Pedestrian Detection Using Structured SVM

Wonhui Kim
Stanford University
Department of Electrical Engineering
wonhui@stanford.edu

Seungmin Lee
Stanford University
Department of Electrical Engineering
smlee729@stanford.edu

1. Introduction

With the advent of smart car and even driverless cars, the importance of intelligent driver's system has been rapidly growing. Accordingly, driver's vision has become one of the most popular issues and especially detecting some obstacles and pedestrians on the road are at the center of the vision problem in order to prevent accidents. Motivated by the importance of such topics, we implemented the human detector in the project.

Our approach is based on Deformable Part Model[1], one of the most powerful method for object detection. In the learning part of the system, we applied Structured SVM (SSVM) instead of Latent SVM (LSVM) which is used in [1]. The major goal of this project is not only to understand how differnt types of SVM can work on the detection problem but also to apply SSVM on our pedestrian detection problem.

Related Work

Dalal and Triggs[7] have developed the idea of histogram of gradient and have achieved excellent recognition rate of human detection in images. They used the concepts of HOG and designed a baseline classifier using a linear SVM. After a few years, Felzenszwalb *et al*. [1] introduced the deformable part model applying latent SVM. The performance of the detector has been significantly improved by combining HOG feature pyramid with deformable part model. Deformable part model has been applied in various papers and it is considered as the most general framework for object detection in 2D image. In [3], the concept of structured SVM first introduced, whose major difference from previous ones is the structured vector form of the output and the loss rescaling in the constraint inequality. Theoretical background of our project is mainly based on those works.

Overview

Our detection method is summarized in two phases: the learning phase and the detection phase.

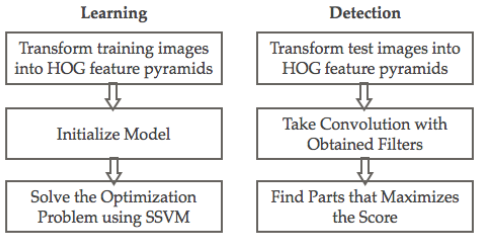

Figure 1. Overview block diagram.

During the whole procedure, we do not use raw images but instead use transformed images into HOG feature space. Since our approach is mainly based on the window scanning, HOG feature pyramid with different levels is constructed for each image.

The model mainly consists of root filter and part filters which should be obtained after the completion of the learning process. Root filter is trained with the coarse resolution whereas part filters are trained with finer resolution. Using the SSVM learning algorithm and given annotation files for the training data which contains ground-truth bounding box coordinates, the optimal root filter and part filters can be obtained.

With obtained root and part filters, the scores for each test image are calculated by taking the convolution between the filters and the HOG feature pyramid of an image. Then we will determine whether some parts of image are desired objects or not based on the given threshold. Finally, we will use precision-recall curve and average precision(AP) to evaluate our model.

2. Deformable Part Model

2.1. Learning

A model is defined as $(F_0, P_1, ..., P_n, b)$ which consists of a root filter F_0, n numbers of part models $P_1, ..., P_n$, and the real-valued bias term b. Each part model P_i consists of a part filter F_i, the location u_i relative to the root filter

and the coefficients vector d_i for evaluating the deformation cost. The ultimate goal of learning is to find the optimal parameters vector $w = (F_0, P_1, ..., P_n, b)$. More details of parameter settings are in [1]. The primal optimization problem of latent SVM with soft margin can be simplified as follows.

$$\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t. \forall i \in \{1, ..., m\} \quad \xi_i \geq 0,$$

$$y^{(i)} h_w(x^{(i)}) \geq 1 - \xi_i$$

where $y^{(i)} \in \{-1, 1\}$. It looks identical to the linear SVM but the difference is in the definition of hypothesis function which involves some latent variables.

$$h_w(x) = \max_{z \in Z(x)} w^T \phi(x, z)$$

$Z(x)$ denotes latent domain of valid placements for the root and part filters specified by x. Latent variable $z \in Z(x)$ contains latent information about the relative part locations and the whole configuration. $\Phi(x, z)$ is the concatenation of features for the root filter, part filters, deformation cost, and the constant 1 for the bias term.

2.2. Detection

A model obtained through training is defined as $(F_0, P_1, ..., P_n, b)$ where F_0 is a root filter, P_i is the i-th part model and b is a scalar bias term. Each part model consists of a part filter F_i, the relative location to the root filter u_i and the coefficient vector for evaluating a deformation cost for each possible placement of the part.

With learned filter parameters $(F_0, P_1, ..., P_n, b)$ and the feature pyramid which is calculated with a new input image x as a part of scanning window approach, we can now implement the detection by evaluating their dot product as a score. By thresholding the scores for all root filter locations, we can finally detect people in a test image.

2.3. Multi-Component Model

In practice, there are some variations in human images due to different poses, directions and actions. Therefore, defining only a single model for the human detector would prove ineffective for such images. Instead of constructing a single model, we split the whole positive training images into N groups according to aspect ratio, the ratio between the width and the height of bounding boxes, and then construct models for each group.

In this project, we trained models for both 1 component and 2 component cases since human images do not contain much variations compared to other objects.

3. Structured SVM

Both linear SVM and latent SVM are binary classifiers where target variable y is binary. Rather than predicting a binary label for each input, SSVM predicts structured responses. Given m training examples $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}$ for i=1,...m, where x denotes the feature vector extracted from i^{th} training image and y is the structured output, the SSVM optimizes w by minimizing a quadratic objective function subject to a set of linear inequality constraints:

$$\min_{w,\xi} \frac{1}{2}\|w\|^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i \quad s.t. \forall i \in \{1, ..., m\},$$

$$\forall y \in Y \backslash y_i : \langle w, \delta\Psi_i(y)\rangle \geq \triangle(y_i, y) - \xi_i$$

There are $m(|Y| - 1)$ numbers of inequalities where w is a parameter vector, $Y \backslash y_i$ is the set of possible outputs excluding y_i and ξ_i is a non-negative slack variable for i^{th} example. $\delta\Psi_i(y) = \Psi(x_i, y_i) - \Psi(x_i, y)$ where $\Psi(x, y)$ represents some combined feature representation of inputs and outputs and $\triangle(y_i, y)$ is a loss function.

With the parameter vector w obtained, we can make a prediction by maximizing the score function F over the response variable for a specific given input x as follows.

$$h_w(x) = \operatorname*{argmax}_{y \in Y} F(x, y; w)$$

$$where\ F(x, y; w) = \langle w, \Psi(x, y)\rangle$$

One of the major advantages of applying SSVM is tunability to specific loss functions. We call this margin rescaling, creating larger margins for the classes of most desirable misclassification. By defining the loss function $\triangle(y_i, y)$ as any appropriate form that is in general proportional to the dissimilarity between two outputs, we can give flexibility to the inequality thereby performing better training compared to the case when just using a constant loss.

4. Experiment

Before implementing our human detector, many things must be decided such as features, parameters, loss functions for SSVM, optimization algorithm, training dataset, and so on. In this section, we focus more on practical issues such as how we specified some important variables and look into the details for the implementation.

4.1. Data Selection

PASCAL VOC 2012 dataset which contains 23,080 training examples with complete annotations is used for the project. For each image, xml file is provided, which contains class label and ground-truth bounding box positions but not for part locations.

It also has images for testing but does not contain a complete set of data labelling files, so we evaluated our system using 9,963 test images from PASCAL VOC 2007 dataset.

4.2. Feature Selection: HOG Descriptor

The accuracy and the effectiveness of HOG feature for human detection has been verified in many works and is currently accepted as one of the most appropriate and generalized feature. We also use HOG feature for this project. More details are in [7].

When applying HOG feature transformation to an image, we obtain 36 transformed images for each gradient orientation and normalization. Due to the window scanning approach applied in this project, we use the concept of HOG feature pyramid instead. With various resolutions of an image, we calculate HOG features for each level of resolution.

4.3. Parameters Setting

Before applying SSVM algorithm to our problem, input and output vectors and also the parameters vector should be clarified. The two most distinctive characteristics of SSVM is the flexibility in choosing loss functions and the form of outputs which can be a structured vector form. We choose the output vector as $y = (y_l, y_b)$ where $y^l \in \{1, -1\}$ and y^b is a four dimensional bounding box labels vector.

4.4. The Loss Function

We can simply choose the loss function for this y and its ground-truth vector $y^{(i)}$ as follows.

$$\triangle(y, y^{(i)}) = 1 - o(y_b, y_b^{(i)}) \qquad if \quad y_l = y_l^{(i)} = 1$$

where $o(y_b, y_b^{(i)})$ is the fraction of overlap area defined by two vectors. Overlap area is computed by dividing the area of intersection as the union of two vectors.

As a possible output y approaches the true output $y^{(i)}$, $o(y_b, y_b^{(i)})$ gets larger thus the loss decreases. With lower values of loss, it would become easy to satisfy the constraint. Thus, the flexibility of the loss function increases the probability to satisfy the constraint for outputs that are close to the ground-truth output. On the other hand, if y and $y^{(i)}$ do not overlap, the loss is maximized with value 1. Since the bound of inequality constraint is more strict, it becomes more likely to violate the constraint. Thus, the flexibility of the loss function decreases the probability to satisfy the constraint for outputs that are far from the ground-truth output.

For negative images, y does not have ground-truth bounding boxes, hence we just use 1 for the loss. In the code implementation, we do not face with the case where y_l and $y_l^{(i)}$ are different, so we do not have to define the loss more in specific for such cases.

4.5. Implementation Detail

As in [1], we train the SSVM using the coordinate descent approach together with the datamining and gradient descent algorithm.

When performing the coordinate descent, we first find the latent variable $z^{(i)}$ for each training example, which maximizes the score calculated with the fixed parameter vector w. That is, for every m training examples, find root and part filter locations $z^{(i)}$ around $x^{(i)}$ having the maximum score.

$$z^{(i)} = \underset{z \in Z(x^{(i)})}{\text{argmax}}\, w^T \phi(x^{(i)}, z)$$

After the completion of the first step, we find the parameters vector w using the stochastic gradient descent. When updating the parameters vector, we can apply the loss function when checking the satisfiability of inequality constraints.

As for the negative training examples, it should be carefully chosen for the parameter update. Since there are a huge amount of possible negative examples, we perform datamining hard negatives which are relatively close to the score of positive examples so misclassified as positive but actually they are negative.

The real implementation does not exactly follow the definition of SSVM. For positive examples, instead of updating the parameter vectors for all the possible locations of bounding boxes, we first choose $z^{(i)}$, the latent variable which maximizes the score, with calculating its overlap area $o(z^{(i)}, y_b^{(i)})$ because otherwise the computation is too expensive. Then, we perform learning with such optimal $z^{(i)}$ by checking whether the constraint satisfied or not according to the value of $o(z^{(i)}, y_b^{(i)})$.

5. Results

5.1. Trained Model

We constructed both one and two component models for human detector. Figure 2 and figure 3 shows the resulting models learned on PASCAL VOC 2012 dataset. 6 parts are used when learning, but we can see that one of them does not take the important role in the person model.

5.2. Evaluation

Since PASCAL VOC 2012 dataset does not contain a complete set of data labellings, we evaluated our system using 9,963 test images from PASCAL VOC 2007 dataset. The dataset specify ground-truth bounding boxes of human location for each image. Based on the calculated scores for each image, we can evaluate the precision and the recall. The precision is the number of true positives divided by the total number of elements labeled as belonging to the positive class, and the recall is the number of true positives divided by the total number of elements that actually belong to the positive class. In other words, the precision is the fraction of the reported bounding boxes that are correct

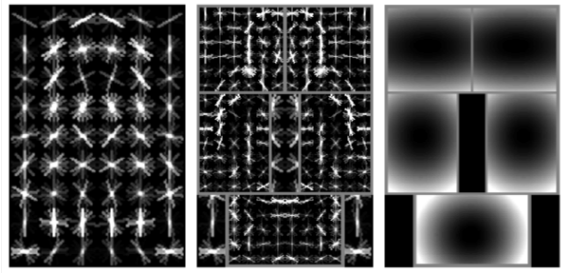

Figure 2. 1-component person model obtained by SSVM. This single model cannot generalize various shapes of person in random test images.

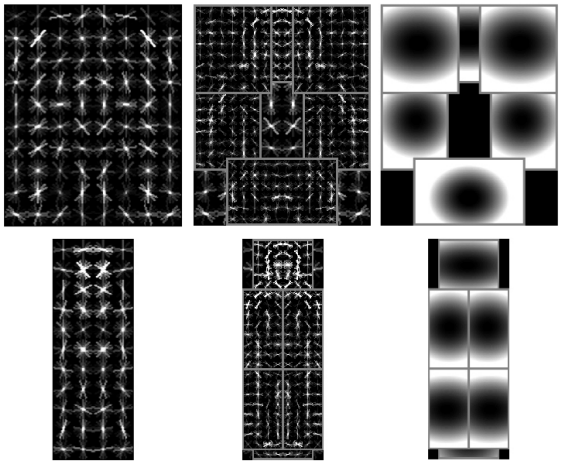

Figure 3. 2-component person model obtained by SSVM. Upper model showing fat-shaped person seems to represent those who are sitting down. Bottom model shows normal shape of person.

detections, while recall is the fraction of the objects found correctly.

Figure 4 shows the precision/recall curves for four different learning procedure. As we can see from the figure, the curves resulting from SSVM is closer to the top right side of the plot than those from LSVM. More specific values are in figure 5.

Depending on the determination method for final bounding boxes, we have two kind of measures. While Test1 column indicates the result only considered root location, Test2 column shows the result considered both root and part locations. Average precisions(AP) resulting from SSVM learning are greater than those from LSVM learning for both 1-component and 2-component models. Especially as for the 2-component model, we have achieved roughly 4% of the performance improvement compared to the original human detector based on LSVM learning algorithm.

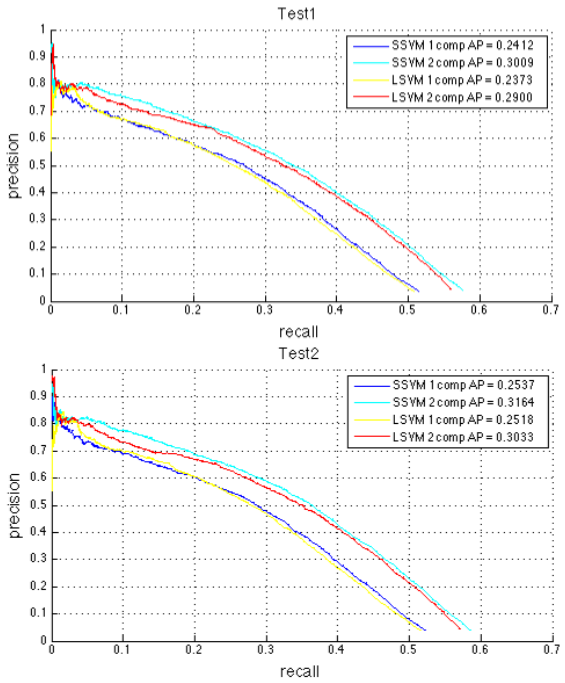

Figure 4. Comparison of precision/recall curves between the models obtained using LSVM and SSVM. The more the curve is placed on the top right side of the plane, the better the detector is.

# Components	N = 1		N = 2	
AP	Test1	Test2	Test1	Test2
LSVM	0.2373	0.2518	0.2900	0.3033
SSVM	0.2412	0.2537	0.3009	0.3164
Improvement	+1.62%	+0.763%	+3.76%	+4.32%

Figure 5. Comparison of LSVM and SSVM with average precisions. N is the number of components of the model, and Test1 and Test2 use slightly different algorithm to decide the final bounding boxes position. Regardless of which test used, SSVM always gives better performance than LSVM.

6. Conclusion

In this project, we have applied SSVM to the human detection problem based on deformable part model. The optimization problem for SSVM is solved though coordinate descent technique, which involves datamining hard negatives and stochastic gradient descent. As a result, we achieved roughly 4% of performance improvement for the 2-component human detector. That is because SSVM gives flexibility to choose the loss function, making the constraint tough for examples far from the ground-truth one while make it loose for those close to the ground-truth vector thus raising the possibility to satisfy the constraint.

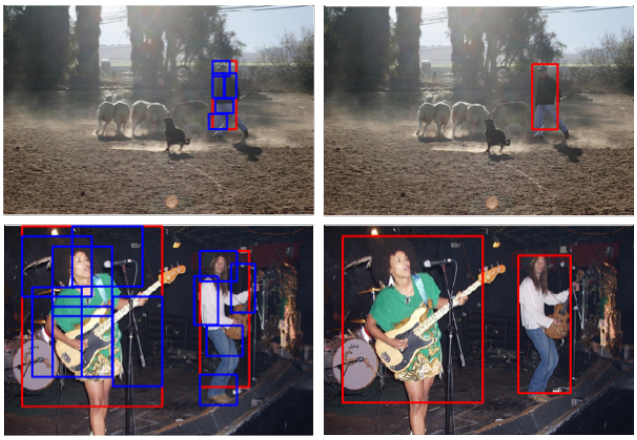

Figure 6. Detection examples using deformable part model learned by SSVM. It is actually hard to detect the difference between resulting images from LSVM and SSVM through the eye, SSVM shows better performance. Images on the left column shows root bounding boxes together with each part, and images on the right column shows the final bounding boxes.

7. Future Work

Due to the heavy load of computation, especially during the learning procedure, we only performed the detection for human. However, our approach can be generalized for other object classes as well, such as vehicles, animals and so forth. In such a case, however, another type of features instead of HOG feature might have to be chosen.

In addition, it would be possible to apply the complete SSVM learning which exactly follows the optimization problem mentioned in section 3 in the implemenation procedure. In this project we only focused on the loss rescaling of SSVM and defined the loss function in a really simple form, but there might be more effective way to design the structured output vector y at the expense of heavier computation. So we are now planning to design the learning parameters in a different way that leads to better performance.

References

[1] P. Felzenszwalb, D. McAllester, D. Ramanan, "A discriminatively trained, multiscale, deformable part model". CVPR, 2008.

[2] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, "Object detection with discriminatively trained part based models". PAMI, 2010.

[3] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, "Large margin methods for structured and interdependent output variables". JMLR, 2005.

[4] B. Pepik, M. Stark, P. Gehler and B. Schiele, "Teaching 3D geometry to deformable part models". CVPR, 2012.

[5] M. Blaschko, C. Lampert, "Learning to localize objects with structured output regression". ECCV, 2008.

[6] , C. Desai, D. Ramanan, C. Fowlkes, "Discriminative models for multi-class object layout". ICCV, 2009.

[7] , N. Dalal, B. Triggs, "Histogram of Oriented Gradients for Human Detection". CVPR, 2005.