

Article

UAV Based Weed Pressure Detection Through Relative Labelling

Sebastiaan Verbesselt¹, Rembert Daems², Axel Willekens^{1,2} and Jonathan Van Beek^{1,*}

¹ Technology and Food Science Unit, Flanders Research Institute for Agriculture, Fisheries and Food (ILVO), 9820 Merelbeke-Melle, Belgium; axel.willekens@ugent.be (A.W.)

² IDLab-Imec, Ghent University, 9052 Zwijnaarde, Belgium; rembert.daems@ugent.be

* Correspondence: jonathan.vanbeek@ilvo.vlaanderen.be

Highlights

What are the main findings?

- A novel labelling approach using relative differences in weed pressure was developed to train a CNN ordinal regression model for weed detection.
- The model achieved strong performance, successfully detecting weed pressure gradients in potato fields.

What is the implication of the main finding?

- The proposed method significantly reduces data labelling time and effort while maintaining high prediction accuracy.
- The approach enables flexible, site-specific weed management decisions, supporting more sustainable and environmentally friendly agricultural practices.

Abstract

Agricultural management in Europe faces increasing pressure to reduce its environmental footprint. Implementing precision agriculture for weed management could offer a solution and minimize the use of chemical products. High spatial resolution imagery from real time kinematic (RTK) unmanned aerial vehicles (UAV) in combination with supervised convolutional neural network (CNNs) models have proven successful in making location specific treatments. This site-specific advice limits the amount of herbicide applied to the field to areas that require action, thereby reducing the environmental impact and inputs for the farmer. To develop performant CNN models, there is a need for sufficient high-quality labelled data. To reduce the labelling effort and time, a new labelling method is proposed whereby image subsection pairs are labelled based on their relative differences in weed pressure to train a CNN ordinal regression model. The model is evaluated on detecting weed pressure in potato (*Solanum tuberosum* L.). Model performance was evaluated on different levels: pairwise accuracy, linearity (Pearson correlation coefficient), rank consistency (Spearman's (r_s) and Kendal (τ) rank correlations coefficients) and binary accuracy. After hyperparameter tuning, a pairwise accuracy of 85.2%, significant linearity ($r_s = 0.81$) and significant rank consistency ($r_s = 0.87$ and $\tau = 0.69$) were found. This suggests that the model is capable of correctly detecting the gradient in weed pressure for the dataset. A maximum binary accuracy and F1-score of 92% and 88% were found for the dataset after thresholding the predicted weed scores into weed versus non-weed images. The model architecture allows us to visualize the intermediate features of the last convolutional block. This allows data analysts to better evaluate if the model "sees" the features of interest (in this case weeds). The results indicate the potential of ordinal regression with relative labels as a fast, lightweight model that predicts weed pressure gradients. Experts have the freedom to decide which threshold value(s) can be used on



Academic Editor: Kevin Tansey

Received: 21 August 2025

Revised: 26 September 2025

Accepted: 10 October 2025

Published: 15 October 2025

Citation: Verbesselt, S.; Daems, R.; Willekens, A.; Van Beek, J. UAV Based Weed Pressure Detection Through Relative Labelling. *Remote Sens.* **2025**, *17*, 3434. <https://doi.org/10.3390/rs17203434>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

predicted weed scores depending on the weed, crop and treatment that they want to use for flexible weed control management.

Keywords: ordinal regression; CNN; relative labels; weed detection; potato; UAV

1. Introduction

Intensive crop production in Europe relies on high chemical input of plant protection products (PPP), which has a high negative impact on environment, biodiversity and human health. Additionally, incorrect use of these PPP can lead to resistant diseases, pests and weeds [1]. More sustainable use of PPP is necessary, and as herbicide application accounts for one third of PPP use in Europe, site-specific weed control (SSWC) could provide an answer. The detection of weeds with images can be converted into prescription maps and applied only where weeds are present [2,3].

Recent advances in weed detection are related to the use of high spatial resolution imagery (<1 cm) from GNSS-RTK equipped unmanned aerial vehicles (UAVs), as they can provide detailed weed maps. The combination with deep learning models has gained momentum as recent studies have shown the potential of commercial low-cost UAVs and open-source software packages in making prescription maps of SSWC [4].

Convolutional neural networks (CNN) are popular deep learning models for computer vision tasks, such as instance and semantic segmentation, object detection and image classification. Their use in recognizing weeds in UAV images has increasingly grown since 2013 as (i) off-the-shelf models and (ii) GPU cloud computing becomes more accessible [5–7]. However, CNNs struggle with the high variability in agricultural datasets and this remains a challenge when building robust deep learning models. The variability can be caused both by variation in the combinations of crops and weeds (i.e., occlusion of crops and weeds, color and texture similarities between crop and weeds, inter-class similarities, intra-class variations because of growth stage variability or phenotypic plasticity) or a variation in the data acquisition (i.e., illumination, shadow effects or motion blur).

CNN models account for this variability with an increased amount of trained parameters and become large, requiring more data collection, labelling and training [6,8]. Researchers have tried to solve this time and energy consuming practice by building available, open-source datasets of different crop and weed species [2,3,9], by generating training data using data-augmentation [2,5,6,9–12] or generating synthetic data [13]. However, even after training on large datasets, models must be retrained to obtain adequate accuracy on their own datasets. Lightweight models, fast labelling and retraining methods to “(over)fit” the model to the dataset of interest are used to overcome this, either through (i) the use of other learning methods (unsupervised, semi-supervised learning, active learning) [2,8,11], (ii) using knowledge of other developed models that performed similar tasks (transfer learning [3,5,6,11–14] and few-shot learning [15–17]), or (iii) by combining different strategies of training data generation, fast labeling and retraining.

In this paper, we propose a CNN ordinal regression model with pairwise ranking to address this problem. Ordinal regression models are categorical but have an ordered structure with distances between ranks being undefined. This represents a combination of classification models and regression models, allowing us to benefit from the best of both worlds. This was visualized in Figure 1 in comparison to common deep learning computer vision tasks. CNN ordinal regression models have been successfully used in a broad range of applications such as facial age estimation [18–22], image aesthetics [22] and relative visibility estimation [23] in foggy image datasets. To the best of our knowledge, we

are the first to use CNN ordinal regression models for weed pressure prediction on UAV image data.

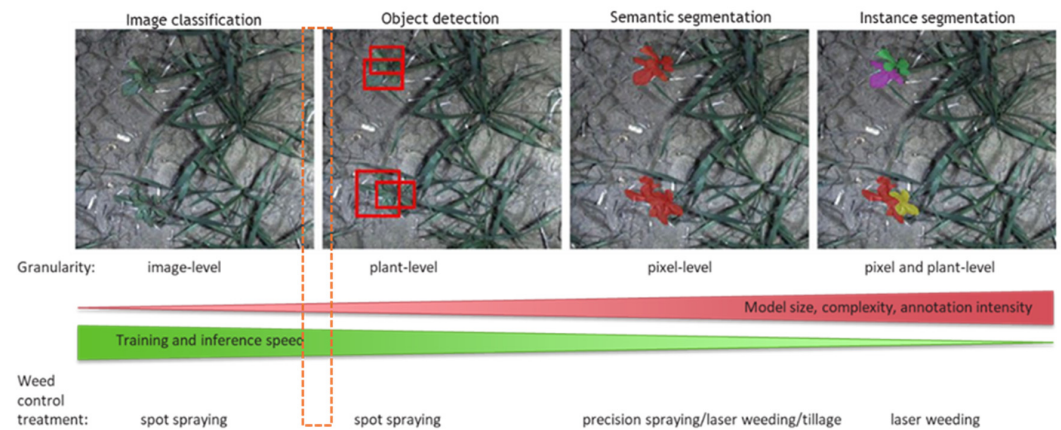


Figure 1. Visualization the most common deep learning computer vision tasks and the trade-offs between level of training/inference speed and model complexity. The newly proposed model (ordinal regression model) is based on these trade-offs, located in between image classification and object detection (indicated by the orange dotted rectangle). Figure adapted from [2].

2. Materials and Methods

2.1. The Dataset

2.1.1. UAV Data Acquisition

The data was collected with two UAV flights on the 27 and 30 May 2022 at an experimental field of the Flanders Research Institute for Agriculture, Fisheries and Food (ILVO) ($50^{\circ}58'57.04''N$, $3^{\circ}46'38.97''E$). The experimental field was uniformly planted and predetermined areas (red boxes in Figure 2) were left untreated to ensure natural distribution of weeds and weed pressure variability throughout the field (Figure 2).



Figure 2. Test field with potato crop, located in Merelbeke, Belgium. Red areas within the field remained untreated with herbicide to artificially introduce weeds within the parcel. The orthomosaic was created using Agisoft Metashape 1.5.5, the map in QGIS 3.22 [24,25].

Images were collected with a DJI M600 PRO (DJI, Shenzhen, China) platform that was equipped with a Sony Alpha 7III camera (Sony, Tokyo, Japan) with a 50 mm lens (FE 50 mm F/1.8 Sony, Tokyo, Japan). A gimbal, type DJI Ronin-MX (DJI, Shenzhen, China), was used to stabilize the camera and keep it in nadir-looking position. The flights took place at an altitude of 20 m to obtain a spatial resolution of 0.2 cm/pixel. During the flights, images of 3984×2656 pixels were taken with 70% overlap.

2.1.2. Data Preprocessing

UAV images were cropped into small image subsections of 256×256 pixels, using Python 3.10 [26]. This resulted in 2350 image subsections of which 14,622 subsections pairs were pairwise labelled and divided into 11,697 (80%) and 2925 (20%), which were used for training and validation of the model, respectively (Figure 3).

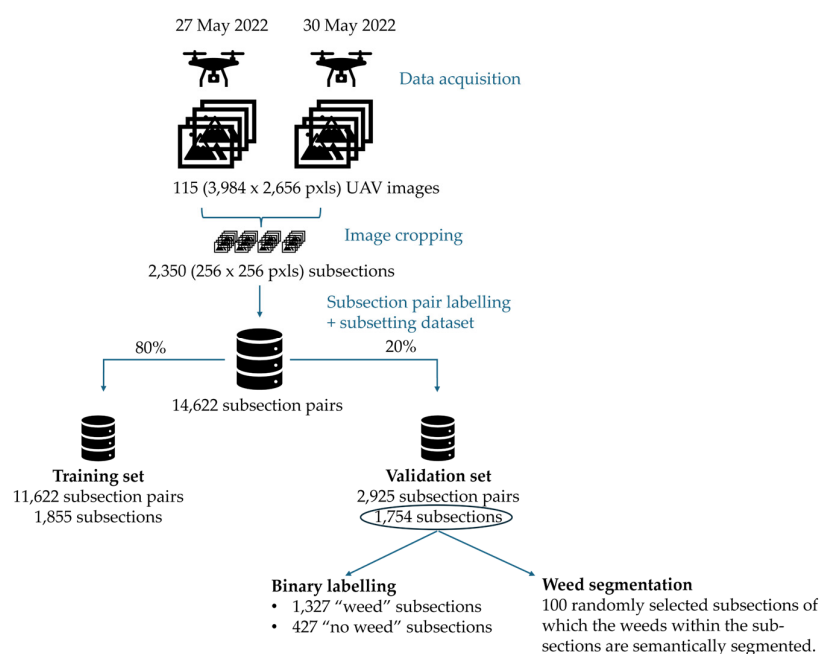


Figure 3. Overview of the steps taken to create the datasets, from data acquisition, image cropping and labelling.

The subsections of the validation set (1754 subsections forming 2925 pairs) were further binary classified into labels 0 (no weed present, do not apply herbicide) and 1 (weeds present, apply herbicide). Of the validation set, 100 subsections were randomly selected for segmentation of weed pixels (Figure 4). This was done by first converting the red, green and blue values of the image into reflectance values, followed by calculating the Excess Green (ExG) index of each pixel [27], as shown in Step B of Figure 4. For this index, a threshold of 0.05 was used, based on visual inspection, to separate pixels into vegetation and non-vegetation pixels (binary tile, Step C in Figure 4). These preprocessing steps were conducted with Python 3.9 in Jupyter Notebook v5 [28–31].

Pixels of the binary raster subsections were manually corrected in QGIS so that only weed pixels received a value of 1, while pixels of potato plants and non-vegetated pixels have a value of 0. From these corrected segmented tiles, the weed pressure was calculated as the ratio of weed pixels over the total amount of vegetated pixels per tile (Step E in Figure 4). The correction on the segmented subsections was carried out with the use of the Serval Plugin [32].

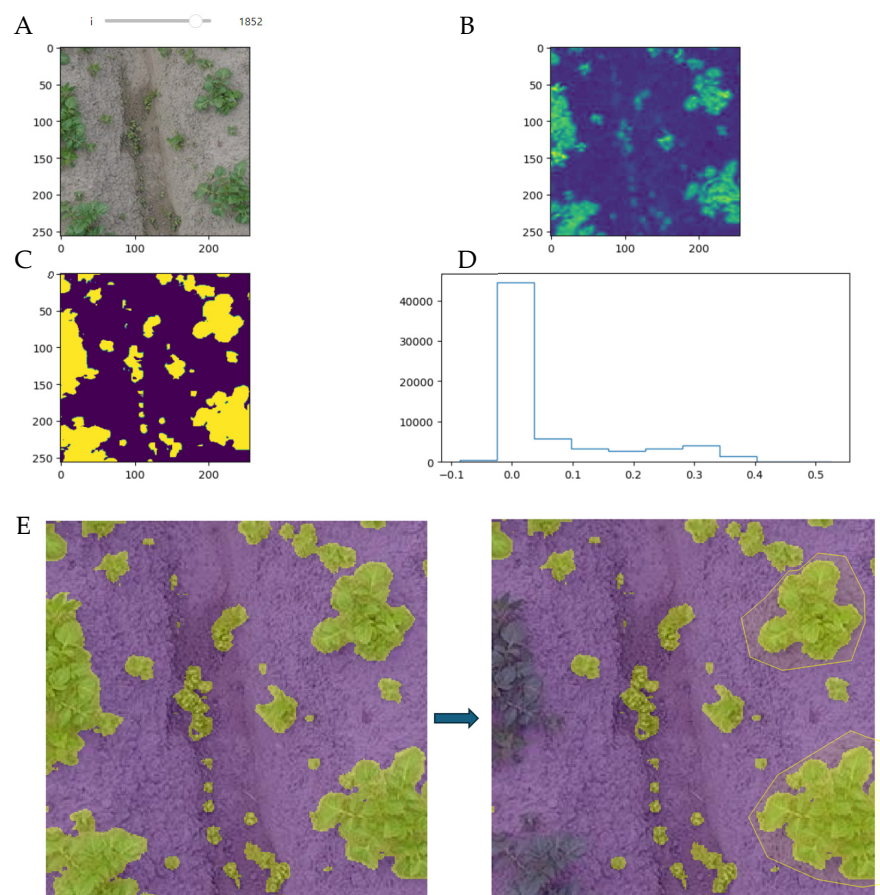


Figure 4. The subsection processing steps for semantic segmentation of the weed pixels. (A) is the original image subsection. (B) is the ExG index for each pixel, with dark blue to brighter green pixels indicating the gradient from low to high values. (C) is the binary subsection after thresholding, with yellow pixels of 1 (ExG values > 0.05) and dark purple values of 0 (ExG values ≤ 0). (D) is the histogram of ExG values of the subsection. (E) is the corrected segmented weed plants.

The model is trained based on pairwise ranking the images: for each image subsection pair, an annotator has to indicate the image with the highest weed pressure. An example for weed pressure in maize is given in Figure 5.

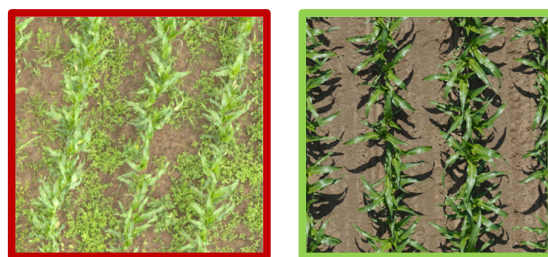


Figure 5. An example of an image subsection pair for which the left image appears to have the highest weed pressure of both images and should be ranked accordingly.

2.2. Ordinal Regression Model

The deep learning model for ordinal regression is implemented using PyTorch [33] and PyTorch Lightning 2.0.3 [34] in Python. It is designed for comparing image subsection pairs to learn a ranking function based on a relative attribute—the amount of weed present in the image. When an image is passed through the model, a theoretical value is computed as “weed pressure” on an ordinal scale. This z-score will be used throughout the model.

The training was also performed with and without data augmentation. Data augmentation is a method of artificially increasing the training set by creating modified copies of the “true” data within the training set. When data augmentation was required, images from the training dataset were randomly flipped vertically and horizontally 50% of the time, while color channels were randomly inverted 20% of the time. This last transformation was added to make sure the model “learns” to recognize weeds based on more than “green” color on a brown background.

2.2.1. Architecture

The model is a CNN which can be divided in different structures:

1. The initial convolutional blocks:

The number of convolutional blocks is configurable as hyperparameter, ranging from two to six blocks. Each convolutional block consists of (i) a 2D convolutional layer for feature extraction (like edges, textures, etc. form the input data), (ii) a 2D max-pooling layer to down sample the feature map, (iii) a group normalization layer (to separate the channels into four groups) and (iv) a ReLU (rectified linear unit) activation function to introduce non-linearity. The number of filters (feature maps) can be further defined as hyperparameters for the first convolutional layer (default value of 32) and subsequent convolutional layers (default value of 64 for convolutional layer two up to six, depending on the total number of specified convolutional blocks).

2. The final convolutional layer:

The last convolutional layer either outputs a single channel (default setting) or continues to a fully connected head. In the latter case, the model has two additional fully connected layers (with a ReLU activation function in between) after the convolution layers.

The final convolutional layer is used to process the final feature maps into a single output, representing the relative ranking of the image.

3. The receptive field calculation:

In order to understand the learning process of the model better, a code was developed to calculate and visualize the receptive field of the last convolutional layer before the final output. When an image x passes through the convolutional blocks (forward pass), the receptive field can be returned for this last stage (if wanted). This allows us to determine how much of the input image that each output unit is “seeing” and helps to ensure proper design for spatial dependencies in the images. Its size is determined by the chosen kernel sizes, strides and dilations within the convolutional backbone.

2.2.2. Optimizer and Loss Functions

1. The Adam optimizer is used to learn the weights and biases. It combines the benefits of Adaptive Gradient Descent and RMSprop, whereby it maintains separate learning rates for each parameter and updates them based on both the first and second moments of the gradients. The default learning rate can be specified as hyperparameter, with a default value of 0.001.

2. The loss function used is a form of the Hinge Loss function. The type of Hinge Loss can be set as a hyperparameter, but the default value is set to 2. This exponentiates the difference, making it a squared hinge loss (L2) by default. Hinge Loss is commonly used for ranking or classification tasks. In this case, it computes the difference (d) between two output values z_1 and z_0 , which represent the relative ranking (i.e., the amount of weeds in image 1 vs. image 0). The function applies the ReLU ($1 - d$) operation, which ensures that the loss is only computed when z_1 is less than z_0 by more than 1, encouraging the model to rank the images correctly.

2.3. Model Performance Evaluation

For the evaluation of the model, several metrics have been proposed:

- During training, the squared hinge loss of each batch of image pairs and the pairwise accuracy are calculated for both the training and validation set. The pairwise accuracy is the number of instances where the pairwise ranks are correctly predicted divided by the total number of instances. The pairwise accuracy could also be calculated for an independent test set but was not used in this case.
- For the 100 segmented image subsections, the linearity and the rank consistency were evaluated by determining the Pearson, Spearman's rank and Kendal rank correlation coefficients between the predicted weed score (z) per image subsection and the number of pixels segmented as "weed".
- Lastly, the validation set was binary labelled into subsections with label 1 (weeds present) and 0 (no weeds present). F1-scores, overall accuracy, precision and recall were calculated for a range of different thresholds on the predicted z -scores (ranging from 0.5 to 1.6, with a step of 0.01, based on the structure of this dataset) to evaluate classification performance. A Mann–Whitney U test with asymptotic approximation with tie correction was used to see if the image subsections with label 0 had a significantly higher predicted weed (z) score compared to image subsections with label 1. Statistical tests were performed with the SciPy package [35].

3. Results

3.1. Model Performance

3.1.1. The Pairwise Accuracy

Model training was conducted for different hyperparameter settings. Pairwise accuracy was used as a metric for model optimization during training (Equation (1)). For all image subsections x_i (with labelled weed pressure rank z_i) and images subsections x_j (with a labelled weed pressure rank z_j), z_i having a higher rank than z_j , the pairwise accuracy of an image pair (x_i, x_j) is:

$$\text{pairwise accuracy} = A = \frac{\sum_{i < j} 1((z_i > z_j) \Leftrightarrow (\hat{z}_i > \hat{z}_j))}{\sum_{i < j} 1} \quad (1)$$

whereby \hat{z}_i and \hat{z}_j are the predicted ranks of x_i and x_j , respectively. The hyperparameters were iteratively evaluated and the hyperparameters of the best pairwise accuracy for the validation dataset is shown in Appendix A.

Training with data augmentation was performed without an early stopping rule (so trained for 100 epochs). The model had a slightly lower maximum pairwise validation accuracy (85.6% versus 86.6%) compared to the model without data augmentation. The latter already reached this validation accuracy within 32 epochs. At the end of their training, both models had a pairwise validation accuracy of 85.2%. Pairwise training accuracy was 90.7% and 87.0% for training without (and early stopping rule) and with data augmentation, respectively. The results are shown in Figure 6. The following results that will be discussed come only from the non-augmented model since no improvement was found for training with data augmentation. A shorter training time is additionally more interesting for applications in agriculture, where processing times must be kept to the minimum.

The pairwise accuracy is calculated by subtracting the predicted weed score of the image subsection with the lower labelled weed pressure from the predicted weed score of the image subsection with the higher labelled weed pressure. This pairwise subtraction or "distance" is visualized in Figure 7.



Figure 6. Results of the training with (purple line) and without (cyan line) data augmentation on the “Weights & Biases” website. From top left to bottom right: evolution of the pairwise training accuracy, pairwise validation accuracy, training loss and validation loss. The transparent lines give the calculated values for each step, while the solid line gives the Gaussian average of the curve by factor 10 (for training) and 1 (for validation).

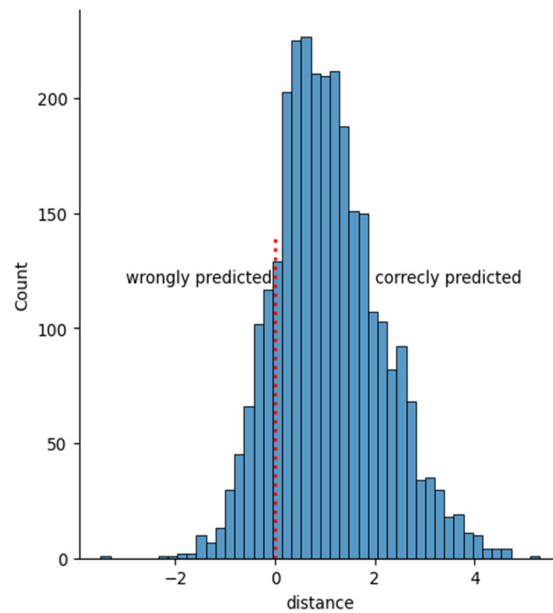


Figure 7. Prediction histogram of pairwise distances on validation dataset after training. Image pairs what have a negative or positive distance are “wrongly” or “correctly” classified by the model, respectively. The proportion that is correctly classified is 85.2%.

3.1.2. Binary Validation

The binary labelling of the validation dataset into subsections with label 1 (weeds present—apply herbicide) and 0 (no weeds present—do not apply herbicide) results in 1327 subsections with label 1 and 427 subsections with label 0. The subsections with label 1 had different rates in weed pressure. The binary scores were compared to the predicted weed pressure (z) scores from the model (Figure 8). Image subsections with label 0 had a significantly lower predicted weed pressure score compared to image subsections

with label 1 (Table 1). F1-score, overall accuracy, precision and recall were calculated for different threshold values for the z-scores, simulated from 0.5 up to 1.6 in increments of 0.01. (Figure 9). The highest values for F1-score (0.92) and overall accuracy (0.88) were found at a threshold of 0.76 and 0.89, respectively.

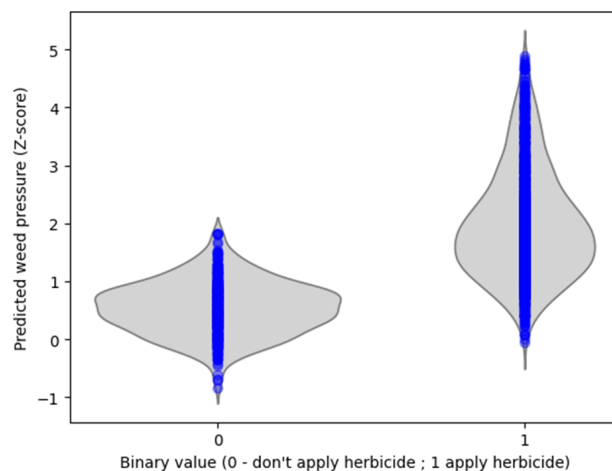


Figure 8. Violin plots of the predicted weed pressure (z-score, *y*-axis) versus binary label for herbicide application (*x*-axis) for the validation dataset.

Table 1. Results of the Mann–Whitney U test statistic with asymptotic approximation for predicted weed pressure (z-score) between the two groups.

| No Herbicide | Herbicide |
|-----------------|-------------------------|
| N1 = 427 | N2 = 1328 |
| U1 = 35,617 | U2 = 531,012 |
| Median = 0.54 | Median = 1.85 |
| IQR = 0.62 | IQR = 1.22 |
| <i>p</i> -value | 4.99×10^{-163} |

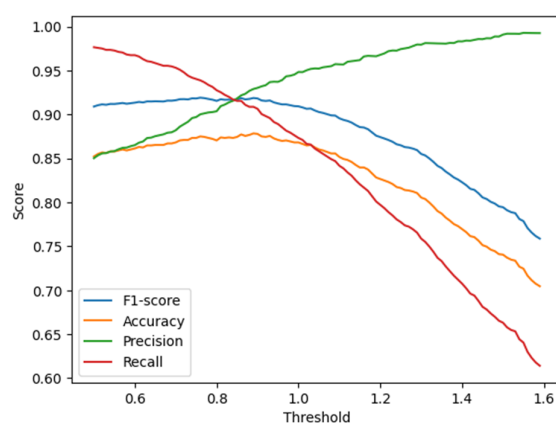


Figure 9. Evolution of precision, recall, overall accuracy and F1-score (*y*-axis) for different threshold values for the predicted z-scores (*x*-axis).

3.1.3. Rank Consistency

The rank consistency of the predicted weed scores (*z*) was evaluated by comparing these scores with the calculated percentage of weed pixels for 100 randomly selected image subsections. The scatterplot is shown in Figure 10. The association was evaluated for linearity (Pearson correlation coefficient, r_p) and monotonic relationship (Spearman’s r_s) &

Kendall (τ) rank correlation coefficients). The three coefficients were significant, with very strong estimated effect sizes (Table 2). This indicates a linear and monotonic association between the two values.

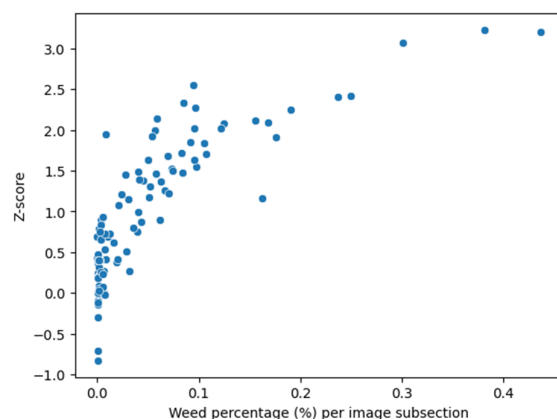


Figure 10. Scatterplot of the predicted z-values (y -axis) and the calculated weed percentage after segmentation (x -axis) for 100 randomly selected image tiles from the validation dataset.

Table 2. The correlation coefficients between the predicted weed scores (z) and the calculated weed percentage after segmentation, for 100 randomly selected image subsections from the validation dataset.

| Statistic | Symbol | Estimate | p -Value ($\alpha = 0.05$) |
|---|--------|----------|-----------------------------------|
| Pearson correlation coefficient | r_p | 0.81 | 2.61×10^{-24} |
| Spearman's rank correlation coefficient | r_s | 0.87 | 2.01×10^{-32} |
| Kendall rank correlation coefficient | τ | 0.69 | 2.38×10^{-24} |

3.2. Receptive Field Calculation: Peeking into the Black Box

The model architecture enables the visualization of the receptive field of the last convolutional block before the final prediction. This receptive field represents the specific portion of the input image that influences the activation of a neuron (or node) in a feature map. By computing and analyzing the receptive field, data analysts gain valuable insights into which regions of the image contribute most to the model's decision-making process. This is particularly crucial in deep learning, where models are often considered "black boxes" due to their complex and non-transparent internal representations. Understanding the receptive field helps in diagnosing model behavior, improving interpretability, and ensuring that predictions are based on relevant image features rather than spurious correlations or artifacts. Ultimately, this enhances trust in the model's decisions and can guide further refinement to improve performance and reliability. An example is given in Figure 11, whereby the input image (left panel) is compared with receptive field (middle panel). The receptive field with a size of 178×178 pixels indicates regions with higher values as yellow (presence of detected features) and regions with lower values as dark blue (absence of features). The final prediction (z -score) will be the average over these values (for the final feature layer). By mapping the features over the input image, we can see which features match which special region within the image (right panel). If the high values (more yellow color) match the regions with weeds within the image, it gives us more confidence in the model. If not, we can see which spatial regions "confuse" the model. It allows us to better understand feature influence, allows debugging and interpretability of the model and (if needed) adjusting the model architecture. For example, if the receptive

field is too small, the model may not capture large-scale patterns. If it is too large, details might be lost.

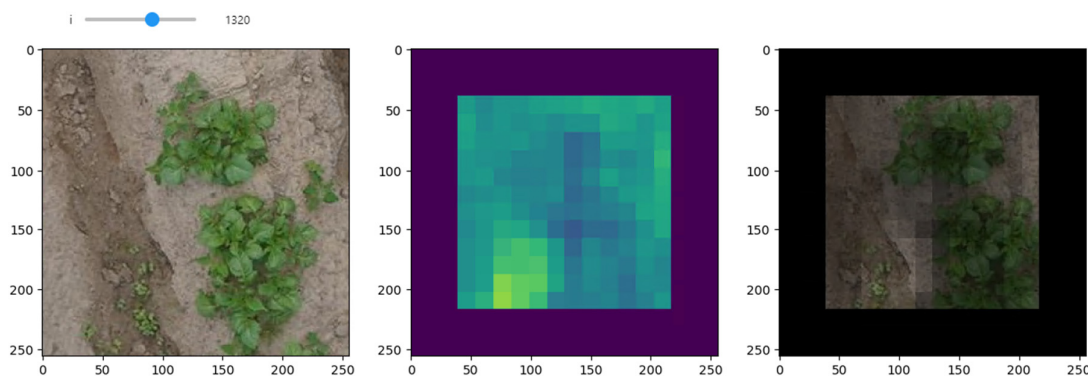


Figure 11. Comparison of the input image subsection (**left panel**) with the receptive field of the finale feature map (**middle panel**). The latter indicates regions with high and low values from yellow to dark blue. The model will calculate the weed pressure (z-score) based on the average of these values. The (**right panel**) maps the feature map over the input subsection. High values are made more transparent, while low values are darker. For this example, the regions with weeds are quite well detected by the model.

4. Discussion

4.1. Challenges of Deep Learning Algorithms

In practice, remote sensing applications have a limited amount of available (labelled) data. Data engineers and scientists are trying to come up with lightweighted models, fast labelling approaches and retraining strategies as solutions for their application. Models are fast retrained or finetuned with a small training and validation dataset of a certain dataset, knowing that the datasets are not independent and the model will likely lose performance when used on another independent dataset. If the retraining and overfitting of lightweighted models has a lower processing time than collecting very large datasets and training robust models, small companies will conduct the first strategy for their business services. Once enough data is collected, they will switch to the training of larger, more general models.

In this paper, we proposed a fast approach for consistent image labelling. The relative reference for each image pair is clear. Image regression, object detection and segmentation (semantic, instance and panoptic) require every object or pixel within the image that belongs to a weed plant to be exactly labelled. This labelling process is time consuming and challenging [2]. Labelling for image classification is easier compared to segmentation, object detection and regression, but gives no information on weed pressure rate within the image. This hinders freedom in choice of herbicide rate application by the end user.

4.2. The Dataset

Labelled datasets, be it for image classification, object detection or segmentation, can be easily converted into a much larger dataset with relative image pairs. Objects and masks can indicate how many weeds are present within the image, which makes ranking of weed pressure possible, resulting in a very large number of combinations pairs for ranking. For example, when the weed pressure (annotated as only one label) is known for n images, the total amount of possibilities (TA) in complete pairwise ranking is [36]:

$$TA = \frac{n(n-1)}{2} \quad (2)$$

This can seriously increase the total amount of data. Not all combinations must, however, be given to the model to achieve a sufficiently performant model (like in this study). For binary labelled image datasets, with x members of label 1 (weeds present) and y members of label 0 (no weeds present), the total labelled dataset (n) changes from:

$$n = x + y \quad (3)$$

to:

$$TA = xy \quad (4)$$

without additional labelling effort. However, this last example is illustrative given that building such a pairwise dataset could result in a model that does not adequately capture the full ranking curve of weed pressures.

In Section 4.1, it was already mentioned that labelling pairwise image subsections are consistent and easy to label. This is true for most cases, except for two conditions:

- rankings are complicated by poor data quality in one or both image subsections (differences in lightning, shadow effects, blurred images, not enough spatial resolution to detect very small weed plants),
- an image subsection pair has similar or very similar weed pressure rate.

Both conditions can create annotation uncertainty. The first condition is common for all computer vision models and is often overcome by removing low quality images from training and validation.

For the second condition, there is no clear approach to overcome this problem. On the one hand, these image pairs could be skipped and the labels could focus on more certain relative differences in weed pressure. This however might lead to annotation laziness. On the other hand, these hard to rank image subsection pairs could be ranked by multiple data annotators (with sufficient domain expertise). It can however also lead to vicious circles when there are triplets of image subsections ($im1, im2, im3$) that have been ranked in the label data in such a way that: $im1 > im2, im2 > im3$ and $im3 > im1$ [36]. Vicious circles will increase the labelling uncertainty and lowers the accuracy metrics (pairwise accuracy and correlation coefficients).

Xun et al. (2022) proposed to first order images in ordinal classes, and to carry out the pairwise ranking based on images from different classes [23]. However, this does not solve the ranking of images at the margins of the ordinal classes. Singh and Chakraborty (2022) worked with a combination of relative and exact labels to partially overcome this problem in facial age estimation [10]. Ferrara et al. (2024) used bias corrections on the evaluator level on the ranked annotations, which can overcome between-evaluator vicious circles, but not within-evaluator vicious circles [37].

We believe that the only “true” way to solve this problem is by segmenting every pixel of weeds present within the subsections and calculating the total number of pixels or calculate the leaf area index of the weeds (weed cover) within the field. Still, if the weed pressure is identical (e.g., 0% or 100%), ranking image pairs will always lower the predicted accuracies since the model cannot handle “equal” weed pressures. However, both solutions are not interesting for practical application due to the labor cost. One could argue that correctly predicting hard to rank pairs is not required for spot spraying, if the model can sufficiently predict the “global” ranking on image subsections from no weed, low weed, medium weed and high weed pressure.

4.3. The Model

The flexible design of our proposed model has some major advantages:

- It is a lightweight model that can be easily trained with a relatively small dataset.

- It makes it easy to shorten or lengthen both the convolutional backbone (for feature extraction) and the specific architecture (one node versus two fully connected layers). The end user can modify this design depending on the complexity of image dataset and the data availability.
- The receptive field calculation of the last convolutional block allows for better evaluation of the model.

The main disadvantage of the model is that the scores are ordinal and have no true meaning. It is not possible to interpret the distance/difference in ranking between image pairs. End users must visually check how predicted z-scores agree with in-field weed pressure. Further, while the receptive field calculation gives insight on how the model “sees” features, the evaluation potential is lower compared to true metrics (like F1-score, Sørensen–Dice score or IoU) of object detection and segmentation pipelines.

This model, like most deep learning computer vision models, has a convolutional backbone. Alternatively, recent studies have shown that Vision Transformers (ViT) that use patch embeddings and transformer encoders have competitive results to state-of-the-art CNN models while requiring less computational costs for training [2,38,39]. CNN image classification models benefit from (i) a smaller size (amount of parameters), (ii) requiring less power, (iii) faster training and inference (and thus application) and (iv) a lower annotation intensity (time needed to annotate an image) compared to object detection and segmentation models [2]. Other deep learning network architectures/backbones that have been successfully used in computer vision, but are not often used for weed detection, are Graph Neural Networks (GNNs) or Graph Convolutional Networks (GCNs) [39]. Instead of using grid or sequence structure like CNNs and ViTs, GNNs and GCNs represent images as a graph with nodes, which allows for better detection of complex and irregular objects [40]. These backbones may be able to be used as an even better feature extractor for weed detection compared to convolutional layers for image ordinal regression tasks.

4.4. Evaluation Metrics

The pairwise accuracy gives a quick measurement of the proportion of image subsection label pairs that have been correctly ranked by the model. Best accuracies were already found within 32 epochs of training (based on the early-stopping rule with patience 10 for the validation loss), resulting in a validation accuracy of 85.2%. Training with data augmentation resulted in a similar validation accuracy after 100 epochs (without an early stopping rule). This accuracy incorporates the labelling uncertainty, mentioned in Section 4.2.

Evaluation of the global ranking performance was done with 100 randomly selected image subsections that were semantically segmented for weed pixels. The model showed strong correlations (Pearson, Spearman and Kendall) with predicted weed pressure (z-score) and weed percentage, indicating it can sufficiently represent the global gradient in weed pressure within the data.

Lastly, the binary evaluation of the validation dataset shows a statistical difference in z-scores between image subsections with and without weeds present (with the latter having lower z-scores). Most optimal thresholding of the z-scores result in maximum F1-score and overall accuracy values of 0.92 and 0.88, respectively. This shows the potential of the model for application purposes. Predicted z-scores can be interpolated over the whole field (e.g., by ordinary kriging), with thresholds being set on the interpolated values based on the total amount of delineated treatment zones and area. These zones can be converted into task maps (shapefile polygons or isoxml) for spotspraying.

The training and validation datasets were, however, not independent from each other, and the validation dataset was imbalanced (427 image subsections without weed

versus 1327 with weed present). Performance of the model should be further evaluated in future research.

The proposed convolutional backbone uses valid padding, which means that the size of the output feature map is smaller than the size of the input data after every convolution operation. For an input image subsection of 256×256 pixels, the final feature map will be 12×12 pixels for valid padding, while it will be 16×16 pixels for same padding. Valid padding can result in the loss of information at the edges of the image. Same padding however results in higher computational costs and can introduce artifacts within the data (dilution of edge-related features) and distort the deep learning. The effect of padding strategy could not be compared fully objectively since it requires retraining the model, which will result in other early stopping results and final pairwise validation accuracies. The problem of padding strategy can however be overcome by keeping the valid padding structure—i.e., cutting UAV images into subsections with overlapping regions for prediction. Interpolation of the output will smoothen the results.

4.5. Further Recommendations

The current model was evaluated solely for ranking general weed pressure in potato without distinguishing between weed types. Future studies could enhance model performance by training on specific functional classes. For example, some studies make a distinction between broadleaved weeds versus grass [8,13] weeds for application purposes, or try to detect specific toxic weed species [41,42]. Additionally, species-specific ranking could be explored to facilitate more targeted weed management strategies.

When applying the proposed methodology, phenological characteristics of weeds and crops will only be accounted for if they are included in the training dataset. In case of a variation in the phenological stage of a crop or weeds, it is recommended to retrain the model before application.

The study utilized the Hinge squared loss function as an optimization approach. However, other loss functions designed for relative ranking, such as Pairwise Logistic Loss [43] and the Bradley–Terry model [37,44], were not investigated. Future research should evaluate these alternative loss functions to determine their effectiveness in improving ranking performance.

Recent studies have increasingly explored the potential of active learning for accelerating model retraining. However, there is limited literature on its application in image ordinal regression [9,10]. One possible approach could involve predicting z-scores within a Gaussian distribution framework (capturing both mean and variance). This additional uncertainty estimation could be valuable for active learning strategies, particularly for uncertainty sampling in image subsection pairs, thereby improving model efficiency and adaptability.

Further research should consider (i) whether the model can be used for other crops and weeds and look at whether it can be used for multi-class and multi-label problems, (ii) a more comprehensive comparison of this model with state-of-the-art computer vision models for weed detection, (iii) other backbone structures for feature extraction, (iv) other loss functions and (v) active learning strategies for ordinal regression for even faster retraining.

5. Conclusions

Weed pressure detection through high-resolution imagery from RTK UAVs combined with supervised CNN models provide a solution for targeted herbicide application. However, CNN models require high-quality labelled data. We proposed a new labelling method, using relative differences in weed pressure in image subsection pairs, to train a CNN ordinal regression model for weed detection in potatoes. The model achieved 85.2% pairwise

accuracy, significant correlation metrics (Pearson $r_s = 0.81$, Spearman's $r_s = 0.87$ and Kendall $\tau = 0.69$) with image subsection segmented weed percentage and a binary F1-score of up to 92%. The architecture also enables visualization of receptive fields for better interpretability. This approach shows promise for efficient and flexible weed control management.

Author Contributions: Conceptualization, R.D.; methodology, S.V. and R.D.; validation, S.V. and R.D.; formal analysis, S.V. and R.D.; investigation, S.V. and R.D.; resources, S.V., R.D. and J.V.B.; software, S.V. and R.D.; data curation, S.V., A.W. and J.V.B.; writing—original draft preparation, S.V.; writing—review and editing, R.D., A.W. and J.V.B.; visualization, S.V.; supervision, J.V.B.; project administration, A.W. and J.V.B.; funding acquisition, R.D., A.W. and J.V.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research has received funding from Flanders Innovation & Entrepreneurship (OpsDrone—VLAIO HBC.2021.0553) and the European Union (SmartAgriHubs—Horizon 2020 grant agreement N° 818182, CODECS—Horizon Europe grant agreement N° 101060179). Furthermore, it was supported by Flanders Make under the SBO project CADAIVISION and the Flemish Government under the ‘Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen’ programme.

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

Acknowledgments: The authors want to express their gratitude towards our colleagues from ILVO: (i) the certified pilot of ILVO Aaron Van Gehuchten for the data collection, (ii) Reinout Godaert for their support in building the data annotation software and the database, (iii) Ruben Van de Vijver and Jana Wieme for sharing their expertise in deep learning modelling of UAV for precision agriculture applications, and (iv) Simon Cool for coordinating the experimental setup.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A

Table A1. Hyperparameters of the best performing pairwise accuracy for the validation dataset.

| Parameter | Value |
|---|---|
| Learning rate | 0.0001 |
| Loss function | squared Hinge Loss |
| Final convolutional layer | single channel |
| Number of convolutional blocks | 4 |
| Number of filters in the 1st layer | 32 |
| Number of filters in the 2nd, 3rd and 4th layer | 64 |
| Image size | 256 × 256 pixels |
| Batch size | 100 |
| Maximum number of workers | 12 |
| Maximum number of epochs | 100 |
| Early stopping rule | When the validation loss does not further decrease after 10 epochs (patience) from the local minimum (to avoid overfitting) |

Appendix B

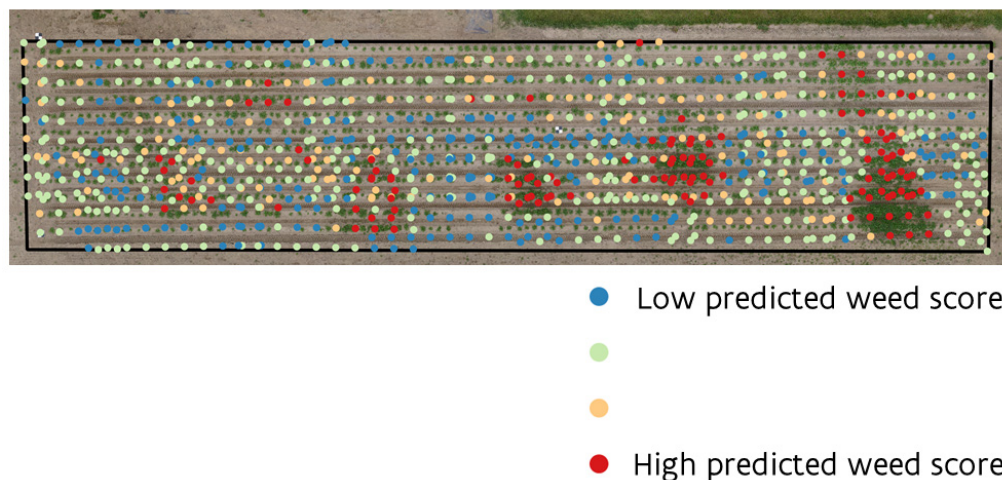


Figure A1. Predicted weed-pressure map based on the described methodology.

References

- Loddo, D.; McElroy, J.S.; Giannini, V. Problems and perspectives in weed management. *Ital. J. Agron.* **2021**, *16*, 1854. [\[CrossRef\]](#)
- Hu, K.; Wang, Z.; Coleman, G.; Bender, A.; Yao, T.; Zeng, S.; Song, D.; Schumann, A.; Walsh, M. Deep learning techniques for in-crop weed recognition in large-scale grain production systems: A review. *Precis. Agric.* **2024**, *25*, 1–29. [\[CrossRef\]](#)
- Zhang, J.; Yu, F.; Zhang, Q.; Wang, M.; Yu, J.; Tan, Y. Advancements of UAV and Deep Learning Technologies for Weed Management in Farmland. *Agronomy* **2024**, *14*, 494. [\[CrossRef\]](#)
- Mattivi, P.; Pappalardo, S.E.; Nikolić, N.; Mandolesi, L.; Persichetti, A.; De Marchi, M.; Masin, R. Can commercial low-cost drones and open-source gis technologies be suitable for semi-automatic weed mapping for smart farming? A case study in NE Italy. *Remote Sens.* **2021**, *13*, 1869. [\[CrossRef\]](#)
- Coulibaly, S.; Kamsu-Foguem, B.; Kamissoko, D.; Traore, D. Deep learning for precision agriculture: A bibliometric analysis. *Intell. Syst. Appl.* **2022**, *16*, 200102. [\[CrossRef\]](#)
- Hasan, A.S.M.M.; Sohel, F.; Diepeveen, D.; Laga, H.; Jones, M.G.K. A survey of deep learning techniques for weed detection from images. *Comput. Electron. Agric.* **2021**, *184*, 106067. [\[CrossRef\]](#)
- Su, J.; Zhu, X.; Li, S.; Chen, W.H. AI meets UAVs: A survey on AI empowered UAV perception systems for precision agriculture. *Neurocomputing* **2023**, *518*, 242–270. [\[CrossRef\]](#)
- Dos Ferreira, S.; Freitas, M.; da Silva, G.; Pistori, H.; Theophilo Folhes, M. Weed detection in soybean crops using ConvNets. *Comput. Electron. Agric.* **2017**, *143*, 314–324. [\[CrossRef\]](#)
- Steininger, D.; Trondl, A.; Croonen, G.; Simon, J.; Widhalm, V. The CropAndWeed Dataset: A Multi-Modal Learning Approach for Efficient Crop and Weed Manipulation. In Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023. [\[CrossRef\]](#)
- Shahi, T.B.; Dahal, S.; Sitaula, C.; Neupane, A.; Guo, W. Deep Learning-Based Weed Detection Using UAV Images: A Comparative Study. *Drones* **2023**, *7*, 624–642. [\[CrossRef\]](#)
- Bah, M.D.; Hafiane, A.; Canals, R. Deep Learning with Unsupervised Data Labeling for Weed Detection in Line Crops in UAV Images. *Remote Sens.* **2018**, *10*, 1690. [\[CrossRef\]](#)
- Genze, N.; Ajekwe, R.; Güreli, Z.; Haselbeck, F.; Grieb, M.; Grimm, D.G. Deep learning-based early weed segmentation using motion blurred UAV images of sorghum fields. *Comput. Electron. Agric.* **2022**, *202*, 107388. [\[CrossRef\]](#)
- Giakoumoglou, N.; Pechlivani, E.M.; Tzovaras, D. Generate-Paste-Blend-Detect: Synthetic dataset for object detection in the agriculture domain. *Smart Agric. Technol.* **2023**, *5*, 100258. [\[CrossRef\]](#)
- Seiche, A.T.; Wittstruck, L.; Jarmer, T. Weed Detection from Unmanned Aerial Vehicle Imagery Using Deep Learning—A Comparison between High-End and Low-Cost Multispectral Sensors. *Sensors* **2024**, *24*, 1544. [\[CrossRef\]](#)
- Ragu, N.; Teo, J. Object detection and classification using few-shot learning in smart agriculture: A scoping mini review. *Front. Sustain. Food Syst.* **2023**, *6*, 1039299. [\[CrossRef\]](#)
- Belissent, N.; Peña, J.M.; Mesías-Ruiz, G.A.; Shawe-Taylor, J.; Pérez-Ortiz, M. Transfer and zero-shot learning for scalable weed detection and classification in UAV images. *Knowl.-Based Systems.* **2024**, *292*, 111586. [\[CrossRef\]](#)
- Wang, S.; Han, Y.; Chen, Y.; He, X.; Zhang, Z.; Liu, X.; Zhang, K. Weed Density Extraction Based on Few-Shot Learning Through UAV Remote Sensing RGB and Multispectral Images in Ecological Irrigation Area. *Front. Plant Sci.* **2022**, *12*, 735230. [\[CrossRef\]](#)

18. Bhattacharya, A.R.; Chakraborty, S. Deep Active Learning with Range Feedback for Facial Age Estimation. In Proceedings of the International Joint Conference on Neural Networks, Padua, Italy, 18–23 July 2022; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2022; pp. 1–9. [[CrossRef](#)]
19. Singh, A.; Chakraborty, S. Deep Active Learning with Relative Label Feedback: An Application to Facial Age Estimation. In Proceedings of the International Joint Conference on Neural Networks, Online, 18–22 July 2021; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2021. [[CrossRef](#)]
20. Cao, W.; Mirjalili, V.; Raschka, S. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognit. Lett.* **2020**, *140*, 325–331. [[CrossRef](#)]
21. Parikh, D.; Grauman, K. Relative Attributes. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011. [[CrossRef](#)]
22. Shi, X.; Cao, W.; Raschka, S. Deep Neural Networks for Rank-Consistent Ordinal Regression Based on Conditional Probabilities. *Pattern Anal. Applic.* **2023**, *26*, 941–955. [[CrossRef](#)]
23. Xun, L.; Zhang, H.; Yan, Q.; Wu, Q.; Zhang, J. VISOR-NET: Visibility Estimation Based on Deep Ordinal Relative Learning under Discrete-Level Labels. *Sensors* **2022**, *22*, 6227. [[CrossRef](#)]
24. AgiSoft Metashape Professional (Software). 2016. Available online: <http://www.agisoft.com/downloads/installer/> (accessed on 7 March 2025).
25. QGIS.org. (Software). (2022). QGIS Geographic Information System. QGIS Association. Available online: <http://www.qgis.org> (accessed on 7 March 2025).
26. Van Rossum, G.; Drake, F.L. *Python 3 Reference Manual*; CreateSpace: Scotts Valley, CA, USA, 2009.
27. Wobbecke, D.M.; Meyer, G.E.; Von Bargen, K.; Mortensen, D.A. *Plant Species Identification, Size, and Enumeration Using Machine Vision Techniques on Near Binary Images*; DeShazer, J.A., Meyer, G.E., Eds.; International Society for Optics and Photonics: Bellingham, WA, USA, 1993; pp. 208–219. [[CrossRef](#)]
28. Kluyver, T.; Ragan-Kelley, B.; Pérez, F.; Granger, B.E.; Bussonnier, M.; Frederic, J.; Kelley, K.; Hamrick, J.; Grout, J.; Corlay, S.; et al. Jupyter Notebooks—A publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*; Loizides, F., Schmidt, B., Eds.; IOS Press: Amsterdam, The Netherlands, 2016; pp. 87–90. [[CrossRef](#)]
29. NumPy Developers. (Software). 2022. Available online: <https://numpy.org/doc/stable/license.html> (accessed on 7 March 2025).
30. Clark, A. *Pillow (PIL Fork) Documentation*; Read the Docs: Portland, OR, USA, 2015; Available online: <https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf> (accessed on 7 March 2025).
31. Bradski, G. The OpenCV Library. Dr. Dobb's Journal of Software Tools. 2000. Available online: <https://github.com/opencv/opencv> (accessed on 7 March 2025).
32. Pasiok, R. Serval—Raster Editing Tools. Lutra Consulting. Available online: https://github.com/lutraconsulting/serval/blob/master/Serval/docs/user_manual.md (accessed on 7 March 2025).
33. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035. [[CrossRef](#)]
34. Falcon, W.; Borovec, J.; Wälchli, A.; Eggert, N.; Schock, J.; Jordan, J.; Skafté, N.; Ir1dXD; Bereznyuk, V.; Harris, E.; et al. PyTorch Lightning. 2019. Available online: <https://github.com/Lightning-AI/lightning> (accessed on 7 March 2025).
35. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Van Mulbregt, P.; SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [[CrossRef](#)]
36. Efimov, V. Introducing to Ranking Algorithms. Towards Data Science. 2023. Available online: <https://towardsdatascience.com/introduction-to-ranking-algorithms-4e4639d65b8/> (accessed on 7 March 2025).
37. Ferrara, A.; Bonchi, F.; Fabbri, F.; Karimi, F.; Wagner, C. Bias-aware ranking from pairwise comparisons. *Data Min. Knowl. Disc.* **2024**, *38*, 2062–2086. [[CrossRef](#)]
38. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929v2. [[CrossRef](#)]
39. Lei, L.; Yang, Q.; Yang, L.; Shen, T.; Wang, R.; Fu, C. Deep learning implementation of image segmentation in agricultural applications: A comprehensive review. *Artif. Intell. Rev.* **2024**, *57*, 149. [[CrossRef](#)]
40. Han, K.; Wang, Y.; Guo, J.; Tang, Y.; Wu, E. Vision gnn: An image is worth graph of nodes. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 8291–8303. [[CrossRef](#)]
41. Zhang, J.; Maleski, J.; Jespersen, D.; Waltz, F.C., Jr.; Rains, G.; Schwartz, B. Unmanned Aerial System-Based Weed Mapping in Sod Production Using a Convolutional Neural Network. *Front. Plant Sci.* **2021**, *12*, 702626. [[CrossRef](#)]

42. Lauwers, M.; De Cauwer, B.; Nuyttens, D.; Maes, W.H.; Pieters, J.G. Multispectral UAV Image Classification of JimsonWeed (*Datura stramonium* L.) in Common Bean (*Phaseolus vulgaris* L.). *Remote Sens.* **2024**, *16*, 3538. [[CrossRef](#)]
43. Lee, J.; Vajjala, S. A Neural Pairwise Ranking Model for Readability Assessment. In *Findings of the Association for Computational Linguistics*; Association for Computational Linguistics: Dublin, Ireland, 2022; pp. 3802–3813. [[CrossRef](#)]
44. Yan, T. Ranking in the generalized Bradley-Terry models when the strong connection condition fails. *Commun. Stat. Theory Methods* **2016**, *45*, 340–353. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.