Wheat Head Detection Using Deep Learning Techniques

Dhruvisha N. Jaiswal[a], Kirtikumar J. Sharma[b], Prashant B. Swadas[c]

[a]*PG Student, Computer Engineering, BVM Engineering College, Vallabh Vidyanagar, Anand - 388120, Gujarat, India*
[b]*Assistant Professor, Department of Computer Engineering, BVM Engineering College, Vallabh Vidyanagar, Anand - 388120, Gujarat, India*
[c]*Associate Professor, Department of Computer Engineering, BVM Engineering College, Vallabh Vidyanagar, Anand - 388120, Gujarat, India*

Abstract

This article aims to study an object detection methodology applied to the Global Wheat Head Detection (GWHD) Dataset. It is important to estimate the density and size of the wheat head from images or manual surveys. This study has been through four dominant architectures of object detection, namely Faster R-CNN, EfficientDet, Detectron2, and YoLov5 to design a novel and powerful wheat head detection model. Model exploration is tuned on optimizing the performance of models. Furthermore, also studied extensive exploratory data analysis, data cleaning, and data splitting and adapted the best data augmentation techniques to the required context. Additionally deliberated semi-supervised learning, specifically pseudo-labeling, to optimize previous supervised object detection models. Besides this study focus on ensemble learning, test time augmentation, bootstrap aggregating, and multi-scale ensemble to reach higher performance. Finally, for the post-processing technique, execute weighted box fusion.

Keywords: Wheat Head Detection, Data Augmentation, Test Time Augmentation, Weighted Boxes Fusion, Faster-RCNN, EfficientDet, Detectron2, YoLov5

1. Introduction

Wheat is the most cultivated cereal crop in the world, and India is the second largest producer in the world [1]. Object Detection deals with detecting instances in images. It has demonstrated the automation of laborious and tedious tasks. The detection of wheat heads in plant images is a crucial piece of work for estimating relevant wheat traits, including wheat head population density. Due to the large amount of data in computer vision applications, object detection requires architectures such as deep learning mode. This article's main goal is to develop a reliable object detection model that can identify wheat heads in an image with numerous wheat heads and generalize to all varieties of wheat crops from around the world. As a result, detection models created for wheat head detection need to be reliable in various growing environments. Many strategies, including pseudo labelling, test time extensions, bagging, multi-scale ensembles, and post-processing algorithms, were combined to improve the performance of detection models [2].

2. Related work

In general, deep learning algorithms strive to learn hierarchical features, analogous to different levels of abstraction [2]. In this section, object detection using deep learning techniques is covered, followed by an explanation of some research on wheat head detection. Wu Wei et al. [3] proposed wheat head detection and enumeration of wheat grains based on a deep learning method and models under several scenarios and scales, which gives us knowledge about wheat grains. According to this study, the number of grains played a crucial role in influencing production. R-CNN is used by authors with a loss under 0.5 and mAP:0.9 to recognize wheat head grains more quickly. The head is detected in less than two seconds. The model is robust to dissimilar backgrounds and different levels of grain crowding. Also, it appears that one of the issues is that the suggested solutions were created for controlled environment shots rather than actual field photos [4]. The focus on the same type of wheat throughout the training and testing of the models is another limitation that results in the models becoming excessively tailored to that particular wheat type [5]. The emphasis on the mAP0.5 measure is a restriction on what head detection algorithms can do. In other words, if the head covers more than 50% of the IoU detection, the detection is judged to

* Dhruvisha N. Jaiswala
E-mail address: dhruvishakali99@gmail.com

be accurate. The GWHD dataset is used in this paper to address the first restriction [1]. That includes photographs taken in the field and exhibits a greater variety of sources, methods of acquisition, environmental factors, and wheat varieties. An alternative metric, mAP0.5:0.75, was adopted to make the second restriction more clear. The GWHD Dataset is the only one with such a high prevalence of overlapping and occluded items. To prevent overfitting and arrive at a universal answer for all wheat photos, significant work was also put into data transformation and regularization of deep models.

In May 2020, the GWHD dataset was released. Kaggle and an AI crowd [6] conducted the GWHD challenge in 2021. Multiple submissions and research happened at that time. The winning paper was also studied. [1].

3. Data Description

The dataset [1] consists of more than 1024×1024 pixel images, each comprising more than 270k distinct wheat heads, along with the appropriate bounding boxes. There are 44 unique measurement terms and images from 11 different nations. A measurement session is a collection of images taken with a particular sensor at the same location over a predetermined period of time (often a few hours). The test dataset will be made up of images from North America (excluding Canada), Asia, Oceania, and Africa and consist of approximately 2000 images. The training dataset will consist of images gathered in Europe and Canada, covering approximately 4000 images [6].

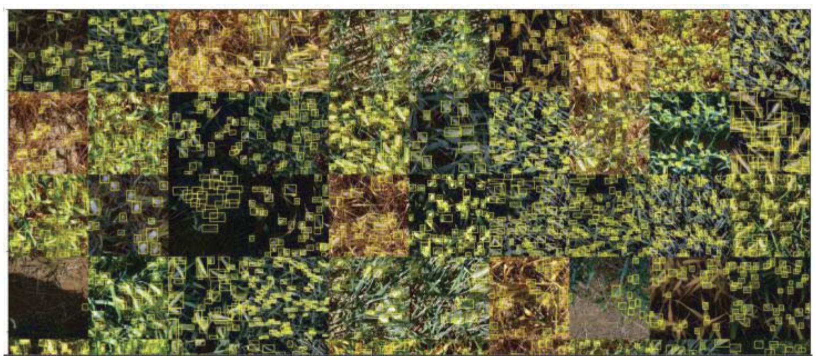


Fig. 1. Some images visualization with bounding boxes from Dataset

4. Methodology

Using object detection models to discover wheat heads in images involves four main processes. The first stage is the Exploratory Data Analysis (EDA), which was incorporated into the design to enhance data collection. Next comes the semi-supervised learning procedure. The third phase ushers in the ensemble one. Lastly, a post-processing block was constructed to maximize the correctness of the entire design, as seen in Fig. 2. Each step will be further explained in the subsections that follow.

4.1 Data transformation (EDA):

Exploratory data analysis was crucial to the work and assisted in producing the best possible inference outcomes. This procedure is circular. Analyze the data, create hypotheses, models, and conclusions, validate them, compare them, and keep analyzing the data until you get satisfactory findings. The techniques listed below for data transformation.

4.1.1 Data analysis:

Data analysis is not only a part of the experimental framework, as it was in the traditional object identification techniques, but it is also an essential stage in the development process. GWHD [6] dataset helps to overcome such problems by providing both diversity and quantity of well-labelled images. The analysis found 49 images without wheat labelling. Also, most wheat heads are smaller in size, and some images have low and high brightness.

4.1.2 Data cleaning:

Data cleaning is the procedure of identifying incomplete, imprecise, erroneous, or unnecessary parts of data and updating, modifying, or removing them [2]. To stop models from being misled and to increase the accuracy of outcomes, data cleansing is a necessary step. Some bounding boxes cover multiple heads, while smaller bounding boxes do not cover any heads. So, by setting a threshold that took into account the range of projected sizes for wheat heads, the enormous and tiny bounding boxes were eliminated.

4.1.3 Data Splitting:

A sufficient validation data set is essential for having reliable validation precision. The data was divided using stratified k-fold splitting. This approach divided the data into k folds, ensuring that each fold had approximately the same number of images per source and bounding box distribution. In that method, cross validation proved more reliable than randomly dividing the data. The training and test sets are guaranteed to include the same percentage of the feature of interest as the original dataset [7].

4.1.4 Data Augmentation:

During experiments, it was discovered that different data augmentation techniques had a stronger impact on increasing the average precision of models [2]. There are numerous ways to alter the same image, such as horizontal and vertical flips, cuts and resizing, rotations, cropping, random erasing, CutOut, CutMix.

4.2 Pseudo Labelling:

During the inference phase of this work, a semi-supervised approach called pseudo-labelling [8] was applied. In this study, the model did not structure a real-time object recognition application; therefore, it was desirable and a viable choice to try pseudo labelling (PL). Pseudo-labelling was used to train an object detector in a supervised manner, then consider adding pseudo labels to the training dataset, retrain the model, and predict new bounding boxes for test data.

4.3 Ensemble Learning:

Ensemble learning combines numerous individual models to acquire better generalization performance. The following ensemble techniques are used in the mentioned model:

4.3.1 Test Time Augmentation (TTA):

Using post-processing techniques, TTA takes several outputs, separates the detected wheat heads, and then merges the results [9]. Whereas other data augmentation tasks are done prior to or during model training, this one is finished during the inference phase. The objective is to provide the same model with numerous versions of the same image, then integrate the findings using a post-processing technique by extracting the detected wheat heads from various outcomes [2].

4.3.2 Bootstrap Aggregation (Bagging):

A machine learning (ML) ensemble algorithm called bootstrap aggregating, commonly known as bagging, aims to improve the precision and stability of ML systems. Moreover, it lowers variance and aids in preventing overfitting [10].

4.3.3 Multi-Scale Ensemble:

Multi-Scale Ensemble Object detection is performed at multiple output layers so that receptive fields match objects of different scales [11]. Several EfficientDet models were trained by the authors [2] using various input measurements, specifically 512×512 (pixels) and 1024×1024 (pixels). To compare the results, train a few models with 1024×1024 input resolution, but most essential, ensemble both strategies.

4.4 Post-Processing Algorithm:

Post-processing steps were needed to generate the right bounding boxes for object detection models. Weighted Boxes Fusion (WBF) was used to combine all predictions from multiple models. The average boxes were constructed using the confidence scores of all the presented bounding boxes.

Workflow diagram:

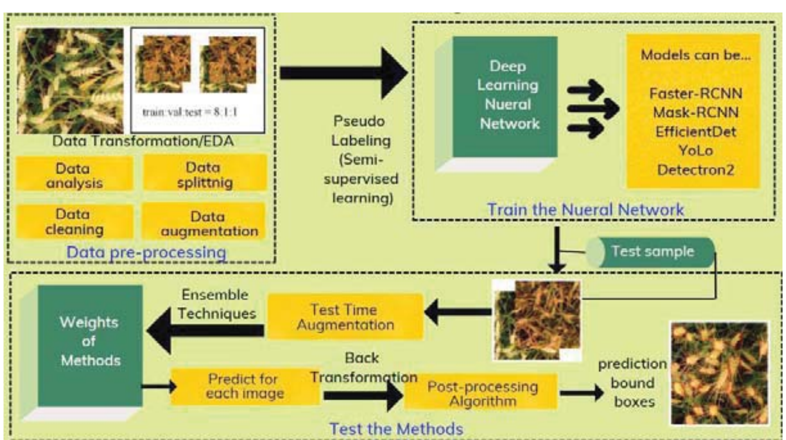


Fig. 2. The workflow of training and testing model detectors for wheat heads.

5. Results & Discussion

In this article, four deep learning algorithms were discussed: Faster R-CNN, EfficientDet, Detectron2, and YoLov5.

Table 1. Following are hyper parameters for Faster RCNN [2]

Hyper Parameters	Description
Backbone	ResNet50
Optimizer	Stochastic Gradient Descent (SGD)
Momentum	0.9
Weight Decay	0.005
Batch size	16

When applying a cleaning on the data and increasing the size of the original dataset by adding one random augmentation for each original image, the mAP increases by around 2%. The best result (68.46%) was achieved by combining the cleaning, multiplying the size of the original dataset by 4, and adding the pseudo labelling [2].

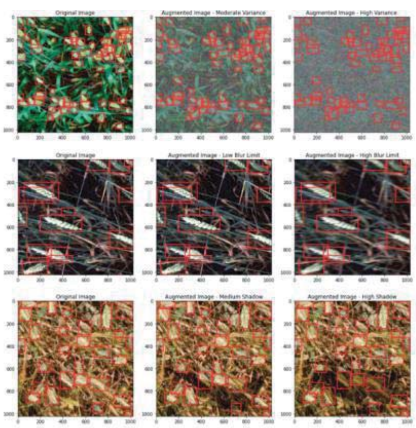


Fig. 3. Samples images from Data Augmentations where the first column represents the original images and the second and third column shows the augmentations with different parameters. [2]

Furthermore, reduced the batch size to 4, augmented the number of epochs to almost 100, employed weighted box fusion as post-processing, and used a learning rate scheduler that was distinct from the one used for Faster R-CNN. Combining bagging with pseudo labelling and augmentation, the result achieved 73.43%. The final experiment combined two models, the first with a 512-pixel image size and the second with a 1024-pixel image size, and applied data augmentation techniques such as TTA, bagging, and multiscale ensemble; the result reached 74.22%, which was the higher result achieved [2, 13, 14].

To elaborate on this topic, execute all of these techniques on another dataset from Kaggle [15]. The dataset is composed of 3422 images of 1024×1024 pixels containing 147k+ unique wheat heads, with the corresponding bounding boxes. The images appear from 7 countries with nine research institutes. The following images are samples of the dataset with bounding boxes. (Fig. 4)

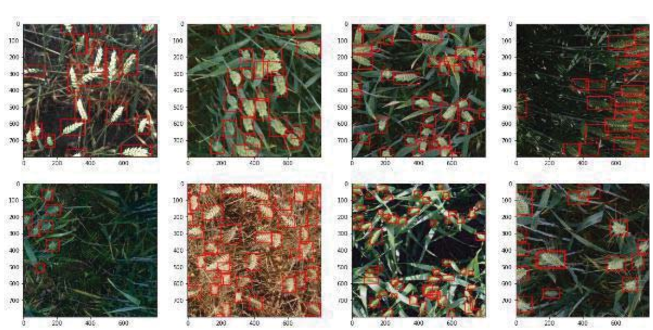


Fig. 4. Sample images from the above mentioned Dataset [15]

Table 2. Following are hyper parameters for respective models.

Hyper Parameters	FasterRCNN	EfficientDet	Detectron2	YoLov5
Backbone	ResNet-50	EfficientNet	ResNet-50	BottleNeckCSP
Batch Size	16	4	16	16
Learning rate	0.5	0.0002	0.001	0.01
Momentum	0.9	0.1	0.9	0.937
Decay	0.001	0.001	0.0001	5e-4
Iteration/Epochs	20 epochs	17 epochs	10000 iterations	1350 iterations
Optimizer	Adam/SGD	AdamW	Adam/SGD	Adam/SGD
IoU Threshold	0.5	-	0.7	0.8

To get better performance out of the data, augmentation techniques were applied using some well-known ones, such as Random Brightness, Flip, Crop, Rotation, Contras , HueSturation, RGBShift, Horizontal and Vertical Flip, Cutout, CutMix.

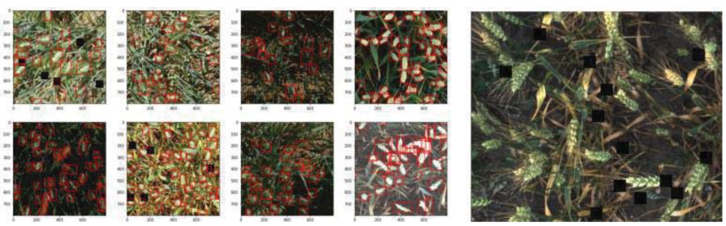
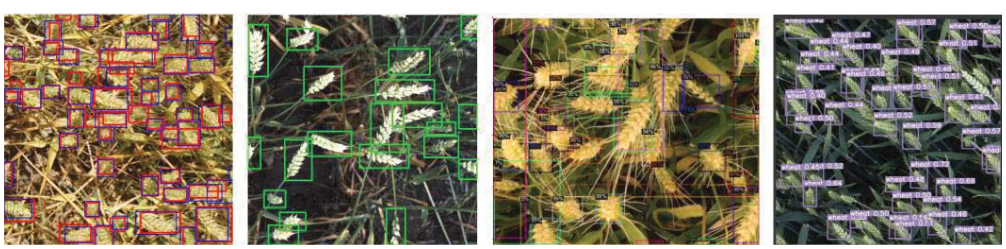


Fig. 5. Some sample images which show the data augmentation on the actual dataset.

Table 3. Performance of the different object detection models.

Model Name	Data cleaning	Data Augmentation	mAP %
Faster-RCNN	Ture	True	56%
EfficientDet	True	True	54%
Detectron2	True	True	-
YoLov5	True	-	64%

The probability of the existing wheat head over each bounding box is also represented in Fig. 6 (c and d). So, as per the performance analysis (Table 3), YoLov5 gives better precision than the other three models.



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

Fig. 6. Results after implementing four studied object detection models. Predicted wheat heads from faster-RCNN (a), EfficientDet (b), Detectron2 (c), and YoLov5 (d).

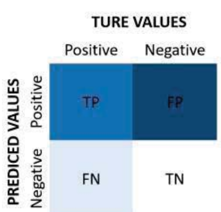
In Fig. 6 (a), the model predicted wheat heads in blue bounding boxes, and ground truth wheat heads are shown in red bounding boxes. The probability of the existing wheat head over each bounding box is also represented in Fig. 6 (c and d). So, as per the performance analysis (Table 3), YoLov5 gives better precision than the other three models.

After implementation of all four models on a small dataset [15], the highest accuracy was about 64%. So, decided to choose a bigger dataset [6] to get more accuracy and tried to get the minimum heads count difference between ground truth values and predicted heads count values. The following results and discussion will be on the GWHD dataset [6] using FasterRCNN and YoLov5 detection models. The dataset consists of 6511 high-resolution RGB images from 47 domains, with 3655 images in the training dataset and 2856 images in the testing dataset. Also a file with 163k+ labelled bounding boxes on train images.

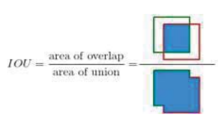
Table 4. Hyper parameters for FasterRCNN and YoLov5 on GWHD dataset [6].

Hyper Parameters	FasterRCNN	YoLov5
Backbone	ResNet-50	BottleNeckCSP
Batch size	4	8
Learning rate	0.0001	0.01
Momentum	0.90	0.98
Decay	0.0005	0.001
Optimizer	Adam	Adam
IoU Threshold	0.5	0.7

Performance Evaluation Metrics:



Confusion Metric [16]:
True Positives (TP): when the true value is Positive and prediction is also Positive.
True Negatives (TN): when the true value is Negative and prediction is also Negative.
False Positives (FP): when the true value is negative but prediction is Positive.
False Negatives (FN): when the true value is Positive but the prediction is Negative.



Intersection over Union (IoU):
IoU is a metric that evaluates the overlap between the ground-truth mask and the predicted mask. IoU is used in object detection to determine if a given detection is valid or not.
Note: IoU metric ranges from 0 and 1 with 0 signifying no overlap and 1 implying a perfect overlap between ground truth and predicted [17].

Fig. 7. IoU formula [17]

$$IoU(A,B) = \frac{A \cap B}{A \cup B}$$

- Precision: Precision (Pre) is a metric that designates the proportion of true positive outcomes. It is formulated as follows: $\text{Pre} = \frac{TP}{TP+FP}$
- Recall: Recall (Re) is a metric that designates the proportion of true positives that were successfully detected. It is formulated as follows: $\text{Re} = \frac{TP}{TP+FN}$
- F1-Score: The F1-Score is calculated as the harmonic mean of precision and recall and is calculated as follows: $\text{F1-Score} = 2\,(\frac{Pre*Re}{Pre+Re})$

mAP :
Mean Average Precision (mAP) is a metric used to evaluate object detection models such as FasterRCNN, YoLo, Mask-R CNN etc.

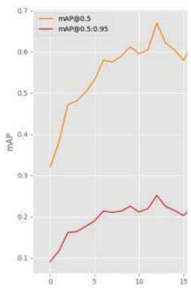
mAP metric is based on 4 sub matrices: Confusion matrix, Intersection over Union (IoU), Precision, and Recall.

$$\text{mAP} = \frac{1}{No.\ of\ Classes}\left(\frac{TP}{TP+FP}\right)$$

FasterRCNN:

The Pytorch Lightning API was used for implementing the FasterRCNN model for detection. Pytorch Lightning is a more structured API to save and load model progress, and it also helps to make machine learning more scalable so one can build more AI models efficiently and quickly [18].

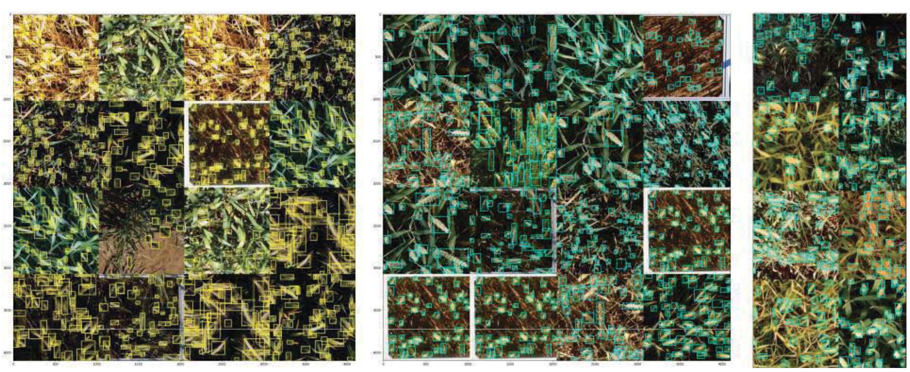
Table 5. Results of FasterRCNN implementation with specific no. of epochs.



Epochs	No. of wheat heads	mAP (%)
1	79,169	36.3
5	89,056	47.9
10	76,055	68.4
15	87,183	59.1

Fig.8. mAP performance of FasterRCNN

The following images are the results of wheat head detection using FasterRCNN (Fig. 9). Table 5 represents the total number of wheat heads in test images, where implementation generates a file with the predicted bounding box string and the total predicted count of every image.



(a) (b) (c)

Fig. 9. Some predicted wheat heads images using FasterRCNN on a test dataset [6]. (a). Wheat head prediction with 5 epochs, and (b). Prediction with 10 epochs. (c). Prediction with 15 epochs.

YoLov5:

To set up the core implementation and environment for YoLov5, use the YoLov5 repository [19, 20]. The wheat head detection model YoLov5 followed a workflow diagram (Fig. 2). Following Table 6 are the results of k-cross validation. During wheat head detection using YoLov5, a total of 4 folds are used: fold0, fold1, fold2, and fold3. Table 6 represents the evaluation metrics during the cross-validation technique applied to the training dataset.

Table 6. Numeric results of data splitting technique for YoLov5

Results	fold0	fold1	fold2	fold3
F1-score	0.64 at 0.514	0.63 at 0.807	0.69 at 0.502	0.64 at 0.494
Precision	0.819	0.956	0.830	0.796
Recall	0.96	0.95	0.95	0.96
Precision/Recall	0.679	0.658	0.669	0.660

Dataset [6] splits into 4 subparts and stores labels for further use as an example of pseudo labelling.

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

Fig. 10. Image visualization after k-cross validation is applied in YoLov5. (a). Generating bounding box debugger for fold0, (b). Mosaic images which are generated through training batches in fold0, (c). Bounding box debugger for fold3, (d). Mosaic images through training batches in fold3.

As per the above discussion, semi-supervised learning techniques produce more accurate outcomes. Here, pseudo labelling is used for semi-supervised learning.

Table 7. YoLov5 mAP performance while performing k-cross validation and pseudo labeling with specific no. of epochs.

K-Fold	-Cross Validation	(epochs)	mAP (%)	Pseudo labeling	(epochs)	mAP (%)
0	fold0	10	89.3	fold0_pseudo0	2	67.9
1	fold1	12	95.0	fold1_pseudo1	5	79.0
2	fold2	10	91.3	fold2_pseudo2	2	66.9
3	fold3	12	88.6	fold3_pseudo3	5	80.6

Wheat head detection implementation trains images from four different splits with four folds, where train images are processed in folds, and then stores images and labels in different folders. Those labels are used in pseudo labelling to predict wheat heads in test images. Pseudo labelling generates labels for test images, and YoLov5, the trained model, predicts the wheat heads in every single test image.

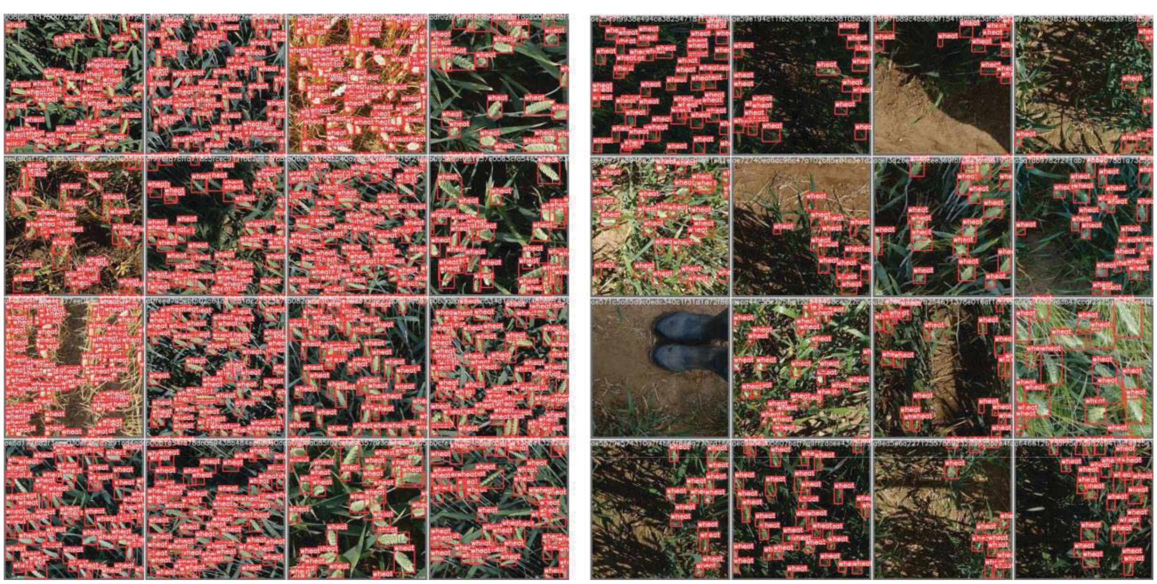


Fig. 11. Outcomes of YoLov5: Wheat heads prediction with labels on test images.

Each fold and each set of pseudo labels generate different outcomes in every test image. Integrating all the results of the four folds and increasing the number of epochs, getting more mAP, as shown in Table 7, and also generating the file of test images with predicted bounding boxes per image and the total count of wheat heads in one image.

Table 8. mAP performance and total number of wheat heads detected in test images using YoLov5.

Epochs	No. of wheat heads detected (Predicted Wheat Heads)	mAP (%)
Cross-validation : 35 Pseudo Labelling : 15	1,09,422	86.82

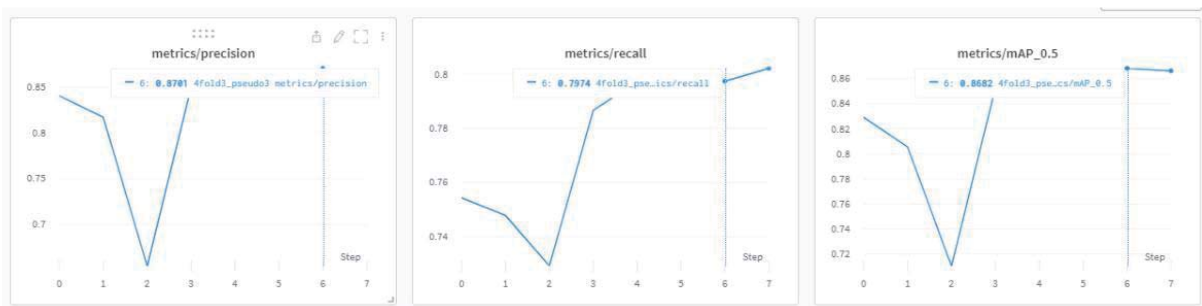


Fig. 12. Final matrics of precision, recall and mAP on test images using YoLov5 detection model.

Table 9. Model comparison on Wheat Head Detection using GWHD dataset [6].

Model Name	Data Cleaning	Data Augmentation	Pseudo Labeling	Test-Time Augmentation	Post processing	mAP (%)
FasterRCNN	True	True	-	True	Ture	68.40
YoLov5	True	True	Ture	Ture	Ture	86.82

Table 10. Some test images wheat heads count comparison with ground truth count values.

image_name	domain	Ground truth count	Predicted count FasterRCNN	Predicted count YoLov5
e38255a86b481f787da4b7ae4a227f6ded7c23b36438af45e63ed67f2dfc4fb4	15	34	35	33
08f2f41ad45dc5235b88179d6c373878c957e4ef5d46afe736c4f4d26b57793b	10	18	23	18
53895351db06d453f47c0602637fb5ad1fd32074059ce6d9645de2b3e7179064	6	63	63	64
fb5f7521da26020909c8786af72305813a468749ddfa21c3d16a98e949510ecb	5	14	13	14
debc17970bfc40679dd3e60c4ceb961614fa37a7b0ad6578ec58f726fb40ccfa	3	43	43	44
3902fc957d935d37319d24438998cdba08624e920f5a8ea58ae62b72f3b372f2	2	25	24	25
035a987f13fd2ad00e8b13f46e59556a384873ace564ea7c56a3a78536f3e2b1	1	23	23	23
2d52563250c95e48817c097e8cb5c53211d9c522d77c709637b24d2ffdaf3f02	4	48	49	49
.	.	.	.	.
.	.	.	.	.
Total Wheat Head Count for 20 images		**928**	**913**	**916**

Table 10 is the basic analysis of the headcount and comparison with true values. Table 10 contains image_name, which is a random 20 test image from the test dataset [6]. So as per the studies of both models: FasterRCNN and YoLov5, the YoLov5 detection model is more accurate for the global wheat head dataset. Also calculating the total count of heads with FasterRCNN and YoLov5, YoLov5 has a smaller difference in the ground truth count as compared to the count generated through

FasterRCNN. FasterRCNN gave 98.38 % and YoLov5 gave 98.71 % accurate counts for 20 sample images from the test dataset.

6. Conclusion

Wheat head detection is a valuable task for wheat production estimation, wheat breeders, and crop management. This study looked into a novel framework for detecting wheat heads in a GWHD dataset, which was released in 2021. This work is mostly concentrated on regularizing deep learning model techniques. To improve the precision of models, semi-supervised learning and ensemble learning techniques were also explored. To enhance training procedures, this study went through the data and analyzed the origins, data types, distribution, diversity, and specificity. Also able to tackle some limitations in the field of wheat head detection, like the unicity of the acquisition methods for the data as well as the lack of precision in the detection. This study uses two different versions of the datasets, which makes the analysis more understandable. Moreover, a comparative analysis was covered. In comparison to a smaller dataset, the study is more accurate with the larger one. In future work, willing to compare methods to further deep learning architectures for wheat head detection.

References:

1. E. David, Shouyang Madec, Pouria Sadeghi-Tehran, “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”. Plant Phenomics- The Science Partner Journal, April, 2020. Dataset - https://zenodo.org/record/5092309#.Y7lcG3ZBzIU.
2. Fares Fourati, Wided Souidene Mseddi & Rabah Attia, “Wheat Head Detection using Deep, SemiSupervised and Ensemble Learning”. Canadian Journal of Remote Sensing- Taylor & Francis, April, 2021.
3. Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X., 2019. “Object detection with deep learning: A review”. IEEE transactions on neural networks and learning systems, 30(11), pp.3212-3232. https://arxiv.org/pdf/1807.05511.pdf&usg=ALkJrhhpApwNJOmg83O8p2Ua76PNh 6tR8A
4. Pound, M. P., Atkinson, J. A., Wells, D. M., Pridmore, T. P., and French, A. P. 2017. “Deep learning for multitask plant phenotyping”. In Proceedings of the IEEE International Conference on Computer Vision, pp. 2055–2063
5. Madec, S., Jin, X., Lu, H., De Solan, B., Liu, S., Duyme, F., and Baret, F. 2019. “Ear density estimation from high resolution RGB imagery using deep learning techniques.” Agricultural and Forest Meteorology, Vol. 264: pp. 225–234. doi:10.1016/j.agrformet.2018.10.013
6. AI crowd, Global Wheat Head Challenge 2021, By University of Saskatchewan. https://www.aicrowd.com/challenges/global-wheat-challenge-2021
7. rohan007,GeeksforGeeks- “Stratified K Fold Cross Validation” https://www.geeksforgeeks.org/stratified-k-fold-cross-validation/
8. Lee, D.H. 2013. “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks.” Workshop on Challenges in Representation Learning, ICML, Vol. 3 (No. 2): p. 896.
9. Nathan Hubens, 2019. “ Test-Time Augmentation and how to perform with Keras”, Feb 12, 2019. https://towardsdatascience.com/test-time-augmentation-tta-and-how-to-perform-it-with-keras-4ac19b67fb4d
10. Breiman, L. 1996. “Bagging predictors.” Machine Learning, Vol. 24 (No. 2): pp. 123–140. doi:10.1007/BF00058655.
11. Yuseok Ban and Kyungjae Lee, 2020 “Multi-Scale Ensemble Learning for Thermal Image Enhancement”. Appl. Sci. 2021, 11(6), 2810; https://doi.org/10.3390/app11062810
12. Sambasiva Rao. K, Medium Article, “Weighted Boxes Fusion — A detailed view”, Feb, 23, 2021. https://medium.com/analytics-vidhya/weighted-boxes-fusion-86fad2c6be16
13. Abhi Lad, Mehul S. Raval, 2022. “Improving Wheat Head Detection: A Data-Centric Approach by Domain Variance Reduction”, https://dl.acm.org/doi/10.1145/3477314.3507190.
14. Movva Nitin Datta1, Yash Rathi, M. Eliazer, SRM Institute of Science and Technology, Kattankulathur, 2021. “Wheat Heads Detection using Deep Learning Algorithms”. Annals of R.S.C.B., ISSN: 1583-6258, Vol. 25, Issue 5, 2021, Pages. 5641 - 5654.
15. Kaggle - Global Wheat Detection challenge 2020. https://www.kaggle.com/competitions/global-wheat-detection
16. Anuganti Suresh, 2020. Medium Article, “What is a confusion matrix?”, Nov. 17, 2020. https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5
17. Kiprono Elijah Koech, 2020. Medium Article, “Confusion Matrix for Object Detection”, Aug 1, 2020. https://towardsdatascience.com/confusion-matrix-and-object-detection-f0cbcb634157#:~:text=IoU%2C%20aso20called%20Jaccard%20 index,detection%20is%20valid%20or%20not.
18. SABREPC.COM, 2022.“Deep Learning and AI : Why You Should Use PyTorch Lightning and How to Get Started”, March 3, 2022. https://www.sabrepc.com/blog/Deep-Learning-and-AI/why-use-pytorch-lightning
19. ULTRALYTICS - YOLOV5, Nov 22, 2022. Github repository. https://github.com/ultralytics/yolov5
20. KSNXR-YOLOV5, June 16, 2022. Github repository. https://github.com/ksnxr/GWC_YOLOv5
21. TAN, M., and LE, Q. V. 2019. “EfficientNet: Rethinking model scaling for convolutional neural networks.” arXiv preprint arXiv:1905.11946

22. TAN, M., PANG, R., and QUOC, V. L. 2020. “EfficientDet: Scalable and efficient object detection.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. doi:10.1109/CVPR42600.2020.01079.
23. OPENGENUS-IQ, “Understanding ResNet50 architecture”, 2022. https://iq.opengenus.org/resnet50-architecture/