

[< Back to Deep Learning](#)

Dog Breed Classifier

REVIEW

CODE REVIEW

HISTORY

Requires Changes

2 SPECIFICATIONS REQUIRE CHANGES

Great job on getting good results. A couple updates on the questions are required then you should be done. Almost there!

Files Submitted

The submission includes all required, complete notebook files.

Step 1: Detect Humans

The submission returns the percentage of the first 100 images in the dog and human face datasets that include a detected, human face.

Step 2: Detect Dogs

Use a pre-trained VGG16 Net to find the predicted class for a given image. Use this to complete a `dog_detector` function below that returns True if a dog is detected in an image (and False if not).

The submission returns the percentage of the first 100 images in the dog and human face datasets that include a detected dog.

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

Write three separate data loaders for the training, validation, and test datasets of dog images. These images should be pre-processed to be of the correct size.

Answer describes how the images were pre-processed and/or augmented.

The submission specifies a CNN architecture.

Answer describes the reasoning behind the selection of layer types.

Good description of the layers used and rationale behind using them.

I've seen good performance with an additional Conv layer, larger initial kernel size, or experimenting with Dropout layers after the max pooling layers.

Choose appropriate loss and optimization functions for this classification task. Train the model for a number of epochs and save the "best" result.

The trained model attains at least 10% accuracy on the test set.

Test Accuracy: 19% (161/836)

Step 4: Create a CNN Using Transfer Learning

The submission specifies a model architecture that uses part of a pre-trained model.

resnet34

The submission details why the chosen architecture is suitable for this classification task.

Please add some details to address the second part of the question asked. Why is this problem suitable for using transfer learning on the current problem?

Train your model for a number of epochs and save the result with the lowest validation loss.

Accuracy on the test set is 60% or greater.

Test Accuracy: 84% (709/836)

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 5: Write Your Algorithm

The submission uses the CNN from the previous step 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 6: Test Your Algorithm

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

Submission provides at least three possible points of improvement for the classification algorithm.

Images look good unfortunately it looks like you missed question 6. Please update with an answer. Was the output correct? Were any misses understandable? How can it be improved?

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

 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](#)

