UDACITY

Return to "Deep Learning" in the classroom

Dog Breed Classifier

REVIEW

HISTORY

Meets Specifications

Congratulations !!

Awesome job on completing your project !! Your project meets all the specifications. All the best for your future projects 👍

Useful Links

1. CNN's for Visual Recognition
2. Deep Conv nets for image classification
3. Large Scale image Recognition using DNN's
4. Transfer Learning
5. Awesome Deep Learning Papers

Suggestions to Make Your Project Stand Out!

TURN YOUR ALGORITHM INTO A WEB APP

- Turn your code into a web app using Flask or web.py

OVERLAY DOG EARS ON DETECTED HUMAN HEADS

- Overlay a Snapchat-like filter with dog ears on detected human heads. You can determine where to place the ears through the use of the OpenCV face detector, which returns a bounding box for the face. If you would also like to overlay a dog nose filter, some nice tutorials for facial keypoints detection exist here.

ADD FUNCTIONALITY FOR DOG MUTTS

- Currently, if a dog appears 51% German Shephard and 49% poodle, only the German Shephard breed is

returned. The algorithm is currently guaranteed to fail for every mixed breed dog. Of course, if a dog is predicted as 99.5% Labrador, it is still worthwhile to round this to 100% and return a single breed; so, you will have to find a nice balance.

EXPERIMENT WITH MULTIPLE DOG/HUMAN DETECTORS

- Perform a systematic evaluation of various methods for detecting humans and dogs in images. Provide improved methodology for the `face_detector` and `dog_detector` functions.

Files Submitted

✓

The submission includes all required, complete notebook files.

Notebook and PDF export are included 👍

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.

Good job discussing the architecture of your CNN.

Suggested Reading : Systematic evaluation of CNN advances on the ImageNet

✓

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.

Step 4: Create a CNN Using Transfer Learning

✓

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

The reason you mentioned for using dropout is not correct. I would suggest you to got through this blog post once to get a better understanding. Below are links to some of the famous papers on different deep learning architectures:

- Le Net 5 - Gradient Based Learning Applied to Document Recognition
- Alex Net - ImageNet Classification with Deep Convolutional Neural Networks
- VGG 16 - VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION
- GoogLeNet - Going Deeper with Convolutions
- Xception : Deep Learning with depthwise separable Convolutions

✓

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 wth the lowest validation loss.

✓

Accuracy on the test set is 60% or greater.

Good job getting an accuracy of 85%.

✓

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.

DOWNLOAD PROJECT

RETURN TO PATH