Pooch Detector: Image Classification with Convolutional Neural Networks

A report by Carlos Gavidia-Calderon

Background

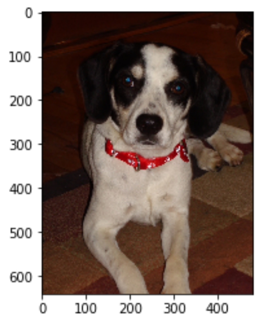
In this project, we will apply a supervised learning approach to an image classification problem. That is, given a training set of labelled images, our machine learning system will learn to assign correct labels to previously unseen images. Among the multiple applications of image classification, we can mention its use for automatic diagnosing of skin cancer, outperforming professionally dermatologists[1].

Problem Statement

We will build an image classifier for dog breeds. When an image is fed to the system, it will output the most likely breed among 133 potential options. These classes are obtained from our labelled training dataset.

Dataset

The dataset is available at the Udacity GitHub repository[2] corresponding to the project proposal. It contains 8351 labelled RBG dog images. Here's an example:



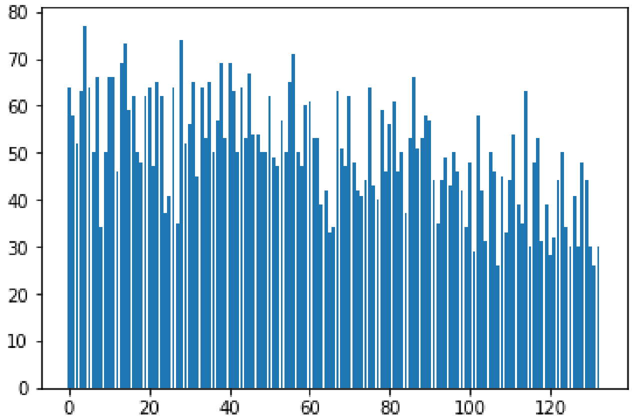
Images in our dataset vary largely in size, so will use a resizing transformation before feeding them to the CNN models (more details on the "Project Design" section).

Here is a histogram of class frequency over the training dataset, where we observe that labels are evenly distributed.

---

[1] Esteva, Andre, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. "Dermatologist-level classification of skin cancer with deep neural networks." Nature 542, no. 7639 (2017): 115-118.

[2] Dataset available at https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip



Proposed Solution

Our solution will be a convolutional neural network (CNN), an approach proven effective for image classification tasks. To increase performance, we will adopt a transfer learning approach, relying on a pre-trained CNN for feature extraction.

Benchmark Model

For comparison purposes, we will also develop a convolutional neural network from scratch, instead of implementing feature extraction with transfer learning. This model will serve as reference, providing a lower-bound on the performance our transfer learning approach can achieve.

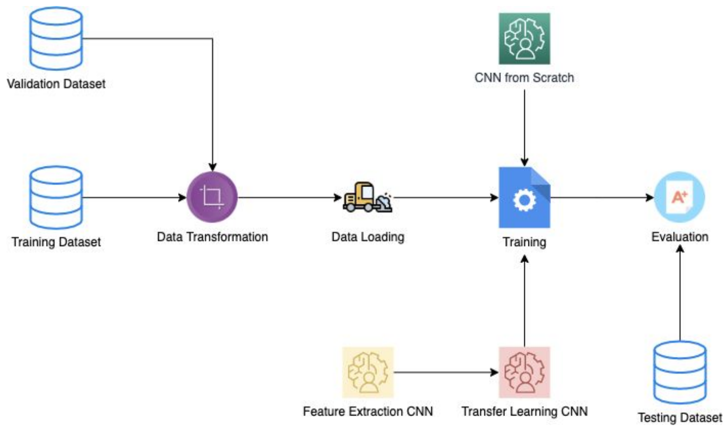
Evaluation Metrics

During training, we will attempt to minimize **cross-entropy loss**, as is common in multiclass classification.

As seen in previous sections, <u>classes in our dataset are balanced</u>. Hence, we can rely on **accuracy** over test dataset to evaluate system performance. Accuracy is defined as the percentage of unseen images correctly classified after training.

Project Design

Our approach is illustrated in the following diagram:



Data Transformation and Data Loading:

The original dataset will be split in three subsets: training, validation and testing. Only training and validation will be used for training purposes.

The raw images from the dataset will need to be pre-processed to be suitable inputs for our CNNs. Pre-processing steps include resizing, cropping, and image normalization. These steps need to satisfy the input requirements of the Feature Extraction CNN.

Regarding data augmentation, we will extend the training dataset by flipping its images.

Convolutional Neural Networks

We will train and evaluate two CNNs: One made from scratch as a reference model and one using a transfer learning approach. We would expect the latter to perform significantly better.

The CNN “from scratch” will be composed of two or more of the following convolutional blocks: a 2D-convolutional layer with a ReLU activation, followed by a max-pooling layer. These convolutional blocks will be followed by a one or more linear layers, where the last linear layer outputs a score per class label.

Regarding the feature extraction CNN for our transfer learning approach, we are considering adopting the VGG-16 model[3], given its effectiveness over the ImageNet dataset. We will use its pre-trained feature extraction layers and append one or more linear layers followed by ReLU activations. The last linear layer will output a score per class label.

Evaluation

The evaluation step calculates cross entropy-loss and accuracy over the testing dataset. Our goal is to obtain at least 66% accuracy. In case initial attempts fall short, we will consider the following solutions: 1) extending training time, by increasing the number of epochs 2) increasing network complexity by adding additional layers and 3) tuning network hyper-parameters like learning rate.

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