Machine Learning Engineer NanoDegree

Capstone Proposal

Yekabote Pavan Kumar
March 25, 2019

Proposal

Domain Background
We've all been there: a light turns green and the car in front of you doesn't budge. Or, a previously unremarkable vehicle suddenly slows and starts swerving from side-to-side.

When you pass the offending driver, what do you expect to see? You certainly aren't surprised when you spot a driver who is texting, seemingly enraptured by social media, or in a lively hand-held conversation on their phone.

According to the CDC motor vehicle safety division, one in five car accidents is caused by a distracted driver. Sadly, this translates to 425,000 people injured and 3,000 people killed by distracted driving every year.

State Farm hopes 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, State Farm is challenging Kagglers to classify each driver's behavior. Are they driving attentively, wearing their seatbelt, or taking a selfie with their friends in the backseat?

Problem Statement
A dataset of 2D images containing a driver, performing an activity, are given. An algorithm has to be developed to classify the activity of driver within the given classes and determine if they are driving attentively, or getting distracted taking selfies and performing other interruptive activities.This can be used to automatically detect drivers engaging in distracted behaviour, while driving, from dashboard cameras.

Datasets and Inputs
Driver images have been captured, where each image contains picture of a driver doing some activity in car. For training, the images containing an activity from any of below classes, will be placed in that class named folder.

The 10 classes to predict are:

c0: safe 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

Following are the file descriptions and the URL's from which files have been obtained.

- Imgs.zip - A zip file containing all train and test images
- driver_imgs_list.csv - A list of training subject ids, class names and respective images.
- sample_submission.csv - A sample format for submission to calculate the evaluation metrics.

The following link will download a zip file of all above listed files:
https://www.kaggle.com/c/5048/download-all

Solution Statement

A convolutional neural network architecture will be developed, using Keras/pytorch, and trained with training images and their respective class labels. Here the network architecture will be optimized to minimize the categorical cross entropy loss. We may include transfer learning method , which takes a model that is already trained on some training data, which includes some patterns relating to our input, and make some customizations to get our required output. Finally, the test images will be passed through the trained network and the obtained results will evaluated based on evaluation policy given in evaluation metrics section.

Benchmark Model

The best model in public leader board score, has categorical cross entropy loss of 0.08689. We are going to take this as our benchmark model. Our model will be built such that, the obtained categorical cross entropy loss will be among top 50% of all submissions made.

Following is the link to public leader board score:
https://www.kaggle.com/c/state-farm-distracted-driver-detection/leaderboard

Evaluation Metrics

The submissions are evaluated based on the categorical cross entropy loss, which is also called as multi class log loss.

$$\text{Multiclass Log loss} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij}\, ln\,(p_{ij})$$

where

- n is the number of images in the dataset
- m is the number of class labels in dataset
- **ln** is the natural logarithm
- y_{ij} is 1, if observation i belongs to class j, otherwise, 0
- p_{ij} is predicted probability that observation i belongs to class j

For each image, we calculate the loss value using loss function, then we will try to minimize the loss , using any of the optimizer functions such as Stocastic gradient descent, Adam etc.

Project Design

As the problem chosen is totally related to computer vision, the deep convolutional networks have been showing state-of-art results in this field. We are going to build a deep convolutional neural network, which can take images as inputs and try to predict the class label. The flow goes as follows:

- Exploring the data
  - Loading libraries
  - load training data and respective labels
  - peek at some samples of data

- Data processing and augmentation
  - Resizing the images
  - Use augmentation techniques to create more examples with some noise
  - Expand the image dimensions
  - Split data into training and validation sets.
  - Feature scaling - Normalization of pixel values of images.

- Creating Model – Deep Convolutional Neural Network
  - Create a convolutional neural network architecture
  - Check out the architecture
  - Compile the model with categorical_cross_entropy as loss function
  - Train the model with training data and evaluate the model on validation data

- Model tuning / Transfer learning
  - Try different architectures to minimize loss
  - Apply transfer learning method to minimize loss
  - Train the model with training data and evaluate the model on validation data
  - Choose best architecture and run the test dataset images through network

- Conclusion

References

[1]https://www.researchgate.net/publication/285164623_An_Introduction_to_Convolutional_Neural_Networks