

Definition

Project Overview:- In this project, we will learn how to build a pipeline to process real-world, user-supplied images. Given an image of a dog, our algorithm will identify an estimate of the dog breed and if an image is of a human, the code will identify the resembling dog breed. We will explore the state-of-the-art CNN models and transfer learning for classification. In this project, we will make the steps towards developing an algorithm that could be used as part of a mobile or web app

Problem Statement:- Our goal is that by completing this project, we will understand the challenges involved in piecing together a series of models designed to perform various tasks in a data processing pipeline. Each model has its strengths and weaknesses, and engineering a real-world application often involves solving many problems without a perfect answer. Since there are many algorithms available but we will try to use pre-trained and light model, so that whenever this model will be deployed as an app it will work fast and accurately.

Datasets and Inputs:- Since this project is available in udacity workspace, so dataset is available there, but if someone want to do on their own they are providing link for download. Dataset contains 13233 human and 8351 dog images. And in both dataset there are images for training, validation, and testing. These images contains required data for better training and testing of model. Datasets contains almost every type of images to learn and detect dog image and human image. The total categories of dog images is 133.

Solution Statement:- To check whether our model is doing good we will test on test images and after that for real-world testing will be perform by feed any random image to model. If model detect dog image then it will display dog with its breed otherwise it will display human and resembling dog breed. And if it doesn't detect neither dog nor human then it display can't detect anything.

Benchmark Model:- For benchmark model we will use OpenCV algorithm to detect human faces. For this we will use some images from the dataset for detection and check how many will it detect exactly. OpenCV provides many pre-trained face detectors. And for dog images we will use VGG16, which is trained on a very large dataset of images for classification and other vision tasks.

Evaluation Metrics:- Percentage of total items classified correctly,

$$\text{True Positive} / \text{total no. of images}$$

Since we are just checking how many out of total images is classified correctly. Because we are also using pre-trained model where dog categories lies between index 151 and 268. So if index is between given index then count 1 otherwise count 0 towards total number of correctly classified. Same is for human face detection but without any index. Because for human face model is purely trained on human faces.

Project Design:- First we will have large dataset of images. Images can be of any size and dimensions. So we will transform them to approx 224*224 pixels, so that we will have enough pixels of images to work with. We will also apply other transformations as well like crop, rotation, flip. After that we will apply normalization. These all operations will apply only on training data, For test and validation we will apply just resize and normalize. So that model will predict better on end user's images and not need to apply all those transformation operations. Since we are using pytorch we have to convert all images to tensor.

Now after that we will use Convolution Neural Network. We will define convolution layers and we will apply maxpooling at each layer. And activation function will be ReLU function. At the end we will flatten the input. For loss function we will use cross-entropy, because Cross-entropy loss measures the performance of a classification model whose output is a probability value between 0 and 1.

For Optimizer we will use Adam Optimizer, because adam is an adaptive learning rate method, it computes individual learning rates for different parameters.

After that we will define training methods to train our model with some number of epochs and calculate training loss and validation loss. Then for testing we will define test method and calculate test loss and accuracy.

For better prediction of the images we will also use transfer learning. Transfer learning make use of the knowledge gained while solving one problem and applying it to a different but related problem. Because in this project we will also predict resembling dog breed if a human is predicted. For that we will use Resnet18 pre-trained model. For this part we will use same optimizer and loss function as in convolution network. Then we will use already defined training and testing methods.

Then we will define a function that will transform image feed by end user, so that image will come in proper dimensions. And after that model will we used to predict breed of dog. Now the model is ready and can be deployed as a web app or phone app.

Reference:-

- ❖ https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html
- ❖ <https://towardsdatascience.com/what-is-transfer-learning-8b1a0fa42b4>