Weed Recognition System for Low-Land Rice Precision Farming Using Deep Learning Approach

Olayemi Mikail Olaniyi, Emmanuel Daniya, Ibrahim Mohammed Abdullahi, Jibril Abdullahi Bala, and Esther Ayobami Olanrewaju

Abstract Precision farming helps to achieve maintainable agriculture, with an objective of boosting agricultural products with minimal negative impact on the environment. This paper outlines a deep learning approach based on Single Shot multibox Detector (SSD) to classify and locate weeds in low-land rice precision farming. This approach is designed for post-emergence application of herbicide for weed control in lowland rice fields. The SSD uses VGG-16 deep learning-based network architecture to extract a feature map. The adoption of multiscale features and convolution filter enables the algorithm to have a considerable high accuracy even at varying resolutions. Using SSD to train the weed recognition model, an entire system accuracy of 86% was recorded. The algorithm also has a system sensitivity of 93% and a precision value of 84%. The trained SSD model had an accuracy of 99% for close-up high definition images. The results of the system performance evaluation showed that the trained model could be adopted on a real rice farm to help reduce herbicide wastage and improve rice production with low chemical usage.

Keywords Precision agriculture · Deep learning algorithm · SSD · Google TensorFlow · Low-land rice

O. M. Olaniyi (✉) · I. M. Abdullahi · E. A. Olanrewaju
Department of Computer Engineering, Federal University of Technology, Minna, Nigeria
e-mail: mikail.olaniyi@futminna.edu.ng

I. M. Abdullahi
e-mail: amibrahim@futminna.edu.ng

E. A. Olanrewaju
e-mail: esteey22@gmail.com

E. Daniya
Department of Crop Production, Federal University of Technology, Minna, Nigeria
e-mail: emma.daniya@futminna.edu.ng

J. A. Bala
Department of Mechatronics Engineering, Federal University of Technology, Minna, Nigeria
e-mail: jibril.bala@futminna.edu.ng

T. Masrour et al. (eds.), *Artificial Intelligence and Industrial Applications*,
Advances in Intelligent Systems and Computing 1193,
https://doi.org/10.1007/978-3-030-51186-9_27

1 Introduction

Agriculture is an important sector of any society since every individual relies on food for survival [1]. Precision agricultural practices help to achieve sustainable agriculture and improve production yield [2, 3]. This technique shows a huge tendency to lessen the number of chemicals used on farms to kill weeds. The concept of precision farming helps apply inputs at the appropriate time, at an appropriate place, in the correct amount, and in a proper manner, all which are targeted for improved crop production. Hand weeding and use of hoes and cutlasses are manual means of weed control on the farms which although, environmentally friendly, result in a high cost of labourer acquisition and are very tedious in nature [4–6]. Herbicide usage dominates the chemical method of weed control and is seen as an alternative approach to weed control [7, 8]. However, researchers have frowned upon uniform spraying and health concerns have been raised on the safety of the crops when consumed [5, 9, 10]. It is obvious that accurate information is needed to identify weeds from crops in order to avoid damage to the crop itself. Therefore, the need arises for a weed control technique that can obtain and analyse information about weeds from the farm, as well as act accordingly to control the weeds on the farm. Watchareeruetai et al. [11] suggested that automated weed control is a suitable technique for weed management by adopting the advance in technology as a tool. Similarly, [12] stated that computer vision abilities help to identify and classify weeds swiftly than manual techniques. Faridi and Aboonajmi [13] and Liakos et al. [14] also suggested that application of machine learning approach directly on farm would help reduce farmer's stress, manage cost, provide flexibility, accuracy, and improve quality of crop production. Due to the development in the computer vision field, it is easier to use automated techniques to acquire information, process them and then utilize the information to separate crops from the weeds [12]. The system computations are based on geometric characteristics such as aspect ratio, shape factor, area or length of the weeds and crops [15, 16]. Colour images are also used to establish the quality of seeds, distinguish weeds from crops, classify plant diseases and other damages that could be made to plants [5, 17]. Textural features also form a powerful discriminate for weeds and crops [18].

In low-land rice production, one of the major ways of weed control is the early application of herbicides. The application which [19] suggested should be done within 40 to 50 days of rice planting. These approaches discussed above addressed broadleaved weeds for upland crops which are not suitable to be used in low-land rice production. Rice is a grass species, whose weeds often looks similar to the rice crop itself and the area of low-land is a water-logged area. These factors make existing methods insufficient to address the weeds in that area. As a result of this shortcoming, this research follows an object detection approach based on a deep neural network algorithm called Single Shot multi-box Detector (SSD). The SSD is one of the fastest algorithms for object detection and it is known for achieving good result on targeted object [20]. The SSD which is based on VGG-16 architecture helps to differentiate weeds from rice of low-land rice field using multiscale features and convolution filters. The remaining of the paper is organised as follows: Sect. 2 presents an overview of the SSD model used while the image recognition process is presented in Sect. 3. Section 4 shows the result obtained from the simulation of the process and Sect. 5 presents the conclusion made and new research directions.

2 Review: Automated Weed Control and Object Detection Techniques

Several research works have been reported on intelligent and automated weed control [5, 10, 18, 21, 22, 23] and real-time object detection techniques [20–28]. In this paper, the review of intelligent and automated techniques is reported in Sect. 2.1, while the review of real-time object detection techniques such as the SSD are reported in Sect. 2.2.

2.1 Automated and Intelligent Weed Control

Several works exist in the area of automated and intelligent weed control. [21] used fast Fourier transform to extract the shape of weeds in the cornfield. The shape features were extracted and used as a discriminate between the weeds and the crop of the cornfield. Similarly, [29] used morphological thresholding techniques, erosion and dilation to differentiate between weed and plants. In addition, [18] adopted the use of textural features to distinguish weeds from vegetables under varying lighting. The researcher used Gray Level Co-occurrence Matrix to calculate textural features which form the feature map. The map was then fed to into a Support Vector Machine algorithm for weed classification. Other approaches have used colour features to separate weeds from crops, by adopting separation of vegetative plants from soil as seen in [5, 10, 22, 23].

2.2 Real-Time Object Detection: SSD Model

Single Shot Detector (SSD) is one of the fastest algorithms for real-time object detection [20]. The network was targeted at solving the problem of region proposed network of faster region with Convolution Neural Network (R-CNN) by using multi-scale features to extract features map and convolutional features for detection [24]. The SSD model consists of two parts, extraction of features map using multiscale features and application of convolution filters to detect objects. In SSD, localization and classification are done once by running the algorithm through the image in one forward pass to give a prediction which consists of an image with a bounding box

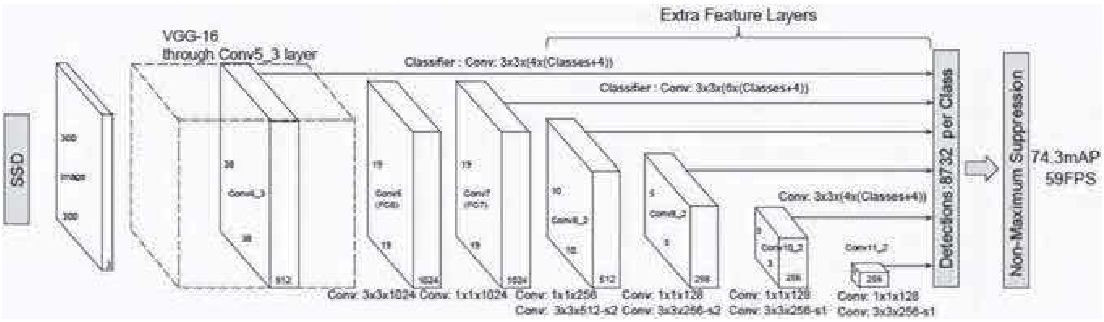

Fig. 1 Network architecture of SSD [24]

and the confidence score of detection. The SSD goes through matching strategies, hard negative limiting, data augmentation and non-maximum suppression to remove prediction duplicates. The network architecture as shown in Fig. 1 is based on VGG-16 architecture which has 16 layers. The figure shows the convolution stages the image goes through so that the features of images can be extracted. Other network architectures are ResNet-101, Inception V2, Inception V3, Inception, ResNet and MobileNet [25, 26–28]. VGG-16 is reported to have high performance on image classification tasks and has improved results on problems that involve using transfer learning.

Rather than use the original fully connected VGG layers, six supplementary layers are added to permit feature extraction at multiple scales and gradually reduce the input size at each successive layer. SSD runs the convolutional network on the image once to get the features map, the features map is then used to predict the bounding box and get the confidence level of the classifier. SSD can learn at different scales using the varying size of filters. The convolution layer acts and operates at different scales to enable it to locate the images at a fast speed. The box drawn has four parameters, namely: the centre, width, height and the vector of probability also known as the confidence level of the classifier. In SSD, predictions are classified as positive matches or negative matches depending on the value obtained for Intersection Over Union (IOU). IOU also known as Jaccard index compares ratio of area of overlap to area of union. Prior which are bounding box of fixed size with close match to distribution of original ground truth are selected in such a way that their IOU value is greater than 0.5. The multibox starts with the prior as initials prediction and starts to regress towards the ground truth boxes. However negative IOU are discarded. The objective loss function of the model is given as the weighted sum of localization loss (loc) and confidence (conf)

$$\mathrm{L(x, c, lg)} = \frac{1}{N}\left(L_{conf}(x, c) + \propto L_{loc}(x, l, g)\right) \quad (1)$$

Alpha ($\propto$) is a weight term which is set to 1 for cross-validation of the model, with a key reason for balancing the influence of the location loss. The criteria for choosing $\propto$ is to select an optimal value that lessen the value of the localization loss and brings the prediction closer to the ground truth. N is the number of matched default boxes and if N = 0 loss is set to 0. The localization loss is the smooth L1 loss between the predicted box and ground truth. The model is regressed to the coordinate of the centre of x and the coordinate of the centre of y (cx, cy) of the bounding box(d). The localization loss tells the mismatch between the ground truth box and the predicted box. The positive match is worked with while the negative match is ignored. The localization loss L_{loc} is given as:

$$L_{loc}(x,\ l,\ g) = \sum\nolimits_{i \in pos m \in \{cx,cy,w,h\}}^{N} \sum x_{ij}^{k}\ smoothL1\left(l_i^m - g_j^m\right) \tag{2}$$

$$g_j^{cx} = \left(g_j^{vcx} - d_i^{cx}\right) / d_i^w \tag{3}$$

$$g_j^{cy} = \left(g_j^{vcy} - d_i^{cy}\right)/d_i^h \tag{4}$$

$$g_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \tag{5}$$

$$g_j^h = \log\left(\frac{g_j^h}{d_i^h}\right) \tag{6}$$

Confidence loss is a SoftMax loss over multiple classes. This loss is used to penalise loss according to the confidence score of the corresponding class for every positive match prediction, while in negative case “class 0” is used which shows no object was detected. Confidence loss is given as:

$$L_{conf}(x, c) = -\sum\nolimits_{i \in pos}^{N} x_{ij}^{k} \log\left(c_i^p\right) - \sum\nolimits_{i \in Neg} \log\left(c_i^o\right) \tag{7}$$

where $c_i^o = \frac{\sum c_i^p}{\sum_p \exp\left(c_i^p\right)}$

The scale of the default boxes for each feature map is given as:

$$S_k = S_{min} + \frac{S_{max} - S_{min}}{m - 1}(k - 1) k \epsilon [1, m] \tag{8}$$

Since a lot of negative matches are made during prediction, due to the fact that we make more predictions than classes of objects; an imbalance is therefore created which has a negative influence on the training process. The process, known as hard negative mining, helps the SSD to use negative sampling to know what makes up a bad sampling. SSD then picks up the top loss and makes the ratio between the negative and positive to be at almost 3:1, this then leads to faster and stable training. The model accuracy is improved by using data augmentation process. Data is augmented using either original image, sampling of a patch (with values of 0.1, 0.3, 0.5, 0.7 or 0.9), or randomly sample of a patch. This part is needed to handle colour distortion and image variant sizes. The last step of the SSD model is non-max suppression and

i. *Initialize VGG 16 network*
ii. *Add auxiliary convolutional layer for feature extraction*
iii. *Reduce object dimension using bounding box regression technique*
iv. *Determine the model objective loss using multibox localization*
v. *Select the multi-box prior at an IOU >0.5*
vi. *Compute class prediction for each predicted bounding box*
vii. *Train model to learn wrong prediction using hard negative mining and learn the image at different IOU at different angles using data augmentation*
viii. *Remove noisy bounding box and ensure most likely predictions are left in the network using non maximum suppression*

Fig. 2 SSD implementation steps

it helps to remove duplicate predictions of the same image by sorting the predictions via their confidence loss. Figure 2 shows the SSD algorithm.

3 Weed Recognition System Development

This section describes the methods used for the development and deployment of the weed recognition system. The weed recognition process consists of image acquisition, labelling of image (data preparation), splitting images into testing and training, generation of TensorFlow record, setting up of model configuration files, training of the selected SSD model, exporting of the frozen inference graph and using of the frozen inference graph to classify new images. The summary of the processes is shown in Fig. 3.

3.1 Image Acquisition into Database

Images of seventeen (17) different types of low-land weed species of rice were obtained from the teaching and research farm, Department of Crop Production, Federal University of Technology (FUT), Minna, Niger State, Nigeria between 18th May, 2018 to 20th August, 2018. Also, images were obtained from the Google server about the same time. Images from the FUT teaching and research farm were taken with a 13 Megapixel Tecno Camon CX air camera with a resolution of 4160 X 3120 pixels and a focus of 0.15 s. The images were taken at different time intervals of rice growth, namely: one week, two weeks, one month and two months. A total of 3866 images were acquired into the database for the recognition process. Figure 4 shows part of the images acquired from the at the FUT teaching and research farm while

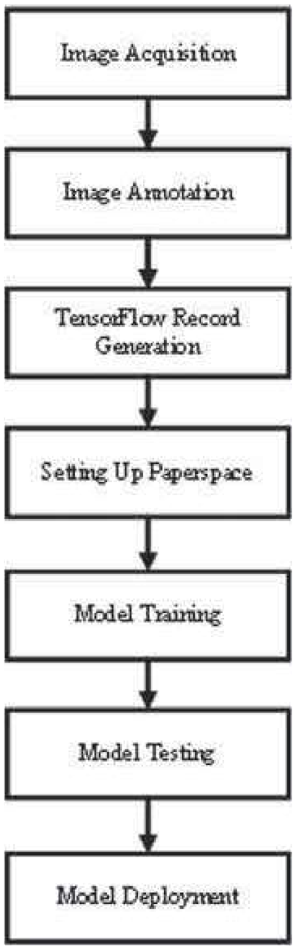

Fig. 3 Image recognition process

Fig. 4 Image acquired from the Cereal and Legume Crops practical plot, of teaching and research farm Department of Crop Production, Federal University of Technology, Minna, Nigeria for data preparation

Fig. 5 shows samples of the images acquired from the Google Server. Figure 6 shows the process chart for the image acquisition process.

Fig. 5 Sample images of weed obtained from Google server

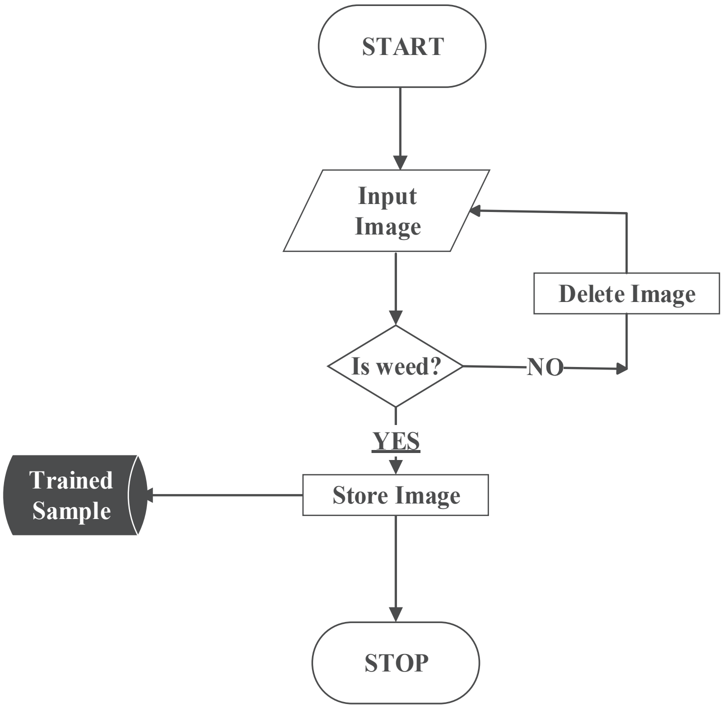

Fig. 6 Process chart of image acquisition and storage system

3.2 Image Annotation

After obtaining a dataset of the images, a process called data preparation was carried out which is a process of putting the data into a suitable form which makes it easier for use. The images of weeds obtained from Google server were resized to a size of 250 × 250 pixels using adobe Photoshop to minimise the size of the images and the memory requirements. A graphical tool for data preparation called ‘Label Image’, was used to draw a rectangular bounding box around the weeds in the image and the boxes were labelled as ‘weed’. This process is called annotation. The essence of the annotation is to draw the bounding box around the image so that the object of interest (weed) can be separated from the background and other objects in the image. The labelling shows the class the object belongs to while the bounding box determines the location of the weed in the image through a process called localization as shown in Fig. 7. This forms the next step in the object recognition process. The labelled image was saved in XML format which contains the four parameters (xmin, xmax, ymin and ymax) which describe a bounding box, needed for the object to be recognized. The four coordinates represent the x coordinate of the centre, the y coordinate of the centre, the height of the object and the width. The labelled image was saved in XML format as a dataset for the training. After the first stage of data preparation the image, the images obtained were divided into test and train. 90% of the data was used for training while the remaining 10% was used for testing. A total number of 4860 and 540 datasets were obtained for training and testing respectively. 10 samples each of the training and testing datasets are respectively shown in Tables 1 and 2.

Fig. 7 Weed annotation with labelling

Table 1 Sample of the training dataset

Filename	Width	Height	Class	Xmin	Ymin	Xmax	Ymax
Aeschynomene indica L. (16).jpg	259	194	weed	7	4	259	162
Aeschynomene indica L. (17).jpg	259	194	weed	21	15	259	162
Aeschynomene indica L. (18).jpg	259	194	weed	6	5	259	159
Aeschynomene indica L. (19).jpg	194	259	weed	3	3	189	259
Aeschynomene indica L. (22).jpg	259	194	weed	48	1	259	186
Aeschynomene indica L. (23).jpg	259	194	weed	3	1	259	193
Aeschynomene indica L. (25).jpg	259	194	weed	1	3	258	194
Aeschynomene indica L. (26).jpg	259	194	weed	36	49	191	194
Aeschynomene indica L. (27).jpg	259	194	weed	15	6	235	192
Aeschynomene indica L. (28).jpg	259	194	weed	2	5	257	191

Table 2 Sample of the testing dataset

Filename	Width	Height	Class	Xmin	Ymin	Xmax	Ymax
Commelina benghalensis L. 001.jpg	230	250	weed	1	1	230	249
Commelina benghalensis L. 002.jpg	250	166	weed	2	1	250	165
Commelina benghalensis L. 003.jpg	250	166	weed	4	5	245	166
Commelina benghalensis L. 004.jpg	250	140	weed	9	1	246	140
Commelina benghalensis L. 005.jpg	250	200	weed	3	2	249	200
Commelina benghalensis L. 006.jpg	250	179	weed	6	3	250	167
Commelina benghalensis L. 007.jpg	250	166	weed	41	12	220	166
Commelina benghalensis L. 008.jpg	250	187	weed	5	4	249	182
Commelina benghalensis L. 009.jpg	250	140	weed	56	5	250	137
Echinochloa colona (L.) Link 001.jpg	257	196	weed	11	7	257	193

3.3 Creating TensorFlow Records

Google TensorFlow (TF) is an open source software library for machine learning application such as a deep neural network. The google TF was chosen for this study because of its large number of trained libraries which are well recognised for machine learning applications and object detection exercises. The TF creation process is shown in Fig. 8. The models for training was obtained from GitHub repository and saved. The test and train images together with respective XML were converted into comma-separated value (CSV) format. The CSV format for the test and train was then converted to TensorFlow record for training the SSD model.

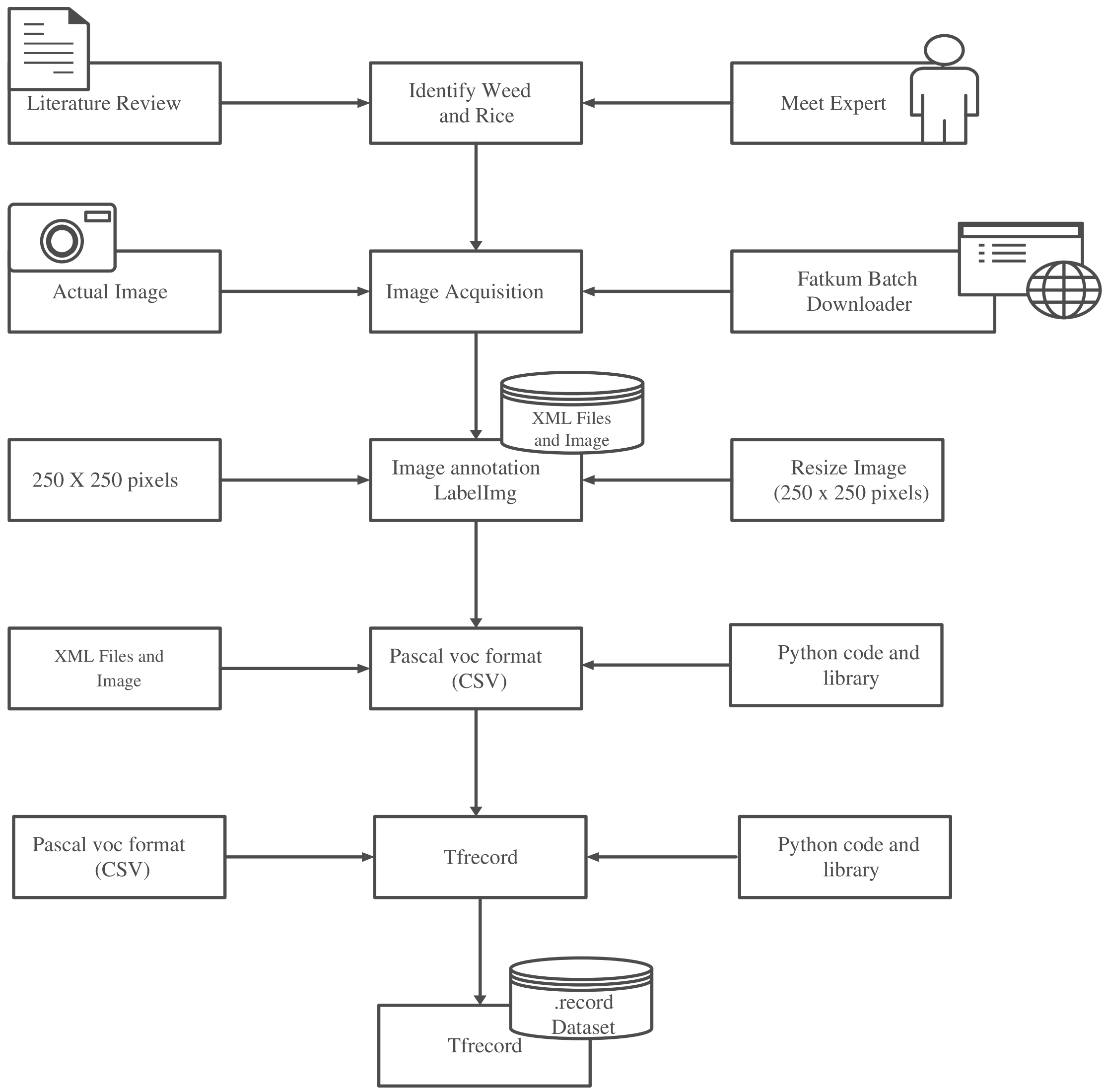

Fig. 8 Dataset generation process

3.4 *Paperspace*

Since the model training cannot be done locally due to hardware constraints, a virtual cloud computing platform was obtained on Paperspace. Paperspace is an online cloud computing platform that gives limitless computing power. The platform is used for deep learning, data exploration and gaming. Paperspace provides the Graphics Processing Unit (GPU) necessary for the training of the deep learning algorithm. GPU is necessary for training deep learning models because they require a lot of computational power to run on, and considerable hardware to run efficiently [30]. In the deep neural network training, two basic operations are performed, the forward and backward pass operations. Weights of the neural network are updated on the basis of the error obtained in the forward pass. For a typical convolution neural network of 16 hidden layers, it has about 140 million parameters.

Table 3 Specification of the paperspace machine

Machine name	Performance specifications
Operating system	Ubuntu 16.04 SSH/terminal only
RAM size	30 GB
Storage size	50 GB SSD
CPU	8 CPU cores plus GPU
Machine IP address	184.105.169.162
Support for AVX instruction	NO

The parameters are the weights and bias. Thinking of the number of multiplication applications, it would take typical system years to completely train the model [31].

Thus, using the GPU can help the model to be trained at once and in a shorter amount of time. The server image of Ubuntu 16.04 Secure Shell (SSH)/terminal was rented for this operation. The details of the server image are shown in Table 3. Basic python libraries such as NumPy, pillow, TensorFlow 1.5, pandas, CUDA drivers from Nvidia, TensorFlow GPU were installed on the machine to enable the SSD model training. The configuration of model selection and configuration files modification was done locally using Python 3.6. The configuration files, the images, the TF records and the SSD model training directory were copied inside a directory and transferred to the Paperspace for the training.

3.5 Model Training

The model configuration and libraries loading were done off line. Table 4 gives the system specification of the laptop used. In the configuration file, the class number, class name, path to TF records and batch size was configured. A class batch size of 12 was used to prevent the RAM from running out of memory due to large data thrown at it for training. A library for object detection ("protocol buffer") was loaded into the model to configure the model for the training on Paperspace. The training commenced with a high error rate of 7 but as the training started the error gradually got reduced until it got to nearly 1. Different checkpoints were made for the training model at every 1000 steps.

Table 4 Offline machine used for model configuration and extraction

System name	HP Pavilion 15-au010wm
RAM size	12 GB
Processor	Intel® Core™ i7
Video graphics	Nvidia® GeForce 940MX
Storage size	1 TB
Support for AVX instruction	YES

The training stopped at 26500 steps. The training results were obtained via google tensor board. Checkpoints of the model were recorded under a trained model after every 1000 steps. A frozen inference graph of the model was created and each checkpoint of the training makes up the result of the training. After the training, the trained network was downloaded to a core i7 system whose specification is in Table 4 because of its support for AVX instruction. The checkpoints together with a protocol buffer text file was used to load the tensor board to see how the training process took place and how the losses were reduced to ensure the data got trained.

3.6 Model Testing

For the neural network testing, the frozen inference graph which was named weed inference graph was loaded with the help of google TensorFlow 1.10. 50 random images from testing folder were used for testing and the result was plotted. The confidence level of each of the tested and recognized images was recorded on the images.

3.7 Model Deployment

The downloaded model was a frozen inference graph which was extracted using a google TensorFlow 1.10. The Model was loaded into Raspberry Pi3 via Virtual Network Computing (VNC) interface after quantization. The offline model was tested with images and the integrated system was tested to note its performance. The performance evaluation of the device was noted for accuracy, precision and sensitivity.

4 Results

The objective function of the SSD model is obtained by the summation of the loss function. The loss function is given as sum of the classification and localization. The goal is to have a loss below 2 for images that are quite similar. The two of them together shows how well the model has learnt. From Fig. 9 the losses encountered by the model was reduced as the training steps increases and which shows that the SSD model was actually learning as the steps increases.

Figure 10 shows the total loss obtained. The total loss which is a combination of localization loss and confidence loss. The two of them together which shows the objective function of the network as shown in Sect. 2. The graph above shows the network learning process as it moves from 0.000 steps to 24000 steps. From the graph

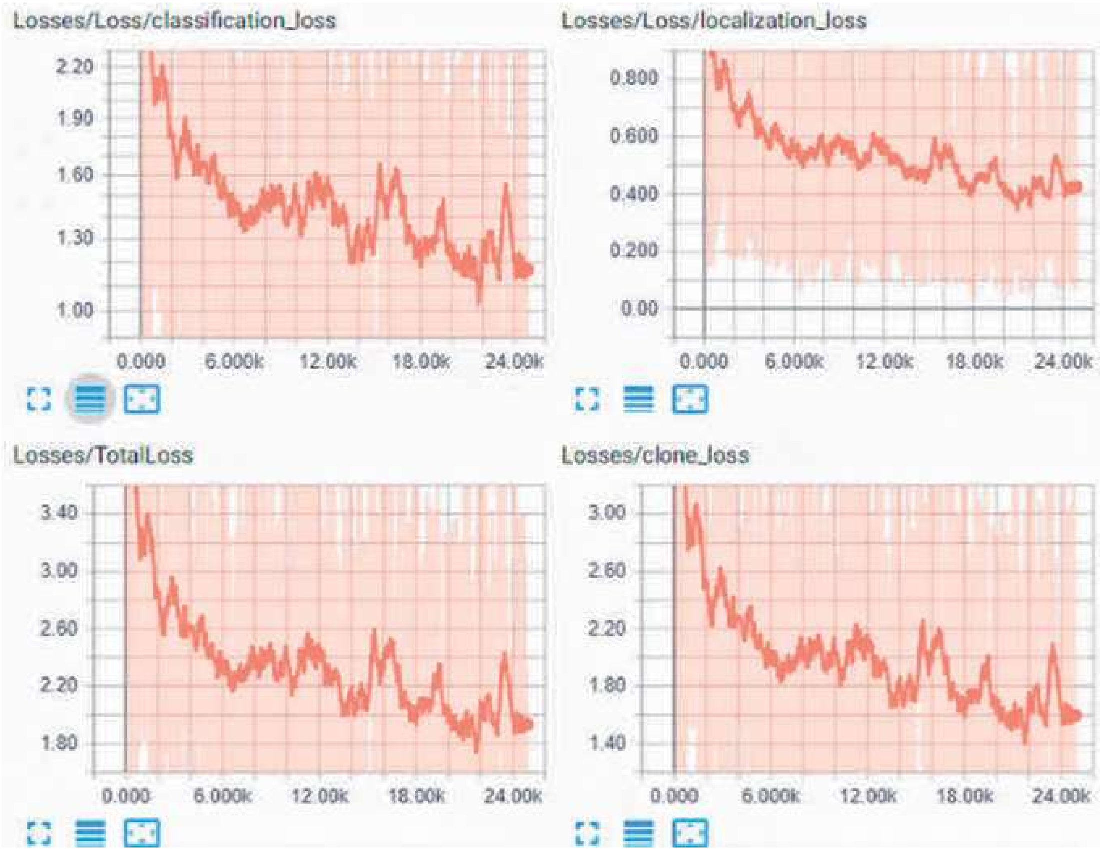

Fig. 9 Training result from tensor board

Fig. 10 Learning process of the deep learning algorithm

obtained from the tensor board, the SSD model was learning as the steps increased and reduction in loss signifies an improvement in the neural network algorithm.

Figures 11 to 13 show the results obtained from the deployment of the model on images from the farm and stand-alone images. The image was able to achieve high

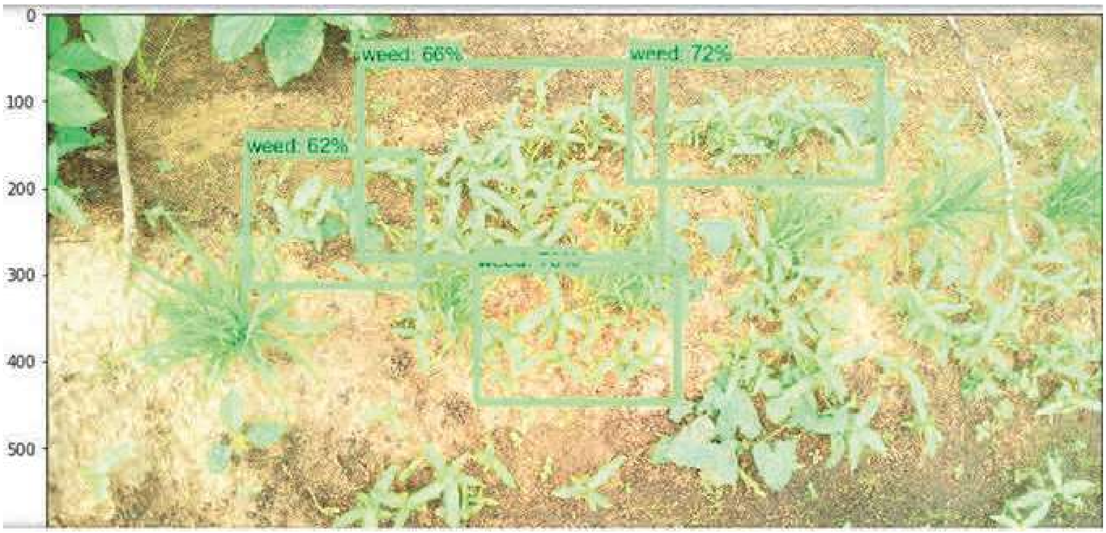

Fig. 11 Result obtained from running the model on weed species of rice obtained from the Cereal and Legume Crops practical plot, of teaching and research farm Department of Crop Production, Federal University of Technology, Minna, Nigeria

accuracy for high definition images and considerable accuracy for low-resolution images obtained from the school farm.

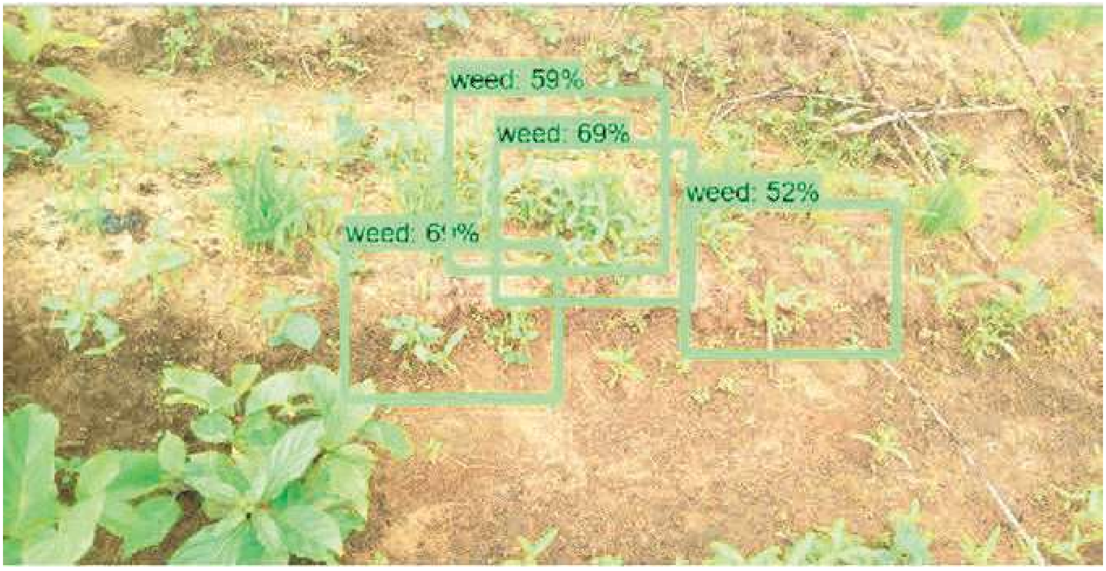

Fig. 12 More results from running the model on the image from the Cereal and Legume Crops practical plot, of teaching and research farm Department of Crop Production, Federal University of Technology, Minna, Nigeria

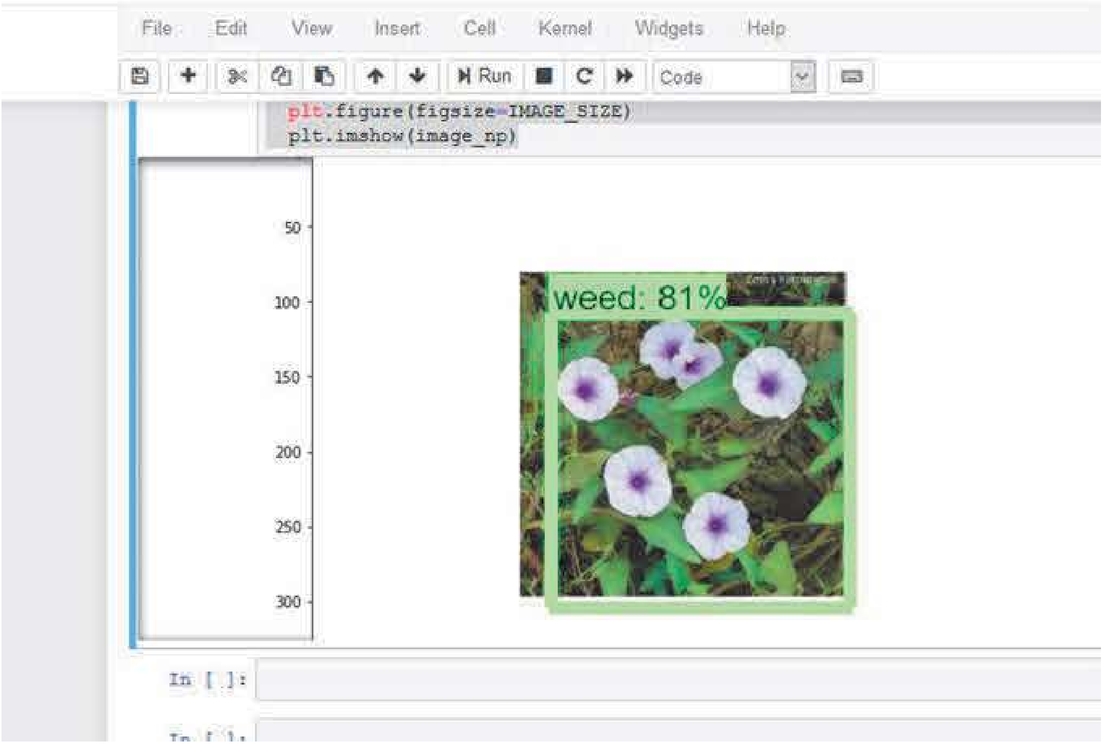

Fig. 13 Running the model on close range image of low land weed obtained from the Google server

4.1 Developed Model Performance Evaluation

The neural network was evaluated for 50 different test samples comprising of weeds and rice based on accuracy, sensitivity and precision as used by Aggarwal & Agrawal [32] and Aibinu *et al.* [33]. Sensitivity measures how well the neural network was able to detect positive case while accuracy measures how well the neural network was able to detect correctly the positive cases and precision measures the repeatability of the measured value. The metrics are represented mathematically as:

$$\text{Accuracy} = \text{TP} / (\text{TP} + \text{FN}) \tag{9}$$

$$\text{Sensitivity} = \text{FP} / (\text{TN} + \text{FP}) \tag{10}$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \tag{11}$$

where:
TP is true positive (number of items classified correctly)
FP is a false positive (number of negative cases classified correctly)
TN is true negative (number of correct cases classified wrongly)
FN is false negative (number of incorrect cases classified wrongly).

5 Conclusion

The SSD model based on three parts which the base convolution using the VGG 16 as the base network, this helped to achieve the lower level features, the auxiliary convolution during the forward pass allowed more feature extraction so as to create a features map that constitute a representation of the image at different scale and size.

Also, the prediction convolution helps to identify objects in the features map leading to the high accuracy of the model. Finally, the single shot multi- ‘box detector trained model showed high accuracy value of 99, 98 and 89% for close up images of weed species taken at high resolution and fairly good accuracy of 72, 69, 59% for weeds found in between rice plants. The model was able to identify the weed species associated with rice and did not misclassify the rice as a weed. In terms of the overall performance, the Single Shot Detector (SSD) model showed 86% accuracy, 84% precision and 92% sensitivity.

With the images of weed taken at very close range, the confidence level of the system becomes better. Application of this method in low land rice can reduce herbicide wastage, its detrimental effect on the environment, increase yield, and a better return of investment for the farmer.

References

1. Diao, X.: Economic importance of agriculture for sustainable development and poverty reduction: Findings from a case study of ghana Xinshen Diao, IFPRI, no. November, pp. 15–17 (2010)
2. Amin, M.S.M., Rowshon, M.K., Aimrun, W.: Paddy water management for precision farming of rice. In: Current Issues of Water Management, InTech (2011)
3. Naresh, R.K., Kumar, Y., Chauhan, P., Kumar, D.: Role of precision farming for sustainability of rice-wheat cropping system in western indo gangetic plains. Int. J. Life Sc. Bt. Pharm. Res. **1**, 1–13 (2012)
4. Rodenburg, J., Johnson, D.E.: Chapter 4—weed management in rice-based cropping systems in Africa. In: Sparks, D.L. (ed), Advances in agronomy, vol. 103, pp. 149–218. Academic Press (2009)
5. Pulido-rojas, C.A., Molina-villa, M.A., Solaque-guzmán, L.E.: Machine vision system for weed detection using **80**, 124–130 (2016)
6. Ajala, A.S., Gana, A.: Analysis of challenges facing rice processing in Nigeria. J. Food Process **2015** (2015)
7. Hossain, M.M.: Recent perspective of herbicide: review of demand and adoption in world agriculture. J. Bangladesh Agric. Univ. **13**(1), 19–30 (2015)
8. Singh, V., Jat, M.L., Ganie, Z.A., Chauhan, B.S., Gupta, R.K.: Herbicide options for effective weed management in dry direct- seeded rice under scented rice-wheat rotation of western Indo-Gangetic Plains. Crop Prot. **81**, 168–176 (2016)
9. Christensen S. et al.: Site-specific weed control technologies, pp. 233–241 (2009)
10. Burks, T.F., Shearer, S.A., Payne, F.A.: Classification of Weed species using colour texture features and discriminant analysis **43**(2), 441–448 (2000)
11. Watchareeruetai, U., Takeuchi, Y., Matsumoto, T., Kudo, H.: Computer vision based methods for detecting weeds in lawns, pp. 287–296 (2006)

12. Tellaeche, A., Burgos-artizzu, X.P., Pajares, G., Ribeiro, A.: A vision-based method for weeds identification through the bayesian decision theory. Pattern Recogn. **41**(2), 521–530 (2018)
13. Faridi, H., Aboonajmi, M.: Application of machine vision in agricultural products. In: Proceedings of the 4th Iranian International NDT Conference (IRNDT), (2017). Retrieved from http://www.ndt.net/?id=21279
14. Liakos, K.G., Busato, P., Moshou, D., Pearson, S.: Machine learning in agriculture: A Review. Sensors, **18**(8), 2674 (2018). https://doi.org/10.3390/s18082674
15. Herrera, P.J., Dorado, J., Ribeiro, Á.: A novel approach for weed type classification based on shape descriptors and a fuzzy decision-making method, pp. 15304–15324 (2014)
16. Lin, F., Zhang, D., Huang, Y., Wang, X., Chen, X.: Detection of corn and weed species by the combination of spectral. Shape and Textural Features, pp. 1–14 (2017)
17. Yang, C.C., Landry, J., Ramaswamy, H., Ditommaso, A., Prasher, S.: Application of artificial neural networks in image recognition and classification of crop and weeds. Can. Agric. Eng. **42**(3) (2000)
18. Pulido, C., Solaque, L., Velasco, N.: Weed recognition by SVM texture feature classification in outdoor vegetable crop images Reconocimiento de maleza por características de textura usando **2017**, 68–74 (2017)
19. Khan, Y.N.: Weed detection in crops using computer vision. Cent. Robot. (2015)
20. Zheng, L., Fu, C., Zhao, Y.: Extend the shallow part of single shot multibox detector via convolutional neural network. In: Proc. SPIE 10806, Tenth International Conference on Digital Image Processing (ICDIP 2018). https://doi.org/10.1117/12.2503001.2018
21. Nejati, H., Azimifar, Z., Zamani, M.: Using fast fourier transform for weed detection in corn fields. In: Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on, pp. 1215–1219 (2008)
22. Liu, H., Lee, S.H., Saunders, C.: Development of a machine vision system for weed detection during both of off-season and in-season in broadacre no-tillage cropping lands. Am. J. Agric. Biol. Sci. Publ. Online **9**(92), 174–193 (2014)
23. Chang, C.L., Lin, K.M.: Smart agricultural machine with a computer vision-based weeding and variable-rate irrigation scheme. Robotics **7**(36), 1–17 (2018)
24. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., Berg A.C.: SSD: Single shot multibox detector. In: European Conference on Computer Vision (pp. 21–37). Springer, cham. (2016)
25. Sun, C., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors
26. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 2818–2826 (2016). https://doi.org/10.1109/CVPR.2016.308
27. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: The Impact of Residual Connections on Learning, pp. 4278–4284
28. Szegedy, C., Com, S.G.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, vol. 37 (2015)
29. Vikhram, G.Y.R., Agarwal, R., Uprety, R., Prasanth, V.N.S.: Automatic Weed Detection and Smart Herbicide Sprayer Robot, vol. 7, pp. 115–118 (2018)
30. Jenny.: Paperspace: Overview (2018)
31. Alzantot, M., Wang, Y., Ren, Z., Srivastava, M.B.: RSTensorFlow: GPU enabled tensorflow for deep learning on commodity android devices. MobiSys. 7–12 (2017). https://doi.org/10.1145/3089801.3089805
32. Aggarwal, N., Agrawal, R.K.: First and second order statistics features for classification of magnetic resonance brain images. J. Signal Inform Process. (3), 146–153 (2012)
33. AIbinu, A.M., Salami, M. J. E., Shafie A.A.: Retina fundus image mask generation using pseudo parametric modeling technique. IIUM Eng. J. **11**(2), 163–177 (2010)