Machine Learning Engineer Nanodegree

Capstone Proposal

Tayyib Alvi

Proposal

According to the CDC motor vehicle safety division, one of five car accidents are caused by distracted drivers. Every year, distracted driving is causing 425,000 injuries and 3,000 deaths. [1]

State Farm's goal is to increase driver's safety and insure their customers by testing to see if a dashboard camera can accurately detect drivers engaging in distracted behaviors. State Farm created a computer vision competition on Kaggle, a platform that provides data science projects and company sponsored competitions. The company is challenging competitors to classify the driver's behavior.

Domain Background

The goal of this project is to implement and train a classifier that can predict the likelihood of what the driver is doing in each picture with an accuracy of over 90%. Using computer vision and machine learning methods, I will give each image scores with range [0-1] for each behavior listed above.

To get a better idea of how we can tackle this problem, I looked into several research papers that have dealt with human activity recognition from still images.

Problem Statement

This article called "A Review of Human Activity Recognition Methods"[2] provides an overview of the techniques that have been applied to perform human activity recognition in still images or videos. It introduced me to several papers relevant to my goal. One paper presents a survey on still image based human action recognition.

Through this paper, I've learned that many researchers used high level cues for still image-based action recognition such as the human body, body parts, action-related objects, human object interaction and the whole scene or context. These cues can be helpful in defining the different types of human actions. In our case, we would like to focus more on the human object interaction

---

[1] Prevention, C. o. (2017, January). *Motor Vehicle Safety*. Retrieved from Distracted Driving: https://www.cdc.gov/motorvehiclesafety/distracted_driving/index.html

[2] Vrigkas, M., Nikou, C., Kakadiaris, I.: A review of human activity recognition methods. Front. Robot. AI 2, 1–28 (2015)

Datasets and Inputs

The dataset consists of a “imgs” folder that has a test and train folder of 640 x 480 jpg files. The images were taken by a dashboard camera. Each image consists of a driver performing a task from one of the distracted tasks. There are no duplicate images in the dataset. Also State Farm removed metadata from each image (e.g. creation dates). State Farm set up these experiments in a controlled environment. While performing each task, the drivers were not driving as a truck dragged the car on the streets.

Below I listed the number of files for each category in the training data and the test data.

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
2490	2267	2317	2346	2326	2312	2325	2002	1911	2129

In total, there are 22,424 training examples. “Safe driving” has the most examples, and “Hair and makeup” the least. It makes sense to have a lot of “safe driving“ examples because State farm is in general interested in finding out if the driver is driving safely or not. They would rather have false negatives than false positives when labeling safe driving. More examples would improve the classifier’s performance in decreasing false positives for safe driving. “Hair and makeup” may be less because it is a task mostly performed by women.

Testing data	79726 files

There is also a csv file that lists training images’ filenames, their subject (driver) id, and class id.

Drivers	26 drivers

It may be hard to accurately train a classifier with 26 drivers due to the differences in driver’s physical traits, especially if the dataset is relatively small for each driver. However, in the end we can create a robust classifier that will make good predictions on people with different sizes, gender etc. The link for the dataset can be found at https://storage.googleapis.com/kaggle-competitions-data/kaggle/5048/imgs.zip?GoogleAccessId=web-data@kaggle-161607.iam.gserviceaccount.com&Expires=1520490902&Signature=ACUebSUCOZ3sMNt254zEoXRuh0cdB9gMTkWMiH2fpYoiAnNqAM5vlQBCHPcbM7S8r8%2FfilDqUo779cUkdixC526wpsxlYPFxG5ofpExedqm%2Bhw3Lt%2FBH9t%2Fr6CIOQzWaFFXuT1%2FkIMw%2BqMcABpmREav%2B0D4My%2BQM0vu8s5HpmD7PBX4DVxJIssHQ1qrcEnkGr%2BbAcGXtE0Tvin4xSTSJO%2BBJHsAzbOj57E9XjL2jLa2Hnp8crgRLMUe9N9OIEYVtx%2FvoBr%2FbVQVyEik9OEANUm0HZut4IrIALRwiiG4Fsbm8LeSLIR50nMSWmcss69gBbEJTCTxgIoi73hw92wxyEA%3D%3D

Solution Statement

A. Analyze DSIFT and HOG features

1. I believe these two approaches are most appropriate for this particular problem. We will not need SC for human contour detection. SC will not give us much information because all the images consist of a driver in a sedentary posture. Also GIST will not be much of help because it focuses on obtaining the background information. In our case, the background is always the same. What's most important for us is to use a feature that can extract detail about the human's interaction with an object and here SIFT and HOG can be helpful.

B. Choose a descriptor to extract from images.

1. In this part, I may have to use PCA to reduce the dimensionality of the features I extract from a large set of images.

2. I would like to find the most efficient and effective feature here that retains most of the image's information.

C. Implement classifiers

1. I hope to implement three classifiers most appropriate for this problem.

D. Train and test classifiers

1. Using the extracted features of all the training images, I like to train each classifier, and test to see which one does a better job.

E. Improve the classifiers and decide on one classifier

1. Using sklearn's GridSearch function, I would like to improve the classifiers and pick the one that does the best.

Benchmark Model

A. We are benchmarking the performance of the models against the performance of an image classification taught in Building Machine Learning Systems with Python[3] by Willi Richert and Luis Pedor Coelho. I am using the same features as this example, and the best performance the book was able to achieve was 67%. For the results, I am expecting a performance of 45-60%. In the book, the images are classified into four classes: cars, animals, transportation, and natural scenes. However, in my case, the problem will be a lot harder in that I will be classifying the images into 10 different classes. Also compared to the four classes mentioned above, the type of classes I will be classifying the images have

[3] Richert W (2013) Building machine learning systems with python. Packt Publishing Ltd, Birmingham

B. very little differences in images. The differences between a image of a car and a bee is easier to distinguish than the difference between a driver talking to a passenger and a driver that's not.
C. The training set includes already manually labeled image instances. We can compare each model's performance with the training data by estimating the accuracy using a Leave One Out cross validation or K Fold Cross Validation.

D. **K Fold Cross Validation** - K Fold(n, n_folds) : divides all the samples in k groups (folds) of samples. This validation creates k-1 training set and leave one out for test.

E. **Leave One Out Cross Validation** - LeaveOneOut(n) : cross validation technique provided by

F. Scikit-learn. The general idea is that if there are n samples, the validation will create n different training and n test sets. The only difference here with the K Fold is that Leave One Out CV will create n models from n samples instead of k models where k < n. Thus Leave One Out can be computationally expensive compared to K Fold if the dataset is very large. Also Leave One Out is known to have a high variance due to the overlap it has in each training set. However it is advantageous in that a 5 or 10 fold CV can overestimate the generalization error when the learning curve for a training set is very steep.
G. Many researchers have suggested using 5 - 10 fold cross validation instead of LOO. I hope to use these two validation for testing my model. However if the dataset is too large, I will use 5 and 10 fold cross validation.

Evaluation Metrics

To evaluate the image classification accuracy, I will perform the following tasks.

- Compute accuracy score

- Create confusion matrix

- Produce precision, recall and f1 score

Accuracy Score - With sklearn, you can compute the accuracy of the classifier with sklearn.metrics.accuracy_score. This will give you an idea of how the classifier is doing. It is simply telling you how many correct predictions the classifier made. However, it will not tell me the details of misclassification, thus we will have to take some extra steps to evaluating the classifier.

Confusion matrix is great in that you can visualize the detailed performance of a supervised learning model with a table layout. It is specifically useful for multi category classifications in which each column represents the instances in a predicted class while each row represents the instances in an actual class. With this visualization one can see clearly if the system is mislabeling instances or "confusing" two or more categories. A perfect model will show a confusion matrix with all the answers in the diagonal of the table. Thus with a confusion matrix, one can easily detect the errors of the model when there are values outside of the diagonal.

Project Design

Features and Algorithms the project will be using are:
- 1st classifier: Haralicks
- 2nd classifier: Linear Binary Patterns
- 3rd classifier: Surf Descriptors
- 4th classifier: Haralicks and Linear Binary Patterns
- 5th classifier: Haralicks, Linear Binary Patterns, and Surf descriptors

After the actual results there will be later refinement state:
1. Refinement to reading large dataset of images
2. Refinement to preprocessing data
3. Refinement to finding the best model with the best set of features