**Classification of Crops and Weeds using Deep Learning.**

A Project work submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS & COMMUNICATION ENGINEERING

By

V. Vishweshwar Reddy 17211A04P4

Y. Shashank Reddy 17211A04P9

K. Rakha Chandre 17211A04K7

V. Vamshidhar 17211A04P7

Under the esteemed guidance of

**Dr. Syed Abudhagir Umar., M.E, PhD**

Associate Professor

****

**Department of Electronics and Communication Engineering**

**B. V. Raju Institute of Technology, Autonomous**

**Vishnupur, Narsapur, Medak(Dt)**

**(Affiliated to JNTU, Hyderabad)**

**2021**

**B. V. Raju Institute of Technology**

**Vishnupur, Narsapur, Medak(Dt) Pin:502313**

**(Affiliated to JNTU, Hyderabad)**

**Ph: 08458-222000, 222001 Fax: 08458-222002**

**Department of Electronics & Communication Engineering**



**CERTIFICATE**

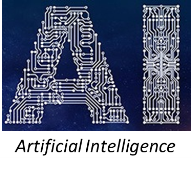
This is to certify that the Project work entitled **Classification of Crops and Weeds using Deep Learning** is being submitted by Mr. V. Vishweshwar Reddy**,** Mr. Y. Shashank Reddy, Mr. K. Rakha Chandre, Mr. V. Vamshidhar in partial fulfillment of the requirement for the award of the degree of **B.Tech. in Electronics & Communication Engineering**, by Jawaharlal Nehru Technological University Hyderabad is a record of bonafide work carried out by him under my guidance and supervision from 2020 to 2021.

The results presented in this project have been verified and are found to be satisfactory.

|  |  |
| --- | --- |
| **Coordinator** | **HOD** |
| **Dr. Syed Abudhagir Umar, Ph.D** | **Dr. Sanjay Dubey, Ph. D** |

**EXTERNAL EXAMINER**

Logo, company name

Description automatically generated

CENTER FOR EXCELLENCE

IN

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

**CERTIFICATE**

This is to certify that thefollowing students of B V Raju Institute of Technology have undertaken the project titled “**Classification of Crops and Weeds using Deep Learning ”** at the Centre for Excellence In Artificial Intelligence and Machine Learning under the esteemed guidance of **“Dr. Syed** **Abudhagir Umar”,** Department of ECE, from 2020 to 2021 and have completed the project successfully.

**TEAM MEMBERS: REGISTERED NUMBERS:**

1. V. Vishweshwar Reddy 17211A04P4
2. Y. Shashank Reddy 17211A04P9
3. K. Rakha Chandre 17211A04K7
4. V. Vamshidhar 17211A04P7

The results presented in this project have been verified and are found to be satisfactory.

|  |  |
| --- | --- |
| **Coordinator** | **HOD** |
| **Dr. Syed Abudhagir Umar, Ph.D** | **Dr. Sanjay Dubey, Ph. D** |

## ACKNOWLEDGEMENTS

We take opportunity to express our indebt gratitude to the persons who contributed for our work, for being our inspiration and guide which led to the successful completion of the project.

We are grateful towards our College Management and our beloved Principal

**Dr. K. Lakshmi Prasad,** for providing us the necessary infrastructure and facilities that ensured smooth and satisfactory execution of the project.

We would like to express out profound gratitude to our Head of the department **Dr. Sanjay Dubey**, **M.E, PhD, MISTE, MIEEE** Dept. of ECE, for his encouragement inspiration and close monitoring and guidance he gave us during the execution of the project.

We express our sincere thanks to **Dr Syed Abudhagir Umar M.E, Phd,** Associate Professor, Dept. of ECE, our guide for his valuable suggestion and motivation in successful completion of project.

We also wish to express our thanks to all the faculty members and laboratory staff who were helpful directly and indirectly for the completion of the project.

V. Vishweshwar Reddy 17211A04P4

Y. Shashank Reddy 17211A04P9

K. Rakha Chandre 17211A04K7

V. Vamshidhar 17211A04P7

**DECLARATION**

We hereby declare that the project entitled “**Classification of Crops and Weeds using Deep Learning**”submitted to **B. V. Raju Institute of Technology,** affiliated to Jawaharlal Nehru Technological University, Hyderabad for the award of the Degree of Bachelor of Technology in Electronics and Communication Engineering is a result of original research done by me.

It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma.

|  |  |
| --- | --- |
| 1. Vishweshwar Reddy 17211A04P4 | 1. K. Rakha Chandre 17211A04K7   A picture containing text  Description automatically generated |
| 1. Shashank Reddy 17211A04P9   Text, letter  Description automatically generated | 1. V. Vamshidhar 17211A04P7   A whiteboard with writing on it  Description automatically generated with medium confidence |

## ABSTRACT

Deep learning is the nucleus in machine learning discipline which uses knowledge representation of learning. Learning can be supervised or unsupervised. Much Deep learning architecture are available which includes deep belief networks, deep neural networks and recurrent neural networks of which it has been applied to most of the fields. The commonly used applications of deep learning are vision related, audio, video, language processing, social media, medical, game and many more programs where they have produced promising accurate results comparable to and in few cases superior to human experts. Smart agriculture is an area that can benefit from the latest advances in expert systems. One of the objective is to remove the weeds by reducing the use of herbicides used, the risk of pollution of crop and water. The image of crop field is given as input training examples. By using the extracted feature, the images with weeds are detected and classified. A deep learning model is developed using convolution neural network to detect weeds with a good accuracy so that the model could be used to detect the weeds in the cucumber crop field in a shorter time.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| 1 | **PROJECT TITLE**  **CERTIFICATE PAGE**  **SPECIAL LAB CERTIFICATE**  **ACKNOWLEDGEMENTS**  **DECLARATION**  **ABSTRACT**  **INTRODUCTION**   * 1. INTRODUCTION   2. PROBLEM DEFINITION   1.3 OBJECTIVE | i  ii  iii  iv  v  vi  1  2  3 |
| 2 | **LITERATURE SURVEY**  2.1 INTRODUCTION  2.2 WEED DETECTION  2.3 CROP DETECTION | 4  5  6 |
|  | **REQUIREMENTS**  3.1 REQUIREMENTS  SPECIFICATION  3.2 HARDWARE REQUIREMENTS  3.3 SOTWARE REQUIREMENTS | 8  9 |
| 4 | **CROP & WEED DETECTION AND CLASSIFICATION**  4.1 INTRODUCTION  4.2 BLOCK DIAGRAM  4.3 CNN  4.3.1 CONVOLUTION LAYER  4.3.2 POOLING LAYER  4.3.3 FLATTEN LAYER  4.4.3 FULLY CONNECTED LAYER  4.4 DATASET  4.5 FLOW DIAGRAM | 10  11  13  18  20 |
| 5 | **WEB FRAMEWORK AND APPLICATIONS**  5.1 WEB FRAMEWORK  5.2 APPLICATIONS | 21  22  23 |
| 6 | **RESULTS AND DISCUSSIONS**  6.1 ACCURACY TABLES  6.2 ACCURACY AND LOSS GRAPHS  6.3 OUTPUTS & PREDECTIONS | 24  25  26  28 |
| 7 | **CONCLUSION & FUTURE WORKS**  7.1 CONCLUSION  7.2 FUTURE WORKS | 30  31 |
| 8 | **REFERENCES**  **APPENDIX** | 32  33 |
|  |  |  |

\

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURES** |  | **PAGE NO.** |
| FIG 4.3 | *Convolution Layers* | 14 |
| FIG 4.3.2 | *Pooling Layer* | 16 |
| FIG 4.3.3 | *Flatten Layer* | 17 |
| FIG 4.3.4 | *Fully connected layer* | 17 |
| FIG 4.4(a) | *Sample image data set* | 18 |
| FIG 4.4(b) | *RGB of image dataset* | 19 |
| FIG 4.5(a) | *Flow Diagram* | 19 |
| FIG 4.5(b) | *Sequence Diagram* | 20 |
| FIG 6.1(a) | *Accuracy and Loss using Adam Optimizer* | 25 |
| FIG 6.1(b) | *Accuracy and Loss using RSMProm Optimizer.* | 26 |
| FIG 6.2(a) | *Accuracy Graph using Adam* | 27 |
| FIG 6.2(b) | *Loss Graph using Adam* | 27 |
| FIG 6.3(a) | *Output Page* | 27 |
| FIG 6.3(b) | *Fig: agri\_0\_057* | 28 |
| FIG 6.4© | *Crop Result* | 28 |
| FIG 6.4(d) | *Fig: agri\_0\_084* | 29 |
| FIG 6.5(e) | *Weed Result* | 29 |

**CHAPTER 1**

**INTRODUCTION**

**1.INTRODUCTION**

**1.1MOTIVATION**

One of the objectives are to remove the weeds by reducing the use of herbicides used, the risk of pollution of crop and water. The image of crop field is given as input training examples.

In order to predict weeds and plants data set is collected and trained using Fully connected layers.

In proposed system we are collecting dataset of weeds and plants for training purpose. Data set is pre-processed, tested, trained and spitted and CNN model is used to train dataset and model is saved to system.

**1.2 PROBLEM DEFINITION**

The unstructured data usually could take a long duration for us to analyse it and mine relevant information which will be used for the analysis of the system. The corporate identify the incredible potential that can result from wasting this valuable information and are increasingly adapt to modern intelligence systems. In order to process the huge volume of data and extract the unique patterns the most common Artificial Intelligence techniques used for processing the data is machine learning algorithms.

Machine Learning provides many categories of algorithms like prediction, regression, clustering etc that generate a high-precision analysis report and patterns with historical data or newly generated data. If medical company wants to detect the disease or occurrence of a specific disease pattern or potential for fraud in its system like forgery of insurance documents, originality of drugs etc it could employ machine learning algorithms and tools for this purpose. The computational algorithm employed into the data analytics system will process all the kind of data in that particular domain and it extracts patterns from the historical data and detects any anomalies in the system.

Deep learning domain is a subset of machine learning and follows a hierarchical level of artificial neural networks to do the process of pattern identification. The artificial neural networks imitates the functionality of the human brain and it contains millions neuron nodes connected to each other like a web. There is a fundamental difference in the way that how the traditional algorithms process the data and the Machine learning algorithms process the data. A nonlinear approach of data processing is employed in Machine Learning and deep learning based application development.

The deep learning techniques uses a nonlinear approach which takes many attributes for processing in parallel. In the case of anomaly/fraud detection in a network based applications it takes many parameters as input which includes time of occurrence of the event, geographical location information, identification n of the device like IP address and Hardware address, type of events and many other feature that is likely to point to the illegal activity.

**1.3 OBJECTIVE**

One of the primary **objectives** in the field of precision agriculture is **weed** detection. Detecting and expunging weeds in the initial stages of **crop** growth with deep learning technique can minimize the usage of herbicides and maximize the **crop** yield for the farmers.

The development of the project will be done considering the following points:

• Research on deep learning and its implementation in CNN

• Develop the code to take the pictures from dataset.

• Develop the CNN code.

• Evaluation and comparison of the optimizers.

To summarize, the project will be considered complete when the neural network achieves an accuracy of more than the 50% when identifying both crop and weeds in an image captured by the FarmBot. Those crops and weeds used for the project are spinach, dandelions and cleavers.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 INTRODUCTION**

The study on CNN has expanded thoroughly and rapidly in recent times in the field of agriculture and many researchers have many terms to describe the combining models involving different algorithms. Ciro Potena, Daniele Nardi, and Alberto Pretto designed a robotics system for automating certain agriculture activities that employ an unmanned ground vehicle (UGV) provided with a high resolution camera of multi spectral capability. This would help to carry out crop/weed detection and classification tasks in the agriculture field without human intervention. Their design explores a channel that consists of two different architectures of convolutional neural networks (CNNs) connected to the input RGB plus Near Infra-Red (NIR) images.

The lightweight version of CNN is used to achieve a fast, scalable, robust, pixel-wise, binary image segmentation, in order to extract the pixels that shows projections of 2D/3D points which belong to green vegetation category. A deeper CNN architecture could be employed to classify the extracted pixels between the two prominent crop and weed classes. A further added important contribution of this work is a new kind of dataset summarization algorithm. The purpose of this algorithm is to select the most informative subsets from a large dataset that better describe the original one automatically. This would enable us to speed up the manual labeling process of the images collection in the dataset. Nima Teimouri, Mads Dyrmann, Per Rydahl Nielsen, Solvejg Kopp Mathiassen,Gayle J. Somerville,and Rasmus Nyholm Jørgensen presented paper on weed identification using CNN. They outlined a new method which estimates the weed species and growth stages of these in some images automatically.

According to various environmental conditions, image of weeds that are grown in the same type of crops are collected. The parameters that are considered may be the types of soils, dark light, medium light and resolution are considered. A set of images are taken for training out of which it separates into nine classes. Then the next sets of images are taken to evaluate the performance of the proposed network. Everything is differentiated with that of the parameters considered. Maximum accuracy of 78% is obtained for the Polygonum species and 46 % for blackgrass. Obtained about 70% accuracy for finding the number of leaves. The new method which is proposed has a good high ability to estimate the weed species.

**2.2 Weed Detection**

Agriculture has always been an essential activity for survival. Over the last century, and more specific, over the last 15 years, agriculture has started to mechanise and digitise; due to this evolution and automation, labour flow was almost totally standardised. Nowadays, after introducing robotics and artificial intelligence into agriculture there is no need of standardization, robots are working collaboratively with humans and learning from them how to realize the basic agriculture tasks such as weed detection, watering or seeding (Marinoudi, et al., 2019).

Weed detection is one of those basic agriculture tasks that are being automatized and digitised, in this case, because of toxicity related to herbicides; so, reducing human intervention will make possible a decrease in the use of herbicides, increasing health care. To achieve this, robots able to detect plants and classify them into crop or weed are now introduced into agriculture (Dankhara, et al., 2019). IoT is present in the communication between a Raspberry Pi, where the processing is done and the camera and sensors are connected, and the Data Server, where the Raspberry Pi sends the information obtained. This paper shows an accuracy of 90%-96% depending on if it is used a Convolutional Neural Network (CNN), a datasheet is being created or it is being used the training set. Daman, et al., (2015) and Liang, et al., (2019) both introduce the use of automation into agriculture to identify weeds, and to do so, they make use of image processing techniques.

. Results were more than successful, after placing plants and weeds randomly, the robot was tested and weeds were almost totally identified and sprayed, taking the processing stage approximately 3 seconds. Liang, et al., (2019) implement image processing in drones instead of robots, that way, they not only detect weeds, but also monitor the growth of crops. By combining image processing and CNN in drones, they get different accuracies depending on the processing, which is from 98.8% with CNN to 85% using Histograms of Oriented Gradients (HOG). All the previously mentioned processes can be done either in static by photos or in real-time by videos. Marzuki Mustafa, et al. (2007), have done a research about the implementation of a real-time.

The crop is recorded and processed, offline, using various image processing techniques and a new developed algorithm that respond correctly to real time conditions. Finally, they achieved an accuracy over the 80%. Not only the weed as a plant can be differentiated, more advanced studies such as Wafy, et al., (2013), differentiate the weeds seeds using Scale-Invariant Feature Transform (SIFT), an algorithm that extracts the interest points from an image; by using this technique, the minimum accuracy they have is 89.2%.

**2.3 Crop Detection:**

Deep Learning neural networks range from deep neural networks, deep belief networks, recurrent neural networks and CNNs. The most usually used are CNN, whose layers apply convolutional filters to the inputs. The networks are rarely created from scratch and most of the ones used on projects are already existing networks such as LeNet, AlexNet, GoogleNet, SNET or CNET (Moazzam, et al., 2019). Moazzam, et al., (2019) offers a summary of seven different studies, all of them use deep learning convolutional networks approaches for the weed/crop identification problem, as shown in Table 1. Even if all the papers mentioned focus on different types of crops, a common element is that most of them only focus on one crop. Studies using deep learning identification of multiple crops and weeds are not common.

Crop and weed detection with the use of deep learning is not yet a usual topic of research, even if there are more and more attempts. There are still many research gaps not considered like the differentiation of different crops and weed combinations. Furthermore, even some major essential crops are lacking in this kind of investigation, as there is still a need of creating big datasets for these crops. Deep learning is still a new a tool for the autonomous agricultural applications, yet it seems to be a promising technique and more accurate than other approaches.

From these researches the needed knowledge about the necessary pre-processing techniques that will be used in this project has been acquired; some of these are filtering, binarization and histograms, a deeper study on them will be done during the development to make sure they suit correctly. Also, through the study of ANNs, some projects using CNNs have been found, being one of those nets AlexNet, the one chosen for this project; by this research, a vision on how to work with these nets has been acquired, as well as the accuracy expected in this kind of projects.

**CHAPTER 3**

**REQUIREMENTS**

**3.1 REQUIREMENTS SPECIFICATION**

Requirements Specification provides a high score to perform training, testing and detection of model efficiently. Software Requirements, Hardware Requirements deal with software, hardware resources respectively that need to be installed on a system to provide optimal functionality of the application. These are necessary before installing dependencies or packages in a system. These requirements are the most common set of requirements defined by any operating system and provide a compatible support to the operating system in developing this tool.

**3.2 HARDWARE REQUIREMENTS**

System Intel core i3. Hard Disk 200 GB. Monitor 15 VGA Color. RAM 2 GB.

**3.3 SOFTWARE REQUIREMENTS**

Operating system Windows XP/7/10

Coding Language Python Development Kit Anaconda navigator

Web Framework Flask

Library TensorFlow, keras OpenCV, Matplot, Numpy, scikit learn

Dataset size 1046 Training Images (2 classes) & 600 Testing Images (2 classes)

**CHAPTER 4**

**Crop and Weed Detection and Classification**

**4.1 Introduction**

The initial layer i.e the input layer of the neural network processes takes the raw input data contains various attributes of variety of data. The input layer passes it on to the next level hidden layers and finally it will be sent to a output layer. Deep learning networks not only takes the text data in the tabular column format and produces an high accuracy output, it also takes the images, audios, videos and time series data and promises a high accuracy output.

The social media Facebook Research division head and Father of network Architecture YannLeCun uses a new architecture which is good at object recognition in image dataset called the Convolutional Neural Network (CNN). The convolutional technique is showing a great success in image processing like multilayer perceptron feed forward neural networks. Also this technique is capable of scaling with data and model size and the model could be well trained with back propagation algorithm. This fundamental idea and the requirement leads to the significance of deep learning as the development of CNNs with large number of layers, which has promised to produce a high accurate detection and classification rate on image and video content. Deep learning methods have several well-defined computational models, which consist of multiple computational layers to learn representations of input data with multiple abstraction levels.

Image recognition is another interesting area of application. The images are represented as a 2D array of pixels. Each pixel with RGB channels or gray scale is feed directly into a convolutional neural network which is trained end-to-end. A CNN consists of alternating layers of convolutions called convolutional layers, and also it contains pooling and sub sampling layers.

**4.2 BLOCK DIAGRAM**

Image Acquisition

Feature Extraction

Pre Processing

Feature Extraction

Classification

Identification

**IMAGE ACQUISION:**

The first step of the process is the Acquisition of the image, Digital imaging or digital image acquisition is the creation of a representation of the visual characteristics of an object, such as a physical scene or the interior structure of an object. The term is often assumed to imply or include the processing, compression, storage, printing, and display of such images.

The **image acquisition** process consists of three steps;

* energy reflected from the object of interest,
* an optical system which focuses the energy
* a sensor which measures the amount of energy.

**PRE-PROCESSING:**

In the next step, the Image pre-processing is the name for operations on images at the lowest level of abstraction whose aim is an improvement of the image data that suppress undesired distortions or enhances some image features important for further processing. ... Image pre- processing use the redundancy in images. As a Machine Learning Engineer, data pre-processing or data cleansing is a crucial step and most of the ML engineers spend a good amount of time in data pre-processing before building the model. Some examples for data pre-processing includes outlier detection, missing value treatments and remove the unwanted or noisy data.

Similarly, Image pre-processing is the term for operations on images at the lowest level of abstraction. These operations do not increase image information content but they decrease it if entropy is an information measure. The aim of pre-processing is an improvement of the image data that suppresses undesired distortions or enhances some image features relevant for further processing and analysis task.

**FEATURE EXTRACTION:**

Feature Extraction includes the feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction.

When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features (also named a feature vector). Determining a subset of the initial features is called feature selection. V The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

**IDENTIFICATION:**

Image recognition is the process of identifying an object or a feature in an image or video. It is used in many applications like defect detection, medical imaging, and security surveillance.

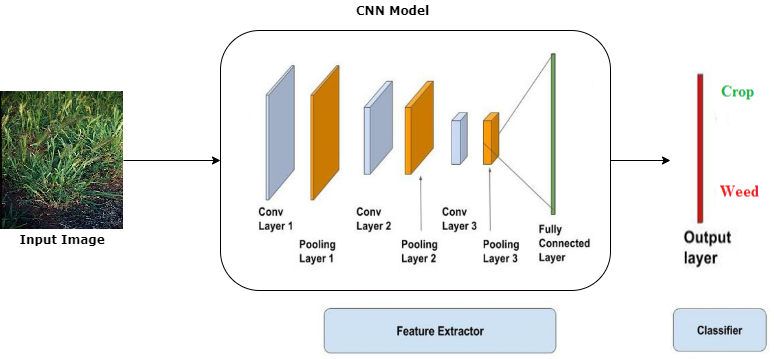
**CLASSIFICATION**

And finally, Image classification refers to the task of extracting information classes from a multiband raster image. The resulting raster from image classification can be used to create thematic maps.

**4.3 CNN**

A **Convolutional Neural Network** (CNN) is a kind of Artificial Neural Network (ANN) used in image detection, classification and processing that is specially developed to process images and videos. The Neural Networks with multiple layers are powerful image analysis and artificial intelligence technique that use deep networks to achieve both generative and descriptive tasks which often using computer vision that includes both collection of images and streaming videos. The multilayer ANN functions like a system of both hardware and software patterned after the process of neurons in our brain. The various earlier methods of artificial networks are not efficient for handling images and those networks enforces the input images should be downsized and its resolutions gets reduced. CNN adopts a methodology in which their neurons are organized in a way like those of the frontal lobe which is the region accountable for meting out visual stimuli in humans and other animals. The inner layers of neurons in our brain are assembled in such a way as to cover the complete visual field which could stay away from the piecemeal image analyzing problem of the earlier conventional neural networks.

**ARCHITECTURE:**



Diagram

Description automatically generated with medium confidence

*Fig 4.3: Convolution Layers*

**Layers Of Convolution Neural Network:**

**4.3.1 CONVOLUTIONAL LAYER:**

Convolution layers are major building blocks used in convolution neural networks. A convolution is the simple application of a filter to an input that results in an activation. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image.

The innovation of convolutional neural networks is the ability to automatically learn a large number of filters in parallel specific to a training dataset under the constraints of a specific predictive modeling problem, such as image classification. The result is highly specific features hat can be detected anywhere on input images.

The convolutional neural network, or CNN for short, is a specialized type of neural network model designed for working with two-dimensional image data, although they can be used with one-dimensional and three-dimensional data.Central to the convolutional neural network is the convolutional layer that gives the network its name. This layer performs an operation called a convolution.

In the context of a convolutional neural network, a convolution is a linear operation that involves the multiplication of a set of weights with the input, much like a traditional neural network. Given that the technique was designed for two-dimensional input, the multiplication is performed between an array of input data and a two-dimensional array of weights, called a filter or a kernel.

The filter is smaller than the input data and the type of multiplication applied between a filter-sized patch of the input and the filter is a dot product. A dot product is the element-wise multiplication between the filter-sized patch of the input and filter, which is then summed, always resulting in a single value. Because it results in a single value, the operation is often referred to as the scalar product.

Using a filter smaller than the input is intentional as it allows the same filter (set of weights) to be multiplied by the input array multiple times at different points on the input. Specifically, the filter is applied systematically to each overlapping part or filter-sized patch of the input data, left to right, top to bottom.

This systematic application of the same filter across an image is a powerful idea. If the filter is designed to detect a specific type of feature in the input, then the application of that filter systematically across the entire input image allows the filter an opportunity to discover that feature anywhere in the image. This capability is commonly referred to as translation invariance, eg. the general interest in whether the feature is present rather than where it was present.

**Rectified Linear Units (ReLU):**

After each conv layer, it is convention to apply a nonlinear layer (or **activation layer**) immediately afterward.

The purpose of this layer is to introduce nonlinearity to a system that basically has just been computing linear operations during the conv layers (just element wise multiplications and summations).In the past, nonlinear functions like tanh and sigmoid were used, but researchers found out that **ReLU layers** work far better because the network is able to train a lot faster (because of the computational efficiency) without making a significant difference to the accuracy.

It also helps to alleviate the vanishing gradient problem, which is the issue where the lower layers of the network train very slowly because the gradient decreases exponentially through the layers (Explaining this might be out of the scope of this post, but seen here for good descriptions).

The ReLU layer applies the function f(x) = max(0, x) to all of the values in the input volume. In basic terms, this layer just changes all the negative activations to 0. This layer increases the nonlinear properties of the model and the overall network without affecting the receptive fields of the convolution layer.

**SYNTAX:**

classifier.add(Convolution2D(filters,(kernalsize\_matrix),input\_shape=(image\_width,image\_height,Colour\_of\_image),activation=’relu’))

**4.3.2 POOLING LAYER:**

It is also referred to as a downsampling layer.

In this category, there are also several layer options, with **maxpooling** being the most popular. This basically takes a filter (normally of size 2x2) and a stride of the same length. It then applies it to the input volume and outputs the maximum number in every subregion that the filter convolves around.

A picture containing chart

Description automatically generated

Fig 4.3.2: Pooling Layer

The reason behind this layer is that once we know that a specific feature is in the original input volume (there will be a high activation value), its exact location is not as important as its relative location to the other features. As you can imagine, this layer drastically reduces the spatial dimension (the length and the width change but not the depth) of the input volume. This serves two main purposes. The first is that the amount of parameters or weights is reduced by 75%, thus lessening the computation cost. The second is that it will control **overfitting**. This term refers to when a model is so tuned to the training examples that it is not able to generalize well for the validation and test sets. A symptom of overfitting is having a model that gets 100% or 99% on the training set, but only 50% on the test data.

**SYNTAX:** classifier.add(MaxPooling2D(pool\_size =(pool\_size\_matrix)))

**4.3.3 FLATTEN LAYER:**

After finishing the previous two steps, we're supposed to have a pooled feature map by now. As the name of this step implies, we are literally going to flatten our pooled feature map into a column like in the image below.

The reason we do this is that we're going to need to insert this data into an artificial neural network later on.

Graphical user interface

Description automatically generated

Fig 4.3.3: Flatten Layer

**SYNTAX:** classifier.add(Flatten())

**4.3.4 Fully Connected Layer:**

The **dense layer** is a neural network layer that is connected deeply, which means each neuron in the dense layer receives input from all neurons of its previous layer. The dense layer is found to be the most commonly used layer in the models.

In the background, the dense layer performs a matrix-vector multiplication. The values used in the matrix are actually parameters that can be trained and updated with the help of backpropagation.

The output generated by the dense layer is an ‘m’ dimensional vector. Thus, dense layer is basically used for changing the dimensions of the vector. Dense layers also applies operations like rotation, scaling, translation on the vector

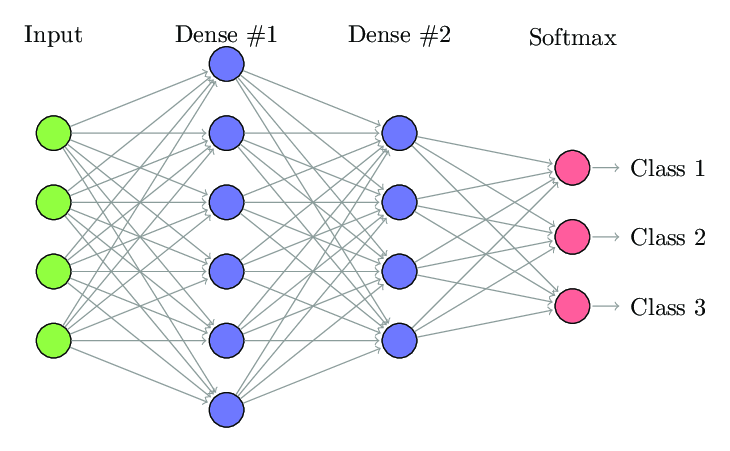


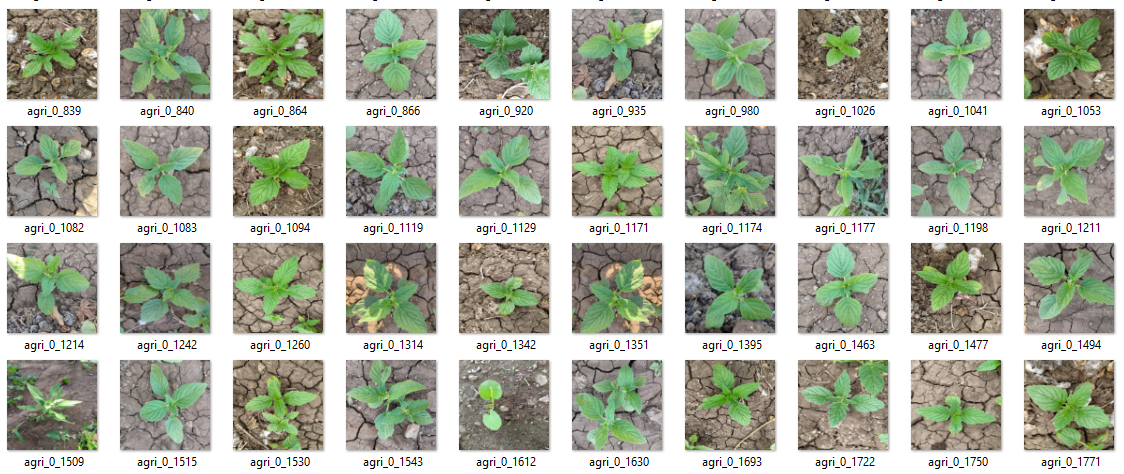
Fig 4.3.4: Fully connected Layer

**SYNTAX:**

**keras.layers.Dense(units,activation=None,use\_bias=True, kernel\_initializer=’glorot\_uniform’,bias\_initializer=’zeros’, kernel\_regularizer=None, bias\_regularizer=None, activity\_regularizer=None, kernel\_constraint=None, bias\_constraint=None)**

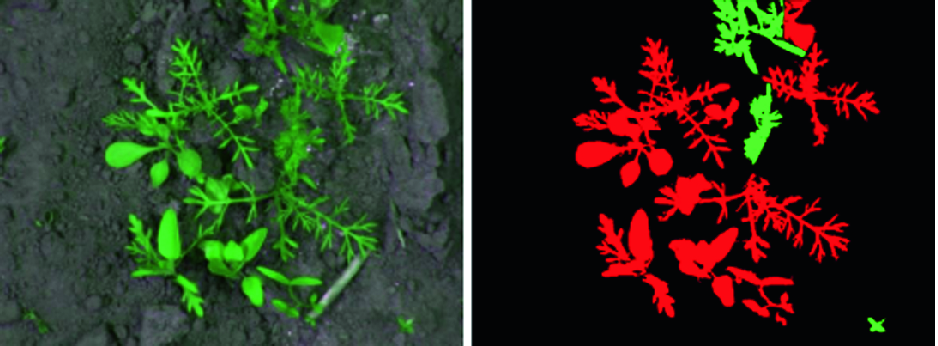
**4.4 Dataset:**

The dataset consists of 1046 images along with annotations available online. These images were acquired by two German researchers Sebastian Haug and Jӧrn Ostermann with the help of autonomous field robot Bonirob in cucumber crop field when the plants were still at their early growth stage. Also, at the time when the images were captured for the data collection, both the weeds and crops were of approximately same size.



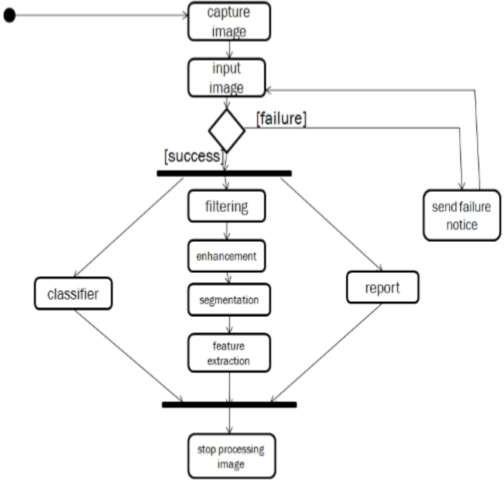
*Fig 4.4(a): Sample image data set*

The dataset contains various features extracted from the given image dataset to calculate whether an image include any indication of weed or not. The features that were extracted represent the individual anatomical part. The method that were utilized for image analysis and construction of 64 color features and 45 texture attributes are based on RBG level co-occurrence of matrix.



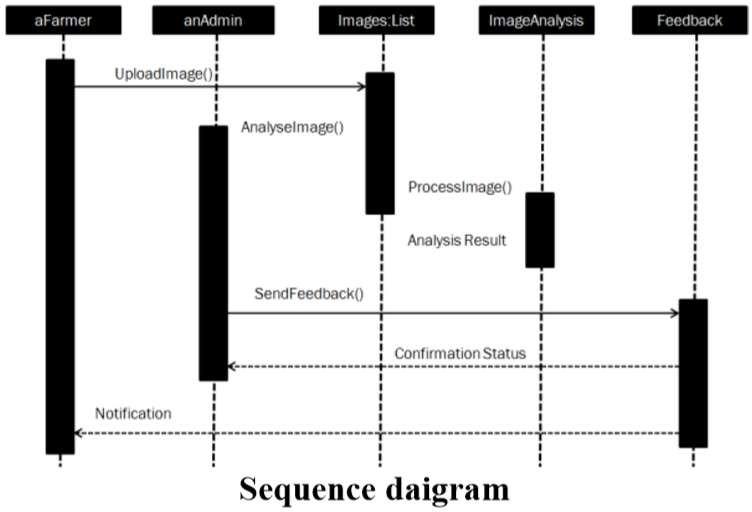
*Fig 4.4(b): RBG image of the image dataset*

**4.5 FLOW DIAGRAM**

The flow of the operations to be performed in a sequential manner are:

*Fig 4.5(a): Flow diagram*

Putting a retrospective in a chronological sequence, we get the sequence diagram as

****

*Fig 4.5(b): Sequence Diagram*

**CHAPTER 5**

**WEB FRAMEWORK AND APPLICATIONS**

**5.1 WEB FRAMEWORK**



**Web framework resources**

* When you are learning how to use one or more web frameworks it's helpful to have an idea of what the code under the covers is doing.
* Frameworks is a really well done short video that explains how to choose between web frameworks. The author has some particular opinions about what should be in a framework. For the most part I agree although I've found sessions and database ORMs to be a helpful part of a framework when done well.
* what is a web framework? is an in-depth explanation of what web frameworks are and their relation to web servers.
* Django vs Flash vs Pyramid: Choosing a Python web framework contains background information and code comparisons for similar web applications built in these three big Python frameworks.
* This fascinating blog post takes a look at the code complexity of several Python web frameworks by providing visualizations based on their code bases.
* Python’s web frameworks benchmarks is a test of the responsiveness of a framework with encoding an object to JSON and returning it as a response as well as retrieving data from the database and rendering it in a template. There were no conclusive results but the output is fun to read about nonetheless.
* What web frameworks do you use and why are they awesome? is a language agnostic Reddit discussion on web frameworks. It's interesting to see what programmers in other languages like and dislike about their suite of web frameworks compared to the main Python frameworks.
* This user-voted question & answer site asked "What are the best general purpose Python web frameworks usable in production?". The votes aren't as important as the list of the many frameworks that are available to Python developers.

**5.2 APPLICATIONS**

A web framework is an architecture containing tools, libraries, and functionalities suitable to build and maintain massive web projects using a fast and efficient approach. They are designed to streamline programs and promote code reuse. To create the server-side of the web application, you need to use a server-side language. Python is home to numerous such frameworks, famous among which are Django and FlasK

**CHAPTER 6**

**RESULTS AND DISCUSSIONS**

**6.1 Accuracy Tables:**

Initially we have worked using two optimizers RMSProp and Adam we have trained the data using both the optimizers, based on the performance levels we have chosen Adam fits best for the validation process as it is giving higher accuracy and very small loss.

The tables shown below are the accuracy and loss obtained during training the model using both the optimizers (RMSProp and Adam) individually.

Table

Description automatically generated

*Fig 6.1(a): Accuracy and Loss using Adam Optimizer*

Table

Description automatically generated

*Fig6.1(b). Accuracy and Loss Using RMSProp Optimizer*

Based on above shown results the high accuracy is obtained for adam optimizer thus we have chosen adam optimizer for prediction the data test for classification between crops and weeds.

**6.2 Accuracy and Loss Graphs**

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Fig 6.2(a) Accuracy Graph using Adam Fig 6.2(b): Loss Graph using Adam

**6.3 Outputs:**

This is the web page will be displayed when you run the prediction code on anaconda prompt by activating tensor flow and run it.

Graphical user interface, text, application

Description automatically generated

*Figure 6.3(a): Output page*

Webpage for predicting wether it is a crop or weed created using Flask web framework with HTML and CSS.

**Prediction:**

**Input:** Crop image taken from the validation set to predict the maodel

A picture containing outdoor, ground, rock, leaf

Description automatically generated

*Fig 6.3(b): agri\_0\_057*

**Output:**

Graphical user interface

Description automatically generated with medium confidence

*Fig 6.3(c): Crop Result*

Thus the trained model predicted the image as “It is a Crop.”

**Input:** Weed image taken from validation data set for prediction

A picture containing outdoor, tree, plant, forest

Description automatically generated

*Fig 6.3(d): agri\_0\_084*

**Output:**

Graphical user interface, application

Description automatically generated

*Fig 6.4(e): Weed Result*

**CHAPTER 7**

**CONCLUSION AND FUTURE OF WORK**

**7.1 CONCLUSION**

Feature extraction concept utilized for reducing the amount of computing and storage resources required to describe and process a large set of data. In the proposed work, CNN classification model is used to extract the important features and the extracted significant features are used for effective training of the model. The proposed model gives an accuracy of 92%. In the near future, steps may be taken to improve the accuracy and to evaluate the model with other parameters. The research methodology proposed in this article would be extended further to other crops and weeds with high dense field contains different categories of crop.

India is a cultured country and about 70% of the residents depend on agriculture. Farmers have large range of variety for select various suitable crops and finding the weeds of a plant. It can be improved by the aid of technological support. By using this Identification tool, lots of money and time can be saved.This identification technique reduces the amount of computation when compared to others. Adaptation to such tools will help in the revolutionization of the farming industry.

**7.2 FUTURE WORKS**

In integrated weed management, farmers employ a diversity of weed-killing techniques, including tillage, cultural practices, and methods for depleting the weed seed bank, rather than depending solely on the spraying of Roundup or another single tool.

Fortunately, weed scientists and other biologists understand variability very well, and Young is now trying to encourage more collaboration between them and engineers. Engineers necessarily focus on machinery—how it’s built, how fast it goes, how well it detects a target. Meanwhile, weed scientists understand plants and their responses to different weed- killing treatments, or the “application side of things,” Young says. “The engineers are bringing biology to their work, but in my opinion they could get there faster if they worked with weed scientists.”

**REFERENCES**

[1] Ahmed, F., Al-Mamun, H.A., Bari, A.S.M.H., Hossain, E., Kwan, P., 2012. Classification of crops and weeds from digital images: A support vector machine approach. Crop Protect. 40, 98–104

[2] Hung, C., Xu, Z., Sukkarieh, S., 2014. Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a uav. Remote Sens. 6 (12), 12037–12054

[3] Siddiqi, M.H., Lee, S., Kwan, A.M., 2014. Weed image classification using wavelet transform, stepwise linear discriminant analysis, and support vector machines for an automatic spray control system. J. Inform. Sci. Eng. 30

[4] Saha, D., Hanson, A., Shin, S.Y., 2016. Development of enhanced weed detection system with adaptive thresholding and support vector machine. Proceedings of the International Conference on Research in Adaptive and Convergent Systems, pp. 85–88

[5] Ishak, A.J., Hussain, A., Mustafa, M.M., 2009. Weed image classification using gabor wavelet and gradient field distribution. Comput. Electron. Agric. 66 (1), 53–6

**APPENDIX**

**Training Process:**

**#Importing Libraries**

import tensorflow

from tensorflow import keras

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Convolution2D

from tensorflow.keras.layers import Flatten

from tensorflow.keras.layers import DensE

from tensorflow.keras.layers import MaxPooling2D

**#Working on CNN layers**

classifier=Sequential()

classifier.add(Convolution2D(32,(3,3),input\_shape=(224,224,3), activation='relu'))

classifier.add(MaxPooling2D(pool\_size=(2,2)))

classifier.add(Convolution2D(32,(3,3), activation='relu'))

classifier.add(MaxPooling2D(pool\_size=(2,2)))

classifier.add(Convolution2D(64,(3,3), activation='relu'))

classifier.add(MaxPooling2D(pool\_size=(2,2)))

**# Minimizing the matrix size**

classifier.add(Flatten())

classifier.add(Dense(units=32,activation='relu')

classifier.add(Dense(units=64,activation='relu'))

classifier.add(Dense(units=128,activation='relu'))

classifier.add(Dense(units=256,activation='relu'))

classifier.add(Dense(units=2,activation='sigmoid'))

classifier.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

**# Loading Image Dataset and Working**

from tensorflow.keras.preprocessing.image import ImageDataGenerator

Train\_datagen=ImageDataGenerator(rescale=1./255,

shear\_range=0.15,

zoom\_range=0.2,

horizontal\_flip=True)

test\_datagen=ImageDataGenerator(rescale=1./255)

#In[6]

train\_path\_dir="agriculture/Train"

test\_path\_dir="agriculture/Test"

training\_set=train\_datagen.flow\_from\_directory(train\_path\_dir,

target\_size=(224,224),

batch\_size=20,

class\_mode='categorical')

test\_set=test\_datagen.flow\_from\_directory(test\_path\_dir,

target\_size=(224,224),

batch\_size=10,

class\_mode='categorical')

**# Model Training**

x=classifier.fit\_generator(training\_set,

steps\_per\_epoch=1000//20,

epochs=20,

validation\_data=test\_set,

validation\_steps=60)

**#Saving the trained model**

classifier.save('try\_one.h5')

**#Ploting the performance**

acc = x.history['acc']

val\_acc =x.history['val\_acc']

loss = x.history['loss']

val\_loss = x.history['val\_loss']

epochs = range(len(acc))import matplotlib.pyplot as plt

plt.plot(epochs,acc)

plt.plot(epochs,val\_acc)

plt.title("Training and validation Accuracy")

plt.figure()

plt.plot(epochs,loss)

plt.plot(epochs,val\_loss)

plt.title("Training and validation Loss")

plt.figure()

**Preparing Web Application:**

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing.image import load\_img, img\_to\_array

import numpy as np

from flask import Flask, request, render\_template

from werkzeug.utils import secure\_filename

import os, sys, glob, re

app = Flask(\_\_name\_\_)

model\_path = "try\_one.h5"

classes = {0:"Crops:-{ About Crops }",1:"weeds:-{ about weeds} "}

def model\_predict(image\_path):

print("Predicted")

image = load\_img(image\_path,target\_size=(224,224))

image = img\_to\_array(image)

image = image/255

image = np.expand\_dims(image,axis=0)

model = load\_model(model\_path)

result = np.argmax(model.predict(image))

prediction = classes[result]

if result == 0:

print("crop.html")

return "Crops","crops.html"

elif result == 1:

print("weeds.html")

return "weeds", "weeds.html"

**Predicting the Output:**

from tensorflow.keras.preprocessing import image

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

import numpy as np

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

import numpy as np

# dimensions of our images

img\_width, img\_height = 224,224

# load the model we saved

model = load\_model('try\_one.h5')

# predicting images

img = image.load\_img('data/agriculture/Train/weeds/agri\_0\_808.jpeg', target\_size=(img\_width, img\_height))

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis=0)

classes = model.predict(x)

print (classes)

# predicting images

#image = image.convert("RGB")

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis=0)

result = np.argmax(model.predict(x))

print(result)