Machine Learning Engineer Nanodegree

Capstone Proposal

Ishaan Upadhyay, June 10th 2020

Domain Background

Dog breed classification is an open issue in the image recognition field, contained as a subset of several datasets such as ImageNet on which some of the best convolutional models have been trained. Dog breed classification is problematic because of some breeds having minimal variation between each other, as well as some breeds having entirely different colours. In this project, the task given is to construct a classifier that can classify dogs into different classes based upon estimated breed, as well as give an estimate of which breed a human being resembles. This could eventually be deployed in a web-app or pipeline.

Problem Statement

Given an image, first, identify whether or not it contains a dog or a human using a detector. If a dog is detected, identify the breed it belongs to. If a human face is detected, identify which dog breed it most closely resembles. The product must be suitable for usage in a web app.

Datasets and Inputs

The 2 datasets used here are both offered directly by Udacity, hosted on Amazon Web Services. The inputs to the program will all be images, both for model training and the final application.

Human images dataset: This dataset contains 5749 folders containing a total of 13233 images. Each folder is labelled using a person's name and contains pictures of that person. This dataset does not contain an equal number of pictures for every person, making the data slightly imbalanced.

Dog images dataset: This dataset contains 133 folders containing a total of 8351 images. Each folder corresponds to a dog breed which can be classified by pretrained models, formatted using a number and the breed name. This dataset is also slightly imbalanced because there is not an equal number of images for each breed.

Solution statement:
First, to determine whether the image contains a dog, human or neither, it will be fed through a face detection algorithm such as OpenCV's feature-based Haar cascade classifiers, followed by a pretrained VGG16 model to detect dogs.

After this, if a dog or human is detected, the image will be forwarded to a ResNet50 model which was trained using transfer learning. In the case of a dog, its predicted breed will be returned. In the case of a human, the closest matching breed will be returned. If neither is detected, the app will say that the image is invalid.

Benchmark models:

- For the CNN model created from scratch, it must exceed 10% accuracy. Even a model simply making a random guess will guess correctly 1 in 133 times, less than 1% accuracy. If it exceeds 10% accuracy, we will know that it is working correctly.

- For the transfer learning model, it must exceed 60% accuracy.

Evaluation metrics:

The loss will be evaluated using cross-entropy loss, a combination of negative log likelihood loss and logarithmic softmax. The cross-entropy loss function is common in classification problems with any number of classes, and works best here.

Accuracy will only be used to evaluate if the model has been trained properly after training is finished.

Project design:

1. Download the dataset and import all the necessary libraries. Pre-processing should be done on all the images, followed by the creation of train/validation/test datasets and any necessary augmentation.
2. Implement face detector and dog detector functions, using Haar feature-based cascade classifiers for the former and a pretrained VGG16 model for the latter.
3. Create a CNN from scratch using torchvision, followed by training, validation and testing. Save the best model.
4. Using the pretrained ResNet50 model, imported from torchvision, create a CNN using transfer learning, and then test it. Create a function that returns the classification from this model.
5. Write a final function that combines the dog detector and face detector functions. If a dog is detected, return its predicted breed from the transfer learning model. If a human is detected, return the closest breed. If neither is detected, output an indication that the image was invalid.

References

Dog Breed Identification. (n.d.). Retrieved from https://www.kaggle.com/c/dog-breed-identification/overview/description

Torchvision¶. (n.d.). Retrieved from https://pytorch.org/docs/stable/torchvision/index.html