Convolutional Neural Networks: Dog Breed Identification

Jianming Han

1. Domain Background

Image classification is the task of taking an input image and outputting a class (dog, car, etc.) or the probability of a class that best describes the image. It faces multiple challenges, including scale variation, viewpoint variation, intra-class variation, image deformation, image occlusion, illumination conditions and background clutter [1].

There are many applications for image classification with deep neural networks. One I'm interested in is autonomous driving, for example, CNNs can be embedded in the systems of autonomous cars to help the system recognize the surrounding of the car and classify objects to distinguish between ones that do not require any action and ones that do. I'm planning to take study in the area of autonomous driving, so I decided to choose the dog breed identification project which is prefect to make preparation and build solid foundation for my future study.

The dog breed identification problem is a classic image classification problem which can be found on Kaggle [2], totally 1282 teams have competed on this problem and shared their solutions. In Literature, some researchers also studied the problem with different solutions. Jiongxin Liu [3] proposed a part localization approach to make dog breed classification and a recognition rate of 67% was achieved. Kaitlyn Mulligan [4] studied this problem by applying a Xception CNN architecture.

2. Problem Statement

The goal of this project is to build a pipeline that can be used to for dog breed identification. The model will take any real-world, user-supplied images as input and make predictions accordingly. Specifically, the model needs two main functionalities:

- Given an image of a dog, the model will identify an estimate of the canine's breed;
- Given an image of a human, the model will identify the resembling dog breed.

To achieve these, the first question needs to be answered is whether a human or a dog is on the picture, so what we need is a human detector and a dog detector. Then we need a model to classify dog breeds, this is a multi-classification machine learning problem.

3. Datasets and Inputs

Considering that it's a supervised learning problem, the dataset used should be well labeled and include both images of dogs and human with sufficient samples.

Luckily the project is provided by Udacity directly, so the dataset is provided within the workspace.

Totally we get 8351 and 13233 images for dogs and human respectively. The dog images are classified into 133 dog categories and split into train (6680 images), validation (835 images) and test (836 images) subsets. The human images are sorted by names.

The average image quantity for each breed is about 63 (≈8351/133). And some breeds contain more and some contain less, it's slightly imbalanced among all the breeds. Also, this might not be sufficient for the model to learn enough features and patterns for each breed, but I'm not sure at the moment, we will see! Dog images are with different image sizes, resolutions and lighting conditions, some contain complete dog body while some just contain a dog head. Human images are in a standard size of 250*250, most images contain only one human face, but some contain more than one human face.

4. Solution statement

In order to conduct the dog breed classification, a model is needed to infer dog breeds. Also mentioned previously, a human face detector and a dog detector are needed to distinguish between human and dog. Basically, the pipeline consists of three parts: human face detection, dog detection and dog breed inference.

- OpenCV model will be used to detect human face in an image because OpenCV provides many pre-trained face detectors.
- To detect dog in an image, a pre-trained VGG16 model will be applied. Given an image, this VGG16 model will return a prediction for the object that is contained in the image.
- Lastly, to efficiently and accurately conduct the dog breed classification, convolutional neural network (CNN) will be applied to construct the model. In deep learning, CNN is a class of deep neural networks, most commonly applied to analyzing visual imagery [5]. It's one of the most popular techniques used in improving the accuracy of image classification.

5. Benchmark model

I searched several models for dog breed identification, and the model from this article [6] shows that the created model from scratch is with an accuracy of 12%, the Xception model that utilizes transfer learning can reach an accuracy of 86% and a loss of 0.4723 on test data.

This will be my benchmark model although I'm not sure whether the model I create will outperform it or not. My goal is not to outperform the benchmark model, but to study the possible methods that can improve the accuracy of dog breed identification. With this benchmark model I can have something fairly enough to compare and crosscheck.

6. Evaluation metrics

Since the problem stated here is a multi-classification problem and the data is slightly imbalanced, the evaluation metrics I used are accuracy evaluation metric and categorical_crossentropy cost function. This

multi-class log loss punishes the classifier if the prediction is different with the actual label and leads to a higher accuracy. The ideal classifier has zero loss and 100% accuracy.

7. Project design

The pipeline consists of three parts: human face detection, dog detection and dog breed inference. To complete this task, the project design is as following:

- **Step 0: Import Datasets.** Import all the images of dog and human, do some statistics and check some samples
- **Step 1: Detect Humans.** OpenCV face detector will be used here, the procedure is
  - Obtain face detector.
  - Load color (RGB) image.
  - Convert image to grayscale.
  - Use pre-trained face detector to detect human face.
  - Return True if a human face is detected and False otherwise.
- **Step 2: Detect Dogs.** A pre-trained VGG16 model will be applied here, the procedure is
  - Obtain VGG16 model.
  - Load and pre-process the image.
  - Send an image to the VGG16 model.
  - Return True if a dog is detected and False otherwise.
- **Step 3: Create a CNN to Classify Dog Breeds (from Scratch)**
  - Specify data loader.
  - Define model architecture.
  - Specify loss function and optimizer
  - Train and validate the model
  - Test the model
- **Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning)**
  - Specify data loader.
  - Define model architecture.
  - Specify loss function and optimizer
  - Train and validate the model
  - Test the model
  - Predict Dog Breed with the Model
- **Step 5: Write My Algorithm.** Write an algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither.
  - 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.
- **Step 6: Test My Algorithm.** Test the algorithm on my own images.

Reference

[1] Convolutional Neural Networks for Image Classification https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-networks-image-classification/.

[2] Kaggle Dog Breed Identification https://www.kaggle.com/c/dog-breed-identification/overview/description.

[3] Liu J, Kanazawa A. Dog Breed Classification Using Part Localization. European Conference on Computer Vision 2012; 172-185.

[4] Mulligan K, Rivas P. Dog Breed Identification with a Neural Network over Learned Representations from the Xception CNN Architecture. 21st International Conference on Artificial Intelligence 2019.

[5] Valueva M.V, Nagornov N.N, Lyakhov P.A, Valuev G.V, Chervyakov N.I. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. Mathematics and Computers in Simulation 2020; 177: 232–243.

[6] Deep Learning: How to build a dog detector and breed classifier using CNN?! https://towardsdatascience.com/deep-learning-build-a-dog-detector-and-breed-classifier-using-cnn-f6ea2e5d954a