Capstone Project Proposal

Dog Breed Classifier – Image processing with Pytorch

Jose Vaides
Machine Learning Engineer Nanodegree - Udacity

- *Udacity Machine Learning nanodegree, Leonberger_06571.jpg*

Project Proposal

Domain Background

Thanks to the sense of sight, human beings are able perceive the world around us in a much wider range than any of our other senses (perhaps the sense of sound is comparable in terms of range). Thanks to our "human" vision, we can identify objects that are at tens if not hundreds of meters of distance, and as the saying "a picture is worth a thousand words" indicates, the amount of information that we can obtain through vision has the potential to exceed that of any other sense.

Although for most human beings, the sense of sight is available from the time we are born, for computers, the ability to process images and obtain information from them is not a simple task. For us, our brains have evolved for millions of years, creating the rules and connections between our neurons that allow us to convert photons of light entering our eyes into valuable insights and information about the environment around us. However, computers didn't have the millions of years we had to develop these rules and algorithms that would allow them to process visual information in the same way human beings do. If we could create these algorithms to allow computers to "understand" and process visual information, there is a vast range of applications where they could be used.

In the late 1950s, neurophysiologists were trying to understand studying how neurons in the visual cortex of cats responded when different images were shown to them. At about the same time, the first digital image scanner was invented by Russel Kirsch. Later, scientists started to study how to derive 3D information from 2D photographs, and with the birth of AI as a discipline in the 1960s researchers believed it would take less than 25 years to create a computer as intelligent as a human being. The prediction was unfortunately not correct, and to the current date the problem of teaching a computer how to process image information in the same way as humans has not been solved yet.

Because of this potential that has not been yet fully exploited, it is that the field of **Computer Vision** was born.

This project will be focused on the domain of **Computer Vision**:
"Computer vision is a field of artificial intelligence that trains computers to interpret and understand the visual world. Using digital images from cameras and videos and deep learning models, machines can accurately identify and classify objects — and then react to what they 'see.'"

- Computer Vision and why it matters, SAS

There have been different attempts performed to tackle the problem of computer vision. Some of these include the work of David Marr in 1982, who developed algorithms for detecting edges, curves and corners but didn't propose any learning process, and Kunihiko Fukushima's work on an artificial neural network with convolutional layers that could recognize patterns without being affected by shifts in position. Later, in 1989 Yann LeCun used a backpropagation algorithm on Fukushima's neural network, and developed LeNet-5, which started to resemble the convolutional neural network architecture used in current times.

Following these attempts, around 1999 the efforts of researchers started to shift from attempting to reconstruct 3D models from 2D images towards feature-based object recognition from images. In 2001 Paul Viola and Michale Jones introduced the first face detection framework working in real time, which implemented learning features to localize faces.

Forward into 2010, the ImageNet Large Scale Visual Recognition Competition (ILSVRC) was started (with yearly repetitions), and became a benchmark with its dataset of more than a million images distributed across 1000 different categories. In this competition, AlexNet, a CNN model, was presented by a team from the University of Toronto, which achieved a breakthrough improvement on the previous error rate of around 26%, to a new record of 16.4% on the ImageNet dataset. Ever since 2012 the winners of the competition have always been Convolutional Neural Networks.

- A Brief History of Computer Vision (and Convolutional Neural Networks), Demush, R.

As described above, in recent years, the development of **Machine Learning** algorithms based on **Artificial Neural Networks** and **Deep Learning** have shown great promise as enablers for Computer Vision applications, among others. Thanks to these new technologies applications that were previously not possible with other algorithms have now become possible, and the performance of other applications has also been greatly improved, enabling them to be used in real world scenarios.

Problem statement

The task of image classification is one of the many use cases where machine learning has been widely used in recent times. Generating a set of rules that can adapt to pictures of different dog breeds or the same breed at different angles, color scales and positions, would be extremely complex. Approaching this problem with a hard coded or rules based system would be very difficult, so it's a great example of where an algorithm that learns from examples can be used.

In this project, the problem being addressed is that of finding out what is the breed of a dog, by providing an application with a picture of the dog. Although the use case might seem trivial, the underlying technology could be used for more significant use cases, such as processing medical images to detect diseases (i.e. classifying images into different disease categories).

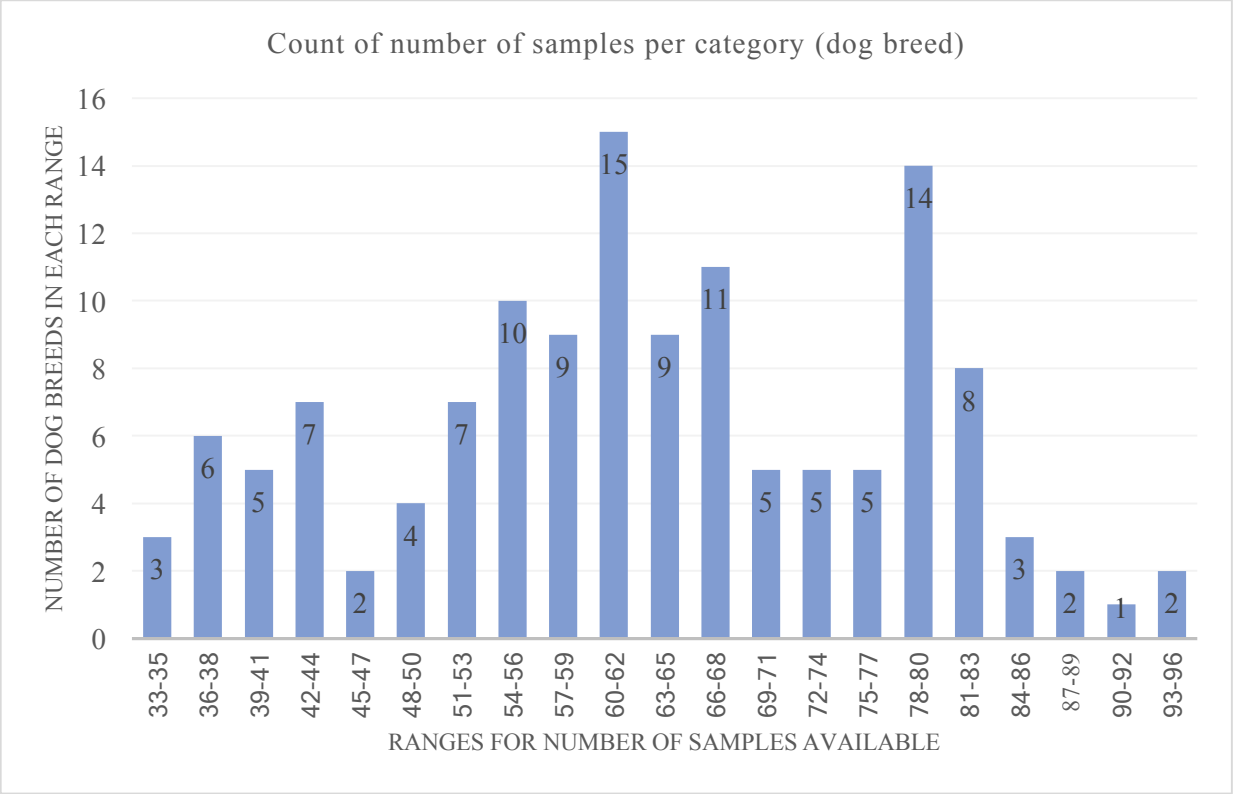
Datasets and inputs

The dataset to be used in the development of the Machine Learning Model will be the set of images provided by Udacity as part of the Machine Learning Engineer nanodegree. This dataset contains more than 8000 images, which are organized into a folder structure, where all the images for each dog breed are included into individual folders. The images are of different sizes, but will be transformed and normalized when being used as inputs for the model.

The distribution for the training, validation and test datasets is the following:

Dataset	Number of samples	% from total
train	6680	80%
test	836	10%
valid	835	10%
Total	8351	

The full dataset contains 133 different breeds of dogs, all 3 sub-datasets (train, valid and test) contain samples for each of the breeds. The total distribution of the samples available by dog breed is the following:



The images in the dataset are of different sizes, therefore they will be cropped to a size of 224x224 pixels, and normalized using torchvision.transforms. The transforms used for the training dataset will include randomly rotating and randomly flipping the images to augment the dataset, by providing positional variations of the pictures to the model.

Solution Statement

As mentioned in previous sections, a Machine Learning algorithm will be used to train a model that will learn how to classify images of dog breeds. More concretely, a model using a convolutional neural network will be trained with the sample data to build the classifier.

The approach for building the model will be to first build a completely new model (from scratch), which will be trained using the provided set of dog breed pictures. The accuracy and other benchmark metrics for this model will be measured, and then a second model will be built using transfer learning. The pre-trained model will be the VGG16 model included in the torchvision library, and the classifier that will be attached to the pre-trained model will also be built using Pytorch.

After the development and training of the models is complete, the model with the best results, based on the evaluation metrics, will be deployed in Amazon Sagemaker, together with a simple website to upload pictures and generate predictions.

The web application will use a file upload button, where the user can choose a picture to be classified. This picture will be uploaded to Amazon S3, where the deployed sagemaker application will transform the picture to a Tensor, and input it to the neural network model. The predicted response will be sent back to the web application as a Json Object containing the name of the breed predicted, and a link to a sample picture of this breed, which will then be displayed by the web application.

Additionally, as part of the project, the classifier will also allow pictures of humans to be used as inputs, which will then be matched by the application to one of the dog breeds that were included in the dataset.

Benchmark Model

The Benchmark Model to be used will be Paul Stancliffe's Dog Classifier (Described in https://medium.com/@paul.stancliffe/udacity-dog-breed-classifier-project-walkthrough-e03c1baf5501) which was also trained with the dataset provided in the same Udacity project, and also classifies the same number of categories (dog breeds).

The benchmark metric detailed in the article from Stancliffe is accuracy and therefore this metric will be used for comparison with his model. Stancliffe's model reached an accuracy of 84.8% on the test dataset.

The VGG16 model by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition", which achieved a 92.7% accuracy in ImageNet, a dataset of more than 14 million images from 1000 classes, will also be used as a reference, although it cannot be used directly as a benchmark (i.e. saying that one model is better than the other), since the number classes and datasets used are different. However, the results of this model can still be used as a benchmark reference to rate the general performance of an image classification model on different datasets.

- *VGG16 – Convolutional Network for Classification and Detection, Neurohive.io*

Although the VGG16 model was trained with a much larger dataset than the one used for the dog classifier, the number of dog breeds used for this project is smaller than the 1000 classes from the VGG16 model. The same level of accuracy might not be reached with a smaller data set, but it still is a good reference to be used for this project.

Set of evaluation metrics

The benchmark metric to be used for comparison will be accuracy, since the benchmark model selected has provided this metric as evaluation of their results.

The metric **F1 Score**, defined as:

$$\left(2 * \frac{precission * recall}{precission + recall}\right)$$

- *Scikit-learn.org*

will also be calculated (although not used as a benchmark) to evaluate the performance of the generated model, since the system will be classifying multiple classes which might have a different number of test images per class, and we are mainly interested in finding out how many of the images are classified correctly.

Other metrics such as recall, and precision will also be calculated but will not be the focus when evaluating the model.

Project Design

The project will be developed using the Jupyter Notebook provided by Udacity as a part of the Machine Learning Engineer nanodegree. The initial Analysis and development will be done using the notebook. The notebook will be run locally, connecting to Sagemaker's API to execute the training scripts and eventually deploy the trained model for testing and for the final live application to be submitted for this project.

Using the Jupyter notebook, the steps to develop the project will be the following:

1. **Data Collection:** This step is covered by the data provided as part of the nanodegree. The data will be uploaded to Amazon S3 so that it can be used as input for the training script.
2. **Data Preparation:** The data will be loaded and transformed using torchvision's transforms.transform and datasets.Imagefolder libraries. For the training dataset, random rotation and flipping of the images will be performed to enhance the dataset with different variations of the images. Additionally, as well as for the test and validation datasets, the images will be cropped, normalized and transformed to tensors to be used by the Model.
3. **Choosing a Model:** The model to be used will be a convolutional neural network built using Pytorch. Initially a model generated from scratch will be used, and later a pre-trained model will also be used. Different architectures will be tested and the one with the best performance results will be selected.
4. **Training the Model:** The model will be trained using the images in the training dataset, and using Amazon Sagemaker's API, in order to use GPUs and speed the training process.
5. **Evaluating the Model:** After the model has been trained, it will be deployed on a Sagemaker endpoint so that predictions can be performed. The test dataset will be stored locally, but the prediction will be used Sagemaker's API and the deployed endpoint. Using the evaluation metrics mentioned previously. A confusion matrix will be generated from which the results will be calculated.
6. **Parameter Tuning:** The parameters will be adjusted in order to improve the evaluation results. The parameters will be adjusted manually through multiple iterations.
7. **Deployment of Live Application:** A simple html file will be made available on the web where it will be possible to upload a picture and obtain a prediction of the dog breed in the uploaded file.

Once the steps in the notebook and have been completed, the project will be migrated into a Github project, where the steps necessary to convert it into a deployed app in Amazon Sagemaker will also be described.

References

1. SAS, Computer Vision and why it matters, URL: https://www.sas.com/en_us/insights/analytics/computer-vision.html#:~:text=Computer%20vision%20is%20a%20field,to%20what%20they%20%E2%80%9Csee.%E2%80%9D
2. Neurohive.io, VGG16 – Convolutional Network for Classification and Detection, URL: https://neurohive.io/en/popular-networks/vgg16/
3. Simonyan, K., Zisserman, A., Very Deep Convolutional Networks for Large-Scale Image Recognition, URL: https://arxiv.org/abs/1409.1556
4. scikit-learn developers (BSD License), sklearn.metrics.f1_score, URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html
5. Udacity, Machine Learning Engineer nanodegree – Dog Breed Classifier, URL: https://classroom.udacity.com/nanodegrees/nd009t/parts/2f120d8a-e90a-4bc0-9f4e-43c71c504879/modules/2c37ba18-d9dc-4a94-abb9-066216ccace1/lessons/4f0118c0-20fc-482a-81d6-b27507355985/concepts/65160313-7054-4ffb-8263-793e2a166d69
6. Marr, D. (2010). Vision: A computational investigation into the human representation and processing of visual information. Cambridge, MA: MIT Press.
7. Demush, R. (2019), A Brief History of Computer Vision (and Convolutional Neural Networks), URL: https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aacc79f3
8. Stancliffe, P. (2019) Udacity Dog Breed Classifier — Project Walkthrough, URL: https://medium.com/@paul.stancliffe/udacity-dog-breed-classifier-project-walkthrough-e03c1baf5501, github: https://github.com/paulstancliffe/Dog-Breed-Classifier