Human Detection with HoG Algorithm

Kyuseok Lee
University of Michigan
eekslee@umich.edu

Abstract—In this project, based on Histograms of Oriented Gradients (HOG) feature extraction algorithm, the DSP system for imager system detecting humans for autonomous vehicle will be implementing. This object-of-interest (OOI) imaging system provide image information with wide field of view, which could cover the field of view where driver could recognized. By adapting this technique, our system will provide human detection result and then warn to driver. As the result, this system could help to decrease the pedestrian fatalities.

Keywords—HoG feature extraction algorithm

I. INTRODUCTION

In 2012, 4,743 pedestrians were killed and an estimated 76,000 were injured in traffic accident in the United States (Figure 1) [1]. As showed figure 2, the most of accident were caused when pedestrian were dashing or standing at outside of driver's field of view. The possible solution is to install the camera system on top of vehicles, which has a human detection function with wide field of view. By adapting this imager system, we could mitigate pedestrian's fatalities. Recently, the most popular human detection algorithm is the HOG with Support Vector Machines (SVM) approach. The advantages of HOG with SVM is upholding invariance to geometric and photometric transformations [2]. We can enjoy this property by adapting HoG feature extraction algorithm on our target system. The paper is organized as follows: section 2 explain the HOG feature extraction algorithm. Section 3 describe the principle of SVM and the process the classifying the new data based on SVM model. Section 4 reports the simulation result and analyze the wrong decision result.
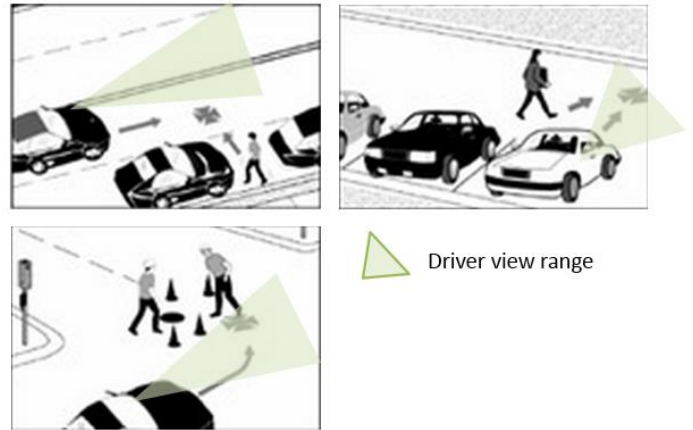

Fig. 2. The pedestrians fatalities case

II. HoG EXTRATION

A. Gradient Histograms

The gradient histogram use the data set [3]. That images set consist with images size 64 pixels width by 128 pixels height. To extract the gradient histograms, we set 8x8 cells within images. Figure 3 shows 8x8 cell drawn in red rectangle.

Fig. 3. Gradient histograms cell size in test image

A gradient vector have to be computed for every pixel on the image by measuring of changing in pixel values along the x-direction and the y-direction around target pixel. Figure 4 shows around pixels value of target pixel.
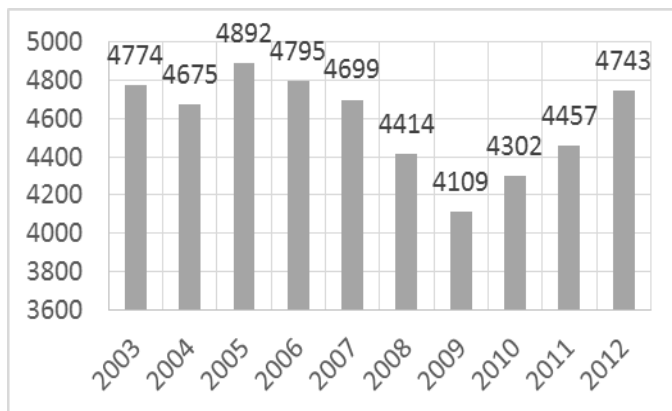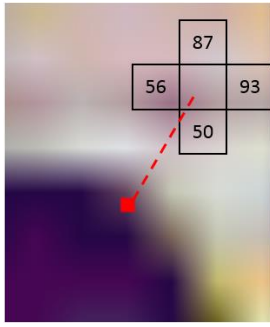

Fig. 1. The number of pedestrians fatalities

Fig. 4. Neighboring pixel's values

Based on these around pixel values, we can easily acquire the magnitude and angle of target pixel. The each magnitude and angle of pixel be expressed as

$$\Delta x = 93\text{-}56 = 37$$
$$\Delta y = 87\text{-}50 = 37$$

$$\text{Magnitude} = \sqrt{\Delta x^2 + \Delta y^2} = 52.3259$$
$$\text{Angle} = \arctan\left(\frac{\Delta y}{\Delta x}\right) = 45° \qquad (1)$$

Within the 8x8 cell, 64 gradient, vectors can be computed and be plotted result into a 9-bin histogram. The histogram ranges from 0 to 180 degrees instead of 0 to 360. These reduced ranges could increase contribution of magnitude of each vector. As the result, we can easily emphasis the properties of HOG, which could be helpful when SVM find optimal hyperplane.

B. Result

Figure 5 shows the HOG result from test image's cell. X-axis indicate the degrees of vector and Y-axis represent the normalized histogram result. For example, the significant gradients component is 90 degrees in figure 5, which mean that y-direction's contrast is major trend in this cell.
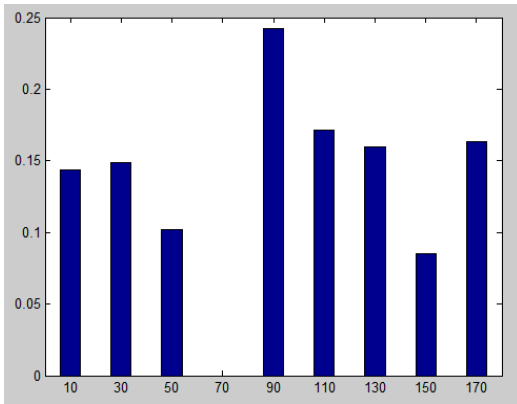

Fig. 5. Histogram of oriented gradients

Figure 6 show whole result of HOG from each cells. In the plot, the histogram of left plots concentrate on 90 degrees because the computation of left cell include valueless point on left side of image. However, this plot could not represent relationship between real image and HOG result. Before SVM

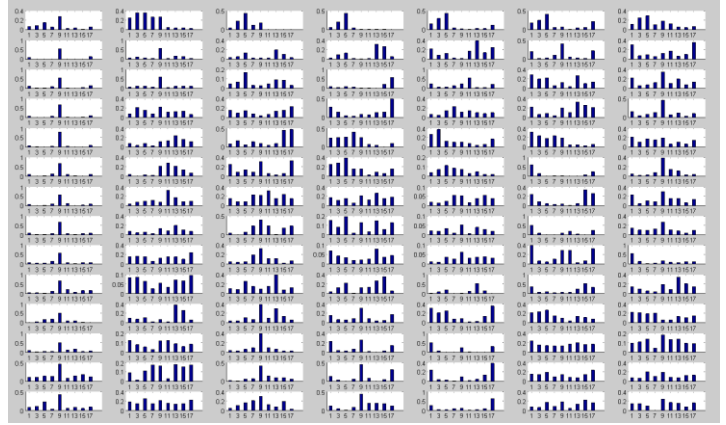training, we need to check the HoG feature extraction result which correspond the raw image.

Fig. 6. Histogram of oriented gradients of full images

Figure 7 show the HOG feature visualization. In the HOG feature visulation, the shape of HOG feature is changed in the region of legs and arms and head of human. Figure 8 show the HOG feature visualization with matlab function. Comparing between two images, the region of legs and arms and head of human show simularity.

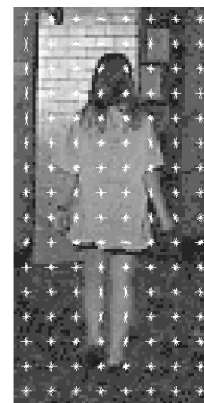Fig. 7. The HOG feature visualization

Fig. 8. The HOG feature visualization from matlab function

III. SUPPORT VECROR MACHINE (SVM)

A. Support Vector Machines

The Support Vector Machine (SVM) was developed for machine learning model with associated learning algorithms that recognize patterns, using for classification data [4]. The basic operation of SVM learning algorithm build a model which categorize examples and assigns new examples into one category or the other based on non-probabilistic binary linear classifier. The SVM model represent examples as points in vector space to separate categories for dividing by a clear gap as wide as possible. The new input data is mapped into that same vector space and predicted to belong to a category based on which side of the gap they fall on. The SVM constructs a hyperplane or set of hyperplanes in vector space, which can be used for classification. To achieve great separation, the hyperplanes have the largest distance to the nearest training example data point, since the larger the margin lead the lower the generation error of the classifier. Margin means the maximal width of the slab parallel to the hyperplane that has no interior data points. Figure 9 show optimal hyperplane and optimal margin in vector space. In this figure, the optimal hyperplane was determined to maxmize the optimal margin for retaining more accurate classification result.
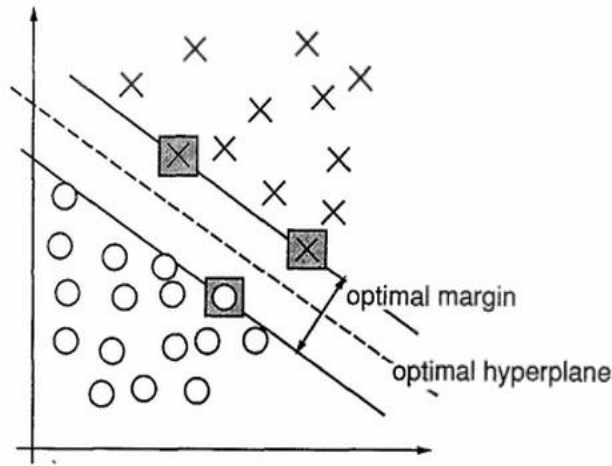

Fig. 9. Example of optimal hyperplane and optimal margin in a 2 dimensional space [4]

With optimal hyperplane, the SVM could classify a new data in vector space. Figure 10 show how SVM classify a new data. The new data (green dot) could be categorized as -1 in this figure since the new data's vector is mapped in the inside of the hyperplane. If the new data's vector is mapped the outside of the hyperplane, the new data could be categorized as +1.
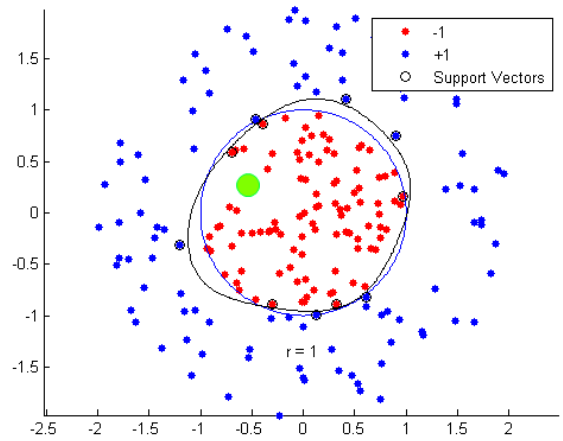

Fig. 10. Example of classfing new data with hyperplane

To improve the performance of SVM, we can adapt a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature space. However we tried to use another kernel trick, we could not achieve any improvement.

IV. SIMULATION RESULT AND ANALYING RESULTS

A. Simulation result and discussion

To train the SVM, we used the 1900 human image samples and 800 non-human image samples. After training the SVM, we test 393 test images that include 250 human images and 143 non-human images with linear hyperplane classification. In this test, we could achieve 0.8% detection error rate with human image samples and 6.3% detection error rate with non-human image samples. The higher detection error rate with non-human image sample is some pattern that have similar pattern like human body cause this problem. Figure 11 show wrong decision images.

Fig. 11. Example wrong decision images

A red circle mean SVM recognize that image have human image in figure 11. In previous HOG result, the result of histogrem show unique feature in legs, arms, and head of human. In left image, we can obseve the window pattern resemble human's leg and human's arm, which might lead the wrong decision. In right human image, the difference of brightness of human body and background is small so SVM might not detect the human in that test image.

Fig. 12. Example wrong decision image

However, as showed figure 12, some wrong decision image did not provide any siginificant clue for guessing reasons. If we gather more sample images for SVM training, we could reduce the detection error rate.

B. DSP tools

We used the 2D convolution to extract magnitude and angel for HOG histogrem. To classify human image, we used SVM and HOG for this project.

REFERENCES

[1] "Traffic safety Facts 2012 Data": http://www-nrd.nhtsa.dot.gov/Pubs/ 811888.pdf

[2] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.

[3] http://pascal.inrialpes.fr/data/human/

[4] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.