UDACITY

< Return to "Deep Learning" in the classroom

DISCUSS ON STUDENT HUB

Dog Breed Classifier

REVIEW

HISTORY

Requires Changes

2 SPECIFICATIONS REQUIRE CHANGES

Avid Udacity Student,

You are almost done. 👍 The submission is quite good and I must admit that you have showcased a good understanding of this project. However, there is one last specification to be fulfilled by providing an answer to `Question 5` in the notebook. You might have l=overlooked it. Please include the answer for the next submission. Good luck and keep up the good work!

Pro Tips

The following links may be of interest:

- How to improve my test accuracy using CNN in Tensorflow.
- A Guide to TF Layers: Building a Convolutional Neural Network.
- Recurrent Neural Networks.

Files Submitted

The submission includes all required, complete notebook files.
All necessary files are present in this submission. Great!

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.

Nice work in using the function `face_detector` and getting the appropriate percentage of the files!

> 98.0 % the images in human_files_short have a detected dog
> 17.0 % the images in dog_files_short have a detected dog

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).

Well done using pre-trained VGG16 Net to find the predicted class for images as seen in the `VGG16_predict()` function and completing dog_detector function.

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

You have achieved superb results. Your algorithm is optimal and does the job expected. Great!

> 1.0 % the images in human_files_short have a detected dog
> 91.0 % the images in dog_files_short have 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.

Awesome! Three separate data loaders for the training, validation, and test datasets of dog images were created. These images were pre-processed to be of the correct size.

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

Nice work describing your pre-processing steps in question 3, including augmentation of the training dataset through randomly flipping and rotating. 👍

Suggestions and Comments

- If you have time, I highly suggest you check out this article about Dog breed classification that utilizes CNN. It is an in-depth study that would help you understand how CNN works in classifying dog breeds. It is quite similar to the project so you might have an easier time understanding this due to your acquired knowledge.
- Here is a study that utilized Data Augmentation using GAN in classifying breeds.

The submission specifies a CNN architecture.

Well done! The submission correctly specifies a CNN architecture with five convolutional layers. This is deep enough to detect complex patterns in the dataset, producing good predictions. Good job using dropout to reduce overfitting.

Suggestions and Comments

I suggest you read this document about Using Convolutional Neural Networks to Classify Dog Breeds. This is a good outside resource to complete your reading of this course.

Pro Tips

The following links may be of great interest to you for further research:

- Keras Tutorial: The Ultimate Beginner's Guide to Deep Learning in Python.
- Deep learning for complete beginners: convolutional neural networks with keras.
- keras tutorial – build a convolutional neural network in 11 lines.
- Image Classification using Convolutional Neural Networks in Keras.

Answer describes the reasoning behind the selection of layer types.

Nice work outlining the steps you took to get to the final CNN architecture!

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

Nice! Architecture was compiled using `CrossEntropyLoss()` as loss function and `SGD` as optimizer. 👍

Please note that the "best" model should be informed by the validation and training datasets. ex. If the training loss decreases but the validation loss does not, then that model may be overfitting the training data and should not be chosen. The model with the lowest validation loss is typically best.

Pro Tips

- The Adam optimizer does sometimes work well but also can generalize poorly. Check out this article for more details.
- It's usually best to try an ada-based method (adam, nadam, etc) and SGD with momentum, and use whichever has better generalization (test/validation set) performance.
- Here is a comparison of some different optimizers.

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

Accuracy on the test set is 14%, this is good for a from-scratch model! 👍

Suggestions and Comments

I would like to share this article with you wherein the discussion is improving the accuracy using CNN. This may help you out in the future. 😃

Step 4: Create a CNN Using Transfer Learning

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

Required

Not bad! However, the submission must specify a model architecture. Please outline the steps you took to get to your final CNN architecture and your reasoning at each step.

Please answer Question 5.

> **Question 5:** Outline the steps you took to get to your final CNN architecture and your reasoning at each step. Describe why you think the architecture is suitable for the current problem.

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

Required

An answer to question 5 is expected here to meet the specification. In your answer, you must detail why the chosen architecture succeeded in the classification task.

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

Well done training 20 epochs of the model with SGD optimizer and CrossEntropyLoss() loss function!

Pro Tips

Note that during training, if the training loss decreases but the validation loss does not, then that model may be overfitting the training data. The model with the lowest validation loss is typically best.

Here are some documents that talk about the choice of the number of epochs:

- How does one choose optimal number of epochs?
- How to train your Deep Neural Network;
- Number of epochs to train on.

Accuracy on the test set is 60% or greater.

Fantastic implementation in calculating the accuracy! `75%` is a great number!

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.

The submission contains a function `predict_breed_transfer()` that takes a file path to an image as input and returns the dog breed that is predicted by the CNN. Fabulous work!

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.

The implemented algorithm in `run()` uses the CNN from Step 5 to detect dog breed. Furthermore, the submission has different output for each detected image type (dog, human, other), and provides either predicted actual (or resembling) dog breed. Well done! 👍

Step 6: Test Your Algorithm

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

The submission tests several human and dog images as well as images with neither a dog nor a human, getting good classifications. Excellent predictions! Well done! 👍

Suggestions and Comments

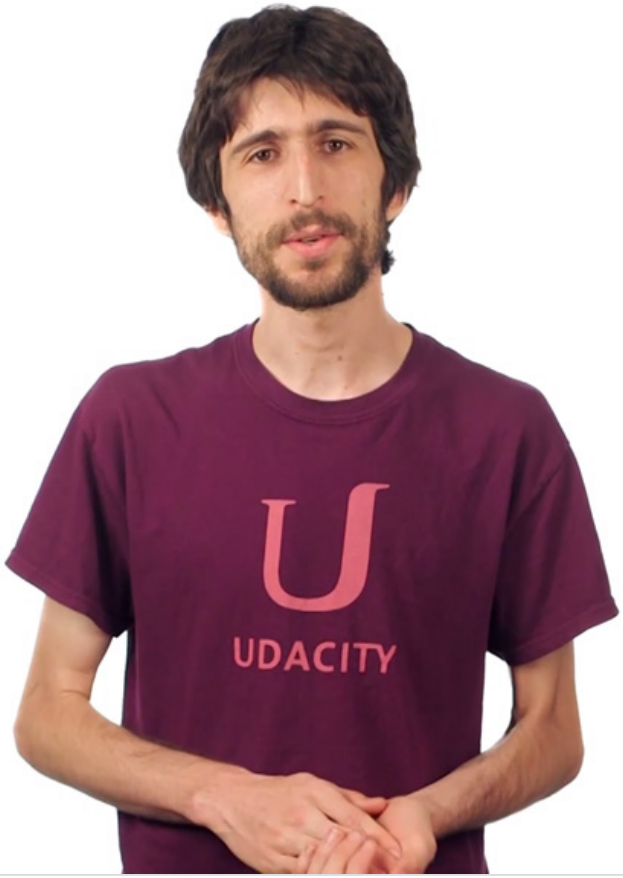
- I would suggest that you utilize a `for loop` in displaying and classifying the images. You can get the length of the folder that contains the images and loop it until it is done classifying all images. Here is a brief explanation of the for loop.

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

I like that you provided some possible improvements that can be made to make the model more robust. Try out some of the improvement points when you have the chance. 😄

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