

[< Back to Deep Learning Nanodegree](#)

# Dog Breed Classifier

## REVIEW

## CODE REVIEW

## HISTORY

### Requires Changes

#### 1 SPECIFICATION REQUIRES CHANGES

Your improvements have been noted. Couple of very minor changes and the project will be complete.  
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.

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.

Output format looks good.

Please, add at least 1 more human image to achieve the 2 human image and 6 total test image specification.

Also, please, add at least 2 more points of improvement as per:

Question 6: Is the output better than you expected :) ? Or worse :( ? Provide at least three possible points of improvement for your algorithm.

Have taken your statement,

"If we needed to recognize hybrids, we could extend our labeled dog data to include labeled images of hybrid breeds and train it to recognize these too."  
as one of them.

For possible suggestions:

<https://machinelearningmastery.com/improve-deep-learning-performance/>

Just a note that, if you have your test images isolated in a folder you could loop through the folder and run `dog_predictor()`. Less typing or cut and paste and slightly cleaner code.

 RESUBMIT [DOWNLOAD PROJECT](#)

Learn the [best practices for revising and resubmitting your project](#).

[RETURN TO PATH](#)