Dual phase convolutional neural network based system aimed at small rice grain dataset for disease identification

Tashin Ahmed [1], **Chowdhury Rafeed Rahman**[1] **Md. Faysal Mahmud Abid**[1]

[1]United International University
Dhaka, Bangladesh
{tahmed153109@bscse, rafeed@cse, mabid143108@bscse}.uiu.ac.bd

Abstract

Although Convolutional Neural Networks (CNNs) are widely used for plant disease detection, they require a large number of training samples while dealing with wide variety of heterogeneous background. In this paper, a CNN based dual phase method has been proposed which can work effectively on small rice grain disease dataset with heterogeneity. At the first phase, Faster RCNN method is applied for cropping out the significant portion (rice grain) from an image. This initial phase results in a secondary dataset of rice grains devoid of heterogeneous background. Disease classification is performed on such derived and simplified samples using CNN architectures. Comparison of the dual phase approach with straight forward application of CNN or Faster RCNN on the small grain dataset shows the effectiveness of the proposed method which provides a five fold cross validation accuracy of 88.11%.

Introduction

As rice grain diseases occur at the very last moment ahead of harvesting, it does major damage to the cultivation process. The average loss of rice due to grain discolouration (Baite et al. 2019) was 18.9% in India. In Bangladesh False Smut was one of the most destructive rice grain disease (Nessa 2017) from year 2000 to 2017. Collecting field level data on agronomy is a challenging task in the context of poor and developing countries. The challenges include lack of equipment and specialists. Farmers of such areas are ignorant of technology use which makes it quite difficult to collect crop disease related data efficiently using smart devices via the farmers. Hence, scarcity of plant disease oriented data is a common challenge while automating disease detection in such areas.

Many researches have been undertaken with a view to automating plant disease detection from the very beginning of deep learning revolution. Sethy, Negi, and Bhoi (2017) applied a threshold based clustering algorithm for this task where they detected defected diseased leaf using K-Means clustering based segmentation. A genetic algorithm was developed by Chung et al. (2016) which was used for selecting essential traits and optimal model parameters for the SVM

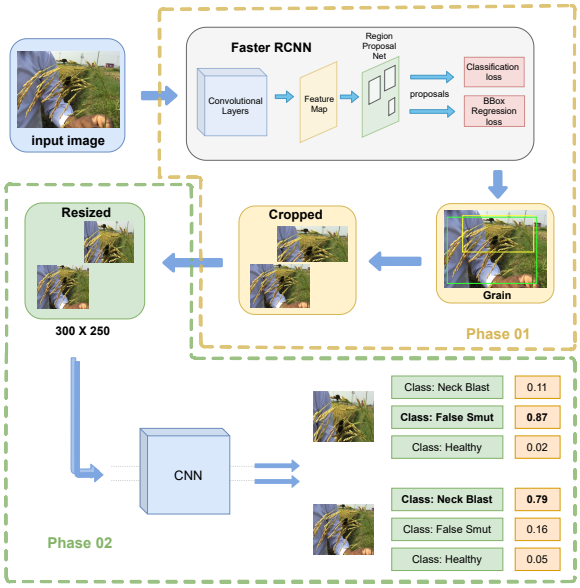


Figure 1: Proposed dual phase approach; Phase one for detection of the significant portion and phase two for classification; a multiclass data have been presented as an example to demonstrate the classification strategy.

classifiers for Bakanae *gibberella fujikuroi* disease. A technique to classify the diseases based on percentage of RGB value of the affected portion was proposed by Islam et al. (2018) utilizing image processing. A similar technique using multi-level colour image thresholding was proposed by Bakar et al. (2018) for RLB disease detection. Deep learning based object classification and segmentation has become the state-of-the-art for automatic plant disease detection. Researchers also experimented with AlexNet (Atole and Park 2018) to distinguish among three classes of rice disease using a small dataset containing 227 images. A similar research for classifying 10 classes of rice disease on a 500 image dataset was undertaken by Lu et al. (2017) using a handmade deep CNN architecture. Furthermore, the benefit of using pre-trained model of AlexNet and GoogleNet has been demonstrated by Brahimi, Boukhalfa, and Moussaoui (2017) when the training data is not large. Their dataset

consisted of nine tomato diseases. A detailed comparative analysis of different state-of-the-art CNN baselines and fine tuned architectures (Rahman et al. 2018) on eight classes of rice disease and pest also conveys a huge potential. It demonstrates two-stage training approach for memory efficient small CNN architectures. Faster RCNN has been applied by Bari et al. (2021) on a real-time approach to diagnose rice leaf disease, although this dataset did not contain natural scene images.

Though the above mentioned researches have a significant contribution towards the automation of disease detection, none of the works addressed the problem of scarcity of (natural scene) data which limits the performance of CNN based architectures for creating a productive solution. Most of the researches focused on image augmentation techniques to tackle the dataset size issue. But applying different geometric augmentations on small size images (Liu and Gillies 2016; Shorten and Khoshgoftaar 2019) result in nearly the same type of image production which has drawbacks in terms of neural network training. Production of similar images through augmentation (Cogswell et al. 2015) can cause overfitting as well.

Our proposed method consists of two phases. The first phase of our proposed method deals with a learning oriented localization architecture. This architecture helps in detecting the significant grain portion of a given image that has a heterogeneous background, which is an easier task compared to disease localization. The detected grain portions cropped from the original image are used as separate simplified images. In the second phase, these simplistic grain images are used in order to detect grain disease using fine tuned CNN architecture. Because of the simplicity of the tasks assigned in the two phases, our proposed method performs well in spite of having only 200 images of three classes.

Our Dataset

Our dataset (balanced) of 200 images consists of three classes - False Smut, Neck Blast and healthy grain class. Some of these images contain both diseases together. A sample image (with heterogeneous background) from each class has been shown in Fig. 2. Data have been obtained and annotated from two different sources: (i) image data from a repository (Rahman et al. 2018) that has undergone previous testing and (ii) field data collected under the supervision of staff from the Bangladesh Rice Research Institute (BRRI) (disease and data collection details in **Appendix A**).

Class	Image Count		Image
	Primary	Secondary	Increment
False Smut	75	85	10
Neck Blast	63	70	7
Healthy	62	64	2
Total	200	219	19

Table 1: Complete dataset and the count difference of primary and secondary dataset.

Table 1 shows detail information about our used data. We consider multi-class images in our dataset (see Fig. 4 of **Appendix A**). In our dual phase approach, a localization algorithm localizes the significant grain portions in phase one. Our secondary dataset comes from this phase - original image broken down into multiple sub-images consisting of significant grain portions. This happens especially in multi-class images (see Fig. 1). No augmentation technique was applied on the training images as these techniques can be prone to overfitting. Supplementary public data related to the paper can be found at **https://zenodo.org/record/7582108**.

Proposed Dual Phase Approach

Fig. 1 shows an overview of our proposed dual phase approach which can learn efficiently from a small dataset of images with significant background heterogeneity. The first stage involves taking the original image, cropping it to a specific size, and then running it through a localization-focused Faster RCNN architecture. From the first stage, two most significant regions have been chosen by the algorithm (see Fig. 1). These regions are cropped and resized to a fixed size. The background of these significant region sub-images are less heterogeneous compared to the original full image. These straightforward significant region sub-images are passed on to CNN model for classification into healthy or a particular disease class.

Stage 1: Localizing Grain Portion

Each input image is resized to 640×480 and is passed through CNN backbone for feature extraction which in turn is passed on to region proposal network (RPN) for generating region proposals. These proposed regions are passed through ROI pooling layer for getting them to fixed size. Finally, RCNN layer decides which of these proposed regions are significant (details in **Appendix B**). The model of this phase only has to predict the significant portions of each image (labeled accordingly). It does not have to worry about predicting class.

Stage 2: Disease Detection from Localized Grain

The Faster RCNN architecture is shown drawing bounding boxes on two significant grain portions in Fig. 1. Each of these frames is cropped and resized to 300×250 before being passed through a CNN architecture. As a result, the single image from the primary dataset has been divided into no more than two images. Each image in the primary dataset goes through the same procedure. This allows for the creation of a secondary dataset with significant grain portions (details in **Appendix A**). To train the CNN architecture, each of these images must be assigned to one of the three classes. As a multi-class data example, the cropped regions in Fig. 1 were predicted by a trained CNN model to represent False Smut and Neck Blast class.

Experimental Setup

We start this section by presenting our algorithm hyperparameters (see Table 5 in **Appendix D** for a better view).

CNN Backbone: ImageNet Pretrained VGG16 model has been used as CNN backbone for feature extraction.

(a) (b) (c) (d)

Figure 2: Background heterogeneity within the dataset demonstrates that when data were collected, different parameters such as unique backgrounds, lights, contrast, and distance were taken into account. The percentage next to the class names represents the overall percentage count. (a) False Smut (37.5%), (b) Neck Blast (31.5%), (c) healthy (31%), (d) multi-class images consisting of Neck Blast and False Smut (13.5% of whole dataset).

Anchor Box Hyperparameters: We use four different size anchor boxes (32, 64, 128, 256 pixels) with four different ratios ((1, 1), $(\frac{1}{\sqrt{2}}, \frac{2}{\sqrt{2}})$, $(\frac{2}{\sqrt{2}}, \frac{1}{\sqrt{2}})$, (2, 2)) for each box. So, the algorithm can propose at most 16 (4×4) anchor boxes per pixel.

Region Proposal Network (RPN) Hyperparameters: Any proposed region with IoU (Intersection Over Union) less than 0.4 with a ground truth object is regarded as an incorrect guess due to the RPN threshold of 0.4 - 0.8, whereas any proposed region with IoU greater than 0.8 with a ground truth object is considered correct. This notion is used for training the RPN layer. Top 200 region proposals from the RPN layer is passed on to the following layers. During non-max suppression, overlapping object proposals are excluded if $IoU > 0.8$.

Learning Rate and Optimizer: Adam optimizer with a learning rate of 0.0001 is used during model training.

We use mAP score (mean average precision) for looking at the phase one localization algorithm performance, while we use accuracy for the stage two classification model. Details of mAP and hardware used for training can be found in **Appendix C**.

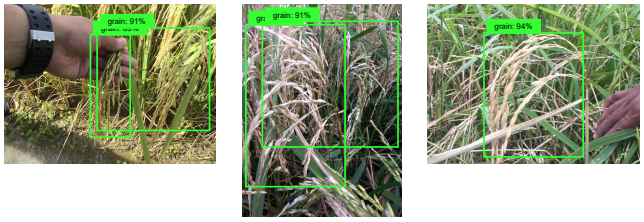
Results and Discussion

The proposed dual-phase approach has been referred to as the **prime experiment**. Straightforward end-to-end classification of disease using CNN has been described as **counter experiment 01**, while using Faster RCNN/ YOLO to directly localize and classify significant portions of each image in a single stage has been referred to as **counter experiment 02**.

Prime Experiment: Dual Phase Approach

Prime experiment has been performed by creating a pipeline of two phases as shown in Fig. 1. Five fold cross-validation has been performed for hyperparameter tuning.

Phase One: Localization of Grains Extracting the significant portion (grain) from a specific image is the goal of phase one. Three different CNN architectures (VGG16, VGG19, and ResNet50) have been tested as the backbone of Faster RCNN. Details of the hyperparameter setting derivation experiments of Faster RCNN have been provided in *Faster RCNN Based Localization Experiments* subsection of **Appendix D**. The goal of these experiments is to choose hyperparameters such that mAP score for significant grain portion localization is maximized. Table 2 shows the five fold cross-validation mAP scores for the different CNN backbones, while utilizing the chosen hyperparameters. Faster RCNN with VGG16 as backbone achieved the best mAP score of 76.32 ± 2.29. Some sample outcomes from phase one are shown in Fig. 3.



(a) (b) (c)

Figure 3: Prime Experiment: Phase One; Sample Outcome: Phase One detected two bounding boxes from (a) and (b) as both boxes meet IoU and accuracy threshold. (c) has only one box as the second box has not met the IoU or accuracy threshold.

Phase Two: Classification Phase two is where the classification result is produced using the image data from phase one. In this phase, three different CNN architectures VGG16, VGG19, and ResNet50 have once more been used for comparison. The best hyperparameter settings from counter experiment 01 (experiment details in **Appendix D**) has been reapplied in this phase. With a validation accuracy of 88.11 ± 3.86, VGG16 stood out as having the best performance as mentioned in Table 3.

Anchor Box Ratio	Anchor Box Pixels	RPN Threshold	CNN Architecture	Overlap Threshold	mAP (%)
(1:1),($\frac{1}{\sqrt{2}}$: $\frac{2}{\sqrt{2}}$), ($\frac{2}{\sqrt{2}}$: $\frac{1}{\sqrt{2}}$),(2:2))	32, 64, 128, 256	0.4 - 0.8	**VGG16**	**>0.8**	**76.32 ± 2.29**
			VGG19	>0.8	70.08 ± 4.54
			ResNet50	>0.8	52.36 ± 5.91

Table 2: Prime Experiment: Phase One

CNN Architecture	Train Loss	Train Accuracy (%)	Validation Loss	Validation Accuracy (%)
VGG16	**0.196**	**94.47**	**0.195**	**88.11 ± 3.86**
VGG19	0.095	89.98	0.093	86.43 ± 2.98
ResNet50	0.367	89.63	0.281	78.00 ± 2.32

Table 3: Prime Experiment: Phase Two

Models	Validation Loss	Validation Accuracy (%)
Faster RCNN	0.312	47.32 ± 5.90
YOLO v5	0.279	68.36 ± 6.43
Proposed	**0.195**	**88.11 ± 3.86**

Table 4: Cross validation accuracy of different object detection methods alongside proposed architecture

Counter Experiment 01

The common approach for rice disease classification from a given image would be to simply pass the image through trained CNN model and perform end-to-end classification. This is what we do in counter experiment 01 (details provided in **Appendix D**). We experiment with five different CNN architectures using transfer learning, fine tuning and adding regularization schemes such as dropout. The best five fold cross-validation accuracy that we could achieve was an accuracy of around 69% using regularized and fine-tuned VGG16 model (see Table 6 of **Appendix D**) which is significantly lower than our proposed dual phase approach performance (around 88% accuracy). It is to note that CNN models with softmax layer at the output are not capable of detecting multiple diseases simultaneously in a single image. In order to simplify things, we consider CNN prediction to be correct for a multi-class image if any one of the present diseases is identified by the model.

Counter Experiment 02

One can argue that instead of using the dual phase approach, we can use a localization algorithm to directly extract and classify the significant grain portions of a given image. In that way, we only need a single phase. If a particular disease exists in the image and if one of the extracted sub-images is labeled as that particular disease by the localization algorithm (other extracted portions can be labeled as healthy, but none of them can be labeled as some other disease), then we can consider a correct classification. In case of the existence of multiple diseases in the image, both diseases need to be labeled in at least one of the extracted sub-images (one sub-image per disease). Comparative performance between Faster RCNN, YOLO v5, and the proposed dual phase approach has been shown in Table 4. We can see that the performance of YOLO v5 is even worse than our best end-to-end CNN model (68% vs 69% accuracy, see previous section), while our proposed approach achieves over 88% validation accuracy. A large amount of training data is necessary for localization algorithms like Faster RCNN and YOLO to converge during training, especially when they have to perform classification besides localization. Faster RCNN and YOLO both failed to outperform the proposed approach because of the current setup's insufficient data count.

Conclusion and Future Work

The main motivation behind this project is to identify a solution for limited labeled agriculture data (supervised task). We offer a solution that can perform well in spite of having a small dataset in a multi-class classification context with heterogeneous image background. Phase one provides a smart localization method that can handle the heterogeneous background present in the real world data of plant disease images. The goal of this first phase is to make the classification task of phase two easier. On a small dataset of rice grain images, an experimental comparison with the use of current CNN architectures has been offered to demonstrate the efficacy of the suggested method. Our proposed approach is model agnostic - one can use any localization algorithm in phase one and any CNN classification model in phase two. Experiments looking at the combination of different architectures in these two phases can improve performance further. Our proposed process is pipeline-based, and phase one may produce false positive or false negative results that are passed on to phase two. Phase two will not be able to properly classify them in such cases. This is one drawback of this system and should be looked into in future.

References

Atole, R. R.; and Park, D. 2018. A Multiclass Deep Convolutional Neural Network Classifier for Detection of Common Rice Plant Anomalies. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, 9(1): 67–70.

Baite, M. S.; Raghu, S.; Prabhukarthikeyan, S.; Keerthana, U.; Jambhulkar, N. N.; and Rath, P. C. 2019. Disease incidence and yield loss in rice due to grain discolouration. *Journal of Plant Diseases and Protection*, 1–5.

Bakar, M. A.; Abdullah, A.; Rahim, N. A.; Yazid, H.; Misman, S.; and Masnan, M. 2018. Rice leaf blast disease detection using multi-level colour image thresholding. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(1-15): 1–6.

Bari, B. S.; Islam, M. N.; Rashid, M.; Hasan, M. J.; Razman, M. A. M.; Musa, R. M.; Ab Nasir, A. F.; and Majeed, A. P. A. 2021. A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework. *PeerJ Computer Science*, 7: e432.

Brahimi, M.; Boukhalfa, K.; and Moussaoui, A. 2017. Deep learning for tomato diseases: classification and symptoms visualization. *Applied Artificial Intelligence*, 31(4): 299–315.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1251–1258.

Chung, C.-L.; Huang, K.-J.; Chen, S.-Y.; Lai, M.-H.; Chen, Y.-C.; and Kuo, Y.-F. 2016. Detecting Bakanae disease in rice seedlings by machine vision. *Computers and electronics in agriculture*, 121: 404–411.

Cogswell, M.; Ahmed, F.; Girshick, R.; Zitnick, L.; and Batra, D. 2015. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Islam, T.; Sah, M.; Baral, S.; and RoyChoudhury, R. 2018. A Faster Technique on Rice Disease Detectionusing Image Processing of Affected Area in Agro-Field. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 62–66. IEEE.

Koiso, Y.; Li, Y.; Iwasaki, S.; HANAKA, K.; Kobayashi, T.; Sonoda, R.; Fujita, Y.; Yaegashi, H.; and Sato, Z. 1994. Ustiloxins, antimitotic cyclic peptides from false smut balls on rice panicles caused by Ustilaginoidea virens. *The Journal of antibiotics*, 47(7): 765–773.

Liu, R.; and Gillies, D. F. 2016. Overfitting in linear feature extraction for classification of high-dimensional image data. *Pattern Recognition*, 53: 73–86.

Lu, Y.; Yi, S.; Zeng, N.; Liu, Y.; and Zhang, Y. 2017. Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, 267: 378–384.

Miah, S.; Shahjahan, A.; Hossain, M.; and Sharma, N. 1985. A survey of rice diseases in Bangladesh. *International Journal of Pest Management*, 31(3): 208–213.

Nessa, B. 2017. *Rice False Smut Disease in Bangladesh: Epidemiology, Yield Loss and Management*. Ph.D. thesis, PhD thesis, Department of Plant Pathology and Seed Science, Sylhet ....

Rahman, C. R.; Arko, P. S.; Ali, M. E.; Khan, M. A. I.; Apon, S. H.; Nowrin, F.; and Wasif, A. 2018. Identification and recognition of rice diseases and pests using convolutional neural networks. *arXiv preprint arXiv:1812.01043*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Sethy, P. K.; Negi, B.; and Bhoi, N. 2017. Detection of healthy and defected diseased leaf of rice crop using K-means clustering technique. *International Journal of Computer Applications*, 157(1): 24–27.

Shorten, C.; and Khoshgoftaar, T. M. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1): 60.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Wilson, R. A.; and Talbot, N. J. 2009. Under pressure: investigating the biology of plant infection by Magnaporthe oryzae. *Nature Reviews Microbiology*, 7(3): 185–195.

Appendix A. Additional Information on Dataset

We have two diseases in the dataset - False Smut and Neck Blast. Neck Blast is generally caused by a fungus known as *Magnaporthe oryzae*. It causes plants to develop very few or no grains at all. Infected nodes result (Wilson and Talbot 2009) in panicle break down. False Smut is caused by a fungus called *Ustilaginoidea virens*. It results in lower grain weight and reduction (Koiso et al. 1994) of seed germination. The Boro rice plant has been chosen for experimental data collection, because Boro species falls under the greatest risk (Miah et al. 1985) of being negatively impacted by Neck Blast and False Smut. When taking pictures, factors like light, distance, and uniqueness were taken into account. Heterogeneity of the background was the primary factor that was considered. Some factors that can spoil the experiment are - illumination, symptom severity, maturity of the plant and diseases. A large versatile dataset can attend on such occasions which can be achieved in the future. The dataset has been kept to three classes for the early stages of the investigation. Also, it is quite challenging and burdensome to collect different rice disease image dataset throughout the year as different diseases occur at different time. So, at this early stage of the investigation three classes is competent to deliver a sufficient result.

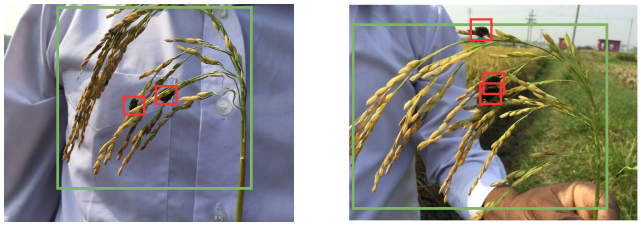


Figure 4: Sample multiclass data; Labeled green for Neck Blast and red for False Smut class

Appendix B. Consecutive Stages of Phase One

Convolutional Neural Network (CNN): In order to avoid sliding a window in each spatial position of the original image, CNN architecture is used in order to learn and extract feature map from the image which represents the image effectively. The spatial dimension of such feature map decreases whereas the channel number increases. For the dataset used in this research, VGG16 architecture has proven to be the most effective. Hence, VGG16 has been used as the backbone CNN architecture which transforms the original image into $20 \times 15 \times 512$ dimension.

Region Proposal Network (RPN): The extracted feature map is passed through RPN layer. For each pixel of the feature map of spatial size 20×15, there are 16 possible bounding boxes (4 different aspect ratios and 4 different sizes mentioned in bold letter in Table 5). So, that makes total 16×20×15 = 4800 possible bounding boxes, RPN is a two branch Convolution layer which provides two scores (branch one) and four coordinate adjustments (branch two) for each of the 4800 boxes. The two scores correspond to the probability of being an object and a non-object. Only those boxes which have a high object probability are taken into account. To remove overlapping bounding boxes and retain the high probability unique boxes, non-max suppression (NMS) is used. The overlap must meet a threshold of 0.8 IoU. Top 200 proposals ranked by object probability from the object proposals are forwarded to the next layers.

ROI Pooling: Each of the 200 selected object proposals correspond to some region in the CNN feature map. For passing each of these regions on to the dense layers of the architecture, each of the regions need to be of fixed size. ROI pooling layer takes each region and turns them into 7×7×512 using bilinear interpolation and max pooling.

RCNN Layer: RCNN layer consists of fully connected dense layers. Each of the 7×7×512 size feature maps are flattened and passed through these fully connected layers. The final layer has two branches. Branch one predicts if the input feature map is background class or significant grain portion. Branch two provides four regression values denoting the adjustment of the bounding box to better fit the grain portion. For each feature map, if the probability of being a grain is over 0.6, only then is the feature map considered as a probable grain portion and the adjusted coordinates are mapped to the original image in order to get the localized grain portion. The overlapping boxes are eliminated using NMS. The remaining bounding box regions are the significant grain portions.

Loss Function: The trainable layers of Faster RCNN architecture are: CNN backbone, RPN layer and RCNN layer. A loss function is needed in order to train these layers in an end to manner which is as follows.

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

The first term of this loss function defines the classification loss over two classes which describe whether predicted bounding box i is an object or not. The second term defines the regression loss of the bounding box when there is a ground truth object having significant overlap with the box. Here, p_i and t_i denote predicted object probability of bounding box i and predicted four coordinates of that box respectively while p_i^* and t_i^* denote the same for the ground truth bounding box which has enough overlap with predicted bounding box i. N_{cls} is the batch size (256 in this case) and N_{reg} is the total number of bounding boxes having enough overlap with ground truth object. Both these terms work as normalization factor. L_{cls} and L_{reg} are log loss (for classification) and regularized loss (for regression) function, respectively.

Appendix C. Hardware, Utilized CNN Models and Evaluation Metrics

For the training environment, assistance has been taken from two different sources.

- Royal Melbourne Institute of Technology (RMIT) provides GPU for international research enthusiasts and they provided a Red Hat Enterprise Linux Server along with the processor Intel Xeon E5-2690 CPU, clock speed of 2.60 GHz. It has 56 CPUs with two threads per core, 503 GB of RAM. Each user can use up to 1 petabyte of storage. There are also two 16 GB NVIDIA Tesla P100-PCIE GPUs available. First phase was completed through this server.
- Google Colab (Tesla K80 GPU, 12GB RAM) and Kaggle kernel (Tesla P100 GPU) have been used for counter experimentation.

Fig. 5 shows architectures and key blocks of the applied CNN architectures. Experiments have been performed using five state-of-the-art CNN architectures which are described as follows.

VGG16 is a sequential architecture (Simonyan and Zisserman 2014) consisting of 16 convolutional layers. Kernel size in all convolution layers is three.

VGG19 has three extra convolutional layers (Simonyan and Zisserman 2014) and the rest is the same as VGG16.

ResNet50 belongs to the family of residual neural networks. It is a deep CNN architecture (He et al. 2016) with skip connections and batch normalization. The skip connections help in eliminating the gradient vanishing problem.

InceptionV3 is a CNN architecture (Szegedy et al. 2016) with parallel convolution branching. Some of the branches have filter size as large as 7×7.

Xception takes the principles of Inception to an extreme. Instead of partitioning the input data into several chunks, it maps the spatial correlations (Chollet 2017) for each output channel separately and performs 1×1 depthwise convolution.

All results have been provided in terms of 5 fold cross validation. Accuracy metric has been utilized in order to compare dual phase approach against implementation of CNN on original images without any segmentation. Accuracy is a suitable metric for balanced dataset.

$$Accuracy = \frac{TP}{TP + FP + TN + FN}^{1} \tag{2}$$

Segmenting the grain portion is the goal of the first phase of the dual phase approach. For evaluating the performance of this phase, mAP (mean average precision) score has been used. Precision, recall and IoU (Intersection over Union) are required to calculate mAP score.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$IoU = \frac{AOI}{AOU}^{2} \tag{5}$$

If a predicted box IoU is greater than a certain predefined threshold, it is considered as TP (true positive). Otherwise, it is considered as FP (false positive). $(TP + FN)$ (FN being false negative) is actually the total number of ground truth bounding boxes. Average precision (AP) is calculated from the area under the precision-recall curve. If there are N classes, then mAP is the average AP of all these classes. In this research, there is only one class of object in phase one, that is the significant grain portion class. So, here AP and mAP are the same.

Appendix D. Experiment Details

Counter Experiment 01

In this experiment, five different CNN architectures were used (see Table 6). Using imagenet pretrained models, three transfer learning methodologies, frozen layer, fine tuning, and fine tuning + dropout have been applied. The freezing layer approach, also known as the default transfer learning method, was used initially. With a validation accuracy of 63.33 ± 2.04, VGG16 performed better than other CNN architectures. After that, fine tuning has been used, which resulted in an increase in validation accuracy for VGG16 of 67.79 ± 3.24. The CNN architectures have been modified to incorporate dropout, which yields a significant improvement of 69.43 ± 3.41 for VGG16. By experimenting with dropout on different positions inside individual CNNs, fine-tuning and fine-tuning + dropout have both been performed repeatedly.

Faster RCNN Based Localization Experiments

Three different CNN architectures have been tested as the backbone of Faster RCNN. Purpose of this experiment is to evaluate Faster RCNN's capability for effective significant portion detection (grain). Additionally, ResNet50 was chosen over Xception and InceptionV3 (which are mentioned in Table 6) due to the lower validation loss along with VGG16 and VGG19 from previous experiment. Pretrained models have been used from COCO and Imagenet. For Faster RCNN, various hyperparameter optimizations have been used to achieve best results are shown in Table 7.

Default settings from Faster RCNN paper (Ren et al. 2015) for anchor box ratio were (1:1), (2:1), (1:2) and anchor box pixels were 128, 256, 512 which produces 3×3=9 anchor boxes per pixel. The default RPN threshold of (0.3 - 0.7), overlap threshold 0.8 and default anchor box ratios and pixels, VGG16 (imagenet pretrained model) provided the best mAP score of 71.0 ± 4.0. After tuning RPN threshold to (0.4 - 0.8), anchor box ratios to (1:1), $(\frac{1}{\sqrt{2}} : \frac{2}{\sqrt{2}})$, $(\frac{2}{\sqrt{2}} : \frac{1}{\sqrt{2}})$, (2:2) and pixel sizes to 32, 64, 128, 256, 4×4=16 anchor

[1] TP: True Positive, FP: False Positive, TN: True Negative, FN: False Negative

[2] AOI: Area of intersection, AOU: Area of union (with respect to ground truth bounding box)

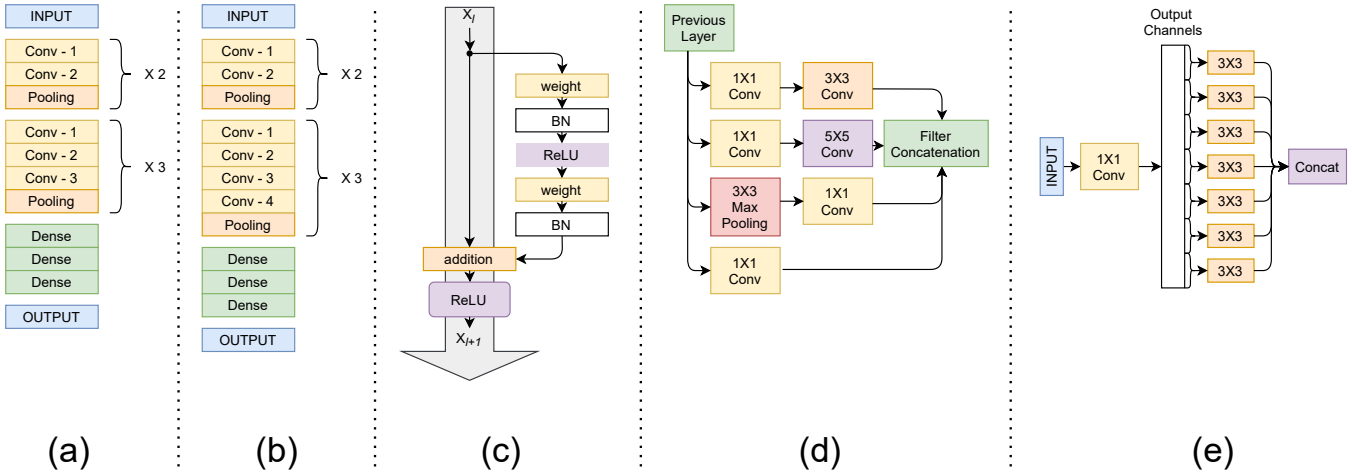


Figure 5: (a): all 16 blocks of VGG16, (b): all 19 blocks of VGG19, (c): Resdiual module of ResNet, (d): Inception module is to act as a multi-level feature extractor in InceptionV3, (e): Extreme module of the Inception module which is utlized in Xception.

Hyperparameter	**Optimized Value**
Anchor Box Count	9, **16**
Anchor Box Size (pixels)	[**32, 64, 128, 256**], [128, 256, 512]
Anchor Box Ratios	[(1,1), (2,1), (1,2)], [**(1,1), ($\frac{1}{\sqrt{2}}$, $\frac{2}{\sqrt{2}}$), ($\frac{2}{\sqrt{2}}$, $\frac{1}{\sqrt{2}}$), (2,2)**]
RPN Threshold	0.3 - 0.7, **0.4 - 0.8**
Proposal Selection	**200**, 2000
Overlap Threshold	>**0.8**, >0.9
Learning Rate	0.001, **0.0001**, 0.00001
Optimizers	**Adam**, SGD

Table 5: Experimented hyperparameters. Bold values were selected for the prime experiment.

Transfer learning Approach	CNN Architecture	Validation Loss	Validation Accuracy (%)
Freezed Layer	**VGG16**	**2.08**	**63.33 ± 2.04**
	VGG19	1.08	43.75 ± 3.43
	Xception	2.34	31.25 ± 4.04
	InceptionV3	9.23	37.50 ± 3.89
	ResNet50	4.47	31.25 ± 3.27
Fine Tuned	**VGG16**	**2.71**	**67.79 ± 3.24**
	VGG19	1.77	55.04 ± 3.00
	Xception	6.29	43.76 ± 1.88
	InceptionV3	7.42	41.73 ± 3.66
	ResNet50	2.47	38.20 ± 1.34
Fine Tuned + Dropout	**VGG16**	**3.47**	**69.43 ± 3.41**
	VGG19	3.11	57.18 ± 2.64
	Xception	5.72	47.17 ± 2.11
	InceptionV3	4.12	48.22 ± 3.14
	ResNet50	2.81	42.31 ± 1.32

Table 6: Counter Experiment 01: CNN

boxes have been produced which provides better outcome than before. This setting improved the mAP for VGG16 (imagent pretrained model) to 76.32 ± 2.29 which is the peak outcome after several optimization.

Pretrained Model	Anchor Box Ratio	Anchor Box Pixels	RPN Threshold	CNN Architecture	Overlap Threshold	mAP (%)
Imagenet	(1:1), (2:1), (1:2)	128,256, 512	0.3 - 0.7	VGG16	>0.8	71.0 ± 4.0
				VGG19		47.06 ± 2.01
				ResNet50		67.14 ± 6.68
	(1:1), $(\frac{1}{\sqrt{2}} : \frac{2}{\sqrt{2}})$, $(\frac{2}{\sqrt{2}} : \frac{1}{\sqrt{2}})$, (2:2)	32, 64, 128, 256	0.4 - 0.8	**VGG16**	**>0.8**	**76.32 ± 2.29**
					>0.9	63.42 ± 2.36
				VGG19	>0.8	70.08 ± 4.54
					>0.9	70.30 ± 2.36
				ResNet50	>0.8	52.36 ± 5.91
					>0.9	40.02 ± 3.03
COCO	(1:1), (2:1), (1:2)	128,256, 512	0.3 - 0.7	VGG16	>0.8	48.32 ± 4.79
				VGG19		32.30 ± 4.83
				ResNet50		46.36 ± 2.04
	(1:1), $(\frac{1}{\sqrt{2}} : \frac{2}{\sqrt{2}})$, $(\frac{2}{\sqrt{2}} : \frac{1}{\sqrt{2}})$, (2:2)	32, 64, 128, 256	0.4 - 0.8	**VGG16**	**>0.8**	**54.24 ± 2.23**
					>0.9	48.0 ± 2.10
				VGG19	>0.8	42.36 ± 1.02
					>0.9	41.07 ± 5.21
				ResNet50	>0.8	30.23 ± 3.0
					>0.9	28.42 ± 4.84

Table 7: Faster RCNN localization experiments