UDACITY CAPSTONE PROPOSAL

DOG BREED IDENTIFICATION

CNN (Convolutional Neural Networks) Project

Domain Background:

Machine Learning has become very popular in recent times. It is new buzz word. All thanks to the new developed algorithms and increasing computation power of computers. The application of Machine learning is never ending in the current world. With big data available there is a need to make out useful information from it.

The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one algorithm — a *Convolutional Neural Network*. It is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNN can learn these filters/characteristics. The architecture of a CNN is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area.

The problem taken of dog breed identification is very popular. Here is the Kaggle link to it for more description: https://www.kaggle.com/c/dog-breed-identification/overview/description

Problem Statement:

This project uses Convolutional Neural Network with Transfer Learning to identify the dog breeds. Transfer learning has advantage that it can decrease time to develop and train a model by reusing the modules of already developed models. Therefore, the model training process speeds up.

The algorithm must achieve three objectives:

i) Dog Detector: given an image of a dog, algorithm will identify an estimate of the canine's breed

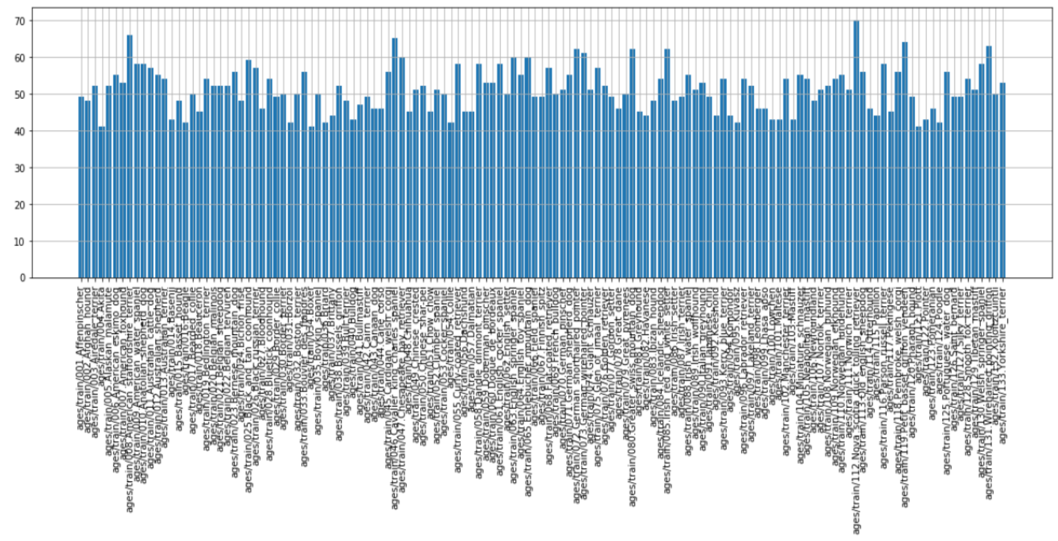
ii) Human Detector: If supplied an image of a human; the code will identify the resembling dog breed.

iii) If the image supplied is identified as neither human nor dog then the algorithm would output "Invalid Image".

Datasets and Inputs:

In the project we want to identify the breed of dog by looking at a image. So, the input format is an image. Dataset for this project has been provided by Udacity which contain images of humans and dogs. The link to the GitHub repo provided by Udacity for the problem: https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification

Dog Images dataset: This dataset has 8351 total images which are sorted in directories as train (6,680 images), test (836 images) and valid (835 images). Each directory (train, test and valid) have 133 folders corresponding to dog breeds. The image size and background are different. Number of images provided for each breed is not same.The no of images (y-axis) for different categories of dog breeds (x-axis) has been shown in the below graph.



Human Images Dataset: This dataset has 13233 total images which are sorted by names of human in 5750 folders. Images background and angles are different. All images of 250x250 size. The given data is unbalanced, some people have only single image while others have many.

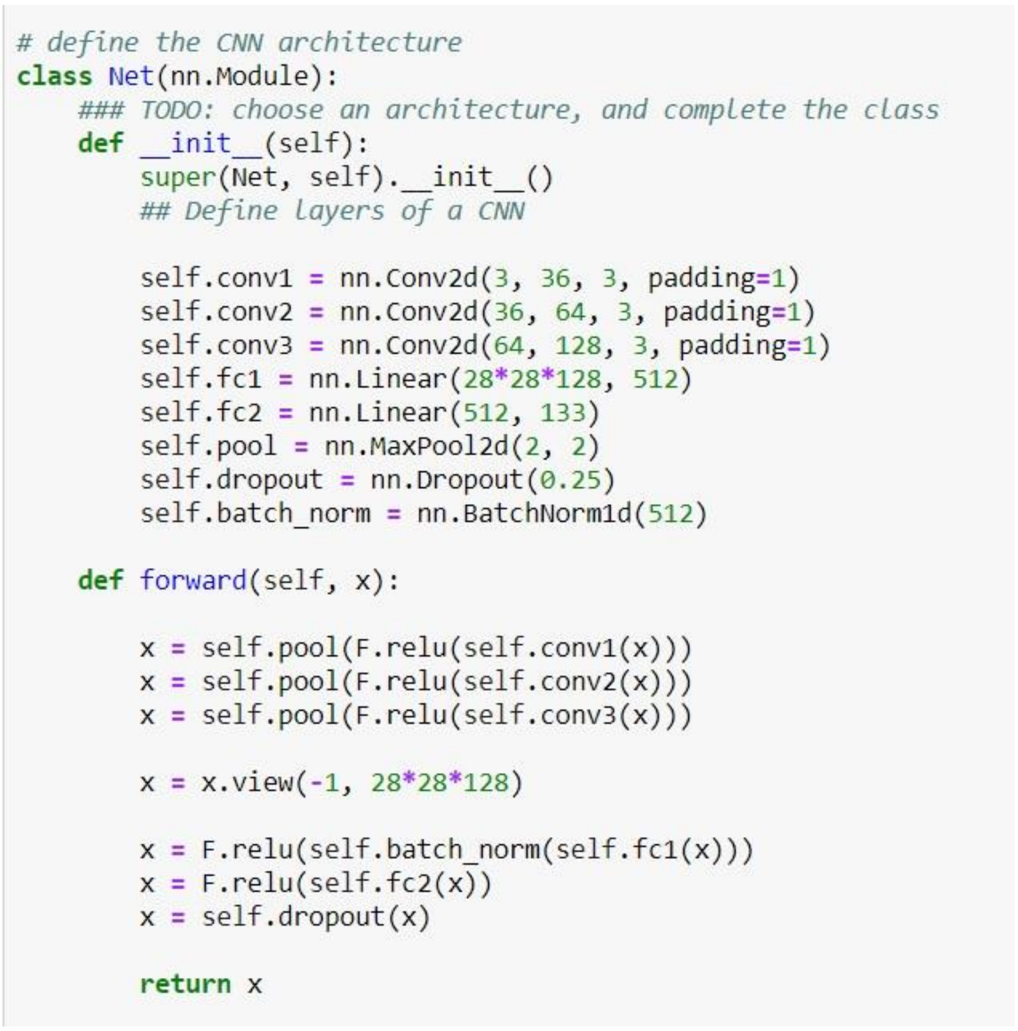
Solution Statement:

We use CNN (Convolutional Neural Network), a Deep Learning Algorithm to solve this problem. It takes up an image, provide weights and biases to various objects of image and help to differentiate one image from another. We first detect human images using an existing algorithm like OpenCV's implementation of Haar feature based cascade classifiers. Then, to detect dog images we use VGG16 model which is pretrained. Transfer learning is used here. There are three steps in transfer learning using pre trained model approach: Selecting source model, reusing model and tuning the model. The ResNet101 architecture is used which is a transfer learning model made with the help of convolution neural networks and residual blocks. The ResNet model is already trained on a huge image dataset, so we can use them to train our dataset by changing its input and output parameters only. At final stage we pass this image to CNN which will best predict the breed out of the 133 breeds. In case the image is of neither human nor dog, it output a message as "invalid image".

Benchmark Model:

CNN to Classify Dog Breeds (from Scartch):

CNN is constructed to classify dog breed from given images.



```python
# define the CNN architecture
class Net(nn.Module):
    ### TODO: choose an architecture, and complete the class
    def __init__(self):
        super(Net, self).__init__()
        ## Define layers of a CNN

        self.conv1 = nn.Conv2d(3, 36, 3, padding=1)
        self.conv2 = nn.Conv2d(36, 64, 3, padding=1)
        self.conv3 = nn.Conv2d(64, 128, 3, padding=1)
        self.fc1 = nn.Linear(28*28*128, 512)
        self.fc2 = nn.Linear(512, 133)
        self.pool = nn.MaxPool2d(2, 2)
        self.dropout = nn.Dropout(0.25)
        self.batch_norm = nn.BatchNorm1d(512)

    def forward(self, x):

        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = self.pool(F.relu(self.conv3(x)))

        x = x.view(-1, 28*28*128)

        x = F.relu(self.batch_norm(self.fc1(x)))
        x = F.relu(self.fc2(x))
        x = self.dropout(x)

        return x
```

This model has 3 convolutional layers. The first layer have in_channels =3 and the final layer gives output size of 128. All the layers have kernel size of 3. ReLu function is also used here.

Evaluation Metrics:

Multi class log loss is used to evaluate the model. It takes in account the uncertainty of prediction based of variation from the actual label. The non-uniformity of dataset is the reason for not taking accuracy as indicator.

Project Design:

1. Import necessary data and libraries. Preprocess the data and create train, test and validate sets. Perform image augmentation on training data.
2. Usage of OpenCV's implementation of Haar feature based cascade classifiers to detect human face.
3. Usage of pretrained VGG16 model to create dog detector.
4. Creation of CNN to classify dog breeds from scratch, train, validate and test the model.
5. Creation of CNN to classify dog breeds using transfer learning with resnet101 architecture. Train and test the model.
6. Write an algorithm to combine dog detector and human detector. If dog is identified, return predicted breed. If human is identified return the resembling dog breed and if neither is detected, provide the output indicating the error.

References:

1. Github repo of Project:
 https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification
2. Resnet101:
https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html#resnet101
3. Imagenet training in Pytorch:
https://github.com/pytorch/examples/blob/97304e232807082c2e7b54c597615dc0ad8f6173/imagenet/main.py#L197-L198
4. Pytorch Documentation
https://pytorch.org/docs/master/
5. CNN
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
6. Log Loss
http://wiki.fast.ai/index.php/Log_Loss
7. Transfer Learning
https://machinelearningmastery.com/transfer-learning-for-deep-learning/
8. ResNet:
https://medium.com/analytics-vidhya/enhance-learning-by-transfering-resnet-architecture-into-big-transform-architecture-44603f537fcf