UDACITY

PROJECT

Dog Breed Classifier

A part of the Artificial Intelligence Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

1 SPECIFICATION REQUIRES CHANGES

Remark

Dear Excellent Student,
Thank you for your submission. I enjoyed going through your work because it was exceptional and impressive. You meet most of our expectations. I have provided feedback below to work you out your errors to meet our specifications. A lot of work was done during this project and i observed you tried out the different models. Implement the suggestions you mentioned and see how far your algorithm can be improved. Keep practising on this and other projects of your own and you will be the best at what you do. Remember, practice makes everything. Check out this blogpost http://machinelearningmastery.com/object-recognition-convolutional-neural-networks-keras-deep-learning-library/ on object recognition CNNs with Keras. You could learn a thing or two from there. Keep up the good work and we are waiting for your resubmission with the changes.

Files Submitted

The submission includes all required files.

All required files were included in the submission. Nice!

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.

Good. The percentage of the first 100 images in human_filenames and in dog_filenames were detected and correctly returned. Ideally, we would like to have 100% of human images with a detected face and 0% of dog images with a detected face but the algorithm falls short as shown in the answers with 99% of human and 11% of dog images.

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

Question 2 is not answered. Please we need to answer all questions to pass this submission

Required

> **Question 2**: This algorithmic choice necessitates that we communicate to the user that we accept human images only when they provide a clear view of a face (otherwise, we risk having unnecessarily frustrated users!). In your opinion, is this a reasonable expectation to pose on the user? If not, can you think of a way to detect humans in images that does not necessitate an image with a clearly presented face?

An opinion on whether Haar cascades for face detection are an appropriate technique for human detection has to be given.

Extra Tips

Some research on Haar Cascades Technique for face detection can be found in this document https://pythonprogramming.net/haar-cascade-face-eye-detection-python-opencv-tutorial/

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.

Awesome! The percentage of the first 100 images in human_filenames and in dog_filenames with a detected dog were detected and correctly returned. 1% of human_filenames detected a dog and 100% of dog_filenames detected a dog. From observations, we notice that the results get better as the algorithm is modified.

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

The submission specifies a CNN architecture.

The CNN architecture was specified and well implemented in the submission.

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

Reasonable number of epochs used. Nice.

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

Good job. Test accuracy is higher than 1%.

```
Test accuracy: 3.3493
```

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).

Using Keras pre-trained model, ResNet-50, the submission downloads the bottleneck features for training, validating and testing.

The submission specifies a model architecture.

The pre-trained ResNet-50 model used as a fixed feature extractor and the bottleneck features are fed as input to the model. Also, a global average pooling layer and a fully connected layer were added to the model with the fully connected layer having one node for each dog category and equipped with a softmax. This was used to specify the model architecture as required. Awesome.

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

The reasons why the chosen architecture was successful in the classification task was clearly stated. Nice.

Suggestions

Even though the first architecture chosen worked out as expected, it would be good to try out or make other attempts with different architectures and find out the reason they aren't working the way they are supposed to be or why they work better. Try doing this next time. I believe it will enhance your knowledge and skills in making better decisions and architectures in the future.

Extra Tips

Check out this paper on Speed/accuracy trade-offs for modern convolutional object detectors and the blog post ImageNet: VGGNet, ResNet, Inception, and Xception with Keras which will give you pointers on why ResNet-50 is more accurate than the other architectures.

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

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

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

Model checkpointing was used to train the model and save the model that attains the best validation loss.

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

The model weights with the least validation loss were loaded as required. Nice.

Accuracy on the test set is 60% or greater.

A test accuracy far greater than the set accuracy of 60% was attained. Excellent

```
Test accuracy: 80.6220%
```

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.

A function that takes a file path to an image as input and returns the dog breed that was predicted by the CNN has been included and well implemented in the submission.

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.

The algorithm implemented clearly uses the CNN to detect dog breeds with different outputs for every detected image be it a dog, human or other images and this provides actual predictions of a specific dog breed or a likely dog breed. Awesome.

Step 7: Test Your Algorithm

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

Tests were made with more than 6 images and at least 2 dogs and 2 humans used with a detailed explanation of what was expected as output and what was gotten, together with suggestions on how to improve the algorithm. Excellent!

RESUBMIT

DOWNLOAD PROJECT

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

RETURN TO PATH

Student FAQ Reviewer Agreement