Project Writeup

A. Project Definition

1. Project Overview

Student provides a high-level overview of the project. Background information such as the problem domain, the project origin, and related data sets or input data is provided.

The goal of the project is using Convolutional Neural Networks (CNNs) to classify dog pictures by the breed !

In this project,we build a pipeline to process real-world, user-supplied images.

Given an image of a dog, the algorithm will identify an estimate of the canine's breed. If supplied with an image of a human, the code will identify the similar dog breed.

To complete this approach, we create a web application in order to play with the created model.

You can find all the steps related to the creation of the different functions and models necessary to our task in the associated notebook dog_app.ipynb.

The main steps are :

- Import the Dataset
- Create a Human Detector Humans
- Create a dog Detector Humans
- Create a CNN to classify Breeds, from scratch --> 2% of accuracy
- Use a CNN to Classify Dog Breeds, with Transfer Learning --> 65 % of accuracy
- Create a CNN to Classify Dog Breeds, , with Transfer Learning with specialized model --> 80% of accuracy
- Sum-up the all in a function

2. Problem Statement

The problem which needs to be solved is clearly defined. A strategy for solving the problem, including discussion of the expected solution, has been made.

- The project's goal is to predict dog breeds (133 types) by creating a model trained using labeled dog images.
- On the first attempt to build a CNN from scratch, the results were poor (around 2% accuracy). This is an expected result because image recognition requires more complex features to detect patterns.
- By using a transfer learning approach, we have improved the model performance. Transfer learning is a machine learning technique that enables data scientists to benefit from the knowledge gained from a previously used machine learning model for a similar task (research.aimultiple.com - see ressource).

3. Metrics

Metrics used to measure the performance of a model or result are clearly defined. Metrics are justified based on the characteristics of the problem.

Accuracy is our main metric. This is the percentage of correct prediction made by the model compared to the test set.

Accuracy main avantages:

- Easier to understand by all, technical and non technical people.
- Classic performance metric for these models detector
- We are on a premise of the technique. We don’t stand for 0,1% of improvement but our range is around 1 to 90%, and this is useful to see improvement.

Other performance metrics that could have been used are the confusion matrix & F-score.

Analysis

4. Data Exploration

Features and calculated statistics relevant to the problem have been reported and discussed related to the dataset, and a thorough description of the input space or input data has been made. Abnormalities or characteristics about the data or input that need to be addressed have been identified.

Datasets

Our Datasets have been provided by Udacity :

- There are 133 total dog categories.
- There are 8351 total dog images.
- There are 6680 training dog images.
- There are 835 validation dog images.
- There are 836 test dog images.

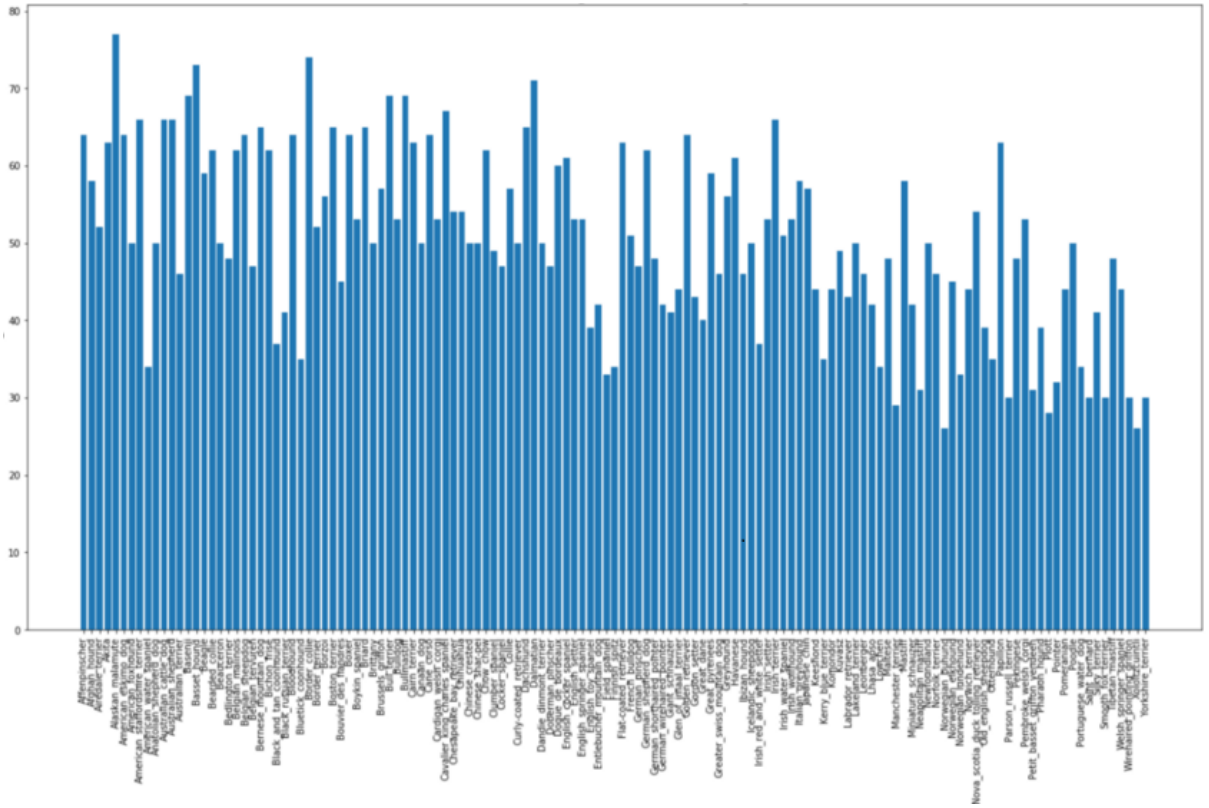
We split our data into two groups by random selection : train and a test set.

There is also a dataset with more than 13000 human images We used in our preprocessing data human detector.

5. Data Visualization

Build data visualizations to further convey the information associated with your data exploration journey. Ensure that visualizations are appropriate for the data values you are plotting.

Number of pictures of dogs in the dataset by breed



Methodology

6. Data Preprocessing

All preprocessing steps have been clearly documented. Abnormalities or characteristics about the data or input that needed to be addressed have been corrected. If no data preprocessing is necessary, it has been clearly justified.

To be able to classify dogs, we need to be able to detect them in pictures.

Therefore, we need to create independent functions :

Face detector

We created this using one of the pre-trained face detectors provided by opencv.
Which will be used to detect humans in the image. one of the required specification og the challenge is to classify humans as a closer dog breed also (without performance issue, it is more for fun). return a boolean.

Dog detector

We created another function ; to detect if a dog is in the picture.using pre-trained ResNet50 which simply returns true/false.

Picture resizing

We also create a function (for maintenance and reliability) which resizes the image to 224x224 pixels. S
Because Keras CNNs require a 4D array as input, with the shape (nb_samples,rows,columns,channels) and as input to some color images, the corresponding tensor would be 3D . So each image would be 224x224x3. And for n samples the tensor needs to be 4D as nx224x224x3.

7. Implementation

The process for which metrics, algorithms, and techniques were implemented with the given datasets or input data has been thoroughly documented. Complications that occurred during the coding process are discussed.

The different steps of the project were built with the help of Udacity comment and progressive learning in order to fulfill the final goal of predicting the dog's breed.

Model Architecture

The CNN algorithm is built using the Keras library.

- We start by initializing the model as Sequential model.
- We choose a type of
- We define an input_shape with the right format for the selected model.
- We define the associated Dense parameter (number of node) with the number of breeds in our dataset = 133.
- We added a dropout to limit the overfitting and it was an improvement for the value 0,05. 0,1 and 0,2 were not.

8. Refinement

The process of improving upon the algorithms and techniques used is clearly documented. Both the initial and final solutions are reported, along with intermediate solutions, if necessary.

Optimiser

We tested to change the optimizer like the 'adam' one following the Keras documentation but 'rmsprop' was better.

Transfer learning using Pre-trained CNN

Wefirst tried to create a model from scratch but it wasn't very good : 2% accuracy. even with some improvement.

Then we used a the VGG16 model.

We tried to download InceptionV3 but it was too big in terms of MB. Therefore we went for the Resnet_50 and it was more than ok. Our model accuracy reached 80% of good answers.

Our model was ready.

Our final model architecture with Resnet_50.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
global_average_pooling2d_3 ( (None, 2048)              0
_________________________________________________________________
dense_3 (Dense)              (None, 133)               272517
=================================================================
Total params: 272,517
Trainable params: 272,517
Non-trainable params: 0
_________________________________________________________________
```

Results

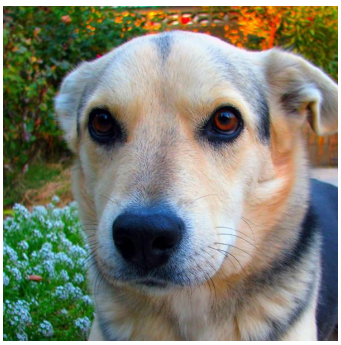
On our three components :

- face_detector was tested. 100% of the human faces have been found.
- dog_detector was tested. 0% of the human faces have been found. 100% of the dogs have been found.
- The Resnet_50 model has an accuracy of 80% to classify the dog picture by its breed.

9. Model Evaluation and Validation

If a model is used, the following should hold: The final model's qualities — such as parameters — are evaluated in detail.
Some type of analysis is used to validate the robustness of the model's solution. For example, you can use cross-validation to find the best parameters.

Show and compare the results using different models, parameters, or techniques in tabular forms or charts.
Alternatively, a student may choose to answer questions with data visualizations or other means that don't involve machine learning if a different approach best helps them address their question(s) of interest.



'This is dog. His breed must be : Labrador_retriever' ***Correct***

'This is a human. His closer dog breed is : Great_dane' ***Correct a human.***

'This is not dog or not human. It must be a mistake, only dogs and humans exist... The closer dog breed : Norwegian_lundehund'

This is correct for a European wolf.

Conclusion

The initial goal is reach : We were able to classify dog breed by using a CNN with a accuarency of more than 80%

The developed function to create this CNN and a web application are also running well.

10. Reflection

Student adequately summarizes the end-to-end problem solution and discusses one or two particular aspects of the project they found interesting or difficult.

The most interesting learning for myself was the use of transfer learning. This is impressive how we can easily step up a model with one of these. They already have some baseline to analyze the data and with a small number of epochs they achieve good results.

I also like the simplicity of keras to be personnalise a model from scratch with a easy paratrable environnement.
Maybe I will use one of these in my job.

11. Improvement

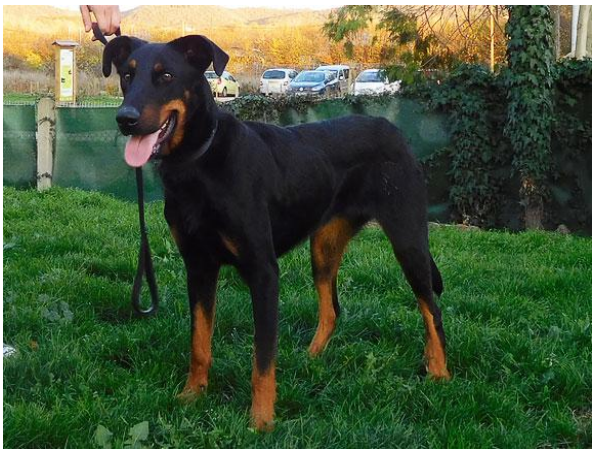
Discussion is made as to how at least one aspect of the implementation could be improved. Potential solutions resulting from these improvements are considered and compared/contrasted to the current solution.

On input data :

- **Add a metadata with the pictures loaded : the sex of the dog.**

Because the sex change sometime the appereace of a dog, he may confuse the model sometime.

As an example for the Beauceron breed, males are 10-15 kg bigger than females and have their ears cut because they are much taller.

- **Improved the number of data input for the close breed.**

None of the models (without different parameters and epoch) have been able to tell me that my beauceron is … not a doberman. And I can't blame it as human-being. It needs special attention for those cases.


On the model:

- **Use InceptionV3.**

It seems to be the greatest pre-trained model for this type of classification. As read online.

Ressources

- https://medium.com/@gopal.iyer0/robot-motion-planning-dsnd-capstone-project-234252e608b9
- https://knowledge.udacity.com/
- https://keras.io/api/
- https://www.simplilearn.com/tutorials/deep-learning-tutorial/guide-to-building-powerful-keras-image-classification-models
- https://research.aimultiple.com/transfer-learning/
- https://keras.io/guides/sequential_model/