TYPE Original Research
PUBLISHED 09 August 2023
DOI 10.3389/fpls.2023.1234616

OPEN ACCESS

EDITED BY
Hao Lu,
Huazhong University of Science and Technology, China

REVIEWED BY
Yanan Li,
Wuhan Institute of Technology, China
Yinglun Li,
Nanjing Agricultural University, China

*CORRESPONDENCE
Hyongsuk Kim
hskim@jbnu.ac.kr

RECEIVED 05 June 2023
ACCEPTED 06 July 2023
PUBLISHED 09 August 2023

CITATION
Ilyas T, Lee J, Won O, Jeong Y and Kim H (2023) Overcoming field variability: unsupervised domain adaptation for enhanced crop-weed recognition in diverse farmlands.
Front. Plant Sci. 14:1234616.
doi: 10.3389/fpls.2023.1234616

Overcoming field variability: unsupervised domain adaptation for enhanced crop-weed recognition in diverse farmlands

Talha Ilyas[1,2], Jonghoon Lee[2], Okjae Won[3], Yongchae Jeong[2,4] and Hyongsuk Kim[2,4*]

[1]Division of Electronics and Information Engineering, Jeonbuk National University, Jeonju-si, Republic of Korea, [2]Core Research Institute of Intelligent Robots, Jeonbuk National University, Jeonju-si, Republic of Korea, [3]Production Technology Research Division, National Institute of Crop Science, Rural Development Administration, Miryang, Republic of Korea, [4]Division of Electronics Engineering, Jeonbuk National University, Jeonju-si, Republic of Korea

Recent developments in deep learning-based automatic weeding systems have shown promise for unmanned weed eradication. However, accurately distinguishing between crops and weeds in varying field conditions remains a challenge for these systems, as performance deteriorates when applied to new or different fields due to insignificant changes in low-level statistics and a significant gap between training and test data distributions. In this study, we propose an approach based on unsupervised domain adaptation to improve crop-weed recognition in new, unseen fields. Our system addresses this issue by learning to ignore insignificant changes in low-level statistics that cause a decline in performance when applied to new data. The proposed network includes a segmentation module that produces segmentation maps using labeled (training field) data while also minimizing entropy using unlabeled (test field) data simultaneously, and a discriminator module that maximizes the confusion between extracted features from the training and test farm samples. This module uses adversarial optimization to make the segmentation network invariant to changes in the field environment. We evaluated the proposed approach on four different unseen (test) fields and found consistent improvements in performance. These results suggest that the proposed approach can effectively handle changes in new field environments during real field inference.

KEYWORDS

crop-weed recognition, domain adaptation, precision agriculture, artificial intelligence, crop phenotyping, agricultural operations

1 Introduction

Deep Learning (DL) techniques have been successful in detecting and recognizing objects in images and videos. These techniques are now being applied to agriculture, particularly in the automatic detection and classification of weeds (Khan et al., 2020). This is a difficult problem because weeds and crops often have similar colors (green vs green), shapes, and textures (Adhikari et al., 2019; Sarvini et al., 2019). Weeds are plants that negatively impact crop growth and yields by competing for resources such as water, sunlight, air, and nutrients. They can also interfere with crop growth through the release of chemicals (Patel and Kumbhar, 2016; Iqbal et al., 2019). Effective weed control is therefore necessary to support crop growth. In addition, what is considered a weed in one setting may be a crop in another. The increasing global population, expected to reach 9 billion by 2050, will require a 70% increase in agricultural production (Radoglou-Grammatikis et al., 2020). However, the agricultural industry will face challenges such as limited cultivation land and the need for more intensive production. Climate change and water scarcity will also impact productivity. Precision agriculture can help address these challenges (Lal, 1991; Seelan et al., 2003).

Farmers must use various strategies to control weeds, including preventative measures (manual weeding), cultural techniques like field hygiene (low weed seed bank), mechanical methods like mowing and tilling, biological methods like using natural enemies of weeds (insects or grazing animals), and chemical methods such as herbicide application (Tu et al., 2001; Melander et al., 2005). Automated weed control systems, which can reduce labor costs and minimize herbicide use, have become desirable as labor costs have increased and concerns about health and the environment have grown (Durmuş et al., 2015; Nicolopoulou-Stamati et al., 2016). Moreover, due to a lack of interest among younger people in joining the agriculture industry, there is a shortage of labor (Sarvini et al., 2019). This shortage, combined with the need for efficient and cost-effective weed control, has made automated weeding methods more necessary than ever before (Lameski et al., 2018).

On other hand automated weed detection systems follow a series of steps to identify and classify weeds in images. These steps include acquiring images, pre-processing them, extracting features, and detecting and classifying weeds (Pantazi et al., 2016; Parra et al., 2020). Deep learning approaches have been successful in achieving accurate results in recognizing crops and weeds in real-world conditions (Li and Tang, 2018). The key challenge in these systems is distinguishing between crops and weeds (Khan et al., 2020; Matloob et al., 2020). These systems typically use fully convolutional networks (FCNs) to perform semantic segmentation, which involves labeling each pixel in an image with a specific class (such as crop or weed) (Parra et al., 2020; Coleman et al., 2022).

One of the main challenges in developing an automatic weed management system is accurately detecting and recognizing weeds in crops. This can be difficult because weeds and crops often have similar colors, textures, and shapes, and may appear differently at different growth stages (Sarvini et al., 2019; Khan et al., 2020; Ilyas et al., 2022). Other challenges include occlusion, variations in color and texture due to lighting and illumination, and the presence of motion blur and noise in images (Sa et al., 2016). The species of weeds can also vary based on geographical location, crop variety, weather conditions, and soil type (Kriticos et al., 2006). All of these factors can make it difficult to classify plants accurately.

Several studies have shown advancements in this area. For example, Tavakoli et al. (2021) utilized marginal loss function in CNN training for better classification. Raja et al. (2019) developed crop signaling for improved detection, and Moazzam et al. (2022) used a CNN ensemble for high accuracy detection in sesame fields. Gao et al. (2020) explored DL-based object detectors for weed detection in sugar beet fields, while Picon et al. (2022) and Peng et al. (2022) investigated synthetic images and RetinaNet adaptations, respectively, for better crop-weed recognition. However, there remains a challenge with these DL models: they often produce confident predictions on the dataset from the source domain (original farm) but underperform on data from different domains (other farms) due to domain shift (Vu et al., 2019). This is further complicated by the high cost of acquiring labeled data for each new domain, especially for semantic segmentation where each pixel must be labeled (Tranheden et al., 2021).

Recent research has explored unsupervised domain adaptation (UDA) to improve the adaptability of crop-weed segmentation systems. Gogoll et al. (2020) devised a method utilizing cycle GANs to regenerate source data in the target domain style while maintaining semantic and structural object consistency. The result was a considerable enhancement in the generalization capabilities of fully convolutional networks (FCNs), resulting in around a 10% increase in the mIOU metric on two different source-target domain pairs.

Similarly, Kendler et al. (2022) tackled the issue of low-level variability in plant disease recognition training data. By dividing images into multiple patches, they increased dataset diversity and improved CNN generalizability without needing environmental modification. This resulted in a 20% improvement in classification accuracy over the baseline. For corn yield prediction across different regions, Ma et al. (2021) presented a CNN training strategy based on unsupervised adaptive domain adversarial training. Li et al. (2021) proposed an intermediate domain approach to decrease the domain gap in maize residue segmentation. However, the application of this approach may be limited as the intermediate domain is problem-specific.

Our approach is based on the idea that the classification of a plant as a crop or weed should not depend on the farm environment, soil type, the specific sensor (camera) used, or other low-level sources of variability. These sources of variability are uninformative for crop-weed recognition, but can significantly affect the predictions of CNNs.

In this paper our aim is to reduce the domain gap between the extracted features, from source and target domain, *via* adversarially optimized deep feature alignment and entropy minimization. Additionally, we introduce a novel regularization technique to improve the convergence of CNNs. In contrast to previous UDA works, we also explore the effectiveness of few-shot training strategy in the context of UDA, called few-shot supervised domain adaptation (SDA). Few-shot SDA involves fine-tuning the model on a small amount of labeled data from the target domain to improve its performance on that domain. The main advantage of few-shot SDA is that it can be used to quickly adapt a model to a new domain with minimal labeled data.

Our main contributions can be summarized as follows:

- A deep adversarial optimized framework for UDA and few-shot SDA.
- Augmentation scheduling strategy for improved regularization and convergence.
- A versatile dataset for fine-grained crop-weed recognition collected from five different fields with different setups.

2 Materials

2.1 Dataset construction

Our proposed approach was tested on a bean field dataset collected over the past one and a half years at five different locations and farms in South Korea using different image acquisition platforms. The dataset includes a number of variations in real-field conditions such as the field seeding bed (Gebrekidan, 2003), environment, weed density, plant scales, and sizes. To evaluate the performance of our approach, we selected five farms with different conditions and data variations as shown in Figure 1 and Table 1.

Beans are a crop that help improve soil health through nitrogen fixation, adding nitrogen back into the soil. Because of this ability, beans are often included in crop rotation plans, as nitrogen is an essential nutrient for growing strong and productive plants (Aschi et al., 2017). In countries like South Korea, where only 20% of the land is suitable for cultivation, it is especially important to use crops that can improve soil health. The collected dataset includes a bean crop and various types of weeds, but for the purposes of the crop-weed recognition task, we have grouped all the weeds into a single category. Table 1 summarized the characteristics of the dataset.

2.2 Field data distribution

In order to make our dataset suitable for domain adaptation, such as the representation shown in Figure 2, we considered the case of data collected at five different locations and fields, designated as F_A, F_B, F_C, F_D, and F_S, as shown in Figure 1. This is a specific example of domain adaptation across various scenarios, in which we aim to build a more robust system by transferring the visual characteristics from one field to another. In this case, we assume that the conditions of each field are different, meaning that each field may have a different weed density, seeding system, image acquisition system, and crop size. The visual characteristics of the fields used for data collection are displayed in Figures 1, 2 illustrates the visual attributes of various seeding bed systems across different fields.

2.3 Source and target datasets

In order to create the source and target datasets for unsupervised domain adaptation (UDA) in our experiments, we designated the field with the largest number of data available i.e., F_S as the source field, and all the other fields (F_A, F_B, F_C, F_D) as the target fields. Based on this grouping, we consider the following combinations across the five fields for evaluation: S→S, S→A, S→B, S→C, S→D. We train the network using data from the source domain (F_S) and test it against all the target domain datasets (F_A, F_B, F_C, F_D).

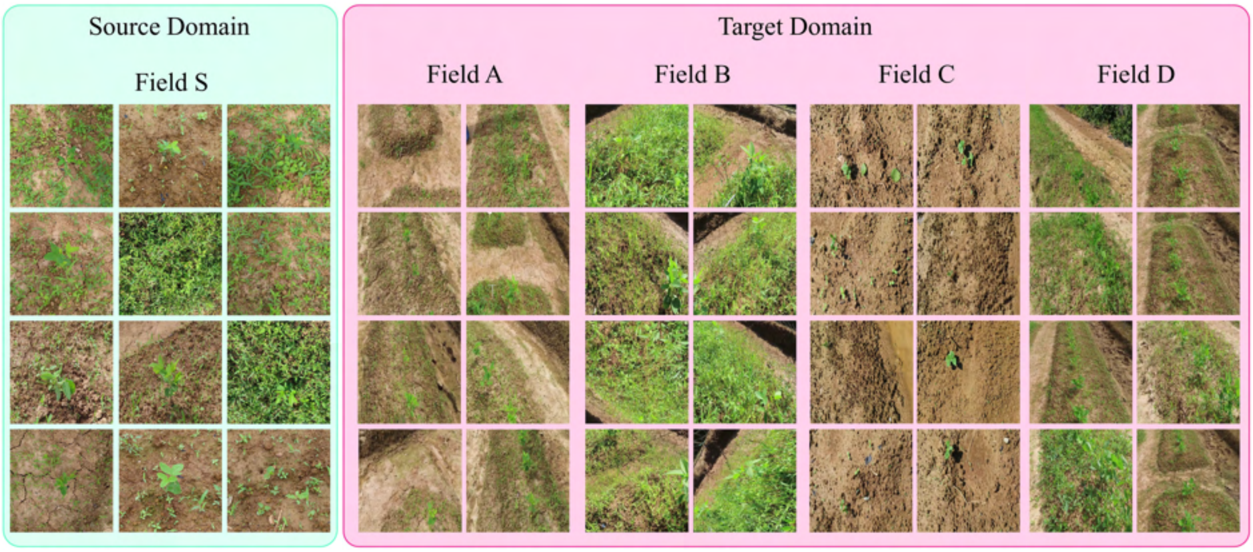


FIGURE 1
Representative images from different fields to collect data. The source domain (Field S) data is collected using handheld cameras in the form of images and are labelled by human annotators. Whereas the target domain data (Field A, B, C and D) is collected from various fields with a camera mounted on a moving platform in form of videos.

TABLE 1 Characteristics of Crop-Weed Recognition Datasets for unsupervised domain adaptation.

Field	No. of Images	Vegetation Density	Image Acquisition Platform	Seeding System
S	1893	Varying	Stationary	Flat Bed
A	202	Medium Dense	Moving	Ridge Bed
B	79	Dense	Moving	Ridge Bed
C	199	Sparce	Moving	Flat Bed
D	157	Medium Dense	Moving	Ridge Bed

3 Methodology

Here, we present our methodology for deep feature adaptation in context of UDA for crop-weed segmentation in unconstrained real-field environments. We also compare UDA approach with few-shot SDA for completeness. This section consists of the following sub-sections: (i) clearly defining the problem statement, (ii) introducing the architecture of the full framework, (iii) explaining the augmentation scheduling strategy which improves the performance of our framework, (iv) defining the learning objectives (loss functions), and (v) providing implementation details.

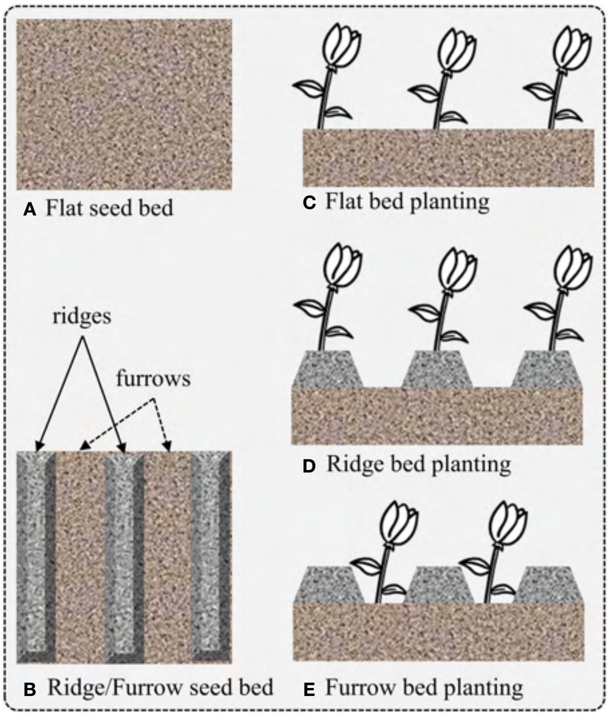


FIGURE 2
Representation of field seeding bed systems depending on the chosen planting method. Different planting methods can lead to varying crop yields for different crops. **(A)** Flat seed bed, **(B)** seeding bed with ridges and furrows, **(C)** plantation on flat seeding bed, **(D)** plantation on ridges and **(E)** plantation on furrows.

3.1 Problem definition

For better generalization we cast our problem as few-shot SDA because UDA can be simply defined as zero-shot SDA. Under these setting consider we are given a labelled soured dataset $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{N}$, where $D_s \subseteq \{F_s\}$ and N is the total number of images in D_S. Similarly, we have target domain datasets, $D_t = \{(x_i^t, y_i^t)\}_{i=1}^{M_t}$, from which we can only access *j* labelled images, here j ϵ {0,1,2,…, M_t} and M_t being the total number of images in *t-th* target domain dataset, and $D_s \subset \{F_A, \ F_B, F_C, F_D\}$. Here $x_i \in \mathbb{R}^{H\times W\times 3}$ is RGB-image and $y_i \in \mathbb{R}^{H\times W}$ is its corresponding label. We define *j*-shot SDA as randomly selecting *j* labelled images from each target domain datasets and using them for finetuning the network. The case of 0-shot SDA (j=0) is equivalent to unsupervised domain adaptation (UDA). For experiments we only consider $j = 0,1,3,M_t$. A graphical illustration that demonstrates the distinctions between UDA and few-shot SDA is displayed in Figure 3.

3.2 Augmentation scheduling

In conventional data augmentation strategies employed for training deep neural networks, a constant probability is applied for data augmentation, which often comprises a mixture of geometric and noise transformations. However, our proposed method diverges from this practice by progressively increasing the frequency of data augmentation as training advances, with each type of augmentation treated distinctly. The concept of increasing the augmentation probability finds parallels in the training of PA-GANs (Zhang and Khoreva, 2019), where both the generator and discriminator of a GAN grow progressively. Starting at low resolution, layers are incrementally added to enhance the resolution over time, thereby enabling the model to initially learn coarse-level structures, and then gradually learn fine-level details as training continues.

In contrast, the proposed technique involves adjusting the intensity or probability of data augmentation over time, but does not involve changing the architecture of the model itself over the course of training. In the augmentation scheduling of GANs, the emphasis is on enhancing the stability and efficacy of training through gradual growth of the model's structure. Conversely, augmentation scheduling focuses on presenting the model with an increasingly diverse and challenging array of training samples over time. While both techniques involve a form of progressive or scheduled change during training, they target different aspects of the training process. The augmentation scheduling technique is primarily about the model, while the augmentation scheduling technique is about the data.

Here we divide different augmentations into three categories depending on their characteristics:

- Geometric augmentations (G), augmentations which effect the entire image-label pair (x^s,y^s).
- Noise (distortion) augmentation (D), which only effect the original image (x^s) and labels (y^s) remain unchanged.

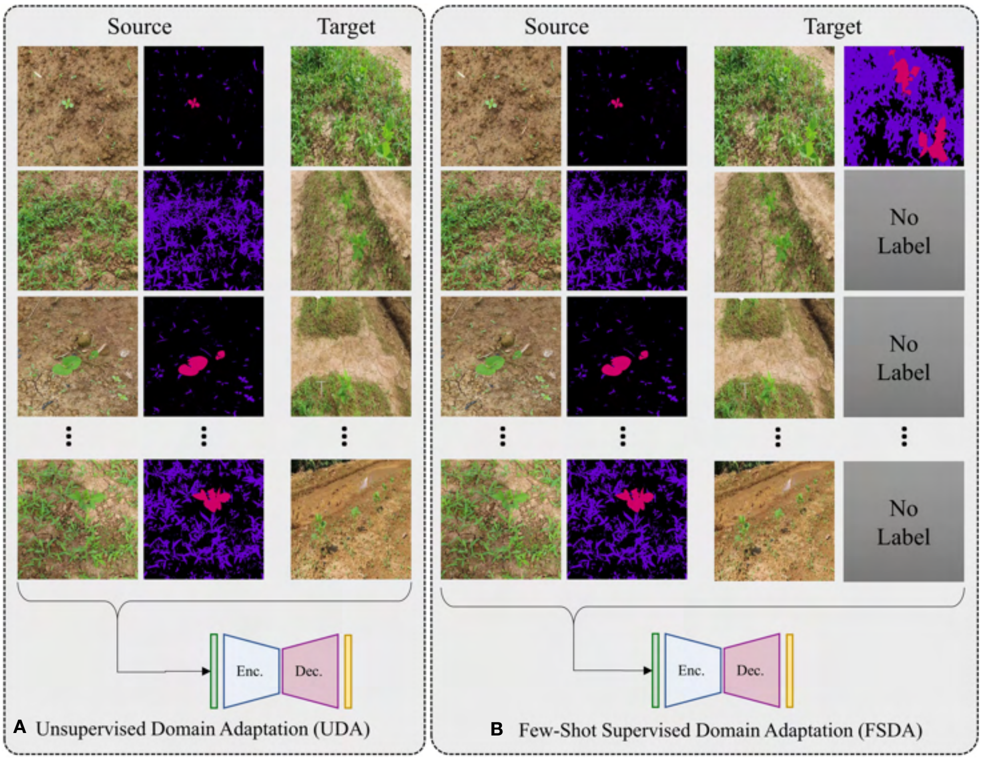

Source
Target
No Label
Enc.
Dec.
A Unsupervised Domain Adaptation (UDA)
B Few-Shot Supervised Domain Adaptation (FSDA)

FIGURE 3
A graphical representation illustrating differences between UDA and few-shot SDA. In UDA **(A)**, there a relatively large number of unlabeled target domain data is available for use during training. In few-shot SDA **(B)**, only a small number of labeled samples (typically one or two) are available for training. The figure shows an example of 1-shot SDA as only one labelled sample is provided.

- Collage Augmentation (C) (Chiang et al., 2019), which generate a collage of multiple image-label pairs (x_i^s, y_i^s) in the dataset. Mathematically it can be expressed as,

$$(x_i^c, y_i^c) = C(\{(x_i^s, y_i^s)\}_{i=1}^{M}, w_c, h_c, b_c) \quad (1)$$

where, C represents the function to generate a collage image-label pair (x_i^c, y_i^c) having width w_c and height h_c, of M images with b_c being the border width (in pixels) between images.

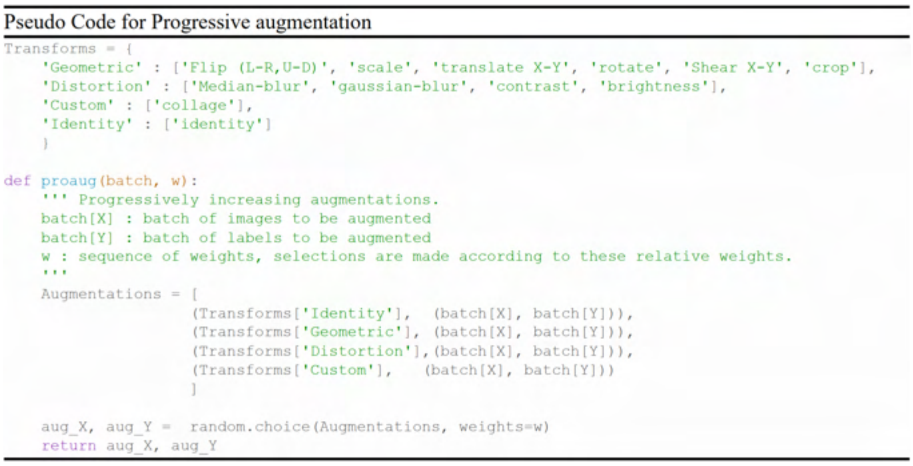
In the early training epochs, we only use the original images (identity augmentation, i.e., id = 1) so that the network can easily and quickly learn simple representations. We only augment the source domain images. Then, we gradually increase the probability of using the other augmentations, starting with geometric augmentations and eventually using all augmentations with specified probabilities (i.e., α, β, $\gamma > 0$). These stronger regularizations make learning more difficult for the CNN and improve its robustness. The probability weights for each type of augmentation can be considered as hyperparameters (i.e., α for G, β for D, and γ for C). The pseudo code for the augmentation scheduling process is shown in Figure 4, and Algorithm 1 Figure 5 summarizes the procedure for integrating augmentation scheduling into the proposed training loop of the framework. It is straightforward to adapt this to a standard training loop. Line graph in Figure 6 shows how the probability of each type of augmentation changes with training epochs for a specified set of hyperparameters. A few examples of data samples that have been augmented using the augmentation scheduling algorithm are presented in Figure 7.

3.3 Network architecture

The proposed framework for addressing the problem of domain shift between source and target domains is depicted in Figure 8. It consists of two subnetworks segmentation network and the discriminator network:

Segmentation Network - The segmentation network (φ), having learnable parameters w, consists of two main parts: an encoder and a decoder. The encoder is made up of a stem convolution block and four stages of feature extraction. The stem block consists of two 7x7 convolutions with a stride of 2. The subsequent four stages are composed of ConvNext blocks (Liu et al., 2022), with the number of channels in each block being $N_{ch}\in\{192, 384, 768, 1536\}$, in that order. Each block is repeated N_s times at each stage, with $N_s\in\{3, 3, 27, 3\}$.

The decoder also has four stages. The first stage uses an ASPP (Chen et al., 2018) module to extract multiscale features from the output of the encoder. The second stage is an upsampling module. In the third stage, the encoder's second stage features are concatenated with the output of the second stage of the decoder through a skip connection (Ronneberger et al., 2015; Badrinarayanan et al., 2017) and are then refined by a dense attention module (DAM) (Ilyas et al., 2021). To control the flow of useful information between the encoder and decoder, the encoder's feature maps are passed through a gating function (G), to reduce the number of feature maps and suppress low-level information, before being added *via* a skip connection. This can



Pseudo Code for Progressive augmentation

```
Transforms = {
    'Geometric' : ['Flip (L-R,U-D)', 'scale', 'translate X-Y', 'rotate', 'Shear X-Y', 'crop'],
    'Distortion' : ['Median-blur', 'gaussian-blur', 'contrast', 'brightness'],
    'Custom' : ['collage'],
    'Identity' : ['identity']
    }

def proaug(batch, w):
    ''' Progressively increasing augmentations.
    batch[X] : batch of images to be augmented
    batch[Y] : batch of labels to be augmented
    w : sequence of weights, selections are made according to these relative weights.
    '''
    Augmentations = [
                    (Transforms['Identity'],  (batch[X], batch[Y])),
                    (Transforms['Geometric'], (batch[X], batch[Y])),
                    (Transforms['Distortion'],(batch[X], batch[Y])),
                    (Transforms['Custom'],   (batch[X], batch[Y]))
                    ]

    aug_X, aug_Y =  random.choice(Augmentations, weights=w)
    return aug_X, aug_Y
```

FIGURE 4
Pseudo code of Augmentation scheduling algorithm.

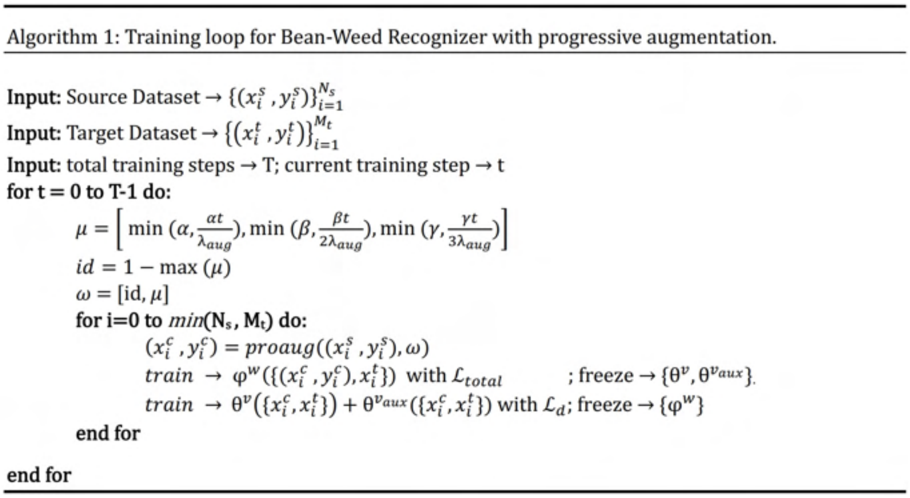
be represented mathematically as, $G(x) = f_r^{1\times1}(x)$, where f is a 1x1 convolutional filter with r channels.

Discriminator Network - PatchGAN (Isola et al., 2017) is used as a fully convolutional discriminator (θ) to classify whether incoming image features are form source domain or target domain. By evaluating smaller patches of the output features rather than the full feature map as a whole allows the PatchGAN to capture fine-grained details in the original image and make more informed decisions. Our framework uses two discriminators for deep feature alignment between the source and target domain features, with one aligning the decoder features (θ^v, having learnable parameters v) and the other aligning the encoder features ($\theta^{v_{aux}}$, having learnable parameters v_{aux}). It was found to be more effective than using only one discriminator at the end of the decoder. Both discriminators (θ) consist of five layers having filter size of 4x4 and a stride of 2, with the number of channels in each layer being {64,128,256,512,1}. Each convolutional layer is followed by instance normalization and a LeakyReLU activation with a negative slope of 0.2.

3.4 Learning objective

Given the augmented source domain labelled pair (x_i^c, y_i^c) the segmentation network (φ^w) predicts a K-dimensional soft segmentation map $p_i = \varphi^w(x_i^c)$, where $p_i \in \mathbb{R}^{H\times W\times K}$ and K is the number of classes present in the dataset. Here each K-dimensional (pixel-wise) vector is a probability distribution over classes. The segmentation network is trained by minimizing the following cross-entropy loss between the ground truth (y_i^c) and the predicted probability map (p_i), given by equation 2.

$$\mathcal{L}_{seg}(x_i^c, y_i^c) = -\frac{1}{N}\sum_{i=0}^{N}\langle\, y_i^c \,.\, \log(\varphi^w(x_i^c))\,\rangle \tag{2}$$



Algorithm 1: Training loop for Bean-Weed Recognizer with progressive augmentation.

```
Input: Source Dataset → $\{(x_i^s, y_i^s)\}_{i=1}^{N_s}$
Input: Target Dataset → $\{(x_i^t, y_i^t)\}_{i=1}^{M_t}$
Input: total training steps → T; current training step → t
for t = 0 to T-1 do:
    $\mu = \left[\min\left(\alpha, \frac{\alpha t}{\lambda_{aug}}\right), \min\left(\beta, \frac{\beta t}{2\lambda_{aug}}\right), \min\left(\gamma, \frac{\gamma t}{3\lambda_{aug}}\right)\right]$
    $id = 1 - \max(\mu)$
    $\omega = [\mathrm{id}, \mu]$
    for i=0 to $min$(N$_s$, M$_t$) do:
        $(x_i^c, y_i^c) = proaug((x_i^s, y_i^s), \omega)$
        $train \rightarrow \varphi^w(\{(x_i^c, y_i^c), x_i^t\})$ with $\mathcal{L}_{total}$ ; freeze → $\{\theta^v, \theta^{v_{aux}}\}$.
        $train \rightarrow \theta^v(\{x_i^c, x_i^t\}) + \theta^{v_{aux}}(\{x_i^c, x_i^t\})$ with $\mathcal{L}_d$; freeze → $\{\varphi^W\}$
    end for
end for
```

FIGURE 5
Training algorithm for proposed framework.

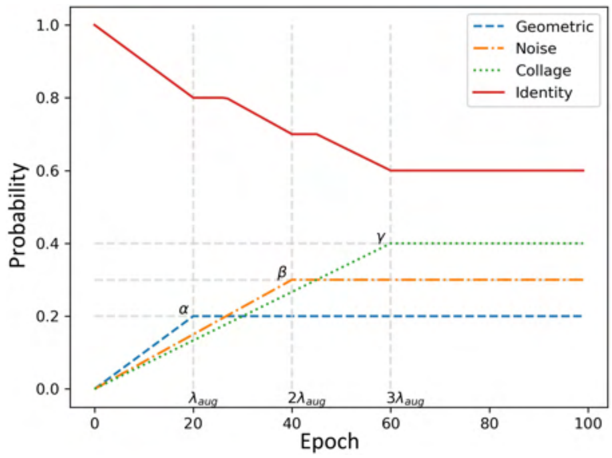


FIGURE 6
Line graph representing the changes in probabilities for each type of augmentation with training epochs for a specified set of hyperparameters, i.e., α=0.2, β=0.3, γ=0.4 and λ_{aug}=20.

For target domain samples (x_i^t) as annotation (y_i^t) are not available, hence these samples can't be used to learn the parameters (*w*) in same way as source domain samples can be used. So, following [28] here we use entropy minimization approach to maximize prediction certainty (lowering surprise) on target domain samples. Given a target domain input (x_i^t) we generate and entropy map (e_i), where $e_i \in \mathbb{R}^{H\times W}$ shows independent pixel-wise entropies of summation of network's predictions p_i (on target domain), normalized between [0,1] range. An example of entropy map is shown in Figure 8 and mathematically expressed by equation 3.

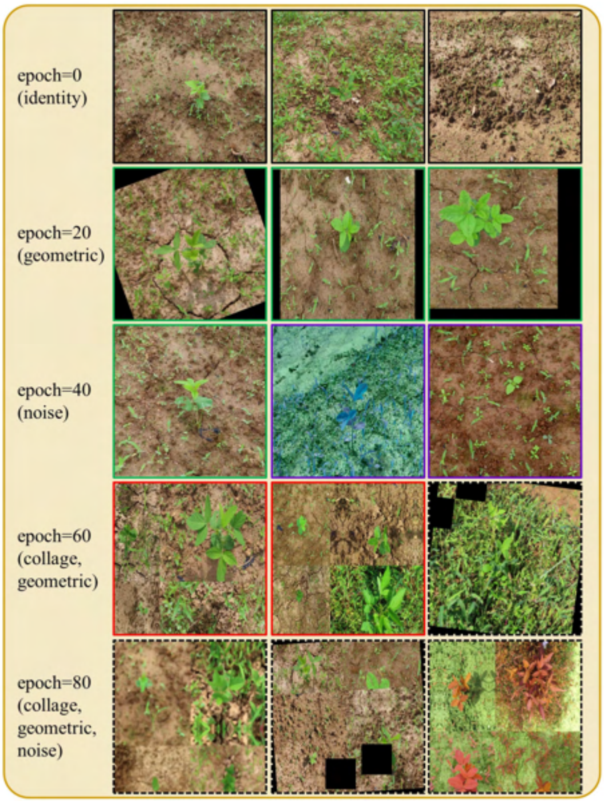


FIGURE 7
Augmentation scheduling in action: images are augmented using various combinations of augmentations at different training epochs. Black boxes depict identity augmentation, green boxes depict geometric (G) augmentation, purple boxes depict noise (D) augmentation, red boxes depict the collage (C) augmentation, and the dashed black boxes represent the application of all augmentations simultaneously.

$$e_i = -\frac{1}{\log(K)}\sum_{k=0}^{K}\varphi^{w}(x_i^t)\;.\log(\varphi^{w}(x_i^t)) \tag{3}$$

However, minimizing entropy directly is ineffective in low entropy regions (Yang and Soatto, 2020). So, we utilize robust entropy minimization, modified *via* carbonnier penalty function which penalizes high entropy predictions more than low entropy predictions when η > 0.5. Utilizing this modified entropy loss (L_{ent}) we update the network's parameters by equation 4.

$$\mathcal{L}_{ent}(x_i^t) = \left(\frac{1}{N}\sum_{i=0}^{N}e_i^2 + 0.0001^2\right)\eta \tag{4}$$

Given the class probability distributions generated from the features output by third stage of encoder and final stage of decoder, represented as $p_{i_{aux}}$ and p_i respectively. These distributions are then passed on to their corresponding discriminators, denoted as $\theta^{v_{aux}}$ and θ^{v} respectively. The goal of these discriminators is to produce domain classification outputs, with a value of 1 indicating the source domain and 0 indicating the target domain. Both discriminators are trained using the cross-entropy loss (L_{ce}). The overall objective for the final discriminator can be expressed as equation 5.

$$\mathcal{L}_d = \mathcal{L}_{ce}(\theta^{v}(x_i^c),\ 1) + \mathcal{L}_{ce}(\theta^{v}(x_i^t),\ 0) \tag{5}$$

Similarly, an equation can be written for the auxiliary discriminator ($\mathcal{L}_{d_{aux}}$), resulting in the total discriminator loss.

$$\mathcal{L}_D = \mathcal{L}_d + \lambda_{aux}\mathcal{L}_{d_{aux}} \tag{6}$$

Now, the adversarial objective for training segmentation network can be written as,

$$\mathcal{L}_{adv} = \mathcal{L}_{ce}(\theta^{v}(x_i^t),\ 1) \tag{7}$$

Both the segmentation and discriminator networks are jointly trained in each iteration. During training, the supervised segmentation loss for source domain samples and unsupervised entropy loss for target domain samples are jointly optimized. The adversarial loss trains the segmentation network to deceive the discriminator by maximizing the probability of target predictions being considered as source predictions. This is achieved by minimizing the cross-entropy loss between the discriminator's predictions for target images and the label of the source domain, which is 1. Therefore, the total loss becomes,

$$\mathcal{L}_{total} = \mathcal{L}_{seg} + \lambda_{ent}\mathcal{L}_{ent} + \lambda_{adv}\mathcal{L}_{adv} \tag{8}$$

In the few-shot SDA scenario, where we have *j* labelled images from the target domain, which are used to fine-tune our model. In addition to the entropy minimization loss described in equation 4,

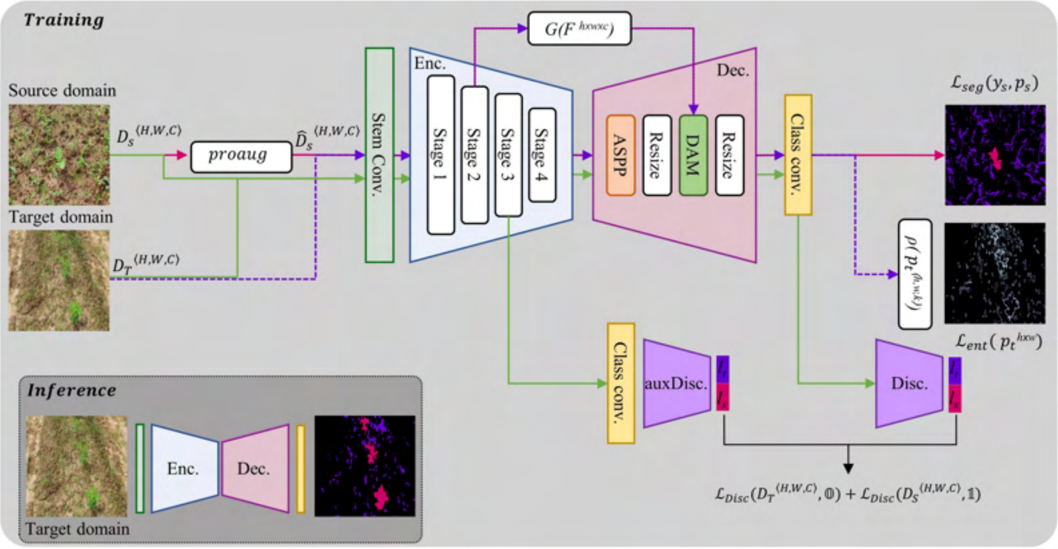


FIGURE 8

Architecture of proposed framework for UDA in crop-weed segmentation. D_S and $\hat{D}_s$ represent the source domain and augmented source domain datasets respectively. During encoder-decoder training, the pink arrows depict the flow of forward and backward gradients for input from the source domain, while the purple arrows represent input from the target domain. The discriminators are kept frozen during this training step. The green arrows show the flow while the discriminators are being trained. At this stage, the encoder-decoder network is kept frozen.

we also incorporate a cross-entropy loss similar to equation 2 for these *j* examples. Let's denote these labeled examples from the target domain as $D_t = \{(x_i^t, y_i^t)\}_{i=1}^{M_t}$ where *i* ranges from 1 to *j*. The additional cross-entropy loss for these samples can be expressed as:

$$\hat{\mathcal{L}}_{seg}(x_i^t, y_i^t) = -\frac{1}{j}\sum_{i=0}^{j}\langle\, y_i^t \,.\, \log(\varphi^w(x_i^t))\,\rangle \qquad (9)$$

Therefore, in the case of *j*-shot SDA, the total loss would be updated to:

$$\mathcal{L}_{total} = \mathcal{L}_{seg} + \lambda_{seg}\hat{\mathcal{L}}_{seg} + \lambda_{ent}\mathcal{L}_{ent} + \lambda_{adv}\mathcal{L}_{adv} \qquad (10)$$

where $\hat{\mathcal{L}}_{seg}$ corresponds to the supervised segmentation loss for the *j* labeled target domain samples, and λ_{seg} is a weight hyperparameter to balance this new term. The model is then jointly optimized for the supervised segmentation loss on both source domain and *j* labeled target domain samples, unsupervised entropy loss for the remaining unlabeled target domain samples, and adversarial loss.

In this way, we effectively use the limited labeled data available in the target domain to guide the model's adaptation process, while still leveraging the entropy minimization approach for the unlabeled target domain data.

3.5 Implementation details

In our implementation we used the PyTorch toolbox and a single NVIDIA RTX-3090 GPU, which has 24GB of memory. The source dataset, which contains a large number of images, was split into a 80% train-validation set and a 20% test set. The target datasets were split into a 70% training set (used only in the case of supervised training for comparison) and a 30% test set.

For training the segmentation network, we employed the SGD optimizer with a weight decay of 5x10^{-4}. For training the discriminators, we used the Adam optimizer with a momentum value of 0.9 and 0.99. We used a cosine decay policy for the segmentation network, with a learning rate of 0.001 and warm start for the first 1000 iterations. For the discriminators, we used a polynomial decay policy with an initial learning rate of 10^{-4}. A detailed list of the hyperparameter settings for the augmentation scheduling and loss function weights can be found in Table 2.

4 Results and discussion

The performance of the proposed method for crop-weed recognition in bean fields was evaluated using the same field data distribution and source and target data splits described in Section 2. To thoroughly evaluate the proposed method, we employed widely used semantic segmentation frameworks, including DeepLab-v3+ and PSPNet (Zhao et al., 2017), with ResNet-101 (He et al., 2016), Xception-71 (Chollet, 2017), and ConvNext-L backbones as baselines. The results of our proposed method were compared with these baselines under the same operating conditions.

TABLE 2 Hyperparameter settings for proposed framework.

Hyperparameter	Value	Hyperparameter	Value
α	0.3	η	2.0
β	0.3	λ_{aux}	0.4
γ	0.3	λ_{ent}	0.5
λ_{aug}	20	λ_{adv}	0.4
λ_{seg}	2		

Firstly, we compared the performance of the proposed framework with traditional segmentation models and other recent unsupervised domain adaptation (UDA) methods. The results indicated that our proposed method performed competitively with these models. Furthermore, we demonstrated how the use of augmentation scheduling further improved the performance of our network. We also conducted ablation experiments to highlight the improvement in results achieved by using augmentation scheduling in comparison to vanilla augmentation.

Lastly, we compared the results of our proposed UDA method under both few-shot self-supervised domain adaptation (SDA) and fully supervised settings. The results showed that our proposed method performed well under both settings and yielded promising results. We evaluate the effectiveness of the proposed framework as well as compare it with other networks utilizing the Intersection-over-Union (IoU) metric, defined by equation 11.

$$mIoU = \frac{1}{N}\sum_{i=0}^{N}\left|\frac{y_i \cap \quad p_i}{y_i \cup \quad p_i}\right| \tag{11}$$

where y_i and p_i represent the ground-truth and predicted segments, respectively.

4.1 Source training only

In the first experiment, we trained semantic segmentation architectures in a simple supervised fashion on the source field (F_S) dataset and compared their performance. In this experiment, all models were trained on the source field dataset and results are reported on its test split (S→S), as shown in Table 3. PSPNet showed the worst performance among all other models when using the same backbone (ResNet-101), while DeepLab-v3+ with Xception-71 backbone performed better than PSPNet. Additionally, integrating the proposed modified decoder into the best-performing model (DeepLab-v3+ with ResNet-101) further boosted performance. It is worth noting that no data augmentation was used in these experiments.

Under the source training only (STO) setting, we also tested the segmentation performance of only source-trained models on other target domain fields (i.e., F_A, F_B, F_C, F_D). The results are reported in Table 3 under columns S→T, where T∈{A, B, C, D}. It can be seen from the table that, even though using better segmentation architectures resulted in considerably better performance on the F_S dataset, the results on the target domain fields did not improve and even got worse in some cases (e.g., the mIOU of field *A* and *C* decreased when using DeepLab-v3+ (ResNet-101) and proposed decoder). This demonstrates the need for unsupervised domain adaptation (UDA) approaches in the field of precision agriculture.

4.2 Unsupervised domain adaptation

In our unsupervised domain adaptation experiments, we used the same data pairs as in previous experiments. We applied the augmentation scheduling algorithm with the hyperparameter values listed in Table 2. The results of these experiments are shown in Tables 3, 4, with and without augmentation scheduling. Overall, we observed a significant improvement in the mIOU score for bean-weed recognition compared to STO methods (as seen in Table 3's top four rows). Our proposed deep feature alignment method without augmentation scheduling performed better on average than previous STO and UDA methods. As shown in Figure 9, using proposed deep feature alignment method resulted in a noticeable improvement in performance compared to using only STO. Additionally, incorporating augmentation scheduling further increased the performance of all models. Specifically, our proposed segmentation model that uses both deep feature alignment and augmentation scheduling outperformed previous best STO models by 8% and previous best UDA methods by 7%. The performance gap was even greater on target fields FA and FD, with improvements of 5.42% and 8.1% respectively.

4.3 Few-shot supervised domin adaptation

In this section, we compared our approach with other conventional few-shot SDA and fully supervised methods. The results are summarized in Table 4. All experiments were conducted under the same conditions. For the fully supervised training, all models were trained using training splits of both the target and source dataset as described in subsection 3.5 (Implementation Details). Under these conditions, our proposed segmentation network showed an improvement of 3% in the mIOU score compared to the DeepLabv3+ model, indicating its superior feature extraction ability. For the few-shot SDA experiments, the model's parameters were fine-tuned using a small number of labeled samples from the target domain. As shown in Table 5, using only one labeled sample (1-shot), our model achieved an accuracy that was almost similar to that of the fully supervised model (80.53% vs 83.6%). Additionally, our proposed method consistently outperformed other SDA methods throughout the few-shot experiments. As seen in Table 5, our method exceeded the best-performing few-shot SDA methods by 2.5% (0-shot), 3.0% (1-shot), and 2.2% (3-shot) for bean-weed recognition. Figure 10 compares the visualization results, demonstrating that our method showed significant improvements in recognizing crops and weeds.

4.4 Vanilla vs. scheduled augmentation

In these experiments, we verify the superior performance of the proposed augmentation scheduling over vanilla augmentation, and the results are summarized in Tables 4, 6. For these experiments we use proposed framework under UDA (0-shot SDA) settings. We experimented with different augmentation probabilities and found that augmenting 30% of all samples during each epoch produced the best results. Starting from the baseline (no augmentations), we first performed random geometric augmentations (G) and observed performance improvement. Then, we performed noise (D) and collage (C) augmentations one by one to see further improvements. A significant increase in performance, 55.36% (baseline) to 71.28%, can be seen when using collage augmentation (C), indicating that

TABLE 3 A comparison of the experimental results on a crop-weed segmentation dataset between traditional semantic segmentation and UDA methods with the use of Vanilla Augmentation.

Backbone	Framework	Method	mIOU (%)					
			S→S	S→A	S→B	S→C	S→D	Average
ResNet-101	PSPNet	–	74.05	66.34	56.83	38.62	52.88	53.66
Xception-71	DeepLab V3+	–	78.7	67.63	59.19	43.05	47.68	56.13
ResNet-101	DeepLab V3+	–	79.57	68.9	59.34	40.43	54.34	57.75
ResNet-101	Proposed	–	80.02	68.53	58.88	41.35	53.7	57.36
ResNet-101	DeepLab V3+	Vu et al., (2019)	**80.95**	63.66	55.62	52.31	47.62	54.80
ResNet-101	DeepLab V3+	Tsai et al. (2018)	75.94	75.57	65.23	45.65	**66.7**	63.28
ResNet-101	DeepLab V3+	Proposed	79.55	74.1	62.6	49.56	64.48	62.18
ConvNext-L	Proposed	Proposed	79.93	**75.67**	**68.54**	**52.36**	63.34	**64.97**

Bold and underlined values represent best and second-best values, respectively.

TABLE 4 A comparison of the experimental results on a crop-weed segmentation dataset between traditional semantic segmentation and UDA methods with the use of Augmentation scheduling.

Backbone	Framework	Method	mIOU (%)					
			S→S	S→A	S→B	S→C	S→D	Average
ResNet-101	PSPNet	–	78.31	77.07	67.36	47.39	62.33	63.53
Xception-71	DeepLab V3+	–	82.5	78.16	62.05	48.49	63.46	63.04
ResNet-101	DeepLab V3+	–	84.81	75.75	69.85	56.88	60.69	65.79
ResNet-101	DeepLab V3+	Vu et al. (2019)	83.99	80.6	73.4	60.86	71.7	71.39
ResNet-101	DeepLab V3+	Tsai et al. (2018)	79.8	78.75	70.8	50.67	79.98	70.05
ResNet-101	DeepLab V3+	Proposed	83.97	84.2	73.52	**61.71**	74.7	73.53
ConvNext-L	Proposed	Proposed	**88.5**	**91.02**	**83.11**	60.3	**81.1**	**78.88**

Bold and underlined values represent best and second-best values, respectively.

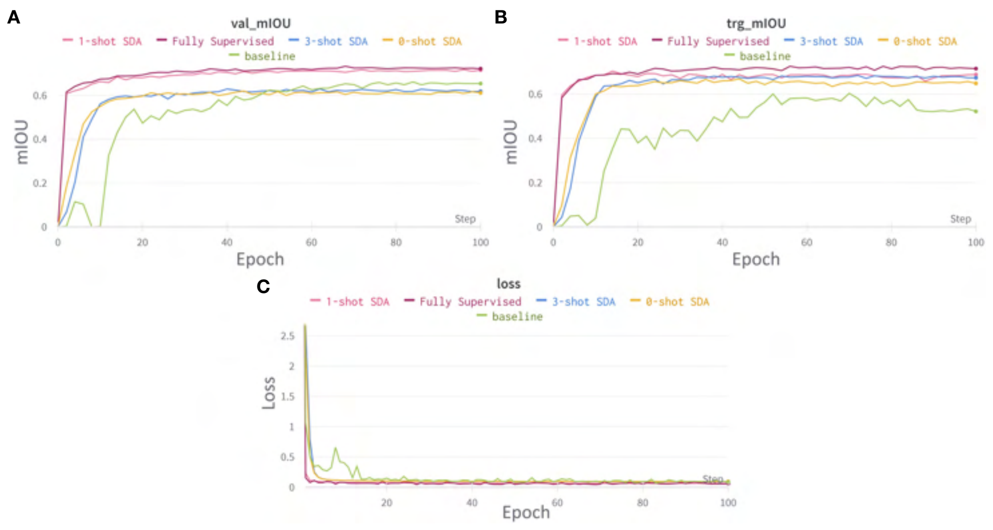


FIGURE 9
Training and loss curves for cross domain adaptation. **(A)** learning curve for mIOU score on source domains validation set. **(B)** learning curve for mIOU score averaged over all target domain test sets. **(C)** Segmentation network's loss curves. Best viewed in color.

TABLE 5 Comparison of mIOU scores for few-shot SDA models with varying values of *k* against fully supervised models.

Backbone	Framework	Method	Strategy	mIOU (%)					
				S→S	S→A	S→B	S→C	S→D	Average
ResNet-101	DeepLab V3+	Vu et al. (2019)	0-shot	83.99	80.6	73.4	60.86	71.7	71.39
ResNet-101	DeepLab V3+	Tsai et al. (2018)	(UDA)	79.8	78.75	70.8	50.67	79.98	70.05
ConvNext-L	Proposed	Proposed		88.5	91.02	83.11	60.3	81.1	78.88
ResNet-101	DeepLab V3+	Vu et al. (2019)	1-shot	86.13	88.71	83.59	58.89	75.05	76.56
ResNet-101	DeepLab V3+	Tsai et al. (2018)	(SDA)	76.49	85.01	73.02	61.59	74.9	73.63
ConvNext-L	Proposed	Proposed		88.7	92.53	84.25	62.5	80.87	80.53
ResNet-101	DeepLab V3+	Vu et al. (2019)	3-shot	88.57	89.66	84.17	60.69	78.62	78.28
ResNet-101	DeepLab V3+	Tsai et al. (2018)	(SDA)	77.36	86.07	75.49	62.99	76.31	75.19
ConvNext-L	Proposed	Proposed		89.01	92.67	85.6	62.9	82.43	81.4
ResNet-101	DeepLab V3+	–	Fully	90.56	93.25	83.5	61.56	83.42	80.43
ConvNext-L	Proposed	–	supervised	**91.62**	**95.24**	**85.48**	**68.67**	**85.39**	**83.69**

Bold and best results. Underlined values represent best results in their respective settings.

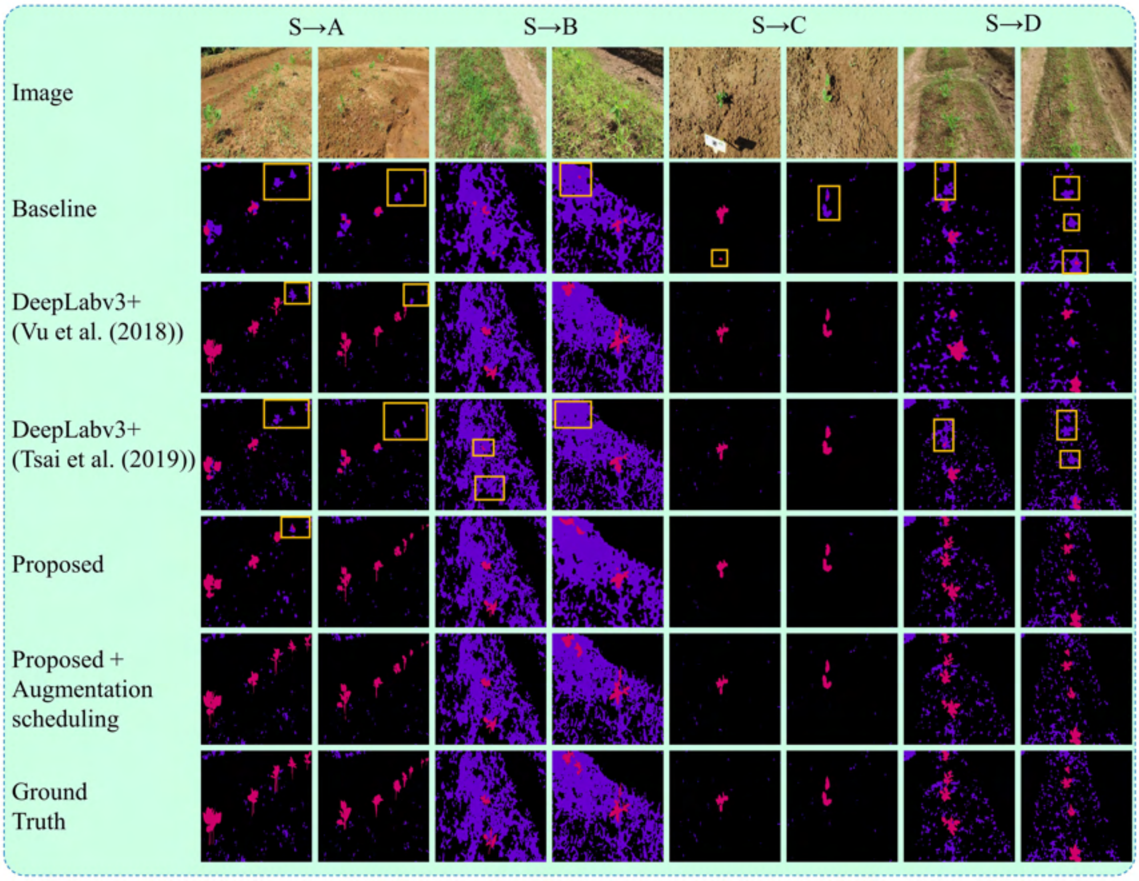


FIGURE 10
Segmentation results on datasets from the target domain under UDA setting. The results include the baseline method, DeepLabv3+ with the UDA algorithms from Vu et al., (2019) and Tsai et al. (2018), and the proposed method with and without augmentation scheduling. The ground truths are also displayed for comparison. Each target field **(A–D)** has two columns, with each column representing a different testing field with varying farm environments such as weed density, seeding bed types, plant sizes, and camera viewing angles. Boxes highlight the crops being misclassified as weeds.

TABLE 6 Effect of augmentation scheduling on performance of proposed UDA framework.

Strategy	Augmentation	mIOU (%)					
		S→S	S→A	S→B	S→C	S→D	Average
Baseline	–	73.9	68.36	56.86	42.33	53.91	55.36
Vanilla (0.3)	Geometric (G)	74.37	82.3	70.59	55.98	72.67	70.39
	Noise/Dist. (D)	77.59	78.49	74.63	55.83	73.42	70.59
	Collage (C)	80.35	84.7	71.33	53.91	75.19	71.28
	G+D	81.03	74.9	66.14	53.55	65.67	65.06
	G+D+C	79.93	75.67	68.54	52.36	63.34	64.97
Progressive (0.3)	G+D	82.59	83.57	75.85	57.75	78.36	73.88
	G+D+C	**88.5**	**91.02**	**83.11**	**60.3**	**81.1**	**78.88**

Bold and underlined values represent best and second-best values, respectively.

the collage augmentation improves the network's generalization on other domains as well. Next, we combined these augmentations at a constant probability (0.3) throughout the training process. It can be seen from Table 2 that performing all augmentations in combination considerably improved the framework's performance compared to the baseline.

However, when using all augmentations at once throughout the training process (i.e., G+D+C), the network's performance drops as compared to when only using G+D. We believe this is because the augmentations are quite strong from the start of training, making it difficult for the network to learn important distinguishing features. To overcome this, we deployed the proposed augmentation scheduling strategy, which fully activates each augmentation after a certain number of epochs (set by the user as a hyperparameter), so that the network can easily and quickly learn simple representations at the start of training. At the end of training, when all augmentations are fully activated, these stronger regularizations make learning more difficult for the CNN and improve its robustness.

As can be seen in Table 6, even without using collage augmentation, the augmentation scheduling algorithm improves the average mIOU by almost 9% compared to vanilla G+D+C. When using all three types of augmentation with progressive strategy, the results improvement is almost 14% as compared to the vanilla augmentation strategy and about a 22% increase when using no augmentation at all.

4.5 Training and loss curves across domains

In Figure 9, the graphs illustrate the training loss (source domain only) and accuracy curves for the proposed domain adaptation for the source domain and average of all the target domains. The system successfully adapted from one domain to the other and was able to effectively recognize both crops and weeds across various seeding bed systems.

5 Visual analysis

The qualitative results of the proposed method are illustrated in Figure 10. The figures present some examples of the system's qualitative performance on the testing dataset from the target and source domains. The system is capable of identifying crops and weeds effectively across different fields, even with varying densities of weeds and different seeding systems. Our approach is robust in addressing the recognition of crops (beans) and weeds, even in complex target (unseen) field environments used for domain adaptation. The underlying reason behind this performance is the utilization of deep feature alignment and augmentation scheduling algorithm which allows the system to incorporate more robust features and context information, leading to more stable and reliable segmentation results.

6 Conclusion

In this research, we presented an approach for unsupervised domain adaptation for crop-weed recognition in an unseen field environment. The main challenge in creating an automatic weed management system is the varying visual appearance of weeds based on factors such as lighting, weather, soil, and seeding bed type. We proposed to address this problem by minimizing the entropy of the network on target domain dataset and aligning the features of both domains through deep feature alignment. Our proposed framework, which is trained in an end-to-end fashion, consists of two main components: a segmentation network for feature extraction and robust entropy minimization and a discriminator network for adversarial training to generate target domain features as close as possible to the source domain. Additionally, we proposed the use of a augmentation scheduling strategy that starts with weak augmentations for quick adaptation to the source domain dataset and gradually increases to stronger augmentations for improved robustness and generalizability. We also demonstrated that the use of collage augmentation improves performance on target domains even further. Our extensive evaluation across four different fields with various environments and plant seeding systems showed an overall performance gain of approximately 10% mIOU on average compared to the baseline. Furthermore, using just one image for fine-tuning in a few-shot SDA setting, our network achieved almost similar performance to that of a fully supervised network, i.e., 80.53% vs 83.6%. A potential direction for future research would be to explore the adaptation of the model for recognition of multiple crops and weeds.

Data availability statement

The original contributions presented in the study are publicly available. This data can be found here: https://github.com/Mr-TalhaIlyas/ARUFE.

Author contributions

TI and HK did conceptualization, investigation, methodology, validation, formal analysis. TI wrote the original draft and then revised. TI and HK supervised, reviewed and edited manuscript. HK, YJ and OW secured funding. TI and JL collected and labelled data, performed investigation and visualization. All authors discussed the results and contributed to the final manuscript. All authors contributed to the article and approved the submitted version.

Funding

This work was supported in part by the Agricultural Science and Technology Development Cooperation Research Program (PJ015720) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1A6A1A09031717 and NRF-2019R1A2C1011297).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Adhikari, S. P., Yang, H., and Kim, H. (2019). Learning semantic graphics using convolutional encoder–decoder network for autonomous weeding in paddy. *Front. Plant Sci.* 10, 1404. doi: 10.3389/fpls.2019.01404

Aschi, A., Aubert, M., Riah-Anglet, W., Nélieu, S., Dubois, C., Akpa-Vinceslas, M., et al. (2017). Introduction of Faba bean in crop rotation: Impacts on soil chemical and biological characteristics. *Appl. Soil Ecol.* 120, 219–228. doi: 10.1016/j.apsoil.2017.08.003

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 2481–2495. doi: 10.1109/TPAMI.2016.2644615

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV))*, Munich, Germany. 801–818.

Chiang, H.-A., Chiang, C.-S., and Shih, C.-S. (2019). Using collaged data augmentation to train deep neural net with few data. In *Proceedings of Medical Imaging with Deep Learning*. London, England. OpenReview.net. doi: 10.1145/3400286.3418244

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251–1258).

Coleman, G., Salter, W., and Walsh, M. (2022). OpenWeedLocator (OWL): an open-source, low-cost device for fallow weed detection. *Sci. Rep.* 12, 1–12. doi: 10.1038/s41598-021-03858-9

Durmuş, H., Güneş, E. O., Kırcı, M., and Üstündağ, B. B. (2015). "The design of general purpose autonomous agricultural mobile-robot:" AGROBO," in *2015 Fourth International Conference on Agro-Geoinformatics (Agro-geoinformatics)*. Istanbul, Turkey. 49–53. IEEE.

Gao, J., French, A. P., Pound, M. P., He, Y., Pridmore, T. P., and Pieters, J. G. (2020). Deep convolutional neural networks for image-based Convolvulus sepium detection in sugar beet fields. *Plant Methods* 16, 1–12. doi: 10.1186/s13007-020-00570-z

Gebrekidan, H. (2003). Grain yield response of sorghum (Sorghum bicolor) to tied ridges and planting methods on Entisols and Vertisols of Alemaya area, Eastern Ethiopian highlands. *J. Agric. Rural Dev. Tropics Subtropics (JARTS)* 104, 113–128.

Gogoll, D., Lottes, P., Weyler, J., Petrinic, N., and Stachniss, C. (2020). "Unsupervised domain adaptation for transferring plant classification systems to new field environments, crops, and robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA. 2636–2642. IEEE. doi: 10.1109/IROS45743.2020.9341277

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Ilyas, T., Jin, H., Siddique, M. I., Lee, S. J., Kim, H., and Chua, L. (2022). DIANA: A deep learning-based paprika plant disease and pest phenotyping system with disease severity analysis. *Front. Plant Sci.* 3862. doi: 10.3389/fpls.2022.983625

Ilyas, T., Umraiz, M., Khan, A., and Kim, H. (2021). DAM: Hierarchical adaptive feature selection using convolution encoder decoder network for strawberry segmentation. *Front. Plant Sci.* 12, 591333. doi: 10.3389/fpls.2021.591333

Iqbal, N., Manalil, S., Chauhan, B. S., and Adkins, S. W. (2019). Investigation of alternate herbicides for effective weed management in glyphosate-tolerant cotton. *Arch. Agron. Soil Sci.* 65 (13), 1885–1899. doi: 10.1080/03650340.2019.1579904

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. Honolulu, HI, USA. 1125–1134. IEEE. doi: 10.1109/CVPR.2017.632

Kendler, S., Aharoni, R., Young, S., Sela, H., Kis-Papo, T., Fahima, T., et al. (2022). Detection of crop diseases using enhanced variability imagery data and convolutional neural networks. *Comput. Electron. Agric.* 193, 106732. doi: 10.1016/j.compag.2022.106732

Khan, A., Ilyas, T., Umraiz, M., Mannan, Z. I., and Kim, H. (2020). Ced-net: crops and weeds segmentation for smart farming using a small cascaded encoder-decoder architecture. *Electronics* 9, 1602. doi: 10.3390/electronics9101602

Kriticos, D. J., Alexander, N. S., and Kolomeitz, S. M. (2006). Predicting the potential geographic distribution of weeds in 2080. *Weed Sci. Soc. Victoria Melbourne Aust.*, 27–34.

Lal, R. (1991). Soil structure and sustainability. *J. Sustain. Agric.* 1, 67–92. doi: 10.1300/J064v01n04_06

Lameski, P., Zdravevski, E., and Kulakov, A. (2018). Review of Automated Weed Control Approaches: An Environmental Impact Perspective. In: S. Kalajdziski and N. Ackovska (eds) ICT Innovations 2018. Engineering and Life Sciences. ICT 2018. Communications in Computer and Information Science. (Cham: Springer), vol 940. doi: 10.1007/978-3-030-00825-3_12

Li, L., Li, J., Lv, C., Yuan, Y., and Zhao, B. (2021). Maize residue segmentation using Siamese domain transfer network. *Comput. Electron. Agric.* 187, 106261. doi: 10.1016/j.compag.2021.106261

Li, J., and Tang, L. (2018). Crop recognition under weedy conditions based on 3D imaging for robotic weed control. *J. Field Robotics* 35, 596–611. doi: 10.1002/rob.21763

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. New Orleans, LA, USA. 11976–11986. IEEE. doi: 10.1109/CVPR52688.2022.01167

Ma, Y., Zhang, Z., Yang, H. L., and Yang, Z. (2021). An adaptive adversarial domain adaptation approach for corn yield prediction. *Comput. Electron. Agric.* 187, 106314. doi: 10.1016/j.compag.2021.106314

Matloob, A., Safdar, M. E., Abbas, T., Aslam, F., Khaliq, A., Tanveer, A., et al. (2020). Challenges and prospects for weed management in Pakistan: A review. *Crop Prot.* 134, 104724. doi: 10.1016/j.cropro.2019.01.030

Melander, B., Rasmussen, I. A., and Bàrberi, P. (2005). Integrating physical and cultural methods of weed control—examples from European research. *Weed Sci.* 53, 369–381. doi: 10.1614/WS-04-136R

Moazzam, S. I., Khan, U. S., Qureshi, W. S., Tiwana, M. I., Rashid, N., Hamza, A., et al. (2022). Patch-wise weed coarse segmentation mask from aerial imagery of sesame crop. *Comput. Electron. Agric.* 203, 107458. doi: 10.1016/j.compag.2022.107458

Nicolopoulou-Stamati, P., Maipas, S., Kotampasi, C., Stamatis, P., and Hens, L. (2016). Chemical pesticides and human health: the urgent need for a new concept in agriculture. *Front. Public Health* 4, 148. doi: 10.3389/fpubh.2016.00148

Pantazi, X.-E., Moshou, D., and Bravo, C. (2016). Active learning system for weed species recognition based on hyperspectral sensing. *Biosyst. Eng.* 146, 193–202. doi: 10.1016/j.biosystemseng.2016.01.014

Parra, L., Marin, J., Yousfi, S., Rincón, G., Mauri, P. V., and Lloret, J. (2020). Edge detection for weed recognition in lawns. *Comput. Electron. Agric.* 176, 105684. doi: 10.1016/j.compag.2020.105684

Patel, D., and Kumbhar, B. (2016). Weed and its management: A major threats to crop economy. *J. Pharm. Sci. Biosci. Res.* 6, 453–758.

Peng, H., Li, Z., Zhou, Z., and Shao, Y. (2022). Weed detection in paddy field using an improved RetinaNet network. *Comput. Electron. Agric.* 199, 107179. doi: 10.1016/j.compag.2022.107179

Picon, A., San-Emeterio, M. G., Bereciartua-Perez, A., Klukas, C., Eggers, T., and Navarra-Mestre, R. (2022). Deep learning-based segmentation of multiple species of weeds and corn crop using synthetic and real image datasets. *Comput. Electron. Agric.* 194, 106719. doi: 10.1016/j.compag.2022.106719

Radoglou-Grammatikis, P., Sarigiannidis, P., Lagkas, T., and Moscholios, I. (2020). A compilation of UAV applications for precision agriculture. *Comput. Networks* 172, 107148. doi: 10.1016/j.comnet.2020.107148

Raja, R., Slaughter, D. C., Fennimore, S. A., Nguyen, T. T., Vuong, V. L., Sinha, N., et al. (2019). Crop signalling: A novel crop recognition technique for robotic weed control. *Biosyst. Eng.* 187, 278–291. doi: 10.1016/j.biosystemseng.2019.09.011

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: N. Navab, J. Hornegger, W. Wells and A. Frangi (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science. (Cham: Springer), vol 9351. doi: 10.1007/978-3-319-24574-4_28

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., and Mccool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *sensors* 16, 1222. doi: 10.3390/s16081222

Sarvini, T., Sneha, T., Gs, S. G., Sushmitha, S., and Kumaraswamy, R. (2019). "Performance comparison of weed detection algorithms," in *2019 International Conference on Communication and Signal Processing (ICCSP)*. Chennai, India. IEEE. 0843–0847. doi: 10.1109/ICCSP.2019.8698094

Seelan, S. K., Laguette, S., Casady, G. M., and Seielstad, G. A. (2003). Remote sensing applications for precision agriculture: A learning community approach. *Remote Sens. Environ.* 88, 157–169. doi: 10.1016/j.rse.2003.04.007

Tavakoli, H., Alirezazadeh, P., Hedayatipour, A., Nasib, A. B., and Landwehr, N. (2021). Leaf image-based classification of some common bean cultivars using discriminative convolutional neural networks. *Comput. Electron. Agric.* 181, 105935. doi: 10.1016/j.compag.2020.105935

Tranheden, W., Olsson, V., Pinto, J., and Svensson, L. (2021). "Dacs: Domain adaptation *via* cross-domain mixed sampling," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, HI, USA. IEEE. 1379–1389. doi: 10.1109/WACV48630.2021.00142

Tsai, Y.-H., Hung, W.-C., Schulter, S., Sohn, K., Yang, M.-H., and Chandraker, M. (2018). "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. Salt Lake City, UT, USA. 7472–7481. IEEE.

Tu, M., Hurd, C., and Randall, J. M. (2001). *Weed control methods handbook: tools & techniques for use in natural areas. All U.S. Government Documents (Utah Regional Depository)*. Paper 533. Available at: https://digitalcommons.usu.edu/govdocs/533

Vu, T.-H., Jain, H., Bucher, M., Cord, M., and Pérez, P. (2019). "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2517–2526. doi: 10.1109/CVPR.2019.00262

Yang, Y., and Soatto, S. (2020). "Fda: Fourier domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Seattle, WA, USA. IEEE. 4085–4095. doi: 10.1109/CVPR42600.2020.00414

Zhang, D., and Khoreva, A. (2019). Progressive Augmentation of GANs. In *33rd Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2881–2890).