UDACITY

‹ Return to Classroom

Dog Breed Classifier

REVIEW

HISTORY

Meets Specifications

Congratulations on meeting all specifications for this project! The next step would be to port the code over to an application or even train the ConvNet to recognize other animals (not just dogs). Keep up the good work!

More useful resources

- https://mashable.com/2018/03/27/nvidia-unveils-ai-supercomputer/#enrelfSQtaq6
- https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0
- https://www.kaggle.com/c/imagenet-object-detection-challenge
- https://www.kaggle.com/c/imagenet-object-detection-from-video-challenge
- https://www.kaggle.com/c/nips-2017-defense-against-adversarial-attack
- https://medium.com/towards-data-science/transfer-learning-using-keras-d804b2e04ef8
- http://cs231n.github.io/transfer-learning/
- http://www.slate.com/articles/technology/future_tense/2016/04/the_philosophical_argument_against_artificial_intelligence_killing_us_all.html
- http://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/parkhi15.pdf

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.

SUGGESTION

I recommend experimenting with batch normalization in the network. Batch normalization works by normalizing the weighted inputs in the network. Without it, only the inputs at the beginning of the network are normalized. To learn more, see this link: https://www.quora.com/Why-does-batch-normalization-help.

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.

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.

Step 7: Test Your Algorithm

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

SUGGESTION

This time around, it doesn't seem as though there's a response to question 6 in the html version of the file. I recommend updating the html version with a response, especially if you intend to upload the file to Github or a blog.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this project

START