

Proposal

Domain Background

Applications of image recognition and classification have started to become integrated into our lives from customs at the airport to surveillance for no-checkout convenience stores ^[1]. Further research and development are taking place in all fields from entertainment, automobile, healthcare, security, retail, construction, beauty, agriculture and many more. We can expect to see more and more applications in the future disrupting the current industries and leveraging technology to support human workers.

In this project, a dog breed classification application will be created using transfer learning on VGG-16 ^[2], a pre-trained convolutional neural net (CNN) model. When supplied with an image of a human, the application will identify the resembling dog breed. Although this project is more towards entertainment, there are other academic research examples of more serious application ^[3] which applies more advanced convolutional neural net (CNN) and deep learning models to automatically identify, count and describe wild animals from cameras. This research is one example of how classification of animals helps us understand the natural ecosystems without disrupting the habitat. Understanding CNN (and similar models) is the first step towards image classification and application of these models can help solve wide range of problems that can be easily integrated into other existing systems and services.

¹ “Amazon Go.” Amazon, Amazon, www.amazon.com/b?ie=UTF8&node=16008589011.

² Very Deep Convolutional Networks for Large-Scale Image Recognition K. Simonyan, A. Zisserman arXiv:1409.1556

³ Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Technical Report 07-49, October, 2007.

Problem Statement

The goal of this project is to create an application that does the following according to the type of input image.

- if a dog is detected in the image, return the predicted breed.
- if a human is detected in the image, return the resembling dog breed.
- if neither is detected in the image, provide output that indicates an error.

To accomplish this goal, we must be able to detect dogs, detect human, classify dogs, classify human as dogs. This means we must develop (or use existing) 3 models which consist of dog detection, human detection and dog breed classification. There is no requirement on the accuracy of the resembling dog breed as there are no set method or metric to classify a human to a resembling dog breed.

Datasets and Inputs

The data set selected by Udacity for this project include a Udacity unique dog breed dataset ^[4] and LFW ^[5] human data set ^[6] originally created and maintained by researchers in University of Massachusetts, Amherst.

The dog breed dataset consists of test, train and valid file each with the same set of 133 different dog breed files with images inside. The dog breed dataset will be used to test the pre-trained model to detect dogs as well as testing the CNN to classify dog breed. Each dog breed folder contains 8 images of the same dog breed. Image size varies however pre-processing of image is not part of this scope (we plan to use the images as they are).

⁴ “dog dataset” <<https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip>>

⁵ Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Technical Report 07-49, October, 2007.

⁶ “human dataset” <<https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip>>

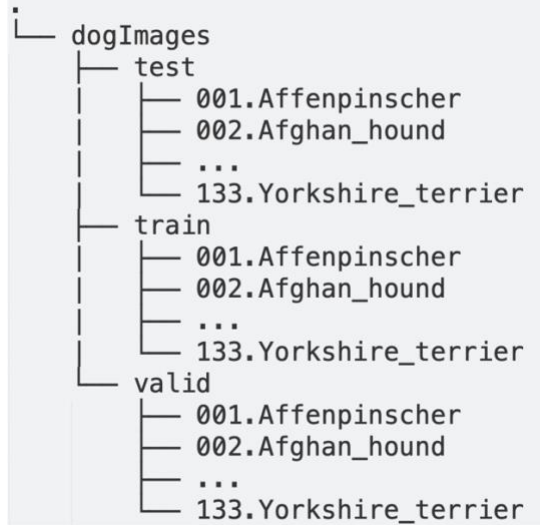


Figure 1 : Dog image folder structure



Figure 2 : LFS human image folder structure

The human face dataset consists of files with the person's first and last name and an image file (or files) in them. The dataset will be used for human face detection for the pretrained model. We don't need to classify people, which means we don't need many images of the same person, so we can conclude that the number of image data is enough for testing human detection model. Overall, we can say that the data provided by Udacity is enough to do all of the task stated in the problem statement.

Solution Statement

In the problem statement, we have specified the 3 tasks we must complete. For each of the task, the CNN algorithms/libraries that will be used will be introduced.

First, we must define a pre-trained model for human detection and dog detection. The project will use OpenCV ^[7] implementation of Haar feature-based cascade classifiers ^[8] for detecting human faces and pre-trained detector haarcascade_frontalface_alt.xml ^[9]. For the dog

⁷ "OpenCV." OpenCV, opencv.org/.

⁸ Viola, P., and M. Jones. "Rapid Object Detection Using a Boosted Cascade of Simple Features." Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 15 Apr. 2003, doi:10.1109/cvpr.2001.990517.

⁹ "haarcascade frontalface" OpenCV, <https://github.com/opencv/opencv/tree/master/data/haarcascades>.

detection, the VGG-16 implemented by torchvision^[10] will be used. We assume the accuracy of detection to be at least 70% (top-1 accuracy using ImageNet^[11]) and we will confirm using the appropriate datasets provided by Udacity.

Next, we will create a CNN model for dog breed classification. We will first develop the model from scratch and evaluate. We will also the minimal test accuracy to pass this portion is set to 10%. The reason for the low accuracy setting is due to the fact that there are set of classes that are very similar visually (even normal people cannot distinguish some categories). There are more than 100 classes, which makes the random guess accuracy to around 1%, so an accuracy of more than 10% is very unlikely to be due to chance. After some trial with models developed from scratch, we will create another model by transfer learning using base model of VGG-16. The minimal test accuracy to aim is set to 60% (by Udacity).

Benchmark Model

The model that will be used as base for dog breed classification (transfer learning) is VGG-16. Some models that can be used as benchmark includes AlexNet^[12], Inception-v3^[13], ResNet-50^[14] to name a few. These models can also be used as benchmarks for detection. The models introduced are very commonly used as benchmark across multiple papers and the comparison of the ImageNet 1-crop error rates (224x224) is provided on torchvision model documentation page.

¹⁰ "Torchvision.models." Torchvision.models - PyTorch Master Documentation, pytorch.org/docs/master/torchvision/models.html.

¹¹ "ImageNet." ImageNet, www.image-net.org/.

¹² Krizhevsky, Alex, et al. "ImageNet Classification with Deep Convolutional Neural Networks - Semantic Scholar." Undefined, 1 Jan. 1970, www.semanticscholar.org/paper/ImageNet-Classification-with-Deep-Convolutional-Krizhevsky-Sutskever/2315fc6c2c0c4abd2443e26a26e7bb86df8e24cc.

¹³ Szegedy, Christian, et al. "Rethinking the Inception Architecture for Computer Vision." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, doi:10.1109/cvpr.2016.308.

¹⁴ He, Kaiming, et al. "Deep Residual Learning for Image Recognition." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, doi:10.1109/cvpr.2016.90.

Evaluation Metrics

This project is unique in the fact that no one can give a "correct" answer to which dog breed a human represent, therefore we can only numerically evaluate human detection, dog detection, and dog classification.

The main evaluation metric that will be used is accuracy and error rate.

	Actual = Yes	Actual = No
Predicted = Yes	TP (true positive)	FP (false positive)
Actual = No	FN (false negative)	TN (true negative)

Given the above confusion matrix, accuracy and error rate is defined as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, Error Rate = \frac{FP+FN}{TP+TN+FP+FN}$$

Project Design

We will follow the dog_app.ipynb file provided by Udacity ^[15] to create this dog breed classification app. The steps are as follows:

1. Import Datasets & Preprocess images as necessary
2. Detect Humans (from pre-trained models)
3. Detect Dogs (from pre-trained models)
4. Create a CNN to Classify Dog Breeds (from scratch with accuracy > 10%)
5. Create a CNN to Classify Dog Breeds (using transfer learning with accuracy > 60%)
6. Write & Test the Algorithm
7. Evaluate & Improve
8. Deploy the model & create API to access model through web app

¹⁵ Udacity. "Udacity/Deep-Learning-v2-Pytorch." GitHub, 28 May 2019, github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification.

