Machine Learning Advanced Nanodegree

Capstone Project

Mayur Nehete

1. Definition:

1.1 Project Overview:

Most of the population in the world survive on the plants for our food, and for medical use. Plants are most important part of our ecosystem. We get food, build shelter, make clothes from plants, three basic things human beings require to survive. So it is very important for human beings to plants more trees. We human should be able to know which type of plant it is. The Farmers only knows the type of plant the plant is just by looking at plant. As plants is important part of our daily file we should be able to classify plants just by looking at the plants. So that we can know the type of plant is, is it hazardous or good for human beings?

So Aarhus University Department of Engineering Signal Processing Group put together dataset and hosted competition on kaggle and asked machine learning and deep learning practitioners to build the model that can identify the 12 different species of the plants at different stages of the growth. So in Future we can build a robot who will be able to identify the type of species.

1.2 Problem Statement:

The dataset is labeled as 12 different species of the plants such as:

1. Black-grass
2. Charlock
3. Cleavers
4. Common Chickweed
5. Common wheat
6. Fat Hen
7. Loose Silky-bent
8. Maize
9. Scentless Mayweed
10. Shepherds Purse
11. Small-flowered Cranesbill
12. Sugar beet

The goal is to predict the likelihood that a plants is of certain species from the provided species.
So it is multi-class classification problem in machine learning terms.

The goal is create the model that would be able to classify plants into the 12 species of plants.

Deep learning is very popular classification techniques use for classification. In these project transfer learning with Deep learning network will be use to train a model to to classify images of plants to their respective species.

Transfer learning refers to the process of using the weights from pretrained networks on large dataset. As the pretrained networks have already learnt how to identify lower level features such as edges, lines, curves etc with the convolutional layers which is often the most computationally time consuming parts of the process, using those weights help the network to converge to a good score faster than training from scratch.

To train a model from scratch successfully, the dataset needs to be huge ,but there is not the case here. The provided dataset from Kaggle is very small, only 4750 images for training and machines with higher computational power with GPU is needed.As I don't have access to GPU at this time we will used the publicly available such as RESNET, InceptionV3, VGG16, Xception pretrained on imagenet challenge. And i'll be using Xception model as it has good accuracy on imagenet challenge and it is very deep model with 126 hidden layers with only 22,910,480 parameters.

1.3 Metrics:

The metric used for this Kaggle competition is MeanFScore, which is weighted average of Precision and Recall.

Precision:

Precision is the ratio of correctly predicted positive labels to the total predicted positive labels.

$$Precision = \frac{\sum_{k \in C} TP_k}{\sum_{k \in C} TP_k + FP_k}$$

Recall:

Recall is the ratio of correctly predicted positive labels to the all actual labels.

$$Recall = \frac{\sum_{k \in C} TP_k}{\sum_{k \in C} TP_k + FN_k}$$

MeanFScore:

$$MeanFScore = F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

MeanFScore score reaches its best value at 1 and worst score at 0. This score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but MeanFScore is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

2. Analysis

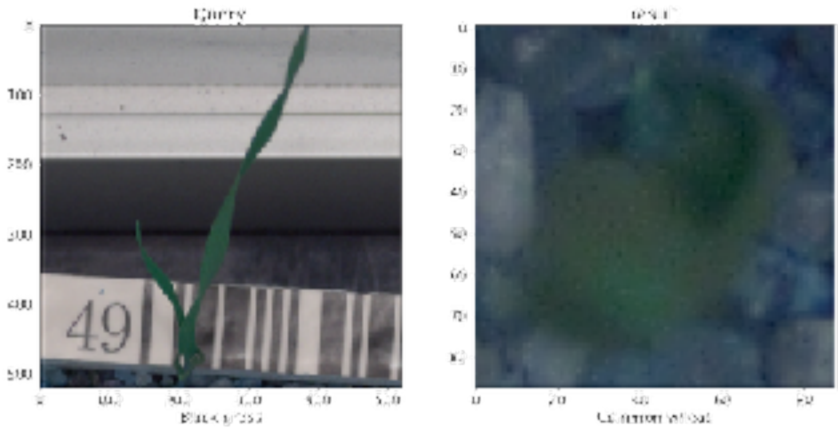
2.1 Data Overview

To create the dataset Aarhus University Department of Engineering Signal Processing Group take picture of plants at there various stages of there growth. This data set is created because to provide researchers a foundation for training weed recognition algorithms.

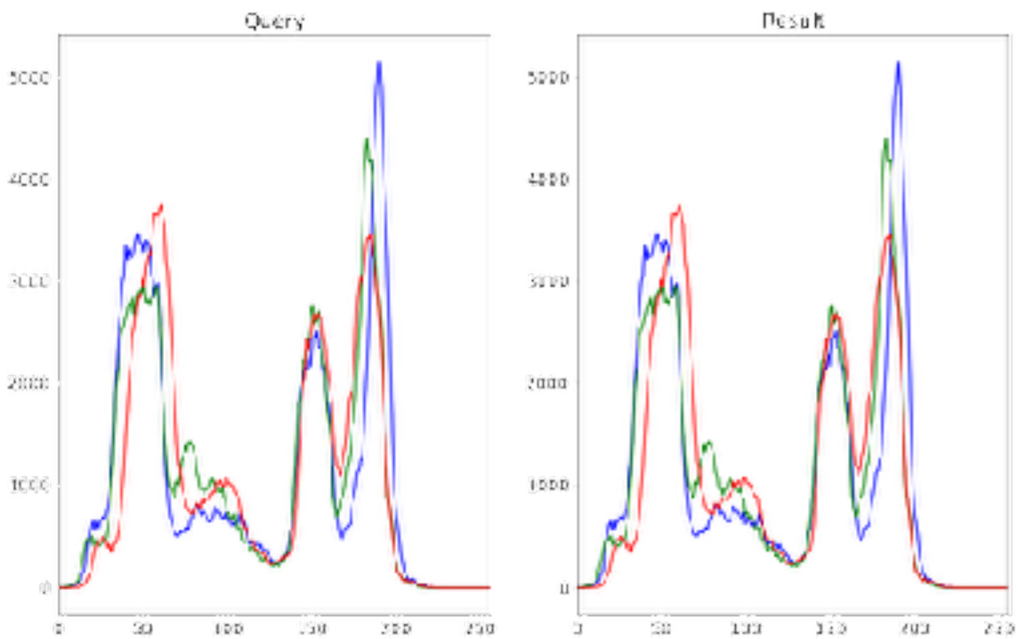
The Dataset features 12 species of the plants that help to improve production go the main crop. The training includes 4750 images and the testing has 794 images.Images are not guaranteed to be of fixed dimensions and the fish photos are taken from different angles. Images do not contain any border.

2.2 Exploratory Visualization

The Histogram represent the colour distribution of the image by plotting the frequencies of each pixel in the 3 channels of the image. In image classification histograms can be used as a feature vector with the assumption that similar images will have similar color distribution. Here we calculate the histograms for the images in the training set and find the most similar image from the histograms with the Euclidean distance metric. Results for a randomly chosen sample image is given below :



Clearly the images are similar in the labels, but they don't look similar. However their histograms are quite similar. Even though there histograms are quite similar there feature cannot be used in the training the model because we can see the histogram completely ignore the shape and texture in the images so can’t be used to train the model.



2.3. Algorithms and Techniques

Transfer Learning :

Transfer learning refers to the process of using the weights a pretrained network trained on a large dataset applied to a different dataset (either as a feature extractor or by finetuning the network). Finetuning refers to the process of training the last few or more layers of the pretrained network on the new dataset to adjust the weight. Transfer learning is very popular in practice as collecting data is often costly and training a large network is computationally expensive. Here weights from a convolutional neural network pretrained on imagenet dataset is finetuned to classify fishes.

Benchmark method :

Support Vector Machine: An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

Layers :

- **Convolution :** Convolutional layers convolve around the image to detect edges, lines. The hyperparameters to the Convolution are the number of filters, filter size, stride, padding and activation functions.
- **MaxPooling :** Pooling layers reduces the size of the images. Maxpooling is applied after the Convolution layer. MaxPooling layer helps to detect edges and lines in the images.
- **Dropout :** Dropout is a technique to prevent the network from overfitting during the training. Dropout is implemented by only keeping a certain number of neuron active in the layer. Its take value between 0 to 1, to tell what percentage of neurons to keep active.
- **Flatten :** Flattens the output of the convolution layers so that it ca be applied to the Dense layers.
- **Dense :** Dense layers are the traditional Deep neural networks. its is applied at the end ending stages of the network.Dense layer maps the output of the convolution layer to the correct prediction.

Activation functions :

Activation layers apply a nonlinearity to the Convolutional or Dense layers.

- **ReLu Activation** : ReLu or Rectified Linear Unit takes max of all the values greater than 0.

- **Softmax Activation :** It is applied to the output layer. It is use in multi class classification to convert the scores into probabilities.

Optimizer :

- **Adam :** Adam (Adaptive moment estimation) is an update to RMSProp optimizer in which the running average of both the gradients and their magnitude is used.In practice Adam is currently recommended as the default algorithm to use, and often works slightly better than RMSProp. In my experiments Adam also shows general high accuracy.

Batch Normalization : Batch Normalization initialised activations in a network to take a gaussian distribution. We use Batchnorm layers right after Dense or convolutional layers.

2.4. Benchmark

Support vector machine: Support vector machine is trained on the features extracted from the Exception model while yields me f1 score of 0.51 on the validation set.

The well designed Deep Neural Network should be able to beat the Support vector Machine baseline score. So the f1 score of 0.51 is baseline for the model. This is reasonable score for beating the baseline model score.

3. Methodology

3.1 Data Preprocessing

As I am using Xception Net for transfer learning images are preprocessed as performed in the Xception model. In Xception model the mean is subtracted from each channel so the values for each channel had a mean of 0. In preprocessing the software expects the input in the format (B,G,R) order but python by default gives the output as (R,G,B) so the image has to be converted from RGB to BGR format. In this dataset images has different sizes and resolution so the input image is resized to 299 * 299 * 3 to reduce size. As the dataset some has different number of samples of each class highest 654 and lowest 221. I am using lowest 221 samples of each class to train the model. The dataset provided by kaggle does not contain Validation dataset, so i am splitting dataset into training set and validation set.The validation has randomly selected 266 samples.

Distribution of Species of Plants

Labels	Count
Black-grass	263
Charlock	390
Cleavers	287
Common Chickweed	611
Common wheat	221
Fat Hen	475
Loose Silky-bent	654
Maize	221
Scentless Mayweed	516
Shepherds Purse	231
Small-flowered Cranesbill	496
Sugar beet	385

Selected Number of samples of each species.

Labels	Count
Black-grass	221
Charlock	221
Cleavers	221
Common Chickweed	221
Common wheat	221
Fat Hen	221
Loose Silky-bent	221
Maize	221
Scentless Mayweed	221
Shepherds Purse	221
Small-flowered Cranesbill	221
Sugar beet	221

3.2 Implementation

During the Images extraction from the zip file the preprocessing is done on the Images after the extraction then preprocessed dataset is split into training and validation set. The features extration is done from the Xception model. After the features extraction from the Xception model the small fully connected dense layer along with dropout layer is applied to predict the final output of the model.

3.3 Refinement

In the initial state i applied my own covet model with 3 convolution layer after i trained it for couple of hours the model was not giving satisfactory result. 18% accuracy on the validation set. So i created RESNET-50 model and train the model but still was not giving me satisfactory result even after running model to 4 hours then i decided to go for Transfer learning as the plant leaves are very small in the image the networks will not learn on the initially started Xception model this will more the 8-10 hours on GPU. as i don't have GPU at this point i shift to pretrained Xception model and extorted features from the Xception model to get features from the model.

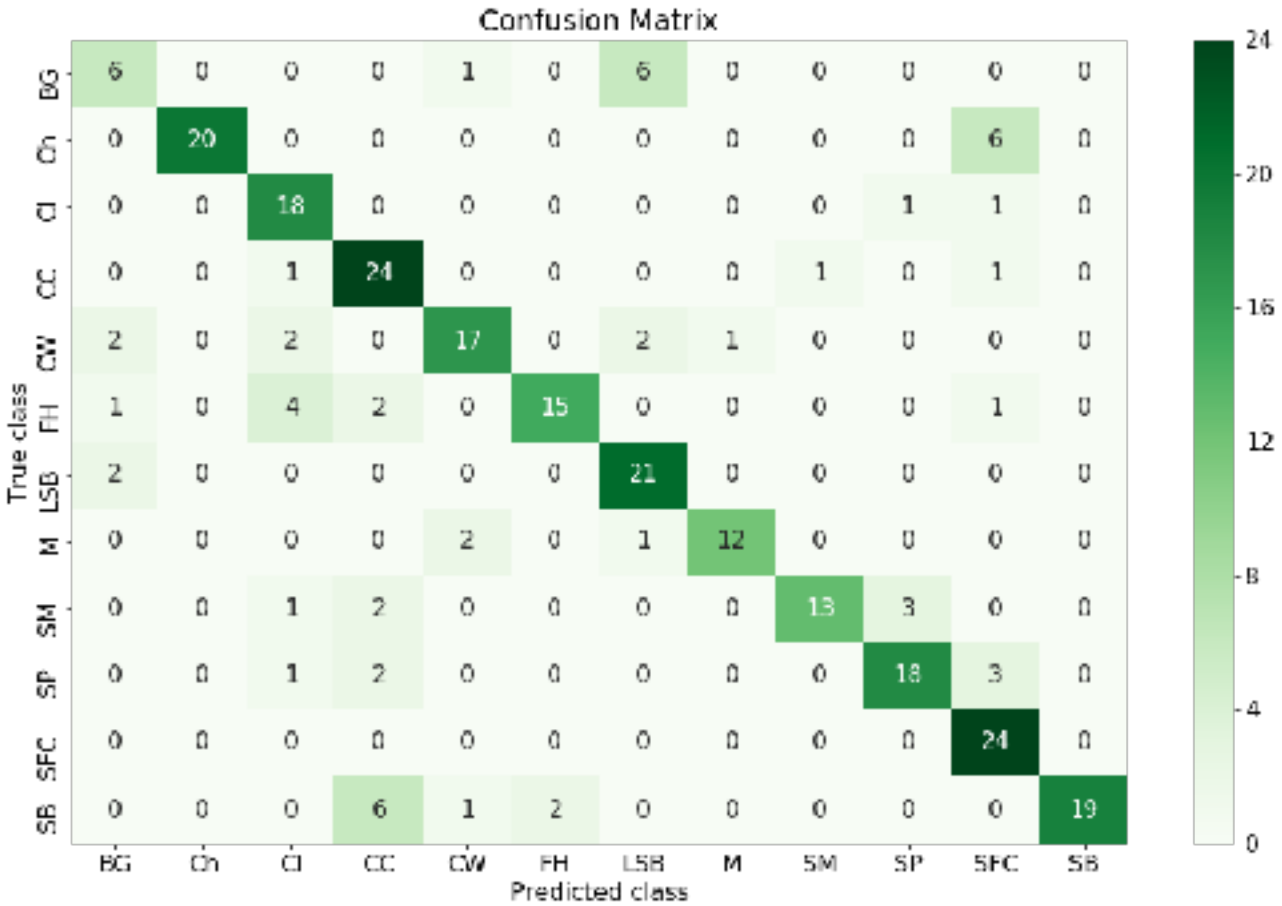
4. Results

4.1 Model Evaluation

The model with transfer learning along with deep neutral network perform very well.To use Xception model for the transfer learning I’ve used the pretrained weights of the Xception trained on imagenet problem.

I have preprocessed all the images according to the Xception architecture. The shape of the images was resized to (299,299) as Xception model was trained on images with shape (299,299). Dense layer on the top of Xception layer is used to predict the validation set of the images.

For evaluation f1 score is used to determine the performance of the model. On feature extracted from the Xception model the SVM is applied as a benchmark which yield a score of 0.51 on validation set and resnet-50 with. The fully connected layer on the same features get a score of 0.84 on the validation set. I have also predicted the confusion matrix to get better understanding of how the model has predicted species of the plants.



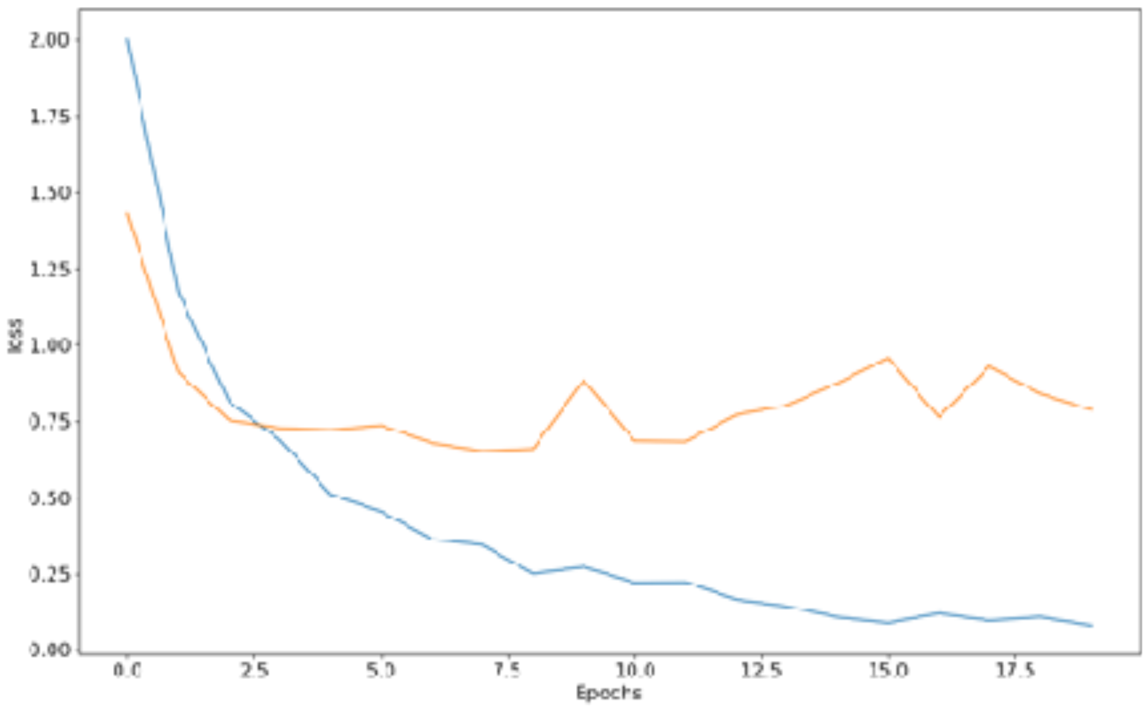
as seen from the confusion matrix of the validation set the model is really good at predicting species, In each class most of the samples are correctly predicted. No other class is dominated the prediction of the species as I use same number of samples of each class for training.

5. Conclusion:

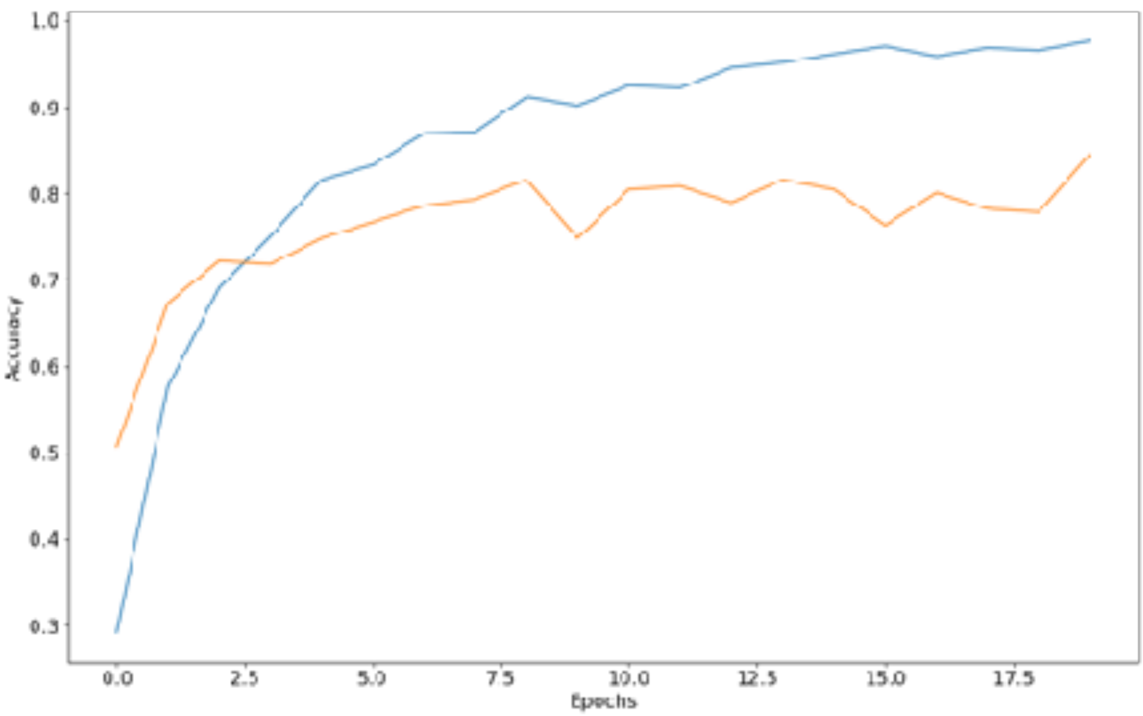
5.1. FreeForm Visualization

I have recorded the accuracy and loss of the model for each epoch, the graphs are shown below. where blue line is of training set and orange line is of validation set.

LOSS



Accuracy



As we can see the training accuracy is nearly 100% and loss is nearly zero.
the validation accuracy is nearly 80% and validation loss is nearly 0.75 near the end of 20th epoch. We also see the trend here where validation loss keeps on decreasing at 8th epoch and the training loss keeps on decreasing/ accuracy keeps increasing , while the validation loss keeps on increasing instead of decreasing. This model is not overfitting on the training set.

5.2 Reflection

For reaching to the end i have tried and use various complex model to classify the image the Support Vector machine was use as baseline to the challenge. I've tried basic convolution with 3 layer to see get a baseline as it is good practise. But perform very low (Code is not present in the Notebook) As the model got me the accuracy of 0.14 even after training for hours. So then i decided to go for transfer learning as a convolution layer from the scratch is not going to able to learn. So i used Xception model to extract features from the model and use the feautres for training the model on the deep neural network.

The most difficult part for me was to get the experiments running on the kaggle. Higher computational time results in lower number of experiments when it comes to neural networks, specially when I'm just figuring out what to do as it's my first experience with deep learning. However now I feel more confident in my skills in using deep learning.

5.3 Improvement

Due to time and computational cost it was not possible for do more experiments with different architecture of the model and not able to use different other architecture of pertained model such as RESNET, VGG-19,etc. it is possible that other model can be more effective. By doing various image processing such that model can pattern easily can improve the accuracy of the classifier.

As the classes where imbalanced by generating augmentation dataset for the class that has number of samples than the other , save them and generate 700 images of each class , this model could have been more robust
i didn't do it because as the size of the image vary from 50*50 pixel to 3000*3000 pixel i use 300*300 pixel so that there should be minimal loss of data, by creating more augmented images my RAM would not be able to handle such large data i did't create more augmented data.

Refrences:

- http://cs231n.github.io/transferlearning/ "CS231, Andrej Karpathy's overview on Transfer Learning"