Robust Wheat Head Detection in Diverse Field Environments

Xiang Gao
Electrical and Computer Engineering
PID: A59015773

Xiaoyu Liu
Electrical and Computer Engineering
PID: A59023313

Abstract

Wheat is a fundamental grain in many food products, making its cultivation and management critical. Accurate detection of wheat heads in field images is essential for estimating crop health and yield. However, this task is challenging due to the dense overlap of plants and varying environmental conditions. This study aims to enhance wheat head detection using convolutional neural networks (CNNs), focusing on creating a generalized solution applicable across diverse growing environments. We utilized datasets from the Global Wheat Detection Competition, SPIKE dataset, and Wheat 2017 dataset, incorporating advanced data augmentation techniques such as custom mosaic augmentation, MixUp, and heavy augmentation to improve model robustness. Our approach involved extensive data preprocessing, integration of external datasets, and a comprehensive model training process using EfficientDet and Faster R-CNN FPN models. The training strategy included five-fold stratified cross-validation, mixed precision training, and pseudo labeling to iteratively refine model performance. Model ensemble techniques and test time augmentation further boosted detection accuracy. The evaluation, based on mean average precision (mAP) across different Intersection over Union (IoU) thresholds, demonstrated significant improvements in wheat head detection accuracy. The EfficientDet-D7 model with larger image sizes showed superior performance, and the iterative pseudo labeling process effectively enhanced model generalization. This research provides a robust framework for wheat head detection, aiding farmers in better crop management and contributing to the quality and availability of wheat-based products.

1 Introduction

Wheat, a staple grain found in many households, is integral to numerous food products, including morning toast and cereal. Its significance as both a food and a crop has made wheat the subject of extensive scientific study. To obtain comprehensive and accurate data about wheat fields globally, plant scientists employ image detection techniques to identify "wheat heads" — the spikes on the plant containing grains. These images are crucial for estimating the density and size of wheat heads across different varieties, enabling farmers to assess crop health and maturity for better field management decisions. However, accurately detecting wheat heads in outdoor field images presents considerable challenges. Dense wheat plants often overlap, and wind can cause blurriness in photographs, making it difficult to distinguish individual wheat heads. Furthermore, variations in appearance due to differences in maturity, color, genotype, and head orientation complicate the detection process. Given the global cultivation of wheat, models developed for wheat phenotyping must generalize across diverse growing environments. Current detection methods, such as one- and two-stage detectors like Yolo-V3 [1] and Faster-RCNN [2], still exhibit biases towards the training region, even when trained on large datasets. In this project, we aim to develop a model for detecting wheat heads from outdoor images of wheat plants, focusing on creating a generalized solution that estimates the number and size of wheat heads. Our objective is to improve the accuracy and robustness of wheat head detection,

enabling farmers to better assess their crops. Successful development of such a model will allow researchers to accurately estimate the density and size of wheat heads in different varieties. Improved detection capabilities will enable farmers to better assess their crops, ultimately enhancing the quality and availability of wheat-based products like cereal and toast for consumers worldwide.

2 Related Work

2.1 Cosine Annealing: Concept and Principles

Cosine Annealing is a learning rate scheduling technique introduced to improve the efficiency and performance of deep learning models during training. This method, detailed in the paper "SGDR: Stochastic Gradient Descent with Warm Restarts" by Ilya Loshchilov and Frank Hutter, leverages a cosine function to modulate the learning rate over the training epochs, providing a cyclical schedule that enhances the model's ability to converge to a better local minimum. [3]

In deep learning, the choice of learning rate significantly influences the convergence and overall performance of the model. Traditional fixed or exponentially decaying learning rates can either lead to slow convergence or risk overshooting optimal solutions. Cosine Annealing addresses these issues by periodically adjusting the learning rate following a cosine curve, ensuring a smooth transition from a higher learning rate to a lower one over a defined period.

The learning rate η_t at epoch t is computed using the following formula:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min}) \left(1 + \cos\left(\frac{T_{\text{cur}}}{T_{\max}}\pi\right)\right) \tag{1}$$

where $\eta_{\min}$ is the minimum learning rate, $\eta_{\max}$ is the maximum (initial) learning rate, T_{cur} is the current epoch within the cycle, and $T_{\max}$ is the maximum number of epochs in one cycle. This equation ensures that the learning rate starts at $\eta_{\max}$, gradually decreases to $\eta_{\min}$, and then resets, following a cosine function. This periodic resetting, known as a "warm restart," allows the model to escape local minima and explore the parameter space more effectively, promoting better generalization.

Cosine Annealing is particularly beneficial for computer vision tasks, which often involve training deep convolutional neural networks (CNNs) that require careful tuning of the learning rate to achieve optimal performance. The smooth decay of the learning rate helps in stabilizing the training process and avoiding oscillations in the loss landscape. Moreover, the warm restarts inject beneficial randomness, which can help the model converge to more optimal solutions by re-energizing the learning process periodically.

The SGDR approach, as outlined by Loshchilov and Hutter, has demonstrated superior performance in various deep learning applications, including image classification and object detection tasks. By providing a more dynamic and responsive learning rate schedule, Cosine Annealing contributes to faster convergence and improved accuracy, making it a valuable tool in the deep learning practitioner's toolkit.

2.2 Weighted-Boxes-Fusion (WBF): Enhancing Object Detection Accuracy

Weighted-Boxes-Fusion (WBF) is an advanced post-processing technique for object detection tasks, designed to enhance detection accuracy by merging the outputs of multiple detection models. Unlike traditional methods such as Non-Maximum Suppression (NMS), which might discard valuable information by retaining only the highest confidence detections and suppressing overlapping ones, WBF aggregates the detection boxes from different models or instances by leveraging their confidence scores to compute a weighted average. This approach results in more precise and reliable detection outcomes.

The principle behind WBF involves the following steps:

- **Collection of Candidate Boxes:** All predicted bounding boxes and their corresponding confidence scores from multiple models are gathered.
- **Weighted Averaging of Boxes:** For each detection box, the method identifies overlapping boxes that exceed a specified Intersection over Union (IoU) threshold. These overlapping

boxes are then combined using a weighted average, where the weights are derived from the confidence scores of each box. This process takes into account the relative confidence of each detection, thereby producing a more accurate representation of the object's location and size.

- **Generation of Fusion Boxes:** The weighted averaged boxes are used as the final detection results, effectively integrating information from multiple sources to improve overall detection performance.

WBF offers several advantages over traditional NMS. By incorporating the confidence scores of overlapping boxes, WBF mitigates the risk of losing critical information and provides a more nuanced fusion of detections. This leads to improved localization accuracy and robustness, particularly in complex scenarios where multiple detectors might produce slightly varying predictions for the same object.

The original paper "Weighted Boxes Fusion: Ensembling Boxes from Different Object Detection Models" by Roman Solovyev, Vadim Buza, and Igor Ginsburg, presents a detailed explanation of the WBF algorithm and its efficacy [4]. The authors demonstrate through extensive experiments that WBF significantly outperforms conventional NMS and other ensemble methods in terms of precision and recall across various datasets.

2.3 Stochastic Gradient Descent (SGD) and Adam Optimizers

Stochastic Gradient Descent (SGD) is a fundamental optimization algorithm widely used in training deep neural networks. It aims to minimize the loss function by iteratively updating the model parameters in the direction of the negative gradient of the loss with respect to the parameters [5].

SGD operates by randomly sampling a mini-batch of data from the training set at each iteration, computing the gradient of the loss function with respect to the mini-batch, and updating the parameters accordingly. The update rule for SGD is defined as:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta_t, x_i, y_i) \tag{2}$$

where θ_t represents the parameters at iteration t, η is the learning rate, and $\nabla_\theta L(\theta_t, x_i, y_i)$ is the gradient of the loss function L with respect to the parameters, computed for a mini-batch (x_i, y_i).

While SGD is simple and effective, it requires careful tuning of the learning rate and may converge slowly in some cases. Techniques such as momentum can be applied to accelerate convergence by accumulating a velocity vector that keeps track of previous parameter updates.

Adam (Adaptive Moment Estimation) is an optimization algorithm introduced by Diederik P. Kingma and Jimmy Ba in 2014 [6]. It combines the advantages of two other extensions of SGD, AdaGrad and RMSProp, to provide efficient and effective optimization.

Adam computes individual adaptive learning rates for each parameter by estimating both the first and second moments of the gradients. The update rules for Adam are defined as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_\theta L(\theta_t) \tag{3}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_\theta L(\theta_t))^2 \tag{4}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{5}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{6}$$

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{7}$$

where: $\mathbf{m}_t$ and v_t are the first and second moment estimates, β_1 and β_2 are the decay rates for these estimates, $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected moment estimates, and ϵ is a small constant to prevent division by zero.

Adam's adaptive learning rate, combined with momentum from the first moment estimate, enables efficient optimization and faster convergence compared to traditional SGD. It is widely used in deep learning applications due to its effectiveness and ease of use.

These optimization algorithms play a crucial role in training deep neural networks for computer vision tasks, providing the foundation for efficient and effective model optimization.

2.4 NVIDIA Apex: High-Performance Training for Deep Learning Models

NVIDIA Apex is a PyTorch extension designed to optimize the training process of deep learning models, particularly for large-scale tasks in computer vision and other domains. It offers tools and techniques for mixed precision training and distributed training, which are essential for leveraging modern hardware architectures efficiently [7].

One of the key features of NVIDIA Apex is its support for mixed precision training. Mixed precision training involves using both 16-bit and 32-bit floating point types in a model to reduce memory usage and increase computational efficiency without sacrificing model accuracy. This technique is particularly beneficial for training large-scale models on GPUs with limited memory capacity.

Apex's Automatic Mixed Precision (AMP) feature simplifies the implementation of mixed precision training by automatically converting certain operations to 16-bit precision while keeping others in 32-bit precision. This approach ensures stability during training and can lead to faster convergence and improved training times.

NVIDIA Apex also provides tools for distributed training, allowing models to be trained across multiple GPUs or even across multiple nodes in a cluster. This is essential for scaling training to handle large datasets and complex models, as it enables parallel processing and accelerates the training process.

Apex's Distributed Data Parallel (DDP) module enhances PyTorch's native distributed training capabilities by providing better performance and usability. It ensures efficient communication and synchronization of gradients and model parameters across multiple GPUs, enabling seamless parallelization of training tasks.

3 Method

3.1 Data Processing

In this project, we employed several advanced data augmentation techniques to improve the robustness and performance of our wheat head detection models. These techniques include Custom Mosaic Augmentation, MixUp, and Heavy Augmentation. Each of these techniques was selected for its ability to enhance the model's generalization capabilities and performance in diverse and challenging conditions.

3.1.1 Custom Mosaic Augmentation:

Custom Mosaic Augmentation involves combining multiple images into a single composite image. This technique is particularly effective for object detection tasks as it significantly increases the variety of object placements and contexts within a single training batch. By randomly selecting and cropping regions from multiple images and then stitching them together, we create more complex training samples that help the model learn to detect objects under varied conditions [8].

In our implementation, the **load_cutmix_image_and_boxes** function was used to load and combine four images into one. This augmentation not only diversifies the training data but also helps the model become more resilient to occlusions and overlapping objects, which are common challenges in wheat head detection.

3.1.2 MixUp:

MixUp is an augmentation strategy where two images and their corresponding labels are linearly combined to create a new training sample. This technique helps in regularizing the model by encouraging it to learn smoother decision boundaries. MixUp also reduces the model's tendency to overfit by providing a form of data interpolation between samples. [9]

During training, MixUp was applied by blending images and their bounding boxes with a random coefficient. This results in mixed images where wheat heads from different images overlap, providing

additional complexity and variability to the training data. This approach improves the model's generalization capabilities and robustness.

3.1.3 Heavy Augmentation:

Heavy Augmentation involves applying a wide range of transformations to the training images to further increase the variability of the dataset. These transformations include random flips, rotations, blurring, sharpening, noise addition, color adjustments, and more. By extensively augmenting the training images, we aim to expose the model to a vast array of potential real-world scenarios, thus improving its adaptability and performance. [10]

In our experiments, heavy augmentation was implemented using the **albumentations** library. The augmentation pipeline included horizontal and vertical flips, random grayscale conversion, Gaussian noise, motion blur, median blur, CLAHE (Contrast Limited Adaptive Histogram Equalization), sharpening, embossing, brightness and contrast adjustments, and hue and saturation changes. These augmentations were applied probabilistically to each training image, ensuring a diverse and challenging dataset for the model to learn from.

These advanced data augmentation techniques played a crucial role in enhancing the performance of our wheat head detection models. By increasing the variability and complexity of the training data, we were able to train models that are more robust and capable of generalizing well to unseen data, ultimately leading to better detection accuracy and reliability.

3.2 Models

In this project, we utilized two state-of-the-art object detection models: Faster R-CNN with Feature Pyramid Networks (FPN) and EfficientDet. These models were selected for their strong performance in object detection tasks and their complementary strengths.

3.2.1 Faster R-CNN with FPN:

Faster R-CNN is a two-stage object detection model that first proposes regions of interest (ROIs) and then classifies and refines these regions. The addition of Feature Pyramid Networks (FPN) enhances Faster R-CNN by allowing it to detect objects at multiple scales more effectively. FPN creates a feature pyramid from a single-scale input, enabling the detection of objects of different sizes using the same network [11].

In our experiments, we used a ResNet-152 backbone with FPN for Faster R-CNN. The model was fine-tuned on our wheat head dataset using a stratified 5-fold cross-validation approach. Each fold was trained with a Stochastic Gradient Descent (SGD) optimizer and an initial learning rate of 5e-3, with a cosine annealing learning rate scheduler to gradually decrease the learning rate over time. Mixed precision training with NVIDIA Apex was employed to speed up training and reduce memory usage. This setup allowed us to leverage the robust feature extraction capabilities of ResNet-152 and the multi-scale detection capabilities of FPN, resulting in accurate and reliable wheat head detection.

3.2.2 EfficientDet:

EfficientDet is a family of object detection models that balance accuracy and efficiency by combining EfficientNet as the backbone with a customized BiFPN (Bi-directional Feature Pyramid Network). EfficientDet models are known for their high performance while maintaining computational efficiency, making them suitable for large-scale detection tasks.

For our experiments, we used EfficientDet-D7 [12], the largest variant in the EfficientDet family, which offers the highest accuracy. The model was trained using an Adam optimizer with an initial learning rate of 5e-4, also employing a cosine annealing learning rate scheduler. Similar to Faster R-CNN, we used mixed precision training to enhance training efficiency. EfficientDet-D7 was chosen for its ability to capture fine-grained details and detect small objects, which is crucial for accurately identifying wheat heads in high-resolution field images.

By utilizing these two complementary models, we aimed to maximize the detection performance for wheat heads in various environmental conditions and image qualities. The combination of Faster

R-CNN with FPN and EfficientDet allowed us to leverage their respective strengths, resulting in a robust and accurate wheat head detection system.

3.3 Pseudo Labeling

Pseudo labeling is a semi-supervised learning technique where a model is trained on a labeled dataset and then used to generate predictions (pseudo labels) for an unlabeled dataset [13]. These pseudo labels are then used as ground truth labels in subsequent training rounds, effectively augmenting the original labeled dataset with additional labeled data. This approach can help improve model performance by leveraging additional data that was initially unlabeled.

In our experiment, we used pseudo labeling to further enhance the performance of our EfficientDet model. Initially, the model was trained on the labeled dataset, and the trained model was used to predict labels for an unlabeled dataset. These predictions were then added to the training set for subsequent rounds of training. This iterative process allowed the model to benefit from the additional data, leading to improved performance and robustness.

4 Experiments

4.1 Dataset description and data format

The dataset utilized in this study consists of three parts, focusing on wheat spike detection using convolutional neural networks (CNNs). The primary dataset is derived from the Global Wheat Detection Competition [14], while two additional datasets, the **SPIKE dataset** and Wheat 2017 [15] dataset, are incorporated to enhance the robustness and generalizability of the model.

The primary dataset comprises images from wheat fields captured across various global locations, annotated with bounding boxes identifying individual wheat heads. This dataset is curated to provide a diverse range of conditions and environments. The images are contained within train.zip and test.zip. The training images are used for training the CNN models, and a small subset of test images is available for preliminary testing and code development, with the majority reserved for official evaluation. Annotations for training images are provided in train.csv, which includes the unique identifier for each image (image_id), the dimensions of each image (width and height), and the bounding box coordinates formatted as a list [xmin, ymin, width, height]. A sample_submission.csv file is also provided to demonstrate the format for submitting predictions.

Not all images include wheat heads or associated bounding boxes. Each bounding box's details are captured in a separate row within train.csv, ensuring easy mapping between image files and their annotations.

To supplement the primary dataset, two additional datasets are incorporated. The SPIKE dataset, sourced from the PBRC, contains images with manually annotated ground truth labels for wheat spike detection. These images are used for training and testing CNN models and are cropped to a size of 1024x1024 pixels to standardize input dimensions. The Wheat 2017 dataset, sourced from the University of Nottingham's plant images repository, includes images of wheat shoots in a glasshouse environment. Annotations for this dataset include spike locations, spikelets, and labels of awned phenotypes. Similar to the SPIKE dataset, the images are cropped to 1024x1024 pixels to ensure uniformity and reduce computational complexity.

To maintain consistency across the datasets, all images are resized to 256x256 pixels where necessary. This resizing ensures that the computational demands of the CNN models are mitigated, allowing for efficient and effective model training and evaluation.

In summary, the integration of these diverse datasets, along with meticulous preprocessing and annotation efforts, provides a robust foundation for developing accurate and reliable CNN models for wheat spike detection in various environments. This comprehensive approach ensures the credibility and effectiveness of the research outcomes.

4.2 Experimental Flow

The experimental flow of this study involves a comprehensive approach to optimize the detection of wheat spikes using convolutional neural network (CNN) models. The methodology encompasses

data augmentation, model training, pseudo labeling, and model ensemble techniques to achieve high detection accuracy.

4.2.1 Data Preprocessing:

We utilized custom mosaic data augmentation to enhance the diversity of training data. This method combines four training images into one, retaining the border information which is crucial for object detection tasks, as shown in **Figure 1**.

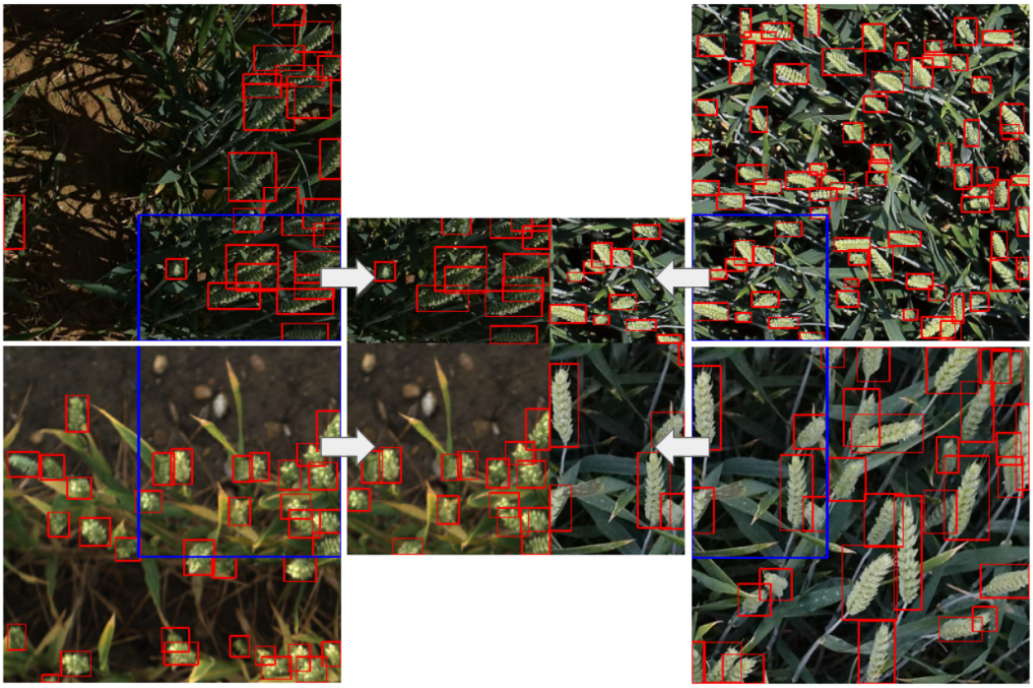

Figure 1: Custom Mosaic Augmentation

Additionally, MixUp was employed to blend two images and their labels, generating new training samples to improve the model's generalization capability. Various heavy augmentation techniques were applied, including RandomCrop, HorizontalFlip, VerticalFlip, ToGray, IAAAdditiveGaussianNoise, GaussNoise, MotionBlur, MedianBlur, Blur, CLAHE (Contrast Limited Adaptive Histogram Equalization) [16], Sharpen, Emboss, RandomBrightnessContrast, and HueSaturationValue. The processed image is shown in the **Figure 2**. These augmentations simulated various real-world scenarios to enhance the robustness of the model. Extensive data cleaning was performed by deleting bounding boxes with width or height less than 10 pixels and fixing excessively large bounding boxes.

4.2.2 External Data Integration:

To further augment the dataset, two external datasets were incorporated: the wheat spikes dataset and the wheat 2017 dataset. Annotations (bounding boxes) were created for all images, and each image was cropped to a size of 1024x1024 pixels. We ensured compliance with licensing requirements by contacting the dataset authors. This integration of external data significantly increased the volume and diversity of the training data, facilitating better model training.

4.2.3 Model Training:

The training process involved several detailed steps to ensure robust evaluation and optimization of the models. Below are the key steps involved in the model training process:

- **Five-Fold Stratified Cross-Validation:** The dataset was split into five folds based on different sources (e.g., usask_1, arvalis_1, arvalis_2). This technique ensures that each

Figure 2: Examples after Apply Mosaic, Mixup, Augmentation

fold is representative of the entire dataset, allowing for robust evaluation of the model's performance across different data distributions.

- **Optimizers: EfficientDet Models:** Trained using the Adam optimizer with an initial learning rate (LR) of 5e-4. Adam is chosen for its adaptive learning rate capabilities, which helps in faster convergence and better handling of sparse gradients. Faster RCNN FPN: Utilized the SGD (Stochastic Gradient Descent) optimizer with an initial LR of 5e-3. SGD is known for its effectiveness in training deep learning models, especially when combined with momentum to accelerate the training process and mitigate oscillations.
- **Learning Rate Scheduler:** A cosine annealing scheduler was used to dynamically adjust the learning rate. This scheduler reduces the learning rate in a cosine manner, promoting stable and efficient training by allowing the model to make large updates in the initial phases and fine-tuning in later stages.
- **Mixed Precision Training:** Implemented using NVIDIA Apex, mixed precision training involves the use of both 16-bit and 32-bit floating-point numbers. This approach accelerates the training process and reduces memory consumption, enabling the training of larger models or the use of larger batch sizes without compromising performance.
- **Warm-Up Phase:** Initially, the models underwent a warm-up phase of 20 epochs. During this phase, the training set was combined with the wheat2017 and spike wheat datasets. The purpose of the warm-up phase is to gradually introduce the model to the training data, stabilizing the model parameters and preventing issues such as divergence due to high initial learning rates.
- **Main Training Phase:** Following the warm-up phase, the main training phase consisted of 80 epochs using the training set combined with the wheat2017 dataset. This phase focuses on thoroughly optimizing the model parameters to achieve the best possible performance on the validation set. The extended training duration ensures that the model has sufficient time to learn and generalize from the augmented and diverse dataset.

4.2.4 Pseudo Labeling:

- **Base Model Training:** Trained EfficientDet-D6 (image size 640) on the training set. Achieved a strong validation average precision (AP).
- **Generating Pseudo Labels:** Used the trained base model to generate pseudo labels for the hidden test set. Combined pseudo labels with the original training set to form an expanded dataset.
- **First Round of Training:** Trained EfficientDet-D6 for an additional 10 epochs using the expanded dataset. Loaded weights from the base model checkpoint.

- **Second Round of Training:** Continued training for 6 more epochs using the expanded dataset.
- **Last step:** Loaded weights from the first round of pseudo labeling checkpoint.

This iterative pseudo labeling process progressively improved the model's performance by continually incorporating additional pseudo-labeled data into the training set.

4.2.5 Model Ensemble and Evaluation:

To enhance the final detection results, predictions from multiple models were combined using Weighted-Boxes-Fusion (WBF), which leverages the strengths of individual models. During inference, test time augmentation (TTA) techniques such as horizontal flip, vertical flip, and 90-degree rotation were applied to further boost model performance.

By integrating sophisticated data augmentation methods, leveraging external datasets, employing advanced training techniques such as mixed precision training and pseudo labeling, and combining model predictions through ensemble methods, we achieved significant improvements in the detection accuracy of wheat spikes. This comprehensive approach ensures robust and reliable model performance, suitable for real-world agricultural applications.

4.3 Results

4.3.1 Evaluation Metrics:

The evaluation of the wheat spike detection model is based on the mean average precision (mAP) at different Intersection over Union (IoU) thresholds. The IoU metric measures the overlap between predicted bounding boxes and ground truth bounding boxes. It is defined as the area of overlap between the predicted and ground truth boxes divided by the area of their union:

$$IoU(A, B) = \frac{A \cap B}{A \cup B} \tag{8}$$

It can be visualized as the following **Figure 3**:

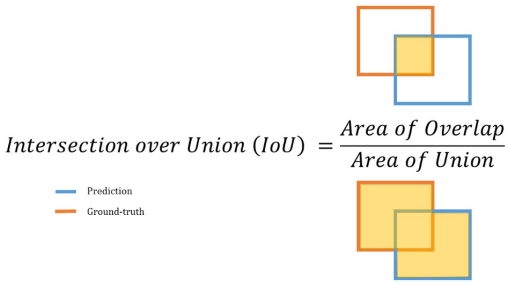

Figure 3: IoU Formula

The evaluation involves sweeping over a range of IoU thresholds, calculating the average precision at each threshold. The thresholds range from 0.5 to 0.75 with a step size of 0.05. A predicted object is considered a "hit" if its IoU with a ground truth object exceeds the threshold.

For each threshold t, precision is calculated based on the number of true positives (TP), false negatives (FN), and false positives (FP):

$$\text{Precision}(t) = \frac{TP(t)}{TP(t) + FP(t) + FN(t)} \tag{9}$$

A true positive is counted when a predicted object matches a ground truth object with an IoU above the threshold. A false positive indicates a predicted object without a matching ground truth object, and a false negative indicates a ground truth object without a matching predicted object.

The mean average precision of a single image is calculated as the mean of the precision values across all IoU thresholds:

$$AP = \frac{1}{|\text{thresholds}|} \sum_{t} \frac{TP(t)}{TP(t) + FP(t) + FN(t)} \tag{10}$$

In the submission, bounding boxes are evaluated in order of their confidence levels. Boxes with higher confidence are prioritized for matches against ground truth objects, which determines the classification of true positives and false positives.

If an image has no ground truth objects, any number of predictions (false positives) will result in the image receiving a score of zero, impacting the mean average precision.

The final score returned by the metric is the mean of the individual average precisions of each image in the test dataset, providing a robust measure of the model's detection accuracy across various scenarios.

4.3.2 Performance and Analysis:

The performance of various models and training strategies was evaluated using the validation average precision (AP) metric. The following sections present the detailed results and analysis, highlighting the effectiveness of different techniques employed in this study.

1. **The Table 1 below summarizes the validation AP achieved by different models and configurations:**

   Table 1: Validation AP achieved by different models and configurations.

Network	Image Size	Fold	Valid AP
EfficientDet-D7	768	0	0.710
EfficientDet-D7	768	1	0.716
EfficientDet-D7	768	2	0.707
EfficientDet-D7	768	3	0.716
EfficientDet-D7	768	4	0.713
EfficientDet-D7	1024	1	0.718
EfficientDet-D7	1024	3	0.720
EfficientDet-D5	512	4	0.702
EfficientDet-D6	640	1	0.716
Faster RCNN FPN-resnet152	1024	1	0.695

2. **Pseudo Labeling Performanc:**
   The pseudo labeling technique significantly improved the model's performance. Below are the results of the pseudo labeling process:
   - **Base Model:** (EfficientDet-D6, Image Size 640, Fold 1) Achieved a Valid AP of 0.716.
   - **Round 1:** Trained EfficientDet-D6 for 10 epochs with the training set and hidden test set (output of ensembling), loading weights from the base checkpoint, resulting in an AP of 0.7633.
   - **Round 2:** Continued training EfficientDet-D6 for 6 more epochs with the training set and hidden test set (output of pseudo labeling Round 1), loading weights from the Round 1 checkpoint, resulting in an AP of 0.7656.
3. **Analysis:**
   The results demonstrate that the EfficientDet-D7 model, particularly with larger image sizes (1024), consistently achieved higher AP scores across different folds. The use of pseudo labeling further boosted the performance, as seen in the significant increase in AP from the base model to the final rounds of pseudo labeling.

   The higher AP scores for larger image sizes suggest that the model benefits from more detailed input, allowing it to better identify and localize wheat spikes. Additionally, the

pseudo labeling technique's iterative approach of incorporating pseudo-labeled data helps in refining the model's predictions and improving its generalization capability.

5 Conclusion

This project presents a comprehensive approach to improving the accuracy and robustness of wheat head detection in field images using advanced convolutional neural network (CNN) techniques. By integrating diverse datasets from the Global Wheat Detection Competition, SPIKE dataset, and Wheat 2017 dataset, we ensured a broad representation of different growing conditions and environments. Our methodology included sophisticated data augmentation strategies, such as custom mosaic augmentation, MixUp, and heavy augmentation, which significantly enhanced the training data's diversity and robustness.

We employed EfficientDet and Faster R-CNN FPN models, leveraging their strengths in object detection tasks. The training process was meticulously designed, incorporating five-fold stratified cross-validation, mixed precision training with NVIDIA Apex, and iterative pseudo labeling. These techniques collectively contributed to the model's improved performance and generalization capabilities. The use of a cosine annealing scheduler for learning rate adjustment and the warm-up phase further stabilized the training process.

Our results demonstrated that the EfficientDet-D7 model, especially when trained with larger image sizes, consistently achieved higher average precision (AP) scores across different folds. The pseudo labeling technique proved to be particularly effective, progressively refining the model's predictions and enhancing its generalization to unseen data.

Overall, this research provides a robust and reliable framework for wheat head detection, which can significantly aid farmers in assessing crop health and making informed management decisions. The successful implementation of these techniques in real-world agricultural applications has the potential to improve the quality and availability of wheat-based products, ultimately benefiting the global food supply chain. Future work could explore further enhancements in model architectures and the integration of additional data sources to continue advancing the field of agricultural image analysis.

6 Implementation

The development of our object detection system leverages established methodologies combined with novel approaches tailored for the scope of this graduate-level project. Central to our implementation are the Fast R-CNN with Feature Pyramid Networks (FPNs) [2] and the EfficientDet [5] architectures, adapted from their respective seminal papers. These frameworks were chosen for their robust performance in handling diverse and challenging datasets.

Our implementation extended beyond the mere adoption of pre-existing architectures by incorporating a customized training regimen. Notably, we implemented a five-fold cross-validation strategy to ensure the generalizability and robustness of our models across different data splits. This method is critical in assessing the performance variability and in preventing overfitting, thus ensuring that our findings are statistically reliable.

Further enhancing our model's capability, we integrated techniques such as Pseudo Labeling [13] to utilize unlabeled data effectively, thereby semi-supervised learning enlarges our training dataset and improves the model's accuracy and robustness. This approach was inspired by successful implementations in recent competitions, where it has shown significant improvements in model performance by leveraging unlabeled data.

Additionally, our models were initialized with weights from pre-trained networks available within the EfficientDet and Fast R-CNN repositories, providing a strong starting point for learning and significantly reducing the training time. Such pre-training is essential in deep learning practices to achieve high performance, especially when labeled data is scarce.

Data preprocessing played a pivotal role in our pipeline, adopting proven techniques from other competitors in the field. This included advanced image augmentation strategies like mixup, mosaic blending, and geometric transformations [16] to increase the diversity of training examples and simulate various object scales and orientations, thereby enhancing the detection robustness.

Each component of our system was meticulously developed and integrated, ensuring seamless functionality and optimal performance. The combination of robust model architectures with advanced training and preprocessing techniques outlines our commitment to adopting advanced practices while pushing the boundaries with innovative data augmentation and training strategies.

References

[1] Pranav Adarsh, Pratibha Rathi, and Manoj Kumar. Yolo v3-tiny: Object detection and recognition using one stage improved model. In *2020 6th international conference on advanced computing and communication systems (ICACCS)*, pages 687–694. IEEE, 2020.

[2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[3] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[4] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107:104117, 2021.

[5] Y Lecun, L Bottou, GB Orr, and KR Müller. Efficient backprop. lncs, 1998.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

[8] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[9] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[10] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[11] 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.

[12] 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.

[13] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014.

[14] Etienne David, Simon Madec, Pouria Sadeghi-Tehran, Helge Aasen, Bangyou Zheng, Shouyang Liu, Norbert Kirchgessner, Goro Ishikawa, Koichi Nagasawa, Minhajul A 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. *Plant Phenomics*, 2020.

[15] 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.

[16] Etta D Pisano, Shuquan Zong, Bradley M Hemminger, Marla DeLuca, R Eugene Johnston, Keith Muller, M Patricia Braeuning, and Stephen M Pizer. Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms. *Journal of Digital imaging*, 11:193–200, 1998.