UDACITY

PROJECT SPECIFICATION

Dog Breed Classifier

Files Submitted

CRITERIA	MEETS SPECIFICATIONS
Submission Files	The submission includes all required files.

Step 1: Detect Humans

CRITERIA	MEETS SPECIFICATIONS
Question 1: Assess the Human Face Detector	The submission returns the percentage of the first 100 images in the dog and human face datasets with a detected human face.
Question 2: Assess the Human Face Detector	The submission opines whether Haar cascades for face detection are an appropriate technique for human detection.

Step 2: Detect Dogs

CRITERIA	MEETS SPECIFICATIONS
Question 3: Assess the Dog Detector	The submission returns the percentage of the first 100 images in the dog and human face datasets with a detected dog.

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

CRITERIA	MEETS SPECIFICATIONS
Model Architecture	The submission specifies a CNN architecture.
Train the Model	The submission specifies the number of epochs used to train the algorithm.
Test the Model	The trained model attains at least 1% accuracy on the test set.

Step 5: Create a CNN to Classify Dog Breeds

CRITERIA	MEETS SPECIFICATIONS
	The submission downloads the bottleneck features corresponding to one of the Keras pre-

CRITERIA	MEETS SPECIFICATIONS
Obtain Bottleneck Features	trained models (VGG-19, ResNet-50, Inception, or Xception).
Model Architecture	The submission specifies a model architecture.
Question 5: Model Architecture	The submission details why the chosen architecture succeeded in the classification task and why earlier attempts were not as successful.
Compile the Model	The submission compiles the architecture by specifying the loss function and optimizer.
Train the Model	The submission uses model checkpointing to train the model and saves the model weights with the best validation loss.
Load the Model with the Best Validation Loss	The submission loads the model weights that attained the least validation loss.
Test the Model	Accuracy on the test set is 60% or greater.

CRITERIA	MEETS SPECIFICATIONS
Predict Dog Breed with the Model	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 6: Write Your Algorithm

CRITERIA	MEETS SPECIFICATIONS
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.

Step 7: Test Your Algorithm

CRITERIA	MEETS SPECIFICATIONS
Test Your Algorithm on Sample Images!	The submission tests at least 6 images, including at least two human and two dog images.

Suggestions to Make Your Project Stand Out!

(Presented in no particular order ...)

(1) AUGMENT THE TRAINING DATA

Augmenting the training and/or validation set might help improve model performance.

(2) TURN YOUR ALGORITHM INTO A WEB APP

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

(3) 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.

(4) 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.

(5) 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.

Student FAQ