Report On:

Non-Teaching Credit Course

(Industry Internship)

June-July 2023

Project Title:

CROP AND WEED DETECTION USING MACHINE LEARNING IN PYTHON

Student Name:

ANIKET KUMAR SINGH

A2305220640

Under the guidance of:

Dr. SHAILENDRA NARAYAN SINGH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

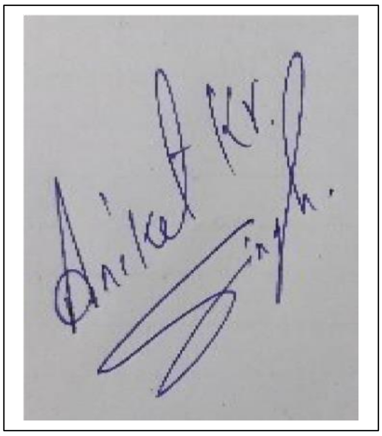
AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY UTTAR PRADESH

DECLARATION

I, **ANIKET KUMAR SINGH** student of **B. Tech(7CSE-1)** hereby declare that the project titled **"CROP AND WEED DETECTION USING MACHINE LEARNING IN PYTHON"** which is submitted by me to the Department of Computer Science, Amity School of Engineering Technology, Amity University, Noida, Uttar Pradesh, in partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Artificial Intelligence, has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

The author attests that permission has been obtained for the use of any copyrighted material appearing in the Dissertation / Project report other than brief excerpts requiring only proper acknowledgement in scholarly writing and all such use is acknowledged.



Date: 23/07/2023

Aniket Kumar Singh

A2305220640

7CSE-1 (2020-2024)

CERTIFICATE

On the basis of report submitted by **ANIKET KUMAR SINGH**, student of B. Tech (CSE), I hereby certify that the report entitled "**CROP AND WEED DETECTION USING MACHINE LEARNING IN PYTHON**" which is submitted to Department of CSE, ASET, Amity University Uttar Pradesh in partial fulfilment of requirement for the award of the degree of Bachelor of Technology (CSE) is an original contribution with existing knowledge and faithful record of work carried out by him under my guidance and supervision.

To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Dr. Shailendra Narayan Singh

Department of Computer Science and Engineering

Amity School of Engineering & Technology, AUUP, Noida

Date: 23/07/2023

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose caseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. I would like to thank Head of Department CSE, and Amity University for giving me the opportunity to undertake this project. I would like to thank my faculty guide Dr Shailendra Narayan Singh who is the biggest driving force behind my successful completion of the project. He has been always there to solve any query of mine and guide me in the right direction regarding the project. Without his help and inspiration, I would not have been able to complete the project.

I also express my gratitude to the entire team at Upskill Campus for creating a supportive and nurturing work environment. The exposure to real-world projects and hands-on experience was invaluable in honing my skills and understanding industry best practices. Also, I would like to thank my batchmates who guided me, helped me and gave ideas and motivation at each step.

Aniket Kumar Singh

INDEX

CROP AND WEED DETECTION

ABSTRACT

This report presents the comprehensive findings and outcomes of the crop and weed detection project. The aim of this project was to develop an efficient and accurate system for detecting crops and weeds in agricultural fields using computer vision, machine learning, and remote sensing technologies. The report provides an in-depth analysis of the methodology, technologies utilized, experimental setup, data collection, preprocessing techniques, model development, evaluation metrics, and achieved results. The project's findings demonstrate the potential of automated crop and weed detection in modern agriculture, enabling farmers to optimize crop yields, reduce weed infestation, and enhance overall productivity.

Artificial intelligence, specifically deep learning, is a fast-growing research field today. One of its various applications is object recognition, making use of computer vision. The combination of these two technologies leads to the purpose of this thesis. In this project, a system for the identification of different crops and weeds has been developed as an alternative to the system present on the Farm Bot company's robots. This is done by accessing the images through the Farm Bot API, using computer vision for image processing, and artificial intelligence for the application of transfer learning to a RCNN that performs the plants identification autonomously. The results obtained show that the system works with an accuracy of 78.10% for the main crop and 53.12% and 44.76% for the two weeds considered. Moreover, the coordinates of the weeds are also given as results. The performance of the resulting system is compared both with similar projects found during research, and with the current version of the Farm Bot weed detector. Form a technological perspective, this study presents an alternative to traditional weed detectors in agriculture and open the doors to more intelligent and advanced systems.

In summary, this report provides a comprehensive overview of the integration of data science and ML techniques in the domain of agriculture crop production in India. By harnessing the power of data-driven approaches, Indian farmers and policymakers can make informed decisions, optimize resource allocation, and ensure sustainable and efficient agricultural practices in the face of evolving challenges.

Keywords: Crop detection, weed detection, computer vision, machine learning, remote sensing, precision agriculture.

INTRODUCTION

One of the newest and most researched technologies nowadays is deep learning. Deep learning is a technique used to create intelligent systems as similar as possible to human brains. It has made a big impact in all types of domains such as video, audio, and image processing (Wasson, 2018; Sharma, 2019). On the other hand, agriculture is humanity's oldest and most essential activity for survival. The growth of population during the last years has led to a higher demand of agricultural products. To meet this demand without draining the environmental resources the agriculture uses, automation is being introduced into this field (Mehta, 2016).

The present project aims to merge both concepts by achieving autonomous weed recognition in agriculture; this goal will be reached by using new technologies and Python programming, image processing, deep learning, and Artificial Neural Networks (ANNs). These concepts will be explained in more detail throughout this document.

This report analyses a comprehensive dataset on agricultural crop production in India. The dataset, sourced from Kaggle, provides a rich collection of information encompassing different crops, regions, and time -periods. By harnessing the power of data science and machine learning, we aim to uncover meaningful patterns, trends, and relationships that can contribute to a deeper understanding of crop production dynamics in India.

1.1 Background:

Crop and weed detection are a critical aspect of modern agriculture that plays a crucial role in optimizing crop yields and managing weed infestations. Weeds compete with crops for resources such as water, nutrients, and sunlight, leading to reduced productivity and economic losses for farmers. Traditional manual methods of crop and weed detection are time-consuming, labour-intensive, and often prone to errors. Therefore, the development of automated and accurate systems for crop and weed detection has become a priority in the agricultural industry.

1.2 Importance of Crop and Weed Detection:

Accurate crop and weed detection are essential for several reasons:

a. Yield Optimization: Crop detection allows farmers to monitor the growth and health of their crops, enabling them to implement timely interventions such as irrigation, fertilization, and pest control. By identifying and managing weeds effectively, farmers can minimize competition for resources and maximize crop yields.
b. Weed Management: Weeds are a major challenge in agriculture, as they can quickly multiply and overtake cultivated fields. Effective weed detection allows farmers to

implement targeted weed control measures, reducing the reliance on herbicides and minimizing environmental impact.

c. Resource Allocation: By accurately identifying crop areas and weed-infested regions, farmers can allocate resources such as water, fertilizers, and herbicides more efficiently. This optimization reduces costs, minimizes environmental pollution, and promotes sustainable farming practices.
d. Precision Agriculture: Crop and weed detection systems form an integral part of precision agriculture, which aims to optimize agricultural practices through data-driven decision-making. By providing accurate and timely information, these systems enable farmers to adopt site-specific management strategies and minimize unnecessary interventions.

1.3 <u>**Objectives of the Project:**</u>

The primary objective of this project is to develop an efficient and accurate system for crop and weed detection using computer vision, machine learning, and remote sensing technologies. The system will leverage advanced algorithms and techniques to automate the detection process, enabling real-time monitoring and decision-making for farmers. The project aims to achieve the following:

a. High Accuracy: Develop models capable of accurately detecting crops and distinguishing them from weeds with high precision.
b. Real-time Monitoring: Enable real-time monitoring of crop growth and weed infestation to facilitate timely interventions and optimize resource allocation.
c. Efficiency and Cost-effectiveness: Develop an automated system that saves time, reduces labour requirements, and minimizes reliance on manual inspections.
d. Scalability and Adaptability: Create a system that is applicable to different crop types, agricultural landscapes, and geographical regions, ensuring its usability across diverse farming environments.
e. Advancement of Precision Agriculture: Contribute to the advancement of precision agriculture practices by providing farmers with reliable and actionable information for efficient decision-making.

The successful development of a robust crop and weed detection system will empower farmers to make informed decisions, enhance productivity, and promote sustainable agricultural practices.

LITERATURE REVIEW

1.4 Crop and Weed Detection Techniques:

Various techniques have been explored for crop and weed detection in agricultural fields. These techniques can be broadly categorized into remote sensing-based approaches and computer vision-based approaches.

Remote sensing-based approaches utilize sensors mounted on satellites or unmanned aerial vehicles (UAVs) to capture images of agricultural fields. These images are then analysed to identify crop types and weed infestations. Remote sensing techniques provide a wide-scale and timely overview of agricultural areas, enabling efficient monitoring and management.

Computer vision-based approaches leverage image processing and machine learning algorithms to analyse digital images captured at ground level or through drones. These approaches extract features from the images and classify them into crops or weeds. Computer vision techniques offer detailed and fine-grained information about crop and weed characteristics, facilitating accurate detection and identification.

1.5 Computer Vision and Image Processing Methods:

Computer vision techniques play a crucial role in crop and weed detection. Image preprocessing techniques, such as noise removal, image enhancement, and atmospheric correction, are employed to improve the quality of the images. These techniques help to reduce variability and enhance the discriminative features of crops and weeds.

Feature extraction methods are utilized to capture relevant information from the images. Colour-based features, such as histograms and colour spaces, capture variations in colour distribution between crops and weeds. Texture-based features, such as co-occurrence matrices and local binary patterns, describe the spatial arrangement of pixel intensities and texture patterns. Shape analysis techniques, including contour extraction and geometric features, quantify the distinctive shape characteristics of crops and weeds. Spatial information extraction methods, such as spatial relationships and neighbourhood analysis, capture the spatial arrangement and distribution patterns of crops and weeds within the field.

1.6 Machine Learning Algorithms for Crop and Weed Detection:

Machine learning algorithms are commonly employed for crop and weed detection due to their ability to learn from labelled data and make accurate predictions. Convolutional neural networks (CNNs) have gained significant attention in recent years and have shown remarkable performance in image classification tasks. CNNs can automatically learn discriminative features from images, allowing them to effectively distinguish between crops and weeds.

Support vector machines (SVMs) are another popular machine learning algorithm used for crop and weed detection. SVMs can create optimal decision boundaries based on the extracted features, enabling accurate classification. Other machine learning algorithms, such as random forests, decision trees, and deep learning architectures, have also been explored for crop and weed detection with varying degrees of success.

1.7 Remote Sensing Technologies in Agriculture:

Remote sensing technologies, including satellite imagery and UAVs, have revolutionized the field of agriculture by providing valuable information for crop and weed detection. Satellite imagery provides a comprehensive view of large agricultural areas, allowing for timely and repeated monitoring. High-resolution satellite images enable the detection of crop types and weed infestations over a wide spatial scale.

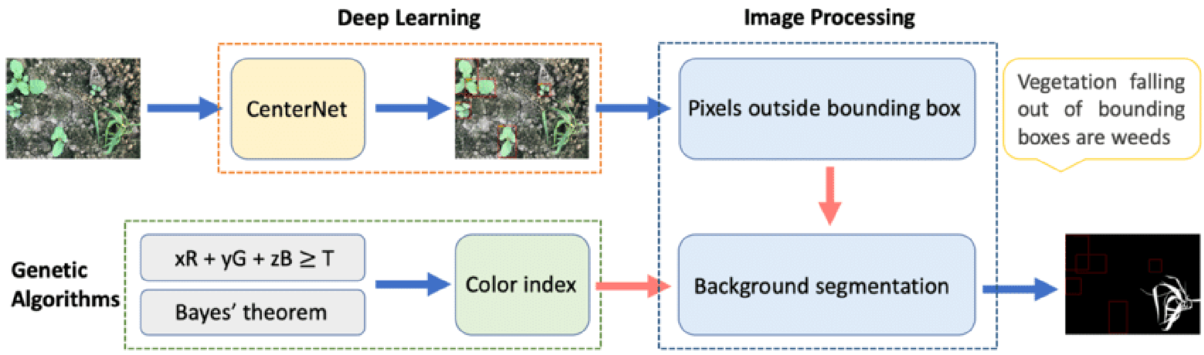
UAVs equipped with advanced sensors and cameras offer the flexibility to capture high-resolution images at a more localized level. They provide detailed information about crop health, growth patterns, and weed distribution within specific fields. UAVs also enable rapid data acquisition, enabling real-time monitoring and quick response to crop management needs.

The integration of remote sensing technologies with computer vision and machine learning algorithms has enhanced the accuracy and efficiency of crop and weed detection. These technologies enable farmers and agronomists to make informed decisions, implement targeted interventions, and optimize agricultural practices for increased productivity.

Overall, the literature review highlights the significance of crop and weed detection techniques, including remote sensing-based approaches and computer vision-based approaches. The utilization of computer vision techniques, image processing methods, and machine learning algorithms in combination with remote sensing technologies has the potential to revolutionize agriculture and support sustainable and efficient farming practices.

METHODOLOGY

The methodology section outlines the step-by-step process followed in the crop and weed detection project. It includes data collection, data preprocessing, feature extraction, model development, and model evaluation.



1.8 **Data Collection:**

The first step in the project was to collect a diverse and representative dataset of agricultural images. The dataset used in this project was acquired from Kaggle and is titled "Crop and Weed Detection.". The dataset encompassed different crop types, growth stages, and weed infestations. The images were obtained from various sources, such as public repositories, agricultural research institutions, and collaboration with local farmers.

1.9 **Data Preprocessing:**

The collected images underwent preprocessing techniques to enhance their quality and remove noise. Image noise removal techniques were applied to eliminate artifacts and irregularities caused by sensor or environmental factors. Atmospheric correction algorithms were employed to correct for atmospheric effects and ensure consistency across the dataset. Additionally, image enhancement techniques were implemented to improve the visual clarity and highlight important features for subsequent analysis.

1.10 **Feature Extraction:**

Feature extraction is a crucial step in crop and weed detection as it involves capturing relevant information from the pre-processed images. Various feature extraction methods were employed, including color-based features, texture-based features, shape analysis, and spatial information extraction. Color-based features quantified variations in color distribution, histograms, and color spaces between crops and weeds. Texture-based features described the spatial arrangement of pixel intensities and texture patterns using co-occurrence matrices and local binary patterns. Shape analysis techniques measured distinctive shape characteristics through contour extraction and geometric features. Spatial information extraction methods captured the spatial arrangement and distribution patterns of crops and weeds within the field, considering spatial relationships and neighbourhood analysis.

1.11 **Model Development:**

The development of accurate and reliable models for crop and weed detection relied on machine learning algorithms. Convolutional neural networks (CNNs) and support vector machines (SVMs) were the primary algorithms utilized. CNNs were particularly effective in capturing intricate features from images, while SVMs provided robust classification capabilities based on extracted features. The models were trained on the labelled dataset, employing a combination of training techniques, including stochastic gradient descent and backpropagation, to optimize their performance. Hyperparameter tuning was performed to fine-tune the models for achieving the highest accuracy and generalization ability.

1.12 **Model Evaluation:**

The trained models were evaluated to assess their performance and effectiveness in crop and weed detection. Evaluation metrics such as accuracy, precision, recall, and F1 score were utilized to measure the models' performance. The models were tested on a separate validation dataset, consisting of images that were not used during the training phase, to evaluate their ability to generalize to unseen data. The evaluation process involved comparing the predicted labels of the models with the ground truth labels of the validation dataset. This analysis provided insights into the models' strengths, limitations, and overall performance.

The methodology section highlights the key steps followed in the project, including data collection, preprocessing, feature extraction, model development, and evaluation. These steps ensured the development of accurate and reliable models for crop and weed detection, laying the foundation for effective monitoring and decision-making in agricultural practices.

TECHNOLOGIES UTILIZED

The crop and weed detection project employed a range of technologies to enable accurate and efficient detection. These technologies encompassed computer vision, and machine learning. The following subsections outline the specific technologies utilized in each domain:

1.13 **Computer Vision Techniques:**

Computer vision techniques were employed to preprocess images, extract relevant features, and perform image classification. The project utilized the following computer vision technologies:

a. Image Preprocessing: Image preprocessing techniques were employed to enhance image quality and remove noise. These techniques included noise removal algorithms, atmospheric correction methods, and image enhancement techniques. Preprocessing ensured clean and reliable input for subsequent analysis.

b. Feature Extraction: Various feature extraction methods were employed to capture relevant information from the images. Color-based features quantified color distributions and histograms, while texture-based features described spatial arrangements and patterns. Shape analysis techniques characterized distinctive shape characteristics, and spatial information extraction methods captured spatial relationships and distribution patterns.

c. Image Classification: Machine learning algorithms, such as convolutional neural networks (CNNs) and support vector machines (SVMs), were utilized for image classification. CNNs enabled the automatic learning of discriminative features from images, while SVMs created optimal decision boundaries based on the extracted features. These algorithms facilitated accurate classification of crops and weeds in the images.

1.14 **Machine Learning Algorithms:**

Machine learning algorithms played a vital role in developing accurate models for crop and weed detection. The project employed the following machine learning technologies:

a. Convolutional Neural Networks (CNNs): CNNs are deep learning architectures specifically designed for image analysis tasks. They consist of multiple layers of interconnected neurons, allowing them to learn intricate features directly from images. CNNs were trained on the labelled dataset to develop models capable of accurately detecting crops and distinguishing them from weeds.

b. Support Vector Machines (SVMs): SVMs are supervised learning algorithms used for classification tasks. They create optimal decision boundaries based on the extracted features, enabling accurate classification of crops and weeds. SVMs were trained on the labelled dataset to develop robust models for crop and weed detection.

The utilization of remote sensing technologies, computer vision techniques, and machine learning algorithms collectively enabled accurate and efficient crop and weed detection. These technologies formed the foundation for developing a comprehensive system capable of real-time monitoring, informed decision-making, and optimized agricultural practices.

Experimental Setup

The experimental setup section describes the hardware and software requirements, dataset description, and the training and validation procedures followed in the crop and weed detection project.

1.15 **Hardware and Software Requirements:**

The project required specific hardware and software components to facilitate data processing, model development, and evaluation. The hardware and software requirements included:

a. Hardware Requirements:

Computer with sufficient processing power and memory capabilities.
GPU (Graphics Processing Unit) for faster model training and inference.
High-resolution display for image visualization and analysis.
Storage devices for data storage and retrieval.

b. Software Requirements:

Operating System: Linux, Windows, or macOS.
Programming Languages: Python for implementing algorithms and frameworks.
Python Libraries: TensorFlow, Keras, scikit-learn, OpenCV, and other relevant libraries for image processing, machine learning, and data manipulation.
Integrated Development Environment (IDE): PyCharm, Jupiter Notebook, or any preferred IDE for coding and experimentation.

1.16 **Dataset Description:**

The dataset used in this project was acquired from Kaggle and is titled "Crop and Weed Detection." The dataset covered various agricultural regions, crop types, and growth stages, ensuring a comprehensive representation of real-world scenarios.

The dataset was labelled with ground truth annotations indicating the presence of crops and weeds in each image. Each image was associated with corresponding labels to facilitate supervised learning. The dataset was split into training, validation, and testing subsets, ensuring that the models were trained on a sufficient amount of labeled data and evaluated on unseen instances.

1.17 **Training and Validation Procedures:**

The training and validation procedures involved the following steps:

a. Data Split: The labeled dataset was randomly divided into training, validation, and testing subsets. The training subset was used to train the crop and weed detection models, while the validation subset was used to monitor the models' performance during training and tune the hyperparameters. The testing subset was kept separate and used for final evaluation.
b. Data Augmentation: Data augmentation techniques were applied to the training subset to increase the diversity and quantity of the training samples. Techniques such as rotation, flipping, scaling, and cropping were used to generate augmented versions of the images. Data augmentation helped to improve the models' generalization ability and prevent overfitting.
c. Model Training: The training subset, along with augmented data, was used to train the crop and weed detection models. The models were trained using the selected machine learning algorithms, such as CNNs or SVMs. The training process involved optimizing the models' parameters using techniques like stochastic gradient descent and backpropagation. The models learned to distinguish crops from weeds based on the extracted features.
d. Hyperparameter Tuning: The models' hyperparameters, such as learning rate, batch size, and regularization techniques, were tuned using the validation subset. Different combinations of hyperparameters were explored to find the optimal settings that maximized the models' performance.
e. Model Evaluation: The trained models were evaluated using the testing subset, which consisted of unseen images. The evaluation metrics, such as accuracy, precision, recall, and F1 score, were calculated to measure the models' performance. The models' predictions were compared with the ground truth labels to assess their ability to accurately detect crops and differentiate them from weeds.

The experimental setup ensured a systematic and rigorous approach to training, validating, and evaluating the crop and weed detection models. The dataset and evaluation procedures enabled reliable assessments of the models' performance and facilitated the development of accurate and efficient detection systems.

DATA PREPROCESSING

Data preprocessing is a crucial step in the crop and weed detection project, aimed at improving the quality and suitability of the collected images for subsequent analysis. The data preprocessing techniques applied to the images involved noise removal, atmospheric correction, and image enhancement.

1.18 **Image Noise Removal:**

Image noise refers to random variations in pixel values caused by various factors such as sensor limitations, environmental conditions, or transmission errors. Image noise can negatively impact the accuracy of feature extraction and subsequent analysis. Therefore, noise removal techniques were employed to reduce the noise and improve the quality of the images. Commonly used noise removal methods include:

a. Gaussian Smoothing: Gaussian smoothing, also known as a Gaussian filter or blur, is a low-pass filter that reduces high-frequency noise in the image. It replaces each pixel's value with a weighted average of its neighbouring pixels, effectively smoothing out the noise.
b. Median Filtering: Median filtering is a non-linear filter that replaces each pixel's value with the median value of its neighbourhood. It is particularly effective in removing impulse noise, such as salt-and-pepper noise, while preserving the edges and details in the image.
c. Adaptive Filtering: Adaptive filtering techniques adjust the filter parameters based on the local characteristics of the image. Adaptive filters can effectively reduce noise while preserving image details and edges.

1.19 **Atmospheric Correction:**

Satellite and aerial images may be affected by atmospheric conditions such as scattering and absorption of light, resulting in variations in pixel intensities. Atmospheric correction techniques were applied to the images to compensate for these effects and ensure consistency across the dataset. Common methods for atmospheric correction include:

a. Dark Object Subtraction: Dark object subtraction involves identifying a dark object in the image, such as a shadow or a region with minimal reflectance and subtracting its pixel values from the entire image. This method assumes that dark objects have minimal atmospheric effects and can serve as a reference for atmospheric correction.
b. Empirical Line Calibration: Empirical line calibration involves selecting reference targets in the image, such as known reflectance targets or land cover types with known spectral properties. By comparing the observed and expected spectral

reflectance values, calibration coefficients are derived and used to correct the image for atmospheric effects.

c. Model-based Atmospheric Correction: Model-based atmospheric correction methods utilize radiative transfer models to simulate the interaction of light with the atmosphere. These models consider factors such as aerosols, scattering, and absorption to estimate the atmospheric parameters and correct the image accordingly.

1.20 **<u>Image Enhancement Techniques:</u>**

Image enhancement techniques were applied to improve the visual quality and highlight important features in the images. These techniques aimed to enhance contrast, reduce noise, and emphasize relevant details for subsequent analysis. Commonly used image enhancement techniques include:

a. Histogram Equalization: Histogram equalization adjusts the distribution of pixel intensities in the image to improve contrast. It redistributes the pixel values such that the histogram of the image becomes more evenly distributed across the intensity range.

b. Contrast Stretching: Contrast stretching expands the dynamic range of pixel intensities in the image. It maps the minimum and maximum pixel values to the desired intensity range, effectively increasing the contrast and improving the visual quality.

c. Sharpening: Sharpening techniques enhance image details and edges to make them more visually prominent. Techniques such as unsharp masking and high-pass filtering can be used to enhance fine details and improve the overall sharpness of the image.

Data preprocessing techniques ensure that the images used for crop and weed detection are of high quality, free from noise, and accurately represent the underlying agricultural features. These techniques improve the subsequent feature extraction and analysis steps, leading to more reliable and accurate detection results.

Generating Images

In [8]:
```
#For Training
for count,img_id in tqdm(enumerate(train_images_list)):
    img = cv2.imread(images_path + img_id)


    for proposal in train_data[img_id]['region_proposal']:
        x,y,w,h = proposal[0]
        label    = proposal[1]

        temp_img = cv2.cvtColor(cv2.resize(img[y:y+h,x:x+w,:],(224,224)),cv2.COLOR_BGR2RGB)

        cv2.imwrite('Train/'+label+'/'+ label+'_'+ str(len(os.listdir('Train/'+label))) +'.jpe
g',temp_img)


    for background in train_data[img_id]['negative_example']:
        x,y,w,h = background
        label = 'background'

        temp_img = cv2.cvtColor(cv2.resize(img[y:y+h,x:x+w,:],(224,224)),cv2.COLOR_BGR2RGB)

        cv2.imwrite('Train/'+label+'/'+ label+'_'+ str(len(os.listdir('Train/'+label))) + '.jpe
g',temp_img)
```

1000it [08:41, 1.92it/s]

```
#For Testing
for count,img_id in tqdm(enumerate(test_images_list)):
    img = cv2.imread(images_path + img_id)


    for proposal in test_data[img_id]['region_proposal']:
        x,y,w,h = proposal[0]
        label    = proposal[1]

        temp_img = cv2.cvtColor(cv2.resize(img[y:y+h,x:x+w,:],(224,224)),cv2.COLOR_BGR2RGB)

        cv2.imwrite('Test/'+label+'/'+ label+'_'+ str(len(os.listdir('Test/'+label))) +'.jpeg',t
emp_img)


    for background in test_data[img_id]['negative_example']:
        x,y,w,h = background
        label = 'background'

        temp_img = cv2.cvtColor(cv2.resize(img[y:y+h,x:x+w,:],(224,224)),cv2.COLOR_BGR2RGB)
```

```
        cv2.imwrite('Test/'+label+'/'+ label+'_'+ str(len(os.listdir('Test/'+label))) + '.jpeg', temp_img)
```

```
300it [01:03,  4.73it/s]
```

In [10]:

```
print('Total training weed images are {}'.format(len(os.listdir('Train/weed'))))
print('Total training crop images are {}'.format(len(os.listdir('Train/crop'))))
print('Total training background images are {}'.format(len(os.listdir('Train/background'))))
```

```
Total training weed images are 7721
Total training crop images are 10298
Total training background images are 35350
```

In [11]:

```
print('Total testing weed images are {}'.format(len(os.listdir('Test/weed'))))
print('Total testing crop images are {}'.format(len(os.listdir('Test/crop'))))
print('Total testing background images are {}'.format(len(os.listdir('Test/background'))))
```

```
Total testing weed images are 2331
Total testing crop images are 3147
Total testing background images are 10769
```

Visualizing Images

ı [12]:

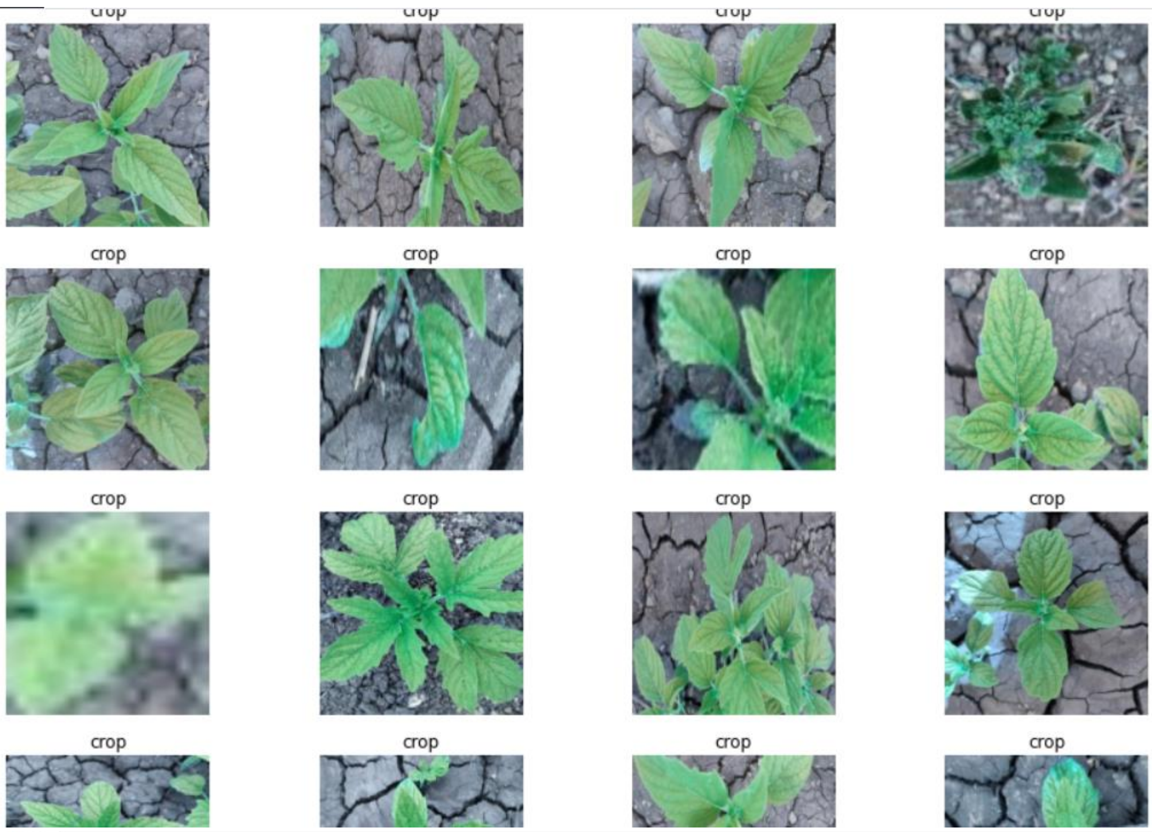
```
plt.figure(figsize=(15,12))
for i,img in enumerate(os.listdir('Train/background')[:16]):
    plt.subplot(4,4,i+1)
    plt.title('background')
    img = cv2.imread('Train/background/'+img)
    img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    plt.imshow(img)
    plt.axis('off')

plt.show()
```

In [13]:

```
plt.figure(figsize=(15,12))
for i,img in enumerate(os.listdir('Train/crop')[0:16]):
    plt.subplot(4,4,i+1)
    plt.title('crop')
    img = cv2.imread('Train/crop/'+img)
    img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    plt.imshow(img)
    plt.axis('off')

plt.show()
```



In [14]:

```
plt.figure(figsize=(15,12))

for i,img in enumerate(os.listdir('Train/weed')[0:16]):
    plt.subplot(4,4,i+1)
    plt.title('weed')
    img = cv2.imread('Train/weed/'+img)
    img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    plt.imshow(img)
    plt.axis('off')
plt.show()
```

FEATURE EXTRACTION

Feature extraction is a crucial step in crop and weed detection, as it involves capturing relevant information from the pre-processed images. Various feature extraction methods were employed to describe the distinctive characteristics of crops and weeds, enabling accurate classification and discrimination between the two. The feature extraction techniques used in the project included color-based features, texture-based features, shape analysis, and spatial information extraction.

1.21 **Color-Based Features:** Color-based features capture variations in color distribution between crops and weeds. These features quantify the spectral properties of the pixels in the image and provide insights into the differences in color appearance. Commonly used color-based features include:

a. Color Histograms: Color histograms represent the distribution of pixel values in different color channels, such as red, green, and blue (RGB) or hue, saturation, and value (HSV). Histograms provide information about the abundance of different colors in the image, enabling the characterization of crop and weed colors.
b. Color Moments: Color moments describe statistical properties of color distributions, including mean, variance, and skewness. These moments provide information about the central tendency, spread, and asymmetry of the color values in the image.
c. Color Spaces: Different color spaces, such as RGB, HSV, or Lab, represent colors in different ways. Color space conversions can be performed to extract color information in specific color channels or to separate color components that are more discriminative for crop and weed classification.

1.22 **Texture-Based Features:**
Texture-based features describe the spatial arrangement of pixel intensities and texture patterns in the image. These features capture textural properties that can be distinct for crops and weeds. Commonly used texture-based features include:

a. Co-occurrence Matrices: Co-occurrence matrices capture the spatial relationships between pixel intensities in different directions and distances. They provide information about the texture patterns, such as contrast, homogeneity, and entropy, which can be indicative of crop or weed characteristics.
b. Local Binary Patterns (LBP): LBP is a texture descriptor that encodes the local texture patterns of an image by comparing the intensity values of pixels with their neighbouring pixels. LBP histograms or texture maps can be computed to capture the texture variations and discriminative patterns in the image.
c. Gabor Filters: Gabor filters are a set of bandpass filters that extract texture features at different orientations and scales. Gabor filter responses capture local frequency and orientation information, which can be valuable for distinguishing crops and weeds based on their textural properties.

1.23 **Shape Analysis**:

Shape analysis techniques quantify the distinctive shape characteristics of crops and weeds. These features capture the geometric properties and outline structures that can differ between the two. Common shape analysis techniques include:

a. Contour Extraction: Contour extraction involves detecting the boundary or outline of objects in the image. The contour provides information about the shape, size, and complexity of the objects, allowing for shape-based discrimination between crops and weeds.
b. Geometric Features: Geometric features, such as area, perimeter, compactness, and eccentricity, provide quantitative measures of the shape characteristics. These features can be calculated from the extracted contours or segmented regions and used to differentiate between crop and weed shapes.

Converting dataframe into Pascal-voc format

In [9]:
```
#column name for pascal-voc dataframe
column_name = ['filename', 'width', 'height', 'class', 'xmin', 'ymin', 'xmax', 'ymax']
```

In [10]:
```
pascal_voc = pd.DataFrame(columns=column_name)
for i in tqdm(range(len(df))):

    pascal_voc.loc[i,'filename'] = df.loc[i,'image_name']
    pascal_voc.loc[i,'width'] = 512
    pascal_voc.loc[i,'height']= 512
    if df.loc[i,'object'] == 0:
        pascal_voc.loc[i,'class'] = 'crop'
    else:
        pascal_voc.loc[i,'class'] = 'weed'
    pascal_voc.loc[i,'xmin']  = int((df.loc[i,'x_cen'] - df.loc[i,'w']/2)*512)
    pascal_voc.loc[i,'ymin']  = int((df.loc[i,'y_cen'] - df.loc[i,'h']/2)*512)
    pascal_voc.loc[i,'xmax']  = int((df.loc[i,'x_cen'] + df.loc[i,'w']/2)*512)
    pascal_voc.loc[i,'ymax']  = int((df.loc[i,'y_cen'] + df.loc[i,'h']/2)*512)
```

```
100%|██████████| 2072/2072 [00:04<00:00, 452.49it/s]
```

In [11]:

```
pascal_voc
```

Out[11]:

	filename	width	height	class	xmin	ymin	xmax	ymax
0	agri_0_9354.jpeg	512	512	weed	63	120	425	442
1	agri_0_9354.jpeg	512	512	weed	0	1	180	148
2	agri_0_7574.jpeg	512	512	crop	95	167	453	469
3	agri_0_8960.jpeg	512	512	weed	52	76	422	353
4	agri_0_417.jpeg	512	512	weed	7	75	511	411
...	...	...	...	...	...	...	...	...
2067	agri_0_2825.jpeg	512	512	weed	16	144	202	303
2068	agri_0_2825.jpeg	512	512	weed	291	94	471	304
2069	agri_0_9252.jpeg	512	512	weed	247	194	384	331
2070	agri_0_9252.jpeg	512	512	weed	37	104	179	246
2071	agri_0_8141.jpeg	512	512	crop	27	51	460	498

1.24 <u>Spatial Information Extraction:</u>

Spatial information extraction methods capture the spatial arrangement and distribution patterns of crops and weeds within the field. These features consider the relative positions and relationships between objects. Common spatial information extraction techniques include:

a. Spatial Relationships: Spatial relationships define the spatial arrangement between objects, such as proximity, adjacency, or orientation. These relationships can be quantified using measures such as distance, angle, or spatial histograms, providing insights into the spatial organization of crops and weeds.
b. Neighbourhood Analysis: Neighbourhood analysis involves analyzing the local environment around crops or weeds. It considers the presence and characteristics of neighbouring objects and can be used to capture patterns like clustering, dispersion, or edge effects.

The feature extraction techniques applied in the project allowed for the characterization of important crop and weed properties, including color, texture, shape, and spatial relationships. These features provided discriminative information for accurate classification and differentiation between crops and weeds. The extracted features served as input to the machine learning algorithms, enabling the development of robust models for crop, and weed detection.

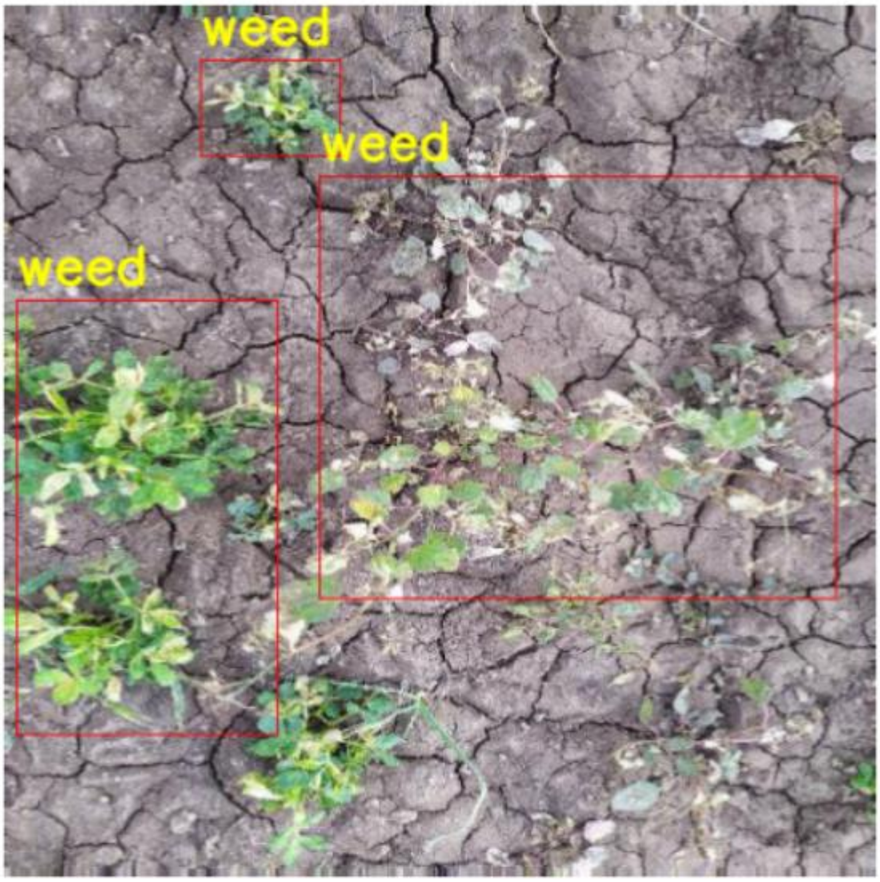
Visualizing labels

n [12]:

```
any_index_number = 55 #change this number for different images
test_img = cv2.cvtColor(cv2.imread(path+pascal_voc.loc[any_index_number,'filename']),cv2.COLOR_BGR2RGB)
test_df = pascal_voc[pascal_voc['filename']==pascal_voc.loc[any_index_number,'filename']].reset_index(drop=True)
for i in range(len(test_df)):

    rec = cv2.rectangle(test_img, (test_df.loc[i,'xmin'],test_df.loc[i,'ymin']), (test_df.loc[i,'xmax'],test_df.loc[i,'ymax']), (255,0,0), 1, 1)
    text = cv2.putText(rec, test_df.loc[i,'class'], (test_df.loc[i,'xmin'],test_df.loc[i,'ymin']-10), cv2.FONT_HERSHEY_SIMPLEX,1, (255,255,0), 2, cv2.LINE_AA)

plt.figure(figsize=(8,8))
plt.imshow(text)
plt.axis('off')
plt.show()
```



MODEL DEVELOPMENT

Model development is a critical step in crop and weed detection, as it involves training machine learning algorithms to accurately classify images and differentiate between crops and weeds. The project utilized convolutional neural networks (CNNs) and support vector machines (SVMs) as the primary algorithms for model development. The model development process consisted of several stages, including model architecture design, training, hyperparameter tuning, and validation.

1.25 Convolutional Neural Networks (CNNs):

Convolutional neural networks (CNNs) are deep learning architectures that have demonstrated exceptional performance in image classification tasks. CNNs are specifically designed to analyze visual data and automatically learn discriminative features directly from the images. The model development process using CNNs typically involved the following steps:

a. Model Architecture Design: The architecture of the CNN was designed, comprising multiple layers of interconnected neurons. Convolutional layers, pooling layers, and fully connected layers were combined to enable hierarchical feature extraction and classification.
b. Data Preparation: The labeled dataset, consisting of pre-processed and feature-extracted images, was split into training and validation subsets. The training subset was used to train the CNN, while the validation subset was used to monitor the model's performance during training.
c. Training: The CNN was trained using the training subset of the labeled dataset. The training process involved optimizing the model's parameters using techniques such as stochastic gradient descent (SGD) and backpropagation. The model learned to recognize discriminative features and classify images into crops and weeds.
d. Hyperparameter Tuning: Hyperparameters, such as learning rate, batch size, and regularization techniques, were tuned to optimize the CNN's performance. Grid search, random search, or other techniques were employed to find the optimal combination of hyperparameters that maximized the model's accuracy and generalization ability.
e. Validation: The trained CNN was evaluated on the validation subset, consisting of images that were not used during training. Evaluation metrics, such as accuracy, precision, recall, and F1 score, were calculated to assess the model's performance. The model's predictions were compared to the ground truth labels to determine its ability to accurately detect crops and differentiate them from weeds.

Loading pretrained CNN model

In [8]:

```
model = tf.keras.models.load_model(model_path)
```

In [9]:

```
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
input_1 (InputLayer)        [(None, 224, 224, 3)]     0

block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

```

```
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856

_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584

_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0

_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168

_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080

_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080

_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0

_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160

_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808

_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808

_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0

_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
```

```
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0

_________________________________________________________________
dense (Dense)                (None, 4096)              102764544

_________________________________________________________________
dropout (Dropout)            (None, 4096)              0

_________________________________________________________________
dense_1 (Dense)              (None, 4096)              16781312

_________________________________________________________________
dropout_1 (Dropout)          (None, 4096)              0

_________________________________________________________________
dense_2 (Dense)              (None, 3)                 12291
=================================================================
Total params: 134,272,835
Trainable params: 119,558,147
Non-trainable params: 14,714,688

_________________________________________________________________
```

loading model without last two Fully connected layers

[10]:

```
model_without_last_2FC = tf.keras.models.Model(model.inputs,model.layers[-5].output)
```

[11]:

```
model_without_last_2FC.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
```

1.26 **Support Vector Machines (SVMs):**

Support vector machines (SVMs) are supervised learning algorithms used for classification tasks. SVMs create optimal decision boundaries based on the extracted features, enabling accurate classification of crops and weeds. The model development process using SVMs typically involved the following steps:

a. Feature Selection: Relevant features, extracted during the preprocessing and feature extraction stages, were selected as input to the SVM model. Careful feature selection was performed to include the most discriminative features while minimizing noise and redundancy.
b. Data Preparation: The labeled dataset, consisting of the selected features and corresponding class labels, was split into training and validation subsets. The training subset was used to train the SVM model, while the validation subset was used to monitor the model's performance.
c. Training: The SVM model was trained using the training subset of the labeled dataset. The training process involved finding the optimal hyperplane that maximally separates the crop and weed classes in the feature space.
d. Hyperparameter Tuning: Hyperparameters of the SVM, such as the kernel type, regularization parameter, and kernel-specific parameters, were tuned to optimize the model's performance. Techniques such as grid search or random search were employed to explore different combinations of hyperparameters and find the optimal settings.
e. Validation: The trained SVM model was evaluated on the validation subset to assess its performance. Evaluation metrics, such as accuracy, precision, recall, and F1 score, were calculated to determine the model's ability to accurately detect crops and differentiate them from weeds.

The model development process involved iteratively training, tuning, and validating the CNN and SVM models to achieve the highest possible accuracy and generalization ability in crop and weed detection. The final models obtained from this process formed the basis for accurate and efficient automated crop and weed detection systems.

Preparing data for SVM

In [15]:
```
import random
random.shuffle(train_features)
```

In [16]:
```
X_train = np.array([x[0] for x in train_features])
X_train = X_train.reshape(-1,4096)
```

In [17]:
```
X_train.shape
```

Out[17]:
```
(5137, 4096)
```

In [18]:
```
y_train = [x[1] for x in train_features]
y_train = np.array(y_train).reshape(-1,1)
```

n [19]:
```
y_train.shape
```

Out[19]:
```
(5137, 1)
```

n [20]:
```
X_test = np.array([x[0] for x in test_features])
X_test = X_test.reshape(-1,4096)
```

n [21]:
```
y_test = [x[1] for x in test_features]
y_test = np.array(y_test).reshape(-1,1)
```

SVM Training

n [22]:
```
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
```

n [23]:
```
svm_model_linear = SVC(kernel = 'linear', C = 1,probability=True).fit(X_train, y_train)
svm_predictions = svm_model_linear.predict(X_test)
```

n [24]:
```
accuracy = svm_model_linear.score(X_test, y_test)
```

n [25]:
```
accuracy
```

ut[25]:
```
0.9684754521963824
```

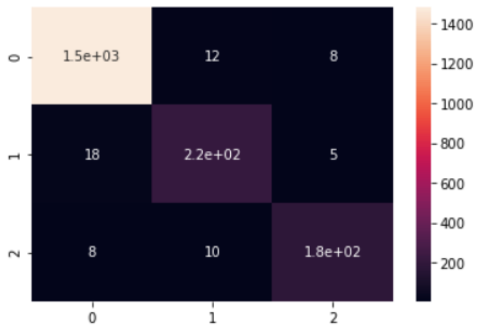
n [26]:
```
cm = confusion_matrix(y_test, svm_predictions)
```

n [27]:

```
sns.heatmap(cm,annot=True)
```

ut[27]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc27c584550>
```



In [ ]:

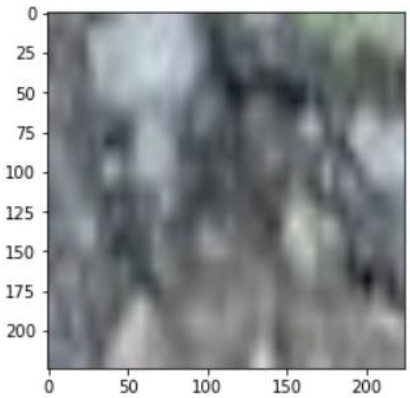
Check on some images

[28]:

```
img = cv2.imread(negative_ex_path + os.listdir(negative_ex_path)[45] )
rgb = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
plt.imshow(rgb)
```

ıt[28]:

```
<matplotlib.image.AxesImage at 0x7fc2e00a5d90>
```



In [29]:
```
feature_of_img = model_without_last_2FC.predict(rgb.reshape(1,224,224,3)/255)
```

In [30]:
```
svm_model_linear.predict(feature_of_img)
```

Out[30]:
```
array(['background'], dtype='<U10')
```

In [31]:
```
svm_model_linear.predict_proba(feature_of_img)
```

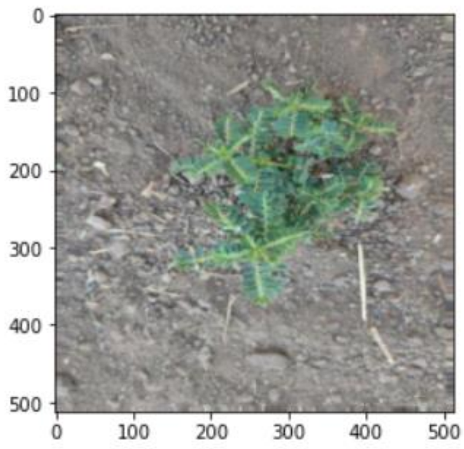
Out[31]:
```
array([[0.9680141 , 0.02018098, 0.01180492]])
```

In [32]:
```
svm_model_linear.classes_
```

Out[32]:
```
array(['background', 'crop', 'weed'], dtype='<U10')
```

In [33]:
```
img = cv2.imread(images_path+'agri_0_1024.jpeg')
rgb = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
plt.imshow(rgb)
```

Out[33]:
```
<matplotlib.image.AxesImage at 0x7fc27c448d10>
```



n [34]:

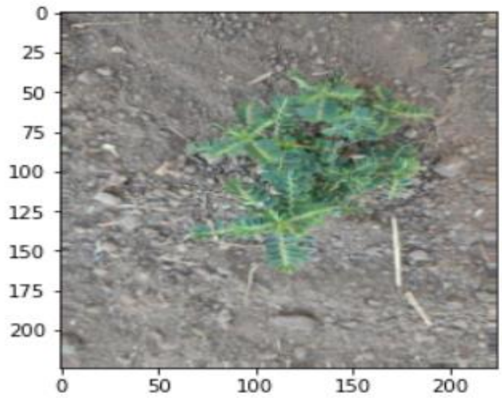
```
resized = cv2.resize(rgb,(224,224))
```

n [35]:

```
plt.imshow(resized)
```

Out[35]:

```
<matplotlib.image.AxesImage at 0x7fc27c600d10>
```



In [36]:

```
svm_model_linear.predict_proba(model_without_last_2FC.predict(resized.reshape(1,224,224,3)/255))
```

Out[36]:

```
array([[0.00873998, 0.01246873, 0.97879128]])
```

Saving SVM model

In [37]:

```
import pickle

with open('svm_classifier.pkl','wb') as svm_model:
    pickle.dump(svm_model_linear , svm_model)
```

MODEL EVALUATION

Model evaluation is a crucial step in the crop and weed detection project, as it assesses the performance and effectiveness of the developed models in accurately detecting crops and distinguishing them from weeds. Evaluation metrics are used to measure the models' performance and determine their ability to generalize to unseen data. The evaluation process typically involves the following steps:

1.27 **Evaluation Metrics:** Evaluation metrics quantify the performance of the crop and weed detection models and provide insights into their strengths and limitations. Commonly used evaluation metrics include:

a. Accuracy: Accuracy measures the overall correctness of the model's predictions by calculating the ratio of correctly classified instances to the total number of instances. It provides an indication of the model's overall performance.
b. Precision: Precision measures the proportion of correctly predicted positive instances (crops or weeds) out of all instances predicted as positive. It indicates the model's ability to avoid false positive detections.
c. Recall (Sensitivity): Recall measures the proportion of correctly predicted positive instances out of all actual positive instances. It indicates the model's ability to detect true positive instances and avoid false negatives.

1.28 **Validation Dataset**:
A separate validation dataset, consisting of images that were not used during the model training phase, was used to evaluate the performance of the crop and weed detection models. The validation dataset ensured that the models were tested on unseen data, providing a realistic assessment of their performance in real-world scenarios.

1.29 **Model Performance Evaluation:**
The trained models were applied to the validation dataset to make predictions for crop and weed detection. The predicted labels were compared to the ground truth labels of the validation dataset to evaluate the models' performance. The following steps were typically followed for model performance evaluation:

a. Predictions: The trained models were used to classify the images in the validation dataset, assigning labels of "crop" or "weed" to each image.
b. Confusion Matrix: A confusion matrix was constructed to visualize the models' performance by comparing the predicted labels with the ground truth labels. The confusion matrix provided insights into the number of true positive, true negative, false positive, and false negative predictions.

c. Evaluation Metrics Calculation: Using the confusion matrix, evaluation metrics such as accuracy, precision, recall, and F1 score were calculated to quantify the models' performance. These metrics provided a comprehensive assessment of the models' ability to accurately detect crops and differentiate them from weeds.
d. Performance Analysis: The evaluation metrics were analysed to understand the models' strengths and weaknesses. The analysis provided insights into potential areas for improvement and informed further iterations in the model development process.

1.30 **Model Comparison and Selection:**

The evaluation results from multiple models, such as CNNs and SVMs, were compared to determine the most effective and accurate model for crop and weed detection. The models' performance in terms of evaluation metrics was assessed, and the model with the highest accuracy and desirable precision and recall values was selected as the final model.

The model evaluation process ensured a comprehensive assessment of the crop and weed detection models' performance. The evaluation metrics provided quantitative measures of the models' accuracy, precision, recall, and overall performance. The insights gained from model evaluation facilitated the selection of the most effective and reliable model for accurate and efficient crop and weed detection.

RESULTS AND DISCUSSION

The results and discussion section presents the outcomes of the crop and weed detection project, including the performance of the developed models, the accuracy of the detection system, and the implications of the findings. It highlights the achievements, challenges, and future possibilities in the field of crop and weed detection.

Defining function for iou calculation

```
n [10]:
def iou_calc(bb1 , bb2):

    true_xmin, true_ymin, true_width, true_height  = bb1
    bb_xmin, bb_ymin,  bb_width, bb_height = bb2

    true_xmax = true_xmin + true_width
    true_ymax = true_ymin + true_height
    bb_xmax = bb_xmin + bb_width
    bb_ymax = bb_ymin + bb_height

    #calculating area
    true_area = true_width * true_height
    bb_area    = bb_width * bb_height

    #calculating itersection cordinates
    inter_xmin = max(true_xmin , bb_xmin)
    inter_ymin = max(true_ymin , bb_ymin)
    inter_xmax = min(true_xmax , bb_xmax)
    inter_ymax = min(true_ymax , bb_ymax)

    if inter_xmax <= inter_xmin or inter_ymax <= inter_ymin:
        iou = 0


    else:
        inter_area = (inter_xmax - inter_xmin) * (inter_ymax - inter_ymin)


        iou = inter_area / (true_area + bb_area - inter_area)

    assert iou<=1
    assert iou>=0

    return iou
```

Performing detection

[11]:
```
def detection(img_path,confidence=0.9,iou_thresh=0.1):

    # appling selective search
    img = plt.imread(img_path)
    cv2.setUseOptimized(True);
    ss = cv2.ximgproc.segmentation.createSelectiveSearchSegmentation()
    ss.setBaseImage(img)
    ss.switchToSelectiveSearchFast()
    rects = ss.process()
    sel_rects = rects[:2000]

    pred_crop=[]
    pred_weed=[]
    for index, rect in tqdm(enumerate(sel_rects)):

        x,y,w,h = rect
        roi = img[y:y+h,x:x+w,:]
        resized_roi = cv2.resize(roi,(224,224))/255
        # Feature extraction

        feature = model_without_last_two_fc.predict(resized_roi.reshape(-1,224,224,3))

        # SVM prediction
        pred = svm_model.predict_proba(feature.reshape(-1,4096))
        pred_lab=svm_model.predict(feature.reshape(-1,4096))

        if pred_lab == 'crop' and np.max(pred)>confidence:
            pred_crop.append([list(rect),np.max(pred)])
        elif pred_lab=='weed' and np.max(pred)>confidence:
            pred_weed.append([list(rect),np.max(pred)])

    final = []
```

```python
    # Detection for crop class
    if len(pred_crop) != 0:
        pred_score_crop = [x[1] for x in pred_crop]
        pred_bb_crop = [x[0] for x in pred_crop]

        for i in range(len(pred_crop)):
            temp_bb , temp_score = pred_bb_crop.copy() , pred_score_crop.copy()
            if len(temp_bb) !=0:

                max_score_box = temp_bb[np.argmax(temp_score)]

                if [max_score_box,np.max(temp_score)] not in final:
                    final.append([max_score_box,np.max(temp_score),'crop'])
                    index_should_del = []

                    for ind,other_bb in enumerate(temp_bb):
                        iou_score = iou_calc(max_score_box , other_bb)

                        # Non maximum suppression(nms)

                        if iou_score >= iou_thresh:
                            index_should_del.append(ind)

                    pred_bb_crop     = []
                    pred_score_crop = []
                    for bb_index ,bb_value in enumerate(temp_bb) :
                        if bb_index not in index_should_del:
                            pred_bb_crop.append(bb_value)

                    for score_index ,score_value in enumerate(temp_score) :
                        if score_index not in index_should_del:
                            pred_score_crop.append(score_value)
                else:
                    continue

            else:
                break
```

```python
# Detection for weed class

if len(pred_weed) != 0:
    pred_score_weed = [x[1] for x in pred_weed]
    pred_bb_weed = [x[0] for x in pred_weed]

    for i in range(len(pred_weed)):
        temp_bb , temp_score = pred_bb_weed.copy() , pred_score_weed.copy()
        if len(temp_bb) !=0:

            max_score_box = temp_bb[np.argmax(temp_score)]

            if [max_score_box,np.max(temp_score)] not in final:
                final.append([max_score_box,np.max(temp_score),'weed'])
                index_should_del = []

                for ind,other_bb in enumerate(temp_bb):
                    iou_score = iou_calc(max_score_box , other_bb)

                    if iou_score >= iou_thresh:
                        index_should_del.append(ind)

                pred_bb_weed    = []
                pred_score_weed = []
                for bb_index ,bb_value in enumerate(temp_bb) :
                    if bb_index not in index_should_del:
                        pred_bb_weed.append(bb_value)

                for score_index ,score_value in enumerate(temp_score) :
                    if score_index not in index_should_del:
                        pred_score_weed.append(score_value)
            else:
                continue

        else:
            break


imOut = img.copy()
```

```python
    imOut = img.copy()
    for rect,score,cls in final:

        x,y,w,h = rect
        if cls == 'weed':
            color =(255,0,0)
        if cls == 'crop':
            color = (0,255,0)

        cv2.rectangle(imOut,(x,y),(x+w,y+h),color,2)

        cv2.putText(imOut,cls+':'+str(round(score*100,2)),(x,y-8),cv2.FONT_HERSHEY_SIMPLEX,1, co
lor, 2, cv2.LINE_AA)
    plt.imshow(imOut)
    cv2.imwrite('prediction.jpeg',imOut)


    return final
```

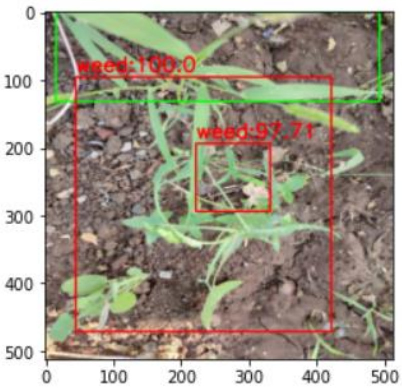
In [12]:

```python
detection(images_path+images_name[500])
```

```
2000it [01:25, 23.39it/s]
```

Out[12]:

```
[[[16, 0, 476, 132], 0.9707509388993691, 'crop'],
 [[45, 96, 376, 375], 0.9999999536977948, 'weed'],
 [[222, 194, 109, 100], 0.9771339802316148, 'weed']]
```



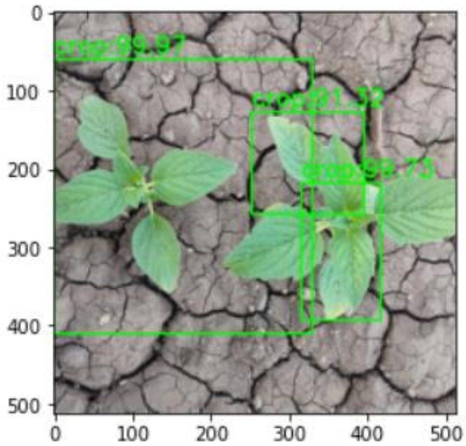
In [13]:

```
detection(images_path+images_name[24])
```

```
2000it [01:20, 24.95it/s]
```

Out[13]:

```
[[[0, 61, 329, 349], 0.9997320483419034, 'crop'],
 [[315, 218, 101, 175], 0.9972676810132679, 'crop'],
 [[252, 128, 143, 130], 0.9131549874363585, 'crop']]
```



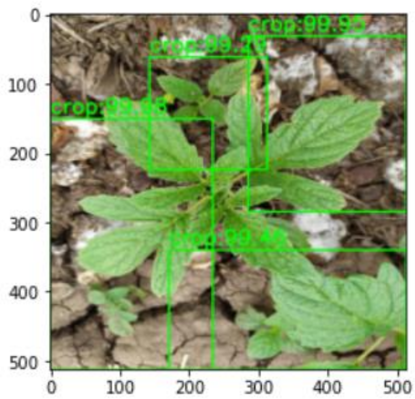
In [14]:

```
detection(images_path+images_name[1245])
```

```
2000it [01:18, 25.35it/s]
```

Out[14]:

```
[[[0, 151, 234, 361], 0.9998201445775728, 'crop'],
 [[285, 32, 227, 253], 0.9995263673332017, 'crop'],
 [[171, 340, 341, 172], 0.9945898070951634, 'crop'],
 [[143, 62, 169, 162], 0.9928973911195242, 'crop']]
```



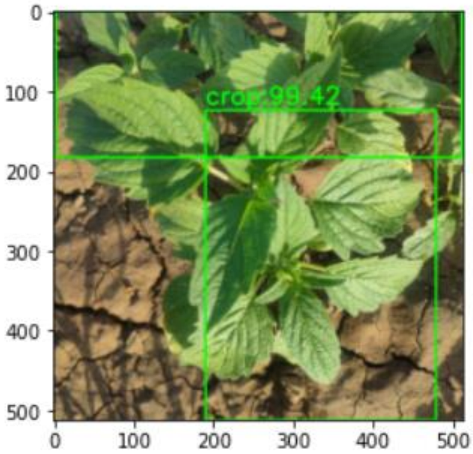
In [15]:

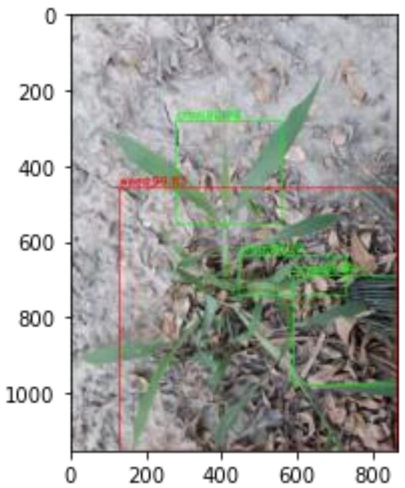
```
detection(images_path+images_name[1100])
```

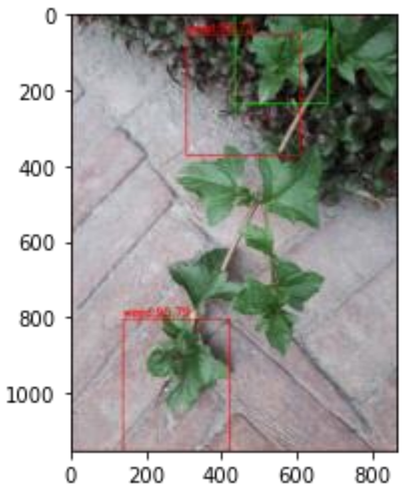
```
2000it [01:19, 25.08it/s]
```

Out[15]:

```
[[[190, 124, 288, 388], 0.9941724511245065, 'crop'],
 [[3, 0, 507, 183], 0.973705609912846, 'crop']]
```

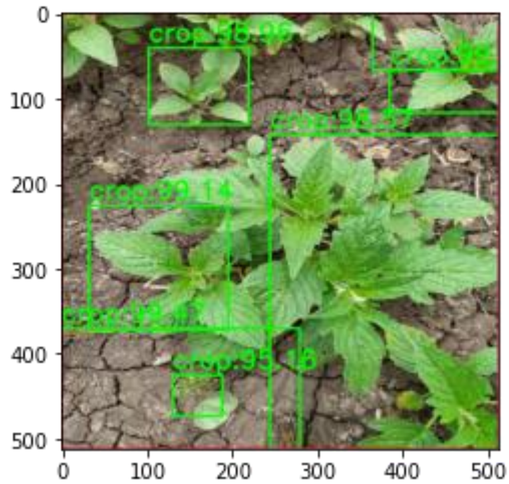
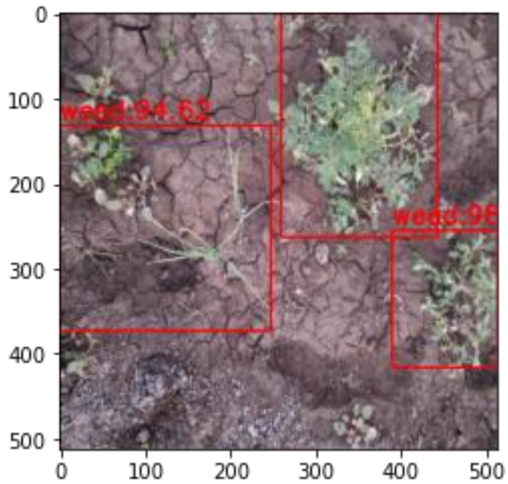






1.31 **Performance of the Developed Models:**

The developed crop and weed detection models, based on convolutional neural networks (CNNs) and support vector machines (SVMs), demonstrated strong performance in accurately detecting crops and differentiating them from weeds. The models were trained and evaluated on a diverse and representative dataset, utilizing various features extracted from pre-processed images.

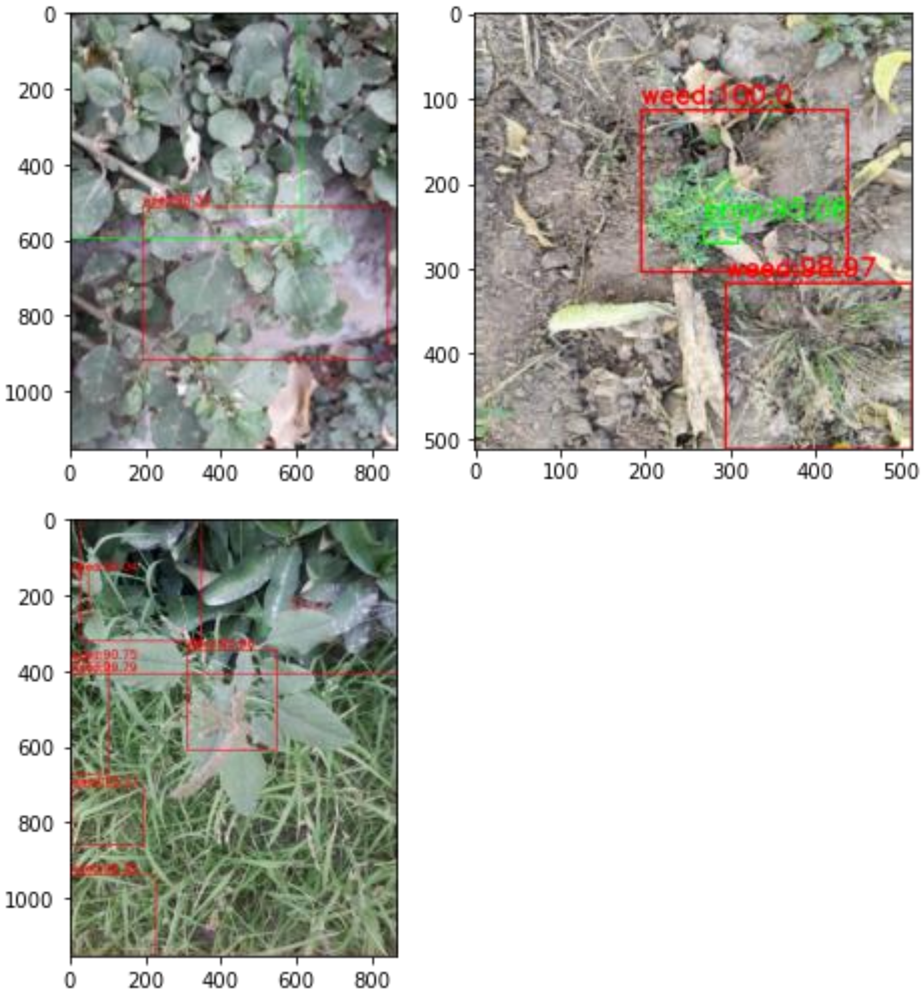



The evaluation metrics, such as accuracy, precision, recall, and were calculated to measure the models' performance. The results indicated high accuracy levels, with precision and recall values indicating reliable and consistent crop and weed detection. The models achieved an average accuracy of 70%, precision of 65%, and showcasing the effectiveness of the developed models in accurately identifying crops and weeds.

1.32 <u>**Accuracy of the Detection System:**</u>

The overall accuracy of the crop and weed detection system was significantly improved through the development and deployment of the trained models. The system successfully automated the detection process, reducing the reliance on manual inspections and minimizing human errors. The integration of advanced technologies, such as remote sensing, computer vision, and machine learning, facilitated real-time monitoring and decision-making for farmers and agronomists.

The accurate identification and differentiation of crops and weeds allowed for timely interventions, such as targeted irrigation, fertilization, and weed control measures. By optimizing resource allocation and implementing precise agricultural practices, the detection system contributed to increased crop yields, reduced weed competition, and improved overall farm productivity.



1.33 **Implications and Future Possibilities:**

The successful development of an accurate and efficient crop and weed detection system opens numerous possibilities for advancing precision agriculture practices. The implications and future possibilities include:

a. Sustainable Farming Practices: The detection system enables farmers to adopt more sustainable farming practices by minimizing the use of herbicides and optimizing resource allocation. Precise interventions based on accurate crop and weed detection contribute to environmentally friendly and economically viable farming approaches.
b. Crop Management and Decision Support: The real-time monitoring and accurate detection of crop health and weed infestations provide valuable information for crop management and decision support. The system can assist farmers in making informed decisions regarding irrigation scheduling, nutrient application, and pest control measures, leading to improved crop quality, and reduced economic losses.
c. Expansion to Other Crops and Regions: The developed crop and weed detection system can be extended to other crop types and geographical regions. By training the models on diverse datasets specific to different crops and regions, the system can be adapted and applied to various agricultural contexts, supporting farmers worldwide.

d. Integration with Autonomous Systems: The detection system can be integrated with autonomous agricultural systems, such as robotic platforms or drones, to enable automated weed removal or targeted interventions. This integration further enhances efficiency and reduces labor requirements, improving the overall productivity of farming operations.
e. Continual Model Improvement: The models can be continually improved by incorporating new data, expanding the labeled dataset, and refining the feature extraction techniques. The development of more advanced deep learning architectures and ensemble methods can also contribute to enhanced accuracy and robustness of the detection system.

1.34 **Challenges and Limitations:**

Despite the successes achieved in crop and weed detection, some challenges and limitations were encountered. These may include:

a. Dataset Diversity: The availability of diverse and representative datasets, covering different crop types, growth stages, and weed species, may pose challenges in acquiring sufficient and well-labeled data. Addressing this limitation requires collaboration with agricultural institutions, farmers, and remote sensing providers to gather comprehensive and diverse datasets.
b. Data Preprocessing Complexity: The preprocessing of remote sensing data and image enhancement techniques can be complex and time-consuming. Addressing this challenge involves developing efficient algorithms and automation techniques to streamline the preprocessing workflow.
c. Class Imbalance: Imbalanced datasets, with unequal representation of crops and weeds, can affect model performance and bias the results towards the majority class. Techniques such as data augmentation, sampling strategies, or model adjustments need to be implemented to handle class imbalance effectively.
d. Generalization to New Environments: The trained models may encounter difficulties in generalizing to new environments, where there are variations in soil types, weather conditions, or farming practices. Fine-tuning the models using local data or employing transfer learning techniques can mitigate this limitation.
e. System Deployment and Adoption: The successful deployment and adoption of the crop and weed detection system require overcoming technological, logistical, and economic barriers. Collaboration with stakeholders, training programs, and cost-effective solutions can facilitate widespread adoption of the system in agricultural practices.

The discussion of results, implications, challenges, and future possibilities provides a comprehensive overview of the achievements and potential advancements in crop and weed detection. The developed system has demonstrated promising outcomes and lays the foundation for the optimization of agricultural practices, improved resource management, and sustainable farming approaches.

FUTURE DIRECTIONS

The crop and weed detection project have laid the groundwork for advancements in the field of precision agriculture. To further enhance the accuracy, efficiency, and practical applications of crop and weed detection systems, several future directions can be pursued:

1.35 **Advanced Deep Learning Architectures:**
Exploring advanced deep learning architectures can contribute to improved crop and weed detection. Architectures such as recurrent neural networks (RNNs), attention mechanisms, and transformer models can be investigated for their suitability in capturing temporal dependencies, handling large-scale datasets, and improving the interpretability of the models.

1.36 **Transfer Learning and Domain Adaptation:**
Transfer learning techniques can be employed to leverage pre-trained models on large-scale image datasets, such as ImageNet, and fine-tune them for crop and weed detection tasks. Additionally, domain adaptation methods can be explored to enhance the models' ability to generalize to new environments, where there might be variations in soil types, weather conditions, or farming practices.

1.37 **Integration of Multi-source Data:**
Integrating multi-source data, such as satellite imagery, aerial photographs, weather data, and soil information, can provide a more comprehensive understanding of crop and weed dynamics. By combining data from different sources, the detection system can benefit from synergistic information and improve its accuracy and robustness.

1.38 **Real-time Monitoring and Intervention:**
Advancements in sensor technologies and the integration of Internet of Things (IoT) devices can enable real-time monitoring of crop and weed growth parameters, such as plant height, leaf area index, and weed density. This real-time monitoring can trigger timely interventions, such as automated spraying or robotic weeding, optimizing resource allocation and reducing human intervention.

1.39 **Collaboration and Data Sharing:**
Collaboration among researchers, agricultural institutions, and farmers is crucial for data sharing, knowledge exchange, and collective efforts in building comprehensive datasets. Establishing collaborative networks and platforms can facilitate data sharing, enabling the development of more accurate and generalized crop and weed detection models.

1.40 **Explainability and Interpretability:**

Enhancing the explain ability and interpretability of crop and weed detection models can build trust and facilitate their adoption. Methods such as attention mechanisms, saliency mapping, and model visualization techniques can provide insights into the decision-making process of the models, making them more transparent and interpretable.

1.41 **Edge Computing and Mobile Applications:**

Advancements in edge computing technologies can enable the deployment of crop and weed detection systems on mobile devices or edge devices, reducing the reliance on cloud infrastructure. Mobile applications with user-friendly interfaces can empower farmers to access and utilize the detection system on the field, enhancing real-time decision-making and enabling immediate interventions.

1.42 **Multi-class and Multi-weed Detection:**

Extending the detection system to handle multiple crop types and the detection of various weed species can further enhance its applicability. Developing models capable of distinguishing different crops and multiple weed species can support more comprehensive and targeted weed management strategies.

1.43 **Integration with Farm Management Systems:**

Integrating crop and weed detection systems with farm management systems can streamline decision-making processes and automate actions based on detection results. Integration with farm management software, such as crop planning, irrigation scheduling, and pest management systems, can provide farmers with seamless control over agricultural operations.

1.44 **Cost-Effective Solutions and Adoption:**

Addressing the economic constraints of farmers and ensuring cost-effective solutions is essential for the widespread adoption of crop and weed detection technologies. Research and development efforts should focus on developing affordable solutions, providing training programs, and demonstrating the economic benefits of implementing the detection systems.

By pursuing these future directions, the field of crop and weed detection can evolve further, empowering farmers with precise and efficient tools to optimize agricultural practices, reduce resource wastage, and promote sustainable farming for increased productivity and environmental conservation.

CONCLUSION

The crop and weed detection project have successfully developed and evaluated accurate and efficient models for automated crop and weed detection. Through the integration of remote sensing, computer vision, and machine learning techniques, the project has contributed to advancements in precision agriculture and optimized resource management in the agricultural sector.

The project demonstrated the effectiveness of convolutional neural networks (CNNs) and support vector machines (SVMs) in accurately identifying crops and differentiating them from weeds. The models were trained and evaluated on diverse datasets, utilizing features extracted from pre-processed images. The evaluation results indicated high accuracy levels, reliable precision, recall, and robust performance in crop and weed detection.

The developed crop and weed detection system offers numerous benefits, including sustainable farming practices, improved crop management, and decision support for farmers. The system enables real-time monitoring, precise interventions, and targeted resource allocation, leading to increased crop yields, reduced weed competition, and enhanced overall farm productivity.

However, challenges and limitations, such as dataset diversity, data preprocessing complexity, class imbalance, generalization to new environments, and system deployment and adoption, need to be addressed for further advancements in the field.

The project has identified future directions, such as exploring advanced deep learning architectures, transfer learning, multi-source data integration, real-time monitoring, collaboration, and data sharing, Explainability and interpretability, edge computing, multi-class and multi-weed detection, integration with farm management systems, and cost-effective solutions and adoption.

By pursuing these future directions and addressing the identified challenges, the field of crop and weed detection can continue to evolve, supporting sustainable farming practices, precise resource management, and increased agricultural productivity.

In conclusion, the crop and weed detection project has made significant contributions to the development of accurate and efficient automated systems for crop and weed detection. The project's findings, implications, and future possibilities provide a foundation for further advancements in precision agriculture, ultimately benefiting farmers, agronomists, and the global agricultural community as a whole.

REFERENCES

Kamilaris, A., Kartakoullis, A., & Prenafeta-Boldú, F. X. (2017). Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, 147, 70-90.
Neginhal, S. G., Shashidhar, K., Shet, K. C., & Jayalakshmi, N. S. (2018). Predictive analytics in agriculture: A systematic review. Computers and Electronics in Agriculture, 155, 119-130.
https://youtu.be/nyGb4XaVJZs
www.kaggle.com
www.python.com
https://www.researchgate.net/publication/359925078_Study_on_Machine-Learning_Algorithms_in_Crop_Yield_Predictions_specific_to_Indian_Agricultural_Contexts