DOG BREED CLASSIFIER (CNN)

PROJECT PROPOSAL

BY NINAD DESHPANDE

November 20, 2020

DOMAIN BACKGROUND

Due to the advancements in computing technologies, machine learning today is not like machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being explicitly programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data and they indeed learn from previous computations to produce reliable, repeatable decisions and results. It's a science that's not new but one that has gained fresh momentum.

While many machine learning algorithms have been around for a long time, image classification is one of the hot topics in the machine learning field. When it comes to image classification, the main advantage of using Convolutional Neural Networks (CNNs) compared to its predecessors is that it automatically detects the important features without any human supervision. Image classification with CNN saw the light of day in 1994 with Yann LeCun's LeNet5. Later it received a huge boost in 2012 with AlexNet by Alex Krizhevsky who won the ImageNet competition. CNNs are effective as the concept of dimensionality reduction suits the huge number of parameters in an image.

A paper on convolutional neural networks (CNNs) to classify dog breeds published by Hsu, David from Stanford University made me realize that this classification is in fact challenging especially due to the minimal inter-class variations and existence of a huge variety of species, also to research on a technology to classify them is crucial. I chose this project to utilize CNNs in order to identify the canine's breed for a given dog image. Moreover, if supplied an image of a human, the code should identify the resembling dog breed. The idea is to build a pipeline that can process the real-world user supplied images and identify an estimate of the canine's breed.

PROBLEM STATEMENT

The aim of the project is to build a machine learning model which can be used to process the user-supplied images and perform the following two tasks:

- **Dog face detection:** Algorithm will identify an estimate of the canine's breed, if an image of a dog is supplied.
- **Human face detection:** Algorithm will identify the resembling dog breed, if an image of a human is supplied.

DATASETS AND INPUTS

The input must be an image. The dataset consists of images of dogs and humans and is provided by Udacity.

- **Dog images:** The dataset has 8351 images which are sorted into train with 6,680 Images, valid with 835 Images and test with 836 images. Each directory has 133 subdirectories corresponding to each dog breed. These images are of different sizes and backgrounds and as the number of images provided for each breed is not the same, the data is not balanced.
- **Human images:** The dataset contains 13233 human images sorted by names of size 250x250 with different backgrounds and postures. As the number of images provided in each folder is different, the data is not balanced.

SOLUTION STATEMENT

CNNs are often used for image classification and recognition because of its high accuracy and we will use it for this multiclass classification as well. The solution will consist of the following three steps.

- Detect dog-images using a pretrained VGG16 model,
- Detect human images using an existing algorithm like OpenCV's implementation of HAAR feature based cascade classifiers,
- Identify an image as a dog or a human, pass it to a CNN to further processing and classification.

BENCHMARK MODEL

- The CNN model created in this project must have the accuracy of at least 10%. As there are 133 dog breeds, a random guess will give a correct answer once in 133 tries, which is equivalent to the accuracy of less than 0.80%.
- To make a good model, the CNN created with transfer learning must have at least 60% accuracy.

EVALUATION METRICS

Multi class log loss will be used for the model evaluation. In this case, accuracy is a not a good indicator to measure the performance because of the data imbalance.

In case of a classification model, we have a multitude of metrics of performance available to optimize our models, quantify their performances, compare them and improve them. The log loss metric takes into account the probabilities underlying our model, and not only the final output of the classification. Thus, it will consider the uncertainty of prediction based on how much it fluctuates from the actual label. This will help us in the model evaluation.

PROJECT DESIGN

1. Import dataset
2. Create train, test and validation dataset after preprocessing the data
3. Detect humans using OpenCV's HAAR feature based cascade classifiers
4. Detect dogs using a pretrained convolutional network for classification and detection, VGG16 model
5. Create a CNN to classify dog breeds from scratch.
6. Create a CNN to classify dog breeds using transfer learning followed by training and testing the model.
7. Write an algorithm which returns the predicted breed, if a dog or a human face is detected and an error when neither of them is detected in an image.

REFERENCES

- GitHub repository of the original project, https://github.com/udacity/deep-learning-v2-pytorch/blob/master/project-dog-classification/
- Eugenio Culurciello. The History of Neural Networks. dataconomy.com. Apr. 19, 2017, https://dataconomy.com/2017/04/history- neural- networks/
- Hsu, David. "Using Convolutional Neural Networks to Classify Dog Breeds", Stanford University. http://cs231n.stanford.edu/reports/2015/pdfs/fcdh_FinalReport.pdf
- PyTorch Documentation: https://pytorch.org/docs/master/
- Muhammed Talo. "Convolutional Neural Networks for Multi-class Histopathology Image Classification", Munzur University Turkey. https://arxiv.org/ftp/arxiv/papers/1903/1903.10035.pdf