Project Proposal

For my capstone project, I propose to build a **visual dog breed detection and classification system**. This system will take as input an image with a dog and **be able to detect the appropriate dog breed.** Additionally, it will also **detect a bounding box in the image within which the dog(s) are located.** I plan to deploy this application as a mobile-compatible web application.

Aside from being a fun, lightweight application for dog lovers, this project also helps me explore state of the art visual classification and detection techniques in the field of Machine-Learning based computer vision.

Background

Deep Neural Networks have revolutionized the field of unstructured data-driven learning (especially Computer Vision and Natural Language Processing). Beginning with AlexNet (see paper here) in 2012 which won the ImageNet Large Scale Visual Recognition Challenge by a significant margin. This work marked the beginning of the modern Deep Learning driven AI era. Since then the field has made significant strides in the field of image classification. Additionally, the architectures which power image classification can be tweaked and reprovisioned to also solve the problem of object detection. The first major and popular breakthrough in this space which offered an end-to-end solution for object detection based solely on image and corresponding labels was Yolo. Since then several families of end-to-end object detectors (including many versions of Yolo) have pushed this research forward.

Additionally, the availability of datasets and the general open-source community have helped speed up the pace of progress. Today, it is easy to find bespoke datasets which can be used as benchmarks for a wide range of visual tasks. Dog breed identification and detection is one such task. Given **the similarity between various dog breeds as well as the existence of many mixed-breeds, this task is not trivial for a traditional vision based learning system**. However, several open-source datasets are now available to help researchers and engineers finetune their solutions.

Datasets

I propose to use the datasets described in my earlier data collection to train a novel visual dog-breed classifier and detector. A brief description of these datasets is provided below.

Stanford Dogs Dataset

The stanford dogs dataset consists of 20,580 images of 120 different dog breeds. The dataset consists of both bounding boxes (for object detection) as well as dog breed labels. Based on discussions and leaderboards at paperswithcode, this dataset has become an important benchmark for dog breed classification.

Tsinghua University Dogs Dataset

Tsinghua University Dogs Dataset is another important benchmark dataset for dogbreed classification and detection. This dataset consists of 70428 images of 130 different dog breeds. Each dog breed has anywhere from 200 to 7449 images represented in this dataset and the sample sizes are roughly representative of frequencies of dog breeds found in China.

As is the case with the Stanford Dogs dataset, this dataset also consists of class labels as well as bounding boxes.

Kaggle Dog Breed Classification Dataset

This is yet another dataset for 20,000 images of 120 different breeds.. However, a drawback of this dataset is that it only consists of labels of dog breeds and not bounding boxes. However, on the plus side, the data is pre-cropped so each image represents mostly only the dog.

Since this dataset is associated with a Kaggle competition, downloading it programmatically requires a Kaggle account. Please follow instructions here on how to use the Kaggle API.

My goal is to use a combination of these datasets to train a visual classifier and detector which can be deployed in a mobile compatible webapp.

Technical Approach

The problem I'm trying to solve involves both detection and classification. With this in mind, this project gives me the opportunity to explore state-of-the-art (SOTA) techniques. Some of these are described below.

Classification

As described above, **multi-class visual classification** is the problem of taking an input image and returning a list of probabilities for the image containing a predefined set of labels.

Vision Transformers

Vision Transformers were the next big breakthrough in the field of image classification. After the advent of the transformer architecture in 2017, vision transformers demonstrated that the transformer architecture could effectively be used for SOTA performance on image classification tasks. More generally, vision transformers can be used to extract image embeddings/features which are useful for a wide variety of downstream tasks. Vision Transformers remain (to this day) SOTA for image classification. I plan on exploring this approach for the classification section of my task.

ConvNeXt

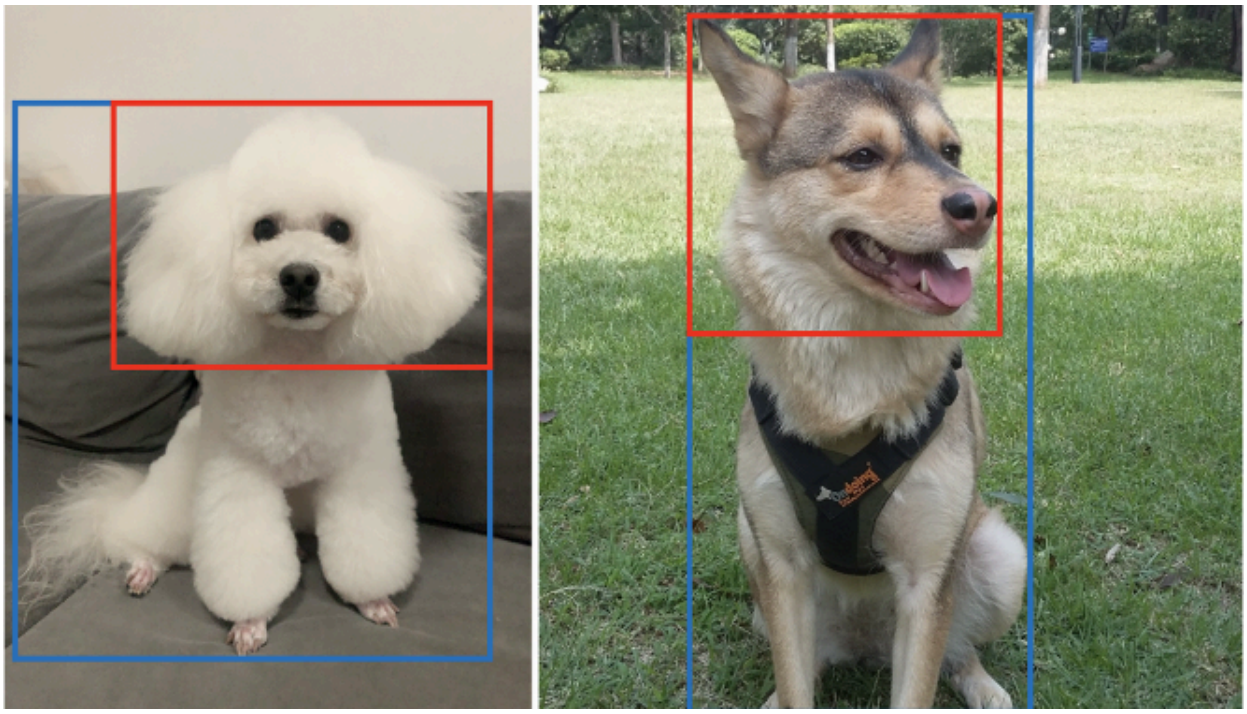
ConvNeXt is a family of purely convolutional architectures released in 2022 which are benchmarked competitively vision transformers. The theory was that pre-training convolutional architectures on the same amount of data as vision transformers can help them close the gap in performance. I also want to explore this family of architectures for image classification.

MobileNet-V2

MobileNet family of models is a family of architectures which are optimized for mobile performance. Since my proposal is to build a mobile friendly web app, it is useful to at least benchmark MobileNet performance for the classification task.

Object Detection

Object detection is the problem of taking an image and regressing bounding box coordinates around predefined categories of objects. Object detection tends to be a harder task as the network is responsible for doing both classification and detection.



Images courtesy of Tsinghua University Dogs Dataset

Some approaches I'd like to explore are described below.

FasterRCNN

FasterRCNN is a family of region proposal networks which were. highly competitive in the mid 2010s. While they are no longer SOTA, they do offer a simple to use and potential benchmark for my project.

CenterNet

CenterNet is the approach I would like to focus on for my project. It is a single stage object detector which uses a novel approach to object detection and tracking. It essentially builds on research in the Objects-as-Points paper and treats each object as a triplet. Each object is represented as **a keypoint and a pair of corners.** Compared to traditional anchor-based approaches in object detection, this approach tends to be much less computationally expensive.

Computational Requirements

Despite the networks above being fairly computationally expensive, training a benchmark on a small (few thousand images) shouldn't take more than a few hours on Google Colab Pro

(especially utilizing the A100 option). I plan on utilizing a smaller dataset sample to explore the various approaches via Google Colab Pro.

Full training is likely to take a few days on high-performance GPUs. For this I plan on utilizing high-performance GPUs available on AWS Sagemaker. An ml.g5.2xlarge node should be more than sufficient for training the full end-to-end classification and detection system.

Deployment

I plan on using a combination of Gradio and AWS Sagemaker to deploy the final mobile-compatible web-app. Use of Gradio simplifies the UI and the necessary boilerplate required to build a mobile-compatible web app. AWS Sagemaker meanwhile offers a quick and easy solution to deployment at scale.