DOG BREED CLASSIFIER (CNN)

PROJECT PROPOSAL

BY NINAD DESHPANDE

15th November 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 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 and 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 goal of the project is to build a machine learning model that can be used to process the real-world, user-supplied images and performs the following two tasks:

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

DATASETS AND INPUTS

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

- **Dog images dataset:** The dataset has 8351 total images which are sorted into train (6,680 Images), valid (835 Images) and test (836 Images) directories. Each directory has 133 folders corresponding to dog breeds. The images are of different sizes and backgrounds. The data is not balanced as the number of images provided for each breed varies.
- **Human images dataset:** The dataset contains 13233 total human images which are sorted by names (5750 folders). Images are of size 250x250 with different backgrounds and postures. The data is not balanced as the number of images provided in each folder varies.

SOLUTION STATEMENT

We can use the Convolutional Neural Network (CNN) for this multiclass classification. CNN is a deep learning algorithm which can take an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and differentiate one from the other. The solution involves the following three steps.

- To detect the dog-images we will use a pretrained VGG16 model.
- To detect the human images, we can use an existing algorithm like OpenCV’s implementation of HAAR feature based cascade classifiers.
- After the image is identified as a dog or a human, we can pass this image to a CNN which will process the image and predict the breed that matches the best out of 133 breeds.

BENCHMARK MODEL

- The CNN model created from scratch must have accuracy of at least 10%. This can confirm that the model is working as a random guess will provide a correct answer roughly 1 in 133 times, which corresponds to an accuracy of less than 1%.
- The CNN model created using transfer learning must have accuracy of 60% and above.

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. Log loss considers the uncertainty of prediction based on how much it varies from the actual label, which will help in evaluating the model.

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