Detection of distracted drivers from dashboard images using convolutional neural networks

Ranganatha Poola Narayana Swamy (rpoolana@buffalo.edu)
Shreeju Jayesh Tanna (shreejuj@buffalo.edu)

October 18, 2017

Abstract

Classification of images obtained from dashboards of cars into 10 possible states of driver (One safe-driving state and 9 distracted states), thereby possibly developing tools to alert drivers when they are distracted. Based on Kaggle competition started by Statefarm. Image data is provided by Statefarm

1 Introduction

The CDC motor vehicle safety division reports that one in five car accidents is caused by a distracted driver (texting, calling, looking back and talking to friends, etc), which translates to 425,000 people injured and 3,000 people killed by distracted driving every year.

A Kaggle competition was setup by Statefarm to improve these alarming statistics, and better insure their customers, by testing whether dashboard cameras can automatically detect drivers engaging in distracted behaviors.

Given a dataset of 2D dashboard camera images, the challenge is to classify each driver's behavior, if they are driving attentively, or if they are distracted.

The image data provided are categorized into below labels:

c0: normal driving
c1: texting - right
c2: talking on the phone - right
c3: texting - left
c4: talking on the phone - left
c5: operating the radio
c6: drinking
c7: reaching behind
c8: hair and makeup
c9: talking to passenger

Sample image from dataset

2 Data

The dataset contains over 20,000 labelled images, out of which 80% are used as training set and 20% are used as validation set. There are over 70,000 unlabelled images that can be used for testing

3 Model

Sequential network with a simple convoluted neural network, a fully connected layer and an output layer.

Layers: The CNN consists of a stack of 3 convolutional layers with ReLU activations followed by max-pooling layers. Fully connected layer consists of 64 neurons with ReLU activations. Output layer has 10 neurons with Softmax activation units to classify the image into one of 10 possible classes

Regularization: Dropout of 50% is used on the fully connected layer to prevent overfitting. Dropout prevents complex co-adaptations on training data, and is an efficient way of performing model averaging with neural networks

Optimization: 'adadelta' is used. It is a per-dimension learning rate method for gradient descent. It dynamically adapts over time using only first order information and has minimal computational overhead. It requires no manual tuning of a learning rate and is robust to noisy gradient information, different model architecture choices and selection of hyperparameters.

Loss: Multi-class logarithmic loss

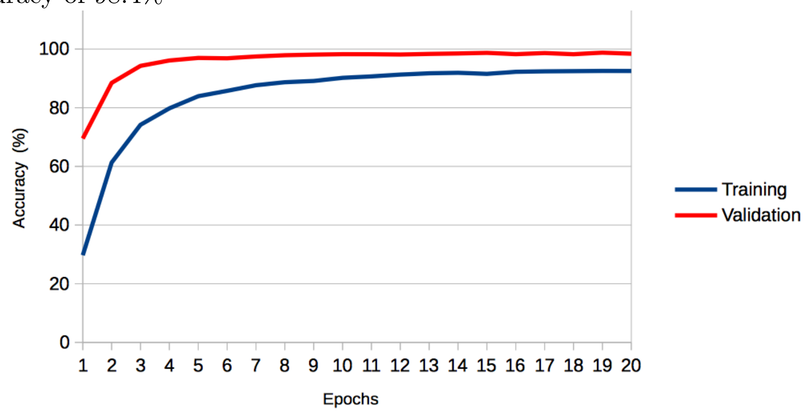
Performance measure: Accuracy and loss

4 Implementation

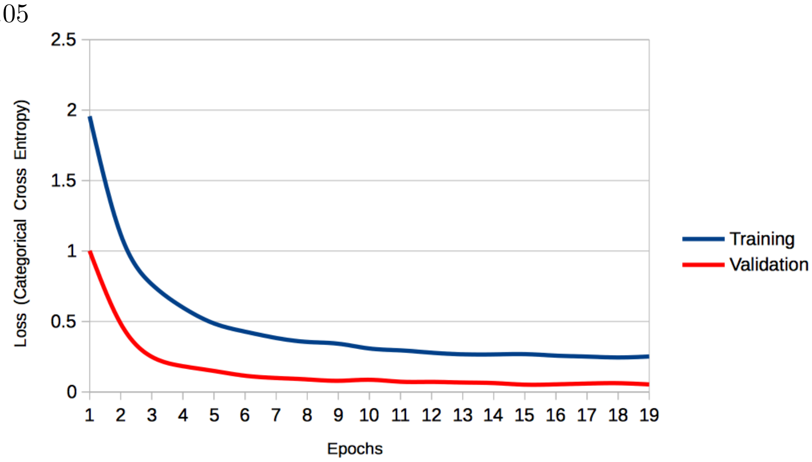
The network is implemented in Python. Keras framework with Tensorflow backend is used to implement our network. OpenCV is used for image processing, numpy for matrix operations.

5 Performance

Accuracy: After 20 epochs, we obtained training accuracy of 92.5% and validation accuracy of 98.4%



Loss: After 20 epochs, we obtained training loss of 0.25 and validation loss of 0.05



References

[1] https://www.kaggle.com/c/state-farm-distracted-driver-detection/data
[2] https://keras.io/getting-started/sequential-model-guide/