

[◀ Return to "Deep Learning" in the classroom](#)[DISCUSS ON STUDENT HUB](#)

Dog Breed Classifier

REVIEW

HISTORY

Meets Specifications

Dear student,

Thanks for submitting the corrected version of your project. It looks like you are very comfortable coding CNNs and transfer learning. As such, this submission meets all requirements to pass.

I hope you found this part of the course useful, and keep up the great work 😊

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.

Great work computing the requested values.

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.

Great job augmenting the training data only

The submission specifies a CNN architecture.

Answer describes the reasoning behind the selection of layer types.

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.

Great accuracy!

Step 4: Create a CNN Using Transfer Learning

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

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

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.

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.

Nice job testing out your algorithm! I would also recommend testing on some images from outside the original dataset. This will give a good indication of the ability of the model to generalize.

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

All very good suggestions

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)

[Rate this review](#)