UDACITY

‹ Back to Deep Learning

Dog Breed Classifier

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Congratulations on meeting all the specifications for this project! I hope you enjoyed working on this project and I wish you all the best for the rest of the course.

More useful resources

- CNN Tricks
- Building An Image Classifier
- Nvidia Unveils A Terrifying AI Supercomputer
- Understanding Activation Functions in Neural Networks
- Transfer Learning using Keras
- Let Artificial Intelligence Evolve
- Deep Face Recognition

Files Submitted

The submission includes all required files.

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

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.

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

The submission specifies a CNN architecture.

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

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

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 specifies a model architecture.

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

AWESOME

Excellent work providing the performance achieved with each of the models!

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.

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

Accuracy on the test set is 60% or greater.

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.

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.

AWESOME

The function has a different output for each detected image type (dog, human). It would have been good to also test images that don't contain a dog or human.

Step 7: Test Your Algorithm

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

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

SUGGESTION

A few ways to improve the algorithm's performance include:

1. Training it on more data
2. Ensemble or combine several models together.
3. Stacking

For further reading on ensembling and stacking, see these links:

1. Ensemble Methods in Machine Learning: What are They and Why Use Them?
2. Stacking Made Easy
3. A Kaggler's Guide to Model Stacking in Practice

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review