Deep Learning Theory Project

Sarthak Vasan(B19BB040) Shivam Zade(B19BB043) Harsh Kumar(B19CSE036)

Plant disease detection

INTRODUCTION

Nowadays there are numerous cases seen of plant diseases affecting the crop production. This is leading more towards food insecurity. To avoid this, early detection of these diseases and their prevention and control is important for management of crop production. Generally, there is a visible pattern present on the diseased crop which can be considered as an abnormality. Leaves are common examples of primary sources for identifying plant diseases.Its a quite famous classification problem so what we did was coded different deep learning architectures to solve this problem and in the end we compared them on different evaluation metrics like accuracy, F1 score, precision and recall to know which of them performed the best.

Resnet50 + SVM

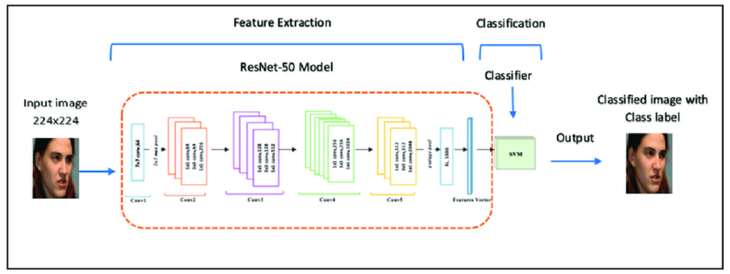

Fig. 1: Resnet50 + SVM architecture

Batch size equal to 8 was used and linear kernel was used. Accuracy obtained on the test set was 96.34%. Firstly the images are passed through the pretraiend Resnet50 and then the output of the last layer was sent to the multiclass SVM classifier and then the data was fitted to the corresponding labels. Finally the results were obtained and F1score, precision and recall was calculated from the predicted and true values. Below is the screenshot of the table containing all the scores.

	precision	recall	f1-score	support
class0	0.96	0.97	0.97	279
class1	0.96	0.96	0.96	240
accuracy			0.96	519
macro avg	0.96	0.96	0.96	519
weighted avg	0.96	0.96	0.96	519

Fig. 2: Scores

The accuracy obtained was 96.34 %

Datasets and Architectures Used

We took the datasets from a github repo : Datasets. We firstly split the dataset into train and test and then created a dataloader to preprocess the images. We used 3 different architectures that are: - pretrained Resnet50 + SVM classifier. In this we used the features extracted from the last fully-connected layer and trained a multiclass SVM classifier for training. Next we finetuned a Resnet50 model training the parameters of last two layers. Thidly we used a simple CNN model with 3 convolutional and 3 pooling layers and with SGD optimizer for the training process. Also the images were rescaled to 224 X 224 size.

Finetuned Resnet50

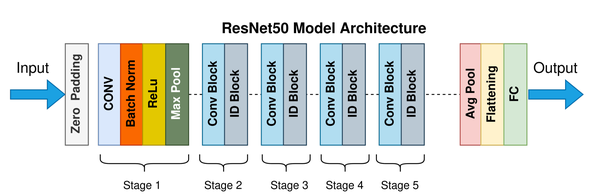

Fig. 3: Resnet50 architecture

Batch size equal to 8 was used and number of epochs were 10. SGD optimizer with crossentropy loss was used. Accuracy obtained on the test set was 95%. The features of last two layers were updated and trained to extract the useful features and then finally the results were obtained and F1score, precision and recall was calculated from the predicted and true values. Below is the screenshot of the table containing all the scores.

	precision	recall	f1-score	support
class0	0.93	0.98	0.95	240
class1	0.98	0.94	0.96	279
accuracy			0.96	519
macro avg	0.96	0.96	0.96	519
weighted avg	0.96	0.96	0.96	519

Fig. 4: Scores

In the end the train loss vs number of epochs graph is attached.

Results

After comapring all the evaluation metrices we can clearly say that Resnet50 with SVM classifier performed the best closely followed by finetuned Resnet50 and then followed by the CNN model.

References

https://github.com/mehra-deepak/Plant-Disease-Detection/tree/master/image_classification/dataset/train
https://pytorch.org/vision/main/feature_extraction.html
https://ieeexplore.ieee.org/document/9399342

CNN

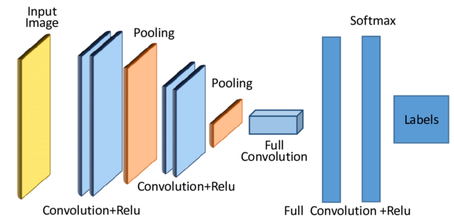

Fig. 6: CNN architecture

Batch size equal to 8 was used and number of epochs were 10. SGD optimizer with crossentropy loss was used and a learning rate of 0.01 was used. Accuracy obtained on the test set was 93%. The architecture contains 3 convolutional layers and 3 pooling layers to extract the features and then finally the results were obtained and F1score, precision and recall was calculated from the predicted and true values. Below is the screenshot of the table containing all the scores.

	precision	recall	f1-score	support
class0	0.91	0.96	0.93	240
class1	0.96	0.92	0.94	279
accuracy			0.94	519
macro avg	0.94	0.94	0.94	519
weighted avg	0.94	0.94	0.94	519

Fig. 7: Scores

In the end the train loss vs number of epochs graph is attached.