Vehicle Detection

COMP 9517 Project – Individual (z5151812 Haonan Zhang)

I. Introduction and Background

A. Understanding of Task

Vehicle detection and statistics are an important component of computer vision. With the popular installation of traffic surveillance cameras, the use of computer vision for intelligent recognization and analysis of vehicles becomes very effective. Vehicle detection takes images as input, analyzes through a series of algorithms, and outputs labelled images with bounding boxes around the recognized vehicles.

B. Understanding of Dataset

The complexity of the dataset for this task is the video recorded on highways, vehicles with different relative distances, and variable traffic conditions. As for the size of datasets, the model training part using the training dataset which contains 1074 clips of 2s videos in 20 frames per second and 3222 annotated vehicles. Model evaluation part using the testing dataset which contains 269 clips of videos with the same format as training data.

C. Task Challenge

The challenge of this technology is the complexity of the camera scene. When the vehicle appears in the photo at a far perspective, the target object occupies a small pixel, and the detection accuracy is low. When the vehicle appears in the close perspective in the photo, the target objects occupy large pixels; this causes a great change in the size of the target object.

D. Literature Review of relevant techniques (Background)

According to previous work, computer vision-based vehicle object detection is divided into traditional machine vision methods and complex deep learning methods. Traditional methods are divided into three categories [1]: background subtraction [2], continuous video frame difference [3], optical flow method [4]. The deep convolutional networks (CNNs) has a strong ability to learn image features and performs well in the field of vehicle object detection.

E. My Solutions

In this article, I focus on the above issues to propose a viable solution, I apply the Histogram of Oriented Gradients to extract image features, select the support vector machine (SVM) as the classifier model to train the dataset.

II. Method (implementation)

A. Preprocessing of Dataset

1) Region of interest (ROI): Extract vehicle image regions from the whole training data set, using the 'bbox' message from the annotation file. Moving pixels to obtain all non-vehicle region images from the dataset, and randomly select equal numbers of images from non-vehicle ROI as negative samples for the classifier. Overall, I extracted 1074 vehicle ROI and 1074 non-vehicle ROI. Figure 1 shows the example of the extracted vehicle and non-vehicle ROI.

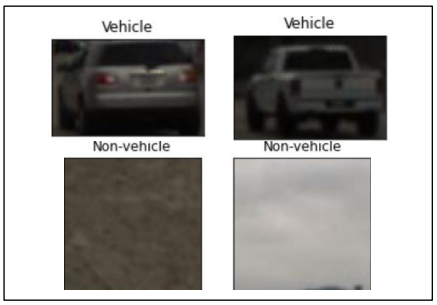

Fig. 1. Vehicle and non-vehicle ROI.

2) Resize dataset: Resize the shape of all data sets to 64 * 64 to ensure that the data sets are of the same size when they are input into the classifier for training. Figure2 below shows the comparison between the original size image and resized image.

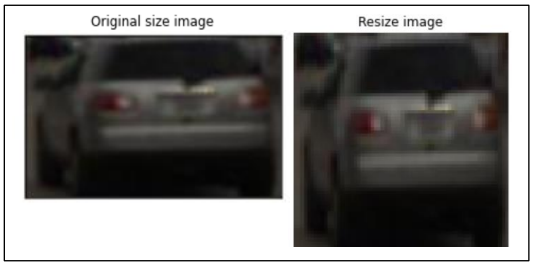

Fig. 2. Example of resize the image.

B. Feature Extraction

I select the histogram of oriented gradients(HOG) method to extract features. HOG is widely used in object detection in computer vision tasks. It describes the appearance and shape of local objects in an image through the distribution of intensity gradients or edge directions. The reason for choosing HOG is that it runs on local units, so it is invariant to geometric and photometric transformations except for the object directly. Rough spatial sampling, fine direction sampling, and powerful

local luminosity normalization function, as long as the objects maintain a rough shape, their movement can be ignored. Therefore, HOG is suitable for vehicle detection in images. Figure3 below shows the comparison between vehicle and non-vehicle image's HOG feature extraction.

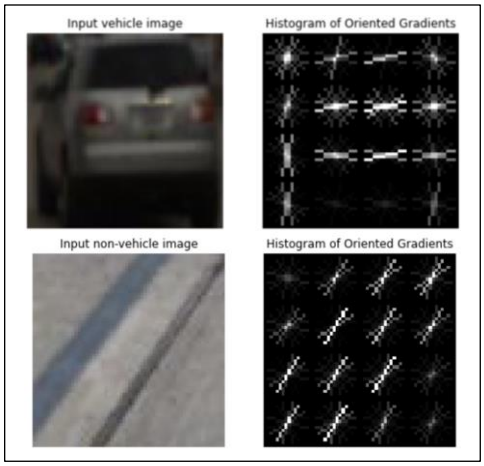

Fig. 3. HOG feature extraction.

C. Classifier

I select the support vector machine (SVM) as the classifier model to train the dataset [5]. The SVM is a supervised machine learning model that has a built-in hyperplane or a group of hyperplanes in a high-dimensional or infinite-dimensional space, which can be used for classification tasks. The latest SVM classifiers algorithms include sub-gradient descent and coordinate descent. When there are many training examples, the sub-gradient method is particularly effective; when the dimensionality of the feature space is high, the coordination is reduced. The sub-gradient descent algorithm (1) work with the following expression.

$$f(\mathbf{w}, b) = \left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i - b)\right)\right] + \lambda\|\mathbf{w}\|^2 \qquad (1)$$

D. Object Detection

For vehicle detection, I used sliding window technology, moving on the picture through a rectangular window, cutting the picture into small rectangular pictures. Then input the image captured by the sliding window one by one into the classifier, and let the classifier determine whether it is a vehicle. As the size of the vehicle in the images varies due to distance, I use variable-size sliding windows, in this way, sliding windows of different sizes can be used in different areas of the image to reduce misrecognition. Figure 4 given below are the display of using variable-size sliding windows.

Fig. 4. Recognize vehicle using variable-size sliding windows.

In order to solve the situation that a car is recognized by multiple windows in above figure 4, heat map technology is used to calculate the number of times the area is detected. After this technology, the vehicle can be recognized by one window (Figure 5).

Fig. 5. Using heat map technology to remove multiple windows.

III. Experiment

A. Dataset

The entire data set is divided into three parts: training set, test set, supplement set. Training and supplement set is used for model training, the test set is used for model evaluation. The format of the training dataset and testing are the same.

1) Sample selection: In my experiment, the 40th image of each clip in the training set is used to extract ROI. I only use the training set for model training, since my computer cannot afford to deal with a huge dataset. I extract 1074 vehicle ROI images as the positive sample for the classifier training. For the negative sample, I extract all the non-vehicle region images and randomly choose 1074 images as the negative training sample. The same amount of positive and negative ROI samples have been chosen to ensure the training accuracy.

2) Shuffle data and normalization: I tried shuffling data and normalize data before model training. The purpose of shuffle is to reduce variance and making sure that models remain general and overfit less; After many tries, it does improve the model precision a bit. Normalization is to bring all the variables to the same range, while in my experiment, apply this step does not change the results, the precision does not improve or reduce, so I drop this procession in my demo. This may because all the training dataset already resized to the same shape before training.

3) Split training data: I use the train_test_split function to make the split of training data and set test_size is set to 0.2 as normal.

B. Feature Extraction

As mentioned in the feature extraction method above (II Method), I have chosen the HOG method to extract features. However, many different feature extraction methods have been tried in my experiment. Local binary pattern(LBP) is a powerful feature for texture classification [6], which not suitable for this task. I tried HOG and Color Histogram, and also try to flatten data. After many tries, using HOG separately performs well enough, while combining it with the color histogram or flatten will reduce the performance. Details of tries gave below in table I.

TABLE I. CLASSIFIER MODEL PRECISION WITH DIFFERENT FEATURE EXTRACTION METHODS

Methods	HOG	HOG + Color Histogram	HOG + Flatten	HOG + Color Histogram + Flatten
Precision	0.98	0.80	0.96	0.97

C. Classifier

In this task, I try different predictive modelling approaches to training data, includes SVM, Decision Tree and Random Forest. All of them performs good, they can generate similar performance by adjusting their parameters. SVM and Random Forest performance a bit better than Decision Tree. I choose one test image to display the precision of model training and the detection results, details given below in Table II.

TABLE II. CLASSIFIER MODEL PRECISION WITH DIFFERENT FEATURE EXTRACTION METHODS

Methods	SVM	Random Forest	Decision Tree
Precision	0.98	0.98	0.95

D. Metrics

In this task, I take distance ($E_{distance}$) and overlap area ($E_{overlap}$) as two metrics. The metric $E_{distance}$ measures the distance between the truth center of test image bounding boxes and the predicted center of test image bounding boxes,

$$E_{distance} = \sqrt{\left(X_{gt} - X_p\right)^2 + \left(Y_{gt} - Y_p\right)^2} \quad (2)$$

Where X_{gt}, Y_{gt} refers to the truth center position of bounding boxes, X_p, Y_p refers to the predicted center position of bounding boxes.

The metric $E_{overlap}$ calculate the area of overlap between the predicted bounding box S_p and ground truth bounding box S_{gt} by the formula used in (3),

$$E_{overlap} = \frac{S_p \cap S_{gt}}{S_p \cup S_{gt}} \quad (3)$$

Where S_p refers to the area of the ground truth bounding box, S_{gt} refers to the area predicted bounding box.

The smaller the value of E, the lower the position offset value between the predicted bounding box and the ground truth bounding box; the larger the value of S, the closer the area of the predicted bounding box to the ground truth bounding box is; satisfied the trend of the two metrics at the same time represents the model detection more accuracy.

IV. RESULTS AND DISCUSSION

A. Results

For the results of vehicle detection, I chose three of my best results and three of the worst results to illustrate the strengths and weakness of my solution. As images are shown in Figure 6, my solution can detect vehicles well, when there are few vehicles in the image scene and the scene is simple. As shown in Figure 7, when the image scene is complex and there are many vehicles, my solution cannot accurately detect all vehicles.

Fig. 6. Three of my best results.

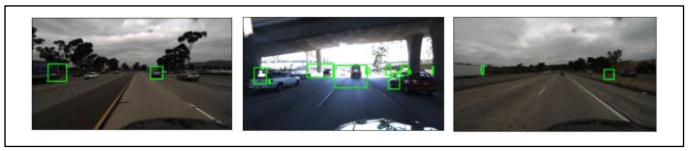

Fig. 7. Three of my best results.

I use distance and area overlap for evaluation. The following table III shows the result of distance and area overlap.

TABLE III. CLASSIFIER MODEL PRECISION WITH DIFFERENT FEATURE EXTRACTION METHODS

Distance	24
Area overlap	32

B. Discussion

I noticed that for images with fewer vehicles, my model can accurately detect vehicles; however, when one image appears in many vehicles, the model cannot accurately detect. This is because 1) Only the training set is used for model training, and 1074 vehicle data sets are not enough to train to a good detection effect. The larger supplementary set should be used for model training since it can provide more complex scenarios and samples, although this requires higher computer performance. 2) The detection technology is not perfect, although I already use variable-size sliding windows to adapt to different vehicle size, the results not good enough. Still should try to add some algorithms to reduce mismatch, like non-maximum suppression will make the detection more accurate.

REFERENCES

[1] Al-Smadi, M., Abdulrahim, K., Salam, R.A. (2016). Traffic surveillance: A review of vision based vehicle detection, recognition and tracking. International Journal of Applied Engineering Research, 11(1), 713–726.

[2] Radhakrishnan, M. (2013). Video object extraction by using background subtraction techniques for sports applications. Digital Image Processing, 5(9), 91–97.

[3] Qiu-Lin, L.I., & Jia-Feng, H.E. (2011). Vehicles detection based on three-frame-difference method and cross-entropy threshold method. Computer Engineering, 37(4), 172–174.

[4] Liu, Y., Yao, L., Shi, Q., Ding, J. (2014). Optical flow based urban road vehicle tracking. In 2013 Ninth International Conference on Computational Intelligence and Security. https://doi.org/10.1109/cis.2013.89. IEEE.

[5] Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks" (PDF). Machine Learning. 20 (3): 273–297. CiteSeerX 10.1.1.15.9362. doi:10.1007/BF00994018. S2CID 206787478.

[6] DC. He and L. Wang (1990), "Texture Unit, Texture Spectrum, And Texture Analysis", Geoscience and Remote Sensing, IEEE Transactions on, vol. 28, pp. 509 – 512