Abstract

Weeds can be defined as any plant or vegetation that interferes with farming or forestry purposes, such as planting crops, grazing animals or planting forests. Weeds compete with plants for one or more aspects of plant growth such as mineral nutrients, water, solar energy and environment and inhibit plant performance. Therefore weed control is very important for plant growth.Traditional weed control methods are cultural weed control, mechanical weed control, biological weed control and chemical weed control. Traditional methods of weed control do not work effectively and the use of accurate farming plays a role. The next problem is finding the family of weeds and the right herbicide to spray the weeds.

To overcome this difficulty, An android/ios app named Kisan Ki Dhosth (KKD) is designed.This application clearly identifies the weed and identify the appropriate herbicide.This can be achieved by using the photograph taken by the farmer. The dataset consist of weeds which are widely seen in Kerala, making it more user-friendly for farmers.Since persistent use of herbicide remain active in environment for long period of time,potentially causing soil and water contamination,home remedies like salt,vinegar,newspaper and boiled water can control be used.

Chapter 1

Introduction

1.1 Overview

A weed is an undesirable plant growing at the wrong place. They tend to degrade the quality of the desired plants.Competition with the desired crops for the most needed resources for the plant, i.e., direct sunlight, soil nutrients, water, and (to a lesser extent) growth space.As a result,weed control is important for agriculture. Methods include plowing, farming-powered farming, cracking or sunburn, deadly wilting at high temperatures, heat, or chemical attack with herbicides.

Among the mentioned methods, the simplest one to implement is to use herbicides.Mostly, a single herbicide would be sprayed in the entire plot.This maybe effective on few weeds but not on all. So exact identification of the weeds must be done and also the method to eliminate it as well.

To tackle this problem, an android/ios app named Kisan Ki Dhosth(KKD) is designed which clearly identifies the weed using the photograph taken by the farmer. The appropriate herbicide would also be advised.The dataset consist of weeds which are widely seen in Kerala, making it more user-friendly for farmers.Since sustainable agriculture is of vital importance, use of herbicides must be reduced. Hence more reliable solution such as home remedies are also provided to eliminate weeds ,if possible.

1.2 Problem Statement

Research works have been done on crop and weed detection from a very long time. Convolutional Neural Networks,Image processing are all the foundation for these researches. But no model exist where the weed classes are correctly classified. The android/ios application is a solution for this problem. It correctly identifies the weed class and also suggest the herbicide or the home remedies available to eliminate it.

1.3 Objective

Given a photograph as input, the goal is to identify the class of the weed and also to recommend the herbicide for it. For it to be more reliable, translation of text to Malayalam is also done.

Chapter 2

Related Works

2.1 UAV-Based Crop and Weed Classification for Smart Farming

The authors are Philipp Lottes, Raghav Khanna, Johannes Pfeifer, Roland Siegwart and Cyrill Stachniss[5]. The year of publication is 2017.In this paper, they address the problem of detecting value crops and typical weeds using a camera installed on an Unmanned aerial vehicle(UAV).

The main purpose of the proposed plant classification is identifying sugar and weed plants in UAV images to provide an accurate monitoring tool for the planting of real fields.The installation of the system is as follows either 4-channel RGB + NIR images or standard RGB images, depending on the UAV sensor setup. They used and tested their system using UAVs in pairs farms, one German and the other in Switzerland and show that their method allows for the analysis of the field and the separation of individual plants.

Figure 2.1: Low-cost UAV used for field monitoring.

The three steps of their approach are: First, they apply a pre-processing to remove the background i.e mostly the soil and get the resultant vegetation. Second, they extract features from vegetation with a combination of an object- based and a keypoint-based approach. Third, they apply a multi-class Random Forest(RF) classification and obtain a probability distribution for the predicted class labels.The evaluation metrics used in this project are ROC curves and Precision-Recall(PR) plots and the datasets are JAI, MATRICE and PHANTOM.

2.2 Mixtures of Lightweight Deep Convolutional Neural Networks

The authors are Chris McCool,Tristan Perez and Ben Upcroft[6].The year of publication is 2017.In this paper,they propose a novel approach for training neural networks that allows us to tradeoff complexity and accuracy to learn lightweight models suitable for robotic platforms such as AgBot II.

The proposed approach is achieved through three-step process.First step is to adapt a pretrained model .Second by using an adapted model and employ model compression techniques.Thirdis by combinining a set of K-lightweight models.This mixture model further enhances the performance of the lightweight models.

To enhance the performance of the compressed models they combine multiple lightweight model as a mixture model of Mix-Agnet and Mix-mini Inception.This approach lead tow impressive results for weed segmentation.The adapted-IV3 model provides state-of-the-art performance,improving accuracy from 85.9percent to 93.9percent.All of their models were implemented in Tensorflow.

The AdamOptimizer was used with a learning rate=le-4,a batch size of b=60 and a dropout rate of 50 percent. When training the lightweight DCNNs they use the same data as was used to fine-tune the complicated network. Weed segmentation is key to enabling integrated weed management for intra-row weed management.This task is made challenging by fact that the weeds can overlap with the crop making accurate classification difficult. The downside to this highly accurate model is that it is current not able to be deployed for real-time processing on a robotic system as it can only process 0.12 frames per second.These model are able to to process 1.83 and 1.07 frames per second.

By increasing the complexity,ie by adding more models, the number of regions that can be processed per second can be decreased. This provides us with a method to tradeoff accuracy and run-time speed. For the highly accurate Adapted-IV3 model we can achieve an accuracy of 93.9percent, however, we are only able to process 155 regions per second. By contrast, the Mix-AgNet K = 4 model can achieve an accuracy of 90.3percent but is able to process 15.8 times regions per second and the Mix-MiniInception K = 2 model can achieve an accuracy of 90.5percent while processing 9.3 times the regions per second.

Future work inckude to examine how to improve the accuracy of the lightweight DCNNs .Alternative formulations of the lightweight DCNN will be examined including if greater depth needs to be introduced into network as this was shown to be important.

2.3 Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields

The authors are Nived Chebrolu, Philipp Lottes, Alexander Schaefer, Wera Winterhalter,Wolfram Burgard and Cyrill Stachniss1[3].The year of publication is 2017. In this paper, they present a large database of agricultural robots for crop differentiation and landscaping and mapping including appropriate crop growth phases to interfere with robots and weed control.

There is a growing interest in agricultural robots and precision farming. For such domains, appropriate data sets are difficult to find, as dedicated fields require care and data collection time is important. On average, they compiled data three times a week, starting with the emergence of crops and ending in a state where the field was no longer available on the machine without damaging the crops. The robot was attached with four camera channels and an RGB-D sensor for capturing more information on the field. Many lidar sensors as well as the global positioning system and wheel encoders provided measurements related to location, navigation, and mapping respectively. All sensors were limited before the data acquisition campaign.

The agricultural robot platform is BoniRob.The BoniRob platform is a multipurpose robot by BoschmDeepField Robotics.Sensors used are JAI AD-130GE camera, Velodyne VLP16 Puck, Kinect One (Kinect v2), Nippon Signal FX8 Le, Ublox GPS GPS provides visibility, depth, 3D laser, GPS and odometry data. It has a removable PC with i7 core processor and 6 GB DDR3 memory and Ubuntu 14.04.

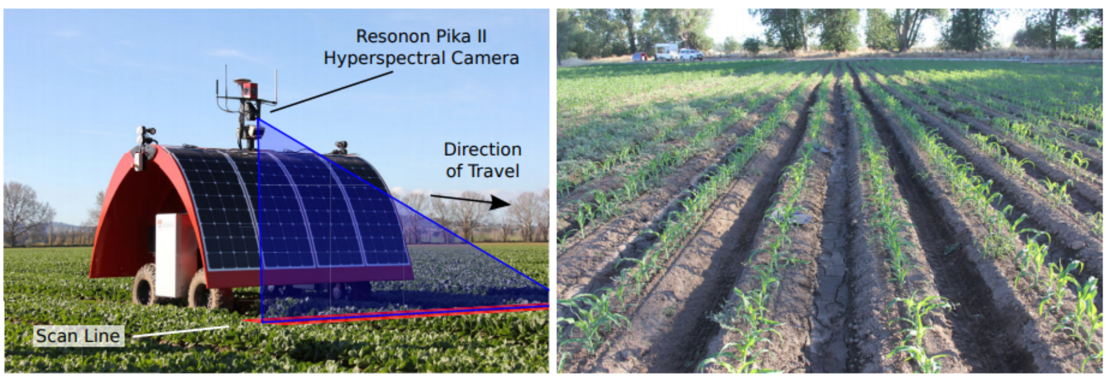
Figure 2.2: BoniRob operating on field.

2.4 Self-Supervised Weed Detection in Vegetable Crops Using Ground Based Hyperspectral Imaging

The authors are Alexander Wendel and James Underwood.The year of publication is 2016[8].In this paper,they present a self-supervised framework for weed detection using hyperspectral data with prior knowledge of seeding patterns using an autonomous mobile ground vehicle.The results of the experiment conducted on corn crop rows to evaluate the system's performance and limitations are also stated

In this paper,high resolution hyperspectral imagery is used.The proposed approach is a 8 step process.The first step is HSI Data collection i.e the hyperspectral data gathered by the ground vehicle.The next step is NDVI Vegetation Segmentation.It is a technique used to seperate the vegetation pixels from background pixels.After segmentation,manual data labelling was done along with automatic generation of training data to make a comparison between the performance.For automatic generation,the segmented data is given as input to Hough transform.It is a feature extraction technique used to locate edges and rows.In this way,the training dataset is generated.Spectral Preprocessing was done on this data to mitigate the effects of varying environmental illumination at different times of the day like when there is direct sunlight or in the evening where there is a shade.Then normalisation was done using PCA for feature extraction and dimensionality reduction on the data.Both LDA(Linear Discriminant Analysis) and SVM(Support Vector Machines) were used as classifiers and implemented using the scikit-learn package.Both SVM anf LDA performed similarly.SVM is a complex but well known classifier, which has been employed with HSI in the past.

The method gives highly accurate per pixel classification.This technique is also adaptable to changes in seasons,geography,illumination etc.This paper demonstrates how to distinguish crop and weed without manual labelling, with a framework that is equipped to handle appearance variability. Further testing is required to confirm and demonstrate how robust the system is to changing conditions compared to static training data.



(a) Platform Overview (b) Typical Corn Crop Rows

Figure 2.3: Hyperspectral camera collecting data from corn crop Rows

2.5 Plant Classification System for Crop/Weed Discrimination without Segmentation

Authors are Sebastian Haung,Andreas Michaels ,Peter Biber,Jorn Ostermann[4].The year of publication is 2014.This paper proposes a mechanical approach to crop classification without segmentation . The system can distinguish crop and weeds from fields where they tend to grow together.This can help in finding different strategies to eliminate weeds in more cost effective way and also with less impact on environment.

The architecture is 6 stage pipeline in which there are online and offline steps.The first step is to collect the pictures for the input.They are collected using a camera mounted on a robot. The next step is background removal,which is done to segment the vegetation from the soil.In the next step, computations are done on the image tiles or image patches.Later the features are extracted from the images generated in the preceding step.the Random Forest Classifier is chosen to train and classify the images.The classifier training is done offline.The Markov Random Field approach is used for smoothening the images and finally interpolation of the sparse results is done.

The method was tested using a database of photographs taken from a live carrot farm with an independent field robot under field conditions.The average classification accuracy was found to be 93.8%. The limitation is that intra class overlap are not split into different plant regions in the final image. Hence the spatial arrangement o crop and weed could be used as a priori information in the future.

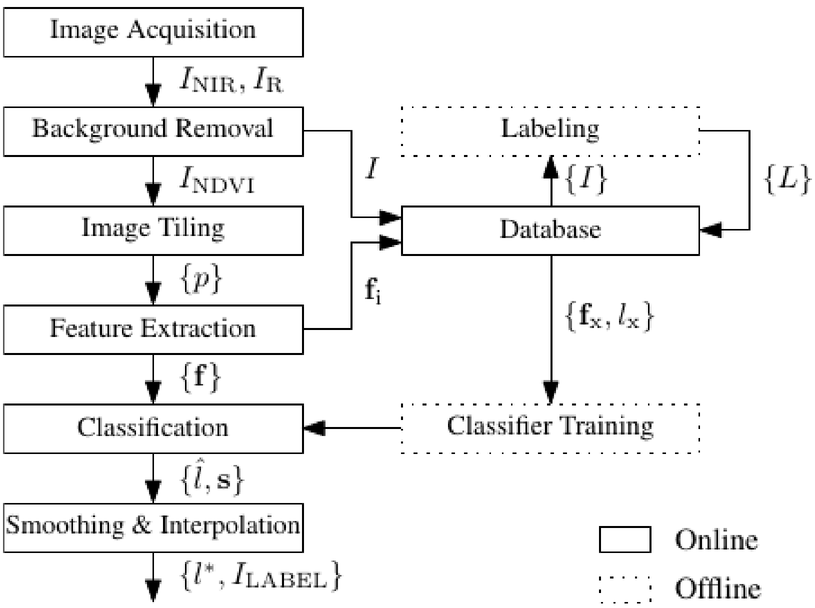


Figure 2.4: Overview of Classification Pipeline for Crop/Weed Discrimination without Segmentation

2.6 SegNet:A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation

The authors are Vijay Badrinarayanan,Alex Kendall and Roberto Cipolla.The year of publication is 2017[2]. They present a novel and practical deep fully convolutional neural network architecture for semantic pixel-wise segmentation termed SegNet.

This core trainable segmentation engine consists of an encoder network, a corresponding decoder network followed by a pixel-wise classification layer.One of the main contributions of this paper is their analysis of the SegNet decoding technique and the widely used fully Convolutional Network. This is in order to convey the practical trade-offs involved in designing segmentation architectures.

The key component of SegNet is the decoder network which consists of a hierarchy of decoders one corresponding to each encoder.They evaluate the performance of SegNet on two scene segmentation tasks, CamVid road scene segmentation and SUN RGB-D indoor scene segmentation.SegNet has an encoder network and a corresponding decoder network, followed by a final pixelwise classification layer.The encoder network consists of 13 convolutional layers which corre- spond to the first 13 convolutional layers.The final decoder output is fed to a multi-class soft-max classifier to produce class prob- abilities for each pixel independently.

They use the CamVid road scenes dataset to benchmark the performance of the decoder variants. This dataset is small, consisting of 367 training and 233 testing RGB images.They use the cross-entropy loss [2] as the objective func- tion for training the network. The loss is summed up over all the pixels in a mini-batch.

To compare the quantitative performance of the different decoder variants, they use three commonly used perfor- mance measures: global accuracy (G) which measures the percentage of pixels correctly classified in the dataset, class average accuracy (C) is the mean of the predictive accuracy over all classes and mean intersection over union (mIoU) over all classes as used in the Pascal VOC12 challenge.

The best performance is achieved when encoder feature maps are stored in full. This is reflected in the semantic contour delineation metric (BF) most clearly.When memory during inference is constrained,then compressed forms of encoder feature maps (dimensionality reduction, max-pooling indices) can be stored and used with an appropriate decoder (e.g., SegNet type) to improve performance.Larger decoders increase performance for a given encoder network.The main motivation behind SegNet was the need to design an efficient architecture for road and indoor scene understanding which is efficient both in terms of memory and computational time. They analysed SegNet and compared it with other important variants to reveal the practical trade-offs involved in designing architectures for segmentation, particularly training time, memory versus accuracy. Those architectures which store the encoder network feature maps in full perform best but consume more memory during inference time. SegNet on the other hand is more efficient since it only stores the max-pooling indices of the feature maps and uses them in its decoder network to achieve good performance.

2.7 Automatic Model Based Dataset Generation for Fast and Accurate Crop and Weeds Detection

The authors are Maurilio Di Cicco, Ciro Potena, Giorgio Grisetti and Alberto Pretto.[9]The year of publication is 2017.In this paper,they addresses the problem of creating large agricultural datasets with pixel-level annotations.It also put forwards an approach to decrease the manual labour required to train a visual model classification system.Large training datasets are created routinely by giving arbitrary values to the important features of the farm land such as crop and weed species, soil type,light conditions etc.

This paper addresses the challenge of creating efficient farming robots which can provide effective crop and weeds classification for precision farming. The strategy is to develop a visuals-based classification system which can distinguish crops from weeds.Large datasets are required for more accuracy in precision farming and creating them is a huge task.Unfortunately, doing pixel-wise annotation maually is a very difficult and time consuming task. In fact, because of this issue, the size of semantic data stocks is generally small. In this paper, they make use of open image engine as a solution to the problem, i.e. by structuring data systematically.

For the procedural dataset generation,Unreal Engine 4[2] was used as graphic engine.A clear-cut model of the target environment was generated and using few real world textures,it creates a large variety of annotated data of plant species using a procedural method.Two datasets are used both collected from a BOSCH Bonirob farm robot moving on a sugar beet field.The first datatset contains 100 images and it has been gathered during the first growth stage of the plants i.e when both the crop and weeds have not developed completely.The second datatset is contains 400 images and it has been gathered after 4 weeks when the plants have developed and are in their advanced growth stages.The performance and quality of the synthetic databases are evaluated using an effective pixel-wise classification CNN,SegNet.The results show that a virtual dataset containing all the weeds has the same level of accuracy of of real-world images.

Figure 2.5: BOSCH BoniRob operating on some field to generate the datasets to be used in experiments.

2.8 Semantic Segmentation of Mixed Crops using Deep Convolutional Neural Network

The authors are Anders Krogh Mortensena, Mads Dyrmannb, Henrik Karstoftc, Rasmus Nyholm Jørgensenc, René Gislumd[7].The paper was published in 2016. Measuring the in-field biomass and plant composition is important for farmers and researchers using close-up crop rotation images, plant species can be classified using image processing.

In this paper, deep convolutional neural networks(DCNN) for semantic segmentation (or pixel-wise separation) of dense classes in RGB(Red, Green,and Blue)images were tested in the event of crop capture and voluntary barley. RGB images from a plot trial using oil radish as catch crops in barley. The images were captured using a high-end consumer camera mounted on a tractor. The images were manually annotated in seven classes: oil radish, barley, weed, stump, soil, equipment and unknown.To artificially increase the size of the dataset, data argumentation was done. Images were flipped and rotated. A modified version of Visual Geometry Group-16(VGG-16) deep neural network was used.

First, the last fully connected layers were transformed into a convolutional layer and the depth was adjusted to accommodate the number of our classes. Second, a deconvolutional layer with thirty-two stride was added between the fully integrated end layer and the softmax partition layer to ensure that the output layer has the same size as the input.The network was supposed to be trained for forty epochs, but ten epochs later it started to overflow so it was disconnected after twnty-three of them. The network model stored after ten epochs has been used to produce results. A pixel accuracy of 79% was shown initially.

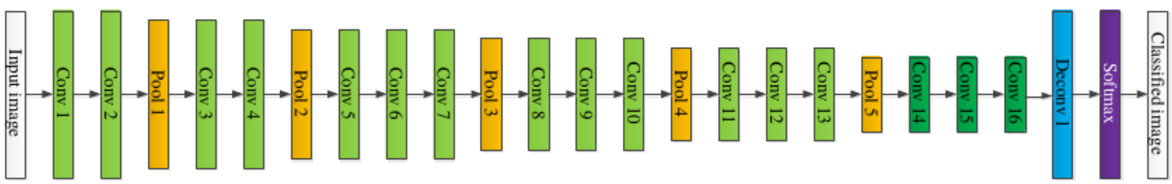


Figure 2.6: Base Network Model

2.9 Comparitive Study

Reference Paper	Methadology	Advantages	Disadvantages	Improvement
1.UAV-Based Crop and weed classification for Smart Farming	Key steps: 1.Vegetation Detection 2.Objects-based vs. Keypoint-based Feature 3. Classification	1.Offers the potential to reduce the amount of agrochemicals applied to fields 2.Low cost 3.Compared to ground vehicles, UAV can cover large areas in a comparably short amount of time.	Using UAV data instead of data recorded with an UGV is more challenging as the imagery is naturally exposed to varying lighting conditions and different scales.	Need more efficient methods to collect UAV data.
2.Plant Classification System for Crop/ Weed Discrimination without Segmentation	A 6 step pipeline structure: image acquisition,back ground removal, image tiling, feature extraction, classification and smoothing.	No plant or leaf segmented input data is required. Overlap is not a special case.	Multiple plant of same class that overlap are not split in to different plant regions in the output label image.	The arrangement of the crops growing in rows can be leveraged to further improve the classification results.
3.Mixtures of lightweight Deep Convolutional neural networks : Applied to agricultural robotics	Training DCNNs that allows to trade-off complexity and accuracy to learn ightweight models suitable for robotic platforms.	1. Improve accuracy from 85.9% to 93.9% 2. Low memory requirement	It can only process 0.12fps. Need specialised hyper spectral cameras.	Light weight DCNNs can be improved.
4. Self Supervised Weed Detection in Vegetable crops using ground based Hyperspectral imaging	Uses hyperspectral data to get accurate per pixel classification of crop and weed without the need for manual labelling.	1.Highly accurate per pixel classification to crop or weed. 2. No need of manual labelling	1.vey small pixels will not be classified due to limited spatial resolution. 2.Slightly less performance compared to manual labelling of data.	The inclusion of spatial information, real time performance and online loop closure for weed control.

5. Agricultural robot dataset for plant classificatioon, localization and mapping on sugar beet filds.	The BoniRob platform is a multi-ourpose robot and is developed for applications in precision agriculture.	Provides researchers with a challenging real-world dataset that helps develop autonomous capabilities for field robots.	None mentioned	The data for this dataset was collected during one crop season. There is a scope to include more crop seasons.
6. Semantic Segmentation of Mixed crops using Deep Convolutional neural Network.	The collected images are trained using a model which consist of 15 convolutional layers, 5 max pooling layers and a soft-max classification layer.	Provide an accuracy of 79% at preliminary stage which is grater than the existing results.	Single straws were not detected and finer details of the objects varied greatly.	A smaller stride will not most likely improve the performance of the network and its ability to capture the finest feature.
7. SegNet: A Deep Convolutional Encoder-Decoder Architecture for the Image Segmentation	Novel and practical deep fully convolutional neural network architecture for semantic pixel-wise segmentation termed SegNet.	Application ranging from scene understanding,inferring support relationships.	Max pooling and sub-sampling reduce feature map resolution.	End-to-end learning of deep segmentation architecture.
8. Automatic Model based Dataset Generation for fast and accurate crop and weeds detection	An approach to minimize the human effort required to train a visual model classification system by creating an explicit model of the target environmental and procedural dataset.	1.Reducing the time and human effort. 2.Can be used even in the presence of a real dataset with a limited amountof data,just as a supplement.	Over fitting phenomina can happen at times.	Rendering of realistic NIR images-a common and useful source of data in the precision agricultural context.

Chapter 3

Design

3.1 Introduction

Design, the initial step in the development phase for any engineered system or product. It may be described as, the process of applying different methods and principles for the purpose of defining a device, a process or a system in enough detail to allow its physical realization.

In early days, software development consisted of just writing code by a programmer in order to solve a problem or to automate a procedure. Nowadays, systems are so large and complex that needs bunch of teams of architects , analysts, programmers, testers and users, who must work together to create the millions of lines of custom-written code that drive our enterprises. There are a sequence of stages in which the output of each stage becomes the input for the next:

1. Project Planning, feasibility study: Establishes a high level view of the intended project and determines its goal.

2. System analysis, requirements definition : Refines project goals into defines functions and operations of the intended applications.

3. System Design: Describes desired features and operations in detail, including screen layouts, process diagrams etc.

4. Implementation : The real code is written.

5. Integration and Testing : Brings all the tasks together into a testing environment.

6. Acceptances, installation, deployment : Final stages of initial development, where the software is put into production and runs actual business.

3.2 Design Methodologies

3.2.1 Dataset and dataset preprocessing

The dataset was generated by collecting images from various sources. This includes online websites (dreamstime.com, Pinterest, iStock, getty images), existing public dataset from the work, "DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning" and also manual clicking of pictures using mobile camera. The images were clicked under different lighting conditions and atmosphere. Eight weeds widely seen in Kerala were selected for our dataset. They are:

1. Cockspur Grass
2. Lantana Camara
3. Mimosa Pudica
4. Oxalis Weed
5. Parthenium
6. Siam Weed
7. Snake Weed
8. Stinging Nettle

A total of 10,400 images were collected with 1300 images of each weed.Out of these 1300 images of each weed, 900 images were taken for training. The validation dataset consist of 300 images and remaining 100 images were for test dataset.

3.2.2 Image Augmentation

The Deep learning model has a huge number of graspable variables compared to the number of images in the training set because of which there is a chance of overfitting problem and also deep learning models performs better with large datasets.In general, the more data we have better will be the performance of the model. So, data augmentation is done which can artificially increase the size of the dataset.This assists in building simpler and robust models which helps to generalize better.This leads to remarkable development in the phase of neural network training and to be more robust and accountable for the testing phase.Different augmentation techniques are applied on the dataset taken after the preprocessing phase, which contains 2-D images of each sample. This helps to enlarge the dataset by increasing the number of samples for each weed type.The Keras deep learning library provides the ability to use image augmentation automatically when training a model using the ImageDataGenerator class.Supported image formats: jpeg, png, bmp, gif. Animated gifs are truncated to the first frame.It generates batches of tensor image data with real-time data augmentation.The data will be looped over(in batches).The target size of our RGB images are (100,100).

The techniques used are:

1. Rotation
2. Shifting(Width and Height)
3. Flipping
4. Shear Intensity
5. Zoom

3.2.3 Deep Learning Phase

Deep learning in the field of computer vision, has achieved a dramatic change, particularly, in image acquisition,classification, segregation and recognition and is considered to achieve superior results, as well as performance on image classification models. Deep Learning uses layers of algorithms for processing, analysis and detection of hidden patterns in data and and visual objects recognition. Information is passed from each layer of a deep network, with output of the preceeding layer provided as input for the next layer. The input layer is the first layer, whereas output layer is the last layer and all the layers between are referred to as hidden layers of the network. Each layer is simple and algorithm-friendly which contains a specific type of activation function. Also, advances in deep Convolutional Neural Network(CNN) have helped to reduce the level of error.

Here, deep learning phase consists of a Convolutional Neural Network(CNN) with 6 layers followed by a single dense layer and the output layer. First 6 layers, which consists of a convolutional 2d layer a max pooling layer, after that comes the flattenning layer and finally a dense layer and output layer.

Transfer learning is a method in deep learning where a neural network is first trained on a problem similar to the problem that is being solved. It is used here to achieve better performance for our Deep Learning Model. It stores the knowledge gained while training a model for image classification problem and applies it to a different classification problem. For example, the knowledge gained while learning to recognize cats could apply when trying to recognize dogs. Transfer learning helps to decrease the training time for a learning model and can also result in lower generalization error.We use the Inception v3 model here for weed classification. It is a widely-used image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. It is based on the original paper: "Rethinking the Inception Architecture for Computer Vision" by Szegedy, et. al.

3.2.4 Performance Metrics

To assess the performance of the proposed approach, the paper measures the Confusion Matrix, Precision, Recall, Accuracy and F1-score. These metrics are defined based on the confusion matrix.
The Confusion Matrix for measuring performance uses the following terms [14]:

- TP (i.e., true positive), which corresponds to the number where the model correctly predicts the positive class. properly classified.
- TN (i.e., true negative), which corresponds to the number where the model incorrectly predicts the positive class.
- FP (i.e., false positive), which corresponds to the outcome where the model correctly predicts the negative class.
- FN (i.e., false negative), which corresponds to the outcome where the model incorrectly predicts the negative class.

		Predicted value	
		True	**False**
Actual value	**True**	True positive	True negative
	False	False positive	False negative

Accuracy
Accuracy is the ratio of correctly predicted instances to the total instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Precision
Precision is the ratio between correctly predicted positive observations to the total predicted positive observations. It indicates the proximity to the expected solution.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

Recall
Recall is the ratio of correctly predicted positive instances to the all instances in actual class - yes. Recall is a degree of the number of relevant results.

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

F1-score
F1-Score is the weighted average of Precision and Recall. F1-score indicates the balance between Precision and Recall.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{10}$$

3.3 System Architecture

The architecture all together has five main phases. The first phase, which is dataset collection and pre-processing, while the second one is data augmentation phase and then deep learning training is used as the third phase.Image samples of eight types of weeds are collected from various sources and and the dataset is generated. It is given to the first phase for pre-processing, where feature extraction takes place, after which only important features are obtained.In the second phase, augmentation using different classical augmentation techniques is done, after which we will obtain an enlarged dataset. This is then given to deep learning phase for classification.In the fourth phase,a database containing information and the specific type of herbicide for each weed is created and connected with the deep Learning model. Finally, An app named "Kisan Ki Dosth" is developed using the flutter framework as Graphical User Interface(GUI).

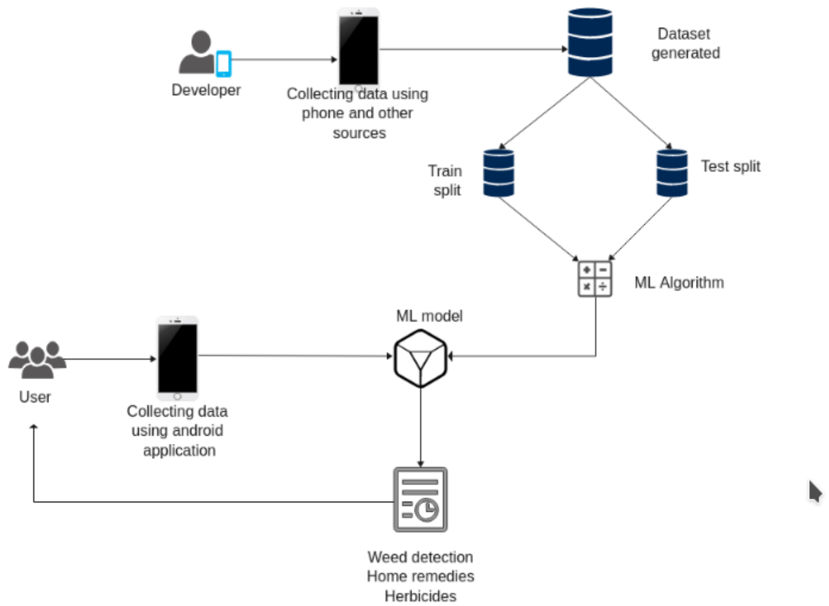
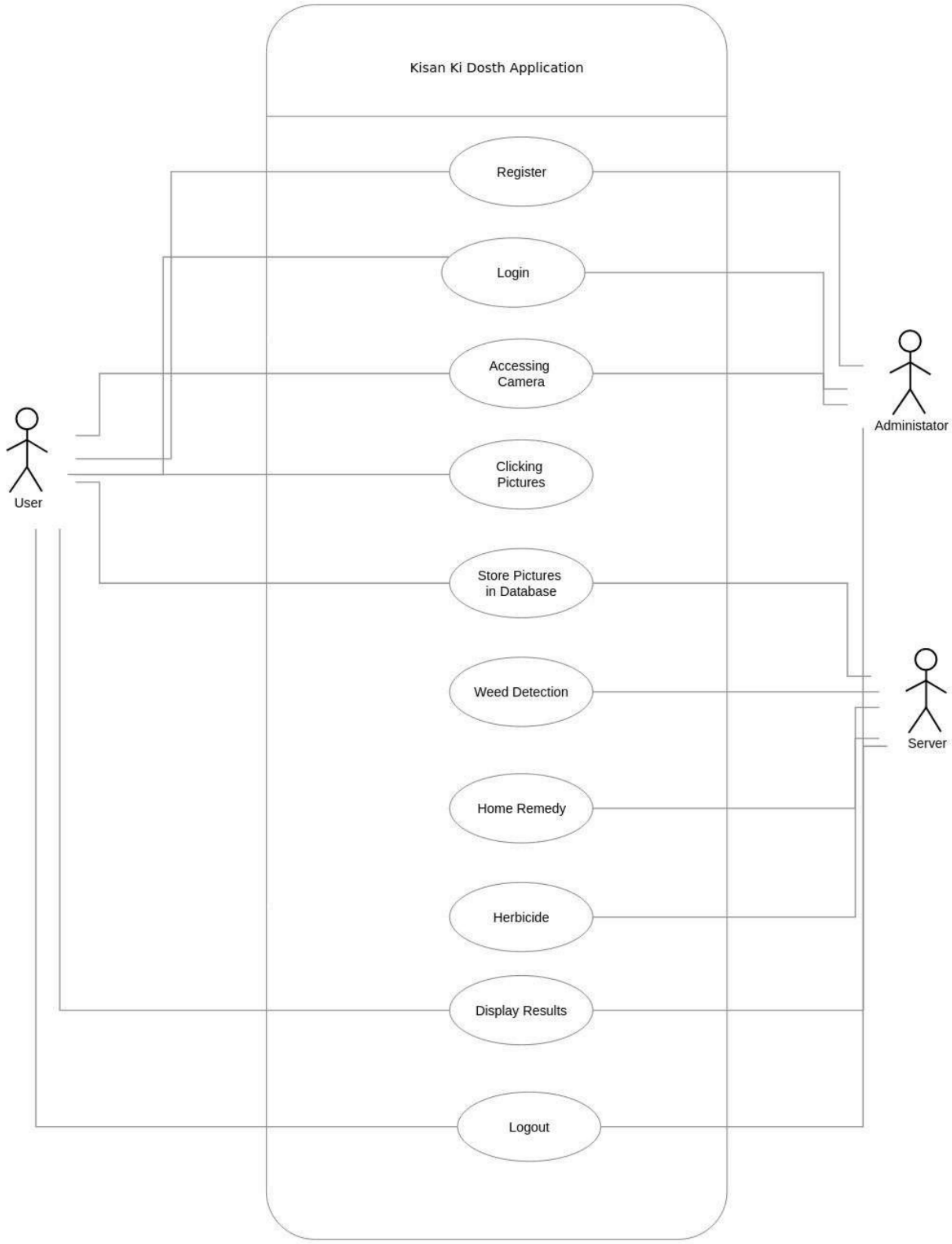
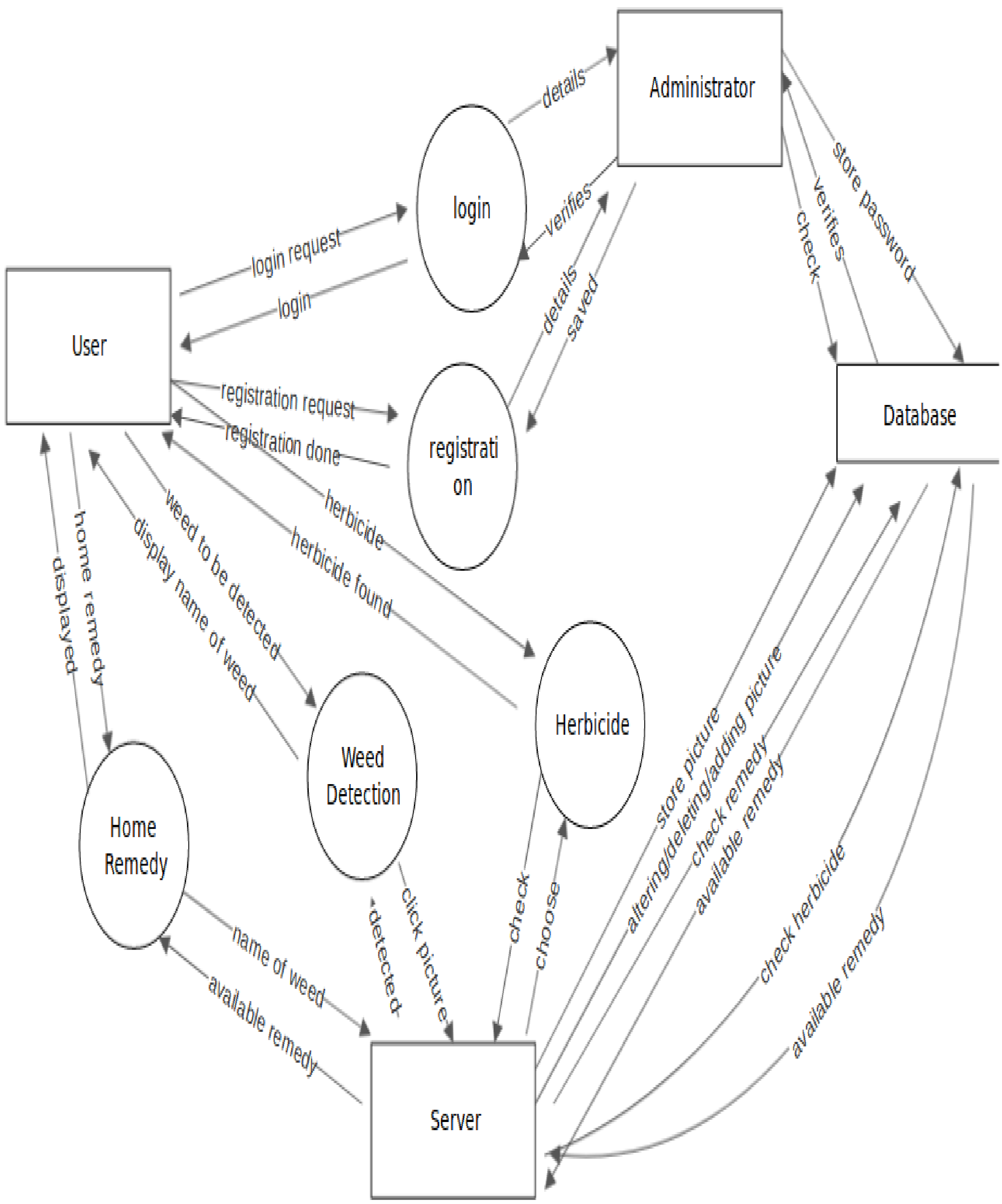


Figure 3.1: System Architecture

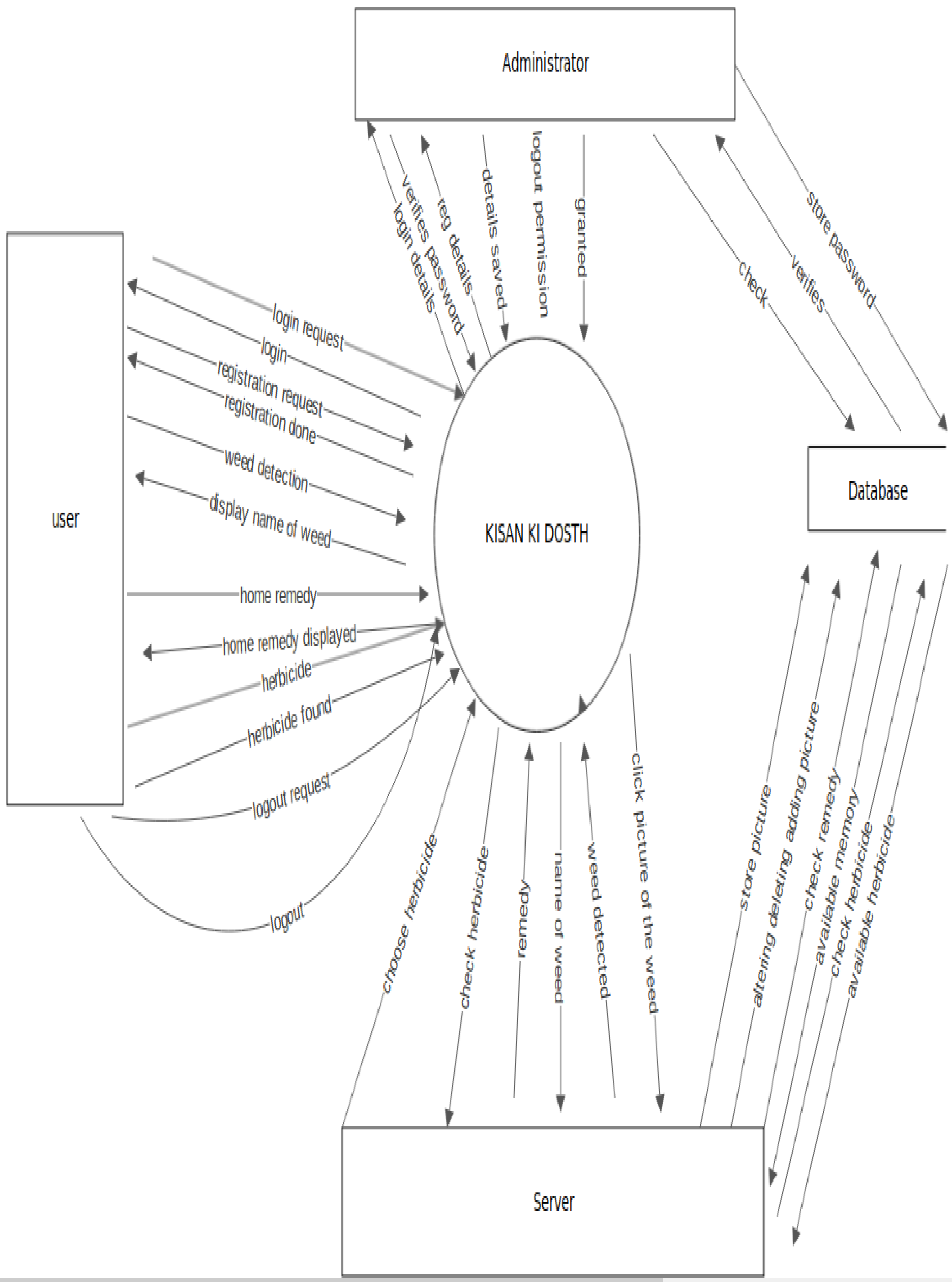
3.4 Use case Diagram



3.5 Data Flow Diagram



Level 0 DFD


Administrator
user
KISAN KI DOSTH
Database
Server
login request
login
registration request
registration done
weed detection
display name of weed
home remedy
home remedy displayed
herbicide
herbicide found
logout request
logout
login details
verifies password
reg details
details saved
logout permission
granted
check
verifies
store password
choose herbicide
check herbicide
remedy
name of weed
weed detected
click picture of the weed
store picture
altering deleting adding picture
check remedy
available memory
check herbicide
available herbicide

Level 1 DFD

Chapter 4

Implementation

4.1 Implementations Details

The implementation stage of a project is when the theoretical design is transformed into a functioning system. It entails meticulous planning, examination of the current system and its limits on implementation, and the design of ways to bring the system up to speed. The more complicated the system being developed, the more time and effort will be necessary for system analysis and design just to get it up and running. The activities are managed according to the plan, and discussions on equipment and resources are held. As a result, additional equipment must be obtained in order to implement the new system. The final and most crucial stage is implementation. The most important stage in assuring a new system's success and giving users trust in its capacity to function and be effective.

4.1.1 Dataset

The dataset was generated by collecting images from various sources. This includes online websites(dreamstime.com,Pinterest,iStock,getty images),existing public dataset from the work, "DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning" and also manual clicking of pictures using mobile camera.The images were clicked under different lighting conditions and atmosphere. Eight weeds widely seen in Kerala were selected for our dataset. They are: Cockspur Grass,Lantana Camara,Mimosa Pudica,Oxalis Weed,Parthenium, Siam Weed,Snake Weed and Stinging Nettle.

A total of 10,400 images were collected with 1300 images of each weed.Out of these 1300 images of each weed, 900 images were taken for training. The validation dataset consist of 300 images and remaining 100 images were for test dataset.

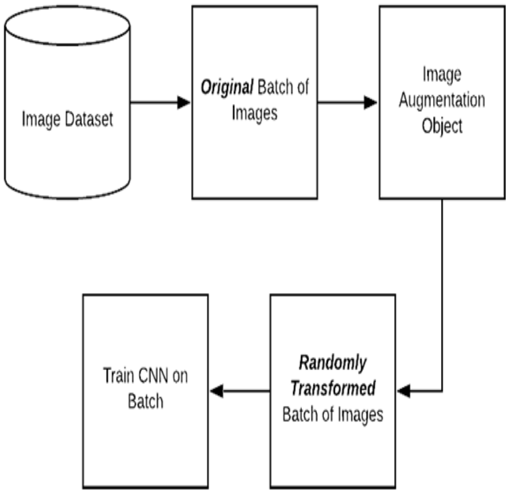
4.1.2 Data Augmentation

Image Data Generator

The Keras deep learning library provides the ability to use image augmentation automatically when training a model using the ImageDataGenerator class. Supported image formats: jpeg, png, bmp, gif. Animated gifs are truncated to the first frame. It generates batches of tensor image data with real-time data augmentation. The data will be looped over(in batches).

The target size of our RGB images are (100,100). First, we load the weed dataset created into memory and encode our labels. Then we define the ImageDataGenerator object with the Data Augmentation we want to achieve. The techniques used are:

1. Rotation
2. Shifting(Width and Height)
3. Flipping
4. Shear Intensity
5. Zoom



This process will thus lead to improvement in the neural network training phase as it makes the dataset to be more invariant to noise, reflection and rotation.The Augmentation techniques and their corresponding values used in our model is shown below.

```
training_datagen = ImageDataGenerator(
      rescale = 1./255,
      rotation_range=40,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True,
      fill_mode='nearest')
```

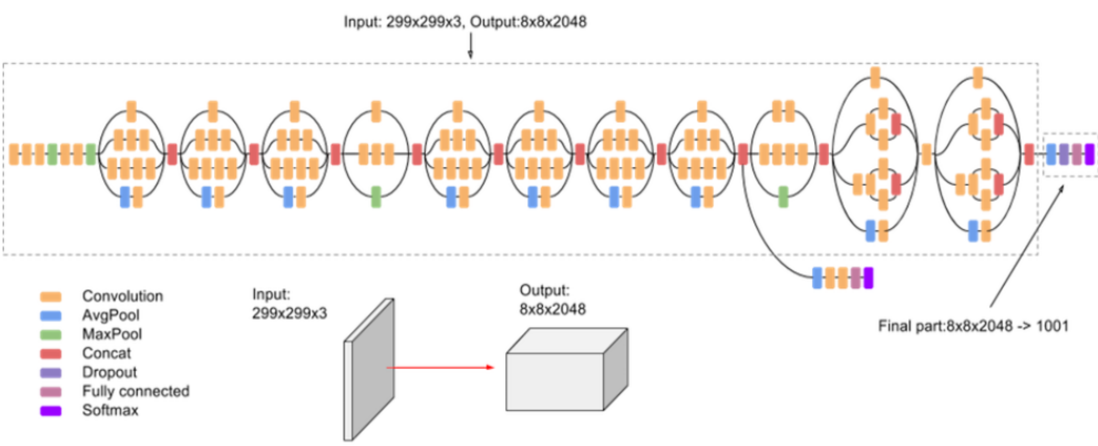
4.1.3 Classification Model

CNN Model

A 2D CNN model is implemented.The layers of a CNN consist of an input layer, an output layer and a hidden layer that includes multiple convolutional layers, pooling layers, fully connected layers and normalization layers.The dimensions of our images were 100x100. The dataset after augmentation phase is taken and is convereted to suitable form for training. This is then split into 80% training and 20% validation sets.The first required Conv2D parameter is the number of filters that the convolutional layer will learn.In the first three layers, including the input layer, 64 filters are used with (3,3) kernal padding and a max pooli.ng layer of size (2,2) .The next three layers uses 128 filters with (3,3) kernal padding.Next,flattening layer is used to flatten the matrix into a single dimensional vector.The dense layer used 512 neurons.The activation function used in all the layers was"relu" except the last layer.The output layer used "softmax" as the activation function.The model was trained using both Adam optimizer and RMSProp optimizer.

Transfer Learning

Transfer learning is a method in deep learning where a neural network is first trained on a problem similar to the problem that is being solved. It is used here to achieve better performance for our Deep Learning Model. Transfer learning helps to decrease the training time for a learning model and can also result in lower generalization error.We use the Inception v3 model here for weed classification. The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batchnorm is used extensively throughout the model and applied to activation inputs. Loss is computed via Softmax. We imported this pre-trained model and used it to train our dataset for better accuracy. A high level diagram of inception model is shown below



4.1.4 Model Evaluation

To evaluate the performance of the proposed approach, several measures are used, including the Confusion Matrix, Precision, Recall, Accuracy and F1-score. The metrics are defined by the confusion matrix. The evaluation parameters are verified by the inbuilt sklearn functions.The model was trained usind both Conventional CNN and Transfer Learning with different optimizers.

4.1.5 User Interface

We developed an Application using the Flutter framework and imported the model as Tensorflow lite into the dart code of our app.The user could take a picture using their mobile phone and the app would correctly identify the class of weed and suggest appropriate measures and home remedies for the treatment.

4.2 Libraries/Applications

Python

Python is a high-level, interpreted programming language that may be used for a variety of tasks. Python's architecture promotes code readability by making extensive use of white space. It's language elements and object-oriented approach are aimed at assisting programmers in writing clear and logical code for both small and large projects. Python is a dynamically typed programming language. It works with a variety of programming paradigms, including, structured, object-oriented and functional.

Scikit-learn

It is a machine learning library that is available for free. Python is a programming language that uses it. Support vector machines, random forests, gradient boosting, k-means, and other classification, regression, and clustering algorithms are included, and it is designed to work with the Python numerical and scientific libraries NumPy and SciPy.

TensorFlow

It is an open-source toolkit that employs data flow graphs to compute numerals and may be used for a variety of tasks, with a focus on deep neural network training and inference.

Keras

It is an open-source software library that provides a Python interface for ANN and serves as an interface for the TensorFlow library.

NumPy

It is a Python library that supports large, multi-dimensional matrices and arrays, as well as a large variety of high-level mathematical functions that may be used to manipulate such arrays.

Matplotlib

It is a charting library for the Python programming language, with Numpy as its numerical mathematic extension. It also includes an object-oriented API that can be used to embed plots into programmes using general-purpose GUI toolkits such as GTK+, Tkinter, Qt, or wxPython.

Chapter 5

Results

5.1 Sample

The user gives an input image to the model using their mobile phone. The type of Weed is predicted as output.

Figure 5.1: Identified Stinging Nettle

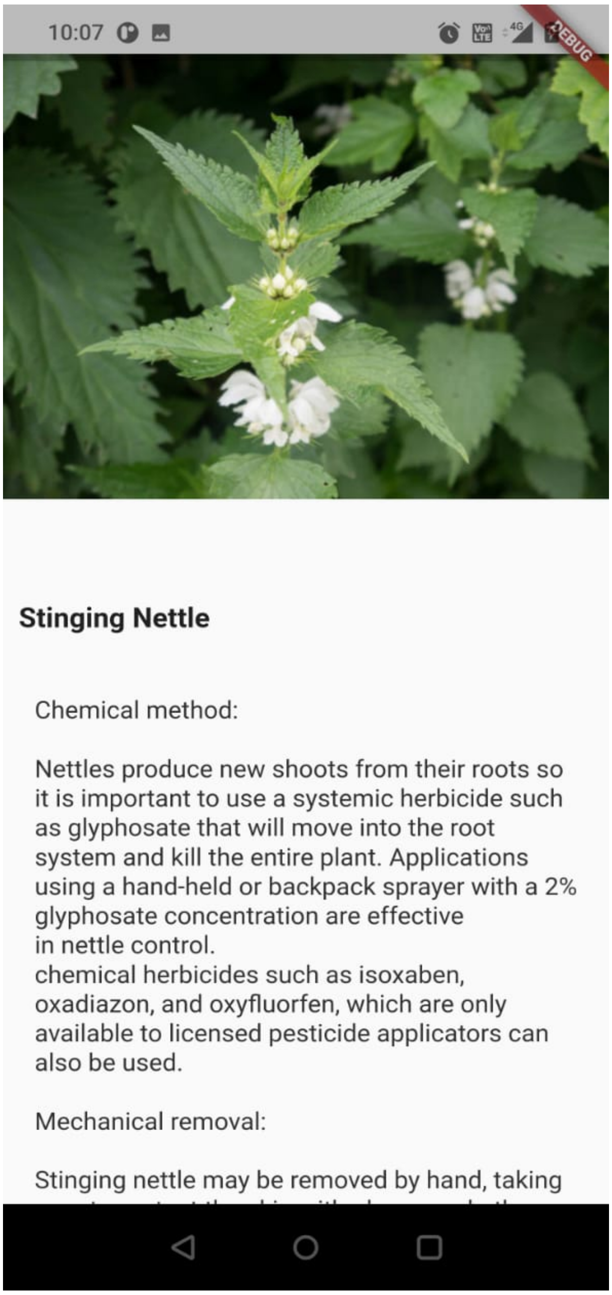


Figure 5.2: Description of Stinging Nettle

5.2 Progress Of Accuracy During Training Phase

5.2.1 CNN with six covolutional Layers

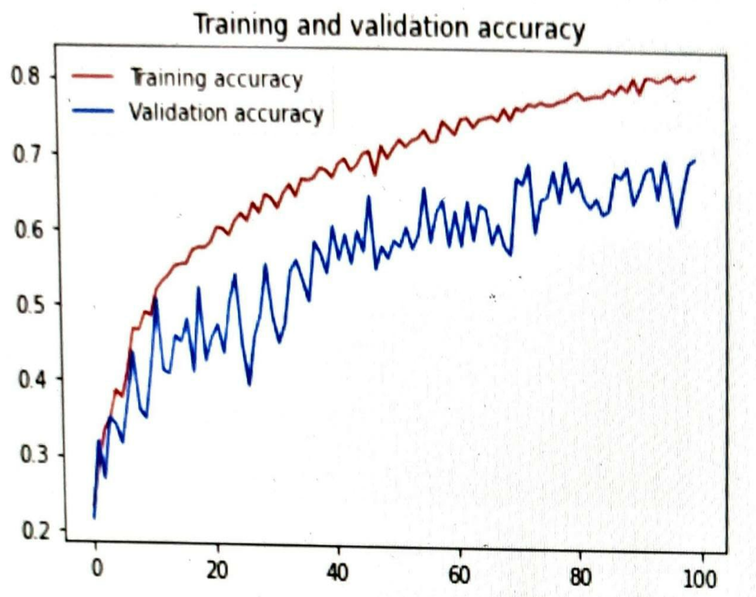


Figure 5.3: RMSprop optimizer with lr=0.001(100 epochs)

5.2.2 Transfer Learning

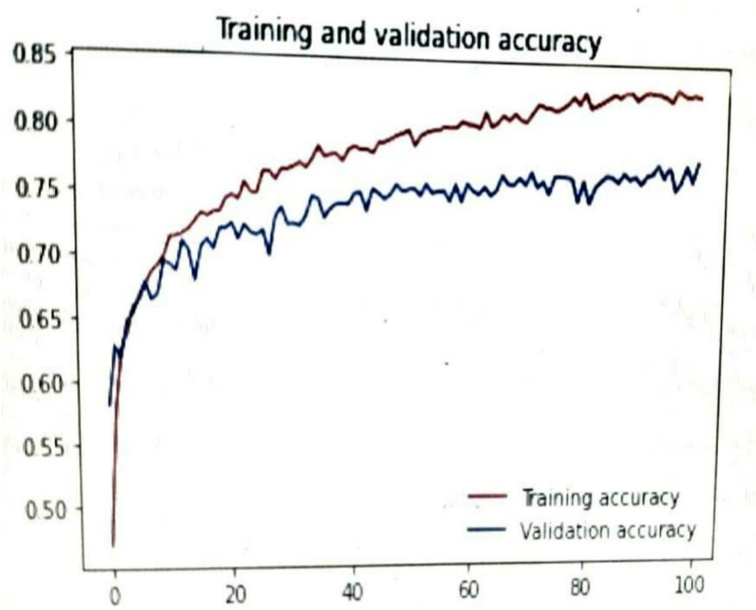


Figure 5.4: RMSprop optimizer with lr=0.001(100 epochs)

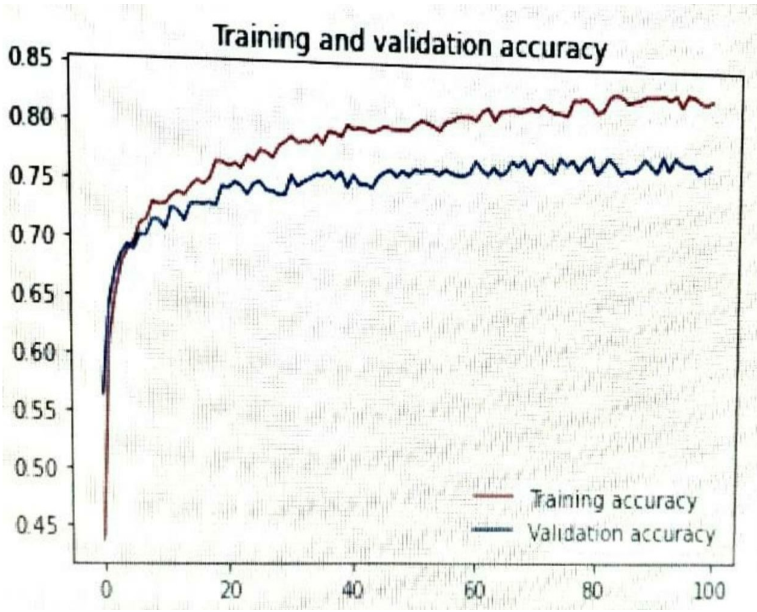


Figure 5.5: Adam optimizer with lr=0.001(100 epochs)

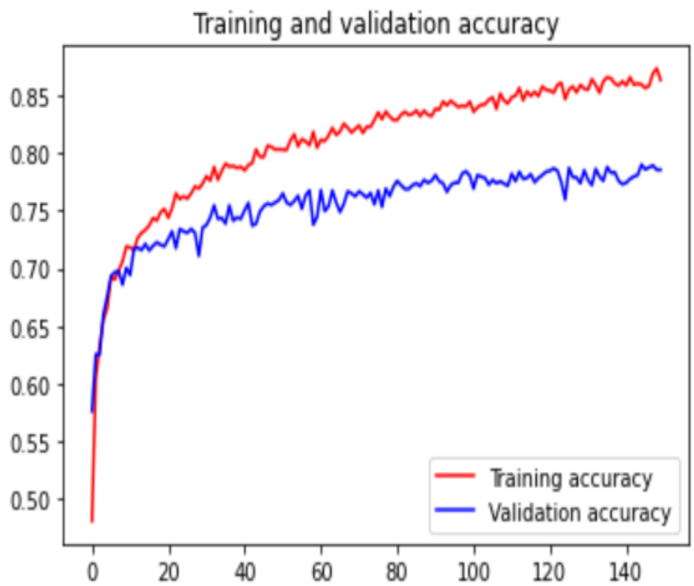


Figure 5.6: Adam optimizer with lr=0.0001(150 epochs)

We observed transfer learning with Adam Optimizer performed better than the other models.

5.3 Confusion Matrix

A confusion matrix is a synopsis of prediction results on a classification problem. The number of correct as well as incorrect predictions are reviewed with count values and pull down by each class. This is the basic to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into errors being made by a classifier but more importantly the type of errors being made. The diagonal values represents the number of outputs correctly predicted for each class and others shows the number of incorrectly predicted outputs. When normalization is done the values are confined to lie in between the range [0,1].

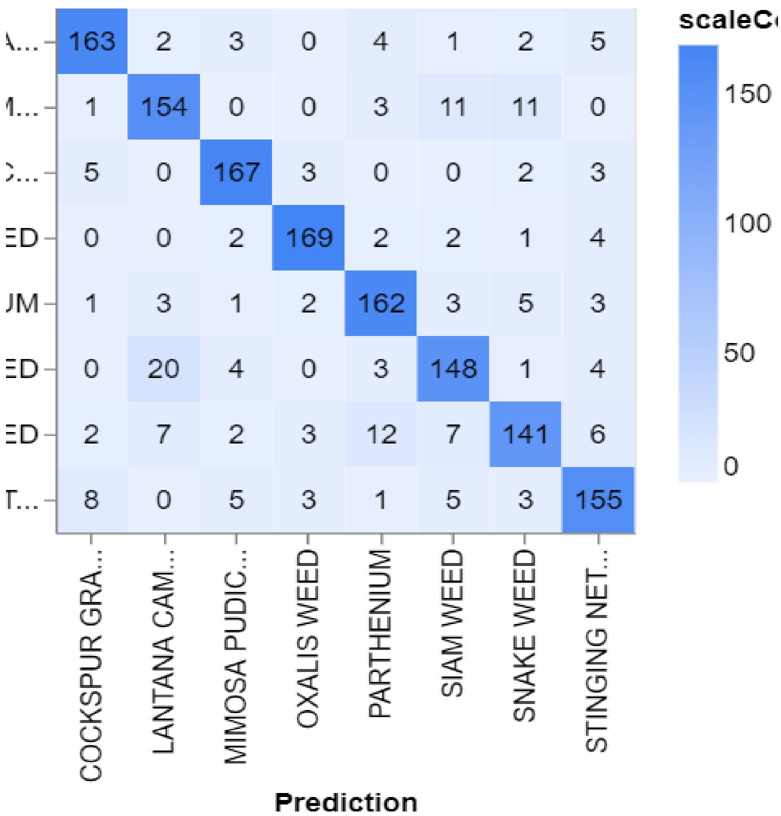


Figure 5.7: Confusion Matrix

5.4 Interpretation Of Performance Measures

The accuracy obtained per class is shown in the figure. 180 test samples were used for each weed class to calculate the accuracy. Among these, Oxalis Weed has the highest accuracy of 94% and Snake Weed has the lowest accuracy of 78%. Precision, Recall, F1-score values for each weed type is also calculated.

CLASS	ACCURACY	# SAMPLES
COCKSPUR GRASS	0.91	180
LANTANA CAMARA	0.86	180
MIMOSA PUDICA	0.93	180
OXALIS WEED	0.94	180
PARTHENIUM	0.90	180
SIAM WEED	0.82	180
SNAKE WEED	0.78	180
STINGING NETTLE	0.86	180

Figure 5.8: Accuracy per class

CLASS	PRECISION	RECALL	F1 SCORE
COCKSPUR	0.9099	0.9055	0.9054
LANTANA	0.8270	0.8556	0.8406
PARTHENIUM	0.8663	0.9000	0.8820
MIMOSA PUDICA	0.9076	0.9227	0.9176
SNAKE WEEDS	0.9388	0.9388	0.9389
SIAM WEED	0.8361	0.8222	0.8314
STINGING NETTLE	0.8493	0.7833	0.8250
OXALIS WEED	0.8720	0.8720	0.8731

Figure 5.9: Precision,Recall and F1-score

5.5 Future Scope

- More number of weed labels can be added to the dataset to identify as many weeds as possible from a photograph.
- More number of pictures can be added to each label to increase the accuracy.
- Accuracy of the model can be improved by using other transfer learning models.
- Instead of a single language, more languages can be added making it more user-friendly and accessible to more users.

5.6 Conclusion

Among the species that endanger agricultural production are weeds that, when invading crops, can cause high losses in yield and quality of yield. Therefore, in order to improve crop production and quality, agricultural weed removal is essential.

Considering that the use of agrochemicals, such as herbicides, has caused significant environmental pollution, due to its widespread use, it has become increasingly important to conduct toxicity testing of these compounds so that its use can be reduced.

An android/ios application for the classification of weeds is generated.It also gives information about the herbicide to be used against the weeds. The dataset contains pictures collected from various sources like internet, cameras etc.Those pictures are processed and the model is trained.The model correctly classifies the photograph of the weed clicked using a smartphone. It may also suggest any organic remedies existing.The project aims to support sustainable agriculture and hence less use of herbicide is preferred.

5.7 Social Relevance

Weeds can be defined as any plant or vegetation that interferes with farming or forestry purposes, such as planting crops, grazing animals or planting forests. Our aim is to reduce reliance on agrochemicals such as herbicides or pesticides due to its side effects and to detect the weeds correctly.An android/ios app named Kisan Ki Dhosth (KKD) is designed.This application clearly identifies the weed and identify the appropriate herbicide.The dataset consist of weeds which are widely seen in Kerala, making it more user-friendly for farmers.There are apps existing which detects the weeds, but they do not provide removal methods specifically for each weed.

5.8 Program Code

```
#Code using inception transfer learning with Adam optimizer
import os
import zipfile
import tensorflow as tf
import keras_preprocessing
from keras_preprocessing import image
from keras_preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt

from tensorflow.keras import layers
from tensorflow.keras import Model

# Using Transfer learning to build the model.
!wget --no-check-certificate \
https://storage.googleapis.com/mledu-datasets/inception_v3_weights
_tf_dim_ordering_tf_kernels_notop.h5 \
-O /tmp/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5

from tensorflow.keras.applications.inception_v3 import InceptionV3

local_weights_file ='/tmp/inception_v3_weights_
tf_dim_ordering_tf_kernels_notop.h5'

pre_trained_model = InceptionV3(input_shape = (100, 100, 3),
                                include_top = False,
                                weights = None)

pre_trained_model.load_weights(local_weights_file)
for layer in pre_trained_model.layers:
  layer.trainable = False
last_layer = pre_trained_model.get_layer('mixed7')
```

```
print('last layer output shape: ', last_layer.output_shape)
last_output = last_layer.output

from tensorflow.keras.optimizers import Adam

#Flatten the output layer to 1 dimension
x = layers.Flatten()(last_output)
#Add a fully connected layer with 1,024 hidden units and ReLU activation
x = layers.Dense(1024, activation='relu')(x)
#Add a dropout rate of 0.2
x = layers.Dropout(0.2)(x)
x = layers.Dense (8, activation='softmax')(x)
model = Model( pre_trained_model.input, x)
model.compile(optimizer = Adam(0.0001),
              loss = 'categorical_crossentropy',
              metrics = ['accuracy'])



from google.colab import drive
drive.mount("/content/drive")

local_zip = '/content/drive/MyDrive/WEED/TRAIN-WEED.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('/tmp/')
zip_ref.close()
local_zip = '/content/drive/MyDrive/WEED/VALIDATION-WEED.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('/tmp/')
zip_ref.close()

TRAINING_DIR = "/tmp/TRAIN-WEED/"
training_datagen = ImageDataGenerator(
      rescale = 1./255,
```

```
    rotation_range=40,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True,
      fill_mode='nearest')
VALIDATION_DIR = "/tmp/VALIDATION-WEED/"
validation_datagen = ImageDataGenerator(rescale = 1./255)
train_generator = training_datagen.flow_from_directory(
TRAINING_DIR,
target_size=(100,100),
class_mode='categorical',
    batch_size=320
)
validation_generator = validation_datagen.flow_from_directory(
VALIDATION_DIR,
target_size=(100,100),
class_mode='categorical',
  batch_size=200
)

class myCallback(tf.keras.callbacks.Callback):
  def on_epoch_end(self, epoch, logs={}):
    if(logs.get('accuracy')>0.98):
      print("\nReached 95% accuracy so cancelling training!")
      self.model.stop_training = True



callbacks = myCallback()
history = model.fit(train_generator,
                    epochs=150,
                    steps_per_epoch=len(train_generator),
```

```
                    validation_steps=len(validation_generator),
                    verbose = 1,
                    validation_data = validation_generator,
                     callbacks=[callbacks]
                    )

#To plot the training and validation accuracy/loss.
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(len(acc))
plt.plot(epochs, acc, 'r', label='Training accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
plt.title('Training and validation accuracy')
plt.legend(loc=0)
plt.figure()
plt.show()
```

Bibliography

[1] P. Lottes, J. Behley, A. Milioto and C. Stachniss, "Fully Convolutional Networks With Sequential Information for Robust Crop and Weed Detection in Precision Farming," in IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 2870-2877, Oct. 2018, doi: 10.1109/LRA.2018.2846289.

[2] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[3] N. Chebrolu, P. Lottes, A. Schaefer, W. Winterhalter, W. Burgard and C. Stachniss, "Agricultural robot dataset for plant classification localization and mapping on sugar beet fields", Int. J. Robot. Res., vol. 36, no. 10, pp. 1045-1052, 2017.

[4] S. Haug, A. Michaels, P. Biber and J. Ostermann, "Plant classification system for crop/weed discrimination without segmentation", Proc. IEEE Winter Conf. Appl. Comput. Vis., pp. 1142-1149, 2014.

[5] P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart and C. Stachniss, "UAV-Based crop and weed classification for smart farming", Proc. IEEE Int. Conf. Robot. Autom., pp. 3024-3031, 2017.

[6] C. S. McCool, T. Perez and B. Upcroft, "Mixtures of lightweight deep convolutional neural networks: Applied to agricultural robotics", IEEE Robot. Autom. Lett., vol. 2, no. 3, pp. 1344-1351, Jul 2017.

[7] K. Mortensen, M. Dyrmann, H. Karstoft, R. N. Jörgensen and R. Gislum, "Semantic segmentation of mixed crops using deep convolutional neural network", Proc. Int. Conf. Agricultural Eng., 2016.

[8] .A. Wendel and J. P. Underwood, "Self-Supervised weed detection in vegetable crops using ground based hyperspectral imaging", Proc. IEEE Int. Conf. Robot. Autom., pp. 5128-5135, 2016.

[9] M. D. Cicco, C. Potena, G. Grisetti and A. Pretto, "Automatic model based dataset generation for fast and accurate crop and weeds detection", Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 5188-5195, 2017.