UDACITY

Return to Classroom

Dog Breed Classifier

REVIEW

HISTORY

Meets Specifications

Bright Learner,
That's a solid submission. A lot of good things has been observed in your work. Well organized answers, simple and optimal implementation. Congratulations! All the rubrics are okay. Nice work! I invite you to visit all the suggestions and tips provided in this feedback. Thank you! Keep this professional spirit.

If you liked this review, can you please do well rate it from the star ratings :) Also, This submission of yours could be nominated for excellence, so it will be helpful if can you tell me the biggest challenge you faced in finishing this project, getting your feedback is always a pleasure, I will be very gladly to hear about your thoughts :) Thanks in advance!

Files Submitted

The submission includes all required files.

The required files for this submission include: `dog_app.ipynb` file, an HTML or PDF export of the project notebook with the name `report.html` or `report.pdf` and a folder containing extra images if any additional images were used.

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 percent of correct human identification: 98%.
The percent of dogs misclassified as humans: 11%.
The above values are acceptable. Good job!

The submission opines whether Haar cascades for face detection are an appropriate technique for human detection.

Using images of faces in different orientations. Also, a previously trained CNN that has already been trained to identify human face. You have provided a good answer! Your opinion is good and reasonable. :+1:

Pro Tips

Here are some documents that provide information on Haar cascades.

- Tutorial face detection;
- https://www.youtube.com/watch?v=88HdqNDQsEk;
- Opencv face detection;
- OBJECT DETECTION : FACE DETECTION USING HAAR CASCADE CLASSFIERS;
- Youtube video;

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.

Great results here! The algorithm is optimal and does the job expected, good work on using `dog_detector`.

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

The submission specifies a CNN architecture.

Superb work in describing the architecture used. It is a good idea to not stray away too much from what you have learned and use a bit of it as a base to learn and formulate better architectures. Nice job on this one.

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

Nice choice with 10 epochs in training the model.

Pro Tips

Here are several 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.

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

A test accuracy with data augmentation of 5.3828% is superb and easily beats the required 1% accuracy! Excellent job!

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 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 correctly downloads the bottleneck features. Nice work!

The submission specifies a model architecture.

Fantastic architecture for your model. It would really be nice if your model outperforms our base model in this project!

The submission details why the chosen architecture succeeded in the classification task and why earlier attempts were not as successful.

Excellent work! We note that, this 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 loss function and optimizer are specified in the compilation of this architecture. That's great!

The submission uses model checkpointing to train the model and saves the model weights with the best validation loss.

Nice work with the model checkpointing!

The submission loads the model weights that attained the least validation loss.

Good job loading the model weights that attained the least validation loss!

Accuracy on the test set is 60% or greater.

Fantastic implementation in calculating for the accuracy! 83.3732% is a great number!
Impressive work in calculating for the 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.

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

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.

Wow! The submission 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.

Step 7: Test Your Algorithm

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

Excellent work with testing your algorithm and getting great results. You did well on detecting and classifying the dog breeds. It is still interesting on how the model classifies humans as dogs based on its algorithm. 😃

DOWNLOAD PROJECT

RETURN TO PATH