Journal of Scientific & Industrial Research
Vol. 81, May 2022, pp. 482-494

Efficient Weed Segmentation with Reduced Residual U-Net using Depth-wise Separable Convolution Network

R Arumuga Arun[1]* and S Umamaheswari[2]

[1]Department of Computer Technology, [2]Department of Information Technology, Anna University-MIT Campus, Chennai 600 044

Received 08 April 2021; revised 03 April 2022; accepted 20 April 2022

Selective weed treatment is a cost-effective method that reduces manpower and usage of the agrochemical, at the same time it requires an effective computer vision system to identify weeds and should be smaller in size to run in resource-constrained devices. To accomplish this, a convolution neural network named Reduced Residual U-Net using Depth-wise separable Convolution (RRUDC) network is proposed in this paper. Residual Depth-wise separable Convolution Block (RDCB) is introduced as a functional unit in both contractive and expanding paths. Residual connection is incorporated inside every RDCB unit. This network employs semantic segmentation to analyze the crop field images pixel-wise. To reduce the parameter size, a depth-wise separable convolution technique is used which curtail the number of parameters generated by the model at a ~1/9 ratio with a very negligible drop in accuracy. The model is trained using Crop Weed Field Image Dataset (CWFID) and then the trained model is pruned to reduce the model size further. It compresses the final model size by around ~70% without affecting the performance. It has achieved segmentation accuracy of ~96%, a lesser error rate with a model size less than 3 MB. It can be compatible with converting the proposed deep learning model into a real-time computer vision application that seems more convenient for farmers in their resource-constrained devices on their agricultural land.

Keywords: Computer vision, Convolution neural network, Deep learning, Pruning, Semantic segmentation, Weed detection

Introduction

The demand for food is growing rapidly due to the increase in the world population. Indian economy will reach greater heights if we cultivate and export more food products from our county. On the other side, the farmers involved in agriculture want to give up farming due to poor income and unbearable loss. So, smart farming techniques are required for the farmers to increase their productivity in agriculture, which helps the farmers to reduce their agricultural expenses. It involves robots in agricultural work to facilitate the automated process and reduces the labor required for agricultural work.[1]

Weed is one of the major threats to crop yield. The presence of weeds in crop fields has anindirect impact on both the quantity and quality of crop yield. Weeds impose degradation in yields in an invisible way. Manual removal of weeds and wide spraying of agrochemical products increase the expenses as well as affect the surrounding crops. Usage of more herbicides may be a cause of toxic produce.

A proper selective weed treatment is required to resolve this issue and it is a very important milestone in smart farming. The success of this process is based on the segmentation accuracy in differentiating weed from crop otherwise it affects surrounding crops. Computer vision techniques can be used to process the agriculture field digital images and extract significant information from them.[2]

The extracted information is passed to the robots. It automatically plucks or sprays herbicides in segmented weed portions in a smooth way without affecting surrounding crops and it also reduces the usage of agrochemicals.[3]

At first, the deep learning model is trained by crop field images. The input data for the training model contains field RGB images and their corresponding annotated images. In this, pixels are belonging to the background, crops, and weeds. So, pixel-wise labeling facilitates smooth and accurate segmentation.[4] Then the trained model is capable to segment new data. The network model architecture is Encoder-Decoder based Convolution Neural Network. In this, the blocks are organized in a way that each decoder block corresponds to its encoder block.[5,6] The encoder encodes the high dimensional image into low dimensional feature maps by extracting the important features and the decoder decodes the feature maps

*Author for Correspondence
E-mail: arun6f.rajesh@gmail.com

back to high dimensional one. To achieve the high accuracy of segmentation, the system model is built based on Unet[7,8], DeepUnet[9], and Residual Blocks.[10]

Most of the research work in the literature addresses achieving high accuracy of segmentation. Furthermore, in agricultural land, the user may have a resource constraints and low computational facilities. So, building a real-time application to aid farmers regarding automated selective weed treatment, require a deep learning model in a smaller size but without compromising the accuracy of classification. The pathway for achieving this is reducing the number of trainable parameters in the model and compressing the model.

The core operation of the convolution neural network is the convolution process. Usually, a convolution process with a large number of filters generates millions of parameters and billions of calculations. On the other hand, it extracts different types of important features from the image. In the process of reducing the number of parameters and computations, if we reduce the number of filters used in the convolution layers, it automatically affects the segmentation accuracy. Depth-wise separable convolution technique[11] is a suitable alternative for the convolution process which reduces the number of parameters at a high rate with a very negligible drop in accuracy.

Further reduction in model size can be done by compressing the trained model using the pruning technique. Thus, the final trained model is applied for filter pruning. It removes the redundant and unimportant filters. The final model size can be reduced significantly without affecting the segmentation accuracy.[12] The segmentation results of various architecture and proposed architecture are shown in Fig. 1 and Fig. 2 respectively.

Related Work

Badrinarayanan *et al.*[13] proposed an encoder-decoder-based SegNet architecture for autonomous driving car applications. In this work, the usage of pixel-wise labeling to segment road scene objects like vehicles, buildings, pedestrians, trees, and other objects for smooth driving of a car, is considered. Both encoder and decoder have 13 convolution layers on each side.

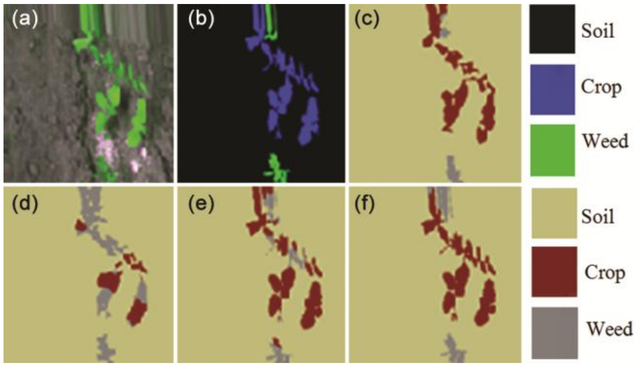

Fig. 1 — Segmentation results of different architecture for comparison: (a) Input RGB image, (b) Target Label image, (c) Prediction image of SegNet512, (d) Prediction image of U-Net, (e) Prediction image of Residual U-Net, (f) Prediction image of proposed RRUDC architecture

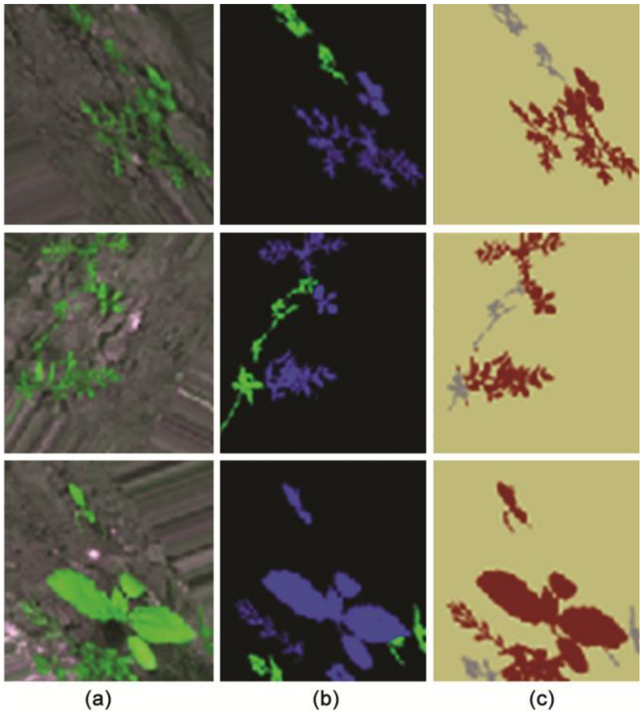

Fig. 2 — Few more segmentation results of proposed RRUDC architecture. (a) Input RGB image, (b) Target Label image, (c) Prediction image

Yasrab *et al.*[14] proposed a CNN model for driver assistance system (CSSA), which employed a semantic segmentation technique. The CSSA architecture was followed the SegNet architecture along with the dropout layer. The model size of CSSA was ~30 MB and achieved average class accuracy (CAA) was 60.2%.

The work proposed by Lottes *et al.*[15] detected vegetation by combining random forest classification and Markov random field in the sugar beet crop and weed dataset. The classification was done on the mobile robots and its perceptron triggers actuators for removing weeds and splashing herbicides on weeds.

In Umamaheswari & Jain[16], the objective was to use the SegNet architecture with pixel-wise segmentation for crop-weed classification and the dataset used here was Crop-Weed Field Images Dataset (CWFID). Here, Segnet256 and Segnet512 were proposed, in which SegNet512 produced an F1 score of ~96% for segmentation and the number of parameters generated by SegNet512 was quite large, that is 16.35 million. Sa *et al.*[17] used the dense semantic weed classification for crop-weed classification and it employed SegNet architecture.

Ronneberger *et al*.[8] proposed U-net architecture, which comprises a contracting path and a symmetric expanding path. The concatenation operation was used here, which is used to combine the feature map from the contractive path with expanding path, since this model used the unpadded convolution, there is a possibility of losing corner features. Xie *et al*.[7] used the UNet architecture in a modified version for segmenting the head-shoulder portion of pedestrians. It generated a huge number of parameters because it used a large number of filters in this architecture. Augustaukas & Lipnickas[18] proposed a pixel-wise road pavement defect detection method whose model structure was the U-Net architecture with the padded convolution process. This was used to detect the pixel-level cracks in the road which helped in the maintenance and monitoring of roads.

The objective of this work by Yang *et al*.[19] was to incorporate residual learning in the U-Net architecture called the Residual Dense U-Net (RDUN). It introduced the Residual DenseBlock (RDB) which enables a residual connection between the convolution layers. It was computationally expensive because it has used 42 convolution layers.

Naqvi *et al.*[10] proposed an ocular recognition system that was based on lite-residual encoder-decoder SegNet architecture. This network used 1000 original images along with data augmentation for training and testing. Li *et al*.[9] proposed a DeepUNet, whose structure was based on the U-Net architecture. In this work, DownBlock, and UpBlock were introduced in the contracting path and expansive path respectively. These blocks incorporated the plus connection between the convolution layers present in the block.

Chen *et al.*[20] proposed a separable convolution-based model with smaller size images like 32 × 32 images. The idea of a depth wise separable convolution instead of the standard convolution in its fire module of a SqueezeNet-DSC is obtained from the work proposed by Santos *et al*.[21] The SqueezeNet-DSC experienced a drop in accuracy of 4% when compared to SqueezeNet.

Mao *et al.*[22] proposed an object tracking system using deep learning and a Siamese network. In this work, the parameter reduction and model size reduction was achieved with depth-wise separable convolution and pruning process respectively. Even though it achieved a size reduction significantly, it used larger kernels like 11 × 11 and 7 × 7 in the network. So, there is a requirement for further improvement for this network architecture.

Huang *et al.*[12] proposed a “try-and-learn” algorithm that trains pruning agents to evacuate unnecessary CNN filters in a data-driven way. It succeeded in evacuating a significant number of filters in various CNN architectures like VGG16, ResNet, FCN32, and SegNet while maintaining the performance at an optimal level. In visual recognition and semantic segmentation tasks, there exists an average drop in accuracy up to 3.4%. The objective of this work by Cai *et al.*[23] was to use dropout techniques to prevent the model from overfitting. In this work, four different dropout methods namely, drop-neuron, drop-channel, drop-path, and drop-layer were proposed.

The objectives of this study are i) to build U-Net architecture with the incorporation of residual learning to increase the segmentation accuracy, ii) to reduce the number of parameters consumed by the model using depth-wise separable convolution, iii) to minimize the error rate using drop-out layer iv) to achieve the reduction in final model size using pruning techniques.

Materials and Methods

The primary objective of the proposed approach is to build an effective computer vision system using a deep learning model to facilitate pixel-wise labeling in-order to classify the crops and weeds accurately from the agricultural land. This can be used to aid robotics to do selective spraying and mechanical weed removal. The proposed model is efficient in all aspects such that achieving high segmentation accuracy, computationally less complex, reduction in model size, and lesser error rate. The workflow of the proposed work is shown in Fig. 3.

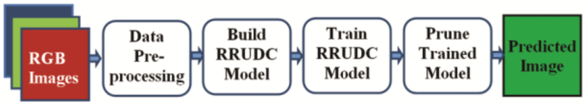

Fig. 3 — Workflow of proposed work

Methodology

Proposed Model Architecture

The proposed Reduced Residual U-Net using Depth-wise separable Convolution (RRUDC) is an Encoder-Decoder based U-Net architecture[8] in which standard convolution blocks are replaced with Depth-wise Separable Convolution blocks to reduce the number of parameters consumed by the model. This architecture comprises of totally 19 convolution layers which are lesser than some of the outstanding architectures: VGG19[24], SegNet[13], DeepUNet.[9] The proposed RRUDC architecture is illustrated in Fig. 4. All convolution layers are using the same padding convolution of constant 3 × 3 filter[24]; which is smaller than the filter size used in the models[18] and the stride value is fixed to 1 pixel. The output size of the convolution process can be found using Eq. (1).

$$S = \frac{F+2p-k}{s} + 1 \qquad \dots (1)$$

where, S and F are the sizes of output and input of the convolution process respectively, p is the padding pixel, k is the size of kernel or filter and s is the stride value.

In the proposed RRUDC, a Residual Depth-wise separable Convolution Block (RDCB) is introduced. Among the 19 convolution layers, 16 convolution layers are organized as 8 RDCB units (in which 4 units are located in the contracting path; 4 units are located in the expanding path), one-layer acts as the middle layer, and two layers are used to input and output the image. In addition to that, a down-sampling layer and dropout layer are added in the contracting path after the RDCB unit, and in expanding path up-sampling layer is added. Going deeper into the contracting path, the number of filters used in the RDCB unit is increased as 32, 64, 128, 256 which is lesser than the number of filters used in the modified U-Net network[7], and the reversed order is used in expanding path. The middle layer contains 512 numbers of filters.

Here, the output of each RDCB unit in the contracting path is integrated with the same level of RDCB unit in the expanding path through the concatenation unit. The concatenation unit, which combines the output of each contracting path RDCB unit with the feature map from the corresponding up-sampling layer, then is fed into the RDCB unit. Hence, the location information is integrated with contextual information to form generalized information which helps to obtain excellent segmentation accuracy.

Drop-channel is an effective method of dropout layer when compared to other dropout variants proposed in the work[23] which restrain the model from overfitting. In this work, the dropout layer is added at the RDCB unit level. It helps to reduce the error rate of the segmentation process significantly. The output

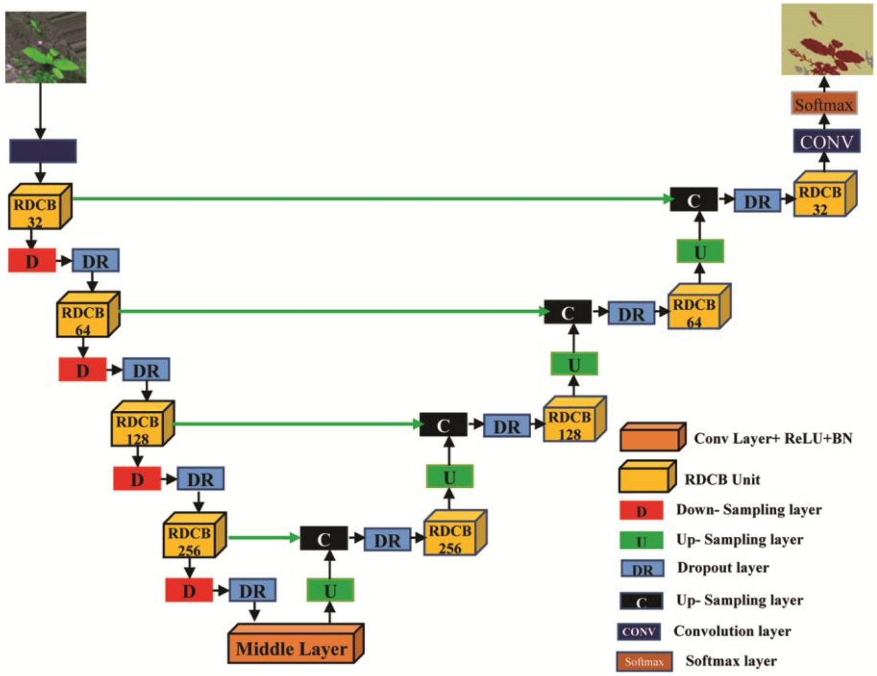

Fig. 4 — The structure of reduced residual U-Net using depth-wise separable convolution (RRUDC) model architecture

of the final RDCB unit on the decoder side is given to the softmax layer via the standard convolution layer which is used to find the class probabilities of each pixel in crop filed image and classify image portion pixel-wise. The detailed RRUDC architecture configuration is shown in Table 1.

Residual Depth-wise Separable Convolution Block (RDCB)

The structure of the RDCB unit is shown in Fig. 5. Each unit contains two convolution layers and one residual connection. Each convolution layer is followed by the ReLU activation layer and Batch Normalization (BN) layer to increase the segmentation accuracy and its outputs are represented by y_1 and y_2. Assuming x, y is the input and output of

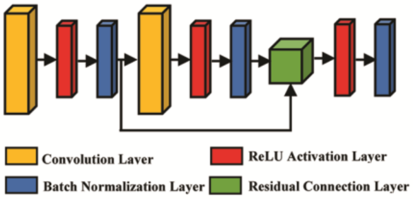

Fig. 5 — The structure of residual depth-wise separable convolution block (RDCB) unit

Table 1 — The configuration of the proposed RRUDC architecture

Block/ Unit Level	Layer	Filter size	No. of layers/ unit	No. of filters	Outputsize
Initial	Conv_IP	3 × 3	1	64	224 × 224 × 64
BLOCK C1	Conv_C1	3 × 3	2	32	224 × 224 × 32
	Rcon_C1	—	1	—	224 × 224 × 32
	Downsamp1	2 × 2	1	—	112 × 112 × 32
	Dropout_C1		(*dropout_rate = 20%)		112 × 112 × 32
BLOCK C2	Conv_C2	3 × 3	2	64	112 × 112 × 64
	Rcon_C2	—	1	—	112 × 112 × 64
	Downsamp2	2 × 2	1	—	56 × 56 × 64
	Dropout_C2		(*dropout_rate = 20%)		56 × 56 × 64
BLOCK C3	Conv_C3	3 × 3	2	128	56 × 56 × 128
	Rcon_C3	—	1	—	56 × 56 × 128
	Downsamp3	2×2	1	—	28 × 28 × 128
	Dropout_C3		(*dropout_rate = 50%)		28 × 28 × 128
BLOCK C4	Conv_C4	3 × 3	2	256	28 × 28 × 256
	Rcon_C4	—	1	—	28 × 28 × 256
	Downsamp4	2 × 2	1	—	14 × 14 × 256
	Dropout_C4		(*dropout_rate = 50%)		14 × 14 × 256
Middle Unit	Conv_MID	3 × 3	1	512	14 × 14 × 512
BLOCK E1	Up_Samp1	2 × 2	1	—	28 × 28 × 512
	Concat_1	—	1	—	28 × 28 × 768
	Dropout_E1		(*dropout_rate= 50%)		28 × 28 × 768
	Conv_D1	3 × 3	2	256	28 × 28 × 256
	Rcon_E1	—	1	—	28 × 28 × 256
BLOCK E2	Up_Samp2	2 × 2	1	—	56 × 56 × 256
	Concat_2	—	1	—	56 × 56 × 384
	Dropout_E2		(*dropout_rate= 50%)		56 × 56 × 384
	Conv_D2	3 × 3	2	128	56 × 56 × 128
	Rcon_E2	—	1	—	56 × 56 × 128
BLOCK E3	Up_Samp3	2 × 2	1	—	112 × 112 × 128
	Concat_3	—	1	—	112 × 112 × 192
	Dropout_E3		(*dropout_rate= 20%)		112 × 112 × 192
	Conv_3d1	3 × 3	2	64	112 × 112 × 64
	Rcon_E3	—	1	—	112 × 112 × 64
BLOCK E4	Up_Samp4	2 × 2	1	—	224 × 224 × 64
	Concat_4	—	1	—	224 × 224 × 96
	Dropout_E4		(*dropout_rate= 20%)		224 × 224 × 96
	Conv_4d1	3 × 3	2	32	224 × 224 × 32
	Rcon_E4	—	1	—	224 × 224 × 32
Output	Conv_OUT	3 × 3	1	3	224 × 224 × 3
	Softmax	—	1	—	224 × 224 × 3

an RDCB unit, the ReLU activation function, and Batch Normalization is represented by σ, convolution operation represented by M_{ci} and the function of the RDCB unit is represented by M_{RDCB} respectively. The mathematical form of y_1 and y_2 are mentioned in Eq. (2) and (3) respectively.

$$y_1 = \sigma\ (M_{c1}\ (x)) \qquad \ldots (2)$$

$$y_2 = \sigma\ (M_{c2}\ (y_1))$$
$$= \sigma\ (M_{c2}\ (\sigma\ (M_{c1}\ (x)))) \qquad \ldots (3)$$

The residual connection layer adds both outputs y_1 and y_2 and then fed into ReLU activation layer and BN layer whose output is represented in the Eq. (4).

$$y = M_{RDCB}\ (x)$$
$$= \sigma\ (y_{1\ +}\ y_2)$$
$$= \sigma\ (\sigma\ (M_{c1}\ (x)) + \sigma(M_{c2}\ (\sigma\ (M_{c1}\ (x))))) \qquad \ldots (4)$$

The residual connection path in the RDCB unit facilitates local residual learning to improve the information flow among architecture blocks. It will improve the representation capacity of the model and will curtail the error rate of the segmentation with the smaller dataset. So, the corner features of the crops and weeds can be predicted accurately.

Pixel-wise Labelling

Pixel-wise labelling is an effective method to analyze crop field images.[13] This helps the model to interpret the image pixel-wise and it facilitates to label each pixel to a certain class. Crop field images may have three categories of pixel components which are crop, weed, and soil. So, it is assumed as three classes (n =3) and its class values are assigned as {soil, crop, weed} = {0, 1, 2}. The model yields pixel-wise prediction P ∈ {0, 1, 2}. Because of this, each pixel in the image assigns to either 0-soil, 1-crop, or 2- weed. By this, weed portion in crop field can easily be detected and it can be either removed or sprayed agrochemical products on it.

Depth-wise Separable Convolution

In general, the model size increases along with this number of parameters if the number of parameters increases. Moreover, a model with a huge number of parameters requires more energy and space. Hence it is impractical to use it in resource-constrained devices that have been used by the farmers on agricultural land.

Convolution is a core operation of a Convolution Neural Network that is used to extract useful information from the image. It involves a huge number of different filters to extract different features from the image. All these extracted features are associated to give a better understanding of the image. But the problem with the standard convolution is that it is computationally expensive. The process of standard convolution is shown in Fig. 6a. The mathematical form for finding the number of multiplications (NM_C) and parameters (NP_C) consumed by the standard convolution is given in the following Eq. (5) and (6) respectively.

$$NM_C = S_k \times S_k \times N_{ic} \times N_{oc} \times S_f \times S_f \qquad \ldots(5)$$

$$NP_C = S_k \times S_k \times N_{ic} \times N_{oc} \qquad \ldots(6)$$

where, S_k and S_f represent the size of the kernel and feature map respectively, N_{ic} and N_{oc} represent the number of input and output channels respectively. Standard convolution is quite large because the size of each convolution filter used in the standard convolution is $S_k \times S_k \times N_{ic.}$

To overcome this, depth-wise separable convolution (DSC) is used instead of standard convolution. This DSC operation is performed in two phases which are depth-wise convolution and point-wise convolution.[20,22] The process of depth-wise separable convolution is shown in Fig. 6b. At first, depth-wise convolution filters whose size is $S_k \times S_k \times 1$ apply to the single input channel of an input image. It will be carried for entire input channels N_{ic} and then stacked together all the outputs to form intermediate feature map G. In point-wise convolution, the intermediate feature map G is convolved with N_{oc} number of filters whose size is $1 \times 1 \times N_{ic}$ to form the final feature map. Hence, it assures that features can be extracted from both spatial and channel levels. The

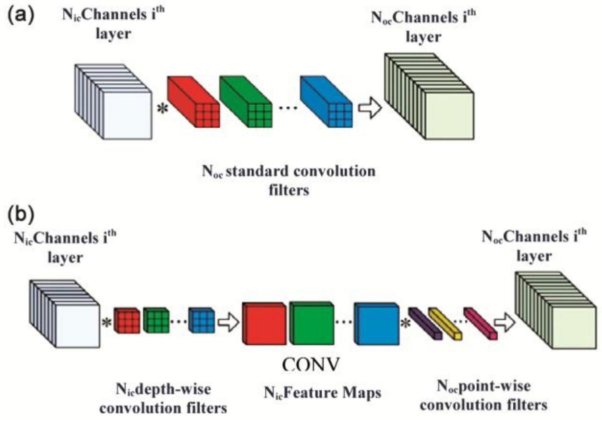

Fig. 6 — The operation of convolution process: (a) Standard Convolution technique, (b) Depth-wise Separable Convolution technique

number of multiplications (NM_{DSC}) and parameters (NP_{DSC}) consumed by the depth-wise separable convolution is given in the following Eqs (7) & (8) respectively.

$$NM_{DSC} = (S_k \times S_k \times N_{ic} \times S_f \times S_f) + (N_{ic} \times N_{oc} \times S_f \times S_f) \quad \dots (7)$$

$$NP_{DSC} = (S_k \times S_k \times N_{ic}) + (N_{ic} \times N_{oc}) \quad \dots (8)$$

The ratio between the standard convolution and depth-wise separable convolution regarding the number of multiplication and parameters are given below in Eqs (9) & (10) respectively. For example, assuming the filter size $S_k = 3$ and the number of input channels $N_{ic} = 64$, we obtain the ratio as 0.1267. This means that the parameters in depth-wise separable convolution are reduced to ~1/9 parameters when compared to that of standard convolution.

$$\frac{NM_{DSC}}{NM_C} = \frac{1}{N_{ic}} + \frac{1}{(S_k)^2} \quad \dots (9)$$

$$\frac{NP_{DSC}}{NP_C} = \frac{1}{N_{ic}} + \frac{1}{(S_k)^2} \quad \dots (10)$$

With the various improvements in the architecture, it can achieve ~88% reduction in parameters and with a very negligible drop in accuracy when compared with models proposed in the work.[21]

Pruning the Model

Usually, the machine/deep learning model is the core component of computer vision applications. It demands a lower size model to build a real-time computer vision application, for helping the farmers, since the target environment (agricultural land) of those applications has resource constraint devices. If the deep learning models are larger sized, they might be successful in achieving better performance in a lab setup, but it finds it hard to be deployed in resource-constrained devices. Even though it has been deployed, it fails to give better results. For this, the proposed deep learning model has been compressed using filter pruning.

After building the final model, the model will be carried to the training process till achieving better validation accuracy and test accuracy. Subsequently, the model will be exposed to the pruning process to reduce the size of the model.[25,22] During the pruning process, iteratively eliminate redundant and insignificant connections and their corresponding unnecessary lowest magnitude weights of the trained model. So the connection between the channels will become sparse. This pruning process achieves significant size reduction in the final model and it can be delivered by less than ~3 MB in size without significant recession in performance.

Data Preprocessing

The dataset used here is the benchmark dataset Crop/Weed Field Image Dataset CWFID).[26] This dataset contains 60 RGB images with their corresponding annotated images. All these images are captured in an organic carrot field. The success of building an efficient deep learning model requires a large number of images to train the model otherwise model is prone to overfit. Data augmentation is the efficient way to overcome this drawback which is used to create a new larger dataset from the existing smaller dataset by creating transformed versions of original images. This helps to upgrade the ability of the model to generalize and improve its performance. Initially, the existing 60 RGB images and annotated images are divided into a 5:1 ratio for the training and testing process. At this rate these images are processed for various augmentation techniques like rotation, horizontal flipping, shifting in height and width, zooming, and shearing which results in the creation of a new dataset containing 750 training images and 100 testing images, and every image is resized to 224 × 224 × 3.

Experimental Environmental Design

Experimental Setup

All the models utilized in this work are based on the Tensor Flow and Keras framework. These models have experimented on the online Google Colab platform which provides NVIDIA TeslaT4 GPU with CUDA Version 11.2, RAM of 13 GB, and Disk Space of 68 GB. Google Colab is a Google product that allows python code to be run on a cloud environment via a web browser, and hence there is no hardware restriction for it. Simple computers with an internet connection are capable of running it.

The proposed network is trained from the scratch using a dataset of 750 RGB images and its equivalent annotated images or labeled images; each image is a size of 224 × 224 × 3. For validation process, validation_split = 0.2 is used. i.e., 20% of the data in the training dataset is allotted for validation. In every epoch, 600 images are used for the training process and 150 images are used for the validation process. For the testing process, a separate test data set is built which contains 100 RGB images and their equivalent labeled images. The proposed system is trained with 80 training epochs with the batch size as 5 images per batch and each epoch contains 120 steps. The loss function used here is categorical_cross_entropy. Since, it has three classes which are crop, weed, and soil, this loss function is quite suitable. This measures the cross-entropy loss between the label and prediction. For this, labeled images are transformed into one-hot representations for facilitating the comparison between prediction and labels. During the training, the loss between the prediction and labels is optimized utilizing an ADAM optimizer with a learning rate of 0.001. It is a stochastic gradient descent method that depends on the adaptive estimation of first-order and second-order moments.

In the proposed model, for the top two RDCB units in the contractive path and expanding path it uses a lesser dropout rate which is 20%, and going deeper into the architecture dropout rate is increased to 50%. Sparsity rate is an important hyper-parameter of the pruning process which specifies the number of weights that have to be pruned away from the model. In the proposed work, the sparsity rate ranges from 50% to 90% in the pruning schedule. After pruning the model, two more training epochs were implemented for fine-tuning the model. The other significant deep learning architecture like SegNet, based on[13], U-Net, used in[7] and Residual U-Net based on[19] is implemented and its architecture configuration details are given in Tables 2–4. These architectures are trained with the same augmented CWFID dataset in the same experimental setup and the performance is evaluated for comparison.

Evaluation Metrics

In this paper, we have used six metrics to evaluate the performance of the model, which are

1. Accuracy (A),
2. Error-rate (E),
3. Overall Precision (OP),
4. Overall Recall (OR),
5. Overall F1 score (OF),
6. Number of Parameters (NoP)

Accuracy (A) states that the percent of the pixels in the prediction image is predicted correctly when compared to the label image. Error-rate (E) is expressed as the rate of pixels in the prediction image is predicted wrongly, when compared to label images. Overall Precision (OP) is the ratio of positive predictions which are actually precise. Overall Recall (OR) is the ratio of actual positive predictions which

Table 2 — The configuration of SegNet architecture

Layer	Kernel size	No. of layers	No. of filters	Output shape
Conv_IP	3 × 3	1	64	224 × 224 × 64
Conv_1E	3 × 3	2	64	224 × 224 × 64
Down_samp1	2 × 2	1	—	112 × 112 × 64
Conv _2E	3 × 3	2	128	112 × 112 × 128
Down_samp2	2×2	1	—	56 × 56 × 128
Conv _3E	3 × 3	2	256	56 × 56 × 256
Down_samp3	2 × 2	1	—	28 × 28 × 256
Conv _4E	3 × 3	2	256	28 × 28 × 256
Down_samp4	2 × 2	1	—	14 × 14 × 256
Conv _5E	3 × 3	2	512	14 × 14 × 512
Down_samp5	2 × 2	1	—	7 × 7 × 512
Up_samp1	2 × 2	1	—	14 × 14 × 512
Conv _1D	3 × 3	2	512	14 × 14 × 512
Up_samp2	2 × 2	1	—	28 × 28 × 512
Conv _2D	3 × 3	2	256	28 × 28 × 256
Up_samp3	2 × 2	1	—	56 × 56 × 256
Conv _3D	3 × 3	2	256	56 × 56 × 256
Up_samp4	2 × 2	1	—	112 × 112 × 256
Conv _4D	3 × 3	2	128	112 × 112 × 128
Up_samp5	2 × 2	1	—	224 × 224 × 128
Conv _5D	3 × 3	2	64	224 × 224 × 64
Conv_OUT	3 × 3	1	3	224 × 224 × 3
Softmax	—	—	—	224 × 224 × 3

Table 3 — The configuration of U-Net architecture

Layer	Kernel size	No. of layers	No. of filters	Output shape
Conv_IP	3 × 3	1	64	224 × 224 × 64
Conv _1C	3 × 3	2	64	224 × 224 × 64
Down_samp1	2 × 2	1	—	112 × 112 × 64
Conv _2C	3 × 3	2	128	112 × 112 × 128
Down_samp2	2 × 2	1	—	56 × 56 × 128
Conv _3C	3 × 3	2	256	56 × 56 × 256
Down_samp3	2 × 2	1	—	28 × 28 × 256
Conv _4C	3 × 3	1	512	28 × 28 × 512
Down_samp4	2 × 2	1	—	14 × 14 × 512
Conv _Mid	3 × 3	1	512	14 × 14 × 512
Up_samp1	2 × 2	1	—	28 × 28 × 512
Concat_1	—	—	—	28 × 28 × 1024
Conv _1E	3 × 3	2	512	28 × 28 × 512
Up_samp2	2 × 2	1	—	56 × 56 × 512
Concat_2	—	—	—	56 × 56 × 768
Conv _2E	3 × 3	2	256	56 × 56 × 256
Up_samp3	2 × 2	1	—	112 × 112 × 256
Concat_3	—	—	—	112 × 112 × 384
Conv _3E	3 × 3	2	128	112 × 112 × 128
Up_samp4	2 × 2	1	—	224 × 224 × 128
Concat_4	—	—	—	224 × 224 × 192
Conv _4E	3 × 3	2	64	224 × 224 × 64
Conv_OUT	3 × 3	1	3	224 × 224 × 3
Softmax	—	—	—	224 × 224 × 3

Table 4 — The configuration of Residual U-Net architecture

Layer	Kernel size	No. of layers	No. of filters	Output shape
Conv_IP	3 × 3	1	64	224 × 224 × 64
Conv _1C	3 × 3	2	64	224 × 224 × 64
R_ConC1	—	—	—	224 × 224 × 64
Down_sampl	2 × 2	1	—	112 × 112 × 64
Conv _2C	3 × 3	2	128	112 × 112 × 128
R_ConC2	—	—	—	112 × 112 × 128
Down_sampl	2 × 2	1	—	56 × 56 × 128
Conv _3C	3 × 3	2	256	56 × 56 × 256
R_ConC3	—	—	—	56 × 56 × 256
Down_sampl	2 × 2	1	—	28 × 28 × 256
Conv _4C	3 × 3	1	512	28 × 28 × 512
R_ConC4				28 × 28 × 512
Down_sampl	2 × 2	1	—	14 × 14 × 512
Conv _Mid	3 × 3	1	512	14 × 14 × 512
Up_samp1	2 × 2	1	—	28 × 28 × 512
Concat_1	—	—	—	28 × 28 × 1024
Conv _1E	3 × 3	2	512	28 × 28 × 512
R_ConE1	—	—	—	28 × 28 × 512
Up_samp2	2 × 2	1	—	56 × 56 × 512
Concat_2	—	—	—	56 × 56 × 768
Conv _2E	3 × 3	2	256	56 × 56 × 256
R_ConE2	—	—	—	56 × 56 × 256
Up_samp3	2 × 2	1	—	112 × 112 × 256
Concat_3	—	—	—	112 × 112 × 384
Conv _3E	3 × 3	2	128	112 × 112 × 128
R_ConE3	—	—	—	112 × 112 × 128
Up_samp4	2 × 2	1	—	224 × 224 × 128
Concat_4	—	—	—	224 × 224 × 192
Conv _4E	3 × 3	2	64	224 × 224 × 64
R_ConE4	—	—	—	224 × 224 × 64
Conv_OUT	3 × 3	1	3	224 × 224 × 3
Softmax	—	—	—	224 × 224 × 3

are identified precisely. Overall F1 score is a metric that considers both OP and OR and it is a harmonic mean of OP and OR. The mathematical forms of the above metrics are mentioned in Eqs 11–15, respectively. The Number of Parameters (NoP) generated by the model is a total number of the learnable elements of all the convolution layers in CNN and its mathematical form is mentioned in Eqs 6 & 8 for standard convolution and depth-wise separable convolution respectively.

$$Accuracy\ (A) = \frac{(OTP+OTN)}{(OTP+OTN+OFP+OFN)} \quad \dots (11)$$

$$Error\text{-}rate\ (E) \ = -Y_j * log(P_j) \quad \dots (12)$$

$$Overall\ Precision\ (OP)\ = \frac{OTP}{OTP+OFP} \quad \dots (13)$$

$$Overall\ Recall\ (OR)\ = \frac{OTP}{OTP+OFN} \quad \dots (14)$$

Table 5 — Training parameters

Parameter	Values
Batch size	5
Steps per epoch	120
Leaning rate	0.001
Optimizer	ADAM optimizer
Loss function	categorical cross_entropy

$$Overal\ F1 = 2 \times \frac{OP \times OR}{OP+OR} \quad \dots (15)$$

where, OTP is Overall True Positive, OTN is Overall True Negative, OFP is Overall False Positive, OFN is Overall False Negative, t is the total number of samples, P_j represents the j-th predicted output of the model, and Y_j represents the corresponding j-th target value.

Results and Discussion

At first, the deep learning architectures SegNet based on[13], U-Net based on[7], Residual U-Net based on[19], and proposed RRUSC (Reduced Residual U-Net using Standard Convolution) networks are implemented with a standard convolution technique. These models are trained using the training parameters mentioned in Table 5. The corresponding results of these models with the Standard convolution technique are recorded in Table 6. The metrics of the proposed architecture RRUSC are considerably good, achieving a segmentation accuracy of 96.14% at validation time and 95.37% at testing time, which is higher than others. The overall F1 score achieved by the proposed RRUSC is also good and better than other models except the Residual U-Net model, which are 97.30% and 97.29% at validation and testing time, respectively. But while considering the number of parameters of the RRUSC model, it is very huge i.e., 5.52 Million even though it is lesser than other models. So, it is computationally very expensive since it is implemented using a standard convolution technique.

To overcome this, all those architectures listed in Table 6 are implemented with depth-wise separable convolution (DSC) instead of standard convolution techniques. The corresponding results of the models with depth-wise separable convolution (DSC) are recorded in Table 7. The parameters comparison of standard convolution and DSC version of different architectures is visualized in Fig. 7. The DSC version of the proposed model, called RRUDC consumes 0.655 million parameters only which was 88% lesser than the standard convolution version. The proposed

Table 6 — Comparison of architecture variants with standard convolution

Architecture Variants	NoP (in Millions)	Validation					Testing				
		A (%)	E (%)	OP (%)	OR (%)	OF (%)	A (%)	E (%)	OP (%)	OR (%)	OF (%)
SegNet[13]	14.145	93.06	11.54	93.46	96.32	94.87	92.77	12.07	93.45	96.31	94.86
U-Net[7]	12.552	95.06	4.14	95.62	99.05	97.30	95.41	8.09	95.62	99.05	97.30
Residual U-Net[19]	12.556	95.78	5.62	95.59	99.22	97.37	95.21	10.07	95.58	99.22	97.36
RRUSC	5.520	96.14	2.89	95.57	99.09	97.30	95.37	7.45	95.57	99.08	97.29

NoP = No of Parameters, A = Accuracy, E = Error rate, OP = Overall precision, OR = Overall, OF = Overall F1 score

Table 7— Comparison of architecture variants with DSC

Architecture Variants	NoP (in Millions)	Validation					Testing				
		A (%)	E (%)	OP (%)	OR (%)	OF (%)	A (%)	E (%)	OP (%)	OR (%)	OF (%)
SegNet-DSC	1.639	92.45	14.74	93.68	96.73	95.18	92.29	14.88	93.67	96.72	95.17
U-Net- DSC	1.450	95.59	6.49	95.69	99.35	97.49	95.23	8.84	94.77	98.60	96.55
Residual U-Net DSC	1.453	95.6	5.20	95.71	99.33	97.49	95.18	8.39	95.70	99.34	97.48
RRUDC	0.655	96.13	2.51	95.43	98.83	97.10	95.40	6.42	95.43	98.82	97.10

NoP = No. of parameters, A = Accuracy, E = Error rate, OP = Overall precision, OR = Overall recall, OF = Overall F1 score

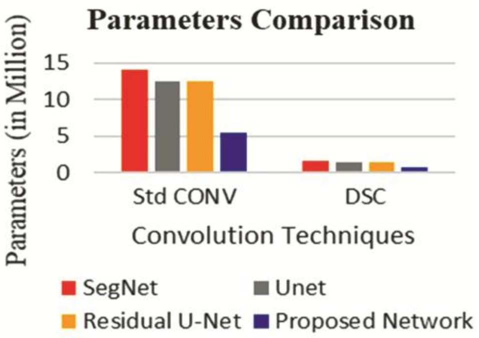

Fig. 7 — Parameters comparison chart between Standard Convolution and DSC

RRUDC architecture's accuracy and error- rate over the epochs during the training and validation phase respectively are shown in Figs 8 and 9.

The proposed RRUDC achieves segmentation accuracy of 96.13% at validation and 95.40% at testing, Overall F1 Score (OF) of 97.10 at both validation and testing time. The drop in accuracy and OF values between the Stand convolution version and depth-wise convolution version of the proposed models are very negligible range when compared with parameter reduction. But the error-rate of the RRUDC is reduced when compared to RRUSC, which is 2.51% at validation and 6.42% at testing. The learning curves of accuracy and error- rate over the epochs for the DSC version of SegNet, U-Net, Residual U-Net, and RRUDC architecture are visualized in Figs 10 and 11 respectively. Hence, the RRUDC attains a better trade-off among the metrics of accuracy, error rate, and parameters consumption.

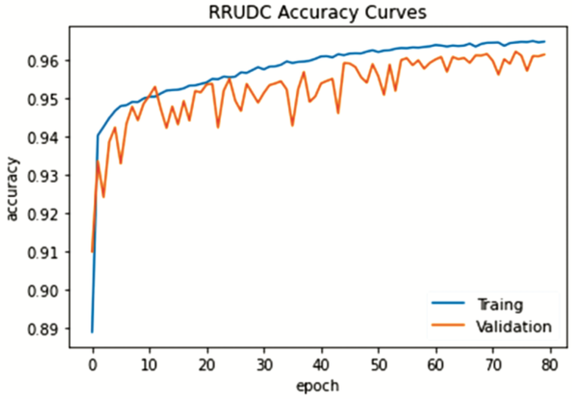

Fig. 8 — Accuracy in RRUDC architecture over the epochs

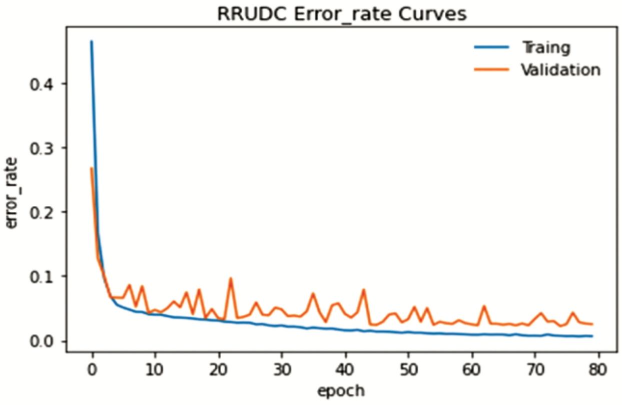

Fig. 9 — Error-rate in RRUDC architecture over the epochs

The OP, OR, and OF values of RRUDC are 95.43%, 98.83%, and 97.10% respectively. These are lesser than U-Net and Residual U-Net but their difference is insignificant.

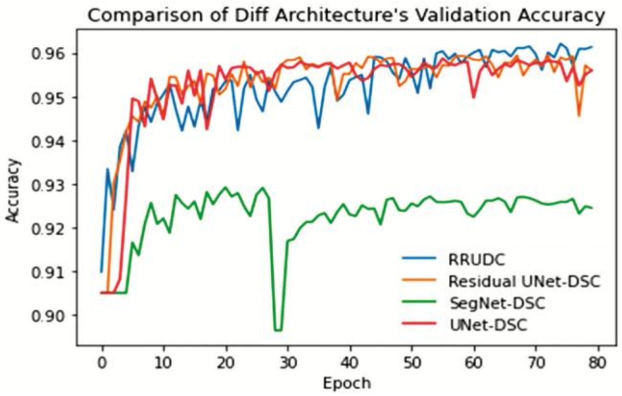

Fig. 10 — Validation accuracy comparison over the epochs

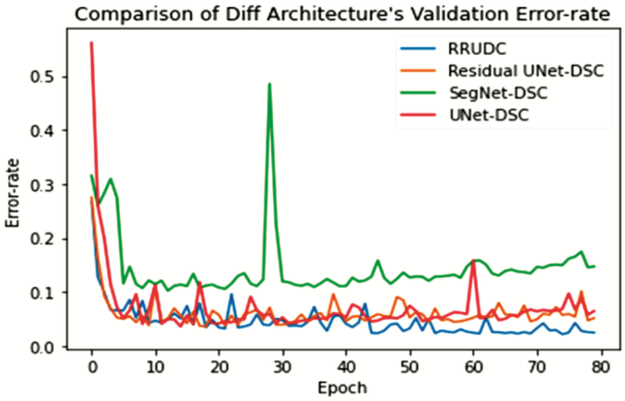

Fig. 11 — Error-rate comparison over the epochs

The sample segmentation images of SegNet_DSC, U-Net_DSC, Residual U-Net_DSC, and proposed RRUDC are shown in Fig. 1. In which, the RRUDC segmentation result (Fig. 1f) extracts the corner features of crop and weed precisely. For further reference, proposed RRUDC segmentation images of a few more crop-field images from the CWFID dataset are shown in Fig. 2. This shows that RRUDC architecture outperformed other architectures in segmentation. After achieving substantial accuracy, the trained model is exposed to the pruning process. The scarsity rate is applied from 50% to 90% in the pruning schedule. After the pruning process, dense connections between the channels become sparse. Hence, the model size is reduced from 7.83 MB to 2.34 MB, since the insignificant weight values are removed from the tensor. The model size and its metrics are recorded in Table 8 and a comparison of the architecture's model size before and after pruning is visualized in Fig. 12.

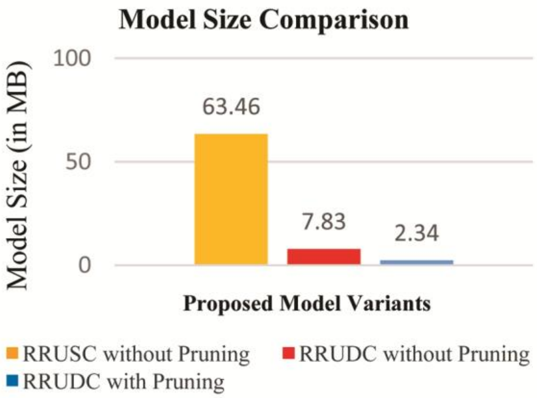

Fig. 12 — Comparison of model size RRUSC Vs RRUDC before and after pruning

Table 8 — Comparison of pruned model and standard model

Model	Model Size (in MB)	Validation		Testing	
		A (%)	E (%)	A (%)	E (%)
RRUSC without Pruning	63.46	96.14	2.89	95.37	7.45
RRUDC without Pruning	7.83	96.13	2.51	95.40	6.42
RRUDC with Pruning	2.34	96.06	2.78	95.44	7.12

A = Accuracy, E = Error rate

It shows that the pruned RRUDC model achieved a better model size reduction with a negligible drop in accuracy and a negligible increase in error rate. Based on the above experiments, the proposed RRUDC architecture outperformed well on the segmentation task when compared to SegNet, U-Net, and Residual U-Net. It achieved better accuracy and lesser error-rate-based, with a lesser number of parameters and reduced model size.

Comparison with Existing work

As part of the crop-weed segmentation task on the CWFID dataset, Hashemi-Beni *et al.*[27]employed FCN-8s, FCN-16s, FCN-32s, and U-Net deep learning models, and McCool *et al.*[28] developed DeepLab v3+ and AgNet deep learning models on their work. The performance comparison of the proposed RRUDC model with the existing work is presented in Table 9. From Table 9, the proposed Pruned RRUDC achieved a better segmentation accuracy of 96.06%, when compared to that of the existing models. In terms of the Number of Parameters (NoP), AgNet[28] generated a very less Number of Parameters, 0.25 Million, when compared to the other models. But on the other side, the parameter reduction was affected its segmentation accuracy, which is dropped to 88.9%. Even the NoP value of the proposed RRUDC is a little higher (0.655

Table 9 — Comparison with existing work

Model Name	A (%)	NoP (Million)	Model Size (MB)
FCN-8s[27]	81.1	—	—
FCN-16s[27]	77.2	—	—
FCN-32s[27]	68.4	—	—
U-Net[27]	77.9	—	—
DeepLab v3+[27]	84.3	—	—
Adapted-IV3[28]	93.9	25	—
AgNet[28]	88.9	0.25	—
Proposed Pruned RRUDC	96.06	0.655	2.34

A = Accuracy, NoP = Number of Parameters
— Not given in the paper

Million) than AgNet (0.25 Million), it significantly improved the segmentation accuracy of crop-weed segmentation. Further, the proposed RRUDC model is pruned for model size reduction. Thus Pruned RRUDC model is come up with a size of 2.34 MB, which can be convenient to deploy on resource-constrained devices used by the farmers.

Conclusions

In this paper, we proposed a Reduced Residual U-Net using Depth-wise separable Convolution (RRUDC) network for crop-weed segmentation. This is Encoder-Decoder based architecture. Residual Depth-wise separable Convolution Block (RDCB) was introduced as a functional unit in which standard convolution technique is replaced with Depth-wise Separable Convolution technique (DSC). With this, the number of parameters is reduced to 12% of parameters used with the standard convolution technique. A dropout layer is added before every RDCB unit which prevents the model from overfitting. Both residual connection and dropout layers help to reduce the segmentation error-rate.

The concatenation unit present in this architecture is used to integrate the location information with contextual information to achieve the precise segmentation of crops and weeds. Further using the pruning technique, the model size is compressed to 30% when compared to the original network without significant degradation of performance. Thus, our proposed deep learning-based RRUDC model outperformed well regarding segmentation results with good accuracy and less error-rate, with reduced model size, 2.34 MB, which is being more convenient to build a computer vision application that can be easily deployed on resource-constrained portable devices in agricultural land.

Future Work

In future work, we shall be planning to build a deep learning-based multi-crop weed segmentation model for segmenting weeds present in the different types of crops in the agricultural land. With this multi-crop weed detection model, we shall be planning to develop a simple and convenient mobile application to help the farmers to identify the presence of weeds on their agricultural land.

References

1 Lottes P, Behley J, Milioto A & Stachniss C, Fully convolutional networks with sequential information for robust crop and weed detection in precision farming, *IEEE Robot Autom Lett*, **3(4)** (2018) 2870–2877.

2 Sampathkumar S & Rajeswari R, An automated crop and plant disease identification scheme using cognitive fuzzy C-means algorithm, *IETE J Res*, (2020) 1–12, doi: 10.1080/03772063.2020.1780163

3 Lottes P, Khanna R, Pfeifer J, Siegwart R, & Stachniss C, UAV-based crop and weed classification for smart farming, *IEEE Int Conf Robotics Autom (ICRA)* 2017, 3024–3031.

4 Shelhamer E, Long J & Darrell T, Fully convolutional networks for semantic segmentation, *IEEE Trans Pattern Anal Mach Intell*, **39(4)** (2016) 640–651.

5 Badrinarayanan V, Handa A & Cipolla R, Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling, *arXiv* (2015) arXiv preprint arXiv:1505.07293.

6 Noh H, Hong S & Han B, Learning deconvolution network for semantic segmentation, *Proc IEEE Int Conf Computer Vision*, 2015, 1520–1528.

7 Xie H X, Lin C Y, Zheng H & Lin P Y, An Unet-based head shoulder segmentation network, *Int Conf Consumer Electronics-Taiwan* (ICCE-TW) 2018, 1–2.

8 Ronneberger O, Fischer P & Brox T, U-net: Convolutional networks for biomedical image segmentation, *Int Conf Medical Image Comput Computer-Assist Interv*, (Springer, Cham) 2015, 234–241.

9 Li R, Liu W, Yang L, Sun S, Hu W, Zhang F & Li W, DeepUNet: A deep fully convolutional network for pixel-level sea-land segmentation, *IEEE J Sel Top Appl Earth Obse Remote Sens*, **11(11)** (2018) 3954–3962.

10 Naqvi R A, Lee S W & Loh W K, Ocular-net: Lite-residual encoder-decoder network for accurate ocular regions segmentation in various sensor images, *IEEE Int Conf Big Data Smart Comput (BigComp)* 2020, 121–124.

11 Chollet F, Xception: Deep learning with depthwise separable convolutions, *Proc IEEE Conf Computer Vision Pattern Recognit*, 2017, 1251–1258.

12 Huang Q, Zhou K, You S & Neumann U, Learning to prune filters in convolutional neural networks, *IEEE Winter Conf Appl Comput Vision (WACV)* 2018, 709–718.

13 Badrinarayanan V, Kendall A & Cipolla R, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, *IEEE Trans Pattern Anal Mach Intell*, **39(12)** (2017) 2481–2495.

14 Yasrab R, Gu N & Zhang X, An encoder-decoder based convolution neural network (CNN) for future advanced driver assistance system (ADAS), *Appl Sci*, **7(4)** (2017) 312.

15 Lottes P, Hoeferlin M, Sander S, Müter M, Schulze P & Stachniss L C, An effective classification system for separating sugar beets and weeds for precision farming applications, *IEEE Int Conf on Robot Automat (ICRA)* 2016, 5157–5163.

16 Umamaheswari S & Jain A V, Encoder–decoder architecture for crop-weed classification using pixel-wise labelling, *Int Conf Artif Intell Signal Process (AISP)* 2020, 1–6.

17 Sa I, Chen Z, Popović M, Khanna R, Liebisch F, Nieto J & Siegwart R, Weednet: Dense semantic weed classification using multispectral images and mav for smart farming, *IEEE Robot Autom Lett*, **3(1)** (2017) 588–595.

18 Augustaukas R & Lipnickas A, Pixel-wise road pavement defects detection using U-net deep neural network, *10th IEEE Int Conf Intel Data Acquisit Adv Comput Syst: Technol Appl (IDAACS)* 2019, 468–471, doi: 10.1109/IDAACS.2019.8924337.

19 Yang X, Li X, Ye Y, Zhang X, Zhang H, Huang X & Zhang B, Road detection via deep residual dense u-net, *Int Joint Conf Neural Networks (IJCNN)* 2019, 1–7.

20 Chen Y, Peng F, Kang X & Wang Z J, Depthwise separable convolutional neural network for image forensics, *IEEE Visual Commun Image Process (VCIP)* 2019, 1–4, doi: 10.1109/VCIP47243.2019.8965671.

21 Santos A G, de Souza C O, Zanchettin C, Macedo D, Oliveira A L & Ludermir T, Reducing squeezenet storage size with depth wise separable convolutions, *Int Joint Conf Neural Networks (IJCNN)*, 2018, 1–6.

22 Mao Y, He Z, Ma Z, Tang X & Wang Z, Efficient convolution neural networks for object tracking using separable convolution and filter pruning, *IEEE Access*, **7** (2019), 106466–106474.

23 Cai S, Shu Y, Chen G, Ooi B C, Wang W & Zhang M, Effective and efficient dropout for deep convolutional neural networks, *arXiv* (2019) arXiv:1904.03392.

24 Simonyan K & Zisserman A, Very deep convolutional networks for large-scale image recognition, *arXiv* (2014), arXiv:1409.1556.

25 Gaikwad A S & El-Sharkawy M, Pruning convolution neural network (squeezenet) using taylor expansion-based criterion, *IEEE Int Symp Signal Process Informat Technol (ISSPIT)*, 2018, 1–5.

26 Haug S & Ostermann J, A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks. *European Conf Computer vision*, (Springer, Cham) 2014, 105–116.

27 Hashemi-Beni L, Gebrehiwot A, Karimoddini A, Shahbazi A & Dorbu F, Deep convolutional neural networks for weeds and crops discrimination from UAS imagery, *Front Remote Sens*, **3: 755939** (2022), *doi:* 10.3389/frsen.2022.755939.

28 McCool C, Perez T & Upcroft B, Mixtures of lightweight deep convolutional neural networks: Applied to agricultural robotics, *IEEE Robot Autom Lett*, **2(3)** (2017) 344–351