Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

Real time detection of inter-row ryegrass in wheat farms using deep learning

Daobilige Su [a,b], Yongliang Qiao [b,*], He Kong [b], Salah Sukkarieh [b]

[a] College of Engineering, China Agricultural University, 100083, Beijing, China
[b] Australian Centre for Field Robotics, The Rose Street Building J04, The University of Sydney, NSW 2006, Australia

ARTICLE INFO

Article history:
Received 25 November 2019
Received in revised form
18 January 2021
Accepted 19 January 2021
Published online 6 February 2021

Keywords:
Ryegrass
Wheat
Agricultural Robot
Crop Weed Classification
Semantic Segmentation
DNN

A key challenge for autonomous precision weeding is to reliably and accurately detect weed plants and crop plants in real time to minimise damage to surrounding crop plants while performing weeding actions. Specifically for a wheat farm, classifying ryegrass weed plants is particularly difficult even with human eyes since ryegrass shows visually very similar shape and texture to the crop plants themselves. A Deep Neural Network (DNN) that exploits the geometric location of ryegrass is proposed for the real time segmentation of inter-row ryegrass weeds in a wheat field. Our proposed method introduces two subnets in a conventional encoder-decoder style DNN to improve segmentation accuracy. The two subnets treat inter-row and intra-row pixels differently, and provide corrections to preliminary segmentation results of the conventional encoder-decoder DNN. A dataset captured in a wheat farm by an agricultural robot at different time instances is used to evaluate the segmentation performance, and the proposed method performs the best among various popular semantic segmentation algorithms. The proposed method runs at 48.95 Frames Per Second (FPS) with a consumer level graphics processing unit, thus is real-time deployable at camera frame rate.

© 2021 IAgrE. Published by Elsevier Ltd. All rights reserved.

1. Introduction

Autonomous weeding is a critical step in precision farming as it directly impacts crop health and yield (Slaughter et al., 2008). Detection of weed plants in crop plant rows reliably and precisely is a critical requirement for precision weeding, which comes in ways of spot spraying, mechanical tillage, mechanical stamping or laser burning, whilst minimising damage to surrounding vegetation (Lottes et al., 2018, 2020).

Wheat is one of the most important crop plants in many regions, including Australia (Golzarian & Frick, 2011). Among several common weed plant species existing in Australian wheat farms, the ryegrass weed plant is the most difficult one to detect and classify against wheat, since both of them have similar leaf shapes and show very similar optical reflectance in visible spectral range, as shown in the left image of Fig. 1(b). Conventionally, to get rid of ryegrass, farmers have to clean the field before seeding, apply specific herbicide, and carry out manual weeding during the growth stage of wheat. Such a process leads to additional costs of materials and labour, and extra usage of herbicide is not environmentally friendly. For robotic autonomous weeding, the visual similarity between wheat and ryegrass also makes off-the-shelf algorithms (Milioto & Stachniss, 2019; Badrinarayanan et al., 2017; Chen

* Corresponding author.
E-mail address: yongliang.qiao@sydney.edu.au (Y. Qiao).
https://doi.org/10.1016/j.biosystemseng.2021.01.019
1537-5110/© 2021 IAgrE. Published by Elsevier Ltd. All rights reserved.

Nomenclature

Name	*Description*
ACFR	Australian Centre for Field Robotics
CNN	Convolutional Neural Net
DNN	Deep Neural Network
FPS	Frames Per Second
IOU	Intersection Over Union
RMSE	Root-Mean-Square Error
$\mathcal{B}(\cdot)$	Batch normalisation
C	Number of Classes for semantic segmentation
$\mathcal{C}_s^{f\times f}(\cdot)$	2D convolution with filter size of $f \times f$ and stride of s
$\mathcal{D}(\cdot)$	spatial dropout operation
$\mathbf{e}_{Lm}$	Output from the main segmentation net after downsampling
$\mathbf{e}_1$	Feature map from the first block of encoder net
$f_c, \tilde{f}$	Frequency of class c and median value of f_c
F1	F1 score, the harmonic mean of the precision and recall
$\mathcal{L}_C(\cdot)$	Linear layer with output dimension of C
$\mathbf{L}_p$	Preliminary segmentation output from the main subnet
$\dot{\mathbf{L}}_m$	Correction or refinement from the middle subnet
$\dot{\mathbf{L}}_s$	Correction or refinement from the side subnet
$\mathbf{L}$	Final segmentation output
$mAcc$	Mean accuracy
$mIOU$	Mean intersection of union
$LReLU(\cdot)$	Leaky ReLU activation function
$MaxP(\cdot)$	Max-pooling operation
$Mask(\cdot)$	Mask of feature map according to locations of pixels
P_c, P	Number of pixels of class c and Total number of pixels
Pr_c, Rc_c	Precision and recall accuracies of class c
s_{pc}	Softmax value of pth pixel and cth class of DNN output
$\mathcal{T}_s^{f\times f}$	Deconvolution layer with filter size of $f \times f$ and stride of s
TP_c, FP_c, TN_c	True positive, false positive and true negative numbers of class c
w_c	Loss function weighting factor for cth class
$\mathbf{x}, \mathbf{y}$	Input features and output features
y_{pc}	One hot ground truth value of pth pixel and cth class
z_c	Logits output from the DNN corresponding to cth class

et al., 2017; Ronneberger et al., 2015; Zhao et al., 2017) work undesirably in detecting ryegrass in a wheat farm.

We propose a novel end-to-end trainable DNN based semantic segmentation method to classify inter-row ryegrass weed plants from wheat in real time by exploiting the geometric locations of ryegrass weed plants. Due to the competition existing against crop plants, weed plants can hardly survive in a densely grown wheat row. Therefore, there are very few intra-row ryegrass weed plants, and the majority of them are inter-row. The pixelwise segmentation results can be used to guide various type of weeding actuators to remove weed plants while avoiding damaging crop plants. The main part of the proposed method is based on the popular encoder-decoder structure, which is widely used in several state-of-the-art approaches (Milioto & Stachniss, 2019; Badrinarayanan et al., 2017; Chen et al., 2017; Ronneberger et al., 2015; Zhao et al., 2017). In addition to the encoder-decoder style main part, we introduce two novel subnets to improve or correct the preliminary segmentation results from the main part of the DNN, as inspired by the refinement architecture adopted in a cascaded manner in LiteFlowNet (Hui et al., 2018).

The objective of this paper is to tackle the challenging problem of semantic segmentation of inter-row ryegrass weed plants in a wheat farm, where conventional methods work undesirably in terms of segmentation accuracy or execution speed. We investigate whether the proposed method with the two novel subnets that exploit geometric location of weed plants can effectively improve segmentation performance both in terms of the pixel-wise accuracy and the object-wise accuracy, compared to conventional state-of-the-art methods. Furthermore, we examine the runtime speed of the proposed method to assure its real-time operation. Overall, we propose a novel method which obtains high segmentation accuracy and real-time execution speed that makes it suitable for robotic weeding.

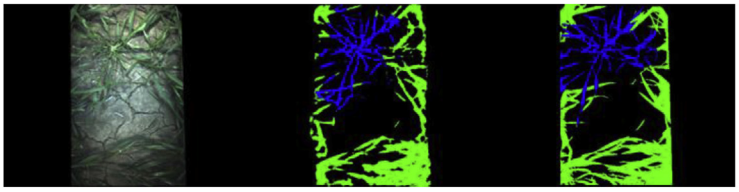
The contributions of this paper are two-fold. Firstly, to the best of the authors' knowledge, the proposed method serves as the first DNN based method to detect and classify ryegrass weed plants in a wheat farm via semantic segmentation, which provides a more accurate localisation of ryegrass compared to bounding box based detection for autonomous weeding action. Secondly, the proposed method introduces two novel subnets on top of a conventional encoder-decoder structured main segmentation net. By employing the two subnets, segmentation accuracy, especially for ryegrass weed plants, can be improved to a large degree, as detailed later in section 4.

The remainder of the paper is organised as follows. In section 2, related works of weed plant detection and crop plant-weed plant classification are discussed. In section 3, the details of the proposed method are described. In section 4, the experimental results of inter-row weed plant detection using the proposed method and performance comparisons with several state-of-the-art methods are presented. section 5 presents conclusions and discussions about further work.

2. Related work

Vision based crop plant-weed plant classification is a common approach for autonomous weeding systems, and there exist many works towards robust classification, detection and segmentation of crop plants and weed plants. Among them, many of the earlier approaches rely on hand crafted features (Haug et al., 2014; Milioto et al., 2017). In order to maximise the separability of crop plants and weed plants, parameters of the hand-crafted features are often optimised to adapt to a

(a) The Digital Farmhand robot developed by the Australian Centre for Field Robotics (ACFR) at The University of Sydney. The robot traverses through wheat bed and the camera attached to it traverses through early, mid or late stage wheat rows under the canopy, while streaming images online.



(b) Left to right: the input image, the segmentation result using the proposed method and the ground truth mask. In segmentation results, blue indicates ryegrass, green indicates wheat and black indicates soil.

Fig. 1 – The agricultural robot and the weed plant detection. (a) Data is collected by an agricultural robot, The Digital Farmhand. (b) Left to right are the captured image, the segmentation result using the proposed method and the ground truth mask.

specific application. When used for a different situation, these features usually cannot be applied directly and need to be adapted further to fit the new situation. Most of them focus on developing complicated nonlinear classifiers, *e.g.* SVM (Guerrero et al., 2012) and boosting (Ahmad et al., 2018) to tackle limitations of the adopted features.

With the advent of deep Convolution Neural Network (CNN), many recent methods for crop plant-weed plant classification adopted DNN (Milioto et al., 2018; Lottes et al., 2017, 2018; Bosilj et al., 2020; Milioto & Stachniss, 2019). Here, we differentiate two terms CNN and DNN, where CNN refers to a neural net with convolutional layer, while DNN refers to a neural net with multiple layers. By adopting CNN, the models overcome the limitations of hand-crafted pipelines described above. They allow to learn feature representations directly from the training data using backpropagation during model training. The DNN based approaches yield richer feature representation which were aggregated over multiple layers of convolutions, pooling operations, and nonlinearities. This makes them achieve superior performance with simple linear classifiers on top of these more complex features, compared to the aforementioned simpler hand-crafted features. Having said that, DNN based methods also have disadvantages. They require a large amount of data, and are computationally expensive to train. Determination of structure, training method, and hyperparameters for DNN is considered a "black art". Furthermore, what a DNN learned is not easy to comprehend.

Based on the output forms, these methods can be divided into three categories, *i.e.* classification, detection and segmentation. Compared to classification and detection, semantic segmentation provides a class label for each pixel of an image, yielding more accurate localisation of objects. It enables applications that require accurate object masks. Specifically for crop plant-weed plant segmentation, Lottes et al. (2017) proposed a method to classify sugar beets against weed plants with RGB-NIR images. Their method relied on pre-segmentation of the vegetation from the background based on the NDVI index. The method used a random forest classifier, which combined appearance and geometric properties. Milioto et al. (2018) presented a CNN-based semantic segmentation approach to separate sugar beet plants, weed plants, and background in crop plant fields. The method exploited existing vegetation indices and can execute the classification task in real time. Milioto and Stachniss (2019) presented Bonnet, which is a stable and easy-to-use tool for deploying a semantic segmentation DNN. Bonnet made DNN technology more approachable in the context of autonomous systems, and demonstrated high segmentation performance for crop plant-weed plant segmentation. Bosilj et al. (2020) investigated the role of knowledge transfer between deep learning based segmentation for different crop plant types. Their method reduced the retraining time and labelling efforts required for a new crop plant. Lottes et al. (2018, 2020) presented DNNs for jointly segmenting both crop plant-weed plant and also their stems. Their work showed that, by

jointly segmenting crop plant-weed plant and stems, segmentation accuracy increased compared to the same architecture of DNN segmenting only one of them. Milioto and Stachniss (2019) proposed a DNN, which produced joint semantic and instance segmentation. The method adopted three decoders that produce reduced resolution semantic segmentation, object centre prediction and matrix learning respectively. Then a post-processing step was used to provide joint semantic and instance segmentation of input image resolution using super pixel upsampling.

Although these methods achieved remarkable segmentation accuracy, crop plants and weed plants present in the dataset used in these methods have clear difference in terms of colour texture or leaf shape. When leaves of crop plants and weed plants have similar colour and shape and are more cluttered as shown in Fig. 1, performance of these methods tend to deteriorate as detailed later in Section 4.

More specifically, for crop plant-weed plant classification related to wheat and ryegrass, Gerhards and Christensen (2003) presented a review on earlier works and a system for site-specific weed plants control in different farms, *i.e.* sugarbeet, maize, winter wheat and winter barley, respectively. The system also included online weed plant detection using digital image analysis. However, their work did not differentiate crop plants against weed plants. Girma et al. (2005) used multi spectrum information to classify cheat and ryegrass weed plants against wheat. The method essentially identified an input image as a cheat, ryegrass or wheat. However it did not localise where the plant was on the image. Wang et al. (2007) presented an approach to classify a multispectral image patch as a crop plant or a weed plant in a wheat farm, then joined many classified image patches to generate crop plant-weed plant map. Golzarian and Frick (2011) presented a method to segment out the vegetation part of image and use PCA to classify wheat, ryegrass and brome grass species. Shapira et al. (2013) also used multi-spectrum images to classify various weed plant species against wheat and chickpea.

Again, these methods classified an image as a predefined plant rather than localising them on the image. In addition, the classification based method does not work if there are two or more species existing in the same image. Our method goes one step further and tackles the semantic segmentation problem of inter-row ryegrass in a wheat farm, which assigns class label for each pixel in the image, and therefore localises weed plants or crop plants more precisely.

3. The proposed method

The proposed method consists of a main segmentation net and two subnets. Details about them are provided in sections 3.1 and 3.2.

3.1. *Main segmentation net*

Many state-of-the-art semantic segmentation methods based on DNN follow an encoder-decoder architecture (Long et al., 2015; Paszke et al., 2016; Romera et al., 2018; Fu et al., 2019; Milioto & Stachniss, 2019). The main segmentation net in the proposed method also follows such an architecture. Particularly, we adopt the specific architecture of Bonnet (Milioto & Stachniss, 2019), which is based on EFRNet (Romera et al., 2018). This particular architecture is favored since it ensures real-time execution by using non-bottleneck-1D modules.

The structure of the main segmentation net is illustrated in Fig. 2. As can be seen from the figure, the main segmentation net follows a sequential structure. Firstly, it uses three encoder blocks to obtain a downsampled feature map. Then, another three decoder blocks are used to upsample the encoded feature map to the input image spatial resolution. Finally, a linear classification layer is used to assign a class label to each pixel of the image using the decoded logits.

For the encoder part of the main segmentation net, each encoder block consists of one downsampling block and several residual blocks. Particularly, as shown in Fig. 3, the downsampling block concatenates outputs of a 5×5 convolution layer with stride 2 and a Max-Pooling layer with kernel size of 2. The formulation of the downsampling block can be written as follows,

$$\begin{cases} \mathbf{c} = LReLU(\mathcal{B}(\mathcal{C}_2^{5\times5}(\mathbf{x}))) \\ \mathbf{m} = MaxP(\mathbf{x}) \\ \mathbf{y} = [\mathbf{c}, \mathbf{m}] \end{cases} \tag{1}$$

where, $\mathcal{C}_2^{5\times5}(\cdot)$ denotes 2*D* convolution function, in which subscript 2 means a stride of 2 and superscript 5×5 represents the filter size; $\mathcal{B}(\cdot)$ represents batch normalisation; $MaxP(\cdot)$ denotes max-pooling operation; $LReLU(\cdot)$ represents leaky ReLU activation function; $\mathbf{x}$, $\mathbf{y}$ denote input and output features respectively; $[\mathbf{c}, \mathbf{m}]$ is concatenation of $\mathbf{c}$ and $\mathbf{m}$. The functionality of the downsampling block is to reduce the spatial resolution of the input image or feature map by a factor of 2. Such a block enables deeper layers to acquire more information while reducing computational cost.

The factorised residual block of the encoder is denoted as the non-bottleneck-1D block (He et al., 2016). The non-bottleneck-1D block essentially consists of two pairs of convolutional layers of non-bottleneck-1D asymmetric kernels. As shown in Fig. 4, the non-bottleneck-1D block can be formulated as,

$$\begin{cases} \mathbf{c}_1 = LReLU(\mathcal{C}_1^{1\times3}(\mathcal{C}_1^{3\times1}(\mathbf{x}))) \\ \mathbf{c}_2 = \mathcal{D}(\mathcal{B}(\mathcal{C}_1^{1\times3}(\mathcal{C}_1^{3\times1}(\mathbf{c}_1)))) \\ \mathbf{y} = LReLU(\mathbf{c}_2 + \mathbf{x}) \end{cases} \tag{2}$$

where, $\mathbf{x}$ and $\mathbf{y}$ represent input and output features as explained before; $\mathcal{D}(\cdot)$ represents spatial dropout operation; $\mathcal{C}_1^{3\times1}$, $\mathcal{C}_1^{1\times3}$ denote convolutional layer, which are with stride 1 and $3\times$ 1, 1×3 kernels respectively. The residually formulated non-bottleneck-1D block improves the learning capacity of DNN, as it significantly reduces the degradation problem present in popular architectures which stack a large amount of layers (Romera et al., 2018).

The design of the non-bottleneck-1D residual module of ERFNet entirely uses convolutions with 1D filters. It is shown that a 2D filter can be represented by a combination of 1D filters, which results in decomposed filters that have intrinsic simplicity, and thereby low computational cost (Alvarez &

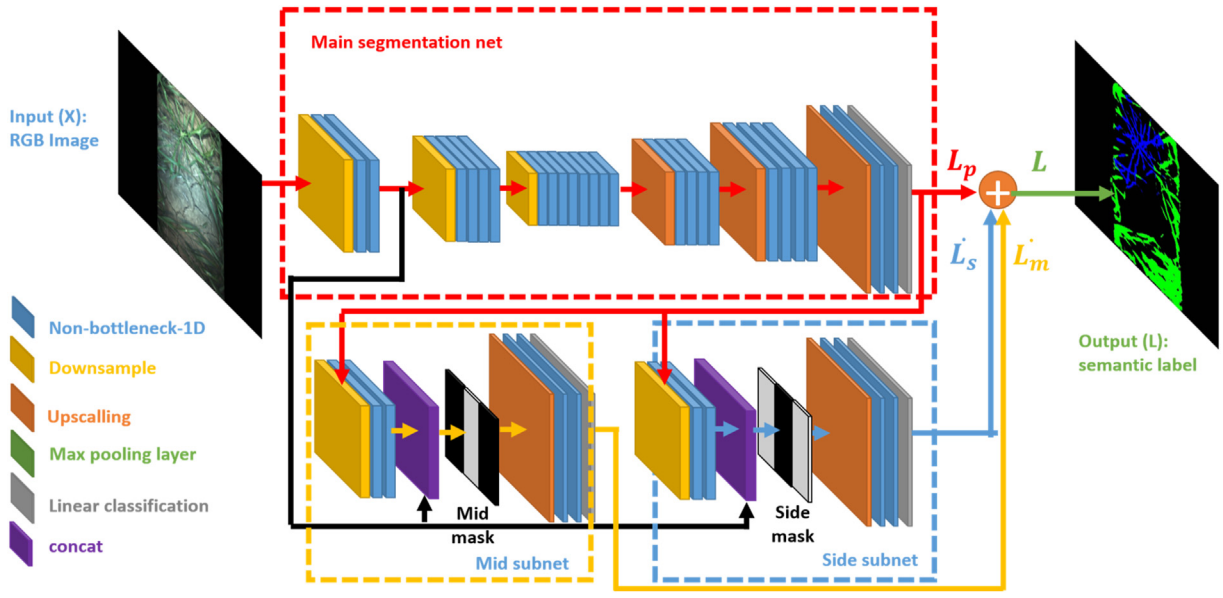


Fig. 2 – The architecture of the proposed DNN.

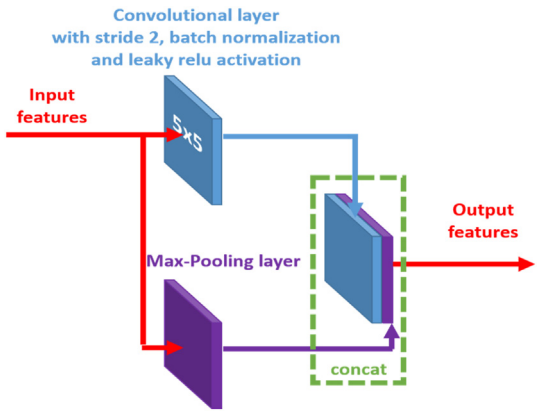


Fig. 3 – The architecture of downsampling block.

Petersson, 2016). Compared to the conventional $N\times N$ convolutional filters, the non-bottleneck-1D residual module also improves compactness of DNN, because filters are shared within each 2D combination. By replacing a $N \times N$ convolutional filter with two $N \times 1$ and $1 \times N$ filters, it reduces the network parameters as well as computation cost. For example, according to Romera et al. (2018), it yields a 33% reduction in network parameters when replacing a 3×3 convolutions with a pair of 3×1 and 1×3 filters. More reduction will be achieved for larger filter sizes. In addition, adopting non-bottleneck-1D residual module also theoretically improves the learning capacity of DNN by inserting a non-linearity between 1D filters (Romera et al., 2018; Alvarez & Petersson, 2016).

On the other hand, decoder is used to upsample the encoded feature map from the previous step and make it match the input resolution of the image. As shown in Fig. 2, an upsampling block and several residual blocks form a decoder segment. Such an upsampling block adopted by ERFNet, different to max-unpooling operation used by SegNet and ENet, includes simple deconvolution layers with stride 2. We can formulate the upsampling block as follows,

$$y = \mathcal{T}_2^{2\times 2}(\mathbf{x}), \tag{3}$$

in which $\mathcal{T}_2^{2\times 2}$ denotes a deconvolution layer with stride 2 and 2×2 kernel. The upsampling block does not require to share the pooling indexes from the encoder due to the usage of the deconvolution layer. Therefore, it can reduce memory and computation overhead (Romera et al., 2018).

As the last step, a linear classification layer is used by the main segmentation net to assign logits of dimension C to each pixel of the image according to the corresponding decoder output. The formulation of the linear classification layer is as follows,

$$y = \mathcal{L}_C(\mathbf{x}), \tag{4}$$

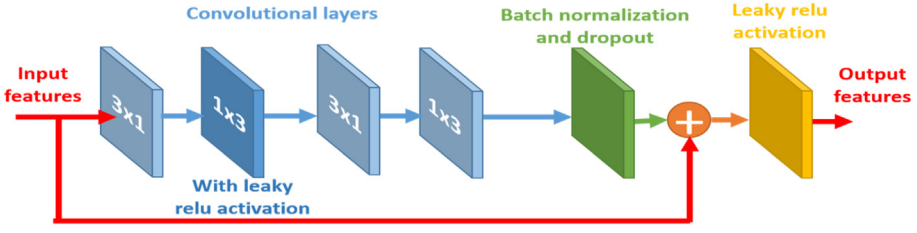
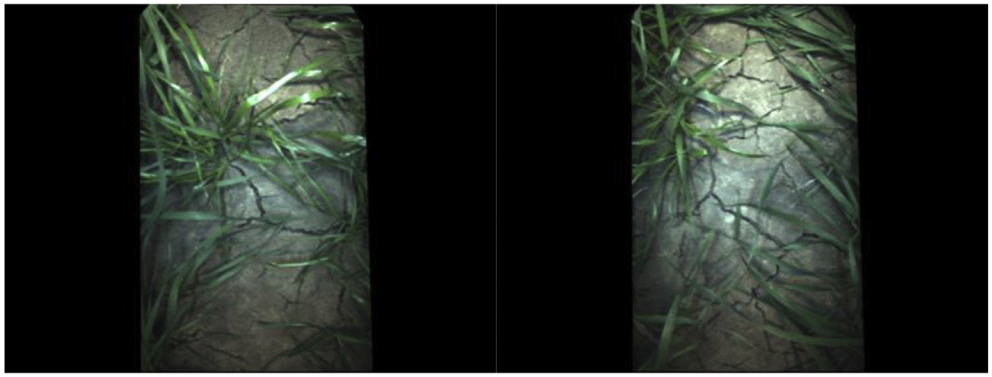


Fig. 4 – The architecture of non-bottleneck-1D block.

(a) Image from 2-week time instance. (b) Image from 2-week time instance.



(c) Image from 4-week time instance. (d) Image from 4-week time instance.

Fig. 5 – Typical images captured during two time instances of the wheat farm. Particularly (a) and (b) were captured at 2-week time instance of the wheat farm, while (c) and (d) were captured at 4-week time instance of the wheat farm. Two sets of images were captured under different weather and lighting conditions, therefore showed severe different brightness.

where $\mathcal{L}_C(\cdot)$ denotes the linear layer with output dimension of C.

The output logits of the main segmentation net provide a preliminary segmentation result $\mathbf{L}_p$. However, this preliminary segmentation result $\mathbf{L}_p$ yields poor segmentation accuracy due to the challenging situation of classifying ryegrass against wheat. Due to the similarity of both leaf colours and shapes between ryegrass and wheat, the main segmentation net often mis-classifies between them. The proposed method uses two subnets to provide refinement or correction to the preliminary segmentation result, as detailed next in Section 3.2.

3.2. *Subnets for correction or refinement*

As shown in Fig. 2, the two subnets use one downsampling block and two non-bottleneck-1D blocks to reduce the spatial dimension of the preliminary result from the main segmentation net, and concatenate it and image feature map from the first encoder layer. Then they mask concatenated result according to the location of middle or two sides of the row. We denote the subnet that extracts the middle part of the row as the middle subnet, and the subnet extracts two sides of the row as the side subnet, respectively. Essentially, the two subnets detect inter-row ryegrass weed plants by treating pixels in the middle of the row and two sides of the row differently. The concatenation layer and masking layer can be formulated as follows,

$$\begin{cases} \mathbf{c}_s = [\mathbf{e}_{Lm}, \mathbf{e}_1] \\ \mathbf{y} = Mask(\mathbf{c}_s), \end{cases} \tag{5}$$

where

$$Mask(\mathbf{c}_s) = \begin{cases} 1 \text{ for pixels in middle} \\ 0 \text{ for pixels in two sides}, \end{cases} \tag{6}$$

for the middle subnet, and

$$Mask(\mathbf{c}_s) = \begin{cases} 0 \text{ for pixels in middle} \\ 1 \text{ for pixels in two sides}, \end{cases} \tag{7}$$

for the side subnet. In Eq. (5), $\mathbf{e}_{Lm}$ is the output logits from the main segmentation after going through the downsampling block of the subnet; $\mathbf{e}_1$ is the feature map from the first block of encoder net; function $Mask(\cdot)$ masks the concatenated feature map according to locations of pixels. Here, simple rectangular regions are used, as shown in Fig. 2 as middle and side masks, to separate pixels that are more likely to be weed plants and crop plants respectively. This is

Table 1 – Summary of four parts of the dataset used in the paper.

Dataset	2-week (part 1)	4-week (part 2)	2 + 4 (part 3)	2 + 4+broadleaf (part 4)
Number of images	174	101	275	517
Resolution	1024 × 768	1024 × 768	1024 × 768	1024 × 768
Train and test images	122, 52	71, 30	191, 82	366, 151
Percentage of background pixels	95.48%	89.60%	92.54%	91.09%
Percentage of rye grass weed pixels	0.95%	1.08%	1.02%	0.57%
Percentage of broadleaf weed pixels	N/A	N/A	N/A	2.00%
Percentage of crop pixels	3.55%	9.30%	6.43%	6.32%

because our camera looks straight downward and the inter-row space between two crop plant rows on the image shows a simple rectangle region. In more complex situations, *e.g.* when the camera looks at ground at a certain angle, custom defined masks can be used to properly define the inter-row space.

The operation above is followed by an upsampling block and two non-bottleneck-1D blocks to upsample the feature map to the spatial resolution of the input image. Then a linear classification layer is used to produce the correction or refinement $\dot{\mathbf{L}}_m$ and $\dot{\mathbf{L}}_s$ to the preliminary segmentation $\mathbf{L}_p$.

Summarising the above operations, the two subnets essentially utilise the first encoder block feature map $\mathbf{e}_1$ and the preliminary segmentation $\mathbf{L}_p$ to compute the error in the coarse preliminary segmentation $\mathbf{L}_p$. Finally, the segmentation result $\mathbf{L}$ is given by summing the preliminary segmentation $\mathbf{L}_p$ and two correction or refinement terms $\dot{\mathbf{L}}_m$ and $\dot{\mathbf{L}}_s$ as follows,

$$L = \mathbf{L}_p + \dot{\mathbf{L}}_m + \dot{\mathbf{L}}_s. \tag{8}$$

The final logits $\mathbf{L}$ goes through a softmax layer to compute each pixel's probability to each class.

3.3. Dataset

The dataset used in this paper was collected by the authors of this paper in a wheat farm in Narrabri, NSW, Australia. Specifically, the Digital Farmhand, a multi purpose agricultural robot developed by the ACFR as shown in Fig. 1(a), was equipped with a downward looking JAI AD-080-GE multi-spectral camera (RGB + NIR camera), installed underneath the robot inside the weeder. The robot traversed through every row in the farm with wheat growing in different stages of their life circles. Images were captured in two different time instances of the wheat growth, which are 2-week time instance and 4-week time instance after the emergence of their first leaves. These two time instances correspond to Feekes scale 1–2 and Feekes scale 3–4, respectively. The camera captured images under the canopies between two rows of wheat continuously at 30 FPS with a resolution of 1024 × 768. The absolute exposure time was set to be 100 μs for RGB camera and 250 μs and 500 μs for NIR camera for 2-week and 4-week time instance wheat farm respectively. Other parameters like F-stop, ISO, shutter speed were automatically set by the camera driver. Each channel of the camera was set to be quantised by 256 levels. Only RGB images are used in this paper. The physical cell size of the sensor was 4.65 × 4.65 μm. The distance from the camera to the ground was 200 – 300 mm, and the camera field of view on the ground was approximately 530 × 400 mm. We cropped out uninformative region consist of weeder wall at the left and right sides of the image. Therefore, the informative part of the camera field of view was around 250 × 400 mm. The robot was moving at approximately 0.2 m s^{-1}. One large LED light and two halogen lights were used to provide ambient lighting for camera images. Images captured under the canopy of wheat farm, especially during the middle and the late growth stages, are usually challenging, since the camera perceived images with limited field of view and leaves of crop plants and weed plants are usually interlaced, as shown in Fig. 5. This increases the difficulty for both accuracy of the DNN and manual labelling of ground truth labels.

In order to collect more images with weed plants, we purposely planted some weed plants in addition to those naturally grown ones. Images captured at these two time instances also encountered different weather and lighting conditions, which introduced severely different brightness values in two sets of images. In particular, artificial lighting was set stronger for images captured at the 4-week time instance than the 2-week time instance. Typical images captured during these two time instances are shown in Fig. 5,

Table 2 – The details of the DNN and training parameters.

Params	Values
kernel no. lyr 1-3	8
kernel no. lyr 4-8	16
kernel no. lyr 9-17	65
kernel no. lyr 18-22	32
kernel no. lyr 23-27	16
kernel no. lyr 28-30	8
kernel no. lyr 32-34	8
kernel no. lyr 37-39	8
train epochs	200
balancing	median freq
focal loss γ	2
learning rate (l.r.)	$1e^{-4}$
l.r. decay	1.5
l.r. decay epochs	10
Adam ε	$1e^{-8}$
batch size	8
input size	512 × 512

Table 3 – The pixel-wise semantic segmentation accuracies of the proposed method and comparison against existing methods.

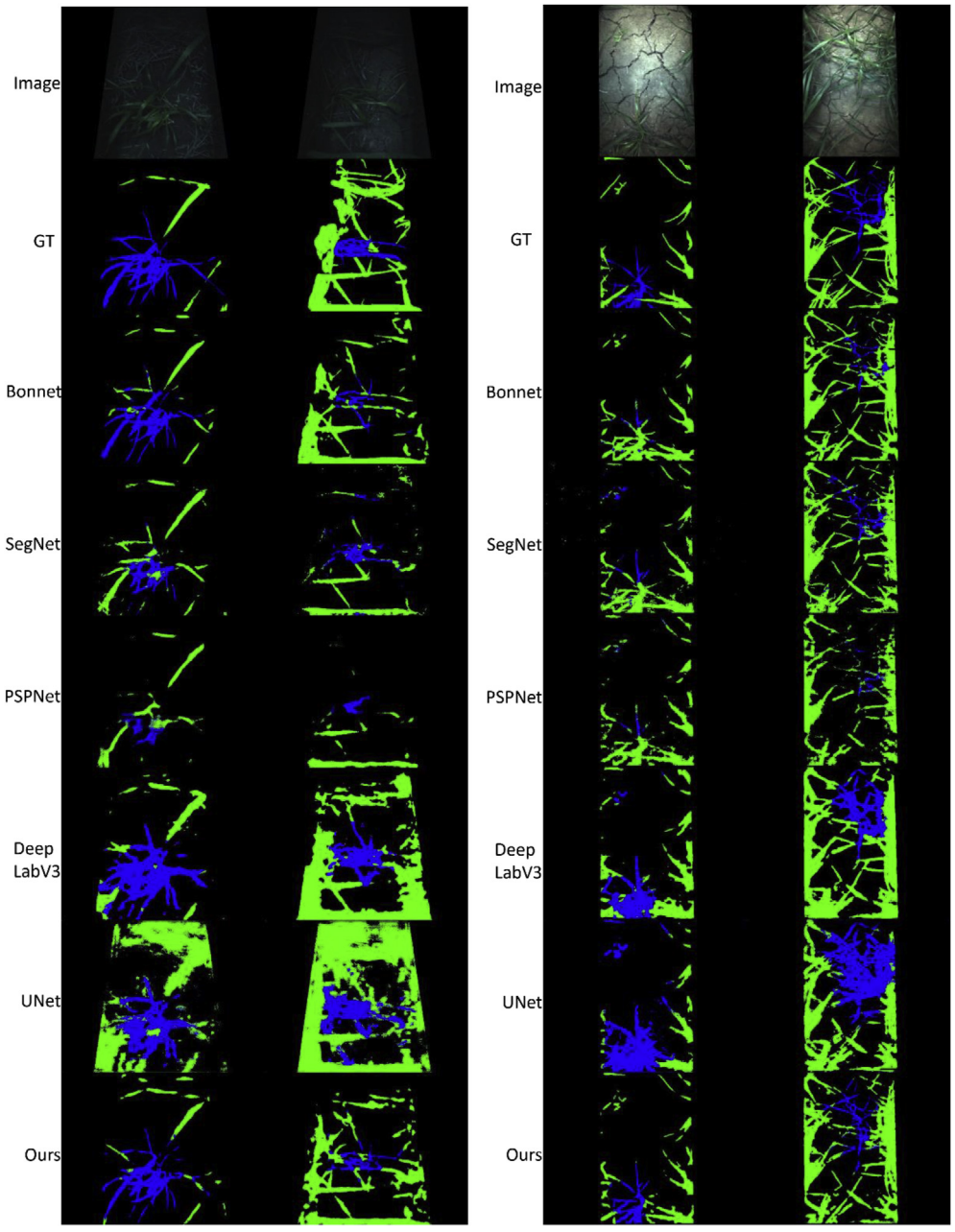
Dataset	Method	mAcc(%)	mIOU(%)	F1 (%)				IOU(%)			
				BG	Wheat	Rye	Broad	BG	Wheat	Rye	Broad
2-week	Bonnet	**97.66**	60.46	**98.87**	**63.31**	54.32	N/A	**97.77**	46.32	37.29	N/A
	SegNet	96.77	50.35	98.41	52.46	31.37	N/A	96.88	35.55	18.60	N/A
	PSPNet	97.07	46.06	98.53	48.41	16.73	N/A	97.10	31.93	09.12	N/A
	DeepLabV3	95.19	55.24	97.56	53.25	50.93	N/A	95.24	36.29	34.17	N/A
	UNet	92.84	48.96	96.35	43.26	41.65	N/A	92.95	27.60	26.30	N/A
	Ours	97.63	**63.10**	98.82	61.52	**64.12**	N/A	97.68	**47.19**	**44.42**	N/A
4-week	Bonnet	96.23	67.79	98.22	81.78	54.72	N/A	96.51	69.18	37.66	N/A
	SegNet	94.10	59.14	97.13	72.97	40.71	N/A	94.42	57.44	25.56	N/A
	PSPNet	94.60	54.83	97.40	72.94	21.61	N/A	94.95	57.40	12.11	N/A
	DeepLabV3	94.12	65.12	96.95	77.49	55.10	N/A	94.08	63.25	38.03	N/A
	UNet	91.69	56.36	96.24	71.61	34.07	N/A	92.76	55.78	20.53	N/A
	Ours	**96.35**	**71.86**	**98.24**	**82.66**	**65.38**	N/A	**96.54**	**70.45**	**48.56**	N/A
2 + 4	Bonnet	97.14	64.00	98.64	74.49	52.17	N/A	97.33	59.36	35.29	N/A
	SegNet	94.87	51.40	97.45	59.71	28.43	N/A	95.04	42.56	16.57	N/A
	PSPNet	96.15	54.82	98.12	65.64	32.33	N/A	96.32	48.86	19.28	N/A
	DeepLabV3	95.46	62.43	97.71	69.68	55.36	N/A	95.52	53.47	38.28	N/A
	UNet	93.22	52.02	96.74	60.88	31.34	N/A	93.69	43.76	18.58	N/A
	Ours	**97.33**	**67.74**	**98.71**	**75.90**	**61.67**	N/A	**97.47**	**61.16**	**44.58**	N/A
2 + 4+Broad	Bonnet	95.64	61.16	97.94	70.21	54.55	72.66	95.97	54.10	37.50	57.06
	SegNet	94.80	53.41	97.59	70.74	48.33	48.17	95.29	54.73	31.87	31.73
	PSPNet	95.81	54.34	98.14	72.87	49.56	47.02	96.36	57.32	32.94	30.74
	DeepLabV3	94.09	62.09	96.85	70.71	56.35	**75.39**	93.91	54.69	39.23	**60.50**
	UNet	90.88	37.40	96.06	52.18	18.72	20.69	92.43	35.30	10.32	11.54
	Ours	**96.22**	**64.21**	**98.18**	**74.20**	**65.95**	68.62	**96.43**	**58.98**	**49.20**	52.24

The bold number indicates the best performing method.

Table 4 – The object-wise semantic segmentation accuracies of the proposed method and comparison against existing methods.

Dataset	Method	mAcc(%)	F1 (%)			Precision (%)			Recall (%)		
			Wheat	Rye	Broad	Wheat	Rye	Broad	Wheat	Rye	Broad
2-week	Bonnet	87.25	87.67	71.42	N/A	**100.00**	**100.00**	N/A	78.04	55.55	N/A
	SegNet	82.35	84.93	20.00	N/A	9.687	**100.00**	N/A	75.60	11.11	N/A
	PSPNet	73.53	71.87	N/A	N/A	**100.00**	N/A	N/A	56.09	N/A	N/A
	DeepLabV3	95.10	93.67	88.88	N/A	97.36	88.88	N/A	90.24	88.88	N/A
	UNet	94.12	92.68	80.00	N/A	92.68	**100.00**	N/A	**92.68**	66.66	N/A
	Ours	**95.71**	**95.55**	**94.86**	N/A	**100.00**	**100.00**	N/A	91.48	**90.23**	N/A
4-week	Bonnet	94.29	94.33	84.61	N/A	89.28	**100.00**	N/A	**100.00**	73.33	N/A
	SegNet	87.14	86.20	57.14	N/A	75.75	**100.00**	N/A	**100.00**	40.00	N/A
	PSPNet	78.57	84.74	N/A	N/A	73.52	N/A	N/A	**100.00**	N/A	N/A
	DeepLabV3	97.14	98.03	92.85	N/A	96.15	**100.00**	N/A	**100.00**	86.66	N/A
	UNet	97.14	96.00	93.33	N/A	96.00	93.33	N/A	96.00	93.33	N/A
	Ours	**100.00**	**100.00**	**100.00**	N/A	**100.00**	**100.00**	N/A	**100.00**	**100.00**	N/A
2 + 4	Bonnet	86.55	84.29	78.94	N/A	92.72	**100.00**	N/A	77.27	65.21	N/A
	SegNet	81.29	84.61	16.00	N/A	85.93	**100.00**	N/A	83.33	08.69	N/A
	PSPNet	78.95	83.20	08.33	N/A	88.13	**100.00**	N/A	78.78	04.34	N/A
	DeepLabV3	95.91	**94.57**	95.45	N/A	9.682	**100.00**	N/A	**92.42**	91.30	N/A
	UNet	94.74	92.68	95.83	N/A	**100.00**	92.00	N/A	86.36	**100.00**	N/A
	Ours	**95.98**	91.99	**100.00**	N/A	94.54	**100.00**	N/A	89.58	**100.00**	N/A
2 + 4+Broad	Bonnet	86.97	82.92	75.00	81.81	81.73	96.00	**100.00**	84.15	61.53	69.23
	SegNet	87.27	88.88	63.15	65.51	80.64	**100.00**	**100.00**	99.00	46.15	48.71
	PSPNet	85.15	89.30	60.71	60.71	84.21	**100.00**	**100.00**	95.04	43.58	43.58
	DeepLabV3	**98.18**	**97.11**	**98.70**	93.15	**94.39**	**100.00**	**100.00**	**100.00**	**97.43**	87.17
	UNet	75.15	71.85	N/A	N/A	57.39	N/A	N/A	96.03	N/A	N/A
	Ours	95.48	90.89	94.86	**95.78**	89.69	**100.00**	**100.00**	92.13	90.23	**91.92**

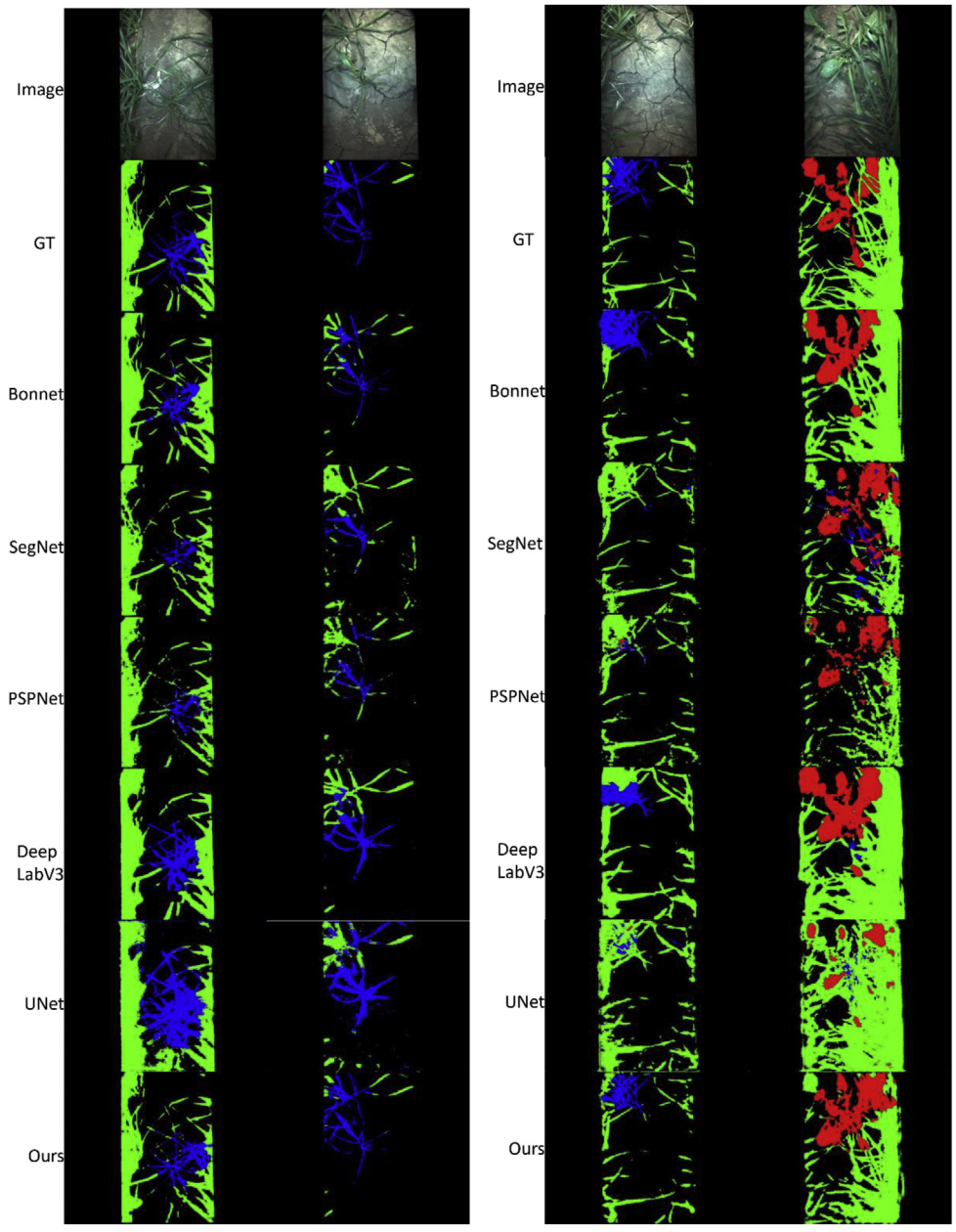
The bold number indicates the best performing method.



(a) 2-week growth stage. **(b) 4-week growth stage.**

Fig. 6 – Sample segmentation results for 2-week and 4-week time instance parts of the dataset. In all label images, black represents background, green represents the crop plant (wheat), blue represents the ryegrass weed plant and red represents the broad-leaf weed plant.

where (a) and (b) were images captured at 2-week time instance, and (c) and (d) at 4-week time instance. Among the more than 10,000 of captured images, 517 images were manually labelled pixel-wise. Specifically these images were labelled with the help of agricultural scientists and NDVI indexes of images. These images were used for detecting ryegrass weed plants in terms of semantic segmentation, which produced masks to classify each pixel in the image to be background, wheat, a ryegrass weed plant or a broad-leaf weed plant. In order to thoroughly evaluate the performance of the proposed method on various situations of captured images, we organised the whole dataset into four parts. In



(a) 2+4 week growth stage.

(b) 2+4 week growth stage with broad-leaf weed plants.

Fig. 7 – Sample segmentation results for 2 + 4 week time instance and 2 + 4 week time instance with broad-leaf weed plant parts of the dataset. In all label images, black represents background, green represents the crop plant (wheat), blue represents the ryegrass weed plant and red represents the broad-leaf weed plant.

the first part, only images from the 2-week time instance were used, and there were 174 labelled images. In the second part, only images from the 4-week time instance were used, and there were 101 labelled images. In the third part, images from both time instances were used, and there were 275 labelled images. In three parts mentioned above, the only weed plant appeared on the images is ryegrass. In the fourth part, images with other broad-leaf weed plants were also added, and there were 517 labelled images. The broad-leaf weed plants have a different categorical label to the ryegrass weed plant. We refer these four parts of the dataset as 2-week, 4-week, 2 + 4 and 2 + 4+Broad in the reminder of

this paper for simplicity. For all four parts, we divided the total number of images into training and test sets by approximately 70% – 30%. Details about the dataset are summarised in Table 1.

3.4. Training and evaluation details

As explained before, the main segmentation net of the proposed method follows Bonnet (Milioto & Stachniss, 2019), whose architecture follows ERFNet (Romera et al., 2018). Therefore, hyperparameters of the main segmentation net were set to be the same as those used by Bonnet. The details of the training parameters are summarised in Table 2, including numbers of different CNN layers, epochs for training the DNN, class balancing method in loss function, gamma parameter used for focal loss, learning rate and its decay parameters, ε of Adam optimiser and the batch size. For the other five state-of-the-art semantic segmentation methods (Milioto & Stachniss, 2019; Badrinarayanan et al., 2017; Chen et al., 2017; Ronneberger et al., 2015; Zhao et al., 2017), we mainly used the default hyper parameters for each method, and slightly tuned few of them to better suit our dataset. All DNNs are trained for 200 epochs for a fair comparison.

Median frequency balancing was adopted in the proposed method to balance different classes with different pixel numbers to weight loss related to different classes, as it was used in (Milioto & Stachniss, 2019). The weighted losses $loss_w$ using median frequency balancing is formulated as follows,

$$loss_w(y,p) = -\sum_{p=1}^{P}\sum_{c=1}^{C} w_c y_{pc} log(s_{pc}), \tag{9}$$

In which s_{pc} is the softmax value, y_{pc} is the ground truth semantic label as a onehot representation, and P is the total number of pixels in the image. The softmax value s_{pc} can be formulated as,

$$s_{pc} = \frac{e^{z_c}}{\sum_{c=1}^{C} e^{z_c}}, \tag{10}$$

in which z_c is the output logits from the proposed DNN. The weighting function w_c for median frequency balancing is formulated as,

$$w_c = \frac{\tilde{f}}{f_c}, \tag{11}$$

where

$$f_c = \frac{P_c}{P}, \tag{12}$$

and $\tilde{f}$ represents the median value of $f_c (c = 1, \ldots, C)$.

In addition to the median frequency balancing, the focal loss function was adopted to penalise the hard examples. Particularly, it is formulated as follows,

$$loss_f(y,p) = -\sum_{p=1}^{P}\sum_{c=1}^{C} y_{pc}(1-s_{pc})^{\gamma} \log(s_{pc}). \tag{13}$$

Finally, the combination of median frequency balancing in Eq. (9) and the focus loss in Eq. (13) were added into the loss function when training the DNN. The combination of two loss can be formulated as,

$$loss_c(y,p) = -\sum_{p=1}^{P}\sum_{c=1}^{C} w_c y_{pc}(1-s_{pc})^{\gamma} log(s_{pc}), \tag{14}$$

with two hyper parameters ε and γ.

The quantitative assessment of the performance of the proposed DNN was conducted via mean accuracy and mean Interaction Over Union (IOU) in addition to the precision, recall, F1 score and IOU of each individual class. Particularly, the mean accuracy ($mAcc$) and the mean IOU ($mIOU$) were computed as,

$$mAcc = \frac{1}{C}\sum_{c=1}^{C} f_c \frac{TP_c}{TP_c + FP_c}, \tag{15}$$

$$mIOU = \frac{1}{C}\sum_{c=1}^{C} \frac{TP_c}{TP_c + FP_c + FN_c}, \tag{16}$$

in which C is the total number of classes; TP_c, FP_c, FN_c are number of true positive, false positive and false negative classification of pixels for cth class in the semantic segmentation problem. In addition to mean accuracy and mean IOU, F1 scores of individual classes were used to evaluate how the DNN performs on a specific class. The F1 score of class c, $F1_c$, is computed as follows,

$$F1_c = \frac{2Pr_c Re_c}{Pr_c + Re_c}, \tag{17}$$

where

$$Pr_c = \frac{TP_c}{TP_c + FP_c}, \tag{18}$$

$$Re_c = \frac{TP_c}{TP_c + FN_c}. \tag{19}$$

denote precision and recall values for class c, respectively.

In addition to the segmentation accuracy, another important metric is the runtime of the method. The runtime is measured as a single image forward pass time through the DNN. The single image forward pass time was evaluated by averaging the forward pass processing time of 200 images. All methods were tested on a Alienware 15 R4 laptop with 8th generation Intel(R) Core(TM) i9-8950HK CPU (6-Core, 12 MB Cache, Overclocking up to 5.0 GHz), 32 GB (2 × 16GB) DDR4

Table 5 – Inference time and inference FPS of the proposed method and comparison against existing methods.

Method	FPS(Hz)	Time (ms)
Bonnet	**63.05**	**15.90**
SegNet	7.07	141.48
PSPNet	19.55	51.15
DeepLabV3	6.03	165.71
UNet	12.10	82.6
Ours	48.95	20.40

The bold number indicates the best performing method.

2666 MHz memory, 512 GB PCIe M.2 SSD Class 40 solid state hard drive, 1 TB 7200RPM HDD, Nvidia(R) GeForce(R) GTX 1080 with 8 GB GDDR5X memory GPU and 240W Power Adapter.

These metrics were used to evaluate the proposed method, and compare it against existing methods.

4. Experimental results

In this section, we evaluate the performance of the proposed method using the evaluation matrix defined in section 3.4, *i.e.* F1 scores in Eq. (17) and IOUs of each individual class, mean accuracy and mean IOU in Eq. (15) and Eq. (16). For comparison purposes, we also show segmentation results of five state-of-the-art semantic segmentation methods, which are Bonnet (Milioto & Stachniss, 2019), SegNet (Badrinarayanan et al., 2017), PSPNet (Zhao et al., 2017), DeepLabV3 (Chen et al., 2017) and UNet (Ronneberger et al., 2015) respectively.[1]

The performance of the proposed method, and a comparison with other five state-of-the-art methods for semantic segmentation, on four different parts of the dataset detailed in Table 1, are summarised in Table 3. In all subsequent tables of this paper, the terms BG, Wheat, Rye and Broad represent background, wheat, the ryegrass weed plant and the broad-leaf weed plant respectively. It can be seen from the table that, the proposed method yields the most favourable overall performance in terms of mean accuracies and mean IOUs in almost all four parts of the dataset. Only in the part of 2-week time instance, the mean accuracy of the proposed method is slightly worse than Bonnet. The proposed method also outperforms the other methods in most of the F1 scores and IOUs for individual classes in four parts of the dataset. Especially, our proposed method achieves notable improvement in F1 scores and IOUs of ryegrass, compared to other methods, in all four parts of the dataset.

In addition to pixel-wise evaluation, another important metric especially for autonomous robotic weeding is the plant level object-wise evaluation, which is commonly used in the literature related to crop plant-weed plant classification for agricultural robots (Milioto et al., 2018; Lottes et al., 2020). In contrast to existing approaches (Milioto et al., 2018; Lottes et al., 2020), we cannot use the class-wise connected components to separate individual crop plants or weed plants, since images are captured under the canopy of the wheat farm, and crop plant and weed plant leaves are all interlaced together as shown in Fig. 5. Therefore, we manually create bounding boxes containing each object of weed plants on every image of test data in four parts of the dataset. Pixels classified as weed plants within the bounding box are treated as one weed plant. Furthermore, all pixels classified as crop plants in the image are treated as one object. Given the above definition, the object-wise evaluation metric in this paper tends to be higher than those used in (Milioto et al., 2018; Lottes et al., 2020). The object-wise evaluation of the proposed method, together with evaluation results of the other five methods, are presented in Table 4. It can be seen from the table that the proposed method achieves better overall performance than other methods in terms of overall mean accuracies and F1 scores of individual classes of 2-week, 4-week and 2 + 4 weeks parts of the dataset. Especially for the 4-week part of the dataset, all ryegrass weed plants in the test data were successfully detected by the proposed method. Its performance was only slightly worse than DeepLabV3 in the 2 + 4+Broad part of the dataset. Our explanation is that the structure of the proposed DNN, which treats the inter-row and two sides of the row differently, does not benefit in the situation of broad-leaf weed plants, which can appear anywhere in the crop plant row in our dataset. Therefore, the proposed DNN loses its advantages in such a situation. However, it is only slightly worse than DeepLabV3, and better than the rest of the methods in terms of overall mean accuracies and F1 scores of individual classes. Furthermore, on the 2 + 4+Broad part of the dataset, Unet does not converge properly during the training stage even after 200 epochs and different random weights initialisation.

Qualitative examples of the segmentation results are provided in Figs. 6 and 7. In both figures, columns top to bottom denote input RGB images, ground truth label images and segmentation results from Bonnet, SegNet, PSPNet, DeepLabV3, UNet and the proposed method. In all labelled images, black represents background, green represents the crop plant (wheat), blue represents the ryegrass weed plant and red represents the broad-leaf weed plant. Particularly in Fig. 6(a), two examples of 2-week part of the dataset are provided. It can be seen that the proposed method yields the most similar segmentation label to the ground truth. Bonnet and DeepLabV3 also provide comparable results. In Fig. 6(b), two examples of 4-week part of the dataset are provided. It can be seen that Bonnet, SegNet and PSPNet miss the greatest part of the ryegrass weed plants in two images, while our method successfully detects them thanks to the two introduced subnets which can capture the inter-row ryegrass weed plants. DeepLabV3 and UNet yield over-segmented results for the ryegrass. Figure 7(a) provides two examples of 2 + 4 week part of the dataset. We can see that SegNet, PSPNet and DeepLabV3 partially miss the top ryegrass weed plant in the right image, and Unet over segmented the ryegrass weed plant in the left image. Again, our method shows better results. Figure 7(b) shows two examples of 2 + 4 week with broad-leaf weed plants. It can be seen that SegNet, PSPNet and Unet miss the ryegrass weed plant in the left image. In the right image, the broad-leaf weed plant is missed by Unet, but is successfully detected by all other five methods. On the 2 + 4+Broad part of the dataset, Unet does not converge properly during the training stage even after 200 epochs and different random weights initialisation as explained before. The reason for the high success rate with broad-leaf weed plants is that they have clearly different leaf shape and texture to wheat and ryegrass, which makes them simpler to be distinguished.

Finally, the runtime performances of the proposed method together with the existing methods are presented in Table 5. We can see from the table that Bonnet achieves the highest

[1] The following open source implementations are used in the experiment. Bonnet: https://github.com/PRBonn/bonnet, SegNet: https://github.com/ykamikawa/tf-keras-SegNet, PSPNet: https://github.com/ykamikawa/tf-keras-PSPNet, DeepLabV3: https://au.mathworks.com/help/vision/ref/deeplabv3pluslayers.html and UNet: https://au.mathworks.com/help/vision/ref/unetlayers.html.

runtime performance with 63.05 FPS. Our method achieves 48.95 FPS inference speed. Both Bonnet and the proposed method are capable of processing images at camera framerate, which is usually 30 FPS for most of the off-the-shelf cameras. On the contrary, the rest of the methods cannot achieve a processing speed of the camera framerate, *i.e.* 30 FPS.

5. Conclusions

A DNN for real time segmentation of inter-row ryegrass weed plants in a wheat farm is proposed. The method exploits the geometric location of ryegrass weed plants, and introduces two novel subnets architecture to a conventional encoder-decoder style DNN to improve the segmentation accuracy, especially for detecting inter-row ryegrass weed plants. Comprehensive evaluation results show that the proposed method with two novel subnets yields superior performance over five other state-of-the-art semantic segmentation algorithms, especially on detecting ryegrass weed plants, both in terms of the pixel-wise accuracy and the object-wise accuracy. In addition, the proposed method also runs at 48.95 FPS with a consumer level GPU, which confirms its real-time deployment at camera frame rate. Future work includes visualisation of learned features that help the DNN to differentiate the ryegrass against wheat, and evaluation of the proposed method on segmentation of other types of inter-row weed plants.

Funding

This work was supported by the Grains Research & Development Corporation Australia [grant number 2018-PROC-9175526].

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

We thank Mr. Guy Coleman and Dr. Michael Walsh for their great support during the preparation of the testbed and the acquisition of the dataset.

REFERENCES

Ahmad, J., Muhammad, K., Ahmad, I., Ahmad, W., Smith, M. L., Smith, L. N., Jain, D. K., Wang, H., & Mehmood, I. (2018). Visual features based boosted classification of weeds for real-time selective herbicide sprayer systems. *Computers in Industry, 98*, 23–33.

Alvarez, J., & Petersson, L. (2016). In *Decomposeme: Simplifying convnets for end-to-end learning*. arXiv preprint arXiv:1606.05426.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(12), 2481–2495.

Bosilj, P., Aptoula, E., Duckett, T., & Cielniak, G. (2020). Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture. *Journal of Field Robotics, 37*(1), 7–19.

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). In *Rethinking atrous convolution for semantic image segmentation*. arXiv preprint arXiv:1706.05587.

Fu, J., Liu, J., Liu, J., Wang, Y., Zhou, J., Wang, C., & Lu, H. (2019). Stacked deconvolutional network for semantic segmentation. *IEEE Transactions on Image Processing*. https://doi.org/10.1109/TIP.2019.2895460.

Gerhards, R., & Christensen, S. (2003). Realtime weed detection, decision making and patch spraying in maize, sugarbeet, winter wheat and winter barley. *Weed Research, 43*(6), 385–392.

Girma, K., Mosali, J., Raun, W. R., Freeman, K. W., Martin, K. L., Solie, J. B., & Stone, M. L. (2005). Identification of optical spectral signatures for detecting cheat and ryegrass in winter wheat. *Crop Science, 45*(2), 477–485.

Golzarian, M. R., & Frick, R. A. (2011). Classification of images of wheat, ryegrass and brome grass species at early growth stages using principal component analysis. *Plant Methods, 7*(1), 28.

Guerrero, J. M., Pajares, G., Montalvo, M., Romeo, J., & Guijarro, M. (2012). Support vector machines for crop/weeds identification in maize fields. *Expert Systems with Applications, 39*(12), 11149–11155.

Haug, S., Michaels, A., Biber, P., & Ostermann, J. (2014). Plant classification system for crop/weed discrimination without segmentation. In *2014 IEEE winter conference on applications of computer vision* (pp. 1142–1149).

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

Hui, T.-W., Tang, X., & Loy, C. C. (2018). Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *2018 IEEE conference on computer vision and pattern recognition (CVPR 2018)* (Vol. 566, pp. 8981–8989).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *2015 IEEE conference on computer vision and pattern recognition (CVPR 2015)* (pp. 3431–3440).

Lottes, P., Behley, J., Chebrolu, N., Milioto, A., & Stachniss, C. (2018). Joint stem detection and crop-weed classification for plant-specific treatment in precision farming. In *2018 IEEE international conference on robotics and automation (ICRA 2018)* (pp. 8233–8238).

Lottes, P., Behley, J., Chebrolu, N., Milioto, A., & Stachniss, C. (2020). Robust joint stem detection and crop-weed classification using image sequences for plant-specific treatment in precision farming. *Journal of Field Robotics, 37*(1), 20–34.

Lottes, P., Hrferlin, M., Sander, S., & Stachniss, C. (2017). Effective vision-based classification for separating sugar beets and weeds for precision farming. *Journal of Field Robotics, 34*(6), 1160–1178.

Milioto, A., Lottes, P., & Stachniss, C. (2017). Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. In *2017 ISPRS annals of the photogrammetry, remote sensing and spatial information sciences* (p. 41).

Milioto, A., Lottes, P., & Stachniss, C. (2018). Real-time semantic segmentation of crop and weed for precision agriculture

robots leveraging background knowledge in CNNs. In *2018 IEEE international conference on robotics and automation (ICRA 2018)* (pp. 2229–2235).
Milioto, A., & Stachniss, C. (2019). Bonnet: An open-source training and deployment framework for semantic segmentation in robotics using CNNs. In *Proceedings of 2019 IEEE international conference on robotics and automation (ICRA 2019)*.
Paszke, A., Chaurasia, A., Kim, S., & Culurciello, E. (2016). In *Enet: A deep neural network architecture for real-time semantic segmentation*. arXiv preprint arXiv:1606.02147.
Romera, E., Alvarez, J. M., Bergasa, L. M., & Arroyo, R. (2018). Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems, 19*(1), 263–272.
Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *2015 International Conference on Medical image computing and computer-assisted intervention* (pp. 234–241).
Shapira, U., Herrmann, I., Karnieli, A., & Bonfil, D. J. (2013). Field spectroscopy for weed detection in wheat and chickpea fields. *International Journal of Remote Sensing, 34*(17), 6094–6108.
Slaughter, D. C., Giles, D. K., & Downey, D. (2008). Autonomous robotic weed control systems: A review. *Computers and Electronics in Agriculture, 61*(1), 63–78.
Wang, N., Zhang, N., Wei, J., Stoll, Q., & Peterson, D. E. (2007). A real-time, embedded, weed-detection system for use in wheat fields. *Biosystems Engineering, 98*(3), 276–285.
Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *2017 IEEE conference on computer vision and pattern recognition (CVPR 2017)* (pp. 2881–2890).