Machine Learning Nanodegree Capstone

Proposal: Dog Breed Classifier

Mohaned Mohamed El-Hadidi

1. Domain background

Even for humans, recognizing dogs according to their breed is a difficult process. In machine learning, the Dog breed classifier is a well-known issue. If a dog image is provided as input, the problem is to identify the resembling dog breed. There are hundreds of breeds, demonstrating that classification is a difficult undertaking because people are incapable of remembering all of them and distinguishing between similar kinds. As a result, a computer is required to complete this task. To complete this goal, a convolutional neural network (CNN) will be deployed.

2. Problem statement

The project's goal is to create a pipeline that can process real-world, user-supplied photos. Given a picture, the algorithm will determine an estimate of the dog's breed. When the image is of a human, the algorithm will select a dog breed that closely matches the person. If neither a dog nor a human can be found, an error message is displayed. As a result, the models in place should be capable of recognizing a dog or human in an image, classifying the dog by breed, and classifying a dog breed that resembles the human.

3. Datasets and inputs

Udacity provided all the datasets used to train, test, and validate the CNN model. The dataset has pictures of dogs and humans.

Dog images dataset: The dog image dataset has 8351 total images which are sorted into train (6,680 Images), test (836 Images) and valid (835 Images) directories. Each of this directory (train, test, valid) have 133 folders corresponding to dog breeds. The images are of different sizes and different backgrounds, some images are not full-sized.

Human images dataset: The human dataset contains 13233 total human images which are sorted by names of human (5750 folders). All images are of size 250x250. Images have different background and different angles.

Both datasets are not balanced as for the dog dataset the number of images provided for each breed varies as well as for the human dataset.

A single portrait of a dog of the matching breed appears in the bulk of the photographs. However, there is a small percentage of photographs in the collection that include both dogs and humans, as well as other animals. The problem is determining whether there is a dog in the image while avoiding misidentifying other animals or humans as dogs, or vice versa.

4. Solution statement

The solution is to create a CNN model that can predict the breed of a dog in a photograph as accurately as possible. To do so, it must first be determined whether a person or a dog is shown in a photograph. The matching dog breed is detected if a person is present. If there is a dog present, a breed estimate is given. We may also leverage existing algorithms to recognize human photos, such as OpenCV's implementation of Haar feature-based cascade classifiers. We could also use a VGG16 model that has been pre-trained to recognize dog photos.

5. Benchmark model

The accuracy of the Convolutional Neural Network that we build from the ground up should be at least 10%. This was chosen in consideration of the difficulty of distinguishing distinct dog breeds even for humans, and so getting high accuracy using bare-CNN is not a simple task. To be effective in a dog breed classifier app, the pre-trained model must have an accuracy of at least 60%.

6. Evaluation metrics

The percentage of accuracy would be an appropriate metric to determine how well our model is working in our problem of classifying the proper dog breed, because it provides a solid measure of how well our model is performing, bound between 0 and 100.

7. Project design

The workflow for the project to approach the solution is as follows:

- Step 1: Import the necessary dataset and libraries, the dataset is provided by Udacity and will be obtained and analyzed for our use case.
- Step 2: Detect human faces using OpenCV’s implementation of Haar feature based cascade classifiers.
- Step 3: We'll employ one of the top pre-trained models from the ImageNet competition, specifically the VGG-16 Model, as described in the section "Solution Statement." The model was trained on a massive dataset with 1000 categories.
- Step 4: Create a CNN to classify dog breeds from scratch, train, validate and test the model. We approach this problem by using pre-trained weights from a ResNet-152 model as we will preprocess the image data by flipping, rotating, resizing and cropping them to have a variety, we will use ResNet-152 model as its convolutional layers trained with such big dataset will help us build our classifier.
- Step 5: CNN model is created using transfer learning to classify the images and test accuracy should be greater than 60%.
- Step 6: Write an algorithm that if a dog is detected, this function returns the projected breed. If a human is identified, the dog breed that most closely resembles it should be returned. If no dog or human is found, an error message is displayed.
- Finally testing the solution model with images from the internet or from datasets.

8. References

1. Original repo for Project - GitHub: https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification
2. Udacity: The dog dataset. url: https://s3-us-west-1.amazonaws.com/ udacity-aind/dog-project/dogImages.zip.
3. Udacity: The human dataset. url: https://s3-us-west-1.amazonaws.com/ udacity-aind/dog-project/lfw.zip.
4. OpenCV. Face Detection using Haar Cascades. https://docs.opencv.org/trunk/db/d28/tutorial_cascade_classifier.html.
5. Ayanzadeh, Aydin and Sahand Vahidnia. Modified Deep Neural Networks for Dog Breeds Identification. May 2018. url: https://www.researchgate.net/profile/Aydin_Ayanzadeh2/publication/325384896_Modified_ Deep_Neural_Networks_for_Dog_Breeds_Identification/links/ 5cd0345ea6fdccc9dd90690c/Modified-Deep-Neural-Networks-forDog-Breeds-Identification.pdf
6. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).