Plant Seedlings Classification:
Convolutional Neural Network vs Pre-trained ResNet, VGG-16 and Ensemble PyTorch

A Project report for the assignment submitted in fulfillment of the requirements for the COMP SCI 7209 Big Data Analysis and Project

MASTER OF DATA SCIENCE

by

Arpit Garg

Register No: A1784072

Semester: III, Year: 2

THE UNIVERSITY of ADELAIDE

SCHOOL OF COMPUTER SCIENCE

THE UNIVERSITY OF ADELAIDE

ADELAIDE - 5000

OCTOBER 2020

Abstract

In this project, we are using deep learning convolution neural network algorithms to classify plant seedlings. The dataset is available on Kaggle. In the dataset, we have different 12 classes of seedlings under train dataset. In particular, we will focus on the parameters and preprocess the parameters before training our model. In modeling, we will use different image processing algorithms such as Convolutional Neural Network, ResNet ,VGG-16 and Ensemble approach. Initially, there are unwanted background noise in the images, we have used various image processing techniques like: segmentation, masking and sharpening to handle this situation . We are using visualisation at different places for comparison. This report focuses on solving the assignment and Kaggle question with detailed analysis and in-depth context related to Deep Learning Image Recognition techniques. Using the approach, we secured a place in the top 10% at the time of submission.

Keywords : Deep Learning, Image Recognition, Convolution, ResNet, Masking, Segmentation

1 Introduction

Agriculture and crops are one of human life 's needs, and demand increases every day. The need of sustainable agriculture sector is crucial requirement of population-increasing world. It requires a step to lower the labour cost while increasing the efficiency. Plant seedling classification is the primary step towards this process. It will help to classify the required seedlings between the weeds and encouraging nutrient, pesticide or medicinal items to be added in timely manner. The main of this project is to differentiate weeds from the crop seedling. For achieving our purpose, we are using the dataset provided on Kaggle for training and testing purposes The Aarhus Signal Processing organization has successfully published a collection of data containing photographs of nearly 960 identical plants of 12 species in multiple growth phases, along with the University of Southern Denmark. Signal Processing Group. [1]
For training purposes, we have 12 different classes. For testing purposes, we have 794 images with the same features excluding classes because that is what our model will classify. The primary purpose is to develop a model that can classify the pants seedlings based on these parameters accurately.

2 Methodology

2.1 Data Pre-processing

In this dataset, we have different images and our target is classification of weed and plant seedling from 12 given categories.We have plotted some images from each category to decide pre-processing step.

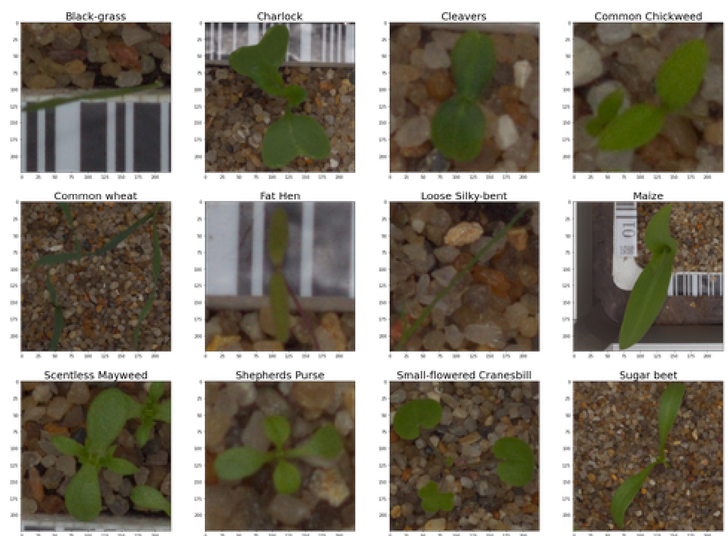


Figure 1: Images from Each Category

From the above plotted Figure 1 we can easily predict that there is lot of background noise in each image which needs to be removed before modelling the data. We only need to focus on leaves, so to remove background, we will use different digital image processing techniques. We have our dataset in two groups training dataset and testing dataset and we will apply pre-processing techniques on both the groups.

2.2 Digital Image Processing

Digital Image Processing means manipulation of images through different algorithms In this case, we need to remove background image and focus on leaves.

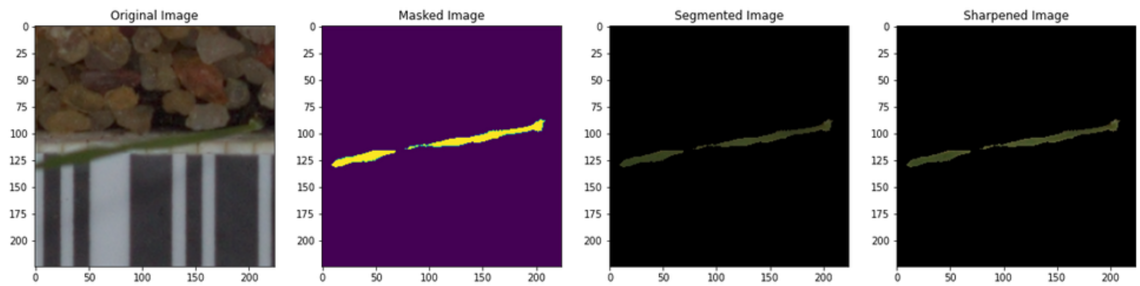


Figure 2: Digital Image Processing

2.2.1 Image Masking

The masking principle is also called spatial filtration. We are concerned only with the filtering operation carried out directly on the image with this definition. The basic method of filtering and application of masks is to transfer the filter mask in an image from point to point.The filter response is determined by a pre-defined relationship at each point (x , y) of the original image. The predefined values of all filters are a norm. In Figure 2, we can observe the masked image of category Black-grass.

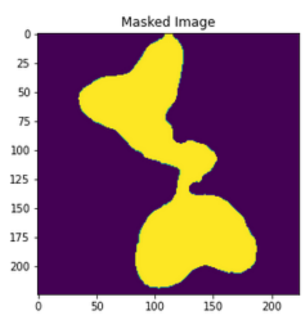


Figure 3: Masked Image of Charlock

2.2.2 Image Segmentation

Image segments are a widely used method in digital photography and analysis, which is often based on the properties of the image pixels to partition an image into different sections or areas. Image segmentation can be based on coloured or shaped resemblance, distinguishing the foreground from the background or clusters pixel areas. In Figure 4, we can observe the segmented image of category Cleavers.

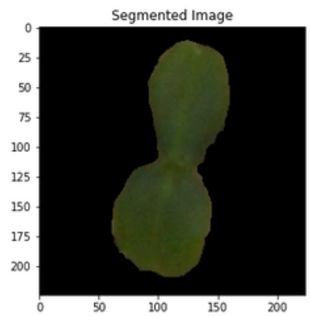


Figure 4: Segmented Image of Cleavers

2.2.3 Image Sharpening

Image sharpening refers to a technology for improving the boundaries and fine details in a photo. Picture sharpening is commonly employed for increasing local contrast and sharpening photos in printing and photographs sectors. In general, sharpening of the image consists of inserting a signal equal to the original image in a high-pass filtered version.

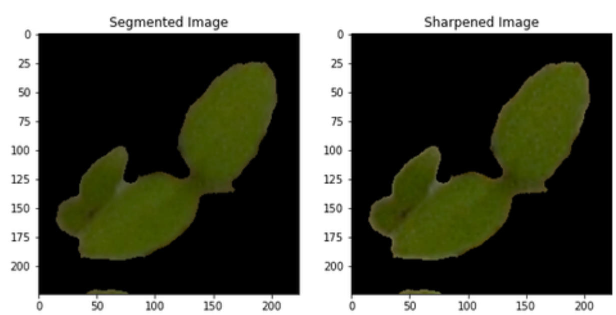


Figure 5: Sharpen Image of Chickweed

Figure 5 shows the segmented and sharpen image of chickweed. We can easily observe the difference between two images.

2.3 Convolution Neural Network (CNN)

Convolutions neural networks (CNN or ConvNets) is one of the major classifications of objects. The identification of objects, reconnaissance faces etc. are some of the fields of frequent use for CNNs. CNNs have two major ele-

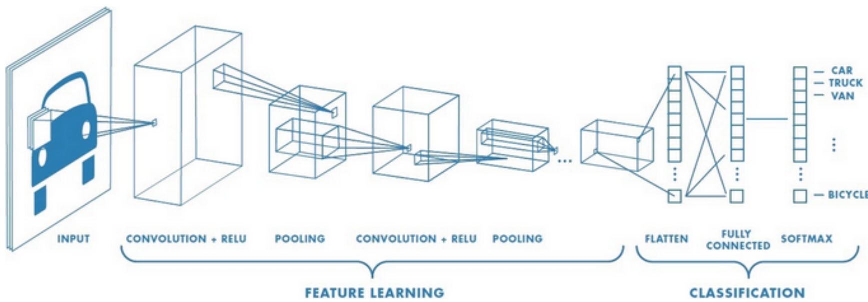


Figure 6: ConvolutionNet Architecture
[2]

ments. 1. Learning features: Transforming, ReLU, Layer pooling stages. In this function learning phase edges, colours, lines and curves are extracted. 2. Classification: In this process, you can find the Fully Connected(FC) layer. They assign a chance to what the algorithm says for the object in the image.

2.3.1 Convolution Layer

The first layer to derived characteristics from an image is Convolution. It retains the link between pixels by using tiny squares of input data to learn image features. Two inputs, for example image matrix and a filter, are necessary mathematically. Diverse operations like edge recognition, blurring and sharpening exist.

Convolution Formula:

$$(f * g)(t) \stackrel{def}{=} \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

2.3.2 Activation Layer

The aim of the Activation function is to minimise ConvNet's non-linearity. It is also important to squash the weighted linear amount from the neurons. ReLU, which is the most commonly used activation function for the rectified linear unit

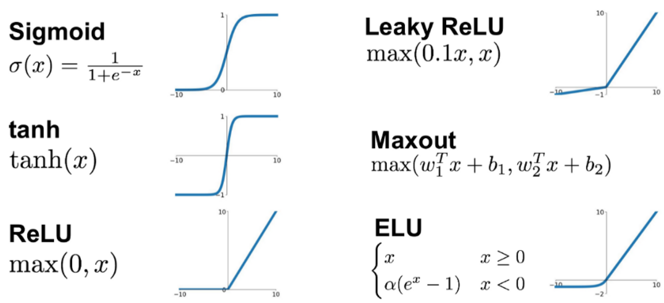


Figure 7: Activation Functions
[4]

2.3.3 Pooling Layer

This helps to minimise the number of parameters in very large images. Spatial pooling, also known as downsampling, decreases the dimensionality of the data.

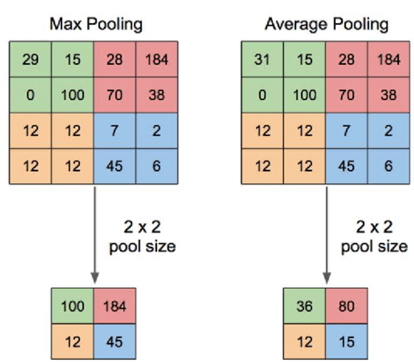


Figure 8: Pooling Types
[4]

2.3.4 Fully Connected Network

Per neuron of this layer is attached to each neuron of the next layer of the previous layer. The conventional multi-layer Perceptron uses a SoftMax feature in the output layer.

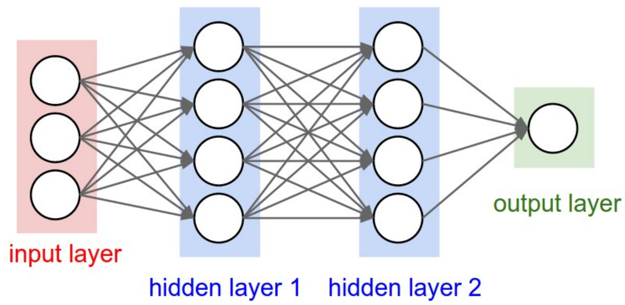


Figure 9: Fully Connected Layer
[4]

2.3.5 Kernel

A kernel is a weight matrix which is randomised to and fitted around the image. A kernel derives from the image a specific feature. In Figure 10, kernel is of size 3 X 3.

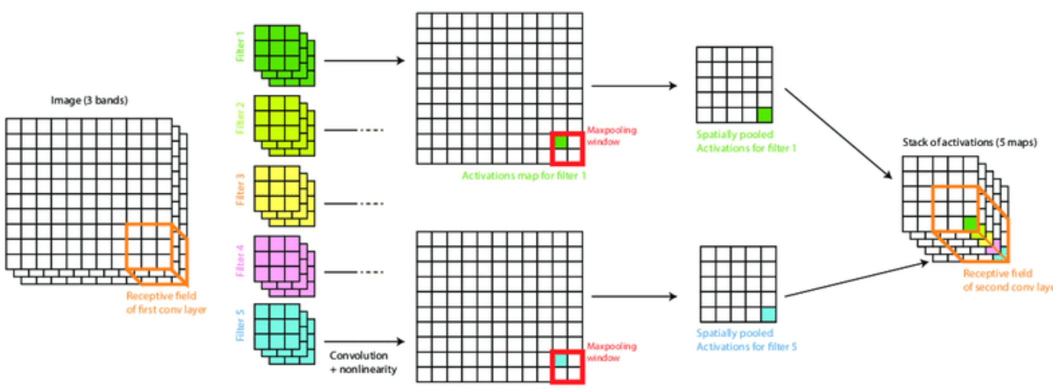


Figure 10: Pooling Types
[5]

2.4 ResNet

If identity mapping is ideal, we can force residues to 0 (F(x)=0) quickly, rather than a stack of non-linear layers to match an identity mapping (x, input = output). In simple language, using a stack of non-linear CNN layers as feature, it is very straightforward to come up with a solution such as F(x) = 0 instead of F(x) = X. So, this function F(x) is called the Residual function. [7] Each ResNet block is either 2 layer deep (Used in small networks like ResNet 18, 34) or 3 layer deep(ResNet 50, 101, 152). ResNet-34 obtained a 5.71% better validation error than BN-inception and VGG. The validation error of resnet-152 is 4.49 percent at the top-5. The Top 5 validity error of 3.57% is obtained with a range of 6 models of varying depths. ResNet has two building blocks:

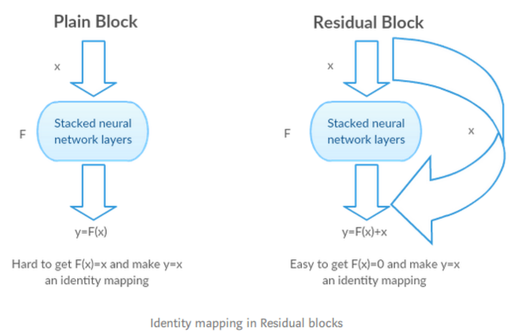


Figure 11: ResNet Architecture
[3]

2.4.1 Identity Block

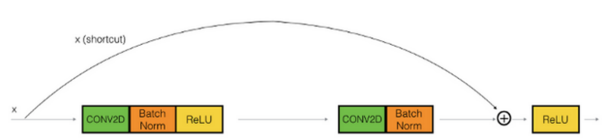


Figure 12: Identity Block
[6]

2.4.2 CNN Block

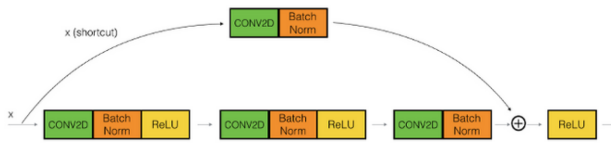


Figure 13: CNN Block
[6]

2.4.3 ResNet Comparison

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Figure 14: ResNet Comparison
[3]

2.5 VGG-16

VGG16 is a CNN architecture used in 2014 for winning the ILSVR(Imagenet) competition. It has until now been considered one of the most impressive model architectures. The most unusual aspect of VGG16 is that they insisted on making 3x3 philtre layers, with a phase 1, and still used the same 2x2 string filter padding and max-pool layer. The entire architecture is constantly assisted by this arrangement of convolutions and max pool layers. At last, the softmax for the production is 2 FC (fully linked layers). The sixteenth in VGG16 corresponds to a weight of 16 layers. This network is a wide network with roughly 138 million (roughly) parameters.

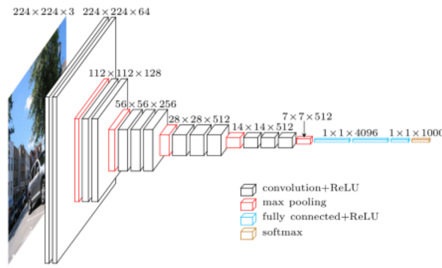


Figure 15: VGG Architecture
[8]

3 Implementation

Implementation of this project can be divided into four parts:

3.1 Data Loading

We have used PyTorch for loading our dataset. PyTorch provides the functionality of ImageFolder for loading all the images within subfolder and automatically assign them ids and classes based on their subfolders with the assigned transformation applied on it. Due to the resources limitation we are only providing resize transformation. For test images we do not have any subfolders as they are our targets, so we need to create separate function (Here SeedlingTestDataset) to read all the images from the folder.

3.2 Exploratory Data Analysis (EDA)

Every project should start with the Exploratory Data Analysis (EDA) to gain insights from the data. It helps to examine the structure of dataset like patterns. For this purpose, we have used different python libraries like NumPy, pandas, matplotlib. We have printed the data shape, length and classes of dataset. We have printed the histogram with count of the data.

We have also displayed all the 12 categories and assigned the numerical id's to each class and displayed the results. For all these puproses we are using other libraries with integration of PyTorch library.

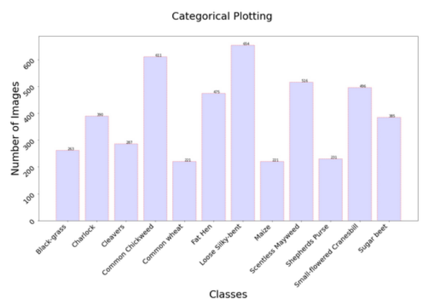


Figure 16: Histogram of Each Category

```
{0: 'Black-grass',
 1: 'Charlock',
 2: 'Cleavers',
 3: 'Common Chickweed',
 4: 'Common wheat',
 5: 'Fat Hen',
 6: 'Loose Silky-bent',
 7: 'Maize',
 8: 'Scentless Mayweed',
 9: 'Shepherds Purse',
 10: 'Small-flowered Cranesbill',
 11: 'Sugar beet'}
```

Figure 17: Classes with Ids

3.3 Data Pre-Processing

Pre-processing parts is very crucial to handle noisy data in the dataset. In this dataset, we have 12 categories, and we need to pre-process data such as removing background noise before modelling. Hence, we have created two arrays, one is data which will contain all the images and other is label which holds the respective label of images. After that we have created dictionaries to hold both these arrays in single variable. After these pre-processing we have displayed some images from each category (Figure 1).
From these images, we can observe that there's a lot of background noises that's need to be removed using Digital Image Processing Techniques mentioned in the Section 2.2. In this section we have applied different image processing techniques like we converted all the images to HSV format then masking is applied followed by segmentation and sharpening. We have applied these techniques on both training and testing dataset to maintain consistency. We then have plotted the images after each image processing technique from each category to show the difference between processes (Figure 2). We are using PyTorch library that means we are dealing with tensors. So we

have converted all out dataset to Tensors and we have divided our training dataset to training dataset and validation dataset using dataloaders. We have used the standard ration of 90:10 for this purpose.

3.4 Data Modeling

In data modeling, we applied simple Convolutional Neural Network (CNN) and ResNet for plant seedling classification. We have started with CNN and creates a very basic model of CNN as explained in Section 2.3. We have trained our model using simple CNN model with early stopping and saved our weights, so we can load the weights directly and get the accuracy on validation dataset. We have calculated and plotted loss and accuracy on validation dataset.

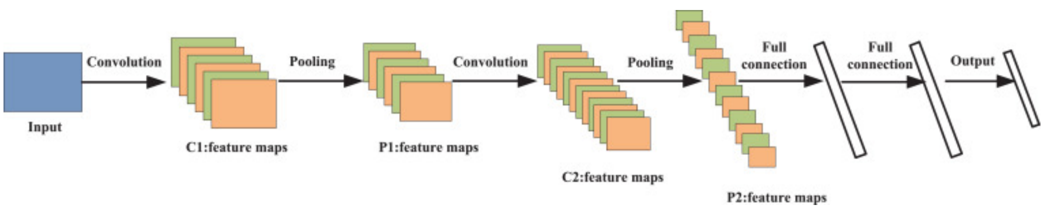


Figure 18: CNN Model

Moreover, we moved to much complex model called ResNet34 and VGG-16. We use the pretrained model ResNet34 and VGG-16 in PyTorch due to resources limitation.(Section 2.4). We have submitted both the csv generated. Our final submission on Kaggle is from ResNet34 due to its high accuracy.

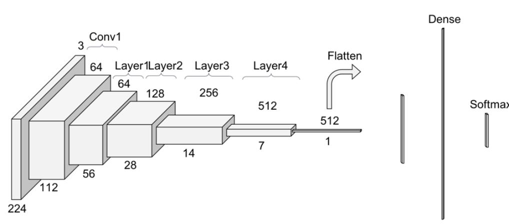


Figure 19: ResNet Model

Note: For generating visuals of models PyTorch torchviz and wandb subscription services are used. (Both are not displayed in final code submission as they require `wandb.ai` WeightsAndBiases subscription and integration with PyTorch through API.)

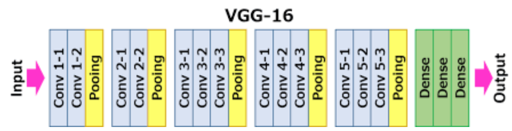


Figure 20: VGG-16 Model

4 Evaluation

For the evaluation purposes, we are mainly focusing on F-Score, and we need high score for each model.The F1 standard formula is the harmonic mean with precision and recall. The F-score for a perfect model is 1.

$$F_1 = 2\frac{Precisión \times Recall}{Precisión + Recall}$$

4.1 Learning Curves

Furthermore, we have also plotted train accuracy, validation accuracy, training loss and validation loss for all Convolutional Neural Network VGG-16 and ResNet34.

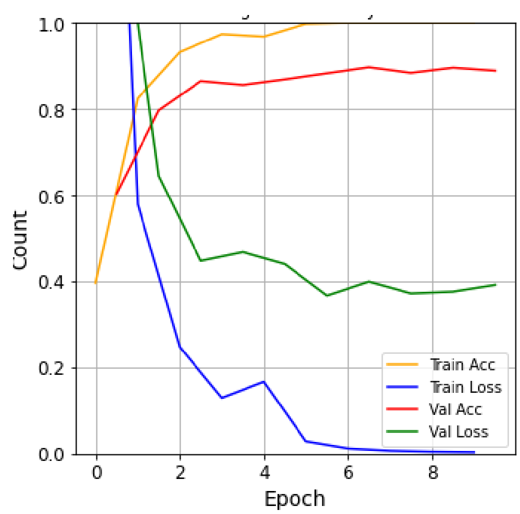


Figure 21: CNN Learning Curve

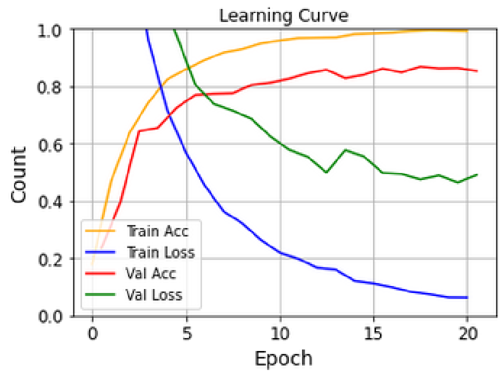


Figure 22: VGG-16 Learning Curve

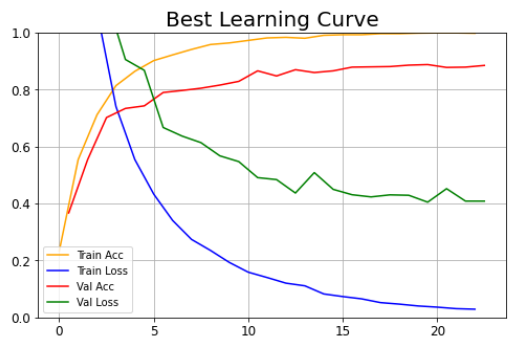


Figure 23: ResNet34 Learning Curve

4.2 Confusion Matrix and ROC Curve

We have also plotted the confusion matrix and Roc Curve of ResNet34 using `wandb.ai` WeightsAndBiases visualisation tool.

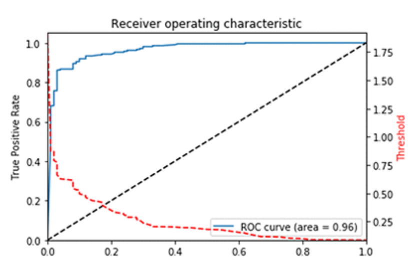


Figure 24: ROC Curve

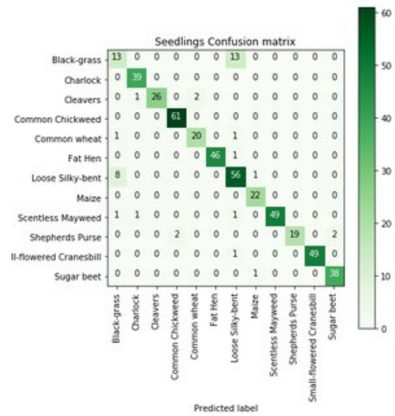
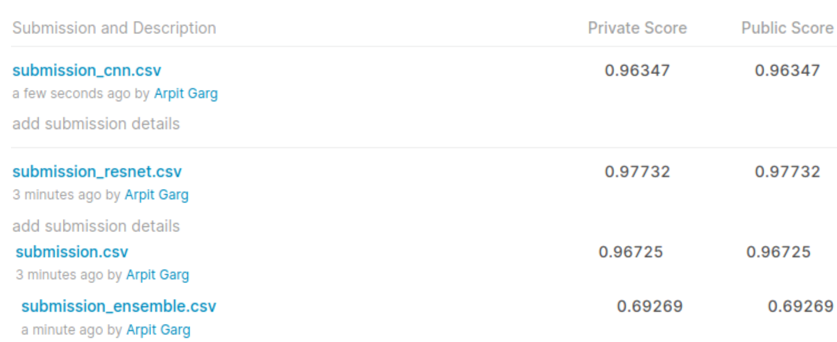


Figure 25: Confusion Matrix

4.3 Evaluation Metrics

We have stored and displayed F-Score for Simple CNN, VGG-16 and ResNet. All to the standards, we have divided the 90% for the training purposes and remaining for cross-validation. For the testing purposes, we have predicted the outputs and uploaded them on Kaggle and got 0.96347 F-Score for CNN,0,96725 for VGG-16 and secured the position in top 20% solutions provided when submitted and got 0.97732 F-Score for ResNet34 and secured place in top 10% at the time of submission. We are getting worst results for ensemble method. FOr evaluation purposes we have also generated output using mode ensemble method.



Submission and Description	Private Score	Public Score
submission_cnn.csv a few seconds ago by Arpit Garg add submission details	0.96347	0.96347
submission_resnet.csv 3 minutes ago by Arpit Garg add submission details	0.97732	0.97732
submission.csv 3 minutes ago by Arpit Garg	0.96725	0.96725
submission_ensemble.csv a minute ago by Arpit Garg	0.69269	0.69269

Figure 26: Kaggle Submission(Scores)

Table 1: Evaluation Metrics

S.No.	Model	F Score
1	Convolutional Neural Network	0.96347
2	VGG-16	0.96725
3	ResNet34	0.97732
4	Ensemble	0.69269

From the above Evaluation section 4, we can observe that we are getting the best results in ResNet34. We are getting good results for Convolution Neural Network and VGG-16 and worst for Ensemble mtehod. We can still improve the results using more complex data augmentation techniques that requires more resources and computation power. We have followed the mentioned approach because of limited resources available.

References

[1] Thomas Mosgaard Giselsson, Rasmus Nyholm Jørgensen, Peter Kryger Jensen, Mads Dyrmann, and Henrik Skov Midtiby. A public image database for benchmark of plant seedling classification algorithms. *arXiv preprint arXiv:1711.05458*, 2017.

[2] Purnasai Gudikandula. A beginner intro to convolutional neural networks.

[3] Prakash Jay. Understanding and implementing architectures of resnet and resnext for state-of-the-art image classification: From microsoft to facebook.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[5] Diego Marcos, Michele Volpi, Benjamin Kellenberger, and Devis Tuia. Land cover mapping at very high resolution with rotation equivariant cnns: Towards small yet accurate models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 02 2018.

[6] mx. Convolutional neural networks week2. deep convolutional models: case studies.

[7] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.

[8] Rohit Thakur. Convolutional neural networks week2. deep convolutional models: case studiesstep by step vgg16 implementation in keras for beginners.