

Udacity Capstone Project

Domain Background

Multiclass classification is a classification task that consists of more than two classes. In multiclass classification, a sample can only have one class (ie. an email is only an email; however it can belong to various categories like primary, social, promotion, marketing, forums or spam).

Apart from regression, multiclass classification is the most herad common machine learning task. In classification (for ex emails), we are presented with a number of training emails divided into K separate classes, and we build a machine learning model to predict to which of those classes previously unseen data belongs to. During training process, the model learns patterns specific to each class and uses those patterns to predict the classification of the future data.

One more reason for this is that today there is a lot of digitized data that we need to successfully implement good ML models.

Dog breed identification problem is well known in the ML community. We can find it on Kaggle: <https://www.kaggle.com/c/dog-breed-identification/overview/description> I will be considering similar problem to solve.

Problem Statement

The aim of the project is to build a pipeline to process real-world, user-supplied images. A convolutional neural network (CNN) will identify an estimate of the dog's breed given an image. When the image is of a human, the CNN will choose an estimate of a dog breed that resembles the human. If neither a dog or a human is detected, then an error message is output. Therefore, the models in place should be capable of detecting a dog or human in an image, classify the dog to its breed and classify a dog breed that the human resembles.

Datasets and Inputs

The dataset is saved under S3(Simple Storare Service) offered by Amazon Web Service and is provided from Udacity.

Human Files location: <https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip>

Dog Files location: <https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip>

Solution Statement

We will use Convolutional Neural Networks (CNN) or ConvNet model to address this problem. It is a Deep Learning Algorithm which accepts input image, assigns importance (learnable weights and biases) to various aspects/objects in the image and helps to differentiate one from the other. To find if the picture is human or not we will use the OpenCV model. And to find if the dog is on a picture we will use a pre-trained torch model i.e VGG16. We will create our CNN model using transfer learning to get great results.

Benchmark Model

- The model created must have accuracy of at least 10%, to make sure that model is working, because random guess will perform correct answer roughly, which corresponds to less than 1%.
- And the model which created with Transfer Learning must have accuracy of 60% and above

Evaluation Metrics

The problem we try to solve is a multi class classification problem. Because our data is unbalanced, using simple accuracy score will not justify. On other platform like Kaggle, multi-class log loss metrics is used to evaluate models. I will be building model using categorical cross entropy loss to increase/decrease weights and accuracy as metric.

Project Design

Following steps will be followed for Project Design :

- Step 0: Import Datasets -> Load necessary libraries and import data sets from relevant resources, make necessary pre-processing and prepare it for CNN models
- Step 1: Detect Humans -> Use Open CV which is loaded with many pre-trained models to perform Human Face detection, I will be using haarcascades detector.
- Step 2: Detect Dogs -> Use pre-trained model VGG-16 to detect dogs in images along with the weights that have been trained on [Image Net](#)
- Step 3: Create a CNN to Classify Dog Breeds (using traditional technique) -> This is a challenging task where assigning breed to dogs from images, as there will be a lot of challenges in classifying which have more similarity
- Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning) -> Use transfer learning to classify dog breed, to attain this create CNN that classifies dog breed and the process

includes specifying max pool, loss function , optimizer etc

- Step 5: Write your Algorithm using the code which is written Step 3 and Step 4 and come up with following results:
 - Predict breed if dog is detected in input image
 - Similar dog breed if human is detected in input image
 - Error if neither is detected
- Step 6: Test Your Algorithm
 - Check the written model and test with random input images

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.