# Onion Crop Disease Detection and Classification

1st Siddharth Khandelwal
*Electrical Engineering*
*Indian Institute of Technology Bombay*
Mumbai, India
190070062@iitb.ac.in

***Abstract*—Over the past decade, deep learning has emerged as a promising approach for automated crop disease detection and classification. This study aims to investigate the performance of different CNN models for onion crop disease classification and detection and to identify the most accurate model. We develop a CNN-based neural network to detect diseases in onion crops and study various baseline models and their performance. We discuss the challenges associated with crop disease detection, such as imbalanced datasets and limited annotated data, and explore techniques to address these challenges, such as data augmentation, transfer learning, and a different sampling technique. We use transfer learning on the Plant Village dataset and then train on our small onion dataset to detect onion diseases. We achieve an accuracy of 92% using the DenseNet baseline and an imbalanced dataset sampler. We also show the performance of our model on classifying onion diseases. The study highlights the potential of CNN models for automated crop disease detection and could contribute to developing precision agriculture technologies.**

***Index Terms*—onion disease detection, CNN, Plant Village**

## I. Introduction

Plants serve as the source of about 80 % of the food consumed by humans and contribute to 98% of the oxygen we breathe. Pests and diseases significantly threaten them, and these have nothing but increased over the past few years. According to the Food and Agriculture Organization of the United Nations (FAO), crop diseases are responsible for losing up to 40% of global food production annually. Each year, plant diseases cost the global economy over $220 billion.

Many of these diseases can spread quickly and over a large area, leading to reduced yields, lower quality produce, or even total failure of crops. In addition to affecting plant health, human health can also be affected due to the introduction of toxic compounds into the food cycle. To conclude, there is a need for continuous research and development of disease-resistant crop varieties and sustainable farming practices to ensure global food security.

Deep learning is a sub-field of machine learning which uses neural networks to learn complex mapping functions from a particular type of input to an output. It is a rapidly evolving field that has significantly advanced in various domains, including computer vision, speech recognition, and natural language processing. Over the years, deep learning methods have found their way into the agriculture industry for crop disease detection and classification, yield prediction, pest counting, and monitoring and controlling soil, water, and temperature conditions. Convolutional neural networks (CNNs) are one class of deep learning networks mainly used for visual imagery. Researchers have used deep learning algorithms to analyze large datasets of images of crops, such as leaves or fruits, to detect the presence and severity of diseases with high accuracy. For instance, several studies have demonstrated the effectiveness of deep learning in detecting diseases in crops such as tomatoes, grapes, and apples [1]–[3]. Adopting deep learning technology in agriculture can improve crop yields, reduce pesticide use, and contribute to global food security.

In this work, we work with the Plant Village dataset and an onion crop dataset for disease detection and classification using transfer learning. We also explore handling imbalanced datasets using a dataset sampler and data augmentations. In the next section, we describe the datasets used in our study.

## II. Datasets

### A. Plant Village

The Plant Village Dataset [4] is a publicly available dataset consisting of 39 different classes of plant leaf and background images. Of 39 classes, 38 belong to different plant leaf images that might be diseased or healthy, and 1 class is images of outdoor backgrounds. The dataset comprises approximately 54000 images, each of 256 x 256 pixels. We show some sample images from the plant village dataset in Figure 1.

### B. Onion Dataset

The onion crop images consist of on-field images of the onion crop taken by a mobile phone or a DSLR camera. It consists of 8 classes, out of which seven classes are different diseases affecting the onion crop like Anthracnose, Basal Rot, Damping off, purple blotch, stemphylium blight, iris yellow spot, bulb rot, and 1 class contain images of healthy onion crop. Some sample images are shown in Figure 2. Table I shows the distribution of the number of images in each class. The images are of different resolutions and consist of images taken from afar and closeup images of the crop.

## III. Model Architecture

### A. Binary Classification

We perform binary classification of the onion images into healthy and diseased images. We clump together the seven diseased classes to form 1 diseased class of images. Figure 3 shows the model architecture implemented. We use transfer learning on several different pre-trained models using the Plant Village and Onion datasets to obtain the final classification results. We use the baseline models which have been trained

(a) Apple Black Rot

(b) Corn Northern Leaf Blight

(c) Strawberry Leaf Scorch

(d) Tomato Leaf Mold

Fig. 1: Sample Plant Village Dataset Images

(a) Stemphylium Blight

(b) Anthracnose

(c) Purple Blotch Disease

(d) Healthy

Fig. 2: Sample Onion Dataset Images

on the ImageNet dataset. We remove the final output layer of these baseline models and add a Linear Block in place of it. The Linear block consists of 2 layers of nodes, the first being 512 nodes, and the second consists of 39 output nodes corresponding to each class in the Plant Village Dataset. Now we use the baseline models as feature extractors for training the Linear Block or fine-tuning the baseline model weights

TABLE I: Number of Images in Class

| Class Name | Number of Images |
|---|---|
| Anthracnose Disease | 20 |
| Basal Rot | 111 |
| Bulb Rot Seed | 304 |
| Damping Off Disease | 187 |
| Healthy | 101 |
| Iris Yellow Spot Virus | 5 |
| Purple Blotch Disease | 9 |
| Stemphylium Blight Disease | 48 |

along with the Linear block parameters. After the training on the Plant Village is completed, we change the last output layer with two output nodes for the disease detection task on the onion images.

### B. Multi-class Classification

Instead of a two-node output layer for this classification, we use a four-node output layer corresponding to classes with many images: Basal Rot, Bulb Rot Seed, Damping Off, and Healthy. We train this model similarly as we do in the case of Binary Classification.

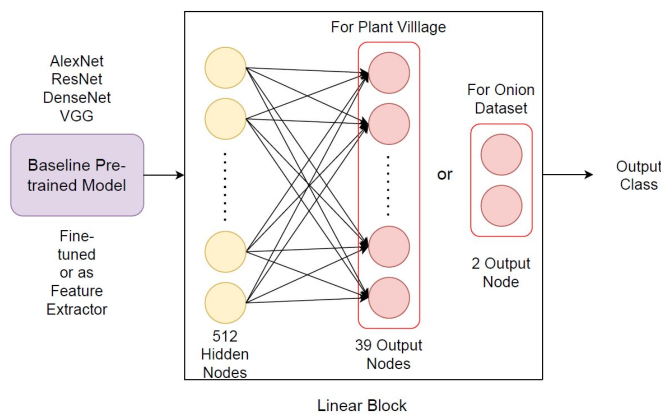



Fig. 3: Model Architecture

## IV. Pre-Processing

We have two pre-processing pipelines for each of the training and validation dataset

### A. Training

We first rotate the image randomly between -30°to 30°. Then a random resized crop is done, where a random image area is cropped and resized to a shape of 224x224. Then we randomly flip the image horizontally with a probability of 0.5. Finally, we normalize the image with the mean and standard deviation of the imagenet dataset.

### B. Validation

In this pre-processing pipeline, we first resize the image into a 256x256 image and then take a center crop of 224x224 out of it. Finally, we again normalize the image with the mean and standard deviation of the ImageNet dataset.

## V. Experiments & Results

### A. Plant Village Pre-training

We experiment with feature extraction and fine-tuning the baseline models with the Plant Village dataset. For fine-tuning, we update the weights of the baseline models every 20 batches, and while feature extracting, we do not update these weights at all and keep the original weights from the ImageNet pre-training. We use a batch size of 32 images. The loss function used is Negative Log Likelihood loss since the model's output is a log softmax over the output classes. We use an Adam optimizer during training with a learning rate of 0.01 for feature extraction and 0.0001 for fine-tuning. We also use a learning rate scheduler which reduces the learning rate by a factor of 0.1 after every five epochs. The results of these are shown in Table II

TABLE II: Plant Village Accuracy

| **Model** | **Accuracy (%)** | | Model Size (MB) |
|---|---|---|---|
| | Feature Extract | Fine-tuning | |
| AlexNet | 92.3 | 96.4 | 225 |
| VGG11 | 91.45 | 97.25 | 500 |
| DenseNet121 | 96.1 | 97.8 | **29** |
| ResNet18 | 94.8 | 97.2 | 44 |
| ResNet50 | 95 | 98.1 | 94 |
| ResNet152 | 96.1 | 98.5 | 226 |
| InceptionNet | - | 96.3 | 98 |

### B. Binary Classification of Onion Crop

We obtain the binary classification of the onion images into diseased and healthy classes. We split the images into a test (30%) and train (70%) split. This split in images is shown in Table III.

TABLE III: Onion Images Split

| **Class** | **Train** | **Test** | **Total** |
|---|---|---|---|
| Healthy | 72 | 30 | 102 |
| Diseased | 449 | 195 | 644 |

*1) Imbalanced Dataset Sampler:* Due to the significant difference in the number of images in both classes, we use an imbalanced dataset sampler to re-balance the class distribution during sampling. If the imbalanced dataset sampler is used, the data loader samples the entire dataset and weighs the sample inversely to its class-appearing probability. Hence the majority class is sampled less during an epoch than the minority class, as shown in Figure 4. Without this dataset sampler, the model cannot learn to predict healthy images due to their low number and predicts each image as diseased.

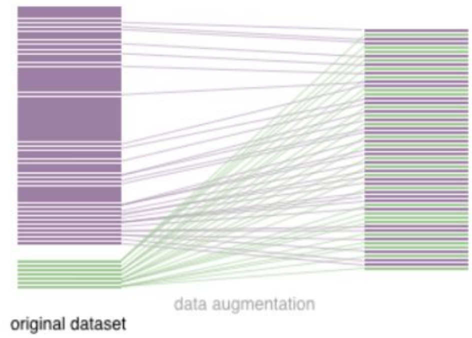



Fig. 4: Imbalanced Dataset Sampler

The results for different baseline models for binary classification of onion images into diseased and healthy classes are shown in Table IV.

TABLE IV: Binary Classification Results

| **Model** | **Accuracy %** | **F-Score** |
|---|---|---|
| DenseNet | **92.4** | **0.9547** |
| ResNet50 | 90 | 0.945 |
| AlexNet | 91.5 | 0.949 |
| VGG11 | 90 | 0.9448 |
| InceptionNet | 83 | 0.89 |

*2) Data Augmentation:* Another experiment we perform is with the CutMix [5] data augmentation. CutMix is an image augmentation strategy in which we replace a patch from an image with another image. The ground truth labels are also mixed proportionally to the number of pixels of the combined image. This augmentation technique has proven to enhance the localization ability of the model by requiring the model to identify the object from a partial view. Some sample images after applying the cut mix data augmentation are shown in Figure 5. The results with and without the cut mix augmentation for the DenseNet baseline are shown in Table V. We observe that cut mix augmentation improves the recall, but all other parameters remain similar, with little better numbers in the case without cut mix.

TABLE V: CutMix Data Augmentation Results

| **Metric** | **DenseNet with Cutmix** | **DenseNet without Cutmix** |
|---|---|---|
| Accuracy (%) | 91.96 | **92.4** |
| Precision | 0.9536 | **0.9889** |
| Recall | **0.9536** | 0.922 |
| F-Score | 0.9536 | **0.9547** |

*3) GradCam [6]:* Gradcam is a technique that uses the gradients of a target class in any classification network, which produces a localization map highlighting the regions in the image which led to the predicted class. We perform GradCam on other onion images to observe whether the model can selectively look at the crops instead of background objects to classify them correctly. The results for the same are shown in Figure 6. The heat maps confirm that the model selectively predicts the correct class by localizing the appropriate regions in the image.

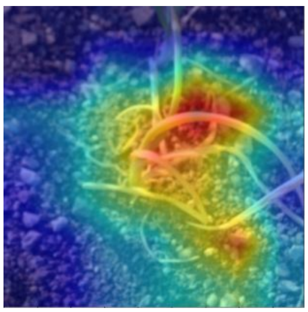
### C. Multi-class Classification of Onion Diseases

We aim to classify the four classes: Basal Rot, Bulb rot Seed, Healthy and Damping off disease using the Densenet baseline model. The confusion matrix for the this classification

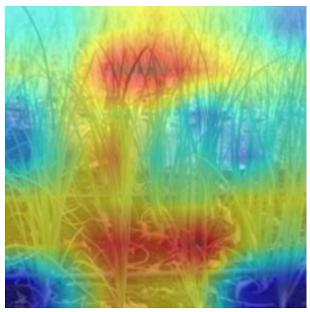
(a)

(b)

Fig. 5: a) Original image of healthy onion crop with a patch of diseased image b) Original image of a diseased onion plant with a patch of healthy onion crop



(a) Diseased Onion Crop



(b) Healthy Onion Crop

Fig. 6: GradCam Heat Maps

is shown in Figure 7. The overall classification accuracy acheived is **73.8%**.

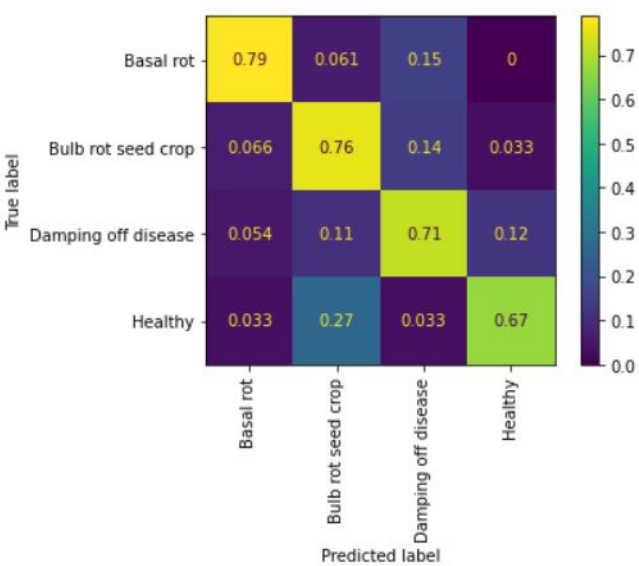



Fig. 7: Confusion Matrix

## VI. Conclusion

We observe that transfer learning on baseline models with the Plant Village dataset helps in disease classification and detection in the onion crop. We analyze different baseline models and estimate the best based on the resultant accuracy and model size. **DenseNet121** model proves to be one of the best-performing models for binary classification achieving an accuracy of 92.4 % and occupying a memory size of only 29 MB. The low memory occupied by the model enables it to be adopted in edge applications which can be deployed directly on the field. The cut mix data augmentation helps bring the classification's recall and precision to the same level, making the model robust to classifying both classes correctly without compromising on accuracy. In the case of multi-class classification, the model architecture achieves an accuracy of around 74% to classify between three types of onion diseases and healthy onion images.

## References

[1] P. Tm, A. Pranathi, K. SaiAshritha, N. B. Chittaragi, and S. G. Koolagudi, "Tomato leaf disease detection using convolutional neural networks," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pp. 1–5, 2018.

[2] B. Liu, Z. Ding, L. Tian, D. He, S. Li, and H. Wang, "Grape leaf disease identification using improved deep convolutional neural networks," *Frontiers in Plant Science*, vol. 11, 2020.

[3] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, 2016.

[4] D. P. Hughes and M. Salathé , "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing," *CoRR*, vol. abs/1511.08060, 2015.

[5] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," 2019.

[6] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, pp. 336–359, oct 2019.