Tom Orth

Udacity Nanodegree: Machine Learning Engineer Capstone Proposal

1. Domain Background

   The domain background for this project is image classification. This has been in the forefront of research and the world. One area is medical image classification (https://arxiv.org/pdf/1704.06825.pdf), where image classification techniques are being used in order to help doctors in the medical industry make better decisions. Another area is urban planning (https://link.springer.com/chapter/10.1007/978-3-642-60105-7_11). A third area is facial recognition (https://arxiv.org/pdf/1804.06655.pdf). The goal of the last two areas is similar to the first; use image classification techniques in order to improve decision making and other tasks related to the field. I have chosen this domain area personally because I do not have much experience with image based machine learning. My background is primarily in NLP.

2. Problem Statement

   Design a model in order to classify different dog breeds. In the case of a human face being present, predict the resembling dog breed. The intended model to solve this task is a Convolutional Neural Network (CNN) (https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53?gi=86d90526cc90), a model that works well for taking images and extracting features out of them. CNNs can be implemented using the Pytorch framework and can compute metrics such as accuracy, precision, recall, and F-measure. Note: When evaluating the model, this would be done solely on dog images since the human images dataset do not provide a label for the dog breed the person most resembles.

3. Dataset and Inputs

   The dataset that will be used as input to the model will be the dogImages dataset provided at https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip to provide dog breeds. Additionally, the human faces dataset, found at http://vis-www.cs.umass.edu/lfw/lfw.tgz, will be used in order to detect a human's face. This is done to ensure that the model only runs on a dog image or human image, as specified by the Dog Breed CNN notebook. The data can be loaded in using Pytorch's dataloader (https://pytorch.org/tutorials/beginner/data_loading_tutorial.html) in order to get the data into the model.

Example dog image:

Example human image:

4. Solution Statement

The solution will be a CNN using transfer learning in order to improve the accuracy of the model. Transfer learning is using a pretrained model that was trained on a large amount of data and then using it for training of a new model on a small amount of data. Some pretrained

models I would look into are VGG-16 (https://neurohive.io/en/popular-networks/vgg16/) and Resnet50 (https://pytorch.org/hub/pytorch_vision_resnet/). The implemented model would use the metrics described in Section 6 and be compared to the benchmark models described in Section 5.

5. Benchmark Models

The solution in Section 4 will be benchmarked against VGG-16 (not with transfer learning) and a CNN from scratch, in order to assess its accuracy. Ideally, the transfer learning implementation should perform better than the VGG-16 standalone and the CNN from scratch.

6. Evaluation Metrics

The evaluation metrics chosen are Accuracy and F-measure. The reason for this is the fact they are two popular metrics in machine learning in order to evaluate a model. Accuracy is a staple evaluation metric for any task performed. However, F-measure is a stricter performance metric which can yield more insight into a model's performance (https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/).

7. Project Design

The project would be implemented in a Jupyter Notebook. First, a dog face and human face detector would be implemented. This is done in order to ensure that the model implemented only runs on a human or dog face image. If an image is not either one, an output of an error is printed during model evaluation. Next, VGG-16 will be loaded into the notebook and test it on the dataset to see it's performance. Afterwards, a CNN from scratch will be implemented and evaluated. Finally, a transfer learning CNN will be implemented and evaluated in a similar manner.