Bus Detection and recognition for visually impaired people

Hangrong Pan, Chucai Yi, and **Yingli Tian**

The City College of New York

The Graduate Center

The City University of New York

MAP4VIP

Outline

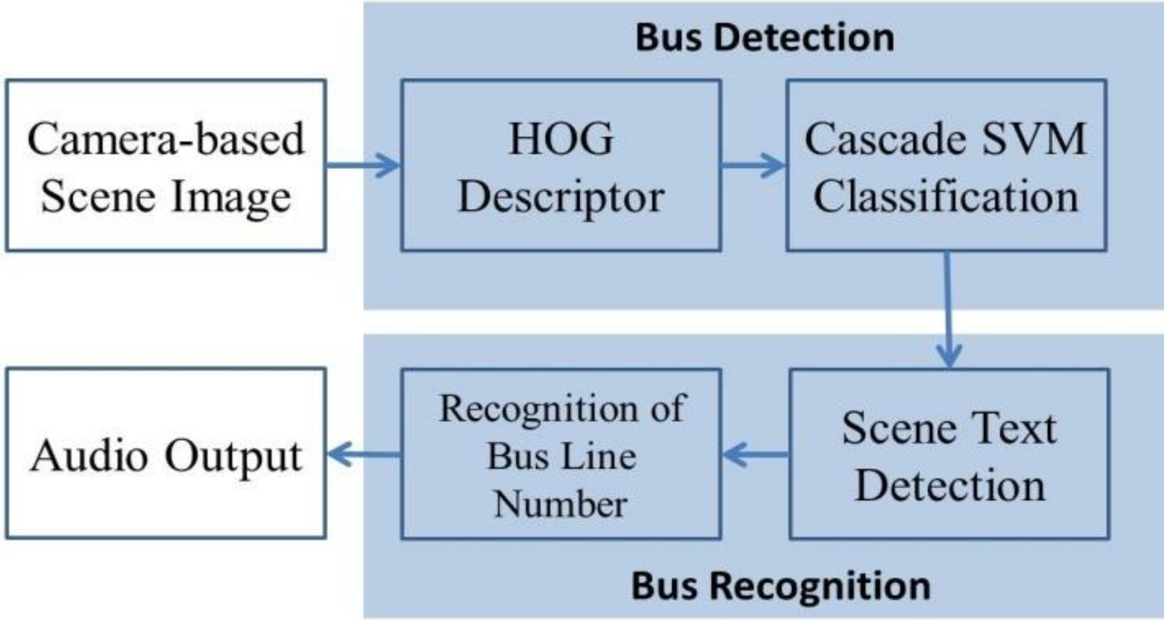
- **Motivation**
- **System Overview**
- **Bus Detection**
  - – Bus Candidate Detection
  - – Feature extraction for bus detection
  - – Cascade Support Vector Machine (SVM)
- **Bus Recognition**
  - – Text information extraction and recognition

Motivation

- **Enhance travel independence for blind or visually impaired people**
- **Assisting blind or visually impaired people to obtain bus information and get on the right bus at a bus stop**
- **Previous work**
  - Satellite signal, wireless network, etc.
  - Pre-installed devices and their maintenance
- **Vision-based travel assistance**
  - Low-cost and portable

System Overview

- **Bus Detection**
  - Localizing the region of bus in camera-based scene image at bus stop
- **Bus Recognition**
  - Extracting text information like bus route number
  - Detecting image regions containing text information, and recognizing text codes in the text regions
  - Audio output of text codes

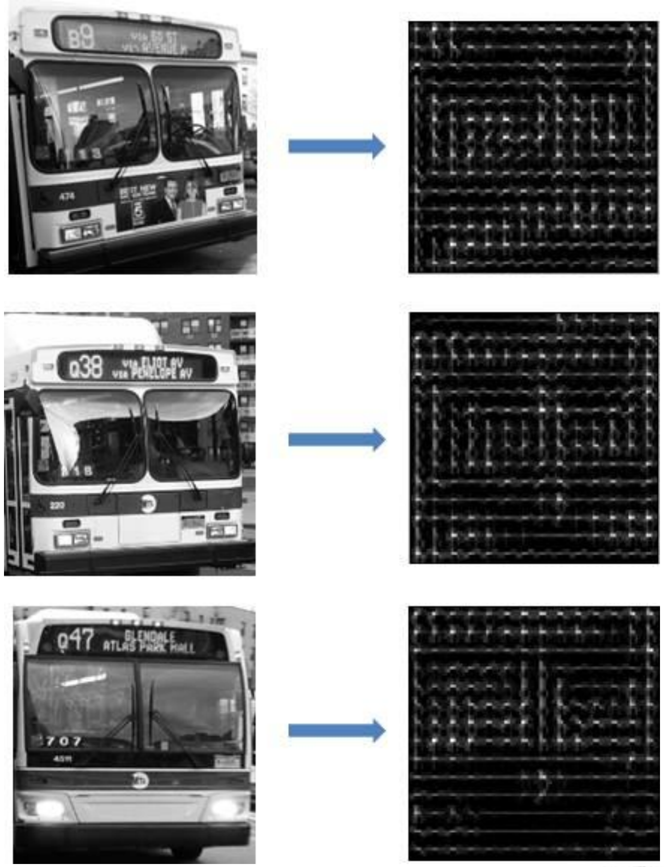


Bus Detection

- **Layout-based Processing**
  - Multi-scale sliding window to localize the candidate regions of bus
  - Background filtering to remove the candidate regions with low edge density
- **Bus Candidate Classification**
  - Feature extraction based on Histogram of Oriented Gradient (HOG) descriptor
  - Detecting bus based on Cascade SVM classifier

Bus Detection

- **Multi-scale Sliding Window**
  - 4 window scales, 32×32, 48×48, 64×64, and 80×80
- **Background Filtering**
  - Measure edge density on Canny edge map
  - Bus region should have higher edge density than background
- **Feature extraction**
  - Histogram of Oriented Gradient (HOG) Descriptor



Bus Detection

- **Detecting Bus**
  - The imbalanced numbers of positive and negative samples
  - Cascade SVM learning

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 $G := GUC_{0;}$

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.

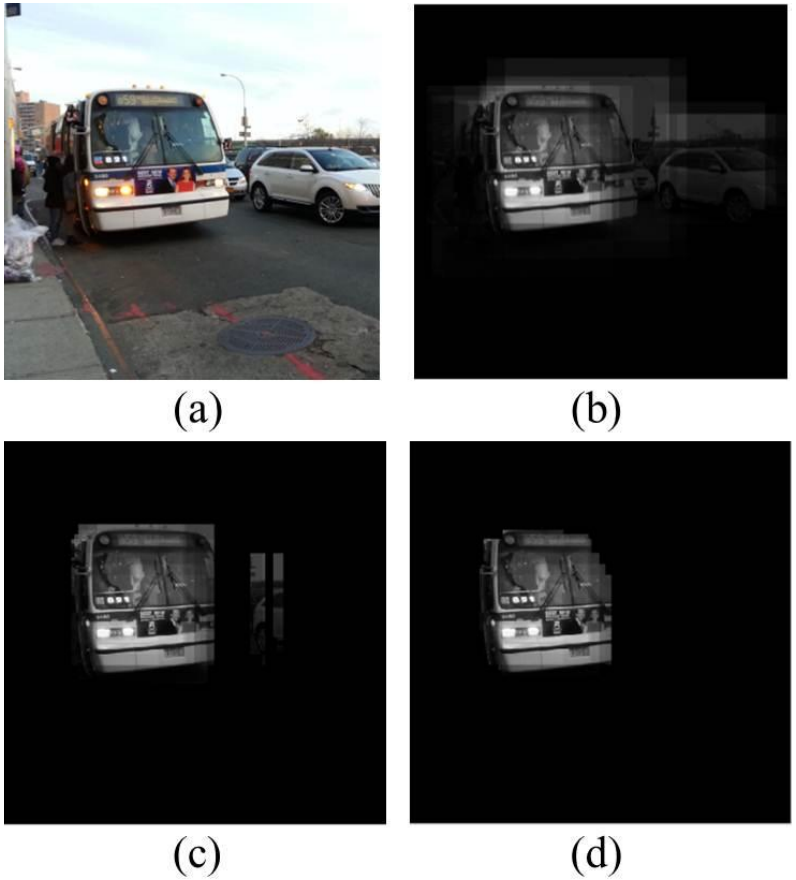
4. We train SVM classifier at current stage as C_i and add it into the cascade classifier as $G := GUC_{i;}$

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

Bus Detection Refinement

- **Each sliding window is assigned a score by Cascade SVM-based bus classifier**

- **Score Map to further filter out false positive bus detection and refine the location of bus candidate regions**
  - The right figure (b-d) shows the results of three different thresholds of score map at 0, 0.25 and 0.5



(a) (b)

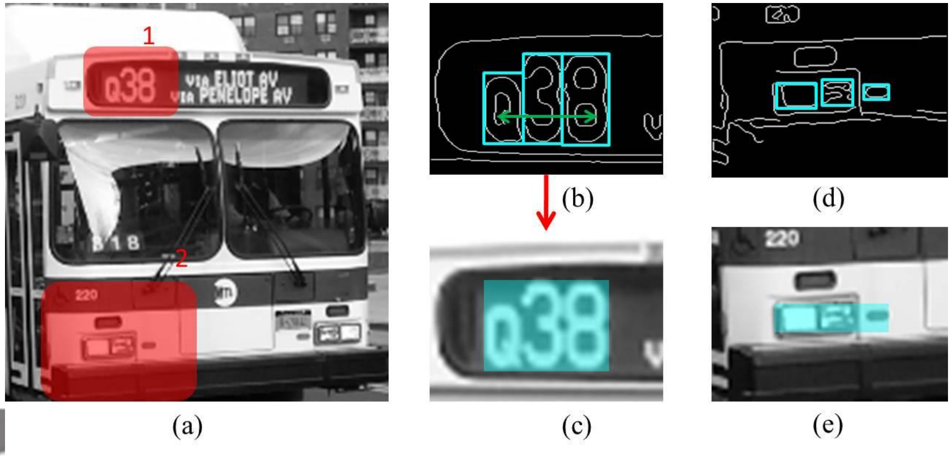
(c) (d)

Bus Information Recognition

- Bus information is important for blind or visually impaired people
  - Bus route number
  - Other information (Local, Express, Not in Service, etc.)
- Scene text extraction
  - Text detection to localize the regions containing text information
  - Text recognition to transform image-based text into readable text codes, for audio output

Bus Information Recognition

- **Scene text detection**
  - Adjacent character grouping [1]
    - Color uniformity
    - Horizontal alignment
  - Text region classification[2]
    - Feature map of character structure, including stroke orientation and edge density
    - Cascade Adaboost learning model to train text classifier



Blue boxes denote the boundary of candidate characters

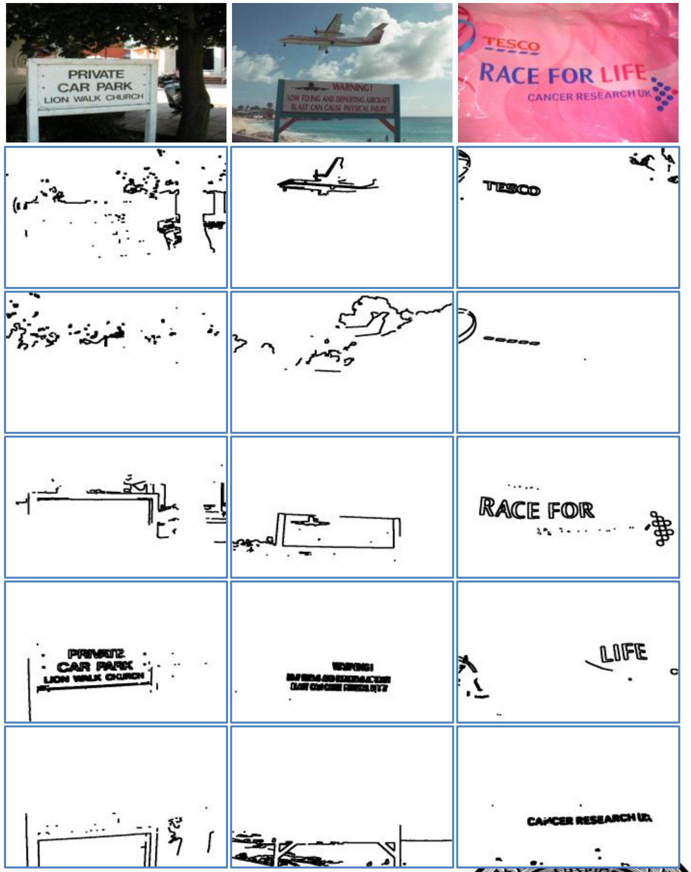
Blue regions denote the candidate text regions

[1] C. Yi and Y. Tian, “Text String Detection from Natural Scenes by Structure-based Partition and Grouping”, *TIP)* Vol. 20, Issue 9, 2011.

[2] C. Yi and Y. Tian, “Assistive Text Reading from Complex Background for Blind Persons”, *CBDAR’11*

Color-based Boundary Clustering

1. Edge detection in scene image, and the object (text/non-text) boundary consists of edge pixels
2. Modeling each edge pixel into a descriptor, related to the pair of colors on both sides of its located boundary
3. Adding spatial position of an edge pixel into its descriptor
4. Gaussian mixture model (GMM) to cluster the close edge pixels in the descriptor space, which have similar color-pair and spatial position.
5. Expectation-maximization (EM) algorithm is employed to optimization the GMM parameters.

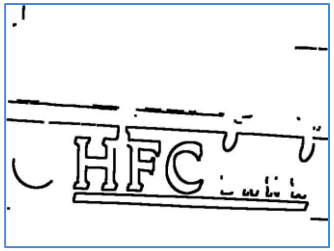


A set of boundary layers, and some layers contain text boundaries where background outliers have been filtered out

Text Character Segmentation

- Filtering out non-text boundary
  - Geometrical properties, e.g. size, aspect ratio, the number of inside boundaries
  - Layout properties, e.g. the number of neighboring boundaries, size/distance relationships with neighboring boundaries

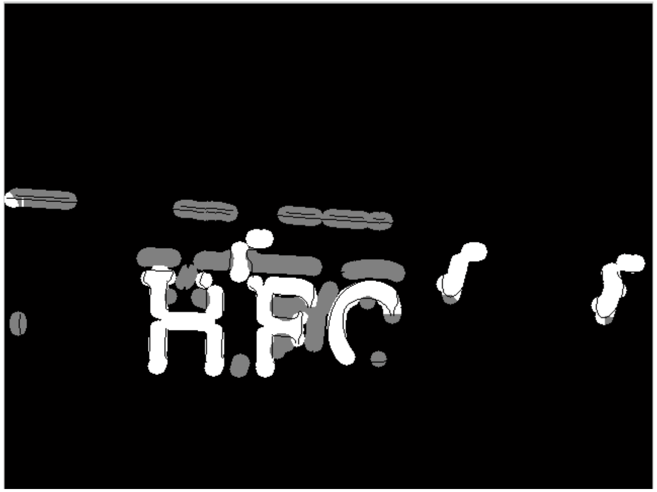
Scene image



A boundary layer

- Restoring the text boundary into connected component of possible text character
  - Binary color assignment, based on the mean color-pair of each boundary layer

- Results: Connected components of possible text characters



Result of color assignment, white and gray represents connected components of text and attachment surface

C. Yi and Y. Tian. IEEE Transactions on Image Processing, 2011, 2012

Horizontal Text String Line Fitting

- Text in natural scene mostly appears in the form of string, rather than a single character

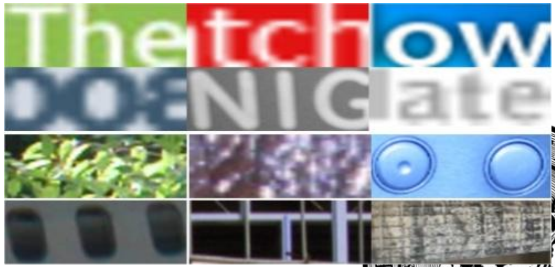
- Text string consists of several characters in approximate alignment
  - Assuming horizontal alignment of text string.
  - Detecting text string in arbitrary text string is feasible, but will introduce more false alarms. Also camera-based text information in real applications satisfies horizontal alignment.

Experiments

- **Bus dataset for training bus classifier and evaluating our demo system**
  - Bus images captured at bus stops
  - 261 positive images containing a bus
  - 434 negative images containing only street view
- **Scene text dataset for training text classifier**
  - Text information captured from natural scene (509 images with average 4 text regions per image)
  - Positive samples are cropped text regions
  - Negative samples are background outliers resembling text structure

(a)

(b)



Results of Bus Detection

- **Evaluating the existence of bus in a camera-based image captured at bus stop**
  - If more than 25% of the image region are classified as positive, we decide that the given scene image contains at least one bus. Otherwise, it does not contain a bus.

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

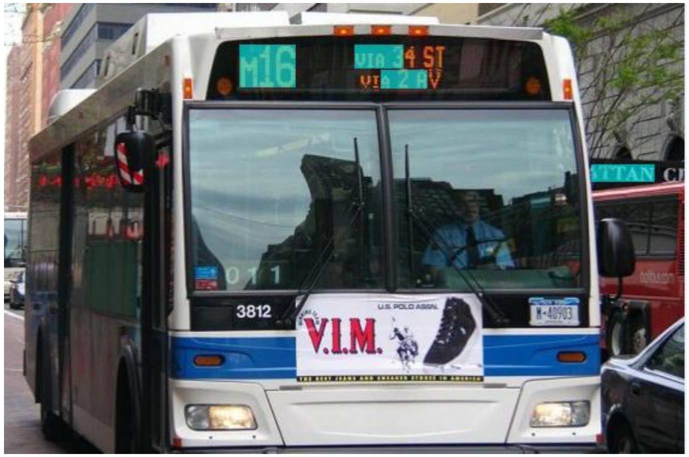
Bus Region Refinement

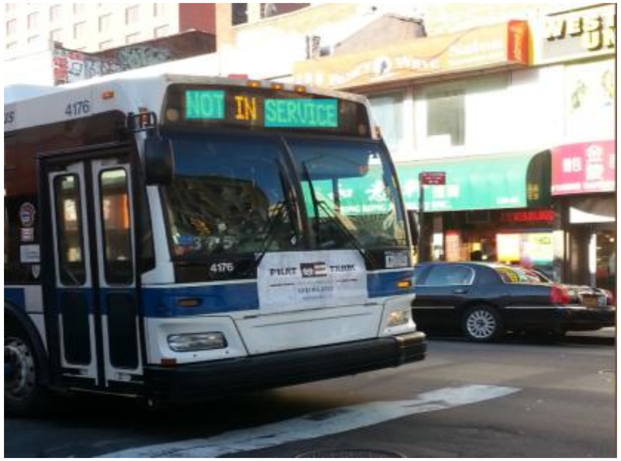
- **Evaluating the accuracy of bus detection in a camera-based image captured at bus stop**
  - A bus region is successfully detected if it has more than 30% overlapping areas with a ground truth bus.
- **Bus images ranging from 358×266 to 864×484 take on average 5 to 20 seconds in bus detection, under un-optimized Matlab code**
  - Two thresholds are set for the score map

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

Results of Bus Information Recognition

M16

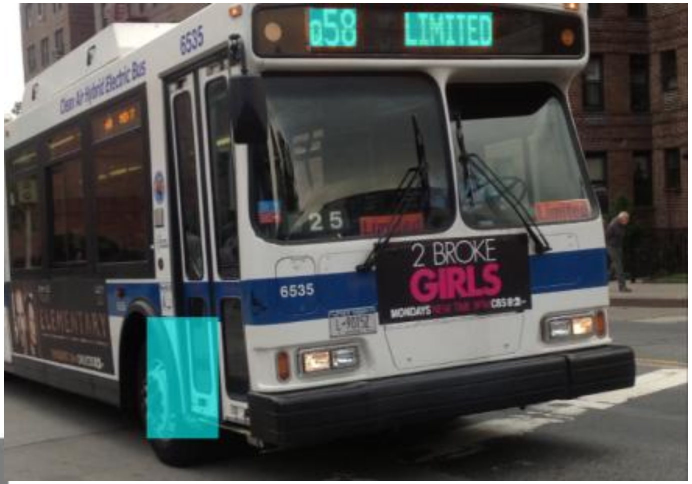


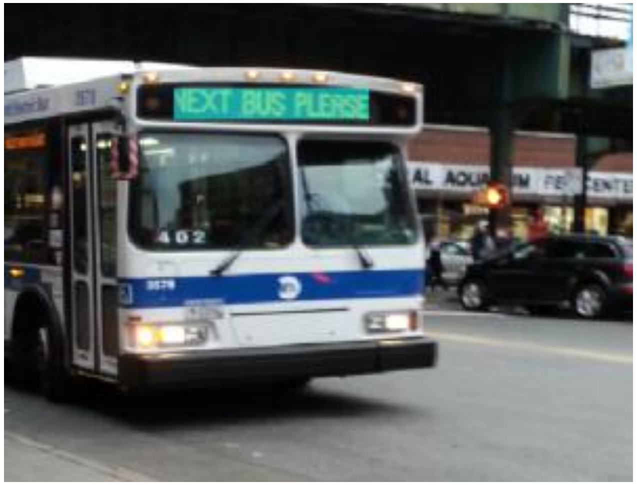


NOT

SERVICE

Q58
LIMITED





NEXT BUS
PLEASE

Summary and Future Work

- Improving the accuracy of text information extraction from the detected region.
- Real-time image/video based bus detection system.
- Integrate with databases of transportation information.
- A user interface study and system evaluation by visually impaired users will also be conducted.

Acknowledgement

- **Supported by:**
  - NSF grant IIS-0957016, EFRI-1137172,
  - NIH 1R21EY020990,
  - FHWA grant DTFH61-12-H-00002,
  - Microsoft Research,
  - CCNY

Questions?