Capstone Project

Priyatham Susarla

Machine Learning Engineer Nanodegree

September 17, 2020

Dog Breed Classifier using CNN

Domain Background

The problem is to identify a breed when a dog image provided as input, will classify it as human besides as it will try to match a similar dog breed that closely matches the picture provided. The aim is to differentiate between human and dog faces initially and then classify dogs based on breed.

The task at hand involves image recognition and classification where uses machine learning techniques to aid computer vision. A convolutional neural network, CNN for short, is preferably used here. This project allows me to build and deploy ML models and also algorithms, so I have chosen this as my capstone project.

Problem Statement

In this project, it is essential to build a data processing pipeline to classify real-world, user-supplied images. Since the user input can contain various types of images algorithm is supposed to perform two tasks:

1) Define an algorithm that defines a dog’s breed based on a picture of a dog.
2) If a human image provided, identify as a human face, and predict a similar dog breed.

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 following datasets provided by Udacity will use:
Dog images dataset: The dog image dataset can download here:
https://s3-us-west1.amazonaws.com/udacity-aind/dog-project/dogImages.zip

Human images dataset: The human dataset can download here:
https://s3-us-west1.amazonaws.com/udacity-aind/dog-project/lfw.zip

The dataset contains images of humans as well as dogs. The dataset contains images of humans and dogs. With 13,233 images of humans and 8351 images of dogs, there seems to be a reasonable amount of variation present in the dataset that sort into a train (6,680 Images), test (836 Images), and valid (835 Images) directories. Each of these directories (train, test, valid) has 133 classes corresponding to dog breeds.

The images are of different sizes and different backgrounds, and some pictures are not full-sized. The data imbalanced because the number of photos provided for each breed varies. Few have four images, and while some have eight. The human dataset is also imbalanced because we have one picture for some people and many pictures for some.

The human dataset will use to detect human faces in images using OpenCV's implementation of Haarfeature-based cascade classifiers. The dog dataset will use to identify dogs in images using a pre-trained VGG-16 model.

Solution Statement

For performing this classification, we can use the Convolutional Neural Network to solve the problem. A Convolutional Neural Network (CNN) is a Deep Learning algorithm that can take in an input image that can be able to differentiate one from the other.
First, to detect human images, we can use an existing algorithm like OpenCV's implementation of Haar feature-based cascade classifiers.

Second, a pre-trained VGG-16 model with trained weights on ImageNet, a popular dataset for image classification, will detect dogs in the user-supplied images.

Benchmark Model

The CNN created from scratch serves as the benchmark for this project and test accuracy of at least 10%. That will 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 an accuracy of 60% and above. This value can use to get accurate results, which can further be improved using similar hyper-parameter tuning methods.

Metrics

The goal here is to compare the performance of my model with that of the benchmark model. Therefore, I would use accuracy as an evaluation metric and also, because the benchmark model only specifies the accuracy. The cross-entropy criterion from PyTorch can provide this evaluation metric.
Precision will use to see how precise our Haar Cascade classifier can identify human faces and not human faces.
Precision = True Positive / (True Positive + False Positive)
Accuracy will be used as a metric to measure the efficiency of our custom-created networks.
Accuracy = (True Positive + True Negative) / Size of all train images

Project Design

Step 1: Import the necessary dataset and libraries, Pre-process the data, and create a 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 a dog detector using a pre-trained 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 the Dog detector and human detector.

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

Step 7: Testing the solution model with images from the datasets to see the model at work.

References

1.Original repo for Project- GitHub:
https://github.com/udacity/deep-learning-v2-pytorch/blob/master/project-dog-classification/dog_app.ipynb
2. Resnet101:
https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html#resnet101
3. PyTorch Documentation:
https://pytorch.org/docs/master/
4. Very deep convolutional networks for large-scale image recognition:
https://arxiv.org/pdf/1409.1556.pdf
5. ImageNet training in PyTorch:
https://github.com/pytorch/examples/blob/97304e232807082c2e7b54c597615dc0ad8f6173/imagenet/main.py#L197-L198
6. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53