HIMAKIRAN KUMAR
01 FEB 2020

Udacity Capstone Proposal : Dog Breed Classifier

HIMAKIRAN KUMAR

PROPOSAL

Domain Background

Convolutional Neural Networks are specialized kinds of neural networks that have been very successful particularly at computer vision tasks, such as recognizing objects, scenes, and faces, among many other applications. CNNs work by discovering low-level and high-level features and use them to classify images. They are most effective when classifying different types of objects say human faces from backgrounds or vehicles from traffic. However when we do use CNNs for classifying different types of the same object, the task becomes difficult to delineate higher features.

Problem Statement

The problem being tackled in this project is to pickup a type of breed of dogs from a given set of dog pictures. It can also be stated that the problem here is to classify dogs using a CNN according to the the type of breeds and use this model to take in a picture of the Dog and predict the breed of dog. Further on given an image of a human face, the nearest type of matching dog breed will be predicted.

Datasets and Inputs

There are two datasets and five pre-trained models available to us.

Dogs dataset : Dog dataset provided by Udacity deep learning v2 PyTorch repository. (https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip)

Humans dataset : Human dataset provided by Udacity deep learning v2 PyTorch repository. (http://vis-www.cs.umass.edu/lfw/lfw.tgz)
There are 13233 total human images and 8351 total dog images.
There are 6680 dog images in the train folder and 836 dog images in the test folder and 835 dog images in the valid folder.
We also have the following five pre-trained models available to us
DogInceptionV3Data.npz , DogXceptionData.npz, DogVGG19Data.npz, DogResnet50Data.npz and DogVGG16Data.npz.

There are 133 breeds of dogs and the breed-wise distribution of the dogs is as follows

```
['103.Mastiff',
 '059.Doberman_pinscher',
 '055.Curly-coated_retriever',
 '031.Borzoi',
 '024.Bichon_frise',
 '049.Chinese_crested',
 '067.Finnish_spitz',
 '130.Welsh_springer_spaniel',
 '019.Bedlington_terrier',
 '115.Papillon',
 '126.Saint_bernard',
 '013.Australian_terrier',
 '116.Parson_russell_terrier',
 '107.Norfolk_terrier',
 '133.Yorkshire_terrier',
 '032.Boston_terrier',
 '108.Norwegian_buhund',
 '028.Bluetick_coonhound',
 '066.Field_spaniel',
 '129.Tibetan_mastiff',
 '005.Alaskan_malamute',
 '102.Manchester_terrier',
 '034.Boxer',
 '068.Flat-coated_retriever',
 '089.Irish_wolfhound',
 '104.Miniature_schnauzer',
 '035.Boykin_spaniel',
 '080.Greater_swiss_mountain_dog',
 '007.American_foxhound',
 '112.Nova_scotia_duck_tolling_retriever',
 '025.Black_and_tan_coonhound',
 '072.German_shorthaired_pointer',
 '062.English_setter',
 '029.Border_collie',
 '045.Cardigan_welsh_corgi',
```

```
'105.Neapolitan_mastiff',
'076.Golden_retriever',
'063.English_springer_spaniel',
'078.Great_dane',
'084.Icelandic_sheepdog',
'023.Bernese_mountain_dog',
'091.Japanese_chin',
'011.Australian_cattle_dog',
'021.Belgian_sheepdog',
'041.Bullmastiff',
'098.Leonberger',
'018.Beauceron',
'020.Belgian_malinois',
'016.Beagle',
'039.Bull_terrier',
'087.Irish_terrier',
'064.English_toy_spaniel',
'123.Pomeranian',
'097.Lakeland_terrier',
'127.Silky_terrier',
'120.Pharaoh_hound',
'006.American_eskimo_dog',
'012.Australian_shepherd',
'070.German_pinscher',
'095.Kuvasz',
'131.Wirehaired_pointing_griffon',
'125.Portuguese_water_dog',
'071.German_shepherd_dog',
'003.Airedale_terrier',
'043.Canaan_dog',
'118.Pembroke_welsh_corgi',
'010.Anatolian_shepherd_dog',
'033.Bouvier_des_flandres',
'106.Newfoundland',
'047.Chesapeake_bay_retriever',
'009.American_water_spaniel',
```

```
'065.Entlebucher_mountain_dog',
'002.Afghan_hound',
'054.Collie',
'093.Kerry_blue_terrier',
'061.English_cocker_spaniel',
'082.Havanese',
'044.Cane_corso',
'056.Dachshund',
'026.Black_russian_terrier',
'132.Xoloitzcuintli',
'094.Komondor',
'022.Belgian_tervuren',
'114.Otterhound',
'036.Briard',
'074.Giant_schnauzer',
'017.Bearded_collie',
'110.Norwegian_lundehund',
'085.Irish_red_and_white_setter',
'069.French_bulldog',
'075.Glen_of_imaal_terrier',
'042.Cairn_terrier',
'004.Akita',
'060.Dogue_de_bordeaux',
'128.Smooth_fox_terrier',
'083.Ibizan_hound',
'117.Pekingese',
'081.Greyhound',
'051.Chow_chow',
'040.Bulldog',
'008.American_staffordshire_terrier',
'046.Cavalier_king_charles_spaniel',
'099.Lhasa_apso',
'090.Italian_greyhound',
'050.Chinese_shar-pei',
'086.Irish_setter',
'037.Brittany',
```

```
'121.Plott',
'014.Basenji',
'030.Border_terrier',
'079.Great_pyrenees',
'096.Labrador_retriever',
'027.Bloodhound',
'048.Chihuahua',
'119.Petit_basset_griffon_vendeen',
'124.Poodle',
'058.Dandie_dinmont_terrier',
'052.Clumber_spaniel',
'038.Brussels_griffon',
'113.Old_english_sheepdog',
'057.Dalmatian',
'053.Cocker_spaniel',
'122.Pointer',
'077.Gordon_setter',
'073.German_wirehaired_pointer',
'088.Irish_water_spaniel',
'111.Norwich_terrier',
'109.Norwegian_elkhound',
'001.Affenpinscher',
'015.Basset_hound',
'101.Maltese',
'092.Keeshond',
'100.Lowchen']
```

Solution Statement

The solution of this project shall consist of the following

- A CNN model built using PyTorch to distinguish Dog breeds.
- A web application that accepts a picture of a dog and predicts the breed type. If human face is detected it shall output the nearest dog breed match.

Benchmark Model

The CNN model built from scratch shall serve as benchmark model to the model built using transfer learning.

Evaluation Metric

The evaluation metric which shall be used is the accuracy for both the benchmark and the solution model as the dataset and the number of images shall remain the same for both the models.

Project Design

We shall follow the steps as given below to realize our solution.

1. Import Datasets. We shall import the given dog and human datasets.
2. Detect Humans. We shall use OpenCV's implementation of Haar feature-based cascade classifiers to detect humans.
3. Detect Dogs. We shall detect the Dogs using VGG-16 model.
4. Creating CNN We shall create a CNN for detecting Dogs.
5. Creating CNN using Transfer Learning. We shall now create a CNN using transfer learning.
6. Algorithm. No we shall write an algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither. Then,
   - if a **dog** is detected in the image, return the predicted breed.
   - if a **human** is detected in the image, return the resembling dog breed.
   - if **neither** is detected in the image, provide output that indicates an error.
7. Testing. Finally we shall test this algorithm on at least six sample images.