Deep learning for reliable identification and counting of wheat ears in robot images from field trials

Ehsan Ullah
Department of Computer Science,NTNU
ehsanu@stud.ntnu.no

Anushka Subedi
Department of Computer Science,NTNU
anushkas@stud.ntnu.no

Dipesh Pandey
Department of Computer Science,NTNU
dipeshp@stud.ntnu.no

***Abstract*—Number of spikes, spikelets per spike, number of spikes per square meter are some of the important metrics for plant breeders and researchers in predicting wheat crop yield. Evaluating the crop yield based on wheat ears counting is still done manually which is labor-intensive, tedious and costly task. Thus, there is a significant need of developing a real-time wheat spikes/ears counting system for plant breeders for effective and efficient crop yield predictions. In this paper, we proposed two deep learning based methods Faster R-CNN and EfficientDet for accurate and computationally efficient localization and counting of wheat spikes/ears in digital images taken using some high-throughput phenotyping techniques under natural field conditions. We used Faster R-CNN with Resnet50 as backbone architecture which produced an overall accuracy 88.7% on the test images. We also used the recent state of the art models EfficientDet-D5 and EfficientDet-D7 having a backbone architectures EfficientNet-B5 and EfficientNet-B7 respectively. The EfficientDet-D5 model produce an accuracy 92.7% on the test images and EfficientDet-D7 produce an accuracy 93.6%.**

***Index Terms*—Wheat Spikes, Deep Learning, Faster R-CNN, EfficientDet**

I. Introduction

One of the most important and widely utilized crop species which are consumed daily by the public. 762.7 million tons of annual wheat production was recorded by [2] in 2020. The Wheat is cultivated every year in around 215 million hectares and the global trade of wheat is estimated nearly 50 billion US Dollar every year [1]. It is estimated that nearly 750.1 million tons of wheat is consumed every year globally [2]. Every coming year the demand of grain is increasing and at the same time extreme weather situations and variation in climate changes increases the risk of uncertain supply of grains. Complex, multivariate, and unpredictable agricultural environments need to be better studied in order to solve these types of challenges by monitoring, measuring/analyzing and constantly evaluating different physical aspects and phenomena. This will help researchers and plant breeders to know and recognize better-yielding and more stress-tolerant plant species.

In recent studies, biologists and breeders rely more on high performance phenotyping techniques to measure the quantitative assessment of crop canopy characteristics [8]. Constraints in plant phenotyping are widely acknowledged as a vital limitation in genetic and plant breeding studies [5], [35]. Initial field-based high-throughput phenotyping technologies was based on direct sensor and image measurements to derive important morphological properties of interest like vegetation indexes from spectral reflectance data [5]. This first generation of High Throughput Phenotyping, while providing great intuition into plant processes, but it is limited in the evaluation of complex characteristics such as plant morphology or growth stage that cannot be assessed by a linear estimation of just pixel data. Although these complex morphological and developmental characteristics are easily distinguished by an eye, it is difficult to evaluate these phenotypes using high-throughput platforms, especially under field conditions used in plant breeding programs.

Using a method comparable to [24], most spikelet and ear counting is performed by hand to date, which relates crop yield to spike and spikelet characteristics without the use of image analysis. In several research problems related to plant phenotyping, conventional Machine Learning methods have been used widely. ML models including SVM, decision trees, Bayesian, and instance base model has been used in crop yield prediction, Disease Detection, Weed Detection, plant species detection, and in crop quality [17]. Some ML based techniques exist to automatically detect heading and flowering in wheat [32] to distinguish growth stages in field-grown wheat, a bag-of-visual-words method is used. Low level characteristics are collected using the SIFT algorithm. Finally, to classify the growth levels in plants, the classification of support vector machines is used.Symptoms of yellow rust disease and nitrogen stress were examined by using hyperspectral features from a five waveband of 20nm hyperspectral imaging system located on the ground [27]. Crop growth characteristics is measured based on-line multilayer soil data of satellite imagery, an unsupervised learning algorithm was used and field variations in wheat yield were predicted [26].

However, Smart agriculture and plant phenotyping have now progressed into the 'big data' era, where massive data is gathered from open field trials, indoor plant phenotyping using advanced platforms such as UAV, satellite imagery, grounded robot vehicles, gantries, etc. With the availability of large amount of data and recent high-end computing power of hardware [15]. Deep learning models are more preferred as its performance increases with the increase in the amounts of data we provide to the model. This is one of the main reason due to which, deep learning approaches took over the traditional machine learning approaches. Secondly, Deep learning surpass the need of manually selecting and defining handcrafted features [15]. Instead deep learning approaches perform optimization in a complete end-to-end way by mapping input data samples to outputs targets. Deep Learning has been applied to hundreds of problems over the past few years. Some of the greatest contributions of deep learning have been in the area of computer vision. It focuses on the interpretation of images and videos, and deals with tasks such as classification of objects, tracking, identification, and segmentation. Deep Learning has outperformed previous approaches used to address specific issues in many fields [25]. Deep Learning algorithms derive meaningful abstract representations of raw data with the use of a hierarchical multi-level learning approach, where more abstract and complex representations are learned at a higher level based on less abstract concepts and representations at the lower level(s) of the learning hierarchy [15]. The models learn to perform classification and detection directly using data in the form of images, text, multi/ hyperspectral data, sound etc. The detection of wheat heads from images in itself is a challenging task as it involves several factors to be taken into account like the observational conditions, genotypic differences and development stages of the plant. Wheat head density (the number of wheat heads per unit ground area) is a major yield component, but because the process of evaluation of this parameter is still manual and labor-intensive, measurement errors of around 10% can be observed. [24] [8] Thus, developing automated image-based methods that can bring this error down is important so that breeders can manipulate the balance between yield parameters in their breeding selections. In this project, we use the Global Wheat Detection Dataset (GWHD) [3] which contains images taken at 90 degree from above of a wheat field with the wheat head annotated using bounding boxes. These images contains occlusions, overlapped wheat ears, blurred background etc which makes it a perfect dataset for training any deep learning model. We used two different deep learning models, Faster-RCNN [31] and EfficientDet [34] for detection of wheat ears and trained them with Global wheat head dataset. The main objective of this study was to build a data driven efficient system which will detect the wheat ears with good performance and accuracy. We also had another unlabelled dataset from NMBU which contained at least 5000 high-resolution RGB raw images during the start of the project. We used a subset of those images in the testing phase to see how our model performs for those images.

The rest of the paper is organized as follows. It will start with the overview of object detection in general and then it will discuss about the previous research work done in wheat spikes/ears phenotyping using deep learning. It is followed by our proposed deep learning models, data preprocessing, model architecture, training, and evaluation methods. Finally, in the last section, we present our results based on our trained models before concluding in the last section.

II. Background

The purpose of generic object detection is to locate and identify current objects in any single image and to mark them with rectangular bounding boxes to demonstrate the confidences of existence. Detection of objects is a method involving classification as well as localization. There are many different object detection models, such as Faster R-CNN, Single shot detectors (SSD) [22], RetinaNet [19], YOLO [30], and the recent state of the art method EfficientDet. These models have different ways to support the detection process, and the differences between how these models working impacts on the performance. Models in the R-CNN [7] family are all region based. The detection happens in two stages. First, the model proposes a set of regions of interests by select search or regional proposal network. The proposed regions are sparse as the potential bounding box candidates can be infinite. Then a classifier only processes the region candidates. The second approach skips the region proposal stage and runs detection directly over a dense sampling of possible locations. In order to solve the critical problem of detecting wheat ears in open field environments researchers have already benefited by using deep learning detection methods.

[24] proposed two different approaches in his study, either using the Faster R-CNN object detector or with the TasselNet local count regression network for detection of wheat ears. Both approaches performed very well giving rRMSE approximately 6%. Faster-RCNN was however, more robust when applied to a dataset collected at a later stage with ears and context showing a different feature due to the higher maturity of the plants. [29] used a stacked hourglass encoder and decoder network, with fully-connected layers and residual blocks for the detection of spikes and spikelets. The architecture of the network was based upon an encoding/decoding structure, in which a series of convolutional operations and spatial downsampling begin by computing a fixed-size feature representation of the image. This feature space is then upsampled back to the original resolution, while lower-level features are re-combined in stages [29].The heatmap output at the end of each hourglass is used to calculate a loss, which guides training of the network. Spikes are located with an F1 score of 0.83 @ 0.1 and 0.89 @ 0.2, spikelets are located with an F1 of 0.88 @ 0.05 and 0.96 @ 0.1 [29]. [8]adapt, train and apply a variant of CNN, hereinafter referred to as Region-based Convolutional Neural Networks (R-CNN), to accurately count wheat spikes in images acquired using land-based RGB imaging platform. Three different types of dataset (Green spike and Yellow Canopy), (Green Spike and Green Canopy), (Yellow Spike and Yellow canopy) were chosen for training the model individually. The datasets were built according to the difference in color of spikes and the wheat canopy in order to get a considerable amount of contrast between wheat spikes and other part of the canopy. The best performing model produced an average accuracy 93.4% and F1 score of 0.95, respectively, when tested on 20 images [8]. Several studies have developed methods for wheat head detection from high-resolution RGB imagery based on machine

and deep learning algorithms. However, these methods have generally been calibrated and validated on limited datasets [3]. Studies like [8], [24], [29] Perform detection on limited set of wheat ears data. [29] perform detection using ACID dataset having 520 images. [8] Perform detection on wheat image dataset having 305 images taken from ground-based vehicle and similarly [24] used 236 high resolution images to conduct his study. High variability in observational conditions, genotypic differences, development stages, and head orientation makes wheat head detection a challenge for computer vision. Further, possible blurring due to motion or wind and overlap between heads for dense populations make this task even more complex [3]. A large, diverse, and well-labelled dataset is necessary to effectively detect the wheat spikes in the wheat images. Recently Through a joint international collaborative effort, A massive dataset is built called Global Wheat Head Detection. It contains 4700 high resolution RGB images and 190000 labelled wheat heads collected from several countries around the world at different growth stages with a wide range of genotypes [3]. The GWHD dataset is publicly available at http://www.globalwheat.com/and aimed at developing and benchmarking methods for wheat head detection.

III. Controlled vs Uncontrolled Environment

Previous Research about image-based wheat phenotyping was performed in two types of environments. In Controlled Environment the phenotyping was done specifically in small pot indoor wheat plots, glass, and greenhouses [3]. Here [16] the growth of a wheat plants is measured for a month by putting the wheat plants in an individual small pots to measure the detailed morphological properties of the wheat plant. The purpose of growing wheat plant in the pot was to reduce the overlapping of spikes. On the other hand, for measuring detailed geometric properties, such as the numbers of awns and wheat ears of plants. The plants were grown in a small pot in a glasshouse [17]. Some studies were carried out using plants grown in large indoor bins of size [120 x 80] cm, 96 plants are grown in a raster with 10 rows of eight plants in each bin [18] The plants are grown closer to field-like conditions and not individually in pots, but the analysis was still carried out in a regulated environment. In this study [19] wheat plants were grown in pots in the climate controlled greenhouse. A uniform background was maintained to increase the accuracy of separation between background and plant regions.

However, analysis of the growth behavior and physical characteristics of single plants grown in small pots is not generally sufficient [20]. One major factor explaining this difference is that under field conditions, plants are subjected to additional pressures, such as competition with neighbors and light source, weeds, water, and nutrients. In Uncontrolled environment, the study of image-based wheat phenotyping is generally performed by considering the realistic environmental conditions where occlusion, overlap, lighting conditions, shadows, and noise etc. is present. Plants quantitative properties are analyzed and measured under realistic environmental conditions. [3] Performed study in uncontrolled environment on 10 wheat varieties subjected to three different fertilizers to analyze it effect on the wheat varieties. In this study [21] the experiment was conducted in two open fields. Six wheat varieties were sown with plot sized of (3 x 1 and 2 x 1)m and nitrogen treatment were applied to analyze their effect. 12 wheat plots of winter wheat whose size was 5 x 2.4m was sown using three different target densities. Same amount of fertilizer was applied to each target to the traits of wheat species [22]. A trial of 120 microplots of 2.0mwidth by 10m long was considered to study the irrigation, nitrogen fertilization and water stress [23].

IV. Proposed Models

In this paper, Along with Faster R-CNN for detecting wheat ears, we also used the recently published state of the art deep learning model purposed by google brain researchers called EfficientDet which has a robust backbone architecture called EfficientNet. [33].

A. Faster R-CNN

Faster R-CNN, developed by Ren et al [31] is an object detection network composed of a feature extraction network which is typically a pre-trained CNN. It consists of two networks: a regional proposal network(RPN) for generating region proposals and a convolutional network which takes the proposed regions to detect objects almost in real-time. Thus, in addition to convolutional neural network, Faster R-CNN has a RPN which is inserted after the last convolutional layer making it different from its predecessors. RPN efficiently predicts region proposals with a wide range of scales and aspect ratios.

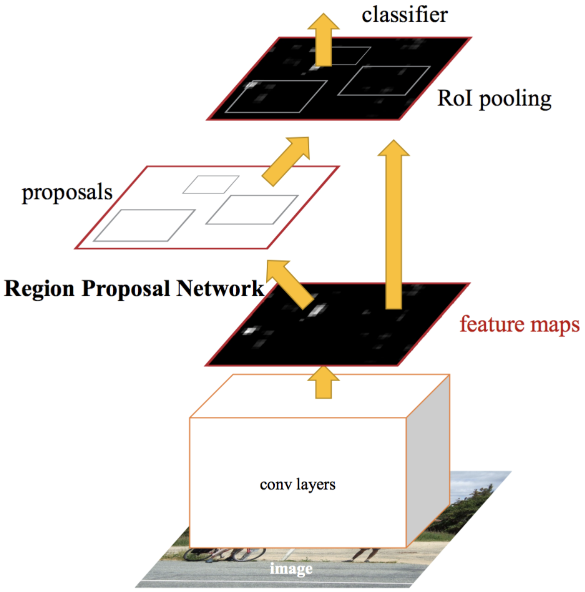

Fig. 1. Basic Architecture of a Faster R-CNN detector

Faster R-CNN is a detector that is learned end to end unlike the earlier variants of the region based detectors that relied on selective search. [36] This is important because the quality of the predictions depend upon the quality of the region proposals. Due to the fast processing capability and a better recognition rate of Faster R-CNN than other region based models, we chose it as one of the methods to detect wheat heads. The basic architecture of a Faster R-CNN detector is represented in Fig 1.

The input to the network are images of size Height x Width x Depth tensors, which are usually passed through a pre-trained CNN, also called backbone, producing a convolutional feature map. In deeplearning, this process is usually achieved by a process called transfer learning, where the weights of a model trained in a huge dataset are used for training a classifier on a smaller dataset.

a) Backbone: The backbone component of Faster R-CNN is the part where transfer learning comes into play. In other words, we use a pretrained Resnet-50 [9] architecture trained on ImageNet dataset for the feature extraction [14]. The output of this component is a set of feature maps, which are learned using a CNN instead of using selective search. These network architectures have been getting better over the years, with increasing number of layers, as well as the number of parameters. MobileNet [10], for example has approximately 3.3M parameters while Resnet50(50 layers) has approximately 23M parameters. In recent years, newer architectures like DenseNet [11] are also focusing on improving results while lowering the number of parameters. In our case, we used the Resnet50 architecture as the backbone.

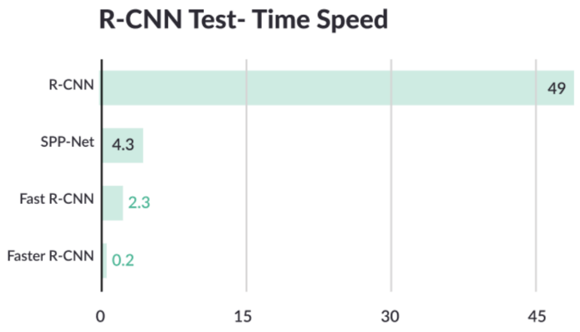

Fig. 2. Comparison of test time among R-CNN models

b) Anchors: Once we get the feature map from the backbone network from the above step, we need to find the regions of interest for classification. This is where anchors come into play. They are used to solve the variable length problem: the idea that there could be bounding boxes of variable shapes and aspect ratios in our images. In this step, we choose a set of anchors as a set of sizes(e.g. 64px, 128px, 256px) and a set of width to height ratios(e.g. 0.5, 1, 1.5). We then finally use all the possible combinations of these sizes and ratios before passing them to the RPN component of the system. For a convolutional feature map of a size W x H, there are WHk anchors in total. [31] In our setup, we used anchors of the following settings, as provided by the Pytorch [28] official implementation of Faster R-CNN:

$$sizes = (32px, 64px, 128px, 256px, 512px)$$
$$aspectratios = (0.5, 1.0, 2.0)$$

c) Region Proposal Network: After the regions of different shapes and sizes are received in the RPN, the next step is to pick the bounding boxes that are actually the correct ones.

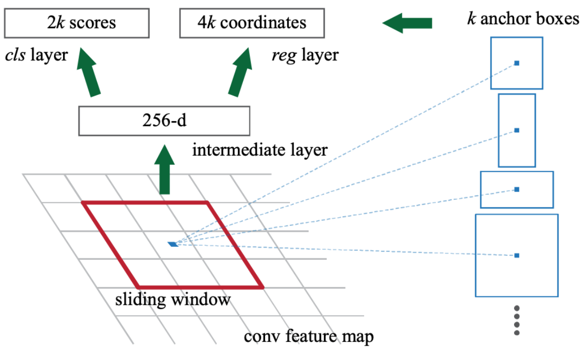

Fig. 3. Region Proposal Network in Faster R-CNN

Fig. 2. shows a basic implementation of the region proposal network. Starting with the convolutional feature map, at the last layer of the CNN, a 3x3 sliding window moves across the feature map and maps it to a lower dimension (e.g. 256-d as shown in the figure). Then, for each sliding-window location, an anchor generator is employed that generates all the possible regions based on the aspect ratios and sizes, as mentioned in the section above. Finally, each region proposal contains an 'objectness' score for the given region, as well as 4 coordinates representing the bounding box location of that region. For each anchor box, the objectness score is checked against a threshold. If the objectness is above the threshold, the box's coordinates get passed forward, else gets discarded.

d) Identify Object Label and Position: Once we have the filtered region proposals, we feed them into another network that resembles a Fast R-CNN. Adding a pooling layer in addition to some fully-connected layers, a softmax classifier is used to classify the objects into N classes in one go. Bounding box-regressors are used to tighten the bounding box of the object. In our case, we just have two classes: one the wheat head, and another the background.

B. EfficientDet

EffificientDet consists of three parts. As shown in Figure 4, The first part is the pre-trained EfficientNet as the backbone architecture of the model. The second part is BiFPN, which do the top-down and bottom-up feature fusion multiple times for the output characteristic of Level 3-7 in EfficientNet. The third part is the classification and detection box prediction network, to regress and classify the wheat ear frame. In deep

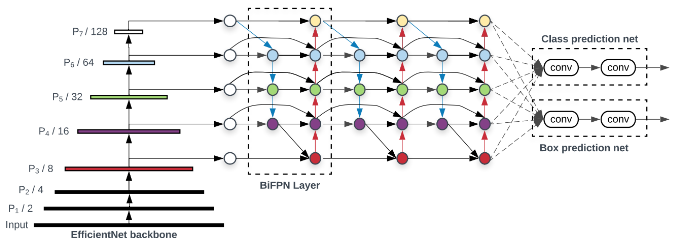

Fig. 4. Architecture of EfficientDet

learning paradigm, models are scaled with one main goal of increasing the model accuracy. For example the ResNet architecture purposed by [9] can be scaled up to ResNet-200 from ResNet-18 base line architecture. Similarly, in this study [13] they used parallelism mechanism where different chain of layers were pipe-lined on separate accelerators. which enable of scaling a variety of different networks to huge sizes efficiently. Using this technique, they train AmoebaNet with 557 million parameters on ImageNet [14] dataset and achieve a top-1 accuracy of 84.4% by making the AmoebaNet baseline architecture 4 times larger. The features extraction model (ConvNets) can be scaled by increasing the model layers (depth) as mentioned in this study [9] or as discussed in this study [38] where they decrease the network model depth by reducing the number of layers and increases the model width. Some of the research is also done on by scaling the model based on spatial resolution or size of the image as discussed in this study [12].

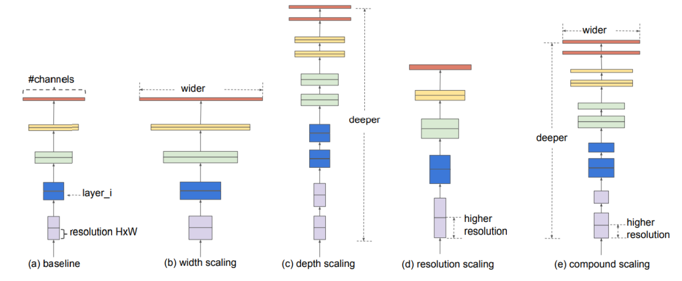

Fig. 5. EfficientNet Compound Scaling

Google Research, Brain Team recently proposed a new method of compound scaling where the network model called EfficientNet which can be scaled simultaneously on all three dimensions height, width and resolution of the image while staying within the constraints of target memory and target FLOPs as shown in figure 5. EfficientNets outperform most of the widely used convolutional neural networks. EfficientNet-B7 surpasses the best existing GPipe accuracy [13], but utilizing 8.4 times fewer parameters and running 6.1 times faster on inference [33]. Compared to the widely used ResNet-50 [9], EfficientNet-B4 attain the top-1 accuracy from 76.3% to 83.0% with similar floating point operation per second(FLOPS) [33]. Besides ImageNet, EfficientNets also performed well on other dataset as well and achieve state of-the-art accuracy in most of the widely used datasets, while restricting the parameters by up to 21 times than existing Convolutional neural networks [33]. Due to all these results, we chose efficientNet as our backbone feature extractor model.

Feature fusion seeks to combine representations of a given image at different resolutions. Typically, the feature fusion networks uses the last few feature layers from the CNN. The Conventional and mostly used top down FPN is inherently limited due to the flow of information in only one-way [18]. PANet [21] address this issued by adding an extra bottom-up path information flow, as shown in Figure 6(b). Recently, NAS-FPN [6] employs neural architecture search to search for better cross-scale feature network topology but it requires thousands of GPU hours during search and the found network is irregular and difficult to interpret or modify, as shown in Figure 6(c). For this part the EfficientDet proposed BiFPN feature fusion, as shown in figure 6(d). which is a multi-scale feature fusion mechanism of combining features at different layers of the backbone architecture. Here BIFPN do the top-down and bottom-up feature fusion multiple times using the output features of Level 3 to Level 7 from EfficientNet [34].

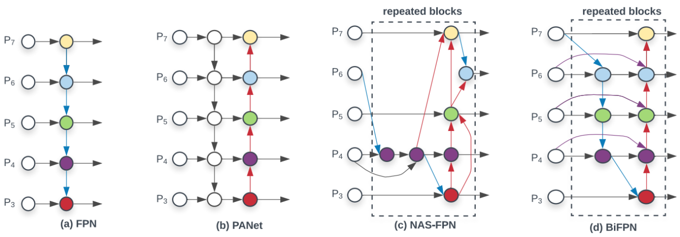

Fig. 6. Feature network design – (a) FPN introduces a top-down pathway to fuse multi-scale features from level 3 to 7 (P3 - P7); (b) PANet adds an additional bottom-up pathway on top of FPN; (c) NAS-FPN use neural architecture search to find an irregular feature network topology and then repeatedly apply the same block; (d) BiFPN top-down and-bottom up approach with better accuracy and efficiency trade-offs.

For optimization, the BiFPN first discard all those feature maps which has only one input. The reason for those feature map is that it contribute less to the overall fusion of different features. Switch connection is built between output node and input feature network layers of the same level for fusing more features to the end feature node with out increasing the complexity and cost. According to the authors they are considering each BiFPN (top-down and bottom-up) path as one feature network layer and it can be used multiple times for more robust and high level feature fusion.

A common practice of fusing features with different resolutions is first resizing the features and then summing them all together. Pyramid attention network [16] introduces Global Attention upsampling module for exploiting high-level feature map to guide low-level features recovering pixel localization similar to this study [6]. All previous methods treat all input features equally without distinction. Different input features are at different resolutions, they usually contribute to the output feature unequally. For network to learn the importance of each input feature map they [34] assigned additional weight to the input. And come up with three different weighted fusion approaches Unbounded Fusion, Softmax based fusion, Fast normalized fusion. The third part is the classification and detection box prediction network, to regress and classify the wheat ear from the images. Based on the above amazing improvement they develop a group of EfficeintDet models with the aim of optimizing both accuracy and efficiency and meet a wide range of resource constraints. With single model and single-scale, EfficientDet-D7 achieves state of-the-art 55.1 AP on COCO test-dev with 77M parameters and 410B FLOPs1 , being 4x – 9x smaller and using 13x – 42x fewer FLOPs than previous detectors [34].

V. Our Work

To perform the wheat head detection and counting, we followed three steps: starting with the exploratory data analysis and preprocessing, followed by training the deeplearning models, and finally using several evaluation metrics to evaluate the results.

A. Data Analysis and Preprocessing

In detecting objects of interest, such as wheat spikes, ambient noise poses significant challenges for computer vision-based techniques. Some challenges include the following: The movements of plants and/or the stability of handheld cameras are likely to cause blurred images.
Due to natural conditions and light variations in the field, dark shadows or sharp brightness can appear in images.
Overlaps between the ears due to the floppy attitude of the ears can also give rise to additional difficulties especially with the presence of awns in certain cultivars.
Over development phases, spikes in various varieties change dramatically, as spikes display no correlation between the early and later growth phases.
Preprocessing is a preliminary phase in the analysis of images, which helps to arrange data properties in order to enable subsequent steps and also to achieve fair final results. At first, the GWHD dataset was analysed. The dataset is gather from several parts of the world, with a total of 4698 squared patches extracted from the 2219 original high-resolution RGB images. It contains 188,495 labelled heads with an average of 20 to 60 heads per image. There are also around 100 images that don't contain any heads to represent actual capturing conditions and make the task more difficult. We tried several data augmentation techniques to improve the performance of our models. In addition to the usual data augmentation methods employed in normal computer vision tasks along with other transform methods, the ones used in our approach were horizontal/vertical flips, cropping and resizing, change to gray, cutout [4], cutmix [37], hue/saturation value changes, and brightness/contrast changes. The data augmentation and image preprocessing will help in producing more samples and variations and help in training the models to decrease overfitting and increase the generalization of our models.

B. Training

1) Faster R-CNN: For training, we used a normal simple random sampling from the dataset we obtained from the above step. We used 80%-20% splitting for training set and validation data. Initially, we started from the pre-trained model on the pedestrian images and did some fine-tuning to adapt to our use case.

We studied the results of the model using a Resnet50 backbone, learning rate of 0.005 and CosineAnnealing scheduler [23] with Stochastic Gradient Descent (SGD) as the optimizer. The main reason behind using a cosine function for the learning function is the idea that for each batch of the SGD, the network should get very close to the global minimum value for the loss, means we don't want the algorithm to overshoot and the learning rate should get smaller helping the loss value settle to some point. Cosine annealing decreases the learning rate following the cosine function and helps in making this global minimum stable.

We also tried the Adam optimizer and tried to see how it performs in comparison to SGD. Using these parameters, we trained the model for 40 epochs with the batch size of 8.

The results are represented in the plots below.

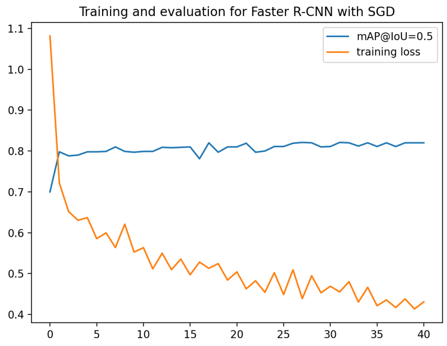

Fig. 7. Faster R-CNN training and evaluation with SGD

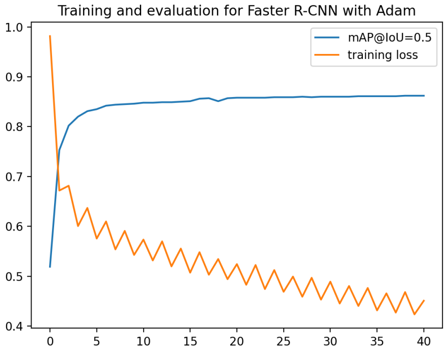

Fig. 8. Faster R-CNN training and evaluation with Adam

2) EfficientDet: EfficientDet- D5 and EfficientDet-D7 were used as our detection models for detecting wheat ears effectively. We used wheat ears GWHD dataset with images of 15 different wheat varieties captured under different environment conditions. We utilized all those images for training and validation of the model. In this study, EffificientDet-D5 and EffificientDet-D7 were trained respectively. We used Pytorch framework version 1.6.0 and Python 3.7. we use the CUDA/10.0.130 version for graphics cards. we trained our model using Idun high performance computing cluster at NTNU Trondheim using only one GPU which was NIVIDIA V100 Tensor Core. The images with input size of 512x512 was introduced to the model and the model is trained for 40 Epochs. The Average loss error on both of the model is saved and its shown in the below figures 9, 10.

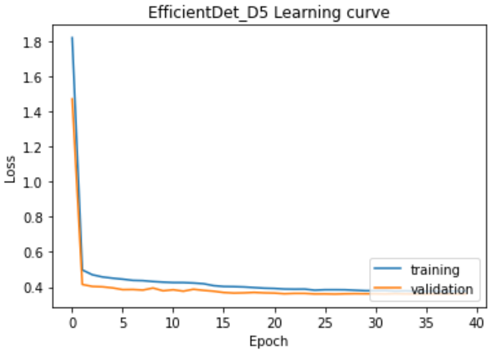

Fig. 9. EfficientDet-D5 training and loss error

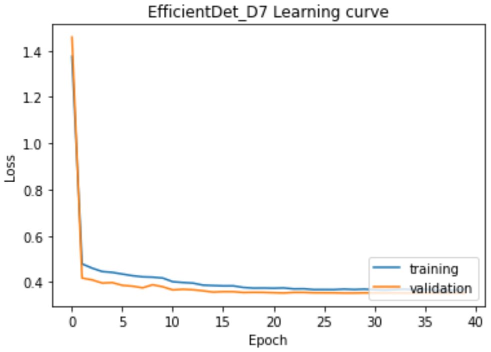

Fig. 10. Efficient-D7 training and loss error

VI. Evaluation and Results

For the evaluation of the results, we used the training error along with the mean average precision (mAP) from the standard MS COCO metrics [20] for the validation set. The mAP values relies on the Intersection over Union(IoU) values. The IoU value is the area of intersection between the actual bounding box divided by their union's area. A True Positive prediction is the one with IoU $>$ threshold, whereas False Positive refers to one with IoU $<$ threshold.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{1}$$

Similarly, we used the regular precision, recall, and accuracy for the test set.

1) Precision and Recall: Precision is the ratio between true positives and all positives, whereas recall is the measure of a model identifying true positives.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\#\text{ ground truths}} \tag{2}$$

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

2) Accuracy: Accuracy, the simplest of the metrics, is the ratio of total number of correct predictions to the number of predictions.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FP} + \text{FN}} \tag{4}$$

We used 10 test images that were not used during the training or evaluation phase and calculated the above metrics.We achieved overall 87.4% accuracy using Faster-RCNN with SGD optimizer and 88.7% accuracy using Adam optimizer. For EfficeintDet Models, The EfficientDet-D5 achieved overall accuracy of 92.7% . EfficientDet-D7 produce better results than Faster-RCNN with Resnet as its backbone architecture and EfficientDet-D5.The EfficeintDet-D7 model achieved 93.6% accuracy on the test images. The results are represented in tables below.

TABLE I
Faster R-CNN with SGD Optimizer on Test Data

ImageID	GT	Detected	Precision	Recall	Accuracy
2fd875eaa	27	24	1.0	0.89	88.9%
51b3e36ab	27	29	0.86	0.93	80.6%
51f1be19e	18	18	1.0	1.0	100.0%
53f253011	31	29	1.0	0.94	93.5%
348a992bb	37	36	0.97	0.95	92.1%
796707dd7	31	23	1.0	0.74	74.2%
aac893a91	24	21	0.95	0.83	80.0%
cb8d261a3	24	21	1.0	0.88	87.5%
cc3532ff6	26	29	0.9	1.0	89.7%
f5a1f0358	28	31	0.9	1.0	90.3%
Total	273	261	0.95	0.91	87.4%

TABLE II
Faster R-CNN with Adam optimizer on Test Data

ImageID	GT	Detected	Precision	Recall	Accuracy
2fd875eaa	27	24	1.0	0.89	88.9%
51b3e36ab	27	29	0.9	0.96	86.7%
51f1be19e	18	18	1.0	1.0	100.0%
53f253011	31	29	1.0	0.94	93.5%
348a992bb	37	36	0.97	0.95	92.1%
796707dd7	31	25	1.0	0.81	80.6%
aac893a91	24	21	0.95	0.83	80.0%
cb8d261a3	24	21	1.0	0.88	87.5%
cc3532ff6	26	29	0.9	1.0	89.7%
f5a1f0358	28	31	0.9	1.0	90.3%
Total	273	263	0.96	0.92	88.7%

GT: The number of ground truth wheat heads

TABLE III
EfficientDet-D5 results on Test Data

Precision, Recall and Accuracy of the EfficientDet-D5 Model					
ImageID	GT	Detected	Precision	Recall	Accuracy
2fd875eaa	27	24	0.88	0.88	88%
53f253011	31	30	0.96	0.96	96%
51b3e36ab	27	25	0.92	0.92	92%
51f1be19e	18	18	1.0	1.0	100%
348a992bb	37	38	0.97	1.0	97%
796707dd7	31	26	0.83	0.83	83%
aac893a91	24	19	0.79	0.79	79%
cb8d261a3	24	24	1.0	1.0	100%
cc3532ff6	26	25	0.92	0.96	92%
f5a1f0358	28	28	1.0	1.0	100%
Total&Results	273	257	92.7%	93.4%	92.7%

TABLE IV
EfficientDet-D7 results on Test Data

Precision, Recall and Accuracy of the EfficientDet-D7 Model					
ImageID	GT	Detected	Precision	Recall	Accuracy
2fd875eaa	27	24	0.88	0.88	88%
53f253011	31	30	0.96	0.96	96%
51b3e36ab	27	25	0.92	0.92	92%
51f1be19e	18	18	1.0	1.0	100%
348a992bb	37	35	0.94	0.94	94%
796707dd7	31	26	0.83	0.83	83%
aac893a91	24	21	0.87	0.87	87%
cb8d261a3	24	24	1.0	1.0	100%
cc3532ff6	26	25	0.96	0.96	96%
f5a1f0358	28	28	1.0	1.0	100%
Total&Results	273	256	93.6%	93.6%	93.6%

VII. Conclusion

Agriculture plays a critical role in the global economy. Pressure on the agricultural system will increase with the continuing expansion of the human population. Digital Agriculture or precision farming, have arisen as new scientific fields that use data intense approaches to drive agricultural productivity while minimizing its environmental impact. The data generated in modern agricultural operations is provided by a variety of different sensors that enable researcher in better understanding of the morphological properties of the crops which leads to more accurate and faster crop yield predictions. In this study we use a data driven deep learning approach for accurate identification and counting of wheat ears/spikes in digital images taken in open field environment. We used two variants of Faster-RCNN, EfficientDet-D5 and EfficientDet-D7 for detecting the target ears/spikes in the wheat crop images where we achieved an accuracy of 88.7% using Faster-RCNN, 92.7% accuracy on EfficientDet-D5 and 93.6% accuracy on efficientDet-D7 respectively. In future the proposed models accuracy can be improved by training these models using high resolutions like 1280x1280 and 1536x1536 instead of images with 512x512 resolutions because these models EfficientDet-D5 and EfficientDet-D7 are pre-trained on the above respective resolutions. The other factor which will improve the accuracy of the models is to introduce less occlusions and background blur in the images.

References

[1] CGIAR wheat in the world. https://wheat.org/wheat-in-the-world/. Accessed: 05-11-2020.

[2] FAO world food situation. https://www.fao.org/worldfoodsituation/csdb/en/. Accessed: 05-11-2020.

[3] Etienne David, Simon Madec, Pouria Sadeghi-Tehran, Helge Aasen, Bangyou Zheng, Shouyang Liu, Norbert Kirchgessner, Goro Ishikawa, Koichi Nagasawa, Minhajul Arifin Badhon, et al. Global wheat head detection (gwhd) dataset: a large and diverse dataset of high resolution rgb labelled images to develop and benchmark wheat head detection methods. *arXiv preprint arXiv:2005.02162*, 2020.

[4] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout, 2017. http://arxiv.org/abs/1708.04552 arXiv:1708.04552.

[5] Robert T Furbank and Mark Tester. Phenomics–technologies to relieve the phenotyping bottleneck. *Trends in plant science*, 16(12):635–644, 2011.

[6] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7036–7045, 2019.

[7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[8] Md Mehedi Hasan, Joshua P Chopin, Hamid Laga, and Stanley J Miklavcic. Detection and analysis of wheat spikes using convolutional neural networks. *Plant Methods*, 14(1):100, 2018.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. http://arxiv.org/abs/1704.04861 arXiv:1704.04861.

[11] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018. http://arxiv.org/abs/1608.06993 arXiv:1608.06993.

[12] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

[13] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in neural information processing systems*, pages 103–112, 2019.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[16] Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid attention network for semantic segmentation. *arXiv preprint arXiv:1805.10180*, 2018.

[17] Konstantinos G Liakos, Patrizia Busato, Dimitrios Moshou, Simon Pearson, and Dionysis Bochtis. Machine learning in agriculture: A review. *Sensors*, 18(8):2674, 2018.

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. http://arxiv.org/abs/1405.0312 arXiv:1405.0312.

[21] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[23] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017. http://arxiv.org/abs/1608.03983 arXiv:1608.03983.

[24] Simon Madec, Xiuliang Jin, Hao Lu, Benoit De Solan, Shouyang Liu, Florent Duyme, Emmanuelle Heritier, and Frederic Baret. Ear density estimation from high resolution rgb imagery using deep learning technique. *Agricultural and forest meteorology*, 264:225–234, 2019.

[25] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1, 2015.

[26] Xanthoula Eirini Pantazi, Dimitrios Moshou, Thomas Alexandridis, Rebecca L Whetton, and Abdul Mounem Mouazen. Wheat yield prediction using machine learning and advanced sensing techniques. *Computers and Electronics in Agriculture*, 121:57–65, 2016.

[27] Xanthoula Eirini Pantazi, Dimitrios Moshou, Roberto Oberti, Jon West, Abdul Mounem Mouazen, and Dionysios Bochtis. Detection of biotic and abiotic stresses in crops by using hierarchical self organizing classifiers. *Precision Agriculture*, 18(3):383–393, 2017.

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. http://arxiv.org/abs/1912.01703 arXiv:1912.01703.

[29] Michael P Pound, Jonathan A Atkinson, Darren M Wells, Tony P Pridmore, and Andrew P French. Deep learning for multi-task plant phenotyping. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2055–2063, 2017.

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[32] Pouria Sadeghi-Tehran, Kasra Sabermanesh, Nicolas Virlet, and Malcolm J Hawkesford. Automated method to determine two critical growth stages of wheat: heading and flowering. *Frontiers in Plant Science*, 8:252, 2017.

[33] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[34] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–10790, 2020.

[35] Mark Tester and Peter Langridge. Breeding technologies to increase crop production in a changing world. *Science*, 327(5967):818–822, 2010.

[36] Jasper Uijlings, K. Sande, T. Gevers, and Arnold Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104:154–171, 09 2013. https://doi.org/10.1007/s11263-013-0620-5 doi:10.1007/s11263-013-0620-5.

[37] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019. http://arxiv.org/abs/1905.04899 arXiv:1905.04899.

[38] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.