Sana Jahnavi
20 October 2020

Machine Learning Capstone Project

Domain Background

The Dog breed classifier is the problem where in if an image of dog or human is supplied we have to identify the resembling breed. The idea is to build a pipeline that can process real world user supplied images and identify an estimate of the canine's breed. This is a multi-class classification problem where we can use supervised machine learning to solve this problem. After completion of the model. I will build a web app where the user can input in the image format and the predictions will be outputted on his screen. I felt this project is very helpful in identifying a dog breed when the owner himself is confused and for people who are tempted to dogs and want to know dog breed to buy it if they have available picture of the dog.I will be using Convolution Neural Network to build this image classifier as they are specialized type of neural network for processing the grid data.

Problem Statement

The main aim of the project is to build a machine learning model that can be used in different web applications, real-world apps to detect the dogs breed. The algorithm has to do mainly two tasks:

1. If given an input of a dog image the model should identify the breed.
2. Given a human image, the code should identify the resembling dog breed.
3. A good accuracy must be achieved

Datasets and Inputs:

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

The images are of various sizes and different backgrounds, some are not full-sized. The data is imbalanced as number of images in each breed differ. 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. This data is also imbalanced as the number of different people's image vary.

Solution Statement

The solution is to design a CNN model that which can estimate breed of a dog correctly as far as possible which is included in the picture. To correctly predict this, it must be recognize beforehand whether a person or a dog is included in a picture. If a person is present, the resembling dog breed must be identified. If a dog is present, an estimate of the dog breed should be outputted.

Benchmark Model

The CNN model created from scratch must have accuracy of at least 10%. This can confirm that the model is working because a random guess will provide a correct answer roughly 1 in 133 times, which corresponds to an accuracy of less than 1%.

The CNN model created using transfer learning must have accuracy of 60% and above.

Evaluation Metrics

For a multi class classification, I feel multi class log loss can be used as an evaluation metrics. 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. Other general metric is the accuracy which is straightforward metric for any machine learning model

Project Design

Sana Jahnavi
20 October 2020

Step 0: Import the required dataset and libraries, Pre-processing the data and creating train, test and validation dataset. Perform Image augmentation on training data.
Step 1: Detect human faces using OpenCV's implementation of Haar feature based cascade classifiers.
Step 2: Make a dog detector using pretrained VGG16 model.
Step 3: Create a CNN to classify dog breeds from scratch, train, validate and test the model.
Step 4: Create a CNN to Classify Dog Breeds using Transfer Learning. Train and test the model.
Step 5: Write an algorithm to combine Dog detector and human detector such that when an image of dog is given it predicts the breed and if human is detected then the resembling breed and throwing an error if neither provided.
Step 6: Testing the algorithm and evaluating it through different metrics

References

1. Original repo for Project - GitHubhttps://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification
2. Resnet101: https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html#resnet101
3. Imagenet training in Pytorch: https://github.com/pytorch/examples/blob/97304e232807082c2e7b54c597615dc0ad8f6173/imagenet/main.py#L197-L198
4. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53