UDACITY

Back to Deep Learning Nanodegree

Dog Breed Classifier

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Greetings Student,
That's a solid submission. A lot of good things has been observed in your work. Well organized answers, simple and optimal implementation. Congratulations! All the rubrics are okay. Nice work! I invite you to visit all the suggestions and tips provided in this feedback. Thank you! Keep this professional spirit.

Files Submitted

The submission includes all required files.

All necessary files are present in this submission. Great!

Step 1: Detect Humans

The submission returns the percentage of the first 100 images in the dog and human face datasets with a detected human face.

The above values are acceptable. Good job!

```
human 99/100
dog 99/100
Human face detection in human_files_short:  1.0
Human face detection in dog_files_short:  0.11
```

```
Human face detection in dog_files_short:  0.11
```

The submission opines whether Haar cascades for face detection are an appropriate technique for human detection.

Your opinion is good and reasonable. 👍

> Answer: As we might be using features which have been learnt from some pre-trained face detectors, it is reasonable to ask the user to provide a clear view of a face. Otherwise our cnn will try to identify face-like features which do not appear on the picture.
>
> Should we want our app to work on pictures which do not clearly show the face of the person, we would need to use features detectors trained on non face-focused images.
>
> Alternatively, we could use the current face-detector up to the very last layers which are too face-detection specific. Then we could retrain the model on a non face-focused data set of images. This would be a good example of learning transfer.

Pro Tips

Here are some documents that provide information on Haar cascades.

- Tutorial face detection;
- https://www.youtube.com/watch?v=88HdqNDQsEk;
- Opencv face detection;
- OBJECT DETECTION : FACE DETECTION USING HAAR CASCADE CLASSFIERS;
- Youtube video;

Step 2: Detect Dogs

The submission returns the percentage of the first 100 images in the dog and human face datasets with a detected dog.

```
human processed 99/100
dog processed  99/100
Dog detection in human_files_short:  0.0
Dog face detection in dog_files_short:  1.0
```

You have achieved superb results. Your algorithm is optimal and does the job expected. Keep it up!

Step 3: Create a CNN to Classify Dog Breeds (from Scratch)

The submission specifies a CNN architecture.

The submission perfectly specifies the architecture used.

Suggestions and Comments

- I suggest you read this document about Using Convolutional Neural Networks to Classify Dog Breeds. This is a good outside resource to complete your reading of this course.

The submission specifies the number of epochs used to train the algorithm.

We note the number of epochs (`epochs = 15`) in this submission.

Pro Tips

Here are several documents that talk about the choice of the number of epochs.

- How does one choose optimal number of epochs?
- How to train your Deep Neural Network;
- Number of epochs to train on.

The trained model attains at least 1% accuracy on the test set.

A good result of your model on the test dataset of dog images.

```
Test accuracy: 5.5024%.
```

Suggestions and Comments

I would like to share this article with you wherein the discussion is improving the accuracy using CNN. This may help you out in the future.

Step 5: Create a CNN to Classify Dog Breeds

The submission downloads the bottleneck features corresponding to one of the Keras pre-trained models (VGG-19, ResNet-50, Inception, or Xception).

The submission correctly downloads the bottleneck features. Nice work!

The submission specifies a model architecture.

Good work! The submission specifies a model architecture.

Pro Tips

Please, I suggest you take a look at this very good document:

- Dog breed image classification with Keras.

The submission details why the chosen architecture succeeded in the classification task and why earlier attempts were not as successful.

Excellent work! We note that, this submission details why the chosen architecture succeeded in the classification task and why earlier attempts were not as successful.

The submission compiles the architecture by specifying the loss function and optimizer.

The loss function and optimizer are specified in the compilation of this architecture. That's great!

The submission uses model checkpointing to train the model and saves the model weights with the best validation loss.

The submission loads the model weights that attained the least validation loss.

Good job loading the model weights that attained the least validation loss!

Accuracy on the test set is 60% or greater.

Superb work! Accuracy on the test set is 80.9809%.

The submission includes a function that takes a file path to an image as input and returns the dog breed that is predicted by the CNN.

The submission contains a function that takes a file path to an image as input and returns the dog breed that is predicted by the CNN. Fabulous work!

Step 6: Write Your Algorithm

The submission uses the CNN from Step 5 to detect dog breed. The submission has different output for each detected image type (dog, human, other) and provides either predicted actual (or resembling) dog breed.

Wow! The submission uses the CNN from Step 5 to detect dog breed. Furthermore, the submission has different output for each detected image type (dog, human, other), and provides either predicted actual (or resembling) dog breed.

Step 7: Test Your Algorithm

The submission tests at least 6 images, including at least two human and two dog images.

The submission tests images. 

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review