Dog breed classifier

Capstone proposal

Pablo Serna

19 February 2020

Domain background

Image classification is one of the fields where computer vision and machine learning have achieved outstanding performance in the 21st century. The main task in this type of problem consists in classifying what is represented in a given picture. This problem can be faced in many ways, but those that made the breakthrough were based on supervised-learning algorithms, and more precisely deep convolutional networks. Particularly, the problem of dog breed classification in pictures belongs to the set of tasks called fine-grained category classification. This name refers to the fact that it assumes that a classifier has found before that the image contains a dog, and then we can classify the breed.

Since the development of AlexNet in 2012 [1], many algorithms have been competing for the quest of excellent image classifications. A classical algorithm, and still a reference nowadays, is called VGG, developed and trained by Oxford's Visual Geometry Group [2]. It consists in a total of *16 layers* of which 13 are convolutional layers (3x3), including activation functions (ReLUs) and 5 max-pooling layers (2x2) in-between, and the remaining 3 are fully-connected layers at the very end. Although very accurate, this neural network is very heavy too. The state-of-the-art for image classification is divided into two sets of algorithms, heavy algorithms with top-accuracy, typically based on ResNet-like architectures [3], and very light algorithms that are focus on efficiency, e.g. mobilenet [4]. However these algorithms aim to classify images in a broad number of categories (1000), and they are not specifically trained for fine-grained classification. To be able to use them in a specific subclass classification, they need to be re-trained or replaced by other, possibly similar, algorithms.

In general, since the problem we face here belongs to the fine-grained classification problems, we need to design a pipeline to classify images in broad classes and then in sub-classes. This is a standard practice in this sort of task and seems a reasonable way to approach it. For example, it is easier for an algorithm to identify landmarks of a face in a picture if another algorithm has found that face in the picture, cropped it and provided it as input.

Classifying dog breeds

Here, Udacity proposes a task that anyone that has had a dog has been asked before. What breed is the dog? In general, this task is not trivial at all, even for humans that can inspect the dog in real life. Can a neural network learn to classify them from their pictures?

To state it in a more precise way the problem to solve is the following:

The algorithm has to classify a picture provided by the user. It has to state whether it is a dog, a human or something else. In the case of a dog in the picture, it has to classify the breed of the dog. In the case of a human, it has to provide which breed of dog the human resembles.

This algorithm can be deployed as a web application, where the user would have the option of uploading a picture and having the answer back. If possible, the web app could also ask for taking the picture directly.

A way to assess the quality of the algorithm would be by estimating statistics of predictions, mainly accuracy, or F_1 score. Note that for some pictures the quality of correct prediction may become ambiguous, for example in pictures with both a human and a dog, or with several dogs of different breeds.

Datasets

For training and testing the neural network, we will be using a dataset of 8351 pictures of dogs, classified in 133 breed. The dataset is split in training set (80%), validation set (10%) and test set (10%). The classes are not completely well balanced. In the training set there are on average 50 images per breed. The class with the least number of images is the Xoloitzcuintli, with 26 images. The breed with the most number of images is Alaskan Malamute, with 77. This dataset can be found in an Udacity's bucket in AWS.

To recognize and distinguish dogs from humans, we will be using the dataset: Labeled Faces in the Wild (LFW) (webpage) [5, 6]. This dataset contains 13233 images of humans labeled by the name of the person.

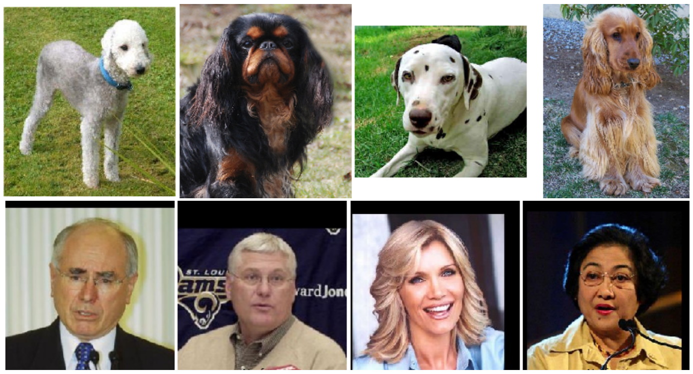


Figure 1: Upper row. Examples of pictures of dogs from the dog dataset. Lower row. Examples of pictures of humans in the human faces dataset.

Solution statement

For this task, we require a two-steps solution. A first step is to recognize what is in the picture: dog, person or something else. A second step, in the case of a dog or a human, is to answer with the most likely dog breed for the individual in the picture.

- For the first step, we can use a standard classifier such as VGG16 [2] or mobilenetv2 [4]. Pre-trained versions of these two networks can be downloaded easily in PyTorch. If the answer is a category different from humans or dogs, then the algorithm will raise an error, otherwise, it will go to the next step.
- For the second step, we could either train a convolutional neural network from scratch or use transfer learning. The first option is not very practical unless we have access to a GPU with no constraint in time. The second option can be exploited to accelerate training since using pre-trained weights for another task will simplify the path of the gradient descent algorithm. On top of that, the dataset is not as large as ImageNet can be, and we would risk encountering other problems if we do not use transfer learning, such as overfitting.

Benchmark

A well-known work published in 2012 [7] attempted precisely this task using *part localization*, i.e., detecting the face of the dog and locating facial landmarks in the picture. This method was in vogue a few years back before the advent of the very deep neural networks. They obtained 67% accuracy in this dataset, with 8351 pictures of dogs classified in 133 breeds. The dataset was created by downloading images from sources such a Flickr, Image-Net, and Google. The similarity with our own dataset is probably not a coincidence.

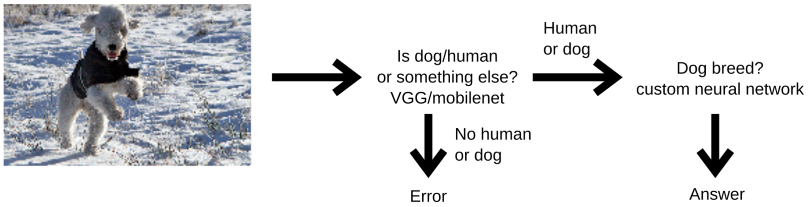
There does not seem to be many other scientific publications related to dog breed classification. The internet, however, is full of projects solving this issue, mostly from previous students of this series of lessons. Since the previously mentioned work is a well-cited paper, we will consider it as our reference. In the scientific literature, however, there are similar works with other species, for example for birds [8].

Evaluation metrics

Given the benchmark cited before, the first quantity we will need to estimate will be the accuracy. However, we may want to monitor other interesting statistical quantities, particularly F_1 score, the harmonic mean of precision and recall. Also, for the first step in the pipeline solution, we will probably be interested in minimizing the false negative answers, since this can be particularly annoying for the experience of the user.

Project design

The overall design has already been presented with the two-steps pipeline solution. We will develop it in a jupyter notebook, using the GPU provided by Udacity and a local one. Once the pipeline is well defined and the classifier trained, we will deploy the model in a simple web app, where we can upload a picture and get it classified.



References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[5] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. 2008.

[6] Gary B Huang and Erik Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. *Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep*, pages 14–003, 2014.

[7] Jiongxin Liu, Angjoo Kanazawa, David Jacobs, and Peter Belhumeur. Dog breed classification using part localization. In *European conference on computer vision*, pages 172–185. Springer, 2012.

[8] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L Alexander, David W Jacobs, and Peter N Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2018, 2014.