Computers and Electronics in Agriculture 219 (2024) 108792

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

# WeedVision: A single-stage deep learning architecture to perform weed detection and segmentation using drone-acquired images

Nitin Rai, Xin Sun *

*Department of Agricultural and Biosystems Engineering, North Dakota State University, Fargo, ND 58102, USA*

ARTICLE INFO

*Keywords:*
Deep learning
Edge device
Image processing
Instance segmentation
Model optimization
Weed identification

ABSTRACT

Deep learning (DL) inspired models have achieved tremendous success in locating target weed species through bounding-box approach (single-stage models) or pixel-wise semantic segmentation (two-stage models), but not both. Therefore, the goal of this research study was to develop a single-stage DL architecture that not only locate weed presence through bounding-boxes but also achieves pixel-wise instance segmentation on unmanned aerial system (UAS) acquired remote sensing images. Moreover, the developed architecture experiments on integrating a novel C3 and C3x module within its backbone for dense feature extraction, as well as ProtoNet (Prototypical network) in its head component for weed masking. Furthermore, the proposed architecture has been trained on five categories of dataset exported using multiple combinations of various dataset augmentation techniques, namely, C1, C2, C3, C4, and C5, for which multiple metrics were assessed on desktop graphical processing unit (GPU) and a palm-sized edge device (AGX Xavier). Results suggest that category C4, a combination of six data augmentation techniques, outperformed the remaining categories by achieving precision scores of 85.4 % (bounding-boxes) and 82.8 % (masking) on a GPU. Whereas, the same model converted to TorchScript was able to achieve 79.1 % and 77 % bounding-box and masking accuracy on an edge device, respectively. The model developed in this research has two potential applications when integrated with site-specific weed management technologies. First, it enables real-time weed detection, allowing for the immediate identification of weeds for spot-spraying applications. Second, it facilitates instance weed masking, aiding in the estimation of weed growth extent in actual field conditions. Moreover, the developed architecture combines both computer vision applications - detection and instance segmentation – to provide comprehensive information about weed growth, eliminating the need for multiple algorithm.

## 1. Introduction

Agricultural weed identification plays a significant role in developing adaptive precision sprayers for real-time applications. Over the past years, various ground as well as aerial-based technologies have been adopted to spot spray herbicide. Due to patch-based growth nature of weeds, spot spraying of herbicide is often preferred when compared to broadcasting application. Therefore, herbicide sprayers are equipped with a computer vision component to deliver real-time weed identification followed by instant spraying dosages. Deep learning-based (DL) object detection models have significantly contributed in detecting and localizing weed presence for precise herbicidal application in real-time scenarios. However, these object detection models pose a limitation in not being able to precisely delineate weeds by creating weed masks. Within weed detection research, achieving bounding boxes (with four coordinates) on an identified weed species is possible, thereby bypassing pixel-wise weed segmentation. In addition, pixel-wise weed segmentation delivers an extended level of information by drawing a weed mask that tends to estimate the extent of weed growth in a dynamic background.

Recently, segmentation models have been deployed to segment weeds from crop plants and soil backgrounds. These segmentation models are mostly semantic-based and do not offer detailed and distinct information about weed species either in images or in a real-time context. Moreover, semantic segmentation may appear similar to instance segmentation but fails to create individual instances of weed masks within the given context. On the other hand, instance segmentation is a novel approach within the field of computer vision that aims to provide a solution to this limitation by segmenting each instance of individual weed species in any context (Hafiz and Bhat, 2020). Past

* Corresponding author.
*E-mail address:* xin.sun@ndsu.edu (X. Sun).

https://doi.org/10.1016/j.compag.2024.108792
Received 5 October 2023; Received in revised form 24 February 2024; Accepted 26 February 2024
Available online 2 March 2024

research has relied on testing various semantic segmentation models for pixel-wise weed segmentation. For example, Mask-RCNN, a two-stage model, was deployed to segment weeds in aerial images (Gromova, 2021). A year later, an improved version of the same two-stage model was developed to segment weeds in sugar beet plantation (Jin et al., 2022). A two or multi-stage segmentation architecture comprises of hierarchical levels of tasks based on the input image fed to the network. First, the initial layers extract relevant features from the region of interest (RoI), and then further classification and localization of the RoI takes place. Generally, the performance of multi-stage models is slow when integrated with edge devices for real-time applications (Li et al., 2022).

Likewise, the application of other segmentation models such as, SegNet, U-Net and DeepLabV3+ have also been used to map weeds in aerial images (Zou et al., 2021b; Hashemi-Beni et al., 2022; Rai et al., 2023). Applicability of U-Net architecture has been also tested with ResNet-34 as a backbone for weed segmentation in drone captured imagery (Genze et al., 2022). However, most of the semantic models are two-stage architectures that are computationally expensive and demand significant graphical processing unit (GPU) memory in order to perform inference task on test set. Additionally, research studies in designing and deploying single-staged architectures to create weed masks in drone captured images and videos need tremendous efforts (Genze et al., 2022). To ease this demand, a state of-the-art single-staged YOLO architectures have been designed that provides an effective means to achieve weed identification and masking even on small edge devices. However, research studies on exploring YOLO-based architectures to identify and pixel-wise segment weeds in aerial images is scarce. Moreover, inferring the hardware feasibility test related to speed, accuracy, memory allocation, and hardware temperatures of instance segmentation models to identify weeds in aerial images and videos needs further research Sharma et al. (2022).

Leveraging the application of DL architecture has its added benefits by delivering in-depth information about multiple weed species. However, deploying such architecture comes at a cost of spending hefty hours to manually annotate aerial images for training purpose (Li et al., 2023b). Additionally, in order to train annotated aerial images, high computation with expensive hardware is required since the aerial images are of high-resolution. This increases the training and decision-making time of site-specific weed management (SSWM) technologies as well. This research study explores the possibility of using manually labeled in-field and greenhouse images in order to train a DL architecture that could detect as well as individually segment multiple classes of weed species in open-field settings. Furthermore, the developed model would be deployed on aerial images to test the generalization and precision accuracy. Therefore, the hypothesis of this research study is based on testing the developed DL architecture that would perform a combined application of detecting and segmenting weeds of intertest. By doing this, reliability of running separate algorithms for multiple tasks will be mitigated resulting in using one optimized model for two computer vision applications – real-time detection and instance segmentation. Most of the algorithms developed to be integrated with site-specific technologies should be optimized in a way that it does not use a lot of computational resources. Therefore, the developed model will simplify this computational process of reducing the hardware and software complexity during execution process. Additionally, having a unified algorithm will promote consistency and coherence in the output, ensuring better integration SSWM technologies. The specific objectives and contributions of this research study are as follows:

1. Develop a single-stage DL architecture that addresses two tasks: detection (bounding-boxes) and instance segmentation (masking) on UAS-acquired images,
2. Generate five categories of UAS-acquired dataset by using a combination of multiple data augmentation techniques,
3. Evaluate the models using multiple prediction scores and identify the best combination of data augmentation techniques, and,
4. Convert the trained model for edge deployment and identify an efficient format that can be used for real-time weed identification on SSWM technologies.

## 2. Materials and methods

This section describes the dataset acquisition and pre-processing steps that were undertaken to prepare the data format for training on the developed instance segmentation architecture. Fig. 1 displays an overview of the steps adopted for this research study. Furthermore, model conversion, metrics calculation, and model edge deployment steps are explained subsequently in each section.

### 2.1. Dataset acquisition

The dataset presented in this study was acquired in the summer of 2021 and 2022. Specific time frame chosen to acquire these images was from mid-May until late September. After sowing the weed seeds in the NDSU-GH, a two-week period was allotted for germination, followed by the data collection process within the greenhouse setting. Following this step, weeds were transplanted in multiple locations to acquire in-field images. The images in the NDSU-GH and field settings were captured using a Canon 90D and DJI Phantom 4 Pro (V2.0), respectively. To capture the in-field images, the DJI Phantom was flown at an average of 12 ft (≈3.7 m), which delivered distinct images of each weed species for manual annotation and masking. Moreover, to diversify the image dataset for training the instance segmentation model, two distributions of the dataset were merged, acquired using a handheld camera and drone technology. The handheld camera delivered images with a resolution of 6960 × 4640 based on manual settings, while the DJI Phantom drone images were captured at 5472 × 3648 and were further clipped and converted to a resolution of 640 × 640. The drone images were clipped at this resolution because it resulted in creating a larger number of weed species instances compared to using a single high-resolution image with fewer annotated examples. Two locations were chosen to acquire in-field images, Agronomy Seed Farm (ASF, Casselton) and Carrington Research Extension Center (CREC). The original number of images used for further pre-processing and augmentation were 767 in number with over 1123, 1339 and 1131 training instances for ragweed (*Ambrosia artemisiifolia*), horseweed (*Erigeron canadensis*) and kochia (*Bassia scoparia*), respectively. Fig. 2 showcases sample images that were annotated to train the instance segmentation model.

### 2.2. Manual weed masking and dataset preprocessing

After the data acquisition process, all the raw images were clubbed into one folder to perform manual weed masking annotation, followed by multiple preprocessing and augmentation steps (Table 1). The data was uploaded to Roboflow, a subscription-based online software that could be used to annotate and export images in multiple training formats. After the dataset was manually annotated using polygon tool, four pre-processing steps were applied on the images. These steps were, auto-orient, resize, auto-adjust contrast, and filter null. To auto-adjust all the image contrast stretching technique was applied with a filter null value of 100 %. This value removed all the null images that were not annotated within the dataset. The preprocessing steps were applied to all classes of weed species. After manual annotation, the total number of polygonal shapes created for kochia, common ragweed, and horseweed were 1786, 3086, and 2320, respectively. Therefore, the total number of manually annotated polygonal boxes was 7192. The dataset was further augmented into six categories with a combination of multiple augmentation techniques which has been described in the next section.

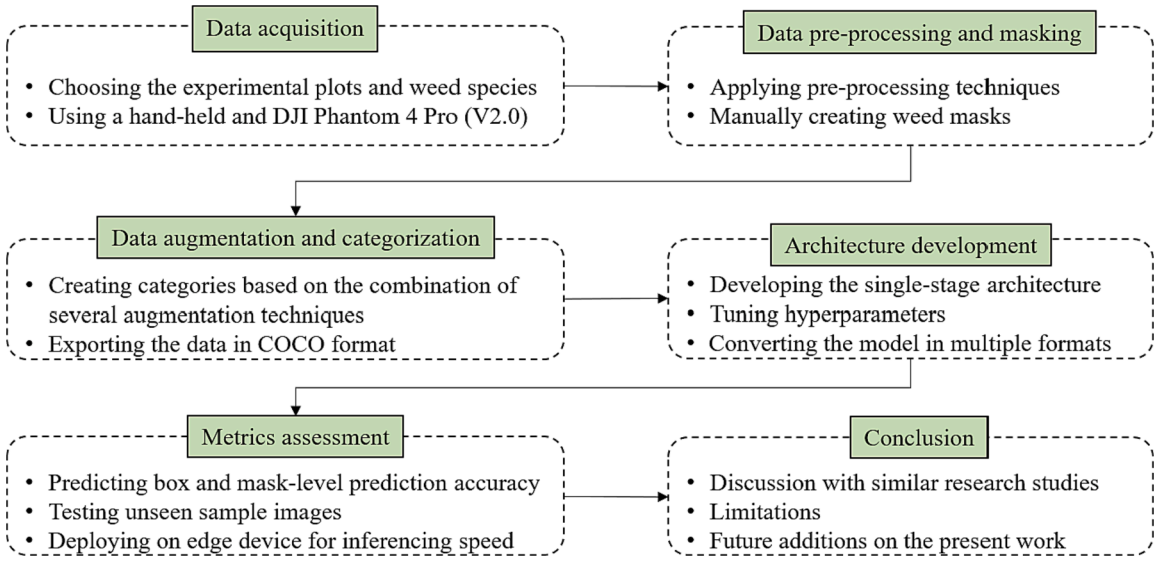


**Fig. 1.** Overview representing the steps adopted in this research study.


**Fig. 2.** Five individual categories of training dataset resulting from several combinations of various augmentation techniques. Following label numbers and color represents weed species, horseweed (0, yellow), kochia (1, green), and ragweed (2, orange).

**Table 1**
Various combinations of augmentation techniques applied on preprocessed images.

| Categories | Augmentation techniques |
|---|---|
| Category (C1) | Flip: Horizontal and vertical<br>90° rotate: Clockwise, counter-clockwise, upside down<br>Rotation: Between −45° and +45°<br>Shear: ±20° horizontal and vertical |
| Category (C2) | Flip: Horizontal and vertical<br>Rotation: Between −45° and +45°<br>Grayscale: Apply to 5 % of images<br>Hue: Between −20° and +20° |
| Category (C3) | Flip: Horizontal and vertical<br>90° rotate: Clockwise, counter-clockwise, upside down<br>Rotation: Between −45° and +45°<br>Shear: ±20° horizontal and vertical<br>Noise: Up to 1 % of pixels<br>Mosaic: Applied |
| Category (C4) | Shear: ±20° horizontal and vertical<br>Blur: Up to 1px<br>Noise: Up to 1 % of pixels<br>Cutout: 5 boxes with 3 % size each<br>Bounding box flip: Horizontal & vertical<br>Bounding box rotation: Between −5° and +5° |
| Category (C5) | Flip: Horizontal and vertical<br>Shear: ±20° horizontal and vertical<br>Hue: Between −30° and +30°<br>Noise: Up to 2 % of pixels<br>Cutout: 5 boxes with 3 % size each<br>Mosaic: Applied<br>Bounding box shear: ±5 horizontal and vertical |

### 2.3. Dataset categorization for training the architecture

To train the developed instance segmentation architecture, a weed dataset was created consisting of five categories based on multiple combinations of data augmentation techniques. As discussed in the previous section, this step was per formed to test and report the best combination of augmentation technique that could be used to achieve the maximum prediction accuracy when deploying the model on the test dataset. Table 1 displays five categories of dataset generated by applying data augmentation techniques. These categories were further used to train the instance segmentation architecture. After all these augmentation techniques were applied on the original dataset, the five categories consisted of training images in the following order; 2,685 (C1), 2,683 (C2), 2,529 (C3), 2,257 (C4), and 2,534 (C5). The percentage allocation for the validation and test dataset was based on the average of all the five categories, which turned out to be 291, with 10 % allocated to both the validation and testing sets.

### 2.4. Architecture development

#### 2.4.1. Single-stage DL architecture for detection and segmentation tasks

As an overview, the architecture developed in this study has been inspired by You Only Look Once (YOLO) models. Currently, YOLO models are state-of-the-art models which are used to detect and segment objects of interest. However, previous researches on weed identification have mostly relied on using YOLO object detection (Wang et al., 2022; Dang et al., 2023; Fan et al., 2023; Coleman et al., 2024; Rai et al., 2024). Therefore, in this study, we aim to develop and optimize a DL architecture that addresses both the tasks, detection and segmentation. YOLO models typically consist of three components, backbone, neck, and head. For object detection models, layers in the backbone component are responsible for feature extraction while the neck layer aggregates all the features and the head component is responsible for drawing bounding-boxes on the identified weed species based on anchor values. However, unlike object detection models, instance segmentation architecture uses a prototypical learning (ProtoNet) architecture (Zhao et al., 2021) that is integrated within the model architecture (head component) to create masks alongside bounding-boxes. Within the network structure, the ProtoNet consists of 2D convolutions and a sigmoid linear unit (SiLU) activation function. The whole structure comprises of three vital arguments, number of input channels, number of output channels in the first and second convolutional layer, and number of masks that needs to be generated in the final layers. For the detection models, the total number of output channel corresponding to 24 for three classes of weeds and is based on Eq. (1).

$$\mathrm{TOC} = (\mathrm{B_{box}} + \mathrm{obj_s} + \mathrm{n_c} + \mathrm{n_{mk}}) \times \mathrm{n_{ac}} \quad (1)$$

where TOC is total number of output channels, $B_{box}$ is number of bounding boxes, $obj_s$ is objectness score, $n_c$ is number of classes, $n_{mk}$ is number of masks, and $n_{ac}$ is number of anchors.

Therefore, as per the equation above, (3 classes + 4 bounding box coordinates + 1 confidence score) × 3 anchor values. But, with the addition of the prototype masks, i.e., 32 mask outputs are also added leading to, (24 outputs + 32 mask outputs) × 3 anchor values, 168 output channels for the instance segmentation architecture.

#### 2.4.2. Final architecture development and hyperparameter tuning

To develop the final architecture, multiple layers of C3 module (concentrated-comprehensive convolution) (Park et al., 2018) (Figs. 3 and 4) were integrated within the backbone component along with one layer of C3x (cross-correlational convolution) module. The C3 module consist of depth-wise and dilated convolutions in its architecture. For a depth-wise convolution, each input channel is convolved with its own individual filters thereby capturing specific spatial patterns separately. Moreover, a dilated convolutional operation is found to increase the receptive field by adding a gap in the kernel filters ensuring that the computational efficiency is not increased without the addition of extra trainable parameters. These two convolutional operations are extremely helpful to capture large spatial information from the input image that demands large contextual modeling. In general, C3 module has been found to increase model performance by 2 % when integrated with the ESPNet architecture (Mehta et al., 2018). For a C3x module, that performs cross-correlational operation, tends to extract features both horizontally and vertically from the incoming input feature maps (the f in figure represents flipped filter). The main intention is to introduce filters that are flipped so that high-level features could be extracted from the layers fed to the network. Overall, the major purpose to integrate C3x module is to impart effective feature learning capability to the model by first convolving the filters vertically and then horizontally thereby improving identification prediction for multi-scaled objects. However, in this study, only one layer of C3x module has been integrated within the last layer of the backbone component (Fig. 3) to avoid unnecessary parameter increment that may result in long hours of model training, increased floating point operations (FLOPS), and large amounts of GPU memory usage.

In Fig. 4, a total of 23 layers of architecture is used that consists of three components subdivided by multiple convolutional, C3, C3x, spatial pyramid pooling fast (SPPF), upsample, concatenation, and segmentation layers. Based on the input image, the feature maps of size 32 × 32 is outputted by the first convolutional layer that shows filters in action as they extract edge (stage 1). As the feature maps are processed in each stage, the filters convolve with each layer thereby extracting more crisp features from the input image. Finally, the extracted feature maps are aggregate to the neck layer (stage 23) and it is sent to the final segmentation layer for classification. This classification layers tend to draw bounding-boxes and mask the identified weed species based on the anchor scores provided to the architecture.

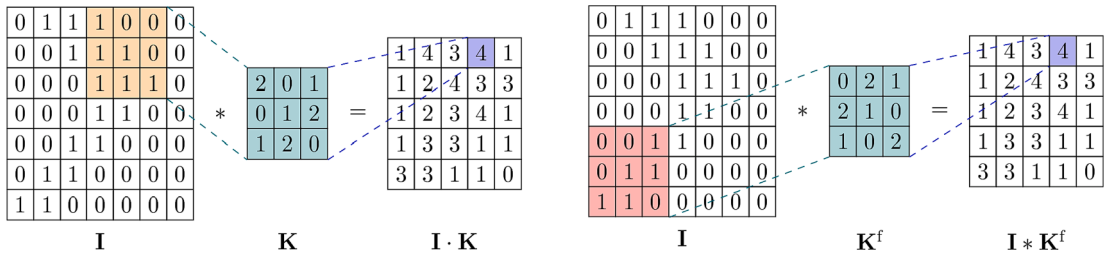


**Fig. 3.** Distinction between two convolution operations, (a) normal convolution, and (b) cross-convolution operation (C3x).

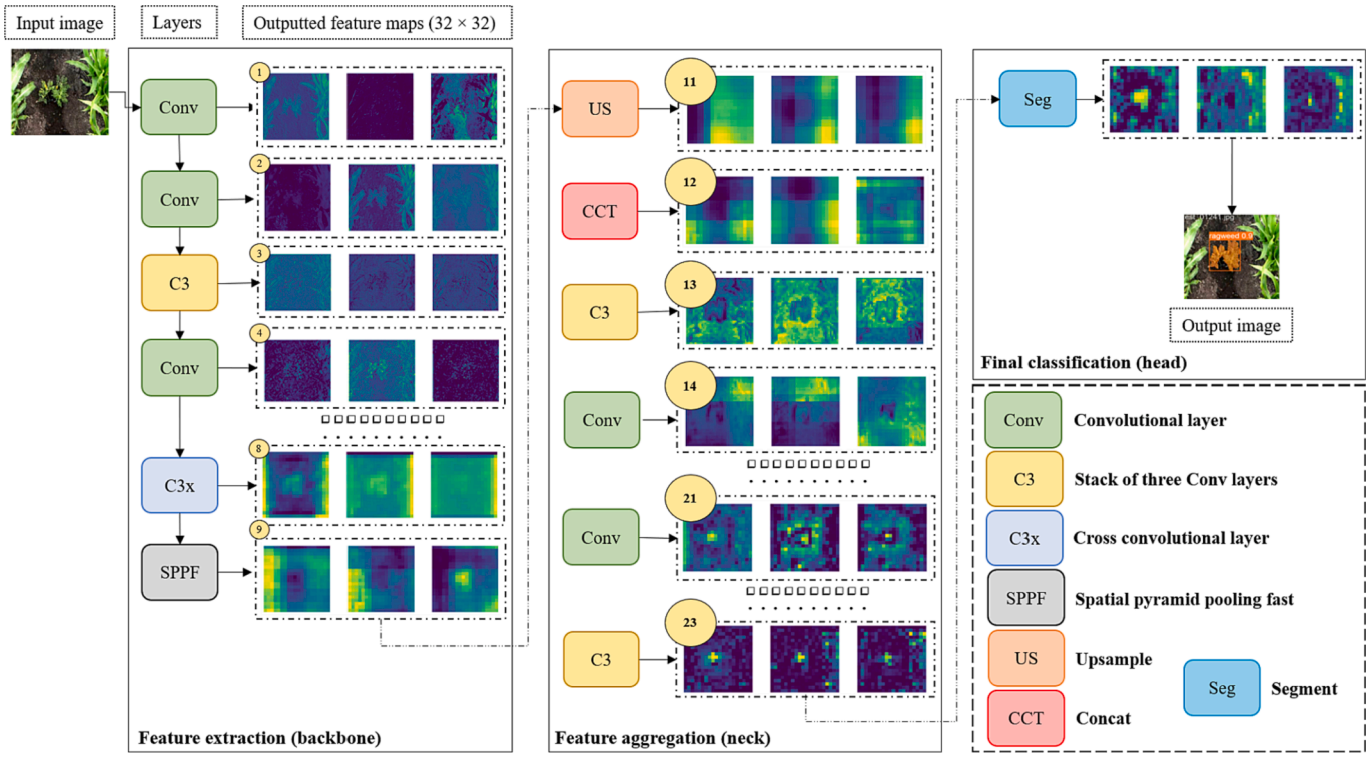


**Fig. 4.** Model architecture along with the outputted sample feature maps per layer.

Hyperparameter values for all the categories during model training process were set same to avoid unfair metrics comparison when reporting the results. While training the architecture, over 16 hyperparameter values were tuned to achieve maximum prediction accuracy (Table 2). Some of the scores for hyperparameter arguments were set based on multiple trials, such as "learning rate'' and "box loss gain,'' after observing the final prediction on the test dataset. The "batch-size'' value was set at 8 because of limited GPU memory constrained. An early stopping criterion was adopted using the "patience'' argument. This argument makes sure that overfitting is avoided by stopping the model training within 100 epochs based on the loss score value. For instance, if the model achieved its lowest loss score at 35 epochs, then this argument checks for the next 100 epochs (until 135) for the loss to decrease then the model keeps training, otherwise it stops. Optimizer "Adam'' was used because of its good performance as well as adeptness in changing learning rate scale for multiple layers in CNN (Kingma and Ba, 2014). The whole experiment (training + testing) was carried out on Intel Core-i7 (12$^{th}$ Gen.), RTX 3060 GB with 16 GB RAM size. Furthermore, the converted models were deployed on AGX Xavier to test edge device feasibility for real-time applications.

**Table 2**
Hyperparameter values chosen for training the developed DL architecture.

| S. No. | Hyperparameter arguments | Values |
|---|---|---|
| 1. | Learning rate | 0.001 |
| 2. | Momentum | 0.937 |
| 3. | Weight decay | 0.0005 |
| 4. | Warmup epoch | 3 |
| 5. | Warmup momentum | 0.8 |
| 6. | Box loss gain | 0.05 |
| 7. | Class loss gain | 0.3 |
| 8. | Object loss gain | 0.7 |
| 9. | IoU threshold | 0.2 |
| 10. | Anchor threshold | 4 |
| 11. | Number of anchors ($n_c$) | 3 |
| 12. | Optimizer | Adam |
| 13. | Cos-lr scheduler | True |
| 14. | Patience | 100 |
| 15. | Batch-size | 8 |
| 16. | Epochs | 300 |

### 2.5. Model conversion formats for real-time edge deployment

The present demand to develop edge-based solutions for agricultural applications is rising because precision weeding technologies are shifting to site-specific management. Deep learning has paved a way in assisting with developing computer vision component for real-time management. However, robust DL architectures face several

limitations regarding edge deployment integration. These limitations are often summarized as slow inferencing due to large number of parameters, high latency, high power consumption and GPU memory usage. Therefore, trained DL architectures can be converted to a format that is compatible with mobile and edge-based devices (Chen and Ran, 2019).

In this research, the developed DL architecture trained on five categories have been converted to two formats for edge deployment use case. Furthermore, these formats have been deployed on the same test images to assess prediction accuracy along with edge device performance during inferencing stage. This step was accomplished to identify a format that could be integrated with edge devices for real-time SSWM applications. These formats were, TorchScript and Open Neural Network eXchange (ONNX). TorchScript (Spisak et al., 2019) is a part of PyTorch framework that could be used to convert the trained models into a more portable, efficient, and optimized format for multiple run time environments. Whereas, ONNX format (ONNX, 2017) offers compatibility in making it easier to execute models across multiple frameworks without the need to re-train the architectures. Therefore, architectures trained on C4 and C5 were considered for metrics evaluation using these formats on an edge device. The edge device chosen to perform inference test was Jetson AGX Xavier with following computational specifications: 8-core Carmel Arm-based 64-bit CPU (2.5 GHz max. frequency), 512-core Volta architecture GPU with 64 Tensor Cores (1,377 MHz max. frequency), 32 GB RAM size, and 32 Terra Operations per Second (TOPS) for AI performance (Nvidia, 2023).

### 2.6. Metrics to assess model performance

The performance of the architectures was assessed using commonly used metrics in object detection and masking research domain (Mohamed et al., 2021). These metrics were, precision of bounding-boxes and masks ($P_b$, $P_m$), recall ($R_b$, $R_m$), mAP ($mAP_b$, $mAP_m$) at 50 % threshold scores, and losses (Eqs. (2)–(6)) where,

$$P_{(b,\ m)} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \tag{2}$$

$$R_{(b,\ m)} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \tag{3}$$

$$mAP_{(b,m)}(@.5) = \frac{1}{w}\sum_{k=1}^{k=w} AP_W \tag{4}$$

$$L_{obj} = 4{\cdot}L_{obj}^{small} + 1{\cdot}L_{obj}^{medium} + 0.4{\cdot}L_{obj}^{large} \tag{5}$$

$$\text{Total losses} = \lambda_1{\cdot}L_{cls} + \lambda_2{\cdot}L_{obj} + \lambda_3{\cdot}L_{loc} \tag{6}$$

$L_{cls}$ is the binary cross-entropy loss that measures the error during classification, $L_{obj}$ is the error measured during objectness task, and $L_{loc}$ is the error measured during localization. $\lambda_1$, $\lambda_2$, and $\lambda_3$ are balance weights, respectively, and are assigned to ensure that the predictions achieved at each scale (small, medium, and large) contribute equally to the total loss.

## 3. Experiment results

### 3.1. Effect of multiple augmentation techniques on metrics evaluation and training losses

Table 3 displays a comprehensive metric evaluation carried out to test the trained models. All the models were named as per the categories described in Table 1 and were further deployed on test images to perform metric evaluation. The metric evaluation was further subdivided into two identification types: bounding box and masking. Precision, recall, and mAP(@0.5) scores for both identification types were assessed. Furthermore, inference timing for all the categories have been included to choose trained architectures that utilize less inference time on the test dataset.

Category C4, which uses a combination of six augmentation techniques, out performs all other categories based on precision, recall, and mAP scores. The precision for both bounding-box and masking in the test set, mainly comprising in-field images, was 85.4 % and 82.8 %, respectively. Another noteworthy observation is the recall metric score of C4 category, which was the lowest (−2.5 % for bounding box and −1.4 % for masking) when compared to all other categories, despite achieving high-precision-recall scores. This indicates that the model trained on C4 category resulted in a decreased number of false positives when deployed to detect and mask target weeds.

The second-best performing category was C5, utilizing a combination of seven augmentation techniques. The precision achieved by the trained architecture on the C5 dataset was 73.4 %, which was 12 % lower than the C4 category for bounding-boxes and 12.4 % lower for masking. However, the recall score, when compared to the precision score was greater than the C4 category. This could result in a greater number of false positives when deployed on test images. The mAP score of the C5 category in bounding-box and masking was the highest, at 63.3 % and 60.5 %, respectively. These scores were +1.4 % and +0.6 % greater than those of C4 category.

Additionally, the inference timing of all the categories was evaluated based on the time required by the model to identify and mask weeds in test images. The C3 category achieved the best inference timing, but it was rejected due to poor model performance on the test dataset. The second-best model, with the best inference timing, was the C5 category. Therefore, based on the results, two categories of the trained architectures are recommended: (1) users seeking accuracy over speed should select the C4 category, and (2) users prioritizing speed over accuracy could opt for the C5 category. They can use this dataset and the trained weights on their custom dataset. Individual learning curves based on training losses were plotted to assess model's performance during the training phase. Training loss graphs were generated in real-time for bounding-boxes and masking categories (Fig. 5). These training loss graphs demonstrate the model's learning or memorizing capabilities during training. As observed, C4 category exhibits a significant drop in the training loss during training phase as compared to other categories. Additionally, the initial loss value of the C4 category was lower than that of other categories, especially C2 and C3, indicating compatible learning ability. As part of the assessment, all the models were trained for a total of 300 epochs, but the graph displays the best weights extracted during a

**Table 3**
Metrics evaluation obtained after deploying the developed architecture on the test dataset.

| Categories | $P_b$, $R_b$ (%) | $P_m$, $R_m$ (%) | $mAP_b$(@0.5) (%) | $mAP_m$(@0.5) (%) | Inference timing (ms) |
|---|---|---|---|---|---|
| C1 | 65.8, 56.6 | 65.5, 56 | 60.7 | 58 | 91.9 |
| C2 | 55.2, 46.5 | 52.5, 45.1 | 50.7 | 49.3 | 86.7 |
| C3 | 61.4, 48.5 | 59.6, 47.6 | 55.2 | 53.3 | 80.8 |
| C4¥ | 85.4, 44 | 82.8, 43.7 | 61.9 | 60.5 | 87.2 |
| C5§ | 73.4, 61.5 | 70.4, 59.6 | 63.3 | 61.1 | 82.1 |

¥ Best results obtained on C4 category.
§ Second-best result obtained on C5 category.

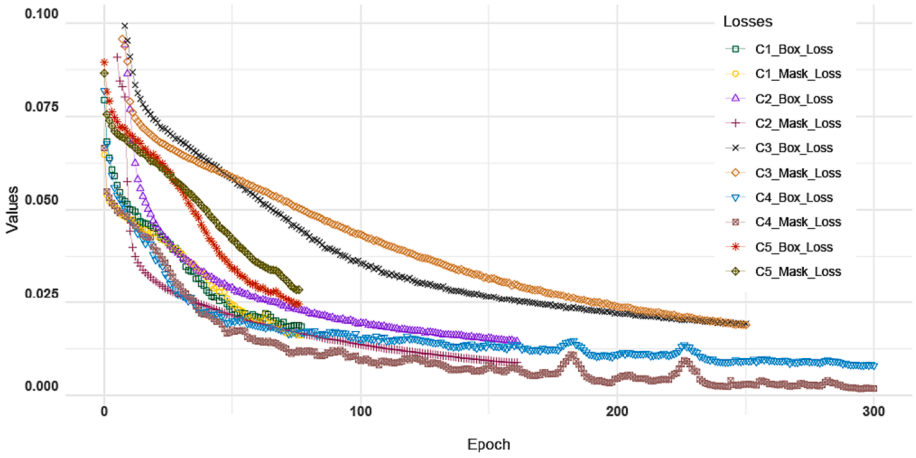


**Fig. 5.** Training epochs vs. bounding-box and mask loss scores signifying real-time learning ability of the model on the training dataset.

specific epoch. For instance, for models C1, C2, C3, C4, and C5, the best weights were extracted at 77, 161, 250, 300, and 78 epochs, respectively. Thereby, the number of hours spent during training the dataset was varied based on the number of epochs.

### 3.2. *Testing the trained architecture on unseen sample images*

Figs. 6 and 7 depicts sample images that were tested using the trained architectures. It consists of labels and predicted weeds with their

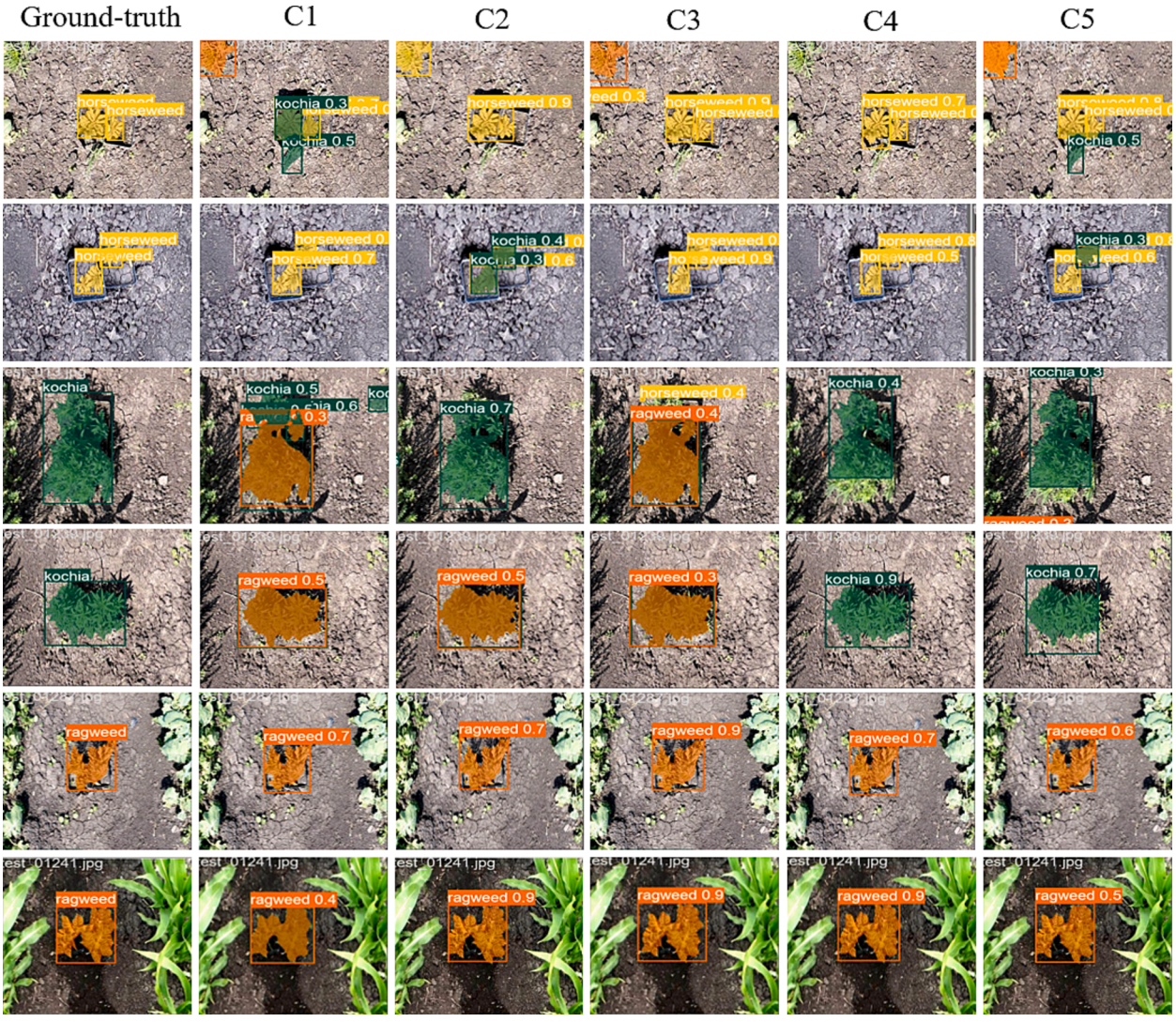


**Fig. 6.** Bounding-boxes and masked weeds as seen on test images represented by yellow color (horseweed), dark green (kochia), and orange (ragweed). Image consists of ground-truth as well as predicted masks with accuracy.

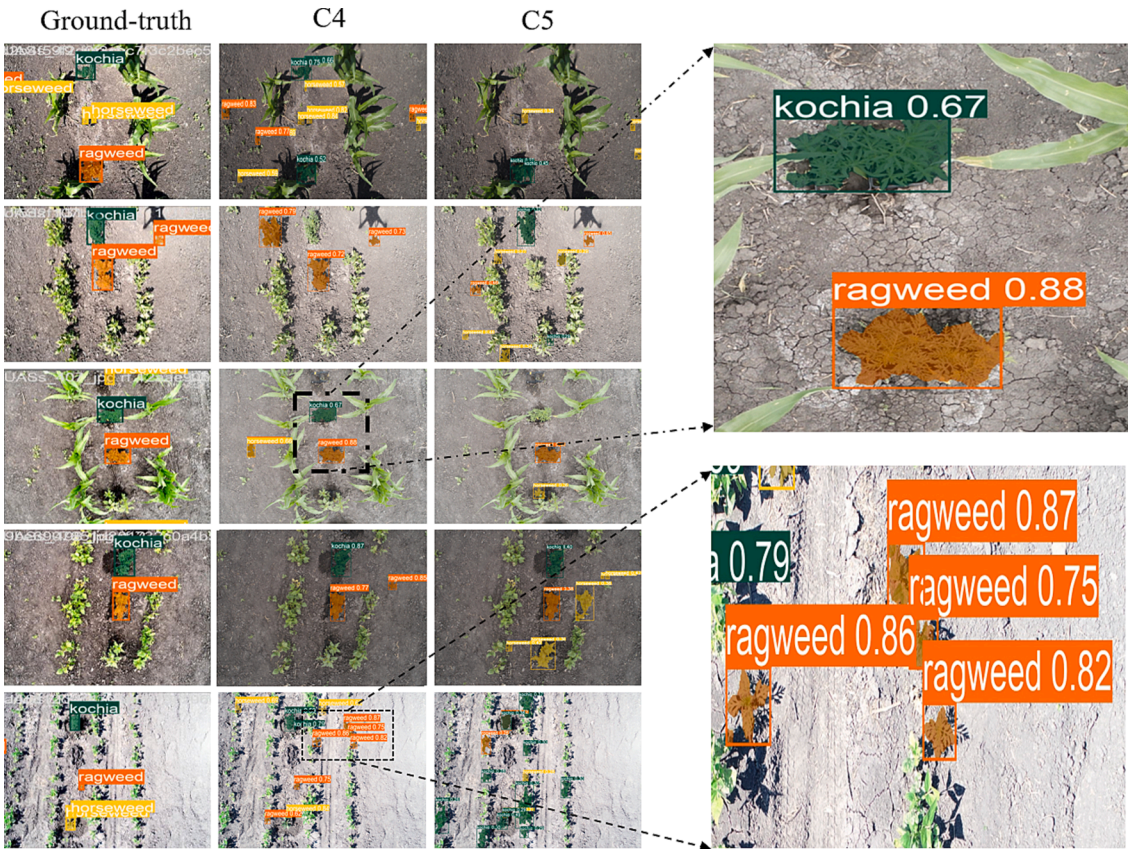


**Fig. 7.** UAS-acquired high-resolution images tested to create individual instances of segmentation masks on three weed species. Following colors represents: yellow (horseweed), dark green (kochia), and orange (ragweed).

corresponding accuracy scores. As evident based on the precision and recall scores from Table 3, models that were trained on the C4 and C5 category yielded better results. However, the C5 category resulted in a greater number of false positives along with correct predictions. For instance, in the first row, the C4 category correctly identified ragweed with over 90 %accuracy. In contrast, the C5 category identified ragweed with 60 % accuracy, along with a class confusion between kochia and ragweed. Similar observations can be made in other categories as well.

Another important aspect of all the models is their performance in terms of class confusion. Models identifying kochia (as seen in the sixth row), in heavily occluded and darker settings resulted in significant class confusion. For instance, the C4 model created two masks, one for kochia and the other for ragweed, with an accuracy of 40 %. Similar observation can be noted in C1 as well as C5 categories, where the model incorrectly predicted kochia as horseweed and ragweed with accuracies of 30 %, respectively. Among all the categories, models trained on C1 category performed the worst. This strongly suggests that data augmentation techniques are beneficial to train a model for robustness; however, careful selection of such techniques are crucial when augmenting image data. This is evident through the contrasting results obtained by the C4 and C5 categories. Model trained on C4 category comprised only six augmentation types with two bounding box-level augmentations, whereas the C5 category was augmented using seven categories with just one type of bounding-box level augmentation.

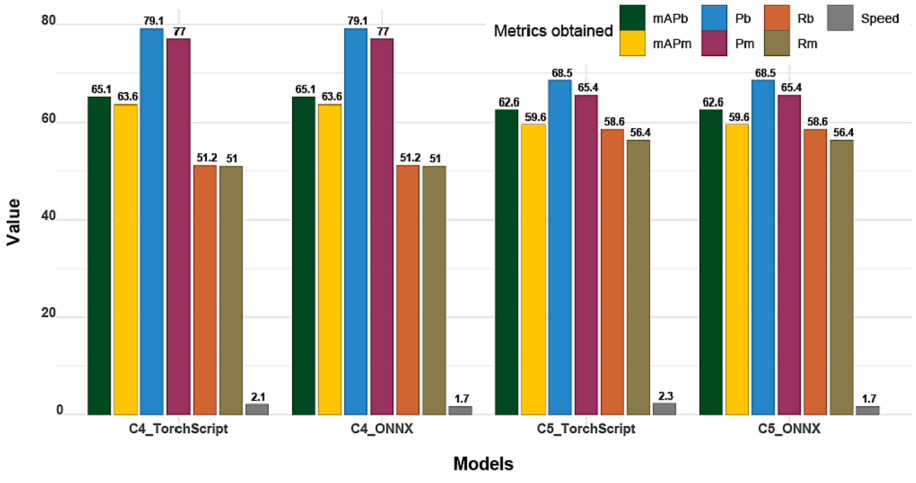


**Fig. 8.** Metrics obtained after deploying the converted formats on Jetson AGX Xavier.

Therefore, based on the categories (Fig. 1), it is suggested that incorporating multiple bounding-box-level augmentation types could improve prediction accuracy in identifying weeds captured using UAS.

### 3.3. Assessing the best format for edge deployment use

Based on Section 2.5, models trained on categories C4 and C5 were converted to TorchScript and ONNX for further processing and metric evaluation. Fig. 8 displays seven metrics that were evaluated to test models' accuracy when inferred on Jetson AGX Xavier. Additionally, for a fair comparison, similar metrics were selected to assess the effect of changing hardware on the prediction accuracy. Only speed (in ms) was added to test if the models were suitable for deployment on small microprocessor for real-time applications. All the models yielded nearly identical results on AGX Xavier.

Considering TorchScript format, among the models that were trained on C4 and C5 categories, C4 achieved an overall precision accuracy of 79.1 % (bounding-boxes) and 77 % (masking), which was 6.3 % and 5.8 % lower than the metrics obtained on another hardware (Table 3). However, C4 recall values for the same were 51.2 %and 51 %, which were 7.2 %and 7.3 % higher than the metrics obtained in Table 3. This signifies that the model detected slightly more false positives on AGX Xavier. For the C5 category, precision for bounding-boxes and masking was 68.5 % and 65.4 %, respectively, which was comparatively lower by 10.6 % and 11.6 % compared to C4 category and 4.9 % and 5 %, respectively. High mAP(@0.5) values were achieved for C4 category in TorchScript format, with values of 65.1 % and 63.6 %, which were higher by 3.2 % and 3.1 % for bounding-boxes and masking, respectively. A decrease in mAP scores was reported for C5 category, which was 0.7 % and 1.5 % less than the mAPs displayed in Table 3 and 2.5 % and 4 % compared to C4 category. The inference speed of the architecture was 2.1 ms and 2.3 ms per image for C4 and C5 category, respectively.

For ONNX models, the precision, recall, and mAP(@0.5) metrics achieved on C4 and C5 category was equal compared to the TorchScript models. However, the inference speed was $\approx$ 1.25X greater than the TorchScript models for both the categories. This signifies that the users can use both the models based on their preferences. ONNX models are recommended if the architecture is to be integrated with ground or aerial robots for site-specific management.

## 4. Discussion

A discussion on the comparative performance of this research with other related works has been presented in Table 4. Ten recently published research articles (2020–2022) were gathered specifically targeting weed segmentation mostly on drone-acquired images. While there are a large number of research articles that focus on weed detection, a selection of segmentation-based algorithms was selected since this study focuses on leveraging single-stage DL architecture for weed segmentation. Table 4, presents these ten papers along with the dataset used, accuracy achieved, and other aspects of the study.

From the table, it can be observed that U-Net has been the most preferred architecture to accomplish weed segmentation tasks, mostly on drone-acquired images. Research studies by Genze et al. (2022) and Zou et al. (2022) have demonstrated that U-Net architectures could be used to segment weeds with above 90 % accuracy, respectively. Despite U-Net's commendable success in achieving such accuracy, the

**Table 4**
Comparative performance of the developed architecture with the recent similar studies on weed segmentation.

| Models | Reference | Dataset | Segmentation type | Architecture type | Accuracy achieved |
|---|---|---|---|---|---|
| | | ***This research*** | | | |
| Our model | – | Self-built | Detection + Instance | Single-stage | Bounding-box and masking precision accuracy of 85.4 % and 82.3 %, respectively |
| | | ***Recent studies*** | | | |
| DNN | You et al. (2020) | Two public dataset, Bonn and Stuttgart | Semantic | – | Mean intersection over union (mIOU) was 89 % on Bonn dataset |
| K-means ML classifier | Gašparović et al. (2020) | UAV acquired orthomosaic images | – | – | Overall pixel-based accuracy of75.5 % and 74.2 % on Subset A and Subset B dataset, respectively |
| DeepSolanum-Net | Wang et al. (2021) | Raw images were acquired using a UAV consisting of 94,436,000 pixels in 33 images. Images were labeled and split into 640 × 640 sub-images with 2,600 images for training purpose | Semantic | – | Accuracy of 90.3 % and with IoU score of 82.7 % was achieved using the proposed architecture |
| Bonnet based on ERFNet | Su et al. (2021) | Combination of ground as well as public dataset comprising of 283 and 150 images as Bonn and Narrabri dataset, respectively | Semantic | – | Mean precision of 78.6 % and 58 % was achieved on Narrabri and Bonn dataset, respectively |
| Modified U-Net | Zou et al. (2021a) | Field images were captured using vehicle-based imaging system with over 600 images split into three subparts, training (300), validation (150), and testing (150) | Semantic | Two-stage | Modified U-Net achieved over 92.9 % accuracy and 92.8 % precision in segmenting weeds |
| U-Net | Zou et al. (2022) | The dataset was acquired using ground vehicle and a tractor-based imaging system with over 10,000 training and 2,500 test sets | Semantic | Single-stage | Precision accuracy of 95.7 % |
| Conventional ML classifiers | Zhang et al. (2022) | Images were collected using a handheld Canon 200D at 30 – 40 cm altitude. Image consisted of seedling lettuce (90 images), lettuce and weeds (30 images) | – | – | Average accuracy obtained was 86.1 % using a combination of genetic algorithm SVM (GA-SVM) classifier |
| U-Net and ResNet-34 | Genze et al. (2022) | UAV images of weeds were collected in Sorghum field comprising of five weed species | Semantic | Single-stage | Macro-averaged results with a precision accuracy of 93 % |
| U-net, FCN-8 s/16 s/32 s, SegNet, DeepLabV3+ | Hashemi-Beni et al. (2022) | Public datasets namely, Crop/Weed Field Image Dataset (CWFID) and Sugarcane orthomosaic dataset | Semantic | Single and two-stage models | Two-stage model, DeepLabV3+ achieved highest accuracy compared to other architectures |
| Mask R-CNN | Sapkota et al. (2022) | UAV acquired images of multiple weed species consisting of 460 images for training and validation while 100 for testing the model | Sematic | Two-stage | Masking accuracy was 80 % on real-field dataset |

architecture was primarily used for semantic segmentation of weeds and was not deployed on edge device. In contrast, the proposed architecture in this paper is capable of achieving instance segmentation and has been trained and tested on multiple hardware including the embedded platform, Jetson AGX Xavier. Moreover, some limitations of the U-Net architecture still persist as a part of model configuration and edge deployment. These are, (a) limited receptive field due to multiple layers of upsampling and downsampling, (b) a fixed size for the input image due to the presence of fully convolutional layers (Ronneberger et al., 2015), and (c) difficulties in deploying U-Net based architectures on edge devices for real-time applications (Safavi et al., 2022). The proposed architecture in this study addresses these limitations by reporting results and metrics based on the test dataset. For instance, the architecture used in this study does not require a single-fixed resolution of input image for training. Users can input any image resolutions based on their preference, although the computational demands for training will be directly proportional to the chosen resolution. Moreover, multi-scaling option could be used to feed random resolution to the network for training (Wang et al., 2022). Out of all the research studies, only Zou et al. (2022) deployed the developed architecture on an embedded device, achieving over 52 FPS for real-time weed segmentation. However, the model was deployed as-is without suitable format conversions for edge device applications. Therefore, the proposed model in this study has been converted into multiple formats to ensure that the architecture and trained weights are compatible with hardware of various sizes and power consumption.

## 5. Limitations and future research directions

Some of the limitations of this study can be summarized in the following points: (a) limited training dataset was used to train the developed DL architecture, (b) a very limited combinations of augmentation techniques were adopted for model training, (c) constant hyperparameters were maintained throughout the training process, (d) the model was not trained using a multi-scale training technique, and (e) evaluating the trained models during real-time performance, and (f) comprehensive comparative test of the trained model with other base models on the dataset used in this study.

In this study, the average number of datasets used to train the architectures was approximately 2,300 with over 7,700 instances of masks for individual weed species. Since instance segmentation architectures demand data intensive training due to the creation of weed masks, future work could involve training with a larger dataset acquired from multiple locations. Additionally, the application of novel modules for mask creation, such as Mask DINO (Li et al., 2023a), which utilizes a transformer framework, should be explored. Its efficiency, especially in terms of edge device deployment, should also be assessed. Furthermore, the combination of various augmentation techniques was quite limited. For instance, category C4 comprised of six augmentation techniques. However, it remains uncertain if adding two more bounding-box transformations, such as noise or cutout, might have resulted in better prediction accuracy. Similar observations could be made with the addition of various other augmentation techniques like solarize, exposure, and contrast adjustments. Currently, Albumentations package consists of various augmentation techniques both ranging from pixel-wise to spatial manipulations. A comprehensive study could be conducted regarding these augmentation techniques, allowing researchers in the future to use specific augmentation types to enhance the model's prediction accuracy for weed detection and segmentation, both on the test dataset and for real-time applications.

Another limitation could be observed in hyperparameter tuning. As mentioned earlier, all the hyperparameters were kept the same during the training phase. This was done to perform a fair comparative test with respect to metrics obtained while testing the model. However, future work could explore the use of other optimizers and label smoothing technique (Zhang et al., 2023) to assess their impact on the model's prediction accuracy. Moreover, the application of genetic evolution algorithms could fine-tune these hyperparameter based on the training dataset. Therefore, future work could involve such algorithms to fine tune these hyperparameter values, allowing for dynamic hyperparameter adjustments (Zeng et al., 2023). Multi-scale training can be leveraged to train the models for better weed prediction on the input images with multiple scales or resolutions (Diwan et al., 2023). Although this study did not include an evaluation of trained models for real-time weed detection and segmentation, as part of assessing their real-time performance, the converted formats were deployed on high-resolution videos of experimental plots recorded during n-field data collection. The model demonstrated promising results (Fig. 8); however, generating masks for individual weed species remained a challenging. While the trained model could produce masks for specific instances of target weed species, these masks were inconsistent or inaccurate in some areas. Considering the speed at which the drone flew and the pace of recorded videos, the task of generating weed masks will require additional training data to accurately depict the target weed species. The efficiency of such autonomous technologies in mapping weed in large fields will be directly influenced by this speed. If the developed computer vision models are not optimized to match this speed, it could lead to two issues: inaccurate estimation of weed growth and increased costs of weed management. Finally, this study did not include a comprehensive comparative test of base models with the developed architecture (Dang et al., 2023; Rai et al., 2024). This omission limits the findings regarding whether the trained models generalize better than the base models when deployed on test datasets. However, it's worth noting that most base models in pre-trained networks are not optimized for identifying or masking weed species in complex agricultural fields. In contrast, the architecture developed in this study was trained solely on weed datasets from scratch. This aspect makes it potentially valuable for other users and researchers who aim to train their datasets for further adaptation and fine-tuning.

## 6. Conclusions

In this paper, a single-stage DL architecture was developed to perform weed detection (bounding-boxes) and instances segmentation (masking). The dataset used in this study has been categorized into five types. These five types were exported using several combinations of data augmentation techniques (Table 1). The developed architecture has been trained on these datasets with a hypothesis to test which combination of dataset category yields better results on bounding-boxes and weed masking predictions. Furthermore, the best models were converted to formats applicable to be deployed on edge devices for SSWM tasks.

As per the results reported, architecture trained on C4 dataset exported with a combination of six augmentation techniques (Table 1), yielded higher accuracy compared to rest of the categories. The precision and recall for bounding-box were 85.4 % and 44 %, respectively. Similarly, precision and recall for masking was 82.8 % and 43.7 %, respectively. The second-high performing architecture was trained on C5 dataset comprising of seven augmentation techniques (Table 1). The precision of bounding-box and masking was 73.4 % and 70.4 %, respectively. Moreover, these categories were further converted to TorchScript and ONNX formats to be deployed on edge device for site-specific applications. Thus, the trained models on C4 and C5 categories could be integrated with SSWM technologies to detect and create individual instances of weed species in UAS-acquired remote sensing images. These decisions made by the model in real-time could then be leveraged to spot-spray herbicide on the weeds of interest. Furthermore, models converted for edge deployment, such as TorchScript and ONNX, are lightweight models that are suitable to be integrated with small microcomputers responsible for carrying out real-time processing in outdoor environments. However, ONNX models were found to be 1.25X faster compared to TorchScript models. Therefore, for edge device use

case, TorchScript models are recommended if accuracy is a preferred, while ONNX models offer a balance between speed and accuracy both.

## CRediT authorship contribution statement

**Nitin Rai:** Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft. **Xin Sun:** Data curation, Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgement

The authors would like to thank NDSU Plant Science Department, NDSU Agronomy Seed Farm, NDSU Carrington Research Extension Center for providing this research greenhouse sample preparation, space, and field experiment assistant. The authors would also like to thank Jenna Kull (Department of Agricultural and Biosystems Engineering) and Dongyu Jin (Department of Electrical and Computer Engineering), for helping with manual data labeling procedure. This material is based upon work partially supported by the U.S. Department of Agriculture, agreement number 58-6064-8-023. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the view of the U.S. Department of Agriculture. This work is/was supported by the USDA National Institute of Food and Agriculture, Hatch project number ND01487.

## References

Chen, J., Ran, X., 2019. Deep learning with edge computing: A review. Proc. IEEE. 107, 1655–1674.
Coleman, G.R., Kutugata, M., Walsh, M.J., Bagavathiannan, M.V., 2024. Multi-growth stage plant recognition: A case study of Palmer amaranth *(Amaranthus palmeri)* in cotton *(Gossypium hirsutum)*. Comput. Electron. Agric. 217, 108622.
Dang, F., Chen, D., Lu, Y., Li, Z., 2023. YOLOWeeds: A novel benchmark of YOLO object detectors for multi-class weed detection in cotton production systems. Comput. Electron. Agric. 205, 107655.
Diwan, T., Anirudh, G., Tembhurne, J.V., 2023. Object detection using yolo: Challenges, architectural successors, datasets and applications. Multimedia Tools Appl. 82, 9243–9275.
Fan, X., Chai, X., Zhou, J., Sun, T., 2023. Deep learning based weed detection and target spraying robot system at seedling stage of cotton field. Comput. Electron. Agric. 214, 108317.
Gašparović, M., Zrinjski, M., Barković, D., Radočaj, D., 2020. An automatic method for weed mapping in oat fields based on UAV imagery. Comput. Electron. Agric. 173, 105385.
Genze, N., Ajekwe, R., Güreli, Z., Haselbeck, F., Grieb, M., Grimm, D.G., 2022. Deep learning-based early weed segmentation using motion blurred UAV images of sorghum fields. Comput. Electron. Agric. 202, 107388.
Gromova, A., 2021. Weed detection in UAV images of cereal crops with instance segmentation.
Hafiz, A.M., Bhat, G.M., 2020. A survey on instance segmentation: State of the art. Int. J. Multimedia Inf. Retrieval 9, 171–189.
Hashemi-Beni, L., Gebrehiwot, A., Karimoddini, A., Shahbazi, A., Dorbu, F., 2022. Deep convolutional neural networks for weeds and crops discrimination from UAS imagery. Front. Remote Sens. 3, 1.
Jin, S., Dai, H., Peng, J., He, Y., Zhu, M., Yu, W., Li, Q., 2022. An improved mask R-CNN method for weed segmentation. In: Proceedings of the IEEE 17th Conference on Industrial Electronics and Applications (ICIEA), pp. 1430–1435.
Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
Li, J., Chen, D., Qi, X., Li, Z., Huang, Y., Morris, D., Tan, X., 2023b. Label efficient learning in agriculture: A comprehensive review. arXiv preprint arXiv:2305.14691.
Li, P., Wang, X., Huang, K., Huang, Y., Li, S., Iqbal, M., 2022. Multi-model running latency optimization in an edge computing paradigm. Sensors 22, 6097.
Li, F., Zhang, H., Xu, H., Liu, S., Zhang, L., Ni, L.M., Shum, H.Y., 2023a. MaskDino: Towards a unified transformer-based framework for object detection and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3041–3050.
Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., Hajishirzi, H., 2018. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 552–568.
Mohamed, E., Shaker, A., El-Sallab, A., Hadhoud, M., 2021. INSTA-YOLO: Real-time instance segmentation. arXiv preprint arXiv:2102.06777.
Nvidia, 2023. NVIDIA Jetson Xavier – A breakthrough in embedded applications. Accessed on: 21st Feb., 2024. URL: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-series/.
ONNX, 2017. Open neural network exchange: The open standard for machine learning interoperability. Accessed on: 21st Feb., 2024. URL: https://onnx.ai/.
Park, H., Yoo, Y., Seo, G., Han, D., Yun, S., Kwak, N., 2018. C3: Concentrated-comprehensive convolution and its application to semantic segmentation. arXiv preprint arXiv:1812.04920.
Rai, N., Zhang, Y., Ram, B.G., Schumacher, L., Yellavajjala, R.K., Bajwa, S., Sun, X., 2023. Applications of deep learning in precision weed management: A review. Comput. Electron. Agric. 206, 107698.
Rai, N., Zhang, Y., Villamil, M., Howatt, K., Ostlie, M., Sun, X., 2024. Agricultural weed identification in images and videos by integrating optimized deep learning architecture on an edge computing technology. Comput. Electron. Agric. 216, 108442.
Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer, pp. 234–241.
Safavi, F., Ali, I., Dasari, V., Song, G., Zhu, T., 2022. Efficient semantic segmentation on edge devices. arXiv preprint arXiv:2212.13691.
Sapkota, B.B., Popescu, S., Rajan, N., Leon, R.G., Reberg-Horton, C., Mirsky, S., Bagavathiannan, M.V., 2022. Use of synthetic images for training a deep learning model for weed detection and biomass estimation in cotton. Sci. Rep. 12, 19580.
Sharma, R., Saqib, M., Lin, C., Blumenstein, M., 2022. A survey on object instance segmentation. SN Comput. Sci. 3, 499.
Spisak, J., Smith, J., Dzhulgakov, D., Qiao, L., Chanan, G., 2019. Introduction to Torchscript. Accessed on: 21st Feb., 2024. URL: https://ai.meta.com/blog/pytorch-adds-new-dev-tools-as-it-hits-production-scale/.
Su, D., Kong, H., Qiao, Y., Sukkarieh, S., 2021. Data augmentation for deep learning based semantic segmentation and crop-weed classification in agricultural robotics. Comput. Electron. Agric. 190, 106418.
Wang, Q., Cheng, M., Xiao, X., Yuan, H., Zhu, J., Fan, C., Zhang, J., 2021. An image segmentation method based on deep learning for damage assessment of the invasive weed Solanum rostratum Dunal. Comput. Electron. Agric. 188, 106320.
Wang, Q., Cheng, M., Huang, S., Cai, Z., Zhang, J., Yuan, H., 2022. A deep learning approach incorporating YOLO v5 and attention mechanisms for field real-time detection of the invasive weed *Solanum rostratum* Dunal seedlings. Comput. Electron. Agric. 199, 107194.
You, J., Liu, W., Lee, J., 2020. A DNN-based semantic segmentation for detecting weed and crop. Comput. Electron. Agric. 178, 105750.
Zeng, T., Li, S., Song, Q., Zhong, F., Wei, X., 2023. Lightweight tomato real-time detection method based on improved yolo and mobile deployment. Comput. Electron. Agric. 205, 107625.
Zhang, Y., Wang, X., Liang, J., Zhang, Z., Wang, L., Jin, R., Tan, T., 2023. Free lunch for domain adversarial training: Environment label smoothing. arXiv preprint arXiv:2302.00194.
Zhang, L., Zhang, Z., Wu, C., Sun, L., 2022. Segmentation algorithm for overlap recognition of seedling lettuce and weeds based on SVM and image blocking. Comput. Electron. Agric. 201, 107284.
Zhao, N., Chua, T.S., Lee, G.H., 2021. Few-shot 3d point cloud semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8873–8882.
Zou, K., Chen, X., Wang, Y., Zhang, C., Zhang, F., 2021a. A modified U-net with a specific data argumentation method for semantic segmentation of weed images in the field. Comput. Electron. Agric. 187, 106242.
Zou, K., Chen, X., Zhang, F., Zhou, H., Zhang, C., 2021b. A field weed density evaluation method based on UAV imaging and modified U-Net. Remote Sens. 13, 310.
Zou, K., Liao, Q., Zhang, F., Che, X., Zhang, C., 2022. A segmentation network for smart weed management in wheat fields. Comput. Electron. Agric. 202, 107303.