Object Recognition

By:

Dhirendra Singh, Shrey Gupta, Vaseem Ahmed

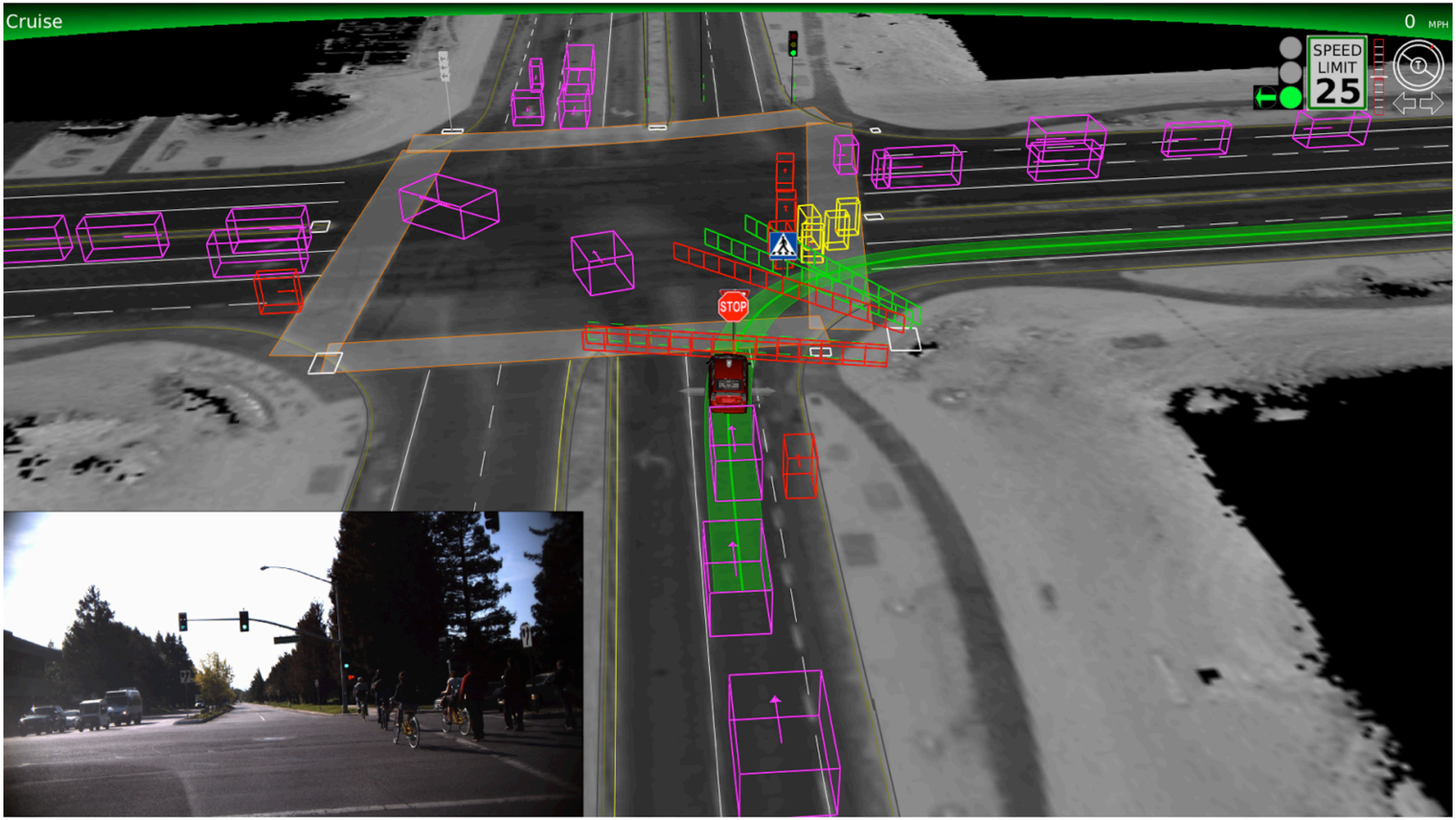
Objective

- Train ML models using annotated images obtained from “LabelMe” data library.
- Test the models on Quebec dataset “BAnQ”.
- Finding objects inside an image.

Uses

- Identifying the presence of an object in an image can be useful in many scenarios e.g.
  - Counting objects in an image
  - Monitoring traffic
- If we can get a bounding box around the object detected, this could be useful in:
  - Object tracking from aerial image
  - Finding location of pedestrian, cars, or other objects for automated driving.

Uses: Automated driving



Courtesy: Google car camera

Dataset

Training data:

"LabelMe"* **Image dataset**: 2920+1133(validation) annotated images.

Annotations of object types: **trees, cars, road, person**. Annotations in the form of polynomial boundaries

Some other datasets:

- PASCAL VOC project
- Caltech dataset

*Russell, Bryan C., and Antonio Torralba. "Building a database of 3d scenes from user annotations." *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.

Examples of Annotated Images

annotated
screen

annotated
car

Dataset

Test data:

Obtained from online *Quebec* archive “BAnQ” downloaded for each of classes **trees, cars, road, person**

General Steps

- Detect points of interest/feature descriptors from a set of annotated objects.
- Extract features from the annotated datasets.
- Feature vector matching between test image and images in the training set.
- Creating a bounding box out of the matched feature points.

Data Preprocessing / Feature extraction

- Use image filters to obtain robust localized features
- filters used:
  1. Laplace of Gaussian: smoothens the image before looking for rapid changes in images: prerequisite for feature extraction
  2. Gabor filter[2] (Currently experimenting): Similar to human visual system for edge detection in spatial domain
- Features:
  1. SURF[1]: : calculates points of interest using LoG.
  2. Histogram of Gradients: calculates features using localized histograms. Are scale and orientation invariant.

[1] Ta, Duy-Nguyen, et al. "Surftrac: Efficient tracking and continuous object recognition using local feature descriptors." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009. APA

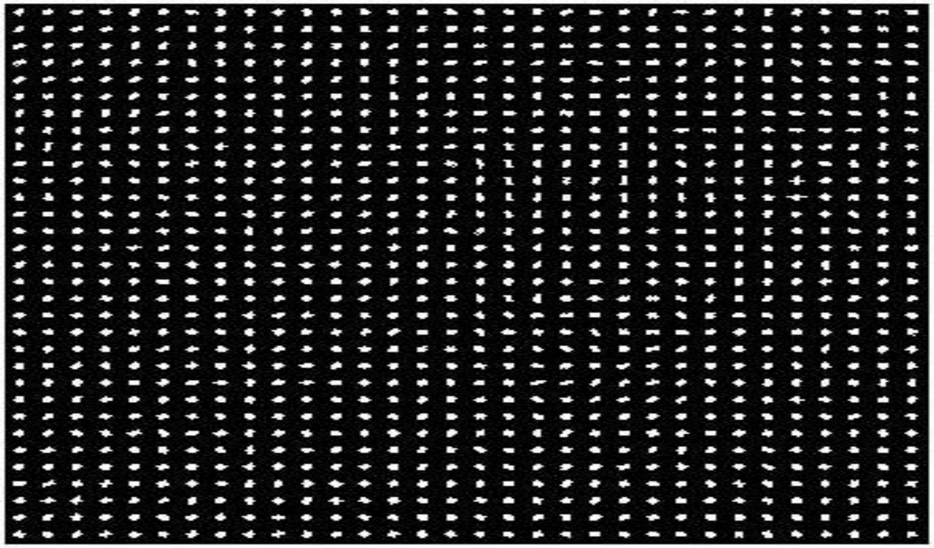
[2] Russell, Bryan, et al. "Object recognition by scene alignment." *Advances in Neural Information Processing Systems*. 2007

Sample original Image

Image after LoG Filter

Sample Image

HOG Features



Method (1^{st})

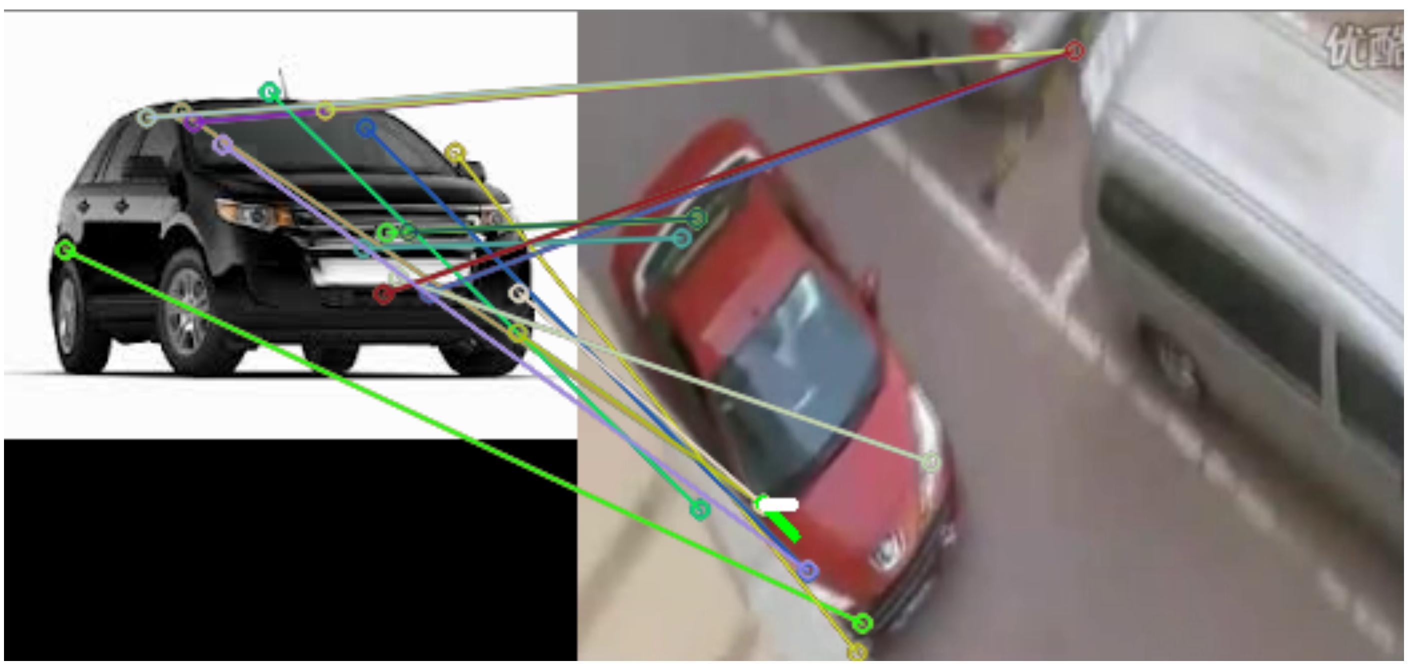
1. Calculate HOG features on the images
2. Train linear SVM model using these features.
3. Calculate HOG features for test data.
4. Test trained model on BAnQ dataset.

Results

1. ~18% accuracy using linear SVM.
2. Low accuracy: we are using entire image to extract HOG features: cropping out interesting region may help.

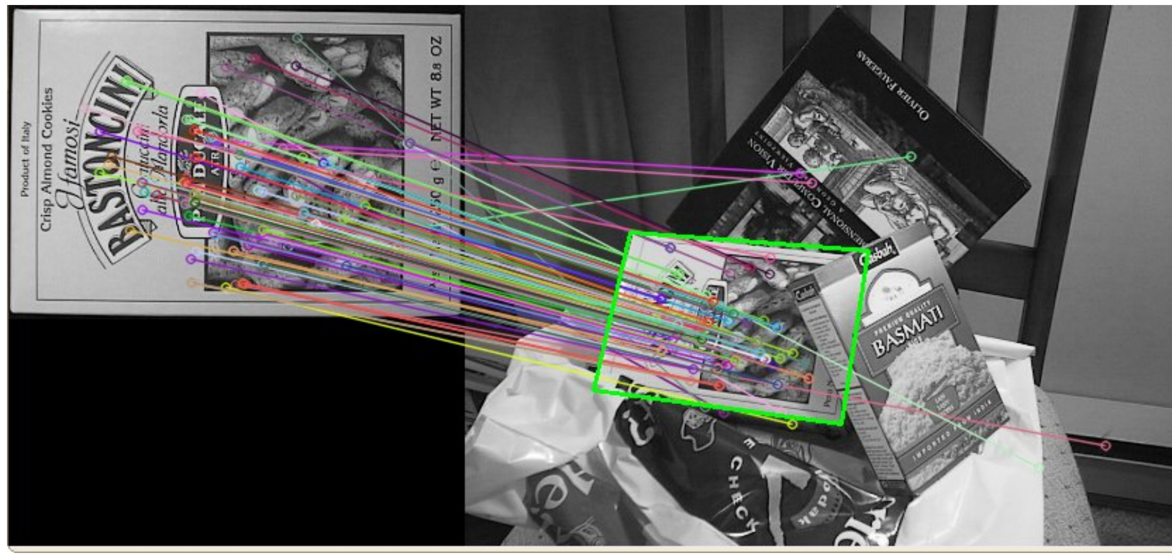
Method (2nd: Ta et.al.)

1. Extract all polygons for an object from labeled images of training dataset, and calculate SURF features on them.
2. Query the database based on similar features (KNN)



Method (2nd: Ta et.al.)

- Obtain bounding box using homography on image



Results

1. ~43% accuracy
2. Accuracy can further be improved using sift based features

Method (3^{rd}):Russell et. al.

1. Extract all polygons for an object from labeled images of training dataset, and calculate HOG features on them.
2. Get K-nearest neighbors using the obtained HOG

Method (3rd):Russell et. al.

- Use the cropped polygons from the Nearest Neighbors to find the bounding box for the object in test image.

Results

1. ~62% accuracy

Future work to improve accuracy for existing test set

- The test dataset is very old: using multiple weak learners (**Adaboost)** on parts of objects like ‘tyre’, ‘screen’, ‘headlights’, might help

THANK YOU