Machine Learning Engineer Nanodegree Capstone Proposal
Dog Breed Classifier

Jasbir Singh
23-Feb-2021

1. Domain Background

Dog breed recognition is a challenging problem considering the number of dog breeds and similarities between some of them. This becomes a complex computer vision task for humans as they need to remember all the dog breed features and distinguish between similar looking features. Normal human mind and memory will struggle to solve this problem and high computing power and memory is required to process this information. Solution to this problem would involve tying together a series of models for different tasks related to identifying humans, identifying dogs and identifying dogs bread resembling human face. Many researchers have contributed to this specific problem related to computer vision and machine learning. Some of published literature can be referred to develop an understanding of the aforementioned field [5] [6] [7] and apply that knowledge to develop a solution which bears closest resemblance to the specific image classification problem.

2. Problem Statement

The aim of the project is to build a pipeline to process real-world user-supplied images. The algorithm is expected to identify dog's breed with certain level of accuracy for a given image and if a human image is passed to it, it will try to predict the closest matching dog breed. If the algorithm is not able to predict human or dog then an error will be returned. Evaluate a model to detect dog breeds, humans and subsequently classify dog breeds that resembles humans.

3. Datasets and Inputs

Datasets provided by Udacity will be used to train, validate and test the CNN models for different tasks. Following datasets will be used:

- ***Dogs Dataset***

  The dogs dataset has 8,351 images which are sorted into train (6,680 images), test (836 images) and valid (835 images) directories. Each of the directories train, test and valid has 131 folders corresponding to different dog breeds. Each image in dog breed folder has different size, color, background, and orientation. Number of images in each dog breed folder is not same.

- ***Human Dataset***

  The human dataset has 13,233 images which are sorted by person name and stored into respective 5,750 folders. 835 images) directories. All the image are of same size but with different background and angles. Each image in dog breed folder has different size, color, background, and orientation. Number of images in each person folder is not same.

4. Solution Statement

Convolutional Neural Network (CNN) model can be used to solve this complex classification problem. CNN is a deep learning algorithm which can be used to create the model and train it using the provided train datasets and get optimal values for learning rate, weights and bias. We can test and validate the model using test and valid data sets. Overall solution consists of multiple steps, detection of human images will be fulfilled using OpenCV's implementation of Haar feature-based cascade classifiers and Pre-trained VGG-16 model will be used for to detect dog images. Identification of dog breeds can be split into two steps, First step involves creating a CNN model with target test accuracy of at-least 10% on test dataset and Second one will use transfer learning to create a CNN model with target test accuracy of at-least 60% on test dataset. Once the image is identified as dog/human, it will passed to the CNN to predict dog bread for dog image and resembling dog bread for human image.

5. Benchmark Model

- CNN model created from scratch is expected to have an accuracy at-least 10% on test data which is better than a random guess. This is baseline accuracy which will be used to apply transfer learning
- CNN model created using transfer learning must have an accuracy of at-least 60% on test data.

6. Evaluation Metrics

Accuracy, error rate, precision, recall and F1 score will be used as the evaluation metics for Convolutional Neural Network (CNN) model.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Error rate = (FP + FN) / (TP + TN + FP + FN)

Precision = TP / (TP + FP)

Recall = TP / (TP + TN)

F1 Score = 2 * ((Precision*Recall) / (Precision + Recall))

TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative

7. Project Design

I will adopt following strategy for project design:

- Step 1: Import the provided datasets and required libraries

  Data imbalance will be addressed during Data preprocessing step. Images will be resized to have a uniform image resolution and data augmentation will be applied using image flips and rotations.

- Step 2: Detect Humans: use OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images.
  - Convert image into grayscale for input
  - Face detector function to detect face returning true/false

    -
  - Step 3: Detect Dogs: use a pre-trained VGG-16 model trained on ImageNet to detect dogs in images.
    - Initialize pre-trained VGG-16 model
    - Predict function returning class index (151 to 268) for the input image path
    - Dog detector function returning true/false for the input image path
  - Step 4: Create a CNN model from scratch using PyTorch to classify dog breeds.
    - Create data loaders for training, validation and test datasets
    - Apply data augmentation using image transformations (Resize, Rotate, Crop, Flip,Normalize)
    - Create CNN model describing convolutional layers, fully connected layers, dropout layer, pooling layer and forward path behavior
    - Specify Loss Function (CrossEntropyLoss) and Optimizer (SGD)
    - Train and validate the model minimizing the validation loss and subsequently save the model with minimal validation loss (50-100 epochs)
    - Test the model and verify the evaluation metrics.
  - Step 5: Create a CNN model using Transfer Learning using ResNet50 to classify dog breeds.
    - Create data loaders for training, validation and test datasets
    - Apply data augmentation using image transformations (Normalize,Resize, Rotate, Crop, Flip,Normalize)
    - Load the pre-trained ResNet50 model and freeze the model parameters
    - Add a fully connected layer with output size of 133
    - Specify Loss Function (CrossEntropyLoss) and Optimizer (SGD)
    - Train and validate the model minimizing the validation loss and subsequently save the model with minimal validation loss
    - Test the model and verify the evaluation metrics.
    - Predict breed function returning class name based on input image path
  - Step 6: Write the algorithm: accept an image file path as input and detect if image contains dog or human. Predict breed for dog image and resembling dog breed for human image and output error if neither dog or human is detected.
  - Step 7: Test the algorithm: Test the algorithm with different human, dog and generic images to see if model provides correct prediction or not.

8. References

1. CNN Model: https://en.wikipedia.org/wiki/Convolutional_neural_network
2. Udacity Dogs Data set: https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip
3. Udacity Human Data set: https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip

4. PyTorch Models - https://pytorch.org/docs/stable/torchvision/models.html
5. Modified Deep Neural Networks for Dog Breeds Identification by Aydin Ayanszadeh and Sahand Vahidnia: https://www.researchgate.net/publication/325384896_Modified_Deep_Neural_Networks_for_Dog_Breeds_Identification
6. Cats and Dogs: 2012 IEEE Conference on Computer Vision and Pattern Recognition: https://ieeexplore.ieee.org/document/6248092
7. Dog breed classification using part localization. Computer Vision: https://dl.acm.org/doi/10.1007/978-3-642-33718-5_13