UDACITY

‹ Back to Deep Learning Nanodegree

# Dog Breed Classifier

REVIEW

CODE REVIEW

HISTORY

SHARE YOUR ACCOMPLISHMENT!

## Meets Specifications

The project is complete.
Excellent experimentation, add-ons and thought put into everything.
Hope you found the material interesting and enjoyable.
Great job and good luck going forward.

### 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.**

Nice experimentation and using dropout.
When you next have the chance, have a look at batch normalization as it is now just about standard and helps with training speed and other aspects of performance.
https://keras.io/layers/normalization/

Also, ELU is gaining in popularity vs Relu activation though they are similar.

https://keras.io/layers/advanced-activations/#elu

✓

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

✓

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

Good % for this section.

## 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.**

Rmsprop again allows for some baseline comparison with earlier models but other optimizers may improve performance. SGD and Adam are commonly used.

https://keras.io/optimizers/

✓

**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.**

Xception is, overall, the best performer of the group in terms of accuracy.

✓

**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.**

Dog before face detector, that is correct ordering.

## Step 7: Test Your Algorithm

✓

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

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

☆☆☆☆☆

Student FAQ