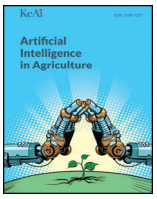
Artificial Intelligence in Agriculture 6 (2022) 189–198

Contents lists available at ScienceDirect

Artificial Intelligence in Agriculture

journal homepage: http://www.keaipublishing.com/en/journals/artificial-intelligence-in-agriculture/



Evaluation of model generalization for growing plants using conditional learning

Hafiz Sami Ullah, Abdul Bais *

Electronic Systems Engineering, Faculty of Engineering and Applied Science, University of Regina, Regina, Saskatchewan, Canada

ARTICLE INFO

Article history:
Received 20 September 2021
Received in revised form 24 September 2022
Accepted 25 September 2022
Available online 28 September 2022

Keywords:
Weed detection
Semantic segmentation
Adversarial training
Late germination
Sugar beet
Crop segmentation
Growing plants
Domain change

ABSTRACT

This paper aims to solve the lack of generalization of existing semantic segmentation models in the crop and weed segmentation domain. We compare two training mechanisms, classical and adversarial, to understand which scheme works best for a particular encoder-decoder model. We use simple U-Net, SegNet, and DeepLabv3+ with ResNet-50 backbone as segmentation networks. The models are trained with cross-entropy loss for classical and PatchGAN loss for adversarial training. By adopting the Conditional Generative Adversarial Network (CGAN) hierarchical settings, we penalize different Generators (G) using PatchGAN Discriminator (D) and L1 loss to generate segmentation output. The generalization is to exhibit fewer failures and perform comparably for growing plants with different data distributions. We utilize the images from four different stages of sugar beet. We divide the data so that the full-grown stage is used for training, whereas earlier stages are entirely dedicated to testing the model. We conclude that U-Net trained in adversarial settings is more robust to changes in the dataset. The adversarially trained U-Net reports 10% overall improvement in the results with mIOU scores of 0.34, 0.55, 0.75, and 0.85 for four different growth stages.

1. Introduction

Weeds are undesirable plants that can be eliminated using herbicides. Traditionally, these herbicides are used uniformly over the whole field. This uniform spraying eliminates the weeds but affects the crop as well. It also consumes an unnecessary amount of herbicide, adversely affecting the environment and human health. The amount of herbicide usage can be decreased by selective spraying. The agricultural robots can perform weed and crop detection to perform precise spraying, which will help reduce herbicides' usage. However, these robots lack a universal weed detection mechanism making them unfit for commercial use. In addition, typical methods for detecting weeds do not translate to out-of-distribution data; therefore, each new weed type or field needs to be updated, which is time-consuming.

Solving the weed detection problem using the colour, skeleton, and area-based hand-crafted features is prone to error, as the plant's shape undergoes variations due to growth, time of the day, weather conditions, and temperature. Therefore, these hand-crafted features-based classifications are only practical in situations with limited variations in the data. The Convolutional Neural Network (CNN) based encoder-decoder architectures are reliable for weed detection using semantic segmentation (Milioto et al., 2017), but a system that can work for multiple stages with different data distribution is still required.

This paper proposes a solution for better model generalization. In this work, we perform experiments to achieve a more reliable model tuning that can be implemented on a robot to select weeds. Furthermore, the resultant model performs better on the data on which it is trained and other growth stages without any additional training requirement. The dataset used for this research consists of images collected from multiple stages of sugar beet crop (Chebrolu et al., 2017). It has nineteen stages which are divided into four sections. The first three sections have five stages each, and the last section consists of the remaining four stages.

In semantic segmentation, given the limited access to labelled data and difficulty annotating at the pixel level, the generalization of the model parameters for the unknown stage of the plant becomes difficult. To articulate that, we explore the capability of different models to segment the crop and weeds from different stages when trained in an adversarial (using CGAN) and classical fashion. The main focus is to train different models to adapt to the unseen information given minimal access to data, i.e., training only for the last four stages. The target dataset has varying illumination, multiple weeds, and different soil texture. Using a fully grown stage for understanding, we are searching for a mechanism to improve the model performance for different stages of the crop. In addition, we focus

* Corresponding author.
E-mail addresses: hsz248@uregina.ca (H.S. Ullah), abdul.bais@uregina.ca (A. Bais).

https://doi.org/10.1016/j.aiia.2022.09.006

on generalizing the model's parameters for changing soil texture and lighting conditions. In this work, our contributions are:

- Providing a thorough study about the effects of changing Generator (G) in CGAN
- Finding the potential of one training scheme over another given limited access to data

We compare the performance of different models when trained in classical fashion, where the model has direct access to the segmentation mask, and in adversarial settings, where the segmentation mask is fed through a discriminator as a condition. The classical training is performed using weighted cross-entropy loss, whereas the adversarial training is conducted using L1 loss and patch-wise discriminator loss.

The CGAN (Isola et al., 2017) is used for adversarial training. The CGAN in (Isola et al., 2017) consists of two individual models, D and G. The D tries to classify the input as fake or real, and the G's objective is to generate realistic fake images to fool D. Fig. 1 shows CGAN settings and how input flows through the model. The experiments are based on both classically and adversarially trained encoder-decoder models. Firstly, we experimented with different encoder-decoder models trained in classical fashion. The input to the model is four channelled (RGB-NIR) images. Secondly, the same models are trained in adversarial settings. We replace G with the earlier used encoder-decoder models. These encoder-decoder models include U-Net (Ronneberger et al., 2015), SegNet (Badrinarayanan et al., 2017), U-Net and SegNet with ResNet-50 as a backbone, and DeepLabv3+ (Chen et al., 2018). A comparison is made between the models and the training mechanism. This comparison helps to fully understand the capability of models and training settings to perform semantic segmentation for growing plants.

The paper is divided into five sections. In Section 2, we present the related work where methods for weed detection are reviewed. The data analysis, experiments, and description of the model architectures are described in Section 3. Section 4 presents the results with a comparison of model performance. Finally, we conclude the paper in Section 5.

2. Related work

Existing weed detection methods could be broadly categorized into two categories; feature-based and CNN based. Furthermore, CNN-based methods are divided into two, one based on classical training and the other with adversarial training.

Haug et al. propose a feature-based detection method using Random Forests Classifier (RFC) (Breiman, 2001). First, they extract Normalized Difference Vegetation Index (NDVI) mask for each overlapping 80 × 80 pixels window, followed by the computation of 15 key points based on the shape of the plant and pixel intensities. Next, users draw class-based polygons which assign the class label to nearby pixels. The features and labels are fed to the classifier for training (Haug et al., 2014). A similar approach is followed by Lottes et al., where Local Binary Patterns (LBP) are used to find statistical features over R, G, and NDVI channels. The LBP operator generates a binary pattern based on the centre pixel value within eight neighbours. Shape-based features are calculated for vegetation masks. These features are then used to train RFC to detect weeds and plants (Lottes et al., 2016). These methods depend on shape-based features, which vary for the same or growing stage of the plant.

Milioto et al. propose CNN based classifier for weed detection. The vegetation mask is used for detecting connected vegetation blobs of size 64 × 64 pixels, and bounding boxes are computed around each blob. The RGB + NIR blobs are used as input to CNN, which classifies them in a vector with the probability distribution of the input image patch as a plant or a weed (Milioto et al., 2017). Mortensen et al. propose a modified version of the VGG-16 (Simonyan and Zisserman, 2014) model to perform the semantic segmentation task achieving pixel accuracy of 79% (Mortensen et al., 2016). Milioto et al. work on weed detection in data having different distributions. Initially, the encoder-decoder-based model, modification of SegNet (Badrinarayanan et al., 2017), is trained for uniformly distributed data, and performance is measured for the same and cross datasets (Milioto et al., 2018).

Cicco et al. propose plant modelling for enhancing the dataset and use the SegNet model (Badrinarayanan et al., 2017) for semantic segmentation of weeds and crops. They design different leaf models and then model other plants based on the type and number of leaves. These plants with varying sizes of leaves representing various growth stages are randomly distributed over the soil. The respective class is spread with the class semantic mask on a transparent background for ground truth. The generated dataset is fed to the SegNet model for segmentation (Di Cicco et al., 2017). This solution helps enhance data to improve accuracy. McCool et al. use encoder-decoder architecture to perform weed, and crop segmentation tasks (McCool et al., 2017). Lottes et al. use a decoder with dilated 3D convolution to visualize the weed and crop. This technique uses a sequence of non-overlapping images chosen using the odometry information. This method proposes the sequential module for extracting the pattern of the plantation. They achieve the mDICE score of 0.86 for the cross-validation dataset (Lottes et al., 2018). The common challenges with the above schemes are variant features utilization and lack of generalization for growing plants.

Gated-shaped CNN (Takikawa et al., 2019) improves the semantic segmentation task by processing information in two streams. Instead of processing colour, shape, and texture information in a single traditional network, a shape-based stream is proposed. The shaped-based stream focuses on gradient computation for the given mask. The intermediate layers in the shape stream are connected using Gated Convolutional Layers (GCL). Finally, features from the regular and shape streams are fused using Atrous Spatial Pooling Pyramid (ASPP). They report an overall mean Intersection Over Union (mIOU) of 80.8% for the cityscape dataset.

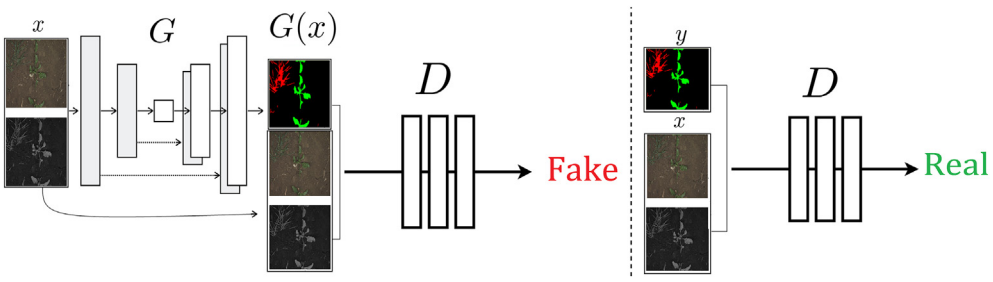


Fig. 1. The CGAN for image translation: G represents generator whereas D represents Discriminator, x represents input (RGB and NIR) image, G outputs RGB semantic map that is fed to D with the fake label. In the next iteration real semantic map is used along with x input to only train D with a real label.

Asad and Bais propose a novel approach for agricultural image labelling with promising results for weed detection (Asad and Bais, 2020). Images from the canola field are first segmented into background and foreground, then the weeds are manually labelled. This labelling mechanism is not fully automatic, but still, it reduces the complexity of manual labelling. The segmentation masks with field images are fed to two different models, SegNet (Badrinarayanan et al., 2017), and U-Net (Ronneberger et al., 2015). With mIOU of 82.9%, SegNet based on ResNet-50 outperforms U-Net architecture. Model-assisted labelling under different challenges is adopted in (Ullah et al., 2021) for crop and weed segmentation in canola. Chen et al. propose an efficient way of semantic segmentation. They use the ASPP module to extract multi-scale information. ResNet and Xception-based backbone is used to extract high-level features. These features are placed at multiple scales in the encoder. The decoder takes low-level information from the feature extractor and combines it with high-level features to target the semantic map. They name their model DeepLabv3+, reporting 89% mIOU for Cityscapes dataset (Chen et al., 2018).

Wang et al. use different image enhancement techniques for improving weed, and crop segmentation under varying lighting (Wang et al., 2020). They use histogram equalization and deep photo enhancer (GAN is used to perform enhancement) for image enhancement. Then deeplabv3+ (Chen et al., 2018) model is trained reporting 88.91% mIOU.

Semantic segmentation can also be done using CGAN. Most of the recent research on CGAN focuses on image translation either from text or an image (Karras et al., 2019; Reed et al., 2016; Wang et al., 2018; Zhang et al., 2017). Reed et al. use GANs to convert textual features to generate an image. The randomly distributed noise with text description is fed to the generator. The discriminator is conditioned to the image with the text attributes. This idea is useful for generating artificial data (Reed et al., 2016). Zhang et al. improve image generation from text using two-stage stack GAN. The stage I generator extracts sketches and the stage II generator refines the results (Zhang et al., 2017). The text description to image translation can be used as a data enhancement technique. In the image-to-image translation, Isola et al. suggest semantic segmentation on the cityscape dataset using CGAN. The scene image is fed to an encoder-decoder-based G, and the output of G is forwarded to D. D is trained for ground truth and generated a segmentation mask image for a given scene image. With pixel accuracy of 86%, generated images look close to the ground truth, but there is noise. The reason for this noise is instability in G (Isola et al., 2017).

Karacan et al. utilize semantic layout to synthesize realistic outdoor images. They use CGAN that learns target content to be drawn inside layout (Karacan et al., 2016). Rezaei et al. propose CGAN for segmenting brain tumours. Utilizing U-Net (Haug et al., 2014)e20 as generator and Markovian GAN (Radford et al., 2015) based discriminator, they report 0.68 mDICE score (Rezaei et al., 2017). Wang et al. use CGAN for high-resolution image translation. They use 70 × 70 Patch-GAN (Isola et al., 2017) for discriminator networks. Their focus is to transform the semantic map into a realistic image. They also report the inverse transformation achieving 63.9% mIOU for Cityscapes dataset (Wang et al., 2018). Regmi et al. work on natural image synthesis and pixel-level segmentation using CGAN. The generator is configured to take an aerial image and convert it to a street map view and the corresponding semantic map. The model consists of one encoder and two decoder streams. They achieve mIOU of 41.3% (Regmi and Borji, 2018). Rammy et al. propose retinal vessel segmentation using conditional patch-based GAN. The conditional sample with noise vector z of latent space is fed to the generator leading to a synthetic map. This synthetic map is used in the discriminator as a fake sample update. They report 98.84% specificity for the proposed method (Rammy et al., 2019). To the best of our knowledge, CGANs were not previously used for weed segmentation. We aim to improve weed detection for the cross stages dataset without relying on hand-designed features and to reduce G instability in CGAN.

3. Data analysis and experiments

In this work, we performed a comparison-based study to understand the effectiveness of model generalization under adversarial and classical training settings. Our focus is to find an architecture as well as a training scheme that could effectively detect weeds and crops at multiple growth stages. In the following subsection, we perform Exploratory Data Analysis (EDA) to understand the underlying pattern and challenges in the data.

3.1. Data analysis

In the dataset, images are collected over the growing season of the sugar beet crop. Fig. 2 analyzes the percentage of Weed and Crop Leaf Area (WLA and CLA) over different growth stages. Eqs. (1) and (2) explains the calculation of percentage CLA and WLA respectively.

$$Percentage\ \ Crop\ \ Leaf\ \ Area = \frac{\sum Y_{crop}}{Total\ \ Number\ \ of\ \ pixels} \times 100 \tag{1}$$

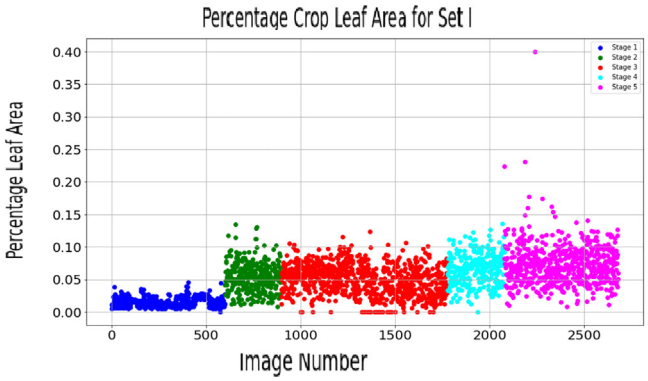
Y_{crop} represents pixels associated with the crop, and the pixel value can either be 1 or 0 for the channel associated with the crop.

$$Percentage\ \ Weed\ \ Leaf\ \ Area = \frac{\sum Y_{weed}}{Total\ \ Number\ \ of\ \ pixels} \times 100 \tag{2}$$

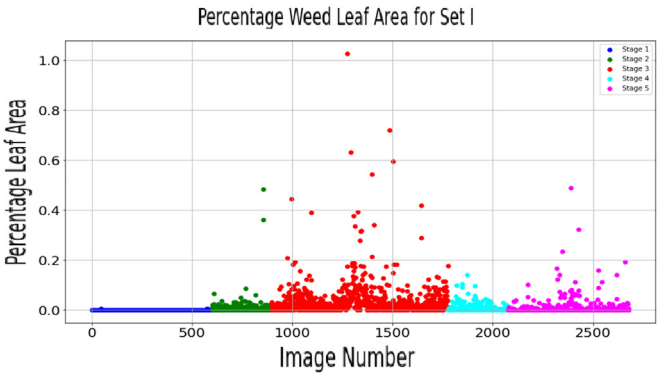
Y_{weed} represents pixels associated with the weed, pixel value 1 or 0 represent weed or no weed, respectively. The pixel value is fundamentally a one-hot encoded vector representing the pixel belonging to a particular class based on its hot value. For example, the vector with values [1,0,0] means the pixel belongs to the first class.

In Fig. 2, the plots in the first column indicate the percentage CLA, whereas the second column reports the percentage WLA. In Fig. 2 (i and ii), CLA and WLA of images from Set I, that consist of five initial timestamps, are reported. The percentage leaf area gives us an indication of changing shape and size. In Fig. 2 (a), for stage one, the crop leaf area is a small fraction (approx. 0.05%), whereas in (b), the weed percentage is near zero, i.e., there is no weed at the early stage. This is represented with blue in Fig. 2(a and b). The weedy plants start popping out at Stage 3 in Set I (Fig. 2(b)), which is more than the crop as the average leaf area is approximately 0.2% compared to the crop that is 0.11%. As we move forward in Set II, Fig. 2 (c and d) show prominent growth for the crop, but for weeds, there is no consistent growth. The approximate mean CLA is near 0.11% for the crop, whereas for weeds, it is close to 0.03%. In Fig. 2(e and f), the growing trend of percentage CLA continues with an approximate mean of 0.65%, whereas the percentage WLA has an approximate mean of 0.10%. In Set IV, the average WLA is about 0.82%, whereas for the crop, the average increases to 4.52% as shown in Fig. 2(g and h). This analysis aims to understand the underlying challenges associated with plant growth, especially when the task is to segment it. Going from a minimal growth stage (where the plant is barely visible) to a stage where the average CLA in the images from Set IV containing a crop plant is almost 10% (CLA of Set IV with crop plants), a lot is happening with the plant's shape. This diversity is complex to generalize if we do not access any prior information. Another critical information that can be extracted from Fig. 2 is the percentage distribution of classes of interest. Looking at WLA and CLA, one can tell that the dataset is highly imbalanced. For example, there are more pixels in the background data than weed and crop. For a model to understand various distributions like the ones explained through Fig. 2, we, unfortunately, sometimes do not have access to enough data.

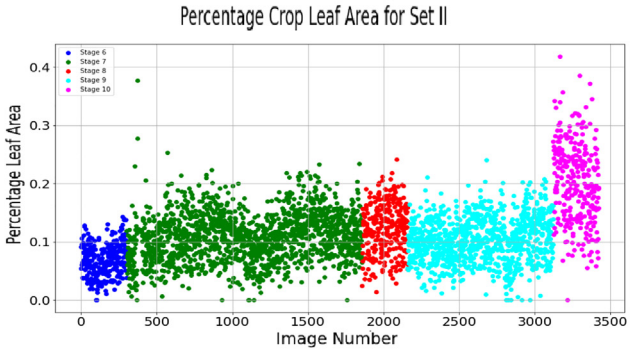
The percentage of pixels for each class and the sets of images is reported in Table 1. For example, Set IV has the highest crop and weed pixels percentage compared to the other three sets. On the other hand, pixels in Set I mainly belong to the background, and only three-



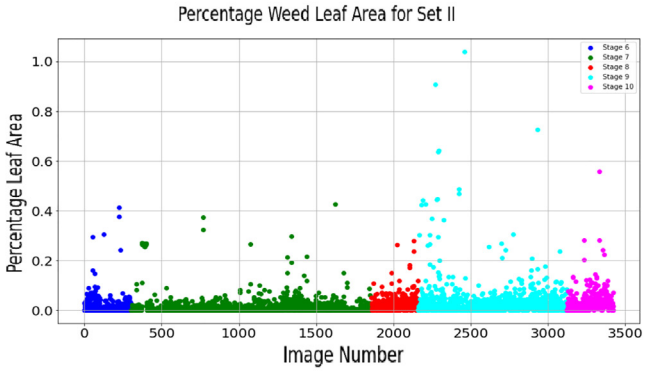
(a) Set I Crop Leaf Area



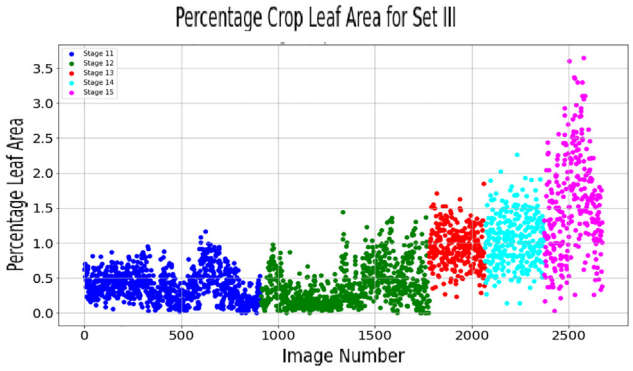
(b) Set I Weed Leaf Area



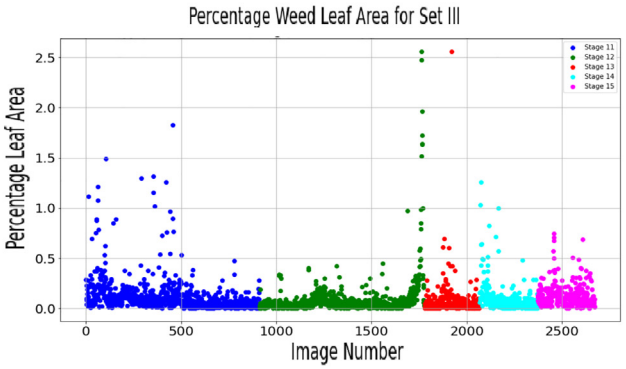
(c) Set II Crop Leaf Area



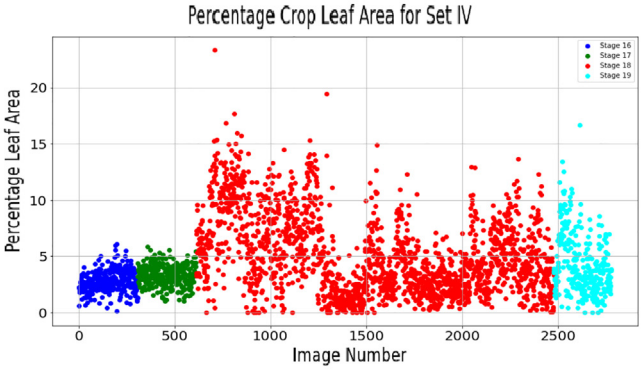
(d) Set II Weed Leaf Area



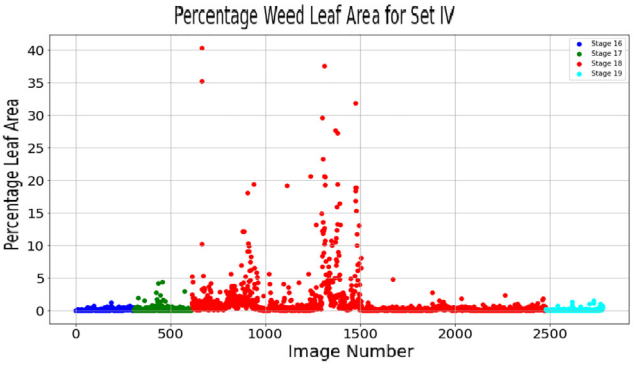
(e) Set III Crop Leaf Area



(f) Set III Weed Leaf Area



(g) Set IV Crop Leaf Area



(h) Set IV Weed Leaf Area

Fig. 2. Data analysis using percentage crop and weed leaf area distribution. A total of 19 timestamps (stages) are grouped to make 4 sets. The first three sets contain 5 timestamps each, whereas the last group (Set IV) has the final three stages.

Table 1
Percentage-wise pixel distribution of Weed, Crop, Background the data.

Set	Images	Percentage Distribution of Pixels		
		Weed	Crop	Background
I	2679	0.02%	0.05%	99.93%
II	3423	0.03%	0.11%	99.86%
III	2673	0.10%	0.65%	99.25%
IV	2800	0.82%	4.52%	94.66%

quarters of a percent have weeds and crop. This analysis is essential to study as it helps define the solution to the problem of data imbalance.

In this work, we address the discussed problem by exploring the generalization capacity of different models under two training settings with minimal exposure to the shape-based diversity of data.

3.2. Experiments

The diversity of a plant's shape is a significant challenge and hard to address if data is unavailable. In our case, although we have access to the data from different plant growth stages, the aim here is to limit access to the data and perform training only over the last few stages. We then evaluate the technique over the remaining data to understand the capacity of different models when trained using two methods. First, different state-of-the-art encoder-decoder models with multiple backbone structures are trained classically. In classical training, the model directly learns the mapping from a given field image to a semantic map, i.e. model has exposure to output segmentation mask. The same networks are trained in the second mechanism using adversarial training settings. While utilizing the CGAN model, we use the D as described in pix2pix (Isola et al., 2017) and change G with encoder-decoder architectures described earlier to see the variation in results. In CGAN, G generates images, and D classifies them as real or fake. D splits the image into patches and decides the patch of the image to be real or fake. This process continues until we get the balance between D and G. The background varies between stages, impacting the model performance. To reduce model dependency on background, we randomly blur the images using a Gaussian filter of size 5 and standard deviation of 1. The random blurring changes the entire image, not only the soil. The plants also get an effect of blurriness which helps in mimicking the vibrations in the ground imagery system.

In both settings, we train the models for 100 epochs with batch size 4 and save the model's best weights. We use Adam optimizer with a learning rate 0.0002 and β_1 0.5. In addition, we use one-hot encoded segmentation masks with weighted cross-entropy for classical training. The weighted cross entropy helps in mitigating the class imbalance problem. In adversarial settings, we use binary cross-entropy loss and mean absolute error (L1 loss) for D and G. It is called PatchGAN loss because the D generates a patch indicating the image as real or fake.

3.3. Objective function for classical training

Semantic segmentation using classical training is common. The size and shape of plants change with growth. If enough data is unavailable, these changes and the soil texture variations are hard to learn. Additionally, the dataset is imbalanced as there are more background pixels than weed and crop. Fig. 3 explains the distribution of the entire dataset. Only 1.46% of a total number of pixels consists of vegetation with 0.23% for weed and 1.28% for the crop. Almost 98.5% of the pixels are from the background class. The weighted cross-entropy loss helps in solving the data imbalance problem. For an unbalanced dataset, weighted cross entropy assigns loss weightage to each class. The weightage is set such that the class with the high number of samples gets a lower weight, and the class with the low number of samples contributes a higher portion towards overall loss. The focus of model tuning is directly related to the penalty for each class. The objective function for classical learning can be written as:

$$Loss(Y_{true}, Y_{pred}) = -\frac{1}{P}\sum_{p=1}^{P}\sum_{c=1}^{C} w_c Y_{true_{p,c}}\ log\left(Y_{pred_{p,c}}\right) \quad (3)$$

where $Y_{true} \in [0,1]$ is the ground truth and $Y_{pred} \in [0,1]$ is the softmax output. P represents the number of pixels and C denotes the total number of classes. w_c is weight assigned to class c.

For this work, the w_c is chosen by taking the inverse of the percentage of the data distribution.

3.4. Objective function for CGAN

The main objective while training CGAN is playing the min-max game. The training is assisted by generating real and fake labels for D. These labels are produced at each step. The real label corresponds to grid of ones and is only given when the ground truth is used. Zeros represent the fake label. They are given when the prediction from the generator is used. The output of D has a shape of 24×32. The labels are generated of the same size as the output of D. Initially, D takes field image and condition as input. The condition is the ground truth that G aims to learn. It can be called a real sample update. Then, D takes field image and prediction from G as input. This is a fake sample update. At the end of each real and fake sample update in D, the weights of G are updated. While updating the parameters for the G, we perform back-propagation through D without updating D's parameters. In other words, we update G with all the layers of D set to non-trainable. The learning rate of D is adjusted to half of the G using loss weight configuration so that G learns faster than D. Collectively the objective of CGAN can be expressed as below (Isola et al., 2017):

$$Loss_{CGAN}(G,D) = \mathbb{E}_{X,Y_{true}}[\ \log D(X, Y_{true})] + \mathbb{E}_{X,Z}[\ \log\ (1 - \mathrm{D}(\mathrm{X}, \mathrm{G}(\mathrm{X}, \mathrm{Z})] \quad (4)$$

X and Z represent the input image and the operation performed on the image, respectively. Y_{true} is the ground truth and only given to D. For G, we use mean absolute error (L1 loss) for parameter updates. L1 loss works well for low-frequency components by improving sparseness in the image. The sparsity is linked with having few pixels with a higher difference to the original. As the D and G work opposite, G restricts the high component and resists D to only model high frequency. For G, our objective equation is:

$$Loss_{L1}(G) = \mathbb{E}_{X,Y_{true},Z}[\|Y_{true} - G(X,Z)\|_1] \quad (5)$$

Overall, the end objective is to play the min-max game as given below:

$$G^* = arg\ \min_G\ \max_D\ Loss_{CGAN}(G,D) + \lambda Loss_{L1}(G) \quad (6)$$

λ is the tuning parameter to assign more weightage to G's loss. The value of λ is set to 100.

The best model is chosen based on the equilibrium of minimum losses of G and D. If D is trained at the same speed as the G, then D will outperform G and getting a balance will be difficult. Fig. 1 shows how the inputs flow through the network. G takes the RGB-NIR field image and generates the target segmentation mask. The output of D represents if each $M \times N$ grid in an image is real or fake. In other words, it gives a penalty that penalizes the D model at $M \times N$ patch of the image. The model architecture and the improvements made are described in the following subsections.

3.5. Model architectures

We use U-Net (Ronneberger et al., 2015), SegNet (Badrinarayanan et al., 2017), U-Net with ResNet-50 backbone, SegNet with ResNet-50

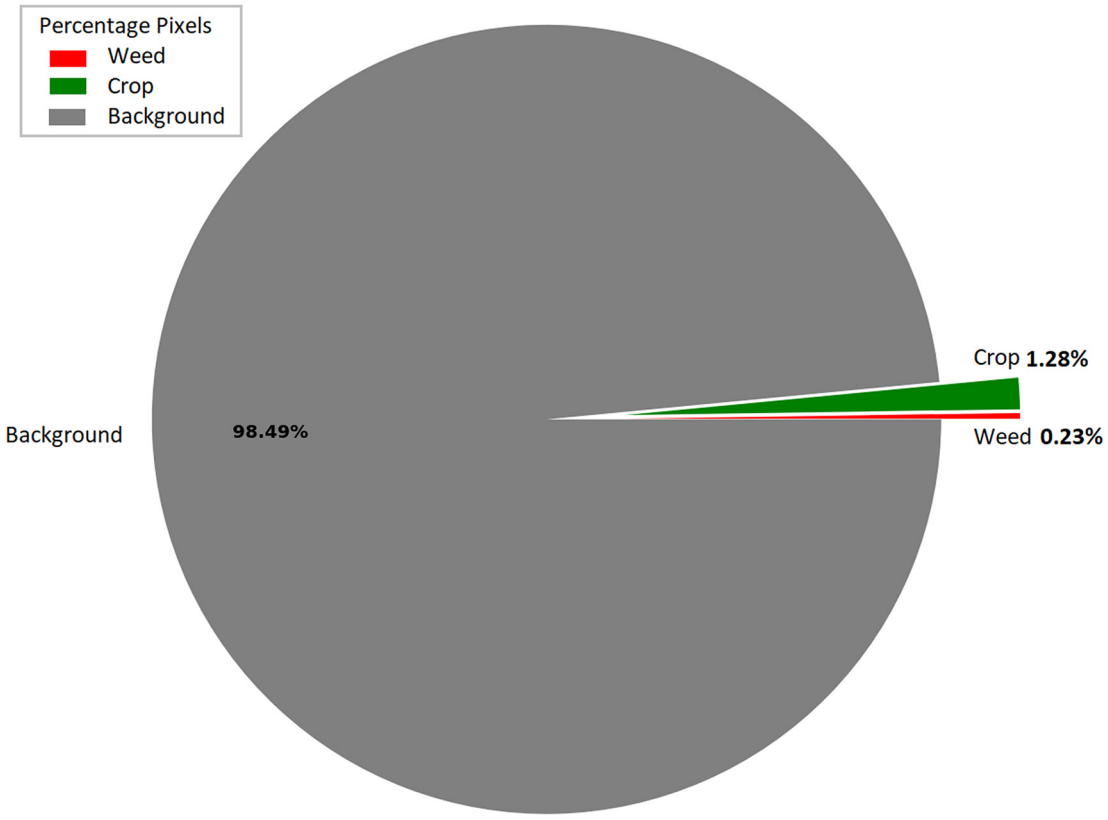


Fig. 3. Percentage of number of pixels for each class in the overall data.

backbone, and DeepLabv3+ (Chen et al., 2018) with ResNet-50 backbone to perform our experiments. The mentioned models are first trained using classical training mechanism and then using adversarial settings.

All the models take an input consisting of RGB and NIR channels. The dimension of the input image is set to 384×512. All the models learn to map the input image to a translation representing the class semantic mask. Each model has two main blocks in the architecture, the encoder and decoder block. The encoder finds the deep features while reducing the spatial size. The decoder performs inverse operation of mapping the features to image translation. The difference between SegNet and U-Net is the type of skip connections (Badrinarayanan et al., 2017; Ronneberger et al., 2015). The U-Net shares more information than SegNet, where only the max-pooling indices are shared in the future layers. The skip connections help to achieve additional stability in spatial configuration. In U-Net, the learnable up-sampling is implemented using transposed convolutional layer. DeepLabv3+ employs the Atrous Spatial Pyramid Pooling (ASPP) module of four different depth-wise separable dilated convolutional layers. Using different dilatation rates in each convolutional layer, this module probes features at multiple scales of the input feature map that helps in maintaining a more diverse spatial configuration. Table 2 compare the models based on trainable parameters. In Table 2 the first column represents the model name, whereas the second column denotes the number of trainable parameters. U-Net requires the highest number of trainable parameters when compared to other reported models. DeepLabv3+ requires only 23.96*M* parameters where *M* represents a million.

Table 2
Trainable parameters for different models.

Network	Parameters
U-Net	36.96 M
SegNet	29.46 M
U-Net-ResNet-50	38.05 M
SegNet-ResNet-50	34.88 M
DeepLabv3+	23.96 M

3.5.1. Generator architecture in CGAN

In CGAN settings, we use U-Net (Ronneberger et al., 2015), SegNet (Badrinarayanan et al., 2017), U-Net-ResNet-50, SegNet-ResNet-50, and DeepLabv3+ as G. The U-Net, SegNet, U-Net-ResNet-50, SegNet-ResNet-50, and DeepLabv3 + -ResNet-50 are named as C-U-Net, C-Segnet, C-U-Net-ResNet-50, C-SegNet-ResNet-50, and C-DeepLabv3+ when trained in adversarial settings. Using different models as G also helps study the variations in the performance for the same D model. The end layer of each decoder is a convolutional layer, after which tanh activation is applied. The output result is in the range of $[-1,1]$, which is then shifted and scaled to $[0,1]$.

3.5.2. Discriminator (patch wise penalizer)

There are altogether six convolutional layers in D with 4×4 filter size and 2×2 stride. First, it takes the source and the target image as input. Then, it uses sigmoid activation to find a 24×32 patch of binary output. The size of the patch can be altered based on the query as given in (Li and Wand, 2016) and (Isola et al., 2017).

4. Results and evaluation

For the evaluation of semantic segmentation tasks, the IOU and DICE score are reliable evaluation metrics to target. We use mean IOU score to compare our performance. The mIOU score can be calculated using Eq. (5).

Table 3
Number of images from each stage used in the test, train, and validation sets.

Stage	Data split		
	Train	Test	Validation
I	0	2679	0
II	0	3423	0
III	0	2673	0
IV	2000	500	300

Table 4
Comparison of classically trained models for different growth stages (mIOU).

	U-Net	SegNet	U-Net-ResNet-50	SegNet-ResNet-50	DeepLabv3+
	mean Intersection Over Union (mIOU)				
Stage I	**0.3296**	0.3267	0.3128	0.3113	0.3240
Stage II	**0.5438**	0.4792	0.3895	0.4029	0.4063
Stage III	0.7199	**0.7719**	0.7412	0.7022	0.7496
Stage IV	0.8088	0.8319	0.8403	0.7924	**0.8455**

$$IOU = \frac{Y_{true} \cap Y_{pred}}{Y_{true} \cup Y_{pred}} \tag{7}$$

We use the sugar beet images from the Bonn field captured at various growth stages in our experiments (Chebrolu et al., 2017). The weeds are mainly dicot and grass. The dataset consists of RGB and NIR images with a three-channel corresponding segmentation mask. The samples are taken over two months, starting from the zero growth stage. Each network is trained on last stage, it is then tested on unseen data from the same and the remaining three stages. We use 2000 images for training, and the rest of the data is used for testing. The data distribution is shown in Table 3. From Table 3 it is clear that only 17% data is used for training, which is a tiny fraction compared to the 80% testing data.

The results in Table 4 and Table 5 refer to classically and adversarially trained models, respectively. Both tables compare model performance based on mIOU score. Together with comparing training schemes, the results also give a bigger picture of the models' overall performance for different plant stages. The C-U-Net results in higher accuracy and IOU score for unknown stages. It reports about 10% overall improvements in mIOU score compared to other models.

Deep networks like U-Net and SegNet with ResNet-50 backbone perform better with classical training. This shows that deeper networks are not suitable for G in CGAN. As SegNet has similar architecture as U-Net, the adversarial training in C-SegNet should work similarly to C-U-Net, but U-Net comes with an advantage of enriched skip connections that makes U-Net preserve more information compared to SegNet. The DeepLabv3+, when trained in a classical training fashion, shows significantly good results for the trained stage, reporting an 84.5% mIOU score. U-Net-ResNet-50 is the second better performance model in the classical setting.

SegNet trained classically works well for the crop at the third stage, reporting a 77% mIOU score. As the plants grow, the mIOU score tends to increase. The score improves from 32 to 84% from the first stage to the last stage. This increasing trend shows that as the crop and weeds grow, they exhibit more resemblance with the training data.

Fig. 4 and Fig. 5 show results from four different stages, with ground truth and predicted segmentation mask. The weeds are red, green represents the crop, and everything else is classified as background. C-U-Net has better overall performance. However, its result is not suitable for the earliest growth stage. The model has difficulty in detecting weeds/crop at an early stage and produces noise instead. This is because of high variations in soil texture as depicted in the 1st row of Fig. 4 and Fig. 5.

Additionally, the difficulty of differentiating weeds and crops at an early stage may be the leading cause of poor performance. The soil variation can be addressed using artificial augmentation. Changing brightness and adding noise in the field images can tackle the soil variations effectively. However, due to the very close resemblance of weed and crop at this stage, it is hard to differentiate between crop and weed at the early stage.

The white rectangular boxes in Fig. 5 show some of the false detection (noise). These false detections are the cost of using CGAN as G does not see the true data distribution, making it unstable. To help G to get stable, we use noise removing technique on random images. Furthermore, we examine that randomly blurring images help us achieve better-generalized parameters for the model, making G stable. We refer to blurring as Z input in Eq. (2) and Eq. (3). The drop-out layers also help in approaching a more stable G.

There is an interesting fact about the dataset. Some of the pixels in-ground truths are either unlabelled or mislabelled. The connecting weeds are labelled such that the area in between the weeds is marked as weed class. These pixels belong to the background class. Moreover, some of the stem pixels are labelled as background. The stem pixels either belong to weed or crop but not the background. Likewise, some pixels are labelled as background, but their actual class is either crop or weed. Nearly all our models detect most of the unlabelled or mislabelled pixel classes in the ground truth. Comparing false labelled data with good predictions raise a concern about results being better than reported. The yellow rectangular boxes in Fig. 4 and Fig. 5 represent the visuals of our claim.

The changing moisture levels of soil may lead to late germination of crops. In Fig. 5 (t) and (x) the green rectangular boxes highlight some of the late germinated plants. Our model detects them as a weed because of mislabelled data. The labels in the dataset do not account for the late germinated plants. It makes the weed and crop segmentation confusing and complex for the model to learn. Using already labelled data with improvements can lead to more noticeable results.

Also, one can see the performance drop of the model in different sets, multiple factors might be hurting the performance, soil moisture, weather, camera settings, Ground Sample Distance (GSD), and time of the day images were collected being some of them. GSD normalization, classical domain adaptation (Histogram matching), and dropping samples to balance the data can be some valuable techniques that might elevate the model performance.

Table 5
Comparison of adversarially trained models for different growth stages (mIOU).

	C-U-Net	C-SegNet	C-U-Net-ResNet-50	C-SegNet-ResNet-50	C-DeepLabv3+
	mean Intersection Over Union (mIOU)				
Stage I	**0.3434**	0.3357	0.3356	0.3336	0.3344
Stage II	**0.5488**	0.3886	0.4757	0.4501	0.4967
Stage III	**0.7500**	0.6756	0.6594	0.5837	0.6183
Stage IV	**0.8472**	0.7355	0.6905	0.6335	0.7023

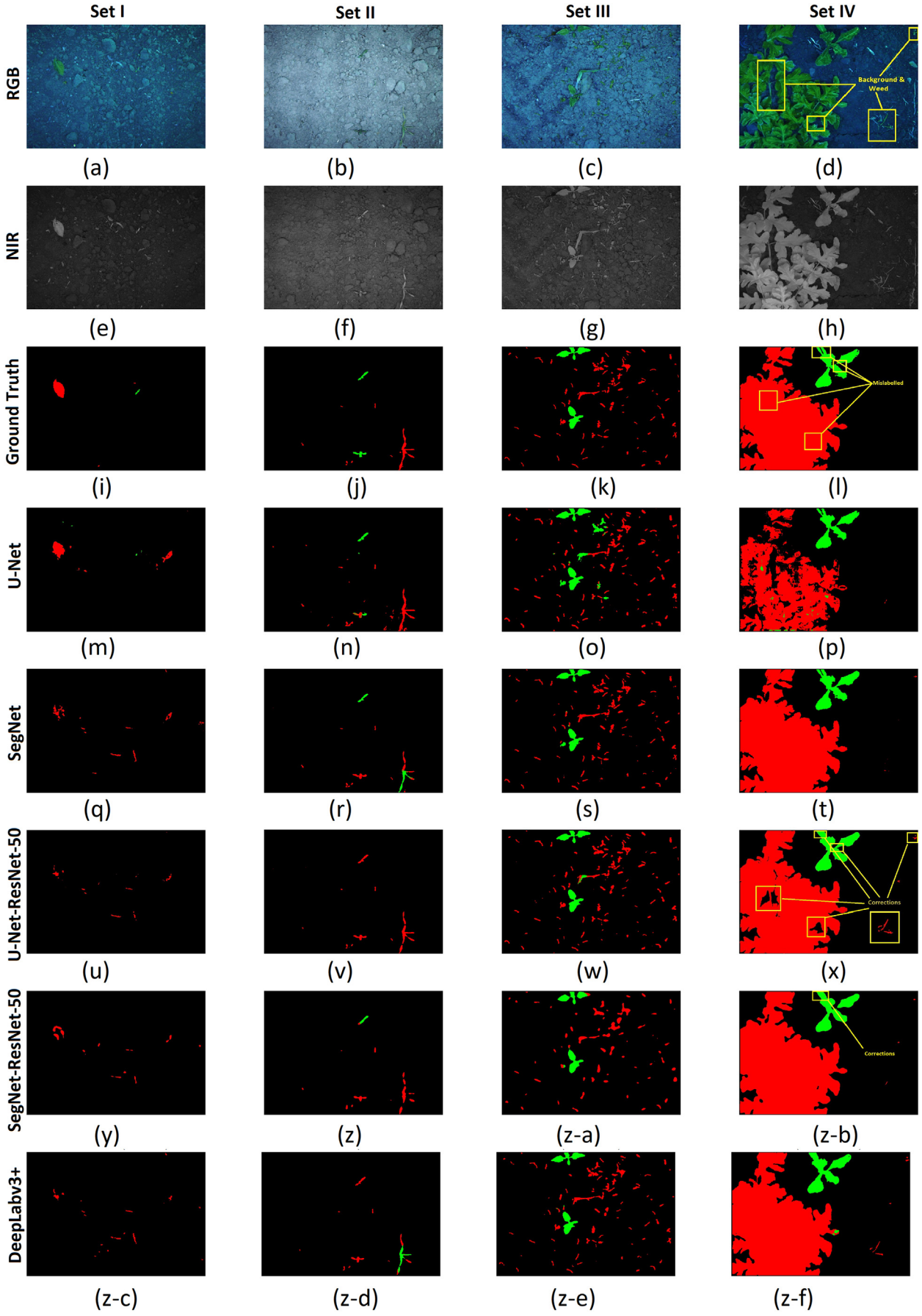


Fig. 4. Comparison of models at four different growth stages: 1st row has RGB field images, 2nd row is NIR mask, and the rest of the rows follow the sequence of Ground truth, U-Net, SegNet, U-Net-ResNet-50, SegNet-ResNet-50, and DeepLabv3+.

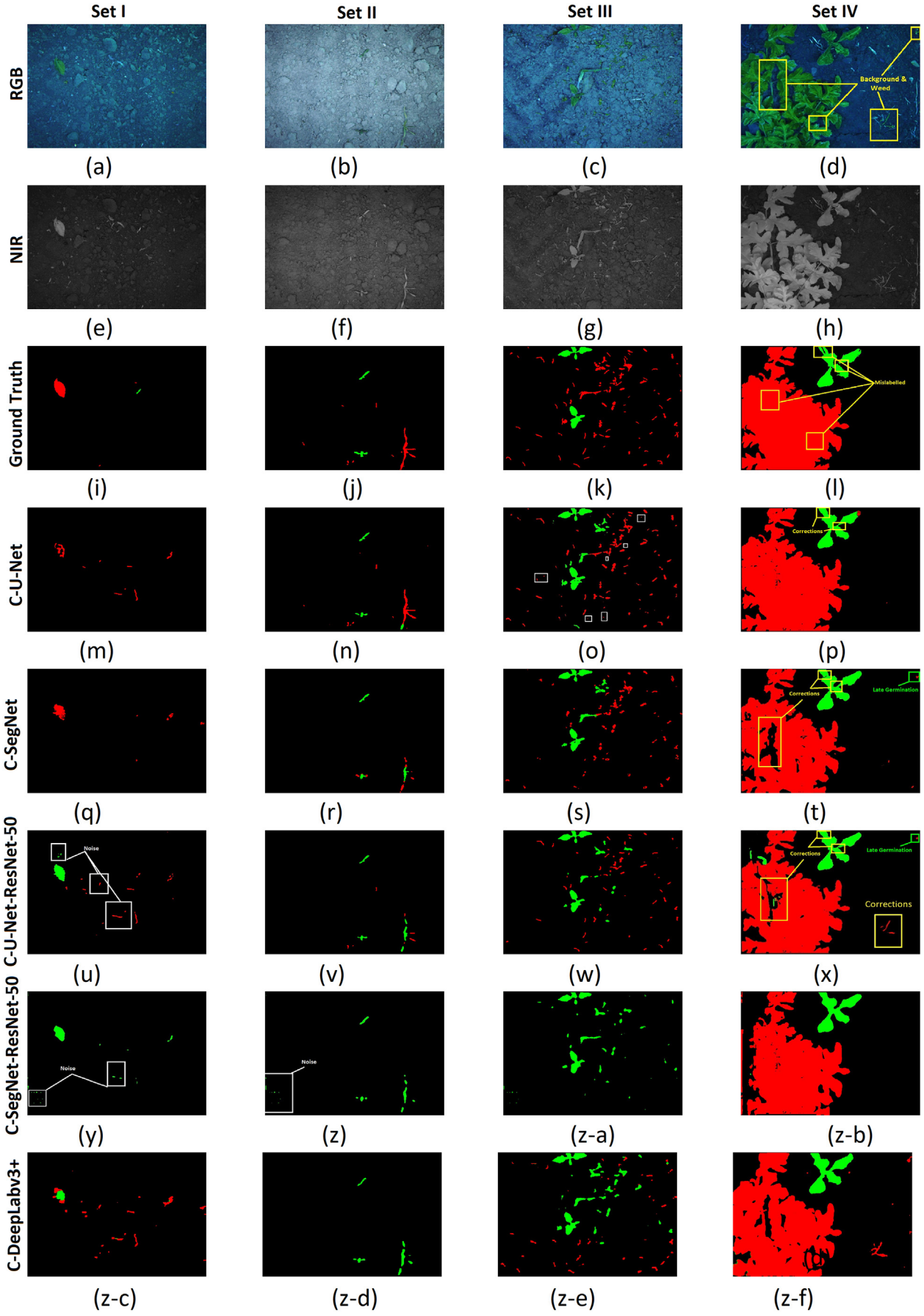


Fig. 5. Comparison of models at four different growth stages: 1st row has RGB field images, 2nd row is NIR mask, and the rest of the rows follow the sequence of Ground truth, C-U-Net, C-SegNet, C-U-Net-ResNet-50, C-SegNet-ResNet-50, and C-DeepLabv3 + .

5. Conclusion

This paper studied the impact of using different generators in CGAN. We performed a semantic segmentation task for weed detection using classical and adversarial learning. We examined different stages of the sugar beet crop by training our models on only one stage. We discuss possible improvements in a dataset that can lead to better results. Using U-Net in CGAN settings leads to more generalized weights than simple classical training, which is evident from the quantitative and qualitative analysis of results collected over different stages. The C-U-Net outperforms in detecting the growth variations in the dataset. Compared with other networks, C-U-Net shows improvements of 10% in mIOU score. The results can be improved if the dataset has equal distribution of all classes and the mislabelled data is corrected. The other way is to use artificial augmentation to enhance the dataset, which will boost accuracy. There is a need to improve the penalizer. PatchGAN loss is not very helpful for training deep models. In particular, our next research will focus on enhancing D and weed detection for an imbalance class data with a focus on late germination of the plant.

CRediT authorship contribution statement

Hafiz Sami Ullah: Conceptualization, Methodology, Software, Writing – original draft. **Abdul Bais:** Supervision, Validation, Conceptualization.

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.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant (RGPIN-2021-04171) entitled "Crop Stress Management using Multi-source Data Fusion.

References

Asad, M.H., Bais, A., 2020. Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network. Inform. Proc. Agricult. 7, 535–545. https://doi.org/10.1016/j.inpa.2019.12.002.

Badrinarayanan, V., Kendall, A., et al., 2017. Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39, 2481–2495.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32.

Chebrolu, N., Lottes, P., et al., 2017. Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. Int. J. Robot. Res. 36, 1045–1052.

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.), Computer Vision – ECCV 2018. Springer International Publishing, Cham, pp. 833–851.

Di Cicco, M., Potena, C., et al., 2017. Automatic model based dataset generation for fast and accurate crop and weeds detection. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 5188–5195.

Haug, S., Michaels, A., et al., 2014. Plant classification system for crop/weed discrimination without segmentation. IEEE Winter Conference on Applications of Computer Vision. IEEE, pp. 1142–1149.

Isola, P., Zhu, J., et al., 2017. Image-to-image translation with conditional adversarial networks. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 1125–1134.

Karacan, L., Akata, Z., et al., 2016. Learning to generate images of outdoor scenes from attributes and semantic layouts. arXiv preprint. arXiv:1612.00215.

Karras, T., Laine, S., et al., 2019. A style-based generator architecture for generative adversarial networks. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 4401–4410.

Li, C., Wand, M., 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. The European Conference on Computer Vision (ECCV). Springer, pp. 702–716.

Lottes, P., Hoeferlin, M., et al., 2016. An effective classification system for separating sugar beets and weeds for precision farming applications. IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 5157–5163.

Lottes, P., Behley, J., et al., 2018. Fully convolutional networks with sequential information for robust crop and weed detection in precision farming. IEEE Robot. Automat. Lett. 3, 2870–2877.

McCool, C., Perez, T., et al., 2017. Mixtures of lightweight deep convolutional neural networks: applied to agricultural robotics. IEEE Robot. Automat. Lett. 2, 1344–1351.

Milioto, A., Lottes, P., et al., 2017. Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. IV-2/W3, pp. 41–48. https://doi.org/10.5194/isprs-annals-IV-2-W3-41-2017.

Milioto, A., Lottes, P., et al., 2018. Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 2229–2235.

Mortensen, A.K., Dyrmann, M., et al., 2016. Semantic segmentation of mixed crops using deep convolutional neural network, in. Proceeding of the International Conference of Agricultural Engineering (CIGR).

Radford, A., Metz, L., et al., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint. arXiv:1511.06434.

Rammy, S.A., Abbas, W., Hassan, N.U., Raza, A., Zhang, W., 2019. CPGAN: conditional patch-based generative adversarial network for retinal vessel segmentation. IET Image Process. 14, 1081–1090.

Reed, S., Akata, Z., et al., 2016. Generative adversarial text to image synthesis. International Conference on Machine Learning, pp. 1060–1069.

Regmi, K., Borji, A., 2018. Cross-view image synthesis using conditional gans. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 3501–3510.

Rezaei, M., Harmuth, K., et al., 2017. A conditional adversarial network for semantic segmentation of brain tumor. International Medical Image Computing and Computer Assisted Intervention (MICCAI) Brainlesion Workshop. Springer, pp. 241–252.

Ronneberger, O., Fischer, P., et al., 2015. U-net: convolutional networks for biomedical image segmentation. International Conference on Medical Image Computing and Computer-assisted Intervention. Springer, pp. 234–241.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint. arXiv:1409.1556.

Takikawa, T., Acuna, D., et al., 2019. Gated-SCNN: gated shape cnns for semantic segmentation. Proceedings of the IEEE International Conference on Computer Vision, pp. 5229–5238.

Ullah, H.S., Asad, M.H., Bais, A., 2021. End to end segmentation of canola field images using dilated u-net. IEEE Access 9, 59741–59753. https://doi.org/10.1109/ACCESS.2021.3073715.

Wang, T.C., Liu, M.Y., et al., 2018. High-resolution image synthesis and semantic manipulation with conditional gans. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 8798–8807.

Wang, A., Xu, Y., et al., 2020. Semantic segmentation of crop and weed using an encoder-decoder network and image enhancement method under uncontrolled outdoor illumination. IEEE Access 8, 81724–81734.

Zhang, H., Xu, T., et al., 2017. StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. Proceedings of the IEEE International Conference on Computer Vision, pp. 5907–5915.