Machine Learning Engineer Nanodegree

Capstone Proposal

Aditya Vikram Bhattacharya
May 16th, 2019

Domain Background

I am instrested in using AI to conserve our environment. The current project is a step to take an initiative towards conserving the environment using AI. Similar to my intrest I found a competition on kaggle with some relevent data set. The current project would require image processing for its execution.

Tehuacán-Cuicatlán Valley is a semi-arid zone in the south of Mexico. It was inscribed in the World Heritage List by the UNESCO in 2018. This unique area has wide biodiversity including several endemic plants. Unfortunately, human activity is constantly affecting the area. A way to preserve a protected area is to carry out autonomous surveillance of the area. A first step to reach this autonomy is to automatically detect and recognize elements in the area. In this work, we present a deep learning based approach for columnar cactus recognition, specifically, the Neobuxbaumia tetetzo species, endemic of the Valley. The current approach is to find columnar cactus recognition.

---

kaggle: https://www.kaggle.com/irvingvasquez/cactus-aerial-photos.

Problem Statement

The impact of climate change on Earth's flora and fauna is a vital way to quantify how human activities such as logging, mining, and agriculture are impacting our protected natural areas and we have need a to assess this. The initiatives taken by the Researchers in Mexico have created the VIGIA project, which aims to build a system for autonomous surveillance of protected areas. The aim of the algorithm is to find a specific type of cactus in areal imagery and carry out its surviellence.

1. Download the given data set
2. Train a classifier that can determine if an image contains cactus or not.
3. Use a different pre trained model other than the one used.
4. Accuracy for the test set.

Datasets and Inputs

The souce for dataset was from kaggle. This dataset contains a large number of 32 x 32 thumbnail images containing aerial photos of a columnar cactus (Neobuxbaumia tetetzo). Kaggle has resized the images from the original dataset to make them uniform in size. The file name of an image corresponds to its id. The problem uses detection of cactus (Neobuxbaumia tetetzo) in frames. the

frames will be used as the input for the convolutional neural network. The number of images in the training set is 17500. The number of images in the test is 4000. Id that we'll assess is 'has_cactus'.

---

data:(https://www.kaggle.com/c/aerial-cactus-identification/data)

Solution Statement

I have thought of using a Convolutional Neural Network for image processing. For creating the model I have thought of using the pre trained model from ResNext.

ResNext : https://github.com/nutszebra/resnext/blob/master/resnext.py

Benchmark Model

The bench mark model that relates to the the project gives an accuracy of 0.977 that was achieved using VGG_16 pre traind model. According to my observation the VGG_16 model is very heavy so I'll try to use a lighter pre trained moel so that we get a more fast functioning model and also would try to beat the bench mark accuracy of 0.977.

VGG_16:https://github.com/tensorflow/models/blob/master/research/slim/nets/vgg.py

Evaluation Metrics

The evaluation metric for both the models to quantify the performance can be determined using ROC. An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

True Positive Rate False Positive Rate True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

TPR = $TP/(TP+FN)$

False Positive Rate (FPR) is defined as follows:

FPR = $FP/(FP+TN)$

- ROC: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
- :True Positive
- TN :True Negative
- FN :False Neagtive
- FP :False Positive

Project Design

Pre Processing

Getting the 4D tensor ready for ResNext-50, and for any other pre-trained model in Keras, requires some additional processing. First, the RGB image is converted to BGR by reordering the channels. All pre-trained models have the additional normalization step that the mean pixel. (expressed in RGB as

$[103.939, 116.779, 123.68]$ and calculated from all pixels in all images in ImageNet) must be subtracted from every pixel in each image.We wouln't require much of cleaning in pre processing. The data is mostly clean that we have.

The Neural Net

Since people have already implemented different methods like VGG_16, VGG_19, fast_ai and fastai.vision. I have thought of implementing Resnext. The input layer followed by a dense layer followed by a drop layer to avoid overfitting. The input layer will have 32 nodes .The first layer will have 64 nodes followed by drop out layer. The next dense layer will have 128 nodes followed by a drop out layer again followed by a layer of 256 nodes with a drop out layer. The current approach is to test the architecture till the time we reach some accuracy of around 0.70. Then we'll keep adjusting the architecture with necessary tweaks here and there.

Training and Testing

The architecture decided will be trained with a batch size say 20. If the processing speed is not satisfactory we might reduce the batch size. Similarly if the number of epochs do not improve after a point. We can reduce the number of epochs and check if the desired accuracy is achived. Post that we can compare the ROC of the model to check performance.