Machine-learning engineer course

Capstone project proposal: dog breed identifier

1. Domain background

Identifying and classifying contents of digital pictures is subsumed under the term computer vision. Computer vision is a branch of machine learning that tries to emulate the understanding and perception of digital images as a human would.
Historically, image classification was a very specialized and resource intensive topic reserved to purpose build and resource heavy application like cancer prediction.
With the advent of deep neural networks, a framework became available that can abstract those problems in a fundamentally different way.
Neural networks can learn from labeled data without engineering the actual meaning of the results into the program. In the end it comes down to recognizing patterns within images. [1]
In my field of work (automotive) computer vision has become a particularly important topic, as autonomous driving based on camera input is almost impossible without a technology like neural networks. This project is the perfect choice to augment my knowledge on this field.

2. Problem Statement

The problem this project addresses is the classification of different dog breeds based on digital images of the specific breed.
The challenge will be to develop a robust classifier that is able to differentiate different dog breeds and detect human faces in case an image with human content is uploaded.

3. Datasets and Inputs

The datasets used in this project are provided by udacity. A total number of 13233 human images and 8351 dog images together with their label are provided.
The dog datasets are available already split as train, validation and test images.
Those will be used directly after normalization in order to train and test the performance of the neural network.
The human faces images consist of individual folders with pictures of personalities.
The human faces dataset will be used to detect if the picture contains a human. The face-detection algorithm will return the number of detected faces or zero in case no human face is recognized.

4. Solution Statement

The problem will be solved using a convolution neural network (CNN).
First, the identification of human faces will be done using the Haar feature-based cascade classifier [2].
Next, we will use the pre-trained VGG-16 network to identify dogs in images generally. The VGG-16 network has 1000 varied categories as output. Some of those categories (index 151-268) already classify dog breeds. We will use this output to generally identify dogs in the images.

Next, a custom CNN will be built with the actual labels contained in the provided dog images. This CNN should surpass 10% test accuracy.
The final solution will be to use transfer learning (using a large pretrained network in combination with a custom fully connected output layer) to combine both the VGG-16 network with our custom labels. A 60% accuracy or higher is the aim.

5. Benchmark Model

The benchmark model for image classification will be the VGG-16 itself.
“VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous models submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another” [3].

The model must accurately predict dogs from images. This will enable us to compare the results of the VGG-16, the custom CNN and finally the transfer learning combination of VGG-16 and a custom output layer.

6. Evaluation metrics

The main evaluation metric will be accuracy. The accuracy is defined as the sum of true positives and negatives divided by the sum of all predictions.
However, when training the CNNs, categorical cross entropy will be used as internal metric for optimization as log loss generalizes better to a multiclass problem [4].

7. Project Design

The workflow of the project follows the general outline:

- Step 0: Import Datasets
  All relevant datasets will be download and loaded

- Step 1: Detect Humans
  OpenCV will be imported to identify human faces on images. In order to identify faces, there will be some light preprocessing (gray-scaling etc.) introduced.

- Step 2: Detect Dogs
  The VGG-16 model outlined under 4. will be loaded and set up to predict labels of images. This contains normalizing and resizing of the images to the specifications expected by the VGG-16 model. The model will return indices of categories of which some are specified as dog breeds. A function will be implemented that decides, whether the index corresponds to a dog breed or not returning true or false.
  We should expect returning true statements for most dog images and false for human faces.

- Step 3: Create a CNN to Classify Dog Breeds (from Scratch)
  A convolution neural network will be built from scratch. Before building the actual model, some image processing will be applied to add variance to the images (rotation, flipping etc.)

- Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning)
  To improve accuracy, the VGG-16 model will be applied in combination with a custom output layer. This will benefit from the exceptionally good benchmark performance (pattern recognition) of the pre-trained model together with the purpose-built output layer to satisfy our classification needs.

- Step 5: Write your Algorithm
  The algorithm will be fed a path to an image, either human or dog. If a dog is detected, then the breed will be returned. Else if a human is detected, the resembling dog breed will be returned.
  If neither a dog nor a human is detected, some error will be thrown.

- Step 6: Test Your Algorithm
  A function will be provided to test the before-mentioned algorithm on at least six images. For evaluation purposes.

Bibliography

[1] I. Mihajlovic, „towardsdatascience.com,“ 29https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e 12 2020. [Online].

[2] „opencv.org,“ 29 12 2020. [Online]. Available: https://docs.opencv.org/master/db/d28/tutorial_cascade_classifier.html.

[3] „https://neurohive.io,“ 20 11 2018. [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/. [Zugriff am 29 12 2020].

[4] R. Agarwal, „towardsdatascience.com,“ 17 9 2019. [Online]. Available: https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226. [Zugriff am 29 12 2020].