agronomy

MDPI

Article

Improving Out-of-Distribution Robustness for Wheat Head Detection: A Lightweight Modified YOLOv13 Approach

Peng Hui , Liyuan Zhang and Qingzhen Zhu *

School of Agricultural Equipment Engineering, Jiangsu University, Zhenjiang 212013, China; 2222116057@stmail.ujs.edu.cn (P.H.); liyuan.zhang@ujs.edu.cn (L.Z.)
* Correspondence: qingzhen_zhu@ujs.edu.cn

Abstract

Wheat head detection is a critical component in high-throughput phenotyping, holding significant application value for wheat yield estimation and breeding analysis. With the continuous advancement of general object detection models, state-of-the-art detectors achieve high accuracy in same-distribution wheat head detection scenarios. However, when applied to cross-distribution environments, their performance often degrades significantly. To investigate this issue, this paper adopts the official division of the GWHD2021 public dataset as the cross-distribution evaluation setting. Using the recently state-of-the-art object detection model YOLOv13 as the baseline, we systematically explore effective approaches to enhance cross-distribution generalization performance in wheat head detection. Specifically, we propose two lightweight modifications to YOLOv13's Full-PAD architecture and HyperACE's core modules, analyzing their potential mechanisms: (1) Replacing scalar gating in Full-PAD with channel-level gating enables finer-grained branch injection control. Concurrently, channel-level gating introduces equivalent stronger weight penalties during training, generating additional regularization effects. Through exploratory controlled experiments aligning weight-penalty strengths, we find that this gain depends on both channel decoupling and the accompanying implicit regularization rather than on decoupling alone. (2) Removing Batch Normalization from HyperACE's core C3AH modules and adopting normalization strategies independent of batch statistics—such as Identity, Group Normalization, or Instance Normalization. Preliminary experiments show that while this results in a small, directionally positive change, the change remains within run-to-run variance; we therefore do not claim BN removal as a reliable standalone improvement. Furthermore, combining channel gating with BN removal does not yield further additive improvements compared to channel gating alone, indicating an interaction effect. To address this, we analyze relevant statistics between channel gating and HyperACE branches, providing an exploratory explanation for this non-additive phenomenon. In summary, this paper delivers empirical evidence and preliminary insights for enhancing generalization performance in the cross-distribution wheat head detection task of the advanced general-purpose detection model YOLOv13.

Keywords: GWHD2021; wheat head detection; cross-distribution generalization; YOLOv13; channel gating; BN-free

Academic Editor: Jayme Garcia Arnal Barbedo

Received: 22 June 2026
Revised: 15 July 2026
Accepted: 16 July 2026
Published: 20 July 2026

1. Introduction

Wheat is a globally important crop and a primary food source for humans, with its production closely linked to global food security [1,2]. Phenotypic parameters of the head,

 https://doi.org/10.3390/agronomy16141370

such as number of heads or the grains per head, are closely related to yield estimation and variety improvement [3]. Achieving rapid and accurate detection and counting of wheat heads in field environments not only supports high-throughput phenotyping analysis, providing key inputs for yield prediction and growth monitoring, but also offers data support for precision agricultural management, resource allocation, and breeding decisions [4]. Traditional wheat head or other crop detection and counting methods primarily include manual counting and conventional image processing: manual counting is time-consuming, labor-intensive, and prone to subjective errors [5–7]; conventional image processing often relies on manually designed features such as texture and morphology, involving complex processes. In natural field environments, it is susceptible to interference from factors like light variation, complex backgrounds, similar colors of heads and leaves, overlapping heads, and changes in imaging scale, resulting in limited robustness [8–10]. In recent years, deep neural networks have become the mainstream method for detecting wheat heads and other crops. These networks learn discriminative features through data-driven learning and demonstrate outstanding performance in complex scene object detection [11–14].

Wheat head detection faces challenges such as complex backgrounds, dense occlusions, and small targets [15,16]. To address issues like missed detections of densely packed small objects and positioning difficulties caused by overlapping occlusions, existing research has primarily enhanced robustness through approaches including multi-scale detection and feature fusion [17], attention mechanism enhancement [18], localization loss modification [19,20], and detection head structure optimization [21,22]. For instance, Meng et al. [23] introduced a micro-scale detection layer within the YOLOv7 framework and integrated convolutional block attention module to enhance the object information while suppressing complex background interference; Dai et al. [24] expanded YOLOv5s' three-scale detection to four scales and introduced the efficient channel attention to enhance suitability for challenging field conditions; Xu et al. [25] incorporated attention mechanisms into YOLOv8 while adopting a dynamic detection head and improved localization loss to enhance small object localization accuracy in complex field environments; Zhenlin Yang et al. [26] applied RT-DETR to wheat head detection and combined it with a lightweight feature pyramid to enhance multi-scale fusion, thereby improving detection capabilities for occluded and overlapping heads; Yuting Zhang et al. [27] introduced a global edge information transfer module architecture based on lightweight YOLOv11n, employing an improved localization loss to stabilize regression of distant small objects and enhance robustness in densely occluded scenarios. Overall, these methods typically achieve significant improvements under identical or similar data collection conditions.

However, domain shifts—differences arising from cross-region, cross-year, cross-variety, and cross-platform data collection—are prevalent in real-world field applications. Specifically, wheat head detection faces severe domain shifts because data collected across different regions and years exhibit distinct phenotypic variations (e.g., ear shapes and awn lengths across varieties), drastically different lighting conditions, and varying background soil colors. When training and testing data are collected under these differing conditions (e.g., sensors/resolution, light intensity, viewing angles, growth stages, density), detection models often exhibit significant performance degradation. Mechanistically, this degradation occurs because deep neural networks are prone to "shortcut learning"—they tend to overfit to spurious domain-specific correlations (such as specific background soil colors) rather than learning the invariant causal features of wheat heads. More broadly, this is an instance of the domain generalization problem in computer vision, in which a model trained on one or more source domains must generalize to unseen target domains [28,29]. Regrettably, many existing wheat head detection studies still primarily aim to improve accuracy and efficiency within the same data distribution. Common approaches include training and testing on data collected under a single condition, randomly resampling the GWHD2021 [30] dataset to create new training/test splits, or reporting results only on the validation set. While these approaches help evaluate upper-bound performance under identical distributions, they inadequately characterize cross-distribution robustness. Out-of-distribution evaluation on GWHD2021 itself remains sparse: the few existing efforts [31,32] rely on classical two-stage or one-stage detectors and a per-domain accuracy metric, leaving the behavior of modern real-time detectors and architecture-level modifications under this shift largely unexplored. Consequently, research on performance degradation patterns, robustness assessment, and enhancement strategies across diverse environments provides both clear data benchmarks and direct practical value. This paper therefore focuses on evaluating and improving the generalization performance of wheat head detection across cross-distribution conditions.

This paper adopts the official partitioning of the GWHD2021 [30] public dataset as its research framework. The dataset is organized into multiple sub-datasets, each representing a group of data collected under identical conditions. The official partitioning of training/validation/test sets emphasizes cross-domain generalization evaluation, providing a reproducible benchmark for cross-distribution research. For the baseline model, we adopt the recently proposed YOLOv13 [33] as our subject of study. Its input resolution is aligned with the original resolution of GWHD2021 images to avoid information loss caused by additional scaling. Based on the official partitioning, we observe a significant generalization gap between the validation set and the test set. Additionally, we supplement performance comparisons under randomly partitioned same-distribution settings to quantify the difficulty of cross-distribution evaluation.

YOLOv13 introduces the Hypergraph-based Adaptive Correlation Enhancement mechanism (HyperACE) into the YOLO series [34,35] to model global multi-to-multi higher-order correlations and enhance cross-location and cross-scale features; it also proposes the Full-Pipeline Aggregation-and-Distribution paradigm (Full-PAD), which distributes and fuses the enhanced features from HyperACE across multiple stages of the backbone, neck, and head. To improve out-of-distribution (OOD) generalization performance while minimizing changes to the overall architecture and training workflow, this paper proposes two lightweight modifications centered on YOLOv13's Full-PAD fusion and HyperACE key blocks, and organizes the study around three hypotheses: (H1) replacing the scalar gating in Full-PAD with channel-level gating improves OOD generalization, through finer-grained channel decoupling together with an implicit regularization effect; (H2) reducing the dependence of HyperACE's key blocks on batch statistics—by removing Batch Normalization [36] and adopting batch-independent normalization such as Identity, Group Normalization [37], or Instance Normalization [38]—improves OOD generalization; and (H3) the two modifications are complementary, so that their gains are additive when combined. Because the OOD gains involved are small and detector training is stochastic, we evaluate the main configurations over an initial set of three repeated independent training runs. From exploratory experiments, we draw the following preliminary conclusions: First, regarding H1, channel-level gating showed a performance gain on the OOD test set, and an exploratory single-run weight-penalty-aligned ablation attributes this gain to both channel decoupling and implicit regularization. For H2, BN removal produces only a small, directionally positive change that lies within run-to-run variance, so we do not claim it as a reliable standalone improvement. H3 is not supported: the combination does not yield further additive gains compared to channel gating alone, indicating an interaction effect. To avoid subjective interpretation, this paper further calculates several metrics between channel gating and HyperACE branches. These exploratory statistics suggest that the current regularization strength and form may be insufficient to simultaneously stabilize the channel fluctuations caused by these two modifications, thereby limiting the overall performance gain. In summary, this paper presents a controlled empirical baseline for the GWHD2021 official cross-domain evaluation, along with two lightweight modifications and an analysis of their interaction, providing practical guidance for improving cross-distribution robustness of object detection models in agricultural vision scenarios.

The main contributions of this paper are as follows:

(1) We establish a controlled empirical baseline for YOLOv13 under the official GWHD2021 cross-distribution partitioning scheme, evaluated over initial three consecutive independent training runs (reported as mean ± standard deviation), and provide a comparison with the random partitioning (i.e., independent and identically distributed) to quantify the generalization gap.

(2) We preliminarily observe that replacing scalar gating with channel-level gated fusion produces a performance gain on the OOD test set, and through an exploratory weight-penalty-aligned ablation, we hypothesize that this gain depends on both finer-grained channel decoupling and an implicit regularization effect.

(3) We reveal a non-additive interaction between channel gating and BN-free modifications. On the one hand, we report that BN removal alone does not yield a reliable gain; on the other hand, we provide an exploratory explanation based on gating and branch statistics, offering empirical conclusions for combining fusion gating and normalization strategies under cross-domain conditions.

2. Methods

2.1. Overview of the YOLOv13 Architecture

This paper uses YOLOv13 as the baseline detector. Unlike the traditional YOLO "backbone–neck–head" architecture [39–45], YOLOv13 introduces a HyperACE (Hypergraph-based Adaptive Correlation Enhancement) module on the multi-scale features (denoted as B3, B4, and B5) from the baseline network to model cross-scale and cross-location feature correlations and generate enhanced features; subsequently, through the Full-PAD (Full-Pipeline Aggregation and Distribution) paradigm, the enhanced features are distributed and fused at multiple locations within the network to further strengthen feature representation. The overall architecture of YOLOv13 is shown in Figure 1.

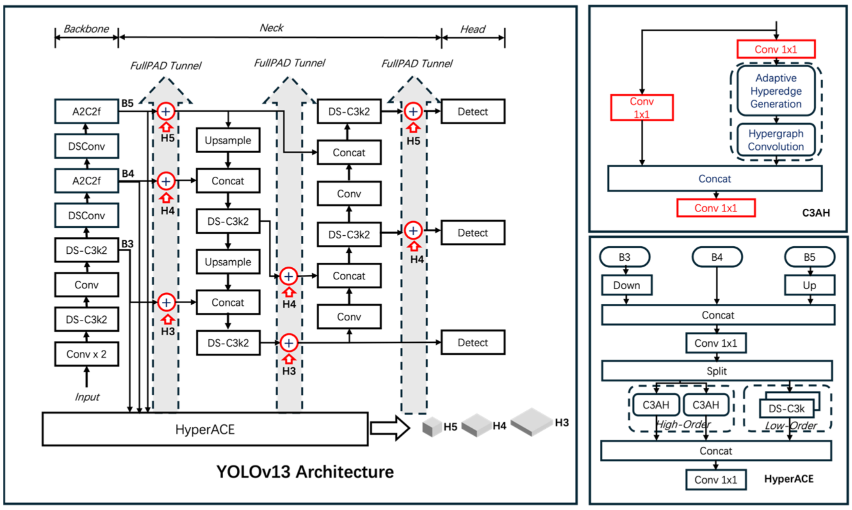


Figure 1. Overall architecture of YOLOv13 and the positions of HyperACE/FullPAD.

After receiving multi-scale features from the backbone, HyperACE simultaneously models local and global information through multiple branch paths and outputs enhanced feature maps for subsequent distribution. Full-PAD, on the other hand, establishes fusion channels (FullPAD Tunnels) at multiple locations—including the connection between the backbone and neck, the internal layers of the neck, and the connection between the neck and head—to inject the HyperACE output into the corresponding layer features after scale and channel alignment. Subsequent modifications in this paper primarily focus on the gated fusion mechanism of Full-PAD and the normalization strategy of the HyperACE core block, as indicated by the red areas in Figure 1.

2.2. Channel-Level Gated Fusion in the FullPAD Tunnel

In the FullPAD Tunnel of YOLOv13, the original features F and the enhanced features H are fused via gated fusion to produce the output $\widetilde{F}$, as shown in Figure 2. The original implementation uses scalar gating:

$$\widetilde{F} = F + \gamma H \tag{1}$$

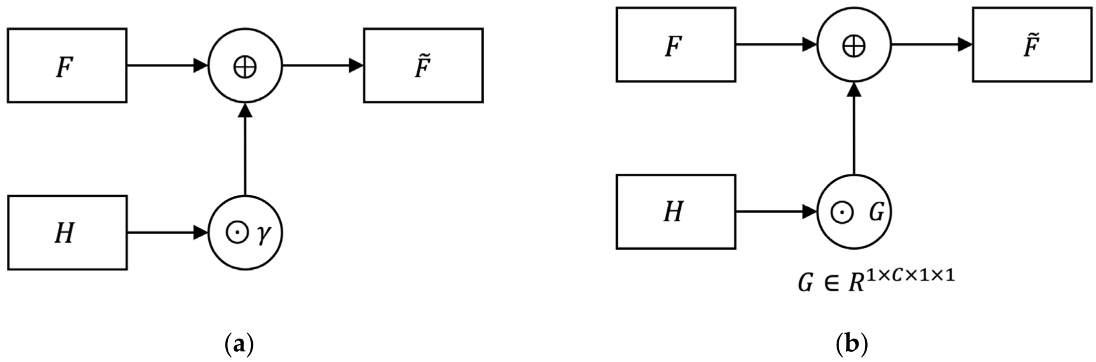


Figure 2. Comparison of FullPAD tunnel gating methods: (**a**) scalar gating, $\widetilde{F} = F + \gamma H$; (**b**) channel gating, $\widetilde{F} = F + G \odot H$.

Here, γ is a learnable scalar used to control the overall injection strength of the augmentation branch. This formulation applies a uniform scaling to all channels, ignoring differences in semantic contributions and stability across channels.

To improve the fine-grained control of the fusion, this paper replaces scalar gating with channel-level gating:

$$\widetilde{F} = F + G \odot H,\ G \in \mathbb{R}^{1\times C\times 1\times 1} \tag{2}$$

Here, $\odot$ denotes element-wise multiplication (applied via broadcasting across the batch and spatial dimensions), and G represents channel-wise learnable gating coefficients. Channel-level gating enables the decoupling of input strengths across different channels: the network can adaptively amplify channels with higher contributions based on the training signal, while suppressing channels with lower contributions or those that are unstable for the task or correlated with background responses. In scenarios involving multi-source data and cross-distribution evaluation, this fine-grained adjustment mechanism may be more effective in improving generalization performance.

From the optimization perspective, let the loss be L, and let the gradient backpropagated to the fusion point be $U = \partial L/\partial \widetilde{F}$. Then, the gradients for the scalar gate and the channel gate are respectively

$$\frac{\partial L}{\partial \gamma} = \sum_{c,h,w} U_{c,h,w} H_{c,h,w} \tag{3}$$

$$\frac{\partial L}{\partial g_c} = \sum_{h,w} U_{c,h,w} H_{c,h,w} \tag{4}$$

Scalar gates are updated by aggregating the gradients across all channels, whereas channel gates allow each channel to independently adjust its injection strength based on its own gradient, making it easier to suppress injection from low-contribution or unreliable channels.

In addition, channel-level gating introduces implicit changes in the strength of regularization. When using weight decay (L2 or decoupled weight decay in AdamW), the penalty for the gating term can be written as

$$WD_{scalar} = \lambda \cdot \gamma^2 \tag{5}$$

$$WD_{channel} = \lambda \cdot \sum_c g_c^2 \tag{6}$$

Here, λ is the decay coefficient. When the order of magnitude of the channel-specific gating terms is comparable to that of the scalar gating term, the total penalty from channel-specific gating becomes stronger, thereby imposing a more significant regularization constraint on the branches. In the experimental section, we will exploratorily distinguish between the contributions of the "channel decoupling" and "implicit regularization" mechanisms by ablating the alignment of gating decay strengths.

2.3. BN-Free Strategy for the HyperACE Core Block

In YOLOv13's HyperACE, C3AH is responsible for modeling global high-order correlations and serves as the core component of HyperACE. In the original implementation, Batch Normalization (BN) is applied after the convolutional layers of C3AH for normalization. During training, BN normalizes features using mini-batch statistics, while during testing, it uses the moving mean and variance accumulated during training. In out-of-distribution (OOD) scenarios, there is often a shift in feature statistics between the training and testing domains, causing the moving statistics used by BN during testing to mismatch the current domain. This may lead to normalization mismatch in BN, which affects generalization performance. To reduce reliance on batch statistics and training-domain moving statistics, this paper adopts a BN-free strategy in the C3AH block: one approach is to directly replace BN with an identity map; the other is to replace BN with normalization methods that do not rely on batch statistics, such as IN (Instance Normalization) or GN (Group Normalization).

Let the input feature be $x \in \mathbb{R}^{N \times C \times H \times W}$. For the BN on channel c, the mean and variance during training are defined as

$$\mu_c = \frac{1}{NHW} \sum_{n,h,w} x_{n,c,h,w} \tag{7}$$

$$\sigma_c^2 = \frac{1}{NHW} \sum_{n,h,w} (x_{n,c,h,w} - \mu_c)^2 \tag{8}$$

The output of BN is

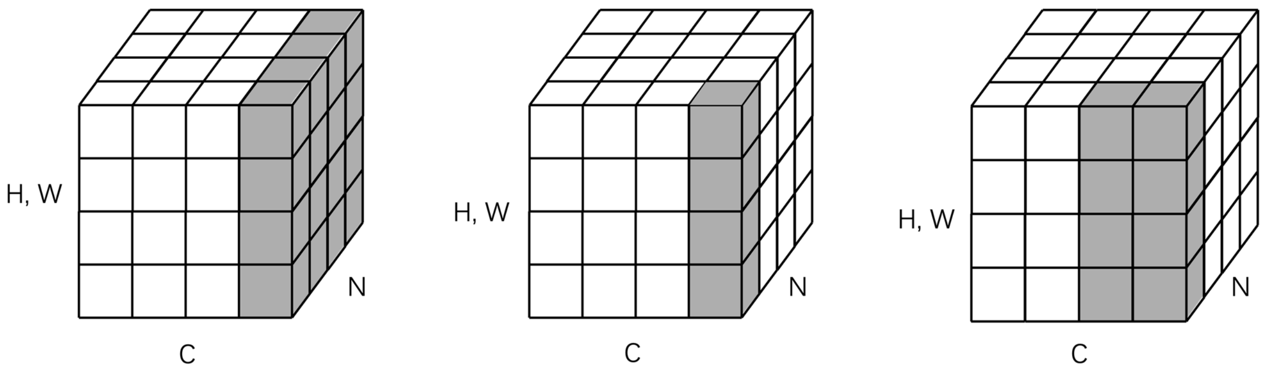
$$\mathrm{BN}(x_{n,c,h,w}) = \alpha_c \cdot \frac{x_{n,c,h,w} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}} + \beta_c \tag{9}$$

Here, α_c and β_c are learnable affine parameters, and ϵ is a numerical stability term. During the testing stage, the cumulative moving statistics $\left(\hat{\mu_c}, \hat{\sigma_c^2}\right)$ accumulated during training are typically substituted for (μ_c, σ_c^2). When there are statistical differences between the testing domain and the training domain, $\left(\hat{\mu_c}, \hat{\sigma_c^2}\right)$ may fail to accurately characterize the feature distribution of the current domain, thereby causing a normalization offset and affecting the response and fusion of subsequent modules.

This paper adopts the lightest BN-free approach, which involves replacing the BN layer with an identity map—that is, simply removing the BN layer:

$$y = x \tag{10}$$

This modification eliminates the original parameters and computational overhead, completely removing the reliance on batch statistics and moving statistics. In addition to directly removing BN, we also tested two alternative normalization methods that do not depend on batch-level statistics: Instance Normalization (IN) and Group Normalization (GN). IN calculates the mean and variance of spatial statistics only on a single sample and a single channel; GN divides the channels into G groups, calculates the mean and variance for each sample and each group within the $(C/\mathrm{G}) \times H \times W$ range, and performs normalization (as shown in Figure 3). A common feature of both is that their statistical scope does not include the batch dimension, thereby avoiding the risk of mismatch caused by moving statistics from the training domain during cross-domain testing.



(**a**) (**b**) (**c**)

Figure 3. Schematic diagram of the statistical scope for BN/IN/GN: (**a**) BN uses batch statistics, while (**b**) IN/(**c**) GN computes statistics within a single sample.

We focus this modification on C3AH because it is the core block that models global high-order correlations over the entire feature map, so its post-convolution BN statistics are particularly exposed to the train–test distribution shift described above. In implementation, we replaced all Batch Normalization (BN) layers originally used in C3AH—specifically at the 1×1 convolutions of the two branches and the 1×1 convolution after concatenation (three locations in total)—with Identity, IN, or GN with varying numbers of groups. In the main experiments, we carry the lightest variant—Identity (direct BN removal)—through our multi-run analysis, since it is the variant that enters the interaction study; the comparison among Identity, IN, and GN is exploratory and single-run, and is reported in Appendix A (Table A1). These experiments examine the impact of removing batch-statistic dependence in C3AH on OOD generalization.

3. Experiments

To evaluate our proposed modifications and their underlying mechanisms, we conducted controlled exploratory experiments. In Section 3.1, we describe the implementation, including the dataset and hyperparameters. In Section 3.2, we evaluate our modifications on the official split and on the split of the same distribution. In Section 3.3, we conduct an exploratory single-run ablation to examine whether the channel-gating gain reflects channel decoupling, implicit regularization, or both.

3.1. Dataset and Experimental Setup

We conducted our experiments using the GWHD2021 public dataset. GWHD2021 is an expansion of GWHD2020: GWHD2020 was constructed by integrating multi-source data from nine institutions across seven countries and three continents; GWHD2021 further supplements the original dataset with images collected by the same institutions across different years, locations, and a broader range of growth stages, adds data from five additional countries, and corrects some annotations from GWHD2020 while removing low-quality images. Furthermore, unlike GWHD2020, which combined images of multiple growth stages from the same location into a single sub-dataset, GWHD2021 adopts a finer-grained classification standard: images from the same experimental unit, the same acquisition session, and the same vector and sensor are treated as a single sub-dataset. Consequently, the 11 sub-datasets from GWHD2020 have been subdivided into 20 sub-datasets in GWHD2021, with 27 new sub-datasets added, bringing the total to 47 sub-datasets.

In terms of data scale, GWHD2021 consists of a total of 6515 images (including three duplicates; 6512 after deduplication). The official split includes a training set of 3657 images (3655 after deduplication), a validation set of 1476 images, and a test set of 1382 images (1381 after deduplication). More importantly, the official GWHD2021 split we adopt exhibits a pronounced cross-domain gap: the validation set consists entirely of GWHD2020 data, the training set is primarily GWHD2020 with only a small amount of new data, and the test set consists entirely of new data, with images from the United States, Mexico, and Sudan appearing only in the test set. This produces substantial distribution differences, so that models performing well on the training and validation sets can degrade markedly on the test set, making this split well suited to our cross-distribution study. We note that the dataset providers additionally define a separate GlobalWheat-WILDS split recommended specifically for distribution-shift studies [31]; evaluating our modifications under that protocol is left to future work.

In addition, we also evaluate the model's performance under the same distribution to characterize its overall detection capabilities. To maintain dataset integrity, we do not exclude any samples. Instead, we perform a random split on the complete, deduplicated dataset (6512 images) using proportions similar to the official split: a training set of 3656 images, and validation and test sets of 1428 images each (the sizes of the validation and test sets are taken as the average of the official validation and test set sizes). To standardize the evaluation data, the random split was generated using a fixed random seed.

In terms of implementation details, we based our work on the official YOLOv13 code, which we modified. YOLOv13 offers model sizes such as Nano, Small, Large, and Extra-Large. Balancing accuracy and computational complexity, we selected YOLOv13-Small (S) as the baseline for our experiments. Because our proposed channel gating and BN-free modifications alter specific internal mechanisms rather than global network capacity adjustments, exploring their transferability to other YOLOv13 variants (e.g., Nano or Large) represents an important theoretical direction, which we leave to future work for empirical confirmation. The training strategy aligns with the official default settings: 600 epochs of training (with early stopping if no improvement is observed for 100 consecutive epochs), using SGD as the optimizer with an initial learning rate of 0.01 and linear decay, along with a linear warm-up during the first three epochs. Data augmentation also follows the official approach, utilizing Mosaic and MixUp. The only difference lies in the input resolution: YOLOv13's default input for COCO is 640×640, whereas wheat ears are mostly small objects. To minimize information loss caused by scaling and better align with the data characteristics, this paper uniformly adopts the original resolution of GWHD2021 (1024 × 1024) as the input, applying this setting to both the baseline and all improved versions.

Because the cross-distribution improvements under study are small relative to run-to-run training variability, we do not rely on a single training run. Instead, the main configurations are evaluated over an initial set of three consecutive training runs with independent random initializations, and we report the mean ± standard deviation over the three runs. The in-distribution comparison, by contrast, uses the single fixed-seed random split described above and is reported from a single run. This asymmetry is intentional: the in-distribution test serves merely as a sanity check to verify that our lightweight modifications do not degrade the model's fundamental fitting capacity, allowing us to allocate our primary computational resources to the more variance-sensitive cross-distribution evaluations. Furthermore, intermediate epoch-by-epoch training logs were not archived, restricting our evaluation to final test-time performance without explicit convergence diagnostics.

In terms of hardware, we use a single NVIDIA GPU (RTX 5090) for training on a cloud computing platform. Since the model modifications are minor, they introduce mathematically minimal parameter changes, maintaining identical FLOPs and per-epoch computational costs across all model versions.

3.2. Main Results Under In-Distribution and Out-of-Distribution Conditions

We first compare the performance differences in the baseline model under in-distribution and out-of-distribution settings, as shown in Table 1. Under in-distribution random splitting, the model achieves high accuracy on both the validation and test sets (exceeding 93% in mAP@0.5), indicating that recent general-purpose object detection models have already demonstrated strong practicability in in-distribution wheat head detection tasks. However, under the official GWHD2021 split (cross-distribution), the model exhibits a significant performance gap between the validation and test sets: the test set mAP@0.5 drops from 93.7% to 74.1%, and the mAP@0.75 shows an even more pronounced decline (from 54.8% to 25.6%). These results indicate that cross-domain distribution differences under the official split lead to a significant degradation in detection performance, suggesting that cross-domain head detection still holds ample scope for further research.

Table 1. Performance comparison of the YOLOv13 baseline model under in-distribution (single run) and out-of-distribution (3 runs) settings (%).

		mAP@0.5	**mAP@0.75**	**mAP@[0.5:0.95]**
In-Distribution	Validation set	93.6	60.0	56.3
	Test set	93.3	59.7	56.2
Official Split (Out-of-Distribution)	Validation set	93.7 ± 0.4	54.8 ± 2.5	53.2 ± 1.1
	Test set	74.1 ± 1.6	25.6 ± 1.7	33.9 ± 1.2

We then evaluated the performance of the two types of lightweight modifications and their combinations on the official cross-distribution test set, as shown in Table 2. Here, "Channel Gating" refers to replacing all scalar gating in Full-PAD with channel-level gating; "Identity" refers to removing the BN in the C3AH branches of HyperACE; and "Channel Gating + Identity" refers to adopting both modifications simultaneously. To strictly control variables, both channel gating and scalar gating use gate parameters initialized to 0, making the model rely more on the backbone–neck–head main path during the early training phase while ensuring consistent initial norms for both gating methods. Table 2 reports each configuration as the mean ± standard deviation over three independent runs. Channel gating yields a preliminary performance gain on the cross-distribution test set, raising the mean mAP@0.5 by 2.7 points and mAP@0.75 by 3.5. BN removal (Identity), by contrast, produces only a small mean increase (mAP@0.5 + 0.7) relative to the run-to-run variance, so we do not treat it as a reliable standalone gain. Combining the two does not yield further additive improvements compared to channel gating alone (75.7 vs. 76.8 mAP@0.5), indicating an interaction effect. We provide exploratory statistics and an interpretation of this non-additive behavior in the discussion (Section 4). Furthermore, given the presence of run-to-run variance, larger-scale experiments will be necessary in future work to rigorously verify these preliminary trends.

Table 2. Performance comparison of the baseline model and improved versions on the official cross-distribution test set (%).

Model	mAP@0.5	mAP@0.75	mAP@[0.5:0.95]
YOLOv13	74.1 ± 1.6	25.6 ± 1.7	33.9 ± 1.2
Channel Gating	76.8 ± 0.5	29.1 ± 0.4	36.4 ± 0.3
Identity	74.8 ± 2.1	26.4 ± 1.4	34.6 ± 1.3
Channel Gating + Identity	75.7 ± 0.5	26.9 ± 1.4	35.0 ± 0.7

Finally, to verify that the modifications do not compromise detection performance under the same distribution, we present the results for each version on a dataset randomly split under the same distribution (Table 3). As can be seen, the differences between versions are minimal, indicating that the modifications in this paper primarily improve robustness under cross-distribution conditions without significantly altering the upper-bound performance under the same distribution.

Table 3. Performance comparison of the baseline model and improved versions on the in-distribution test set (%).

Model	mAP@0.5	mAP@0.75	mAP@[0.5:0.95]
YOLOv13	93.3	59.7	56.2
Channel Gating	93.5	59.8	56.4
Identity	93.5	59.5	56.3
Channel Gating + Identity	93.4	59.6	56.2

3.3. Sources of Gains from Channel-Level Gating

In Section 2.2, we noted that the cross-distribution gains from channel-level gating may stem from two sources: (1) fine-grained branch injection control enabled by channel decoupling; and (2) an enhanced implicit weight penalization effect resulting from the increased dimensionality of channel gate parameters. To exploratorily distinguish between these two mechanisms, we further designed two sets of comparative experiments:

Channel decoupling only: Channel gating is employed, but the weight decay coefficients of the gating parameters are scaled by the number of channels so that the total weight penalty scale roughly aligns with scalar gating.

Weight penalty enhancement only: Scalar gating is retained, but its weight decay coefficients are amplified by the number of channels so that the total weight penalty scale roughly aligns with channel gating.

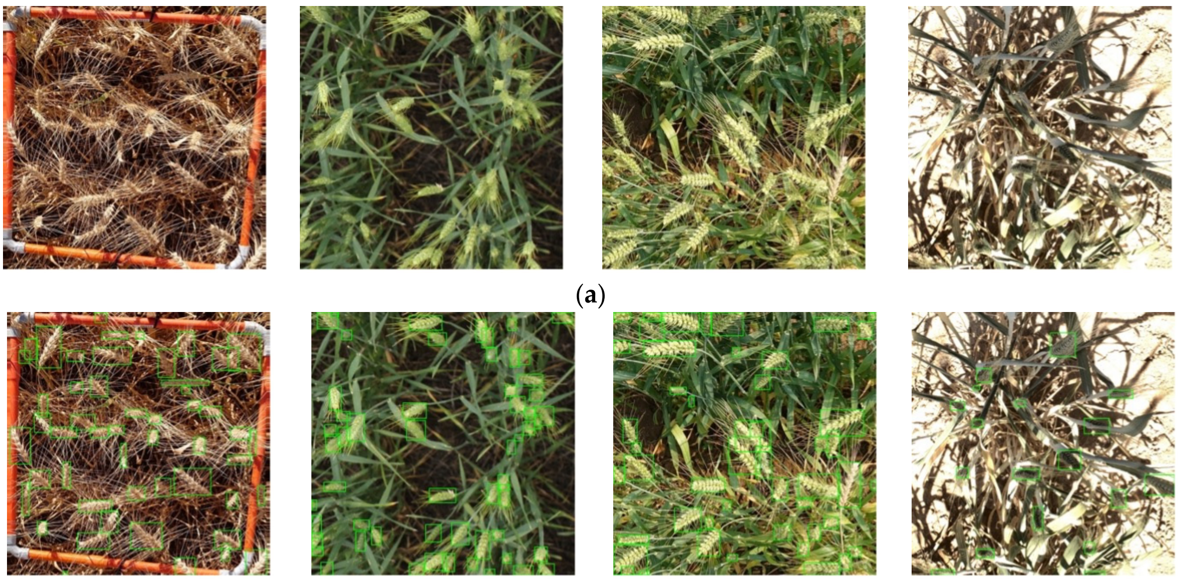
The results are shown in Table 4. (For the baseline and channel-gating configurations, the values are the median of the three runs in Table 2; the two ablation variants were each trained once. Consequently, Table 4 serves merely to qualitatively explore potential trends.) Performance improves when moving from scalar gating to “scalar gating + enhanced weight penalty”; introducing channel gating on top of this (which combines channel decoupling with implicit regularization) yields even better performance. Conversely, when channel decoupling was introduced alone while realigning the weight penalty to the scalar gate strength, model performance (75.1%) fell back into the noise margin of the baseline (75.0%). This preliminarily suggests that the benefit of channel decoupling is contingent on the accompanying increase in regularization rather than arising independently.

Table 4. Ablation comparison of channel gating mechanism (official cross-distribution test set, %).

Model	mAP@0.5	mAP@0.75	mAP@[0.5:0.95]
Scalar Gating (i.e., YOLOv13)	75.0	26.1	34.4
Channel Gating	76.9	29.3	36.6
Channel Gating (weight penalty divided by the number of channels)	75.1	25.1	34.1
Scalar Gating (weight penalty multiplied by the number of channels)	75.7	28.3	35.7

3.4. Visual Analysis

To provide a more intuitive understanding of the typical error types observed in the baseline model and to illustrate where our modifications take effect, we purposively selected several representative examples from the official GWHD2021 test set where visual improvements were most pronounced. As explicitly noted, these visual results only illustrate qualitative tendencies in these specific scenarios; they supplement rather than establish the quantitative conclusions. Figure 4 presents the comparison results under identical inference settings (imgsz = 1024, conf = 0.25, NMS IOU = 0.7). Each column in the figure corresponds to a single example, listed from top to bottom as follows: (a) original image, (b) manual annotations (Ground Truth), (c) predictions from the baseline YOLOv13s model, (d) predictions from the channel-gated model, and (e) predictions from the Identity (BN-free) model.



(**a**)

(**b**)

Figure 4. *Cont.*

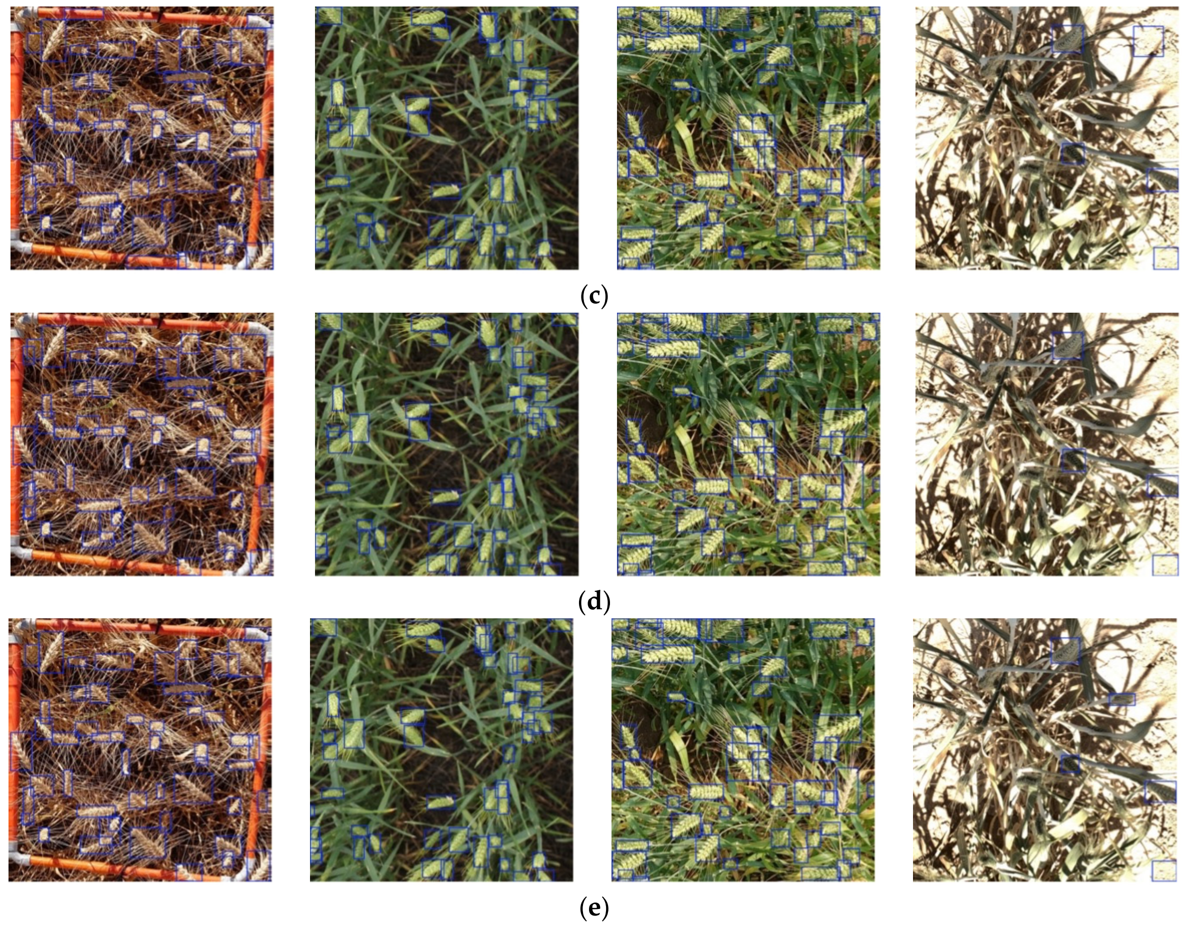


Figure 4. Visual comparison of examples from the official GWHD2021 cross-distribution test set (imgsz = 1024, conf = 0.25, NMS IOU = 0.7). Each column represents a single instance; from top to bottom: (**a**) original image, (**b**) Ground Truth, (**c**) YOLOv13s, (**d**) Channel Gating, (**e**) Identity (removing BN).

Example 1 (first column, complex background/similar texture scenario): The baseline model tends to misclassify objects in the background, such as pipes and leaves, as wheat heads; the Identity version reduces some background misclassifications in this example, while channel gating further suppresses additional background misclassifications and improves detection of obscured or blurred heads in the top-left corner.

Example 2 (second column, highly blurred and edge-truncated scenario): The image is generally blurred, and only partial edges of some heads are visible; the baseline model misses many instances; both improved models successfully detected some of the missed objects in this example, but they still struggled with areas of severe overlap and occlusion.

Example 3 (third column, overlapping scenario): The image is relatively clear, but some heads of wheat overlap, resulting in insufficient detection by the baseline model; the improved models achieved more complete detection of the overlapping areas, demonstrating a certain degree of enhanced robustness in this example.

Example 4 (fourth column, challenging background and high-brightness interference scenario): The baseline model produces false positives for high-brightness soil clumps and exhibits low overall recall; both the channel gating and Identity versions reduce some false positives in the high-brightness background in this example and detect more heads.

4. Discussion

To further explore why the combination of channel-level gating and BN-free (Identity) does not yield additive gains, we analyzed the cross-sample variability of HyperACE's output features on the official test set, as well as the branch injection intensity at each injection point of FullPAD. Note that extracting these post hoc diagnostic statistics across the test set requires loading the saved final model weights. Because we did not archive the weights for all repeated runs, these statistics are calculated using a single representative run (the median run) for each configuration. To contextualize these metrics and prevent confusion with the mean values in Table 2, the specific mAP@0.5 scores for these median runs are Baseline (75.0%), Identity (75.4%), Channel Gating (76.9%), and Combined (75.6%). For HyperACE's output H, we first calculated the mean value of each image in channel c:

$$m_{i,c} = \frac{1}{HW}\sum_{h,w}|H_{i,c,h,w}| \tag{11}$$

Accordingly, the inter-sample coefficient of variation for channel c is defined as

$$\mathrm{CV}_c = \frac{\mathrm{Std}_i(m_{i,c})}{\mathrm{Mean}_i(m_{i,c}) + \epsilon} \tag{12}$$

Finally, by taking the average across the channel dimension, we obtain

$$\mathrm{CV}_{channel} = \mathrm{Mean}_c(\mathrm{CV}_c) \tag{13}$$

This is used to characterize the relative level of fluctuation in the HyperACE output across the channel dimension. On the other hand, to quantify the injection strength of the enhanced branch in Full-PAD, we define the injection ratio:

$$\rho = \mathrm{E}\left[\frac{\|\mathrm{gate} \odot H\|_1}{\|H\|_1 + \epsilon}\right] \tag{14}$$

Specifically, we calculate the average of the ratio of the first-order norms of the feature maps before and after gating for each sample (where the gate is either a scalar gate or a channel-level gate).

Table 5 presents the statistical results for the $\mathrm{CV}_{channel}$ values across different models. Compared to the baseline YOLOv13 (0.30), both removing BN (0.32) and channel-level gating (0.34) increase the channel-level volatility of HyperACE's output; when the two are combined, the volatility further increases, reaching the maximum value among the four settings (0.37). These results are consistent with an interaction between BN removal and channel-level gating at the channel level: the combined setting shows the largest channel-level fluctuation, introducing higher uncertainty into the subsequent Full-PAD fusion.

Table 5. Channel fluctuation in HyperACE output feature maps across different models.

Model	CV_Channel_Mean
YOLOv13	0.30
Identity	0.32 (+0.02)
Channel Gating	0.34 (+0.04)
Identity + Channel Gating	0.37 (+0.07)

Table 6 further illustrates the injection ratios ρ at each injection point for FullPAD. It can be observed that channel gating substantially reduces the branch injection ratios at all injection points, with a decrease of up to roughly an order of magnitude at high-injection points such as P1 and P3, while the overall average injection ratio decreases by approximately 4–5 times (from 0.25–0.26 to about 0.046–0.057). This indicates that channel-level gating tends to impose stronger injection constraints on the enhancement branch after decoupling the channels. At the same time, within the same gating type (scalar gating or channel gating), the effect of removing BN on ρ at most injection points is relatively limited; however, the differences exhibit a certain degree of injection-point specificity, with more pronounced changes potentially occurring at individual injection points.

Table 6. FullPAD branch injection ratios for different models (P1–P7 represent the seven fusion injection points in the order of the network's forward inference).

Model	P1	P2	P3	P4	P5	P6	P7	Mean
YOLOv13	0.6035	0.4529	0.3608	0.0210	0.2993	0.0048	0.0065	0.2498
Identity	0.3667	0.4380	0.5190	0.0960	0.2903	0.0999	0.0167	0.2609
Channel Gating	0.0481	0.1409	0.0352	0.0131	0.1408	0.0199	0.0002	0.0569
Identity + Channel Gating	0.0384	0.1034	0.0287	0.0099	0.1282	0.0127	0.0003	0.0459

Combined with the exploratory gating ablation in Section 3.3 (where channel gating fails to demonstrate gains when the regularization penalty is aligned with the scalar gate strength, but performance partially recovers upon increasing the gating-related regularization penalty), we preliminarily hypothesize that the generalization benefits of channel gating depend on sufficiently strong regularization constraints to suppress the channel-level instability it introduces; when combined with Identity, channel fluctuations in the enhanced branches further increase, and the original regularization strength may be insufficient to stabilize both sources of fluctuation simultaneously, resulting in the combination failing to yield additive benefits. It is important to note that channel gating and Identity may introduce different types of statistical variations, and a single regularization term may not be able to suppress both simultaneously. Future work could systematically investigate the relationship between different regularization methods and different types of fluctuations without altering the main architecture, and explore stronger gating regularization or more suitable no-batch statistical stabilization techniques (e.g., applying magnitude constraints to HyperACE outputs or imposing structured sparsity on gating parameters) to verify whether both sources of fluctuation can be suppressed simultaneously.

5. Conclusions

To address the issue of generalization performance degradation in cross-distribution wheat head detection, this paper establishes an out-of-distribution (OOD) evaluation setup based on the official cross-distribution split of the GWHD2021 dataset. Using the advanced real-time object detection model YOLOv13 as a baseline, we propose two lightweight modifications while minimizing changes to the architecture and training workflow.

First, we replace the scalar gating in FullPAD with channel-level gating, achieving a preliminary performance gain in OOD testing. Exploratory ablation results suggest that these gains stem from both the finer-grained injection control enabled by channel decoupling and variations in the strength of gating-related regularization (weight decay); second, we remove the Batch Normalization (BN) layer from the key blocks of HyperACE (replacing it with an identity map) and compare it with batch-free statistics strategies such as IN/GN (Appendix A). Removing BN yields only a small mean increase relative to the run-to-run variance, so we do not treat it as a reliable standalone improvement. Additionally, we find that the combination of channel-level gating and removing BN does not yield additive gains. Exploratory statistical analysis of fluctuations in HyperACE output channels and injection ratios at various injection points in FullPAD reveals that both types of modifications increase the relative channel-level fluctuations in the enhanced branch, with the greatest fluctuations observed when combined. Meanwhile, channel-level gating substantially reduces injection intensity at each injection point. Building on our preliminary hypothesis that the benefits of channel-level gating rely on stronger regularization constraints, we speculate that under combined settings, the current regularization strength and form might be insufficient to simultaneously stabilize fluctuations from different sources, thereby limiting overall gains.

Overall, using the official cross-distribution split of GWHD2021—which itself spans many regions, years, and acquisition platforms—this paper presents two low-cost, easily implemented modifications and analyzes their interaction. Notably, channel-level gating demonstrates a preliminary performance gain in cross-distribution robustness for wheat head detection. The evaluation is nonetheless limited to a single detector size (YOLOv13-S) and to the official Challenge split rather than the GlobalWheat-WILDS protocol, so transfer to other detectors, to the WILDS protocol, and to larger models remains future work. Within this scope, the results are relevant to applications such as high-throughput phenotyping, breeding evaluation, and yield estimation.

Author Contributions: Conceptualization, P.H.; methodology, P.H.; software, P.H.; validation, P.H., L.Z. and Q.Z.; formal analysis, P.H.; investigation, P.H.; resources, L.Z. and Q.Z.; data curation, P.H.; writing—original draft preparation, P.H.; writing—review and editing, Q.Z.; visualization, P.H.; supervision, L.Z. and Q.Z.; project administration, L.Z. and Q.Z.; funding acquisition, Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Jiangsu Province (Grant No. BK20230548) and was a project funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD-2023-87).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

As an exploratory complement to the main analysis, we examined other batch-independent normalizations in the C3AH blocks. We replaced BN with Instance Normalization (IN) and Group Normalization (GN, with eight and one groups). The key difference among these approaches lies in their statistical scope: IN performs statistics on a single channel of a single sample, GN ($g = 8$) performs statistics across channels within a group, and GN ($g = 1$) performs statistics across all channels of a single sample. In Table A1, the YOLOv13 baseline and the Identity variant are the median of the three runs reported in Table 2, whereas IN and GN were each trained once. Given the single-run nature of the IN and GN experiments, and considering the substantial run-to-run variance observed in the main text, we cannot treat these point estimates as a reliable ranking. We report them only to indicate that none of the batch-independent alternatives we tried produced a clearly larger gain than removing BN, which is why the main text focuses on the Identity variant.

Table A1. Performance comparison of different BN replacement strategies (official cross-distribution test set, %).

Model	mAP@0.5	mAP@0.75	mAP@[0.5:0.95]
YOLOv13	75.0	26.1	34.4
Identity	75.4	27.4	35.2
IN	75.5	28.1	35.7
GN ($g = 8$)	74.6	26.0	34.2
GN ($g = 1$)	75.4	27.5	35.2

References

1. Wei, L.; Yang, H.; Niu, Y.; Zhang, Y.; Xu, L.; Chai, X. Wheat biomass, yield, and straw-grain ratio estimation from multi-temporal UAV-based RGB and multispectral images. *Biosyst. Eng.* **2023**, *234*, 187–205. [CrossRef]
2. Zhang, Q.; Chen, Q.; Xu, W.; Xu, L.; Lu, E. Prediction of feed quantity for wheat combine harvester based on improved YOLOv5s and weight of single wheat plant without stubble. *Agriculture* **2024**, *14*, 1251. [CrossRef]
3. Fernandez-Gallego, J.A.; Kefauver, S.C.; Gutiérrez, N.A.; Nieto-Taladriz, M.T.; Araus, J.L. Wheat ear counting in-field conditions: High throughput and low-cost approach using RGB images. *Plant Methods* **2018**, *14*, 22. [CrossRef] [PubMed]
4. Madec, S.; Jin, X.; Lu, H.; De Solan, B.; Liu, S.; Duyme, F.; Heritier, E.; Baret, F. Ear density estimation from high resolution RGB imagery using deep learning technique. *Agric. For. Meteorol.* **2019**, *264*, 225–234. [CrossRef]
5. Khan, Z.; Shen, Y.; Liu, H. Object detection in agriculture: A comprehensive review of methods, applications, challenges, and future directions. *Agriculture* **2025**, *15*, 1351. [CrossRef]
6. Jiang, L.; Wang, Y.; Wu, C.; Wu, H. Fruit distribution density estimation in YOLO-detected strawberry images: A kernel density and nearest neighbor analysis approach. *Agriculture* **2024**, *14*, 1848. [CrossRef]
7. Duan, Y.; Han, W.; Guo, P.; Wei, X. YOLOv8-GDCI: Research on the phytophthora blight detection method of different parts of chili based on improved YOLOv8 model. *Agronomy* **2024**, *14*, 2734. [CrossRef]
8. Zuo, Z.; Gao, S.; Peng, H.; Xue, Y.; Han, L.; Ma, G.; Mao, H. Lightweight detection of broccoli heads in complex field environments based on LBDC-YOLO. *Agronomy* **2024**, *14*, 2359. [CrossRef]
9. Zhang, F.; Chen, Z.; Ali, S.; Yang, N.; Fu, S.; Zhang, Y. Multi-class detection of cherry tomatoes using improved YOLOv4-tiny model. *Int. J. Agric. Biol. Eng.* **2023**, *16*, 225–231. [CrossRef]
10. Pei, H.; Sun, Y.; Huang, H.; Zhang, W.; Sheng, J.; Zhang, Z. Weed detection in maize fields by UAV images based on crop row preprocessing and improved YOLOv4. *Agriculture* **2022**, *12*, 975. [CrossRef]
11. Wang, A.; Xu, Y.; Hu, D.; Zhang, L.; Li, A.; Zhu, Q.; Liu, J. Tomato yield estimation using an improved lightweight YOLOv11n network and an optimized region tracking-counting method. *Agriculture* **2025**, *15*, 1353. [CrossRef]
12. Wang, J.; Gao, Z.; Zhang, Y.; Zhou, J.; Wu, J.; Li, P. Real-time detection and location of potted flowers based on a ZED camera and a YOLOv4-tiny deep learning algorithm. *Horticulturae* **2021**, *8*, 21. [CrossRef]
13. Ji, W.; Pan, Y.; Xu, B.; Wang, J. A real-time apple targets detection method for picking robot based on ShuffleNetV2-YOLOX. *Agriculture* **2022**, *12*, 856. [CrossRef]
14. Ji, W.; Gao, X.; Xu, B.; Pan, Y.; Zhang, Z.; Zhao, D. Apple target recognition method in complex environment based on improved YOLOv4. *J. Food Process Eng.* **2021**, *44*, e13866. [CrossRef]
15. Guan, Y.; Pan, J.; Fan, Q.; Yang, L.; Yin, X.; Jia, W. CTWheatNet: Accurate detection model of wheat ears in field. *Comput. Electron. Agric.* **2024**, *225*, 109272. [CrossRef]
16. Guan, S.; Lin, Y.; Lin, G.; Su, P.; Huang, S.; Meng, X.; Liu, P.; Yan, J. Real-time detection and counting of wheat spikes based on improved YOLOv10. *Agronomy* **2024**, *14*, 1936. [CrossRef]
17. Li, R.; Sun, X.; Yang, K.; He, Z.; Wang, X.; Wang, C.; Wang, B.; Wang, F.; Liu, H. A lightweight wheat ear counting model in UAV images based on improved YOLOv8. *Front. Plant Sci.* **2025**, *16*, 1536017. [CrossRef] [PubMed]
18. Li, R.; Wu, Y. Improved YOLOv5 wheat ear detection algorithm based on attention mechanism. *Electronics* **2022**, *11*, 1673. [CrossRef]
19. Guan, Y.; Pan, J.; Fan, Q.; Yang, L.; Xu, L.; Jia, W. Generalized focal loss WheatNet (GFLWheatNet): Accurate application of a wheat ear detection model in field yield prediction. *Agriculture* **2024**, *14*, 899. [CrossRef]
20. Yan, X.; Ma, Q.; Feng, Q. A method for detecting the number of wheat ears based on the improved YOLOX model. *Remote Sens. Lett.* **2024**, *15*, 423–432. [CrossRef]
21. Shen, X.; Zhang, C.; Liu, K.; Mao, W.; Zhou, C.; Yao, L. A lightweight network for improving wheat ears detection and counting based on YOLOv5s. *Front. Plant Sci.* **2023**, *14*, 1289726. [CrossRef] [PubMed]
22. Bi, Z.; Li, Y.; Guan, J.; Zhang, X. Real-time wheat detection based on lightweight deep learning network RepYOLO model. *INMATEH—Agric. Eng.* **2024**, *72*, 601–610. [CrossRef]
23. Meng, X.; Li, C.; Li, J.; Li, X.; Guo, F.; Xiao, Z. YOLOv7-MA: Improved YOLOv7-based wheat head detection and counting. *Remote Sens.* **2023**, *15*, 3770. [CrossRef]
24. Dai, F.; Xue, Y.; Huang, L.; Huang, W.; Zhao, J. Application of efficient channel attention and small-scale layer to YOLOv5s for wheat ears detection. *J. Indian Soc. Remote Sens.* **2024**, *52*, 1751–1759. [CrossRef]
25. Xu, J.; Sun, Y.; Zhang, S.; Sun, D.; Xiang, Z. Wheat-YOLO: A real-time and high precision object detection for wheat. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*; IEEE: Piscataway, NJ, USA, 2024; pp. 2165–2170.
26. Yang, Z.; Yang, W.; Yi, J.; Liu, R. WH-DETR: An efficient network architecture for wheat spike detection in complex backgrounds. *Agriculture* **2024**, *14*, 961. [CrossRef]
27. Zhang, Y.; Liu, Z.; Guo, X.; Li, C.; Teng, G. Wheat head detection in field environments based on an improved YOLOv11 model. *Agriculture* **2025**, *15*, 1765. [CrossRef]

28. Zhou, K.; Liu, Z.; Qiao, Y.; Xiang, T.; Loy, C.C. Domain generalization: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 4396–4415.
29. Wang, J.; Lan, C.; Liu, C.; Ouyang, Y.; Qin, T.; Lu, W.; Chen, Y.; Zeng, W.; Yu, P.S. Generalizing to Unseen Domains: A Survey on Domain Generalization. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 8052–8072.
30. David, E.; Serouart, M.; Smith, D.; Madec, S.; Velumani, K.; Liu, S.; Wang, X.; Pinto, F.; Shafiee, S.; Tahir, I.S.A.; et al. Global wheat head detection 2021: An improved dataset for benchmarking wheat head detection methods. *Plant Phenomics* **2021**, *2021*, 9846158. [CrossRef] [PubMed]
31. Koh, P.W.; Sagawa, S.; Marklund, H.; Xie, S.M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R.L.; Gao, I.; et al. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *International Conference on Machine Learning (ICML)*; PMLR: Cambridge, MA, USA, 2021; pp. 5637–5664.
32. Jena, S.; Rajendran, S.S.; Seemakurthy, K.; Poornachari, P. Domain penalisation for improved Out-of-Distribution Generalisation. *arXiv* **2024**, arXiv:2408.01746.
33. Lei, M.; Li, S.; Wu, Y.; Hu, H.; Zhou, Y.; Zheng, X.; Ding, G.; Du, S.; Wu, Z.; Gao, Y. YOLOv13: Real-time object detection with hypergraph-enhanced adaptive visual perception. *arXiv* **2025**, arXiv:2506.17733.
34. Jocher, G.; Qiu, J. Ultralytics YOLO11. Available online: https://github.com/ultralytics/ultralytics (accessed on 5 May 2026).
35. Tian, Y.; Ye, Q.; Doermann, D. YOLOv12: Attention-centric real-time object detectors. *arXiv* **2025**, arXiv:2502.12524.
36. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2015; pp. 448–456.
37. Wu, Y.; He, K. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*; European Computer Vision Association: Cham, Switzerland, 2018; pp. 3–19.
38. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Instance normalization: The missing ingredient for fast stylization. *arXiv* **2016**, arXiv:1607.08022.
39. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
40. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
41. Jocher, G. Ultralytics YOLOv5. Available online: https://github.com/ultralytics/yolov5 (accessed on 5 May 2026).
42. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO series in 2021. *arXiv* **2021**, arXiv:2107.08430.
43. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
44. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2023; pp. 7464–7475.
45. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. Available online: https://github.com/ultralytics/ultralytics (accessed on 5 May 2026).