Udacity Capstone Project Report

Machine Learning Engineer Nanodegree
Developer : Sakthivel Rathinasamy
Date : 01-May-2020

Dog Identification Application Using Convolutional Neural Network(CNN)

Project Overview:

- The Dog identificaiton app is one of the default project suggested by the Udacity, which helps in learning and building skills in machine learning.
- The problem is to identify a dog if dog image is given as input, if supplied an image of a human, we have to identify the resembling dog image.
- The object is to build the model and the web app that can process real world user input images and identify an estimate of the dogs breed.
- This is a multi-class classification problem where we can use supervised machine learning to solve this problem.

Problem Statement :

Objective 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.

Exploratory Data Analysis(EDA):

Dog images dataset:
https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip

Below are the dog images split up
test Images : 836
valid Images :835
train Images : 6680
Total Images : 8531

Dog images varies with size and background.

Human images dataset:
https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip

Total Images : 13233
All the images are 250X250 pixel size and 96 dpi resolutions.

Total Folders : 5750
Single image can have one human face or multiple human face as well.

Project Design :
Step 1: Import the necessary dataset and libraries, Pre-process the data and create train, test and validation dataset.
Step 2: Human Face detector using the OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images
Step 3: Dog Face dectector using the VGG-16 model, along with weights that have been trained on ImageNet.
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 ,train and test the model.
Step 6: Dog breed classification algorithm :

i. If a dog is detected , return the predicted breed.
ii. If a human is detected , return the resembling dog breed.
iii. If neither is detected , provide output as other thing.

Step 7: Validate the algorithm using sample images and test.

Approach:

Convolutional Neural Network is a type of artificial neural network used in image recognition and processing that is specifically designed to process large pixel data. Neural Networks mimic the way our nerve cells communicate with interconnected neurons and CNNs have a similar architecture. In our project for sloving the problem we will be using CNN

Object Detection using Haar feature-based cascade classifiers is an effective object detection method.It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. In our case have both dogs and human images to test.

- Step 1: Human Face detector using the OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images.
- Step 2: Dog Face dectector using the VGG-16 model, along with weights that have been trained on ImageNet.
- Step 3: Create a CNN to Classify Dog Breeds from scratch
- Step 4: Create a CNN to Classify Dog Breeds using transfer learning

Final Metrics :

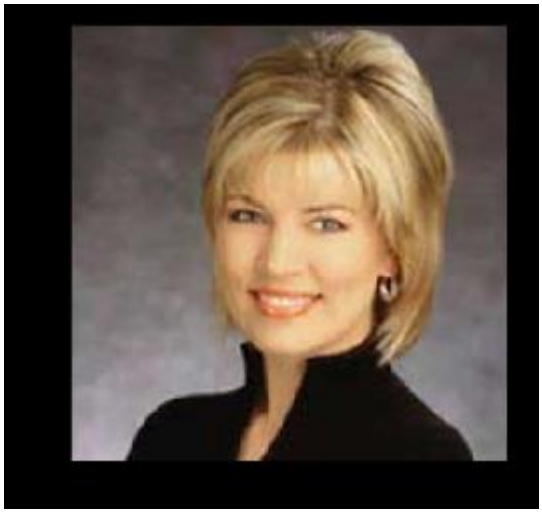
Model Scratch Test Loss is 4.7%
Model Scratch Test Accuracy from the scratch is 5%.

Model Transfer Learning Test Loss is 0.86%.
Model Transfer Learning Test Accuracy is 76%.

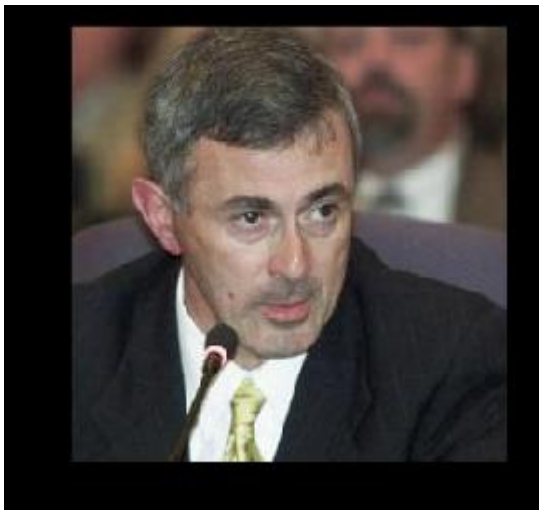
Based on the above output, Transfer learning is performing better.

Below are the output results of the running the app to test the model transfer learning:

Predicted breed: Boykin spaniel



Predicted breed: Glen of imaal terrier



Predicted breed: Akita

Predicted breed: Mastiff

Predicted breed: Mastiff

Predicted breed: Mastiff

References :

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html

https://www.kaggle.com/c/dog-breed-identification

https://www.mathworks.com/help/deeplearning/ref/resnet101.html

https://www.programcreek.com/python/example/104830/torchvision.transforms.Normalize

https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035