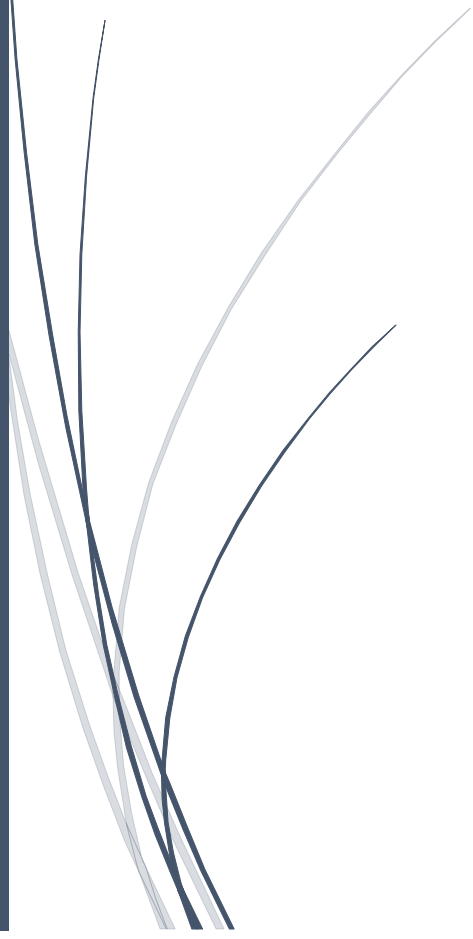
10/26/2020

Dog Breed Classifier

Capstone Proposal



Jeremiah Ifeanyichukwu
UDACITY MACHINE LEARNING ENGINEER NANODEGREE PROJECT

Domain Background

The field of computer vision deals with providing means for computers to understand and identify the information contained in images in a way similar to how humans do. This has been applied in different ways like helping self-driving cars make sense of their surroundings, detecting facial features in pictures, detecting spatial features for augmented reality, and as in the case of this project, recognizing and classifying (the content of) images to provide information for machine processes or human consumption.

The usefulness of image recognition for humans directly is underscored by the fallible nature of human memory and the sheer amount of information available. This project which deals with the classification of dogs by their breeds, is a good instance. At the time of writing, the American Kennel Club recognizes over 193 breeds of dogs [1]. An average person will find identifying the different breeds to be a daunting task. This is where a good image recognition solution will shine through.

Convolutional Neural Networks (CNNs) are deep learning algorithms inspired by the workings of the visual cortex of the brain. Popular CNN architectures like ResNet50 and AlexNet have been shown to deliver great performance on the problem of image classification [2]. This performance can likely be replicated to some degree for the particular problem tackled in this project**.**

Problem Statement

This project aims to create an android application that will serve as a pipeline for processing user-supplied images and classifying them by dog breeds based on the image content. Specifically, when supplied any given image by a user, the app should provide an output based on the following scenarios:

- If a dog is detected in the image, an estimate of the dog's breed is provided.
- If a human is detected, an estimate of the dog breed that the human most resembles should be the output.
- If neither a human nor a dog is detected, the app will present an error to that effect.

Datasets and Inputs

Inputs

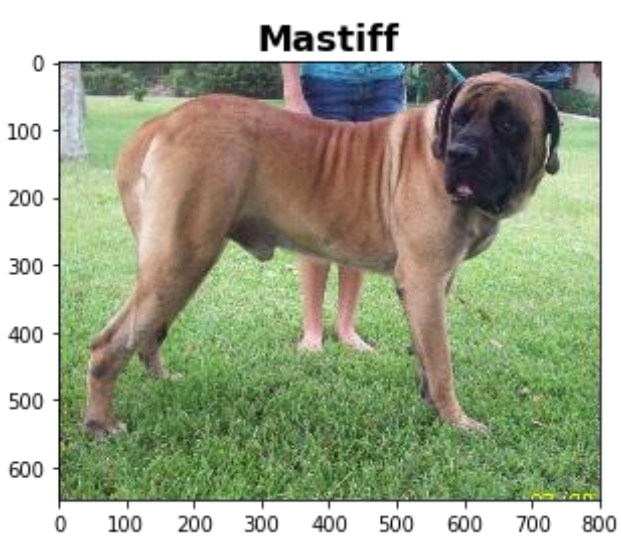
The inputs for the final solution will be any user-supplied image a phone camera or sourced from a user's gallery, etc.

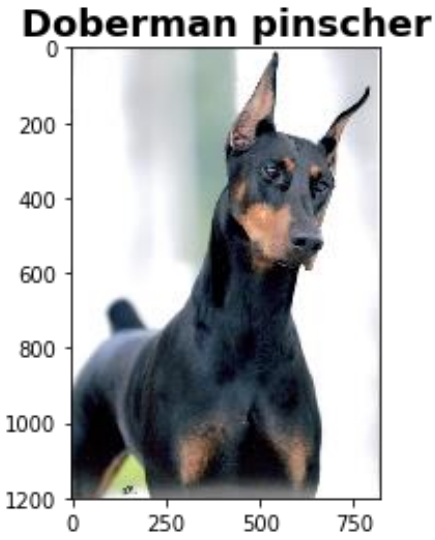
Datasets

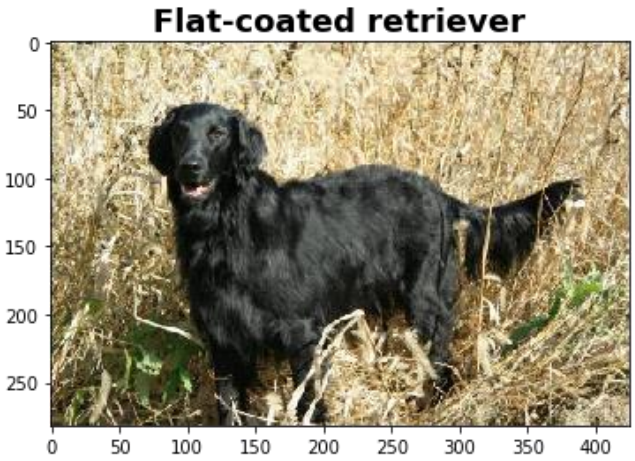
The ML algorithms used for this project will be trained, validated, and tested on two datasets gotten from Udacity, which contain images of dogs of different breeds and humans respectively.

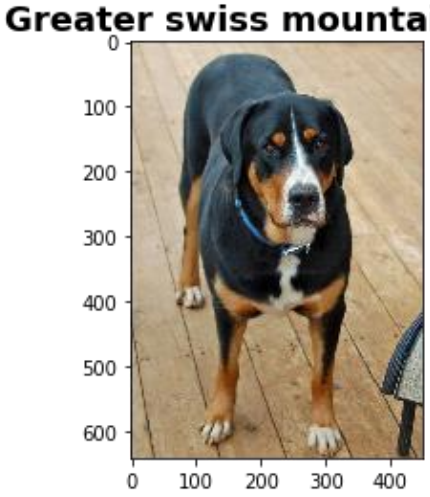
The Dog Dataset

The dog dataset contains a total of 8,351 dog images classified by their 133 different breeds. These images will be used to train the algorithm for classifying the dog breeds. Sample dog images from the dataset are shown below.

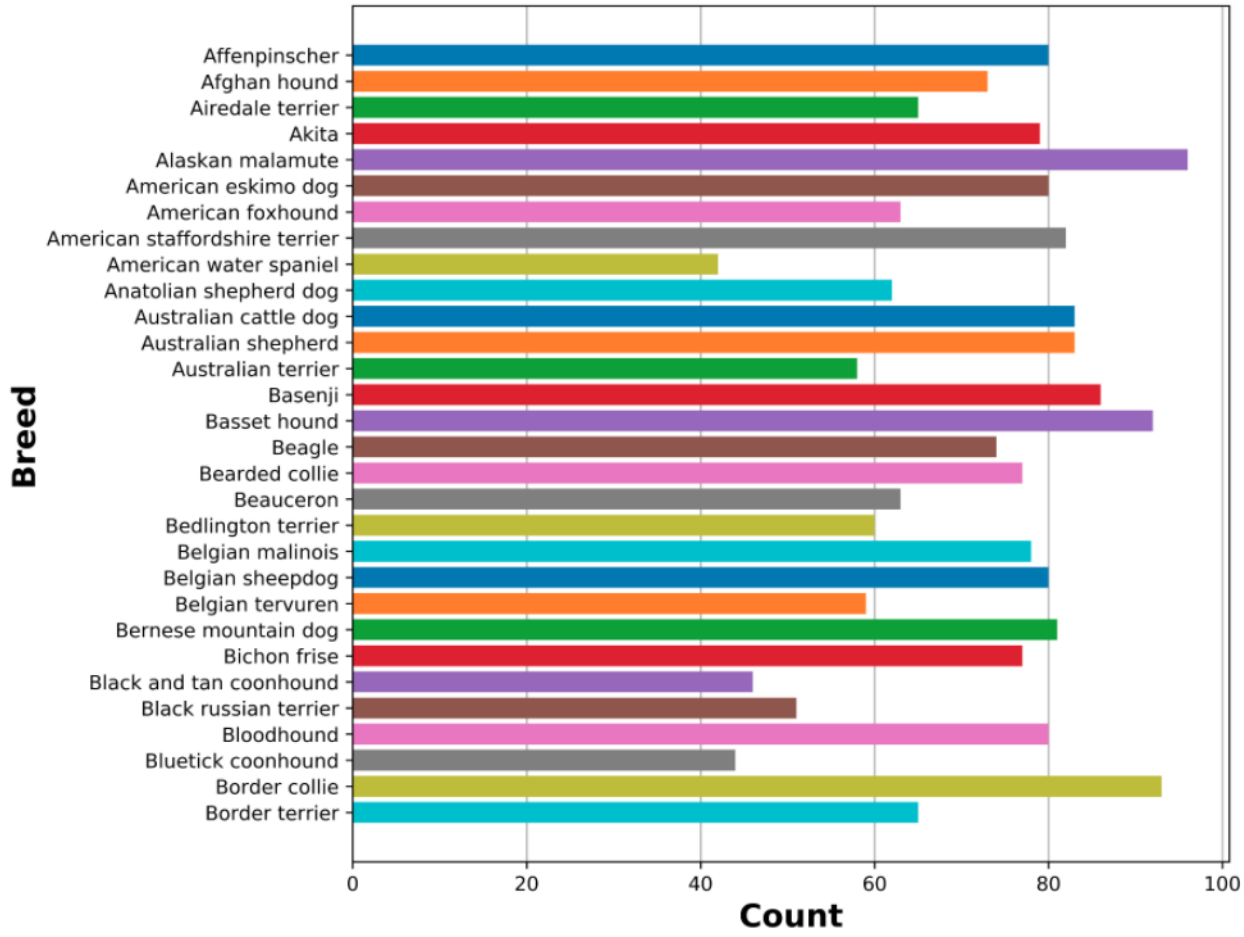

Mastiff


Doberman pinscher


Flat-coated retriever

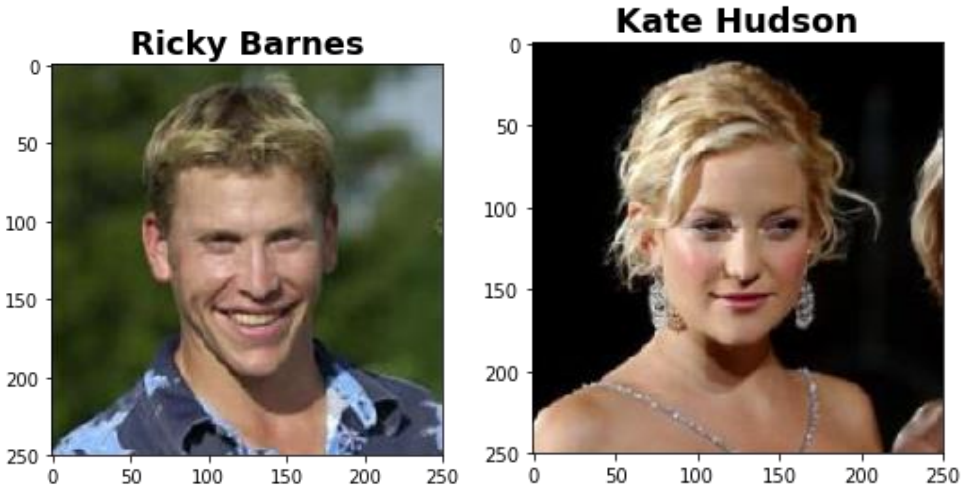

Greater swiss mountain dog

The image below shows the breed distribution of a sample of 30 breeds



THE HUMAN DATASET

The [human dataset] contains 13233 pictures of humans. There is no classification needed here as these will be used for estimating the breed a human face most resembles, and this is quite subjective. The human faces will however be used to test the accuracy of the human/dog detector. Here are sample images from this dataset.



Proposed Solution

When the user first supplies an image, the solution will check the image to detect either a human or a dog. The human face detection will be handled using OpenCV's implementation of Haar feature-based cascade classifiers (to determine if the supplied image contains a human face). If a human face is detected, a trained CNN will make an estimate of the breed of dog that human most resembles from the set of 133 breeds the CNN will be trained with. If no human face is detected, the image will be scanned to see if it contains a dog, using a pretrained CNN like ResNet50 or GoogLeNet depending on which performs better. If a dog is present in the image, an estimate will then be made about the dog breed using the custom CNN model trained in this project. Otherwise, the app will indicate that the image has no dog or human face present. This logic will be contained in an android application for ease of use.

Benchmark model

The model trained in this project will be benchmarked against the results of a dog breed classification project by Stanford researchers LaRow *et al* [3]. They also attempted to tackle this problem with the use of a CNN. Their model achieved an accuracy of 52%.

Evaluation metrics

Since the model is to be deployed in a real-time prediction pipeline, the performance of the model generated will be measured by its accuracy. For more context, the Top-5 and Top-10 accuracy (how often the correct breed is identified in the top 5 or top 10 guesses) may also be observed.

Project Design

The workflow to create the solution can be broken down into the following steps:

- In-depth data analysis and preparation:
  The images in the dataset will be analysed and any necessary preprocessing steps for better model performance, such as transforms and scaling, will be investigated.
- Human face detection:
  The OpenCV face detection technique mentioned in the proposed solution will be implemented and the performance tested against the datasets given.
- Dog detection:
  Different pretrained models such as VGG-16, ResNet50 and GoogLeNet will be tested for detecting whether an image contains a dog or not. The best performing model will be noted.
- Implementation and testing Dog breed classification model(s):
  Here a CNN to classify the dog images by breed will be created from scratch. However, since this problem has already been tackled with similar architecture (CNNs), transfer learning from existing Neural Net configurations will also be explored to get a best performing model.
- Transfer models to TensorFlow-Lite:
  The models to be used for the various steps of the pipeline will need to be converted to a format suitable for deployment in a native android mobile application. They will be converted using the TensorFlow Lite converter.
- Implement android recognition pipeline:
  The features developed above will be combined into a single flow in an android app for real-time inference.

REFERENCES

[1] "Dog Breeds - Types of Dogs - American Kennel Club," American Kennel Club, [Online]. Available: https://www.akc.org/dog-breeds/. [Accessed 24 October 2020].

[2] N. Sharma, V. Jain and A. Mishra, "An Analysis Of Convolutional Neural Networks For Image Classification," *Procedia Computer Science,* vol. 132, pp. 377-384, 2018.

[3] W. LaRow, B. Mittl and V. Singh, "Dog Breed Identification," 2016. [Online]. Available: https://web.stanford.edu/class/cs231a/prev_projects_2016/output%20(1).pdf. [Accessed 26 October 2020].