry out a procedure of pattern recognition. Among these all in this project, we used HOG feature descriptors to extract the features of an image and SVM classifier used for the purpose of classification.

3. Overview of the Method

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

Figure 2: Examples of Positive training images

Figure 3: Examples of Negative training images

This chapter describes an entire project overview in brief. Human detector usually includes two phases: Learning phase & Detection phase. In the learning phase, Support Vector Machine (SVM) learning will be done using Histogram of Oriented Gradients (HOG) feature vectors of training data set. Training data set consists of positive (human) and negative (non-human) images. In the detection phase, Human/non-human classification will be done on the test image based on SVM classifier.

Figure 1: Overview of the project

3.1 Learning Phase

In the learning phase, Support Vector Machine (SVM) learning will be done using Histogram of Oriented Gradients (HOG) feature vectors of training data set. Following steps are to be done in the learning phase. Collect the training data set which contains positive and negative images of 64*128 pixels.

1. Compute Histogram of Oriented Gradient (HOG) feature descriptors for each and every image available in the training set. Computation of HOG will describe in the next chapter.

2. Support Vector Machine (SVM) learning will be done using iterative process, with the help of available feature vectors. A function will be find out in the learning process, it is used in the classification of humans from other objects in the images.

In the detection phase, Human/non-human classification will be done on the test image based on SVM classifier. For any test image, the feature vector is computed on densely spaced windows at all scales and classified using the learned SVM. Following steps are to be done in the detection phase.

1. In which image we want to detect the humans that will be captured by using the webcam called Test image.

2. Compute Histogram of Oriented Gradients (HOG) feature descriptor across all scales and window locations and the locations and scales of all positive windows are saved [1] (window size 64x128).

3. Compute the dot product of Test image HOG descriptor vector to the linear discriminant function/vector obtained in the learning process.

4. If the value greater than zero human detected in the image. The resulting modes give the final detections and the bounding boxes are drawn using this final scale with the rectangular bounding box, otherwise not.

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

4. HOG & SVM

This chapter gives an overview of our feature extraction method, which is summarized in figure 4. The method is based on evaluating normalized local histograms of image gradient orientations in a dense grid. The basic idea involved in HOG is local object appearance and shape; these are characterized by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions. This is implemented by dividing the image window into small spatial regions ("cells"), for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell [1]. For better invariance to illumination, shadowing, etc., contrast-normalization should be done before using them. This can be done by accumulating a measure of local histogram "energy" over somewhat larger spatial regions ("blocks") and using the results to normalize all of the cells in the block. We will refer to the normalized descriptor blocks as Histogram of Oriented Gradient (HOG) descriptors. Sliding the detection window with a dense (in fact, overlapping) grid of HOG descriptors and using the combined feature vector in a conventional SVM based window classifier gives our human detection.

A. Gradient Computation For gradient computation, first the grayscale image is filtered to obtain x and y derivatives of pixels using sobel operator function in OpenCV with these kernels [1]:

After calculating x,y derivatives, the magnitude and orientation of the gradient is also computed:

One thing to note is that, at orientation calculation rad2deg(atan2(val)) method is used, which returns values

between [-180°,180°]. Since unsigned orientations are desired for this implementation, the values which are less than 0° is added up with 180°.

Figure 4: HOG algorithm

B. Orientation

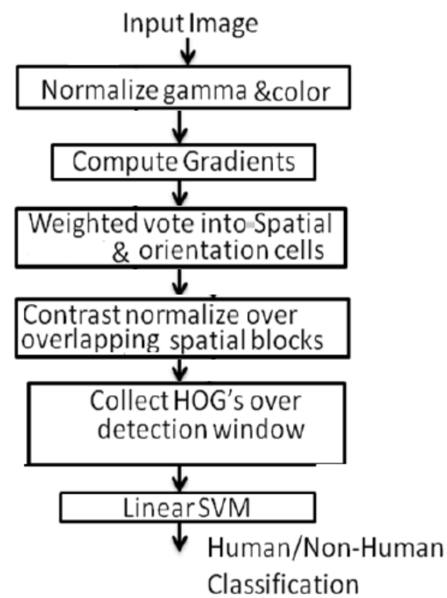
The next step is to compute histograms for each cell. Cell histograms later use at descriptor blocks. 8x8 pixel size cells are computed with 9 orientation bins for [0°,180°] interval. For each pixel's orientation, the corresponding orientation bin is found and the orientation's magnitude |G| is voted to this bin.

C. Descriptor Blocks To normalize the cells orientation histograms, they should be grouped into blocks. From the two main block geometries, the implementation uses R-HOG geometry. Each R-HOG block has 2x2 cells and adjacent R-HOGs are overlapping each other for a magnitude of half-size of a block i.e., 50%. D. Block Normalization Although there are three different methods for block normalization, L2-Norm normalization is implemented using the formulae [1][9]:

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

E. Classifier

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis [12]. Given a set of training examples, each marked as belonging to one of two classes, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. When a test image given to the classifier [13]. It extracts the Histogram of oriented features and compute the distance between linear discriminant function (vector) and computed feature vector. If the distance is greater than zero(>0), human detected in the image otherwise not.



p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

5. Experimental Results

Human detection system is successfully implemented on DM3730 by taking input as image (.jpg) and giving output as image with rectangular bounded box when human is detected. An operating system Ubuntu 12.04 is ported on to the Beagleboard-xM with DM3730 processor [10]. A Linux kernel image (uImage) is created using Linux kernel 2.6.32 which is compatible with DM3730. USB Webcam and keyboard devices are interfaced with Beagleboard-xM through USB ports. Monitor is connected to Beagleboard-xM through HDMI/DVI-D. Figure 7 shows the hardware setup of the Beagleboard-xM DM3730 with connections and μSD card. After Ubuntu OS loaded, enter the commands to initialize the webcam, capture the image and display the output result. $sudo fswebcam -r 320*240 -S 20 test.jpg $sudo ./humandetection test.jpg $sudo fbi

Figure 5: Hardware setup

detection.jpg

In the above commands, the first command is to initialize the webcam and capture the image of 320*240 pixel size. Second is to run the .exe file (here humandetection is the .exe file of source code). Third is to display the output result.

Figure 6: Output of the project

5.1 Small Training Dataset and Analysis

A small dataset for training is constituted from 25 positive and 25 negative images.

5.2 Testing Results

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

Table 1: Testing results on small training dataset

Test Image Database (Total No.of Positive images:288)	
Human	229
No Human	59 (0.208) false negative
Test Image database (Total No.of Negative Images:454)	
Human	34 (0.07) false positive
No human	420

5.3 Large Training dataset and Analysis

A large dataset for training is constituted from 100 positive and 100 negative images.

Testing results:

Table 2 Testing results on small training dataset

Test Image Database (Total No.of Positive images:288)	
Human	268
No Human	18(0.04) false negative
Test Image database (Total No.of Negative Images:454)	
Human	10 (0.02) false positive
No human	434

5.4 Performance Measures:

Precision and recall measures are widely used for evaluation of the classification tasks. They are defined as follows:

$$p = \frac{TP}{TP+FP}. \qquad r = \frac{TP}{TP+FN}.$$

Precision p is the number of correctly classified positive examples divided by the total number of examples that are classified as positive. Recall r is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set. Where,

TP: The no.of correct classifications of the positive examples (True Positive)

FN: The no.of incorrect classifications of the positive examples (False Negative)

FP: The no.of incorrect [10] www.ti.com

[11]M. Enzweiler and D.M. Gavrila, “A mixed generative-discriminative framework for pedestrian classification,” IEEE Int’l Conf. Computer Vision and Pattern Recognition, 2008.

[12]O.Tuzel, F.Porikli, and P.Meer, “Human detection via classification on Riemannian manifolds,” IEEE Int’l Conf. Computer Vision and Pattern Recognition, 2007.

[13]E. Osuna, R. Freund and F. Girosi, “An Improved Training Algorithm for Support Vector Machines,” To appear in Proc. of IEEE NNSP’97 , Amelia Island, FL, 24-26 Sep., 1997. classifications of the negative examples (False Nositive)

TN: The no.of correct classifications of

Table 3 Performance measures

	Classified Positive	Classified Negative
Actual Positive	0.93 (TP)	0.04 (FN)
Actual Negative	0.02 (FP)	0.95 (TN)

the negative examples (True Negative)

precision p = 0.978 recall r = 0.958 (from formulae) accuracy = (TP+TN)/(TP+TN+FP+FN) = 0.97 Table 4 represents the execution times on Intel PC and DM3730 Processor with two different test image sizes. From the

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

table results we can analyze that the algorithm is optimized on hardware with 1GHz processor and 512MB of RAM.

Table 4 Comparisons between HOG with SVM and SIFT

Execution Time (640*480px)	On PC (Intel pentium, 2Gb RAM)	On Board (1Ghz processor,512Mb RAM)
	16-21ms	100-150ms
Execution time (320*240px)	On PC	On Board
	8-15ms	70-85ms

Table 5 represents the comparison between performance measures of proposed algorithm i.e., HOG with pre trained SVM features and SIFT features. The proposed algorithm shows the best results interns of precision,

Table 5 Comparison of performance measures

	HOG+SVM	SIFT
Precision	0.978	0.77
Recall	0.958	0.82
Accuracy	0.97	0.8

recall and accuracy.

Analysis:

The results assist in assessing the increase in performance of the HOG descriptor, when a large dataset is used for training the classifier. A change froms a smaller training set to a larger training set, while using the histogram of HOG output vector, certainly increases the evaluation parameters precision and recall. Comparison results shows that HOG descriptor performs well in Human detection than SIFT features.

6. Conclusion & Future Scope

To detect the people in static images HOG is an overall suitable feature descriptor, since it can describe an object without the need to detect smaller, individual parts of a person (i.e. the face). Additionally, the HOG features of an image are not affected by varying illumination conditions. In this project, an implementation of the Histogram of Oriented Gradients in static images has been described. A dataset was built for training an SVM classifier. We have successfully implemented Human detection system on Beagleboard-xM DM3730 which is very useful and basic step in Automatic digital content management, Video surveillance, Driver assistance systems and can be used in many other applications. The test results showed that the detecting humans in the static images using HOG-SVM is quite satisfactory Possible extensions to this study include expanding the dataset used by the SVM with additional images. The amount of errors in the detection process can be further reduced by adding more training images. The use of pre-processing algorithms to eliminate or reduce the amount of noise in the images will also improve the detection process.

References

[1] Navneet Dalal , Bill Triggs, "Histograms of Oriented Gradients for Human Detection," in Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1, p.886-893, June 20-26, 2005.

[2] David G. Lowe, "Distinctive image features from scale-invariant keypoints,"

International journal of computer vision, 60, 2004.

[3] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," The 9th ICCV, Nice, France , volume 1, pages 734-741, 2003.

[4] Andriluka, M., Roth, S., & Schiele, B. (2008), "People-tracking-by-detection and people-detection-by-tracking," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1{8).

[5] V. de Poortere, J. Cant, B. Van den Bosch, J. de Prins, F. Fransens, and L. Van Gool., "Efficient pedestrian detection: a test case for svm based categorization," in Workshop on Cognitive Vision , 2002.

[6] R.C. Gonzalez and R. E, Woods, Digital Image Processing , Pearson Education, 3rd Edition, 2008.

[7] G. Bradski and A. Kaehler, Learning OpenCV , OReilly Publications,2008.

[8] BeagleBoard-xM Rev C System Reference Manual , http://beagleboard.org, Revision 1.0, April 4, 2010.

[9] N.Dalal, B.Triggs and C.Shmid, "Human detection using oriented histograms of flow and appearance," in proc European Conference on Computer Vision , 2006.

[10] www.ti.com

[11]M. Enzweiler and D.M. Gavrila, "A mixed generative-discriminative framework for pedestrian classification," IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2008.

[12]O.Tuzel, F.Porikli, and P.Meer, "Human detection via classification on Riemannian manifolds," IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2007.

[13]E. Osuna, R. Freund and F. Girosi, "An Improved Training Algorithm for Support Vector Machines," To appear in Proc. of IEEE NNSP'97 , Amelia Island, FL, 24-26 Sep., 1997.