Drivers, Focusing on Your Driving!

Yundong Zhang
CS229 Machine Learning Project Poster
Department of Computer Science
yundong@stanford.edu

Stanford University

Abstract

This project aims to build a computer vision system to detect and alarm the distracted driver. Using the dataset provided by Kaggle, we are interested in using machine learning algorithm to solve the problem. Several methods have been tested and the Convolution Neural Net proves its state-of-the-art performance. Specifically, utilizing the pre-trained VGG network we are able to achieve an accuracy of around 90%, with cross-entropy lose 0.24052.

Motivation

Drivers are supposed to be focusing on driving by law. However, it is very common to see drivers doing something else while driving: texting, drinking, operating the radio, talking on the phone and etc. This distracted behaviors easily cause crash incidents. In US, each day there are over 8 people killed and 1,161 injured in crashes due to a distracted driver, translating to 423,765 people injured and 2920 people killed each year. In China, 30 percent of the car incidents are caused by using cell phones. We hope to design an alarm system that can detect the distracted behavior of car drivers by using a dashboard camera.

Data Preparation

We use the Kaggle Dataset: State Farm Distracted Driver, which consists of 10 classes of image data taken in a car for training. There are in total 22424 training images (labeled) and 79727 test images (unlabeled), each is of size 640x480 pixels.

Figure 1: From left to right: (1) talking on the phone (right); (2) texting (left); (3) drinking; (4) hair and makeup; (5) operating the radio

Approach

SVC

step 1. Bounding Box on human body To make the SVC classifier focus on the indicative features, we train a bounding box according to the HOG features and crop the image. The images are then re-sized to 224x224 for further processing.

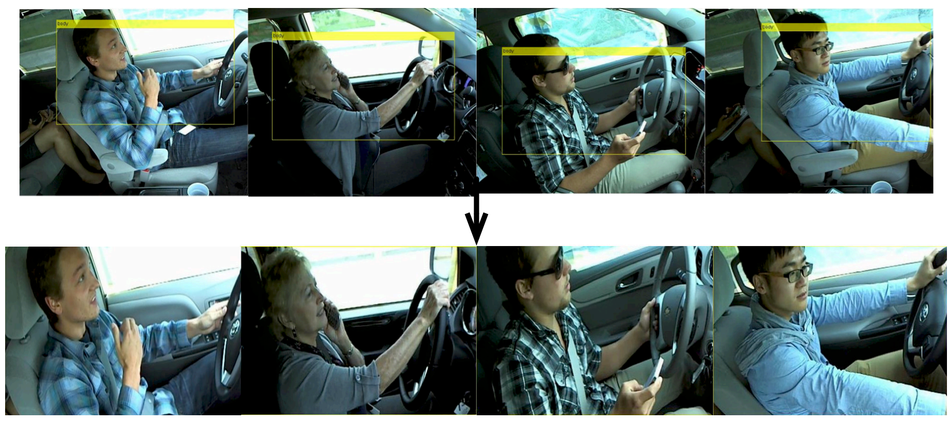

Figure 2: Top: Original Images; Bottom: Images cropped by body using HOG bounding box

step 2. Use PCA for noise removal, dimension reduction & decorrelation To make the small objects more influential in the classification–95% variance is preserved.
step 3. Use Grid-search & K-fold cross-validation to tune SVC hyper-parameters: kernel size, penalty term C, learning rate.

Deep Learning using CNN

1. Training CNN from scratch
A simple network structure similar to MNIST digit recognition, consists of three convolution layers connected by Maxpooling.

2. Transfer learning using VGG-16

- mean normalization and random rotate +/- 10 degrees;
- 25 Epoch with early-stopping, 5-fold Cross-validation;
- Learning rate start with 1e-4, decay with 1e-6, batch size 12;

3. VGG-CAM Network

- Remove the layers after VGG-conv5, resulting in a resolution of 14x14;
- Add a convolution layers with 1024 filters of 3x3, followed by an average pooling layer of 14x14;
- Adam optimizer, batch size 16, learning rate start with 1e-5, decay with 1e-6.
- Visualisation: Map the prediction score back to the added convolution layers, and up-sampling to the input shape.

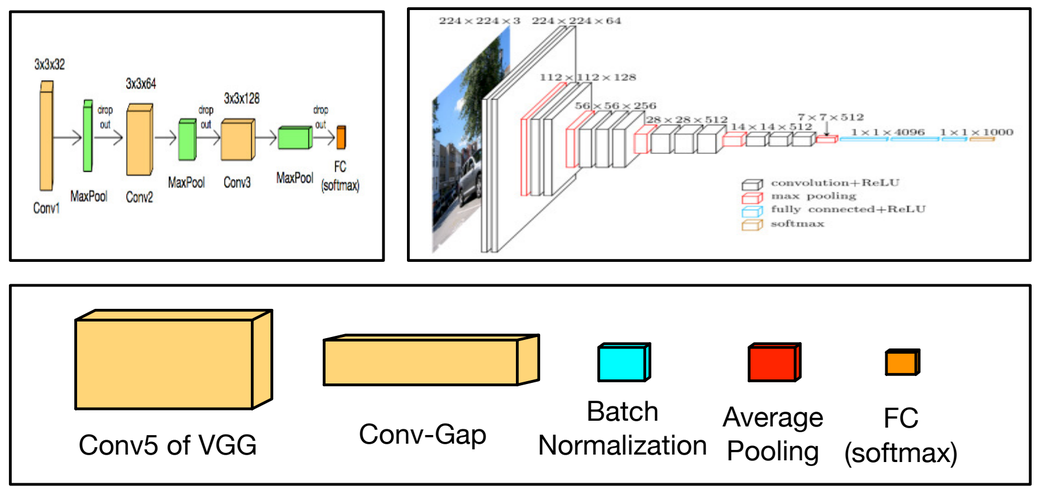

Figure 3: Top left: scratched CNN; Top right: Network in Network VGG-16; Bottom: VGG-CAM

Semi-supervised Learning

Use Pseudo-label method (equivalent to Entropy-regularization): label test-set image with the class that is predicted with the highest probability. We infuse the train set with additional 10,000 images from test-set.

Result

	1	2	3	4	5	6	7	8	9	10
1	2174	24	56	13	1	2	0	3	7	209
2	32	2078	69	23	4	0	27	3	1	30
3	18	2	2192	0	0	0	17	3	68	17
4	34	4	29	2271	3	0	1	0	4	0
5	55	0	28	11	2133	0	1	3	60	35
6	19	1	9	3	0	2253	0	3	0	24
7	0	18	18	11	1	2	2218	0	54	3
8	2	1	18	1	0	0	3	1942	10	25
9	43	8	95	2	50	1	96	83	1503	30
10	371	3	86	7	1	11	9	70	121	1450

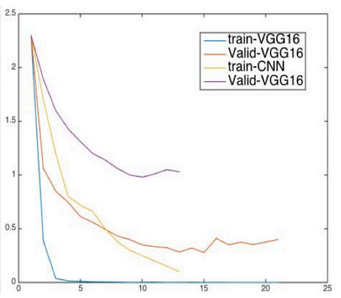

Figure 4: Left: confusion matrix of VGG-16; Right: Learning curves

Model	local accuracy	local log-loss	Kaggle Score
Pure SVC	18.3%	1.94	1.98
SVC + PCA	34.8%	1.73	1.69
SVC + PCA + HOG	40.7%	1.47	1.53
Scratched CNN	63.3%	0.98	1.02
VGG-16	90.2%	0.28	0.34
VGG-CAM	91.3%	0.27	0.32
Ensemble VGG-16 & VGG-CAM	**92.6%**	**0.23**	**0.24**

Table 1: Model Score

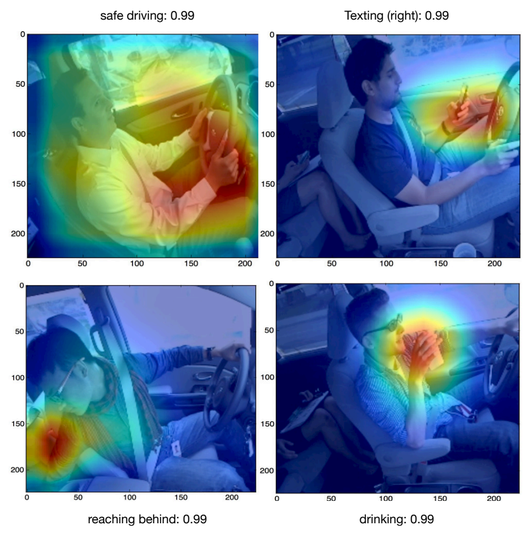

Figure 5: Visualize what the Neural Network is looking for

Discussion

- The ensemble model reduce the variance of the model and achieve the highest Kaggle score, which ranks top 8
- The HOG feature bounding box does not seem to improve the performance of VGG16 in my experiment, mainly due to the quality of cropping is not so good. FRC would be a better candidate.
- Might be better to use even more heavier data argumentation, e.g. zero-out a random box in the image, to prevent over-fitting.