Improved Detection and Analysis of Wheat Spikes Using Multi-Stage Convolutional Neural Network

M. A. Batin[1], Muhaiminul Islam[1], Md Mehedi Hasan[1*], and Stanley J. Miklavcic[2]

[1]*Department of Robotics and Mechatronics Engineering, University Of Dhaka, Dhaka 1000, Bangladesh*

[2]*Phenomics and Bioinformatics Research Centre, University of South Australia, Adelaide 5095 AUS*

**E-mail: mmhasan@du.ac.bd*

Received on 26 June 2022, Accepted for publication on 17 January 2023

Abstract

High throughput plant phenotyping is the advanced scientific approach for rapid phenotyping of plant traits, especially high consumable grains or crops, which is designed to process a high volume of data in a short time for plant breeders and cultivars to utilize. Detection and counting of crop traits such as plants, fruits,wheat or rice spikes, sorghum head, and plant diseases is more advanced research in this field, where real-world data are collected using aerial and land-based imaging platforms equipped with a variety of geospatial sensors, and their statistical analysis is conducted using Artificial Intelligence (AI) and Deep Learning-based solutions. In this paper, we contributed to solving such a challenge of phenotyping by detecting and counting wheat spikes from land-based imaging by applying a Region-based Convolutional Neural Network (CNN) model. Our method employs the use of CNN to extract features from the imaging platform and the learning model is trained todetect and count wheat spikes in field images basedon these extracted features. Using the publicly available SPIKE dataset to train and test our model, our proposed method achieved 98% average precision and 91% average F1 score on the test set. Our results show a significant improvement of 2.9% and 11.2% in detection accuracy as well as 1% and 3% in average precision metric over state-of-the-art Faster Region-based Convolutional NeuralNetwork (Faster-RCNN), and RetinaNet, respectively, andhave the potential to significantly benefit plant breedersby facilitating the selection of wheat varieties with high yields.

Keywords: Plant Phenotyping, Neural Network, Spike detection, Field imaging, Machine Learning

1. Introduction

Modern AI-based agricultural technologies are used to observe plants scientifically and systematically. Plant Phenotyping facilitates us to understand how a plant grows and its development in various conditions. It describes the study of plant structure and function which depends on the dynamic interaction between genetic background and environment. Again, phenotyping in- cludes the idea of measuring and analyzing observable plant characteristics which are advantageous in the agri- cultural aspect for selecting crops [1]. Though pheno- typing is not the latest innovation, manually monitoring features like plant height, growth, tolerance, resistance, nitrogen content, biomass, and yield counts takes a lot of time and effort. With high throughput plant phenotyping using image-based sensors, reconfigurable harvesting tools, and drones generates much more information in less duration. Modern plant phenotyping is an emerging science that gives important details on how genetics,epigenetics, environmental stresses, and crop manage- ment may influence the selection of plants that are fitfor their surroundings. It is feasible to crop productivity to meet the demands of the increasing human population by implementing high throughput phenotyping [2].

Wheat is one of the world's three most important crop species, which is trading for nearly $50 Billion globally on an annual basis [3]. It is the main source of nutrition for 2.5 billion people living in 89 nations [4]. As the population in the world is increasing and so the demand for cereal crops like Wheat, Sorghum, Millets, Maize, and Rice is also increasing as these are the main grain meals for the majority of the people. Among these, wheat is traded larger than all other crops combined and is produced on the largest amount of land of any food crop. So, it can be said that Wheat is the backbone of food security [5]. So, this emphasizes the importance of finding wheat plant kinds that are hardier and yield more production while also improving tolerance to biological and chemical challenges.

The formation of the spike or ear is a critical plant physiological phase in the growth of wheat. So the spike numbers detected in a wheat field got an enormous attraction in modern days for high throughput plant phenotyping. In Fig. 1, a general workflow of high throughput plant phenotyping is shown, where images can be collected from land-based or aerial imaging or both to conduct computer vision tasks such as ob- ject detection, instance segmentation using methods like image processing, machine learning or deep learningfor further data processing. These data acquisition and assessment processes can be used to carry out high throughput phenotyping tasks such as spike detection, plant growth, seedling count, vegetation indexing, and density analysis. In the early days of phenotyping, deep learning approaches were applied for spike detection in a controlled environment including indoor agriculture and vertical farming. However, it has always been a more challenging task to detect spikes in real-world scenarios.

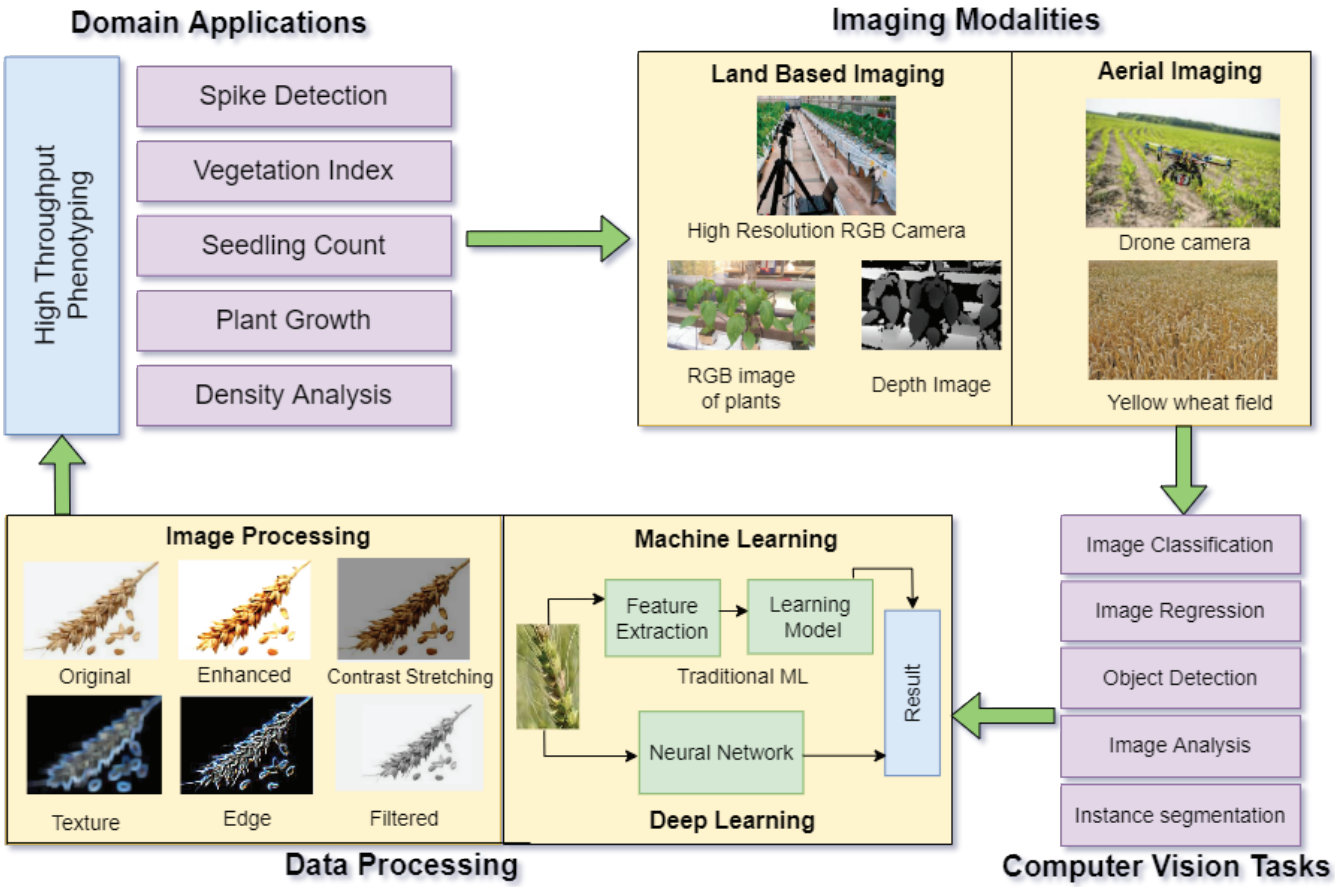

Fig. 1: High Throughput Phenotyping process workflow

Development of region-based convolutional neural network (R-CNN) is a step towards solving the problem of detecting objects in complex scenarios through the combined use of region proposals with CNN features [6]. Utilizing the computed CNN features from extracted region proposals results in a far better detection and localization of objects in images than other previous methods that didn't rely on those region proposals. This success naturally can be translated to the challenging task of detecting wheat spikes from real-world field images with state-of-the-art precision and accuracy.

Many deep neural networks, Fast R-CNN, Faster R-CNNN, RetinaNet, YOLOv5 (You Only Look Once), etc. are developed for wheat spike detection and counting [3, 7, 8]. These models are trained with a large spike dataset so that later it would be able to detect spikes from images. The detected spikes can describe the distinctive features of wheat varieties, and the results can then be utilized for smart farming, data intelligence, and predictive analysis using Artificial Intelligence (AI) in real- time. So, our purpose here is to apply a deep learning method to detect wheat spikes from real-world field images with better accuracy and precision than existing state-of-the-art methods and to employ our method in further applicable scopes of AI in agriculture.

Our proposed method accomplished an average accu- racy of 84% and the F1 score is 0.91 in detecting wheat spikes on SPIKE dataset [3]. The approach generatesa detailed list of bounding box regions for recognizing spikes in photos that are not noticeably visible in the training phase. A robust deep learning study requires extensive training with large, increased data sets. In our study, we intended to develop a detection method that can classify wheat spikes with better accuracy. We have compared our results with two existing approaches developed by hasan et al. [3] wen et al. [7] and claimto be more efficient, feasible, and robust in the field of spike detection in a complex environment.

1.1 Background

In the field of phenotyping, many research works are ongoing for the detection, analysis, and counting of wheat spikes. Here, we will discuss some recent works that have been done on plant phenotyping using Deep Learning methods. Among researchers studying plant phenotyping, there has been a significant amount of literature on suggested techniques. Some novel deep learning approaches are proposed for the detection and recognition of wheat spikes.

Hasan et al. [3], in their paper, proposed a fine- tuned Region-based Convolutional Neural Network(R- CNN) model for classifying and analyzing wheat spikes from field-based images. The authors here used Faster R-CNN as the

model structure to be trained on a training set. A pretrained VGG-16 model was used as the backbone architecture to extract features from input images which were then fed to the Region Proposal Network (RPN) to regress bounding boxes and the Classification Network to classify the bounding boxes as either spike or background. A spike dataset named SPIKE was developed by capturing in-field images using high definition RGB cameras for training the Faster R- CNN model. The model achieved an average accuracy and F1 score of 93.4% and 0.95. But the model was not applied to oblique-view images which makes it prone to low accuracy when it comes to partially occluded spikes, especially in high-density regions.

Misra et al. [4] developed SpikeSegNet, a novel deep learning method for detecting, recognizing, and counting multiple wheat spikes. For preparing the dataset, images were captured with a high-resolution camera and later cropped for detecting regions of interest. This model ar- chitecture consists of two feature subnetworks, the Local Patch extraction Network (LPNet) and the Global Mask refinement Network (GMRNet). The proposed model achieved an average precision, accuracy, and robustness of 99%, 95%, and 97% for counting spikes. In a dataset with illuminated images, the model achieved sufficient robustness, with no significant drop in segmentation performance. But performance here drops when a spike overlaps with another in the image, then the model counts two or more spikes as one [4].

Zhao et al. [8] proposed a more advanced and de- veloped method, improved YOLOv5 to detect spikes precisely in UAV images, and overcome false spike detection caused by occlusion conditions. High-quality in-field images were taken by a UAV equipped with a high-quality RGBD camera for the dataset. After providing an input image to the network, the backbone module extracts its features, the neck module generates a multi-scale, multi-channel feature pyramid based on Path Aggregation Network (PANet), and finally the head module outputs detection boxes with a confidence score indicating the category and coordinates of wheat spikes contained. The average accuracy of wheat spike detection in UAV images is 94.1%, which is 10.8% higher than the standard YOLOv5. Yet the spikes here are detected as points instead of bounding boxes so the height of the spike cannot be described from the detection method. This same deficiency was found in another Convolutional Neural Network (CNN) based model, WheatNet, pro- posed by Khaki et al. [9]. For counting wheat heads this method is accurate and robust for different conditions in the field. Images were collected from 10 different locations around with a high spectrum RGB camera. The proposed model uses a truncated MobileNetV2 model as a lightweight backbone for extracting features with various scales which are then merged to counter varia- tions in image scale. This model uses significantly fewer parameters and so runs very fast. The characteristics make the model lightweight enough to be used in in- field, mobile platforms.

Wen et al. [7] developed a technique based on Spik- eRetinaNet in order to detect and count small dense objects in complex scenes. SpikeRetinaNet is an opti- mized version of the RetinaNet model and is consisting of three crucial steps: the usage of BiFPN for more effective integration of multiscale features, DSA block for improved network refinement, and the application of Soft-NMS to solve the occlusion problem. The au- thors trained and tested their method using the Global Wheat Head Detection (GWHD) dataset augmented with Wheat-Wheatgrass Spike Detection (WSD) images. The model achieved wheat spike mAP and count detection rates of 92.62% and 92.88%, respectively.

The field of phenotyping is predominantly assigned with analyzing the spike dataset for detection and recog-nition of wheat spikes. Increasingly, plant biologists and breeders rely on high-throughput phenotyping methods to evaluate various plant traits, which are then used to examine the plant's response to various external condi- tions and treatments in an effort to improve grain yield. In this research, to gather a diverse dataset containing images of wheat spikes in real-world field conditions, we applied the SPIKE dataset [3] which is accurate, and complete to develop a standard model for this approach. Since most existing methods are struggling with detecting partially occluded or visible parts of a dense region, our main contribution to this research is to develop an improved end-to-end deep learning method to get better accuracy detection. We applied the Cascade R-CNN architecture [10] for the detection by fine-tuning the hyper parameters and customizing the model to suit the need of our spike detection task. In doing so, we observed an improvement over existing state-of-the-art methods developed by Hasan et al. [3] and Wen et al. [7]; a 2.9% and and 11.2% increase in Mean Average Precision (mAP), 8% and 13% increase in Recall, 9% and 15% increase in Accuracy metric as well as 5% and 9% increase in Average F1 score for detection count. Moreover, the experimental results demonstrate its significance in various analysis tasks such as yield estimation, plant growth measure, genotype traits, etc.

2. Methodology

2.1 Proposed Method

Our goal in this work is to develop an improved method for detecting and counting wheat spikes from in-field land-based imaging. To achieve this, we need a fast and accurate system that can detect spikes in challenging field conditions as well as test the robustness of the model in terms of detecting those spikes. "Fig. 2 shows a workflow diagram for such a system."

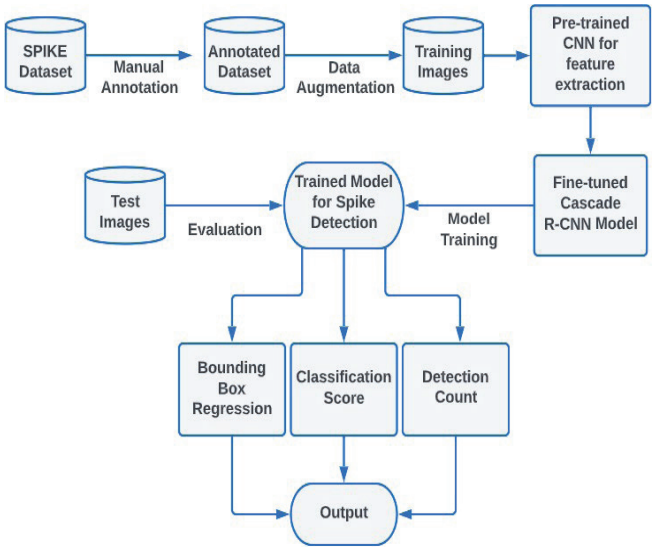

Fig.2: General work-flow diagram of our proposed system.

2.2.1 Experimental Setup

In this paper, we adopted the SPIKE dataset developed by Hasan et al. [3] The SPIKE dataset was obtained by capturing images of wheat fields of 90 plots, 18 rows and 5 columns using a mobile land-based imag-ing platform over four months, from July 21, 2017 – November 22, 2017. The plots were planted with 10 spring wheat (Triticum aestivum L.) varieties (Drysdale, Excalibur, Gladius, Gregory, Kukri, Mace, Magenta, RAC875, Scout, Yitpi). This makes the dataset morerobust in detecting spikes of different wheat varietiesi.e., spikes of different shapes and sizes.

The mobile platform to capture images was a steel- framed, four-wheeled wagon with a central overhead rail for mounting imaging sensors. An 18.1-megapixel Canon EOS 60D digital camera was mounted on the rail to capture images of the plots from a slight oblique view (55 degrees from the horizontal overhead rail). Theimages captured had a resolution of 5184 × 3456 pixels or an image resolution of 0.4mm per pixel.

2.2 The SPIKE Dataset

The SPIKE dataset contains, in total, 335 images of ten wheat varieties at three different growth stages. Each image is expertly annotated to denote the bounding boxes of wheat spikes resulting in a total of approximately 25,000 annotations. The three different growth stages correspond to three different situations for spike and canopy color shown in Fig. 3:

- Green Spike Green Canopy (GSGC)
- Green Spike Yellow Canopy (GSYC)
- Yellow Spike Yellow Canopy (YSYC)

Fig. 3: Examples of training images captured at three different growth stages.

For our purpose in this paper, we took exactly 320 images with proper annotation from the SPIKE dataset and split them into train, validation and test sets randomly. We ensured proper distribution of different types of images in train, validation and test sets to eliminate biases while testing our model.

Table 1: Number of images from each growth stage for train, validation and test set

IMAGES	GSYC	GSGC	YSYC	TOTAL
TRAIN	222	34	34	290
VALIDATION	9	3	3	15
TEST	9	3	3	15
TOTAL	240	40	40	320

Each image in dataset has dimension of 2500x1500 pixels and our dataset contains over 22,000 annotations of wheat spikes. For a standard evaluation and testing purpose, we converted the original annotation format of the SPIKE dataset to COCO format [11] usedas a standard format to evaluate object detection models. Here, each bounding box annotation is a list of coordinates denoting [x, y, width, height]. Fig. 4 shows the bounding box annotations of a training image and as it can be seen, the wheat spikes have been properly annotated with tight bounding box regions.

Fig. 4: Example of bounding box annotations of a training image.

2.2.2 Experimental Model Setup

In this paper, as a method for detecting and counting wheat spikes from field images, we applied the Cas-cade R-CNN

architecture [10] and fine-tuned the hyper- parameters to obtain an optimized model applicable to our problem domain. Cascade R-CNN is a multi-stage extension of the R-CNN architecture [6], where detectors further down the cascade architecture are sequentially more discriminating against detections that are close to false positives. These stages of the R-CNN architecture are trained successively, with the previous step's output used to train the subsequent stage.

2.2.2.1 Data Augmentation

Before using it for the training of our model, we augmented the dataset with some standard data aug- mentation techniques to avoid overfitting the model. These data augmentation techniques include Resizing (to 1333x800 size), Random Flipping (with 0.5 probability being vertical or horizontal), and padding (upsampled to a multiple of 32). The augmentations help create virtual copies of the wheat spike images to improve model generalization capability.

2.2.2.2 Model Architecture

Essentially, Cascade R-CNN is an extension of the two-stage architecture of the Faster R-CNN [12, 13], shown in Fig. 5. The first stage, a proposal sub-network ("H0"), produces preliminary detection hypotheses, known asobject proposals, after being applied to the entire input image. For our purpose, this is done using a Region Pro-posal Network (RPN) [12]. The second stage, a regionof interest detection sub-network ("H1"), processes these hypotheses to detect regions that might contain objects. This stage is denoted as the detection head. The final stage assigns a final classification score ("C") and abounding box ("B") to each detection. In Fig. 5, forboth of the architectures, "I" stands for input image,"conv" stands for backbone convolutions, "pool" stands for region-wise feature extractor, "H" stands for networkhead, "B" stands for bounding box regressor, and "C" stands for bounding box classifier. In both architectures, "B0" is the proposal bboxes.

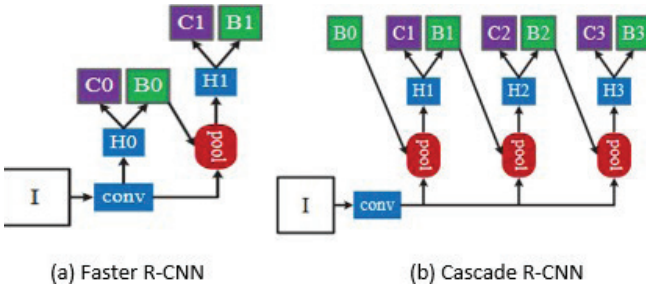

Fig. 5: Abstract representation of the architectures of Faster R-CNN and Cascade R-CNN methods [10].

For the backbone architecture of our network, we used a ResNet-50 [14] model pre-trained on ImageNet dataset to extract features from our input image. In addition to using a Feature Pyramid Network (FPN) [13] as the sub-network to arrange the features into a multi-scale, multi- channel feature pyramid, we implemented the Balanced Feature Pyramid (BFP) proposed by Pang et al. [15].

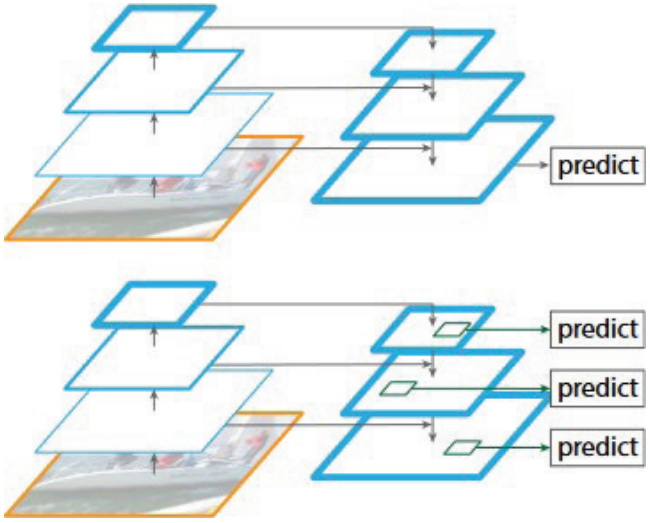

Fig. 6: Architecture representation of Feature Pyramid Network (FPN) [13].

In Fig. 6, the top diagram shows a top-down and skip connection architecture, where a single high-level featuremap with fine resolution is created to make predictions, whereas the bottom diagram represents FPN with similararchitecture, where predictions are made separately on feature maps of all levels.

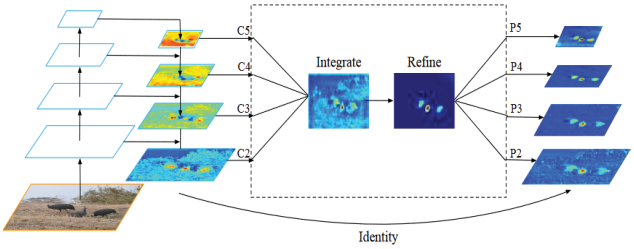

Fig. 7: Pipeline and heat map visualization of balanced feature pyramid [15].

For the classification network, the proposed cascaded architecture is used where for bbox regression and classi-fication loss, SmoothL1Loss and CrossEntropyLoss were used respectively in each stage.

SmoothL1Loss (Huber loss),

$$l_n = \begin{cases} 0.5(x_n - y_n)^2 & ,\text{if}|x_n - y_n| < 1 \\ |x_n - y_n| - 0.5 & ,\text{otherwise} \end{cases} \quad (1)$$

Cross Entropy Loss,

$$\log\_loss = \frac{1}{N}\sum_{i=1}^{N} -(y_i * log\,(p_i) + (1 - y_i) * log\,(1 - p_i)) \quad (2)$$

2.2.2.3 Model Configuration

Taking in an input image, our model extracted and arranged features using FPN and BFP sub-networks into a multi-scale pyramid with 5 different scales, each con- taining 256 feature channels. The features were extracted using a pretrained ResNet-50 model. These features were then fed into the RPN sub-network with an Anchor- Generator. An anchor is a bounding box generated at each point of the feature map with a specific scale and aspect ratio. At each point, 3 anchors were generated. These 'anchors' or object proposals are then processed in the region-of-interest sub-network. Here we used CascadeRoIHead with 3 stages. Each stage takes the output from the previous stage as its input for bounding box regression to localize the wheat spikes and assign a classification score to correctly classify each bbox as either spike or background.

At the training stage, the region proposal network applies Non-Maximum Suppression (NMS) for every generated anchors with an IoU threshold of 0.7 and max-imum number of proposals set to 1000 per image. Then each of the detection head stages samples and assigns a prediction score to all the proposed regions. This is done by setting an IoU threshold to classify between a positive sample (spike) and a negative sample (background). Unlike Faster R-CNN where a single detector does this regression and classification task, Cascade R-CNN employs multiple cascaded detectors with increasing IoU thresholds for better detection and localization of the wheat spikes. For our purpose, we employed three cascaded detectors with IoU thresholds of 0.5, 0.6, and 0.7 respectively. All the stages predict a confidence score for each of the detected bounding boxes, with Stage 3 prediction score being the final output score for each detection. For testing, the R-CNN network samples out detections with low confidence scores and use NMS to output detected bounding boxes closest to ground truth. In our case, the score threshold was set to 0.5 and IoU threshold to 0.5 for best performance.

2.2.2.4 Hyperparameters

In this paper, we used a certain configuration for our model hyperparameters after several empirical tests. Here for the optimization of model parameters, the Stochastic Gradient Descent (SGD) algorithm was used. Because training might be unstable at initial iterations, a warmup method for learning rate was used where the initial learning rate = lr * warmup ratio, which in our case is 1e-4. The learning rate was decreased by a factor of 10 after epochs 167 and 229. Also, we grouped the training images into a batch size of 2 images per iterationduring training.

Table 2: Values of the learning hyper-parameters

Hyperparameters	Type	Value
Optimizer	SGD	momentum=0.9 weight decay=0.0001
Learning Rate		lr = 0.01
Learning rate policy	Step	step = [167,229]
Warmup policy	Linear	warmup iterations = 500 warmup ratio = 0.01
Epochs		Number = 250
Classification stages	Cascaded	Number = 3
IOU threshold		Stage 1 = 0.5 Stage 2 = 0.6 Stage 3 = 0.7
Score threshold	RCNN	Score thr = 0.5

2.2.2.5 Implementation

To easily implement the model architecture of our choice as well as efficiently process dataset augmentation, training, and testing pipelines, we have opted to use the MMDetection codebase developed by Chen et al. [16] MMDetection is open-source object detection and instance segmentation codebase developed with PyTorch. The main advantage of this codebase is its ease of use and modular nature when it comes to model representa- tion. Also, the codebase contains several popular single- stage, two-stage, and multi-stage object detection and instance segmentation methods, which is helpful for con- veniently implementing our choice of model architecture.

3. Results and discussions

3.1 Detection and Count Results

To test the performance of our trained model, we utilized Mean Average Precision (mAP), the standard evaluation metric for COCO-formatted datasets. The mean average Precision, which quantifies the method's precision at various recall levels, can be represented as follows:

$$mAP = \frac{1}{101}\sum_{r_i\in\{0,0.01\ldots,1\}} \max_{r_i:r_i\geq r} p(r_i) \quad (3)$$

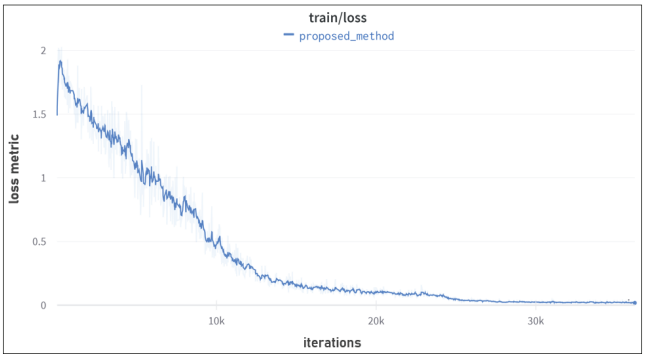

Fig. 8: Training loss of the method over 250 epochs

In other words, it is the average precision of 101 equally spaced Recall levels [0.1, 0.01,..., 1]. $p(r_i)$ represents the

Precision at Recall r_i that was measured. The Precisionat each Recall level r_i is interpolated using the highest epochs Precision for which the corresponding Recall exceeds *r*.

After training for 250 epochs, we logged the training loss of our model. As it can be seen from the graph in Fig. 8, after approximately 30k iterations or 200 epochs of training, the training loss reaches a constant value which means the benefit of further training is negligible. We also evaluated the trained model using the Cascade R-CNN method on each test image. The model performed sufficiently well to detect all the wheat spikes. For evaluation, the following metrics were used for each image:

- $Precision = TP/(TP + FP)$ measures how many of the detections made are actual spikes.
- $Recall = TP/(TP + FN)$ measures how many actual spikes in the image are successfully detected.
- $F1\ Score = (2 * Precision * Recall)/(Precision + Recall)$ "is the weighted mean of precision and recall, it measures the model's robustness."

Here, true positive (TP)—when the model correctly detects a region as a spike, false positive (FP)—when the model incorrectly detects a background region as a spike or detects the same spike as multiple ones; and false negative (FN) —when the model incorrectly classifies an actual spike as background. In contrast, true negative (TN)—correctly classifying background which is always considered 'zero' and does not contribute to the model's performance evaluation.

Table 3 shows the counting result of the detected spikes by our proposed method for each test image. As it can be seen, our method achieved an average precision of 98% across all test images which indicates the model's ability to detect regions that contain wheat spikes. An average recall of 85% means that the model can detect most of the spikes present in the images. Our method also achieved a high average F1 score of 91% showing that the model is robust enough to detect spikes in different challenging conditions.

Table 3: Count and evaluation of spike detection using the Cascade R-CNN model tested on SPIKE test set (15images)

Image	GT	Detected	TP	FP	FN	Precision	Recall	Accuracy	F1-score
GSGC test2.jpg	72	68	67	1	5	0.99	0.93	92%	0.96
GSGC test4.jpg	76	69	68	1	8	0.99	0.89	88%	0.94
GSGC test5.jpg	62	56	56	0	6	1	0.9	90%	0.95
GSYC test199.jpg	78	70	68	2	10	0.97	0.87	85%	0.92
GSYC test220.jpg	83	73	70	3	13	0.96	0.84	81%	0.9
GSYC test242.jpg	78	68	64	4	14	0.94	0.82	78%	0.88
GSYC test320.jpg	84	71	69	2	15	0.97	0.82	80%	0.89
GSYC test383.jpg	77	67	65	2	12	0.97	0.84	82%	0.9
GSYC test417.jpg	88	74	73	1	15	0.99	0.83	82%	0.9
GSYC test421.jpg	79	72	72	0	7	1	0.91	91%	0.95
GSYC test437.jpg	72	65	64	1	8	0.98	0.89	88%	0.93
GSYC test480.jpg	91	77	74	3	17	0.96	0.81	79%	0.88
YSYC test1.jpg	85	72	69	3	16	0.96	0.81	78%	0.88
YSYC test3.jpg	71	59	59	0	12	1	0.83	83%	0.91
YSYC test6.jpg	63	53	52	1	11	0.98	0.83	81%	0.9
Total	1159	1014	990	24	169	-	-	-	-
Average	-	-	-	-	-	0.98	0.85	84%	0.91
Standard dev.	8.36	6.79	6.24	1.24	3.81	0.02	0.04	0.05	0.03

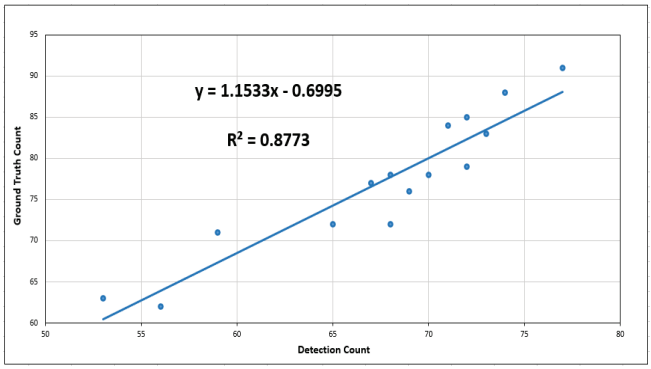

Fig. 9. Ground truth versus Detection count plot.

Fig. 9 shows the relationship between the ground truth count of spikes and the detected number of spikes by our method, for each of the 15 test images.In the graph, the horizontal axis represents the number of detected spikes by the proposed method and the vertical axisrepresents the number of manually counted spikes inthe test images. Our proposed method yields a close- to-perfect estimate of the number of spikes per image (the slope of the line is 1.1533, and the intercept is -0.6995). The model yields a high R2 value of 0.88, demonstrating a strong linear relationship between the manually counted ground truth and the detection outputsof our method.

In Fig. 10, it can be seen that our method can even detect spikes that are partially visible at the edges. But the method struggles to detect partially occluded spikes and achieved poor detection performance for spikes that are crowded together.

3.2 Comparison with Existing Method

We also compared our proposed method against a state-of-the-art method developed by Hasan et al. [3], which uses a Faster R-CNN architecture for their model, as well as another state-of-the-art method developed by Wen et al. [7] that utilised a model based on the RetinaNet architecture, and logged the loss metric and detection performance for comparison of these models. In Fig. 11, the blue line represents the loss of the proposed method and the red and magenta lines represent the training losses of the existing methods.

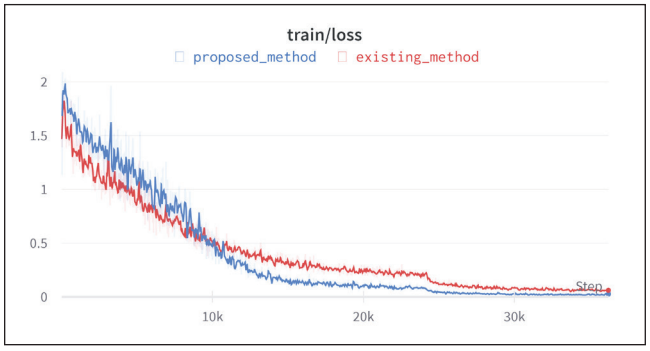

Fig. 11. Comparison of training loss over 250 epochs.

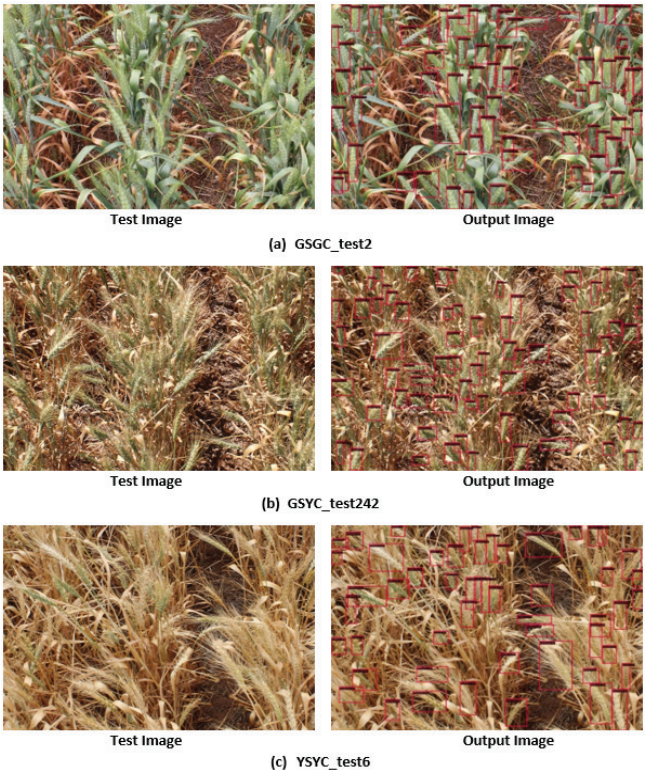

Fig. 10. Examples of the generated output image from a test image.

The X-axis indicates the number of training iterations and the Y-axis indicates training loss. As it can be seen from the graph, our proposed method is better at optimizing the loss function and resulted in a more optimized training loss value. Better optimization of the loss metric can lead to better detection performance and precision which can be seen in the graphs included inthe following figures.

In Fig. 12, the blue line represents the mAP50 of the proposed method and the red and magenta lines represent the mAP50 of the existing methods. The X-axis indicates a number of training iterations and the Y-axis indicate mAP50 of the validation set. Here mAP50 refers to mAP calculated at the IoU thresholdof 0.5. The graph indicates the improvement in the precision of our method. After just 69 epochs (10k iterations) of training, the proposed method surpassedthe Faster R-CNN's [3] detection performance in terms of mAP at IOU threshold of 0.5.

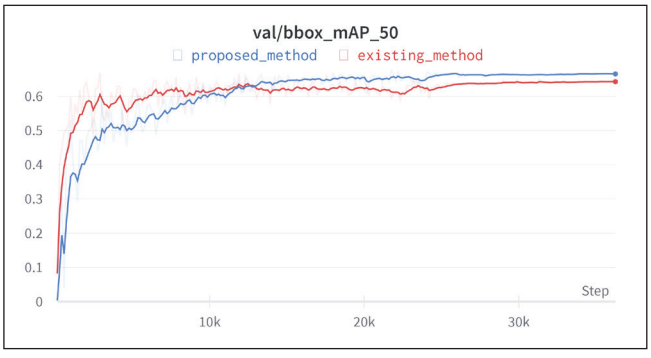

Fig. 12. Comparison of mAP50 of the validation set over 250 epochs.

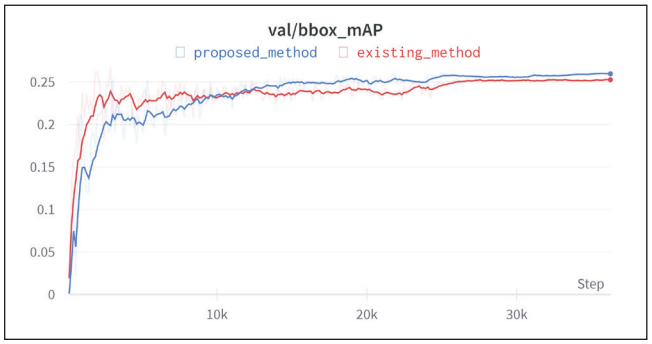

Fig. 13. Comparison of mAP of the validation set over 250 epochs.

In Fig. 13, the blue line represents the mAP of the proposed method and the red and magenta lines represent the mAP of the existing methods. The X-axis indicates a number of training iterations and the Y-axis indicate the mAP of the validation set. Here mAPis calculated over 10 levels of IoU threshold, [0.5: 0.05: 0.9]. Our proposed method has better localization performance than the existing method which contributes to higher mAP shown in the graph in Fig. 13. Similar to the graph in Fig. 12, it's also evident that the proposed approach started scoring better mAP value for the validation set after just a few epochs and ultimately resulted in a better score after 250 epochs of training.

Table 4 shows the exact values of the evaluation metrics after taking the best-performing model and testing it on the test set containing 15 images. Our proposed method achieved a higher mAP50, 2.9% better than Faster R-CNN [3] and 11.7% better thanRetinaNet [7]. It also resulted in 1.1% and 4.14% better Mean Average F1 scores which is calculated using the following equation,

$$Mean\ Average\ F1\ Score\ = \frac{2\ *\ mAP\ *\ AR@100}{mAP\ +\ AR@100}$$

We also manually counted the detection results to compare total detection, TP, FP, and FN values of the experimented methods as well as calculate average precision, recall, and accuracy metrics. As can be seen from Table V, our method can detect significantly more spikes present in test images. Also, a lower FN value means it can detect spikes that are otherwise difficult to detect for the existing method.

Our proposed method, utilizing the Cascade R-CNN architecture for wheat spike detection and counting, offers a significant improvement over some existing state-of-the-art methods such as the ones which employ the use of Faster R-CNN [3] and RetinaNet [7]architectures. As it can be seen from the results of our conducted experiments, our method scores an mAP of 0.672, which is 2.9% and 11.2% better than the existing standard methods developed by Hasan et al.[3] and Wen et al. [7], respectively. Also, when it comes to average detection accuracy, our method scores 84% (9% and 15% improvement over Faster R-CNN and RetinaNet, respectively) as well as a significant improvement in the average recall metric, 85% versus 77% and 72% compared to Faster R-CNN and RetinaNet, respectively. It also sees a 1% and 3% improvement in the average precision metric.

Analyzing the output images of our method, it can be observed that our proposed method offers an im-provement over the existing methods when it comes to detecting spikes that are partially visible in the images, especially at the edges. The use of multi-stage cascaded R-CNN architecture that utilizes a sequence of detectors trained with increasing IOU thresholds (0.5, 0.6 and 0.7 in our case) attributes to the superior performance of our proposed method among all the compared methods. Furthermore, careful fine-tuning of the model's hyperpa- rameters using the validation set contributed to the im- proved detection rate. However, it requires an exploration of other components in the network architecture, namelythe loss function, the region proposal network (RPN), and the non-maximum suppression (NMS) algorithm, to improve detection performance when it comes to detecting spikes that are partially occluded or crowded together.

Table 4: Comparison of evaluation metrics of experimented methods

Method	mAP [0.5: 0.05: 0.9]	mAP50	AR@100 [0.5: 0.05: 0.9]	Mean Average F1 score (%)
RetinaNet [7]	0.219	0.56	0.306	25.53
Faster-RCNN [3]	0.249	0.643	0.335	28.57
Proposed Method	0.257	0.672	0.351	29.67

Table 5: Comparison of counting performance of experimented methods

Method	Total					Average			
	GT	Detected	TP	FP	FN	Precision	Recall	Accuracy	F1 score
RetinaNet [7]	1159	865	825	40	339	0.95	0.72	69%	0.82
Faster R-CNN [3]	1159	911	886	25	273	0.97	0.77	75%	0.86
Proposed Method	1159	1014	990	24	169	0.98	0.85	84%	0.91

4. Conclusion

In this paper, we have proposed an improved method for detecting wheat spikes from land-based field images. Our proposed method uses the Cascade R-CNN architecture for detecting spikes from images that are captured in real-world field conditions. After many iterations of empirical testing, we fine-tuned our model's architecture and optimized the hyper-parameters to suit the need for our spike detection task. Thus, it offers a significant improvement over other existing standard approaches such as the one developed by Hasan et al [3] and Wenet al [7]. This allows us to more accurately detect and count spikes in challenging photographic scenarios such as partial occlusion, visibility, and crowded regions. Because our method makes use of a dataset derived from real-world land-based field imaging that contains complex scenarios, it can be used to detect spikes in other real-world scenarios as well. Furthermore, the improved performance of our method could be applied to applications that use different imaging systems, such as UAVs and satellites. Moreover, accurate detection and count of various phenotyping traits like wheat spike and spikelet, paddy and sorghum head is necessary as it helps smart farming and agricultural advisory applications to make more accurate decisions regarding crop breeding and management. Our research is a step towards betterment in this field of e-agriculture. Further opportunity for research includes but are not limited to training and testing the proposed method on various other datasets for wheat spike detection and conducting an ablation study to improve the method's accuracy and performance.

Acknowledgement

Plant phenotyping research is conducted in collaboration with the Phenomics and Bioinformatics Research Centre at the University of South Australia, Australia.

References

1. "The Big Idea – Phenotyping – Envision – College of Agriculture Magazine at Purdue University." https://shorturl.at/flpwL, 2017. [Online; accessed 23-June-2022].
2. M. P. Cendrero-Mateo, O. Muller, H. Al- brecht, A. Burkart, S. Gatzke, B. Janssen, B. Keller, N. Ko¨rber, T. Kraska, S. Matsubara, J. Li, M. Mu¨ller-Linow, R. Pieruschka, F. Pinto, P. Rischbeck, A. Schickling, A. Steier, M. Watt, U. Schurr, and U. Rascher, "Field Phenotyping," in *Terrestrial Ecosystem Research Infrastructures* (A. Chabbi and H. W. Loescher, eds.), pp. 53–81, Boca Raton, FL : CRC Press, 2017.: CRC Press, 1 ed., Mar. 2017.
3. M. M. Hasan, J. P. Chopin, H. Laga, and S. J. Miklavcic, "Detection and analysis of wheat spikes using Convolutional Neural Networks," *Plant Methods*, vol. 14, p. 100, Dec. 2018.
4. T. Misra, A. Arora, S. Marwaha, V. Chinnusamy, A. R. Rao, R. Jain, R. N. Sahoo, M. Ray, S. Ku- mar, D. Raju, R. R. Jha, A. Nigam, and S. Goel, "SpikeSegNet-a deep learning approach utilizing encoder-decoder network with hourglass for spike segmentation and counting in wheat plant from visual imaging," *Plant Methods*, vol. 16, p. 40, Dec. 2020.
5. H. Xiong, Z. Cao, H. Lu, S. Madec, L. Liu, and C. Shen, "TasselNetv2: in-field counting of wheat spikes with context-augmented local regression net- works," *Plant Methods*, vol. 15, p. 150, Dec. 2019.
6. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object De- tection and Semantic Segmentation," pp. 580–587, 2014.
7. C. Wen, J. Wu, H. Chen, H. Su, X. Chen, Z. Li, and C. Yang, "Wheat Spike Detection and Counting in the Field Based on SpikeRetinaNet," *Frontiers in Plant Science*, vol. 13, p. 821717, Mar. 2022.
8. J. Zhao, X. Zhang, J. Yan, X. Qiu, X. Yao, Y. Tian, Y. Zhu, and W. Cao, "A Wheat Spike Detec- tion Method in UAV Images Based on Improved YOLOv5," *Remote Sensing*, vol. 13, p. 3095, Aug. 2021.
9. S. Khaki, N. Safaei, H. Pham, and L. Wang, "WheatNet: A lightweight convolutional neural net- work for high-throughput image-based wheat head detection and counting," *Neurocomputing*, vol. 489, pp. 78–89, June 2022.
10. Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving Into High Quality Object Detection," pp. 6154–6162, 2018.
11. T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dolla´r, "Microsoft COCO: Common Objects in Context," Tech. Rep. arXiv:1405.0312, arXiv, Feb. 2015. arXiv:1405.0312 [cs] type: arti- cle.
12. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems*, vol. 28, Curran Associates, Inc., 2015.
13. T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariha- ran, and S. Belongie, "Feature Pyramid Networks for Object Detection," pp. 2117–2125, 2017.
14. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Resid- ual Learning for Image Recognition," Dec. 2015. Number: arXiv:1512.03385 arXiv:1512.03385 [cs].
15. J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards Balanced Learning for Object Detection," pp. 821–830, 2019.
16. K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open MMLab Detection Toolbox and Benchmark," *arXiv:1906.07155 [cs, eess]*, June 2019. arXiv: 1906.07155.