Human Detection in RGB Images

Ish Rishabh
Department of Information and Computer Science
University of California, Irvine
Irvine, CA - 92612
irishabh@uci.edu

Arjun Satish
Department of Information and Computer Science
University of California, Irvine
Irvine, CA - 92612
asatish@uci.edu

Abstract

Human detection is a challenging classification problem which has many potential applications including monitoring pedestrian junctions, young children in school and old people in hospitals, and several security, surveillance and civilian applications. Various approaches have been proposed to solve this problem. We have studied and implemented a scheme using Histogram of Oriented Gradients (HOG), as part of CS273A Machine Learning coursework for Winter 2008 quarter. The results are encouraging and we intend to improve our implementation so as to integrate it with another ongoing project at UCI. The INRIA Person dataset [9] was used for training and testing the classifier.

1 Introduction

1.1 E2E systems

Our motivation for detecting the presence of human beings in images originates from its necessity in automated systems like Environment to Environment (E2E) communication system. E2E systems are futuristic communication systems that connect an environment to another environment and provide automation for many functionalities. Hence, if a user is present in an environment, the status of the user is automatically updated, based upon which several scheduled tasks can be performed. The presence of user can be sensed through optical sensors like cameras. There may be several cameras placed around in the environment in order to sense the presence of a person.

1.2 Need for human detection

Although many other detection methods can be used for detecting human presence, like audio sensing, face detection, blob detection, motion detection and so on, there are distinct

reasons because of which a detector specific for human detection is needed.

- A person may not be facing the camera. Hence detectors that use features for face detection may fail in such scenarios.
- A person may be sitting without significant movement and so motion based detectors may fail. This rules out using multitemporal images for detection.
- Techniques like blob detection cannot be used because if a non-human object appears/disappears from the scene, blob detector may give false alarms.
- Audio sensing will not be reliable because we cannot assume that the user will continuously make sound.

This report is organized as follows. Section 2 provides a literature survey of important work in this field. Detailed explanation of our implementation is provided in Section 3. Section 4 summarizes the results.

2 Previous work

Previously, this problem has been tackled by many different approaches. In 1999, David Lowe [1] presented the theory of SIFT which enabled us to detect and describe a range of local features to help in object recognition. The SIFT features are local and based on the appearance of the object at particular interest points, and are invariant to image scale and rotation. This was later improved by him in [8]. In 2000, Papageorgiou et. al. [2] came up with a Haar wavelet based feature space and Support Vector Machines for classifying test input images for detecting objects. There has been much work related to this, improving on its results. More recently, Mikolajczyk et al [3] worked on human detection by modeling it as a fixed assembly or parts. an object is classified as human if it has "two hands, a torso and a head visible in the correct alignment". Feature extraction and learning was done using the Integral Image and the Adaboost algorithm [4]. An advantage of this approach was the feasibility of using it in scenarios where people would be occluded, for example, in pedestrian scenes. Liebel et al [5] use a combination of local and global cues via probabilistic top-down segmentation to determine pedestrians in a complex scenarios. Dalal and Triggs [6] use a simple set of features, namely the Histogram of Oriented Gradients, to determine the location of humans in RGB images. They use these features to train a soft linear SVM classifier, which performs extremely well in classifying general RGB images for human detection. We find this technique very intuitive and hence, the use it in this project. Though this approach has been adopted to video streams [7], we would be exploring it with RGB images. Such histogram-based features have also been applied with deformable models to provide good results on the INRIA Person database[10].

3 Implementation

We consider an image $\mathbf{I}_{W\times H}$ to be a 2-dimensional array of pixels. There are W columns and H rows in each image. Each pixel is a triple comprising the RGB values of the color that it represents. Hence, the image has three color components: $\{I(R)_{W\times H},\ I(G)_{W\times H},\ I(B)_{W\times H}\}$.

3.1 Feature extraction: Histogram of Oriented Gradients

We have used Histogram of Oriented Gradients [6] as features for detecting human presence. Choosing gradient-based features makes the scheme robust to illumination variations whereas use of orientation information to define features provides robustness against contrast variations.

Basic idea behind these features is to divide an image into tiles called *cells* and then extract a *weighted* histogram of gradient orientations for each cell. Following subsections provide details of each step.

3.1.1 Defining multiple resolutions

Since there may be resolution difference between images used for training the classifier, and those of new target images, features should be extracted from an image at multiple levels of resolution. Resolution level is determined by a shrinkage factor γ ($\gamma = 0.95$) that defines the amount by which the image size is reduced in each dimension, as compared to the size in previous level. Thus, the size of an image in level l is $w(l) \times h(l)$ such that, $w(l) = \gamma^l W$ and $h(l) = \gamma^l H$. Features are extracted from cells at each resolution level from 0 to an upper limit L.
We represent an image at level l as $\mathbf{I}^l$ which comprises three color channels $I^l_R,\ I^l_G$ and I^l_B.

3.1.2 Gradient computation

For image at each level l, we determine a gradient image $\mathbf{G}^l$ as follows:
$\mathbf{G}^l = \{G^l_{mag}, G^l_\theta\}$, such that,
$G^l_{mag}(x,y) = \sqrt{(I^l_{c*}(x+1,y) - I^l_{c*}(x-1,y))^2 + (I^l_{c*}(x,y-1) - I^l_{c*}(x,y+1))^2}$
and $G^l_\theta(x,y) = \frac{\pi}{2} + \arctan\frac{I^l_{c*}(x,y-1) - I^l_{c*}(x,y+1)}{I^l_{c*}(x+1,y) - I^l_{c*}(x-1,y)}$, where $c* = \arg\min_{c\epsilon\{R,G,B\}} \sqrt{(I^l_c(x+1,y) - I^l_c(x-1,y))^2 + (I^l_c(x,y-1) - I^l_c(x,y+1))^2}$.
Here, (x,y) denotes the location of a pixel such that $1 < x < w(l)$ and $1 < y < h(l)$. Gradient values for all pixels at the boundary of the image are defined to be zero (both magnitude and orientation).

It should be noted that $G^l_{mag}(x,y)$ has only one component as it retains the maximum gradient magnitude value amongst all color components at pixel (x,y). Similarly, $G^l_\theta(x,y)$ retains the orientation value for that color component for pixel (x,y).

3.1.3 Computing histogram of gradient orientations

Gradient orientation values lie between $[0\ \pi)$. This range can be discretized into 9 *bins* of size $\frac{\pi}{9}$ each. Now, the image at each level is divided into non-overlapping *cells* of size $p \times p$. For each cell (cx, cy), we compute a 9 element array $f(cx,cy)[\ ]$. Each of the elements corresponds to one of the *bins* in which the orientation of a pixel in a cell falls. Thus, each pixel is said to vote for one of the bins in the histogram. This vote is weighted by the magnitude of the gradient at that pixel. The following equation shows this:

$$f(cx,cy)[b] = \sum_{x=cx\times p+1}^{(cx+1)\times p} \sum_{y=cy\times p+1}^{(cy+1)\times p} [I\{\lceil \frac{G^l_\theta(x,y)}{\pi/9} \rceil = b\} \times G^l_{mag}(x,y)].$$

Here $1 \leq b \leq 9$ and $I\{.\}$ is the *identity* function.

We also define *energy* of a cell (cx, cy) as

$$E(cx,cy) = \sum_{x=cx\times p+1}^{(cx+1)\times p} \sum_{y=cy\times p+1}^{(cy+1)\times p} (G^l_{mag}(x,y))^2.$$

Now, in order to retain spatial information between neighboring cells, we append features of neighboring cells to each cell features and also normalize all these features with the sum of energies of all neighboring cells. This neighborhood of cells is called a *block*. We have considered 2×2 block such that the top, left and top-left cells are included in the

neighborhood of a cell. Hence, we define an overall HOG feature vector (at level l) for a cell (cx, cy) as follows:

$$HOG^l_{cell}(cx, cy)[] = \frac{[f(cx-1,cy-1)[] \mid f(cx,cy-1)[] \mid f(cx-1,cy)[] \mid f(cx,cy)[]]}{E(cx-1,cy-1)+E(cx,cy-1)+E(cx-1,cy)+E(cx,cy)}$$

Here, $\mid$ denotes *concatenation* of features. It should be noted here that the final HOG feature vector has a dimension of $4 \times 9 = 36$. It should also be noted that image at each level generates $(\lfloor \frac{w(l)}{p} \rfloor - 1) \times (\lfloor \frac{h(l)}{p} \rfloor - 1)$ number of cell features, as features from cells in the topmost row and in the leftmost column in image cannot be generated because their left and top neighbors are not defined.

3.2 Classifier

We have used soft-margin linear SVM with $C = 0.01$ as suggested in [11] to train our classifier. Here C is the *penalty* that we pay in allowing slack variables in a soft-margin SVM. 25 positive example images and 25 negative example images were used to generate the training set. The cell size was chosen as 8×8. Each image was of size 32×96 and only a single level of resolution ($L = 1$) was used to generate the training examples. Thus, each image generated 33 cell features. Overall, there were 825 positive vectors and 825 negative training vectors. 800 vectors were randomly selected from each set and then given for training to the SVM classifier. Hence, a total of 1600 training examples were used to train the classifier.

4 Results

We have used INRIA Person dataset [9] for training and testing our classifer. The results were tested on 500 images containing human beings and 450 other images that did not contain human beings. The size of each image was 32×96 and 33 feature vectors were generated from each image. Hence, there were 16500 positive examples and 14850 negative examples for testing the classifier.
We have calculated the accuracy using the number of correctly classified positive and negative *cell* vectors. Table 1 shows the accuracy of classification.

	Positive examples	**Negative examples**	**Overall**
Total	16500	14850	31350
Correctly classified	11337	9597	20934
Accuracy	68.7%	64.6%	66.78%
Error rate	31.3%	35.4%	33.22%

Table 1: Table showing classification accuracy

Figure 1 shows the classification results on various images with and without human beings. Cells that do not belong to *human* class have been painted in white. Cells along the top and left margin of the image have been blanked in the tested images as they cannot be classified because of the reason given in Section 3.1.3.

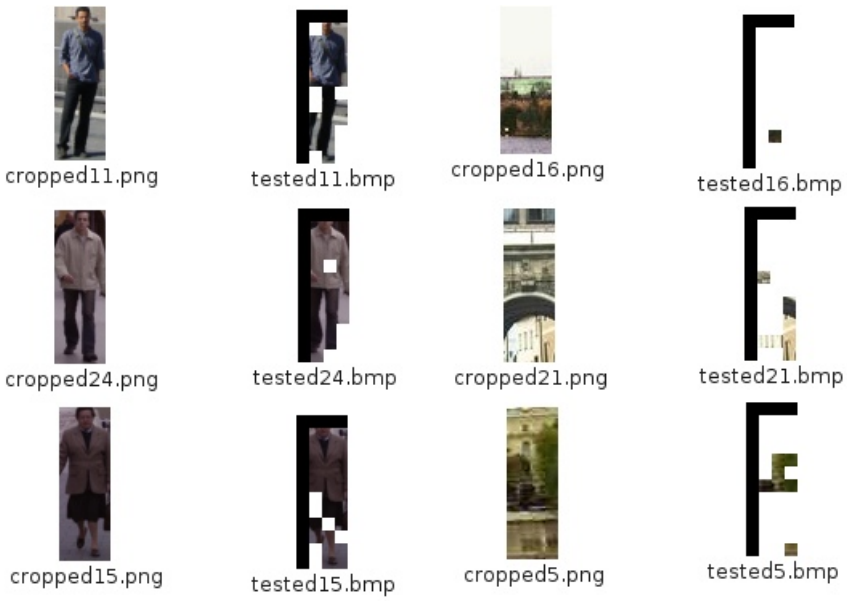

Figure 1: **Column 1:** Test images with human beings. **Column 2:** Classified images. **Column 3:** Test images without human beings. **Column 4:** Classified images.

5 Conclusion and future work

Histogram of Oriented Gradients features have been used to detect whether an image contains human beings or not. Soft margin SVM was used to train the classifier on the features. Though the accuracy is reasonable, there is still a lot of scope for improvement.

Going ahead, we would like to train a classifier for detecting human beings in *indoor* images as E2E applications are usually run indoors. We would also like to study the effect of different pose of humans in the images. In most of the training images, the people were standing upright. Also, in these images, full height of people was covered. We would like to study, for instance, the case when only the upper torso of a person is visible in an image.

6 Acknowledgements

We wish to thank Prof. Deva Ramanan for helping us with understanding and implementating the HOG scheme. We would also like to thank our colleagues in Experiential Systems Laboratory for their help.

7 References

[1] Lowe, D.G. (1999) Object recognition from local scale-invariant features.

[2] Papageorgiou, C. & and Poggio, T. (2000) A Trainable System for Object Detection *International Journal of Computer Vision.*

[3] Mikolajczyk, K. & Schmid, C. & Zisserman, A.(1995) Human Detection Based on a Probabilistic Assembly of Robust Part Detectors.*Computer Vision, ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004: Proceedings*.

[4] Viola, P. & Jones, M.(2001) Rapid object detection using a boosted cascade of simple features.

[5] Leibe, B. & Seemann, E. & Schiele, B.(2005) Pedestrian detection in crowded scenes *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference*.

[6] Dalal, N. & Triggs, B (2005) Histograms of oriented gradients for human detection *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference*.

[7] Dalal, N. & Triggs, B. & Schmid, C. (2006) Human detection using oriented histograms of flow and appearance. *European Conference on Computer Vision*.

[8] Lowe, D.G. (1999) Distinctive Image Features from Scale-Invariant Keypoints (2004) *International Journal of Computer Vision* .

[9] Dalal, N. (2005) INRIA Person Dataset *http://pascal.inrialpes.fr/data/human/* .

[10] Felzenszwalb, P. & McAllester, D. & Ramanan, D. *A Discriminatively Trained, Multiscale, Deformable Part Model*.

[11] Dalal, N. (2006) *Finding People in Images and Videos*. Ph.D. Thesis.