A PRIMARY TRAVELLING ASSISTANT SYSTEM OF BUS DETECTION AND RECOGNITION FOR VISUALLY IMPAIRED PEOPLE

Hangrong Pan[1], Chucai Yi[2] and Yingli Tian[1,2]
[1]Dept. of Electrical Engineering, The City College of New York
[2]Dept. of Computer Science, The Graduate Center
The City University of New York, USA
e-mail: {hpan05@ccny.cuny.edu, cyi@gc.cuny.edu, ytian@ccny.cuny.edu}

ABSTRACT

It is a challenging task for many blind and visually impaired passengers travelling by bus. To assist visually impaired passengers to travel more independently, we design a computer vision-based system to detect and recognize bus information from images captured by a camera at a bus stop. Our system is able to notify the visually impaired people in speech the information of the coming bus, and detect the route number and other related information which is depicted in the form of text. In bus detection, histogram of the oriented gradient (HOG) descriptor is employed to extract the image-based features of bus facade. Cascade SVM model is applied to train a bus classifier to detect the existence of bus facade in sliding windows. In bus route number recognition, we design a text detection algorithm on the basis of layout analysis and text feature learning, and then recognize the text codes from detected text regions for audio notification. This algorithm is able to compute the image regions containing text information. Experimental results demonstrate the effectiveness of our proposed algorithm for bus detection and route number recognition.

Index Terms—travelling assistance, sliding windows, cascade classifiers, linear SVM, text detection and recognition

1. INTRODUCTION

Public transportation plays an important role in the daily life of a city. For example, more than 55% of the people in New York City commute to work every day by public transportation system, including railway, subway, and buses. However, it is a challenging task for blind or visually impaired people to take the public transportation because they cannot perceive the coming bus and read the bus route number. In this paper, we present a primary framework of a camera-based bus detection and recognition system, as shown in Fig. 1. This system is able to assist visually impaired people to obtain the bus information at a bus stop. We define that bus detection is to find out the regions of a bus facade in scene image, and bus recognition is to recognize the bus route number from the detected bus facade.

Bus detection and recognition based on satellite signals [10] or wireless network communication [11] has been developed in some bus stations. However, this scheme requires pre-installation of the sensors and periodic maintenance. Thus vision-based technology will provide an alternative mean to detect and recognize the bus. It will significantly reduce the equipment expense and improve the efficiency and popularity of this travelling assistance application.

Fig. 1. Camera-based scene image, which is captured by a visually impaired user, to extract the information of bus.

Researchers in computer vision field have proposed some research work on bus detection and recognition for visually impaired person, such as vehicle categorization system in the highway, extraction and recognition of vehicle license plate. Yohida *et al.* proposed a recognition algorithm for different vehicles using computer graphic models [1]. In [2], a real-time classification system is designed for vehicles in different lanes of highway. The categorization is implemented by detecting the structure regions using horizontal edges, combined with regions mergence according to colors and positions of vehicles. Zhang *et al.* proposed a real-time face detection and recognition method [3]. Classification is achieved by the cascade verification

modules which are based on the face skin information and face structure features.

In this paper, we present a primary travelling assist system to help visually impaired people independently obtain information of the bus they are waiting. The system progressively extracts bus information from scene images, from bus facade to text region.

The remaining of the paper is organized as following. Section 2 presents an overview of our system. Section 3 describes the process of bus detection based on HOG descriptor and cascaded SVM model. Section 4 describes bus route number recognition based on text detection and recognition. Section 5 summarizes the descriptions of the datasets in our experiments and discusses the experimental results. Section 6 concludes the paper and introduces future work.

2. SYSTEM OVERVIEW

Our proposed system contains two main components, bus detection and bus route number recognition. Fig. 2 depicts the flowchart of systematic process. Bus detection applies sliding window and bus classifier to search for bus facade in camera-based scene images. The bus classifier is obtained from HOG-based feature extraction and Cascaded SVM learning model. Within the detected regions, we further perform bus recognition to extract the information of bus route number which is usually located on the top part of bus facade. In bus route number recognition process, a scene text extraction algorithm is designed to localize and recognize the text information.

In this paper, we present a demo system to evaluate the two components respectively. The performance of text-based bus recognition depends on the accuracy of bus facade regions extracted by bus detection. To retrieval accurate text information like bus route number, the camera-based scene image should have relatively high resolution and little motion blur.

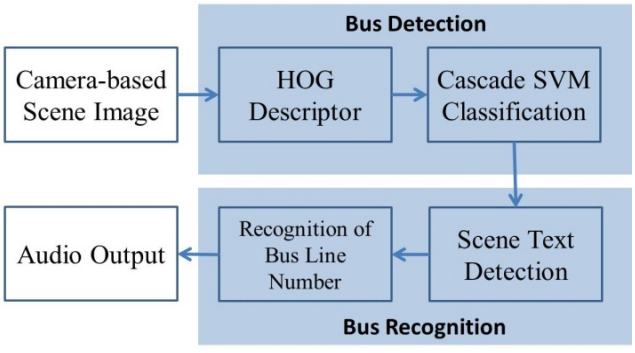

Fig. 2. The flowchart of our proposed travelling assistant system for bus detection and recognition.

3. BUS DETECTION

3.1 An Overview of Bus Detection

Bus detection is to find out the regions of city buses from scene images, which are captured by the visually impaired people waiting buses in the bus station. Since the system aims to help visually impaired people perceive the buses that are heading to them, we set the front part of the bus as our detection target.

The bus detection process can be divided into the following 5 steps:

(1) Preprocessing of camera-based scene images in bus station;

(2) Multi-scale sliding windows to extract sub windows as candidate bus facades in different positions and scales;

(3) Edge distribution evaluation to filter out the sub windows of background outliers, such as road and sky;

(4) HOG feature extraction from the obtained sub windows, and bus classifier is applied to detect the existence of a bus facade in the sub windows;

(5) Computing score map for bus facade detection.

We imitate the viewpoint of visually impaired people in bus station and collect natural scene images by a smart phone. Some preprocessing method is first applied to filter out some obvious background regions. In our experiments, the input images are down-sampled into a proper size which makes a trade-off between detection efficiency and accuracy. It ensures the image resolution while reduces the computational cost of bus facade detection. Fig. 3 illustrates the flowchart of our proposed bus detection algorithm, including edge-based background filtering, HOG feature extraction and learning, and score-based false positive filtering. Those sub-windows that go through the filters and the classifier will be processed in the next step.

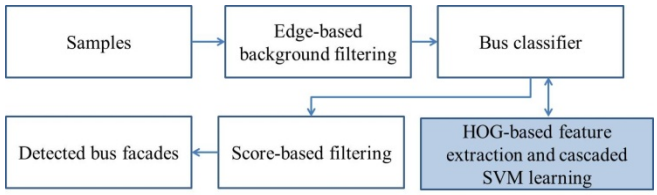

Fig. 3. The process of bus detection, including edge-based background filtering and score-based filtering. HOG-based feature is used to train a bus classifier in cascaded SVM model.

3.2 Sliding Window to Detect Candidates of Bus Facades

Sliding windows is a traditional method used in object detection. The purpose of sliding windows approach is to search the possible target in the image. In our system, the target is defined as bus facade. In the absence of the prior knowledge of the locations and scales of the target in the image, we carry out multi-scale scheme to obtain sub-windows in different positions and scales.

In our experiments, we set 4 window scales, 32×32, 48×48, 64×64, and 80×80. The original scene image is adaptively resized to ensure that the size of bus facades does not exceed 80×80. We slide the fixed size window in both horizontal and vertical directions. The sliding stride is set as 1/10 of the window size.

Each sub window corresponds to an image patch cropped from original scene image. A robust bus classifier is applied to detect whether the image patch contains the target bus facade or only background outliers. This bus classifier is learned from a training set of bus facades and background outliers.

3.3 Background Filtering Based on Edge Distribution

According to our observation, most camera-based scene images captured in bus station contain road and sky as background, which appears in smoother texture and less edge information than bus facade.

Thus we define a set of rules based on edge distribution to filter out some sub windows. In order to distinguish the target bus facade from the background, edge detection is performed on the sub-windows which are obtained from sliding window process.

In our experiment, canny edge detector is employed to generate the edge map of scene image, because it is able to detect both strong and weak edge in scene image. Then we count the number of edge pixels for each sub-window, and remove the sub-windows with less edge pixels than a pre-defined threshold. This threshold is set as the mean number of edge pixels in all sub-windows.

3.4 Feature Extraction and Learning based on HOG Descriptor

After edge-based background filtering, we obtain a set of sub-windows that probably contains bus facades. All these windows are normalized into size 80×80.

To detect the existence of bus facade, we learn a bus classifier from a training set, which contains image patches of bus facades as positive training samples, and image patches of background outliers as negative training samples. To model the appearance and structure of a bus facade and distinguish it from background outlier, HOG descriptor is extracted from all the training samples as the features of bus facades, as shown in Fig. 4.

HOG is a very popular feature descriptor that used for the purpose of object detection and recognition [4]. Local object appearance and shape information can be described by the distribution of intensity gradients or edge directions. In our system, HOG descriptor is chosen because it is able to generate very representative and discriminative information of the bus facade.

Based on HOG descriptor extraction, each sub-window is mapped into a feature vector, which is considered a point in feature space and prepared for feature classification. Then we perform feature learning and classification to obtain a robust bus classifier. The first task is to prepare a training set, in which bus facades are used as positive samples and background outliers are used as negative samples. The positive samples are collected from the image patches containing bus in camera-based scene image. The negative samples are collected from the image patches without bus information, and we make these negative samples diverse enough to represent different background outliers.

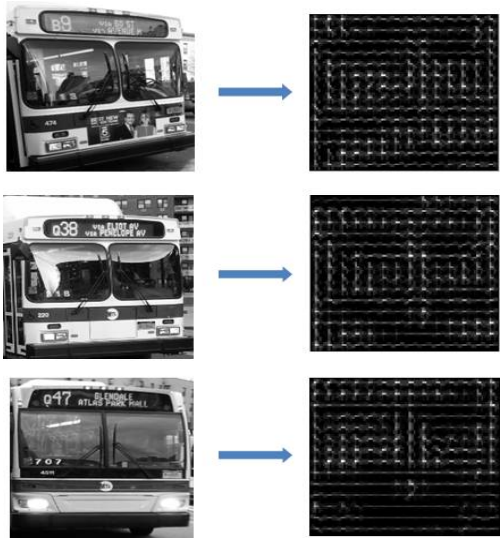

Fig. 4. Examples of HOG features extracted from patch windows of bus facades.

In this training set, the number of background outliers (negative samples) is far more than the number of bus facades (positive samples). To handle the imbalanced training process, we adopt cascade learning model, which divides the set of negative samples into several parts and performs the training stage-by-stage between all positive samples and a part of negative samples. The implementation is listed as the following steps.

(1) The negative set is divided into N parts, and each part has the same number of samples as the positive set. A cascaded classifier G is initialized as empty set;

(2) One part of negative samples is taken to be initial negative set, and we train an SVM classifier C_0 from the initial negative set and the positive set. Then the stage classifier is added into the cascade classifier as $G := G \cup C_0$;

(3) At the i-th stage, we select negative samples which are incorrectly classified by current cascade classifier G, and combine them into the updated negative set. If the number of incorrectly classified negative samples is less than that of positive samples, the whole process ends and the current cascaded classifier G is output.

(4) We train SVM classifier at current stage as C_i and add it into the cascade classifier as $G := G \cup C_i$;

(5) Set $i := i + 1$, and repeat the process from Step (3).

The cascaded classifier consists of several stage classifiers. Each stage classifier is trained from positive samples and the negative samples which cannot be correctly classified by previous stage classifiers. Give the cascaded classifier and a testing sample to be detected the existence of a bus facade, we input the testing sample into each of the stage classifiers. If all of them determines it as a positive sample, the testing sample is classified as a bus facade. If one of the stage classifiers detects it as a negative sample, the testing sample is classified as a non-bus facade.

3.5 Score Map to Refine the Location of Bus Facade Regions

For a testing sample that is classified as bus facade, we assign a score value, which is the mean of the SVM scores from all stage classifiers. According to our experiments, a higher score value leads to a higher probability of a bus facade. To further remove the false positive target bus facades, we discard the samples classified as positive but with a lower score than a predefined threshold.

In the original scene image, we restore the sub-windows corresponding to the samples that are classified as positive by the cascaded classifier. Then a score map is obtained by assigning the classification score to the sub-windows. The calculated score map is normalized into the range from 0 to 1. On the normalized score map, we then apply a score threshold to suppress the false positive bus detections. The threshold can be set from 0 to 0.5, as shown in Fig. 5.

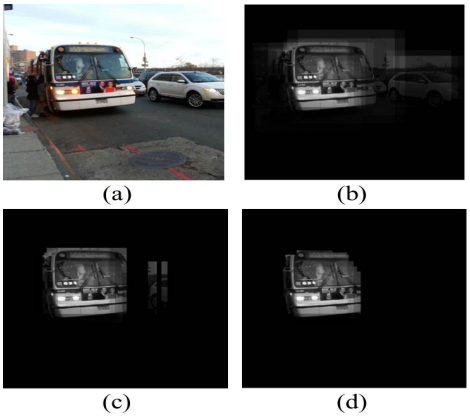

Fig. 5. (a) Original image; (b-d) Images after edge-based background filtering with score thresholds at 0, 0.25 and 0.5 respectively.

4. BUS INFORMATION RECOGNITION

To inform the visually impaired people of the status of a coming bus, we should extract text information from the detected bus facade. The text information usually presents the bus route number and other important notifications like service change or emergency notice. To effectively extract the text information for an audio output, we employ an algorithm of scene text extraction to localize and recognize the text information in cropped patch of bus facade.

Scene text extraction is divided into two main steps, scene text detection and scene text recognition. Scene text detection is to localize the image regions containing text characters and strings. Scene text recognition is to read text codes from the text regions, and prepare them for audio output.

According to our observations, text notification in bus facade is composed of digital dots, very different from the text in scan documents or object labels. A group of text features is defined to distinguish text information from non-text background outliers. Based on these text features, we perform layout analysis and text classification to detect text regions from the obtained bus facade region.

Text information in bus facade mostly appears in the form of text strings in horizontal alignment. So we can adopt the adjacent character grouping method in [7] to find out the possible text strings according to the layout organization. In this method of layout analysis, canny edge detection is first performed to compute candidate text characters in the form of bounding boxes of character boundaries. We remove the candidates in unsatisfied sizes, aspect ratios and locations. In the remaining candidates, we generate adjacent group by combining three consecutive text characters in horizontal alignment as shown in Fig. 6. Each adjacent group is considered as a sample, either positive sample (containing text) or negative sample (containing non-text background outliers).

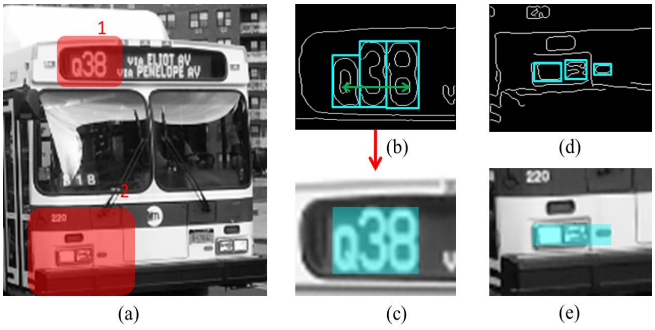

Fig. 6. The process of adjacent character grouping. (a) Original image of bus facade with text information. Adjacent groups are obtained in the two shaded regions. (b, d) Finding out three consecutive bounding boxes of candidate character boundaries in edge map. (c, e) An adjacent group in the form of candidate patch.

By applying the above method of layout analysis to natural scene image containing text, many adjacent groups are generated to compose a training set. In this training set, the number of negative samples is far more than that of positive samples. Thus we continue employ the cascaded learning model. The difference is that we apply an Adaboost learning model to generate the stage classifiers in learning a text classifier. This text classifier is able to detect whether an adjacent group obtained from layout analysis truly

contains text or not. The learning process is divided into two main steps as presented in [8], feature extraction and cascaded Adaboost learning. Text features are extracted by Haar-like filters from feature maps of edge distribution and stroke orientation. Then cascaded Adaboost model as in [6] is applied to learn a text classifier on the imbalanced training set. By using the text classifier, we can further remove the false positive text regions from layout analysis.

After scene text detection, the regions of text information in bus facade are localized. We crop out the text regions and transform them into readable text codes by optical character recognition (OCR) software. The recognized text codes are output by audio notification to the visually impaired people.

5. EXPERIMENTS

5.1 Dataset

Two datasets are involved in our experiments, which are bus dataset and text dataset respectively.

The bus dataset is self-collected. This dataset is used for training bus classifier and evaluating the accuracy of bus facade detection. It consists of camera-based natural scene images, captured in bus stations by a Samsung Galaxy III cell phone. The photos are taken when the buses are arriving at or leaving the bus station, so the captured buses in this dataset are not static but moving at a low speed. This dataset consists of two parts. The first part of dataset is used for training the bus classifier and text classifier. Both positive samples and negative samples are extracted from 20 original photos. Positive samples are windows patches obtained from the bus regions while the negative samples are selected randomly from non-bus regions. There are 131 positive sample patch windows and 328 negative samples used for training. Another part consists of 236 photos taken near or next to the bus stations to test the performance of the detection system. Number of images with bus (positive images) is 130 and number of images with no bus (negative images) is 106. The dataset includes different scales, view angles of the buses, also different lighting conditions but all of the images are captured in the daytime.

The text dataset is Robust Reading Dataset [9] used for the competition of scene text detection in the International Conference on Document Analysis and Recognition 2003 (ICDAR 2003). This dataset is used for training text classifier. It contains 509 natural scene images with multiple patterns of text information. On average 4 text regions exist in each image, and we use all these text regions as positive training samples to learn a robust text classifier.

5.2 Results and Discussions

To evaluate the performance of our proposed method, we carry out three experiments on our collected bus dataset. The first experiment is to detect whether a given scene image contains bus or not. First, we extract sub-windows by sliding window method and apply edge-based background filtering to remove the background sub-windows. Next, in the remaining sub-windows, we apply the bus facade bus classifier to detect the existence of bus facades. If more than 25% are classified as positive, we decide that the given scene image contains at least one bus. Otherwise, it does not contain a bus. By comparing with the ground truth information, the accuracy of bus existence is calculated and presented in Table 1. The detection accuracy of bus image is 81.48%, and that of non-bus image is 80.19%. The overall performance is calculated to be 80.93%.

Table 1. Accuracy of bus detection.

	Num of images	Correctly detected	Detection accuracy
Images with bus	130	106	81.48%
Images without bus	106	85	80.19%

In addition to detecting the bus existence, we further carry out the second experiment to evaluate the detection accuracy, by comparing the detected bus facade regions with the ground truth bus facade regions. These ground truth regions are manually annotated when collecting the bus dataset. In this experiment, we assume that a bus facade region is successfully detected if the ratio of its overlapping area with an annotated bus facade region is more than 30%. Table 2 shows the result when respectively applying thresholds 0 and 0.5 of the score map. Once we increase the threshold of the score map, the accuracy drops due to the decreasing of the passing area in the score map, as we can see from the example in Fig. 7. The possibility that the bus falls on the region of detected area also decreases.

Fig. 7. Some example images from the dataset. (a) Images with bus. (b) Images without bus.

Bus images ranging from 358×266 to 864×484 cost on average 5 to 20 seconds in bus facade detection. Some negative images containing little texture or structure resembling bus appearance could be quickly removed in the process of Cascade SVM classification.

Table 2. Detection accuracy in the positive images after applying score threshold to the score map.

Threshold	Number of images	Bus region detected	Accuracy
0	106	104	98.11%
0.5	106	79	74.53%

In the third experiment, scene text detection algorithm is applied to the scene images in bus dataset for extracting travel-assistant text information. This dataset contains 438 text regions of bus route numbers and destination addresses, according to our manual labeling. Currently, our demo system that is not fully functional detects 94 of the labeled text regions. The performance will be largely improved by generating high-resolution and deblurred bus images, and combining bus detection and recognition in an optimized way in future system. Our designed algorithm spends about 1 second on average in extracting text from a bus facade. Fig. 8 presents some example results of scene text detection in the bus facades. The cyan boxes represent the detected text regions. Thus text information is extracted to inform the visually impaired people of the bus route information.

Fig. 8. Some example results of scene text detection on bus facades. The cyan region represents the detected text regions.

6. CONCLUSION AND FUTURE WORK

In this paper, we have developed a primary image-based detection system to assist visually impaired people to independently travel by bus and obtain the route information at a bus station. The proposed system can achieve an accuracy of 80.93% in detecting the bus existence in a scene image. The accuracy of bus region detection can achieve 98.11% by setting the score threshold to be 0. An enhanced algorithm can be proposed to further increase the detection accuracy and also eliminate the falsely detected results during the detection.

We also have applied a scene text extraction algorithm to the detected bus facades, and successfully retrieve the text information of bus route number. Some pre-processing schemes will be employed to obtain high-resolution and deblurred bus images for more accurate text extraction.

The future work will include implementation of a real-time video based bus detection system and improve the accuracy of recognizing text information from the detected region. A user interface study and system evaluation by visually impaired users will also be conducted.

7. ACKNOWLEDGEMENT

This work was supported in part by NSF grant IIS-0957016, EFRI-1137172, NIH 1R21EY020990, FHWA grant DTFH61-12-H-00002, and Microsoft Research.

8. REFERENCES

[1] T. Yoshida, S. Mohottala, M. Kagesawa and K. Ikeuchi, "Vehicle Classification System with Local-Feature Based Algorithm Using CG Model Images", IEICE Trans. Information and Systems, vol.E85-D, No.11, pp. 1745-1752, Nov 2002.

[2] M. Shaoqing, L. Zhengguang, Z. Jun and W. Chen, "Real-time Vehicle Classification Method for Multiple-lane Road", IEEE International Conference On Industrial Electronics and Applications (ICIEA), pp. 960-964, May 2009.

[3] P. Zhang, "A Video-based Face Detection and Recognition System using Cascade Face Verification Modules", IEEE Applied Imagery Pattern Recognition Workshop, Washington DC, pp. 1-8, Oct. 2008.

[4] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", *Computer Vision and Pattern Recognition*, Vol. 1, pp. 886-893, Jun. 2005

[5] C. Chang and C. Lin, "LIBSVM: a Library for Support Vector Machine", 2001. http://www.csie.ntu.edu.tw/ ~cjlin/libsvm

[6] P. Viola and M. J. Jones, "Robust real-time face detection," In IJCV 57(2), pp. 137–154, 2004.

[7] C. Yi and Y. Tian, "Text String Detection from Natural Scenes by Structure-based Partition and Grouping", In *IEEE Transactions on Image Processing (TIP)*, Vol. 20, Issue 9, pp.2594-2605, 2011.

[8] C. Yi and Y. Tian, "Assistive Text Reading from Complex Background for Blind Persons", In *ICDAR Workshop on Camera-based Document Analysis and Recognition (CBDAR)*, Springer LNCS-7139, pp.15-28, 2011.

[9] http://algoval.essex.ac.uk/icdar/Datasets.html

[10] http://www.oregon.gov/ODOT/HWY/ITS/Pages/its_benefits_pub_trans.aspx

[11] H. Zhou, K. Hou, D. Zuo, and J. Li, "Intelligent Urban Public Transportation for Accessibility Dedicated to People with Disabilities," *Sensors*, Vol. 12, No. 8, pp. 10678-10692, 2012.