Convolutional Neural Networks (CNN) project
Dog Breed Identification

Vikash Rawat
October 2020.

Domain Background

The Dog breed classifier is a well-known problem in ML. The aim of the project is to develop a dog breed classifier using Convolutional Neural Network (CNN). The final program should be able to accept an image, if a dog is detected in the image, it will provide an estimate of the dog's breed If a human is detected, it will provide an estimate of the dog breed that is most resembling. This is a supervised learning problem and because we have our dog images divided into breed classes we will use classification predictive modeling more precisely multi-class predictive model.. After completing this model, I am planning to build a web app where user can input an image and obtain prediction from this model. This project gives me an opportunity to build and deploy ML models, so I have chosen this as my capstone project.

Problem Statement

The goal of the project is to build a machine learning model that can be used within web app to process real-world, user-supplied images. The algorithm has to perform two tasks:

Dog face detector: Given an image of a dog, the algorithm will identify an estimate of the canine's breed.

Human face detector: If supplied an image of a human, the code will identify the resembling dog breed.

Datasets and Inputs :

For this project, the input format must be of image type, because we want to input an image and identify the breed of the dog. The dataset for this project is provided by Udacity. The dataset has pictures of dogs and humans.

Dog images dataset: The dog image dataset has 8351 total images which are sorted into train (6,680 Images), test (836 Images) and valid (835 Images) directories. Each of this directory (train, test, valid) have 133 folders corresponding to dog breeds. The images are of different

sizes and different backgrounds, some images are not full-sized. The data is not balanced because the number of images provided for each breed varies. Few have 4 images while some have 8 images.

Human images dataset: The human dataset contains 13233 total human images which are sorted by names of human (5750 folders). All images are of size 250x250. Images have different background and different angles. The data is not balanced because we have 1 image for some people and many images for some.

Solution Statement

For performing this multiclass classification, we can use Convolutional Neural Network to solve the problem

. A Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The solution involves three steps.

First, to detect human images, we can use existing algorithm like OpenCV's implementation of Haar feature based cascade classifiers.

Second, to detect dog-images we will use a pretrained VGG16 model.

Finally, after the image is identified as dog/human, we can pass this image to an CNN which will process the image and predict the breed that matches the best out of 133 breeds.

Benchmark Model

For our benchmark model, we will use the Convolutional Neural Networks (CNN) model created from scratch with an accuracy of more than 10%. This should be enough to confirm that our model is working because random guess would be 1 in 133 breeds which are less than 1% if we don't consider unbalanced data for our dog images.

Evaluation Metrics

For this multi class classification, Multi class log loss will be used to evaluate the model. Because of the imbalance in the dataset, accuracy is a not a good indicator here to measure the performance. Log loss takes into the account of uncertainty of prediction based on how much it varies from actual label and this will help in evaluating the model.

Project Design

Step 1: Import the necessary dataset and libraries, Pre-process the data and create train, test and validation dataset. Perform Image augmentation on training data.

Step 2: Detect human faces using OpenCV's implementation of Haar feature based cascade classifiers.

Step 3: Create dog detector using pretrained VGG16 model.

Step 4: Create a CNN to classify dog breeds from scratch, train, validate and test the model.

Step 5: Create a CNN to Classify Dog Breeds using Transfer Learning with resnet101 architecture. Train and test the model.

Step 6: Write an algorithm to combine Dog detector and human detector.

- If dog is detected in the image, return the predicted breed.
- If human is detected in the image, return the resembling dog breed.
- If neither is detected, provide output that indicates the error.

References

1. https://en.wikipedia.org/wiki/Convolutional_neural_network
2. https://www.kaggle.com/c/dog-breed-identification/overview/description
3. https://github.com/udacity/dog-project
4. https://www.kdnuggets.com/2018/04/right-metric-evaluating-machine-learning-models-1.html
5. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234