Jingyi Han
June 7th, 2021

Udacity Machine Learning Engineer Nanodegree Capstone Proposal
Convolutional Neural Networks Application in Dog Identification App

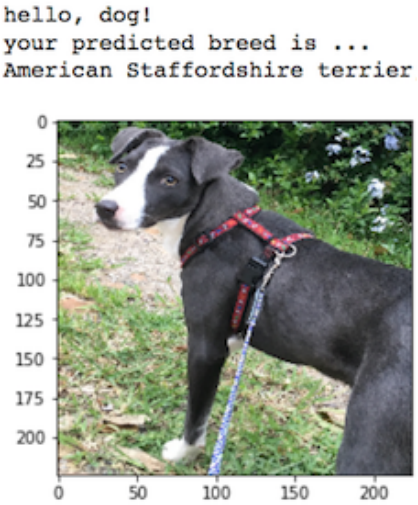
Domain background

Image recognition, as a subset of computer vision, allows machines to gain an understanding of digital images using artificial intelligence algorithms. The improvement in the image recognition field enables us to automatic relevant daily tasks, enhance efficiency and bring fun into our daily life. This capstone project will investigate dog breed identification and build an algorithm to produce an estimation of a dog breed given a doge image or dog breed that best ensembles the provided human image.

Problem Statement

The objective of this project is to develop a dog breed identification algorithm that could be used as part of a mobile or web app. Taken user-supplied images as input, the program would provide the estimation of the dog's breed. When the algorithm detected a dog in the supplied image, it will provide the estimation of the dog breed. However, if a human is detected, it will provide an estimate of the dog breed that is most resembling. The image below displays the potential sample output of the finished project.

Figure 1: Sample output [1]



Datasets and Inputs

Dog images are downloaded for dog breed identification. 133 distinct breeds will be included and separated into a training set, validation set, and testing set. A total of 6,680 images are included for model training, 835 images are included for validation, and 836 images are included for testing. For training, Alaskan Malamute has the highest number of images which contains 77 images, and Norwegian Buhund has the lowest number of images which contains 26 images. Human images are included as additional testing data set. 13,233 total images of humans will be downloaded for 5,749 distinct people. Out of this dataset, George W. Bush has the highest number of images, which contains 530 images in the folder. The rest of people have as least 1 image stored in the folder.

Solution Statement

Jingyi Han
June 7th, 2021

Firstly, two pre-implemented detection models will be used to identify humans and dogs in user supply images. Specifically, OpenCV's implementation of Haar feature-based cascade classifiers [2] will be used to detect human faces in images, and the pre-trained VGG-16 Model [3] will be used to detect dogs in images. Secondly, algorithms for dog breed classification will be implemented. Convolutional Neural Networks (CNN) [4] has been a well-studied deep learning algorithm in the domain of computer vision that enabled machines to conduct tasks such as image recognition and natural language processing. Hence, a CNN algorithm is selected for the purpose of dog breed identification using user-provided images and will be implemented in two ways, from scratch [5] and using transfer learning [6] with PyTorch, as solutions to the dog breed identification problem.

Benchmark Model

A random guess will provide a correct answer approximately 1 in 133 times, which corresponds to an accuracy of less than 1%. Hence, the CNN algorithms should have an accuracy higher than the random guess. In addition, the CNN algorithm created from scratch will be used as the benchmark model. CNN algorithm implemented using transform function should improve the model performance.

Evaluation Metrics

Test accuracy will be calculated as an evaluation metric of the model performance. As suggested by the Udacity project notebook [1], certain test accuracies must be reached for each implementation. For the CNN algorithm created from scratch, a test accuracy of at least 10% must be attained. For the CCN algorithm created using transfer learning, at least 60% accuracy on the test set must be attained.

Project Design

Step 0: Import dog images dataset and human faces dataset described in Datasets and Inputs.
Step 1: Use OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images.
Step 2: Use VGG-16 pre-trained model to detect dogs in images.
Step 3: Create a CNN to Classify Dog Breeds from Scratch in PyTorch
Step 4: Create a CNN to Classify Dog Breeds using Transfer Learning in PyTorch
Step 5: Write the algorithm that accepts a file path to an image and first determines whether the image contains a human or a dog and returns the predicted breed through combining the Human detection model, dog detection model, and the CNN classification model. The algorithm would provide an error indicator when the provided image does not contain a human or bog.
Step 6: Test the algorithm.

Reference

[1]https://github.com/udacity/deep-learning-v2-pytorch/blob/master/project-dog-classification/dog_app.ipynb
[2]https://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html
[3]https://pytorch.org/vision/stable/models.html
[4]https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
[5]https://pytorch.org/tutorials/beginner/blitz/neural_networks_tutorial.html
[6]https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html