Project Report

on

Detecting Rice Crop Diseases with YOLO

Submitted By

Adapa Yasaswi **(20481A0503)**

Annam Manoj Venkata Sai **(20481A0509)**

Daram Srenitha **(21485A0504)**

1.Introduction

1.1.Overview :

Increasing grain production is essential to those areas where food is scarce. Increasing grain production by controlling crop diseases in time should be effective. To construct a prediction model for plant diseases and to build a real-time application for crop diseases in the future, a deep learning-based image detection architecture with a custom backbone was proposed for detecting plant diseases.

Rice is one of India's most significant food crops. Rice is the most well-known food ate by far most of the populace consistently. Not withstanding, diseases annihilate a significant part of the harvest every year. Subsequently, early recognition and investigation of such sicknesses will assist us with having a better return and a better harvest.

In order to get a good amount of crop, we need to detect the disease at the earliest. Basically, crop disease diagnosis depends on different characteristics like color, shape, texture, etc. Here the person can capture the images of the crop and then the image will be sent to the trained YOLO model. The model analyzes the image and detects crop diseases like Bacterial Blight, Leaf Smut,White Tip.

Fig-1 : Bacterial Blight

Fig-2 : Leaf Smut

Fig-3 : White Tip

1.2.Purpose :

Rice crops are a huge asset in terms of food consumption for everybody. Therefore, it is basic to guarantee that the rice crops are liberated from sickness. On the off chance that contamination emerges, it is basic to analyze illnesses at a beginning phase. All things considered, data is moved through a few layers in deep learning.

The previous layer's result is utilized as contribution for the following layer. It utilizes enactment capacities to spread the gained highlights starting with one layer then onto the next until the result layer gives the expected results. The areas that follow sum up endeavors on plant sickness characterization and discovery, alongside its advantages and downsides.

The YOLO model will detect the disease with which the rice crop has been infected.In this project a deep learning model is developed which will interpret the given rice crop image and detect whether the rice crop has been affected by Bacterial Blight or Leaf Smut or White Tip. We offer a strategy for recognizing illnesses utilizing the YOLO object identification algorithm in this project.

2.Literature Survey

2.1.Existing Approaches :

There are several existing methods to detect rice crop diseases.Some of them are :

Remote Sensing : Remote sensing techniques using satellite or drone imagery can be employed to monitor large rice fields. Different spectral bands are analyzed to detect changes in plant health and identify potential diseases.

Spectroscopy : Near-infrared (NIR) and infrared spectroscopy can be utilized to measure the biochemical composition of plant tissues. By analyzing the spectral signatures, researchers can identify diseases based on characteristic changes in the spectra.

Image Processing and Computer Vision : Computer vision techniques can be used to process images of rice plants and detect disease symptoms. Deep learning algorithms, such as Convolutional Neural Networks, have shown promising results in automatically identifying diseases from images.

DNA-based Diagnosis : Polymerase Chain Reaction (PCR) and other DNA-based methods can identify the presence of specific pathogens or pathogens' DNA in the rice plants, enabling accurate and early disease detection.

IoT and Sensor Technologies : Deploying IoT devices and sensors in the fields allows continuous monitoring of various environmental factors (temperature, humidity, etc.) and plant health parameters. Any abnormality can indicate the presence of diseases.

Machine Learning Models : Utilizing historical data on rice crop diseases, machine learning models can be trained to detect diseases based on various input features like environmental conditions, plant characteristics, and disease symptoms.

It is important to note that the effectiveness of these methods may vary depending on the specific disease, the location, and the available resources. Researchers and technologists continue to work on improving the accuracy and accessibility of these disease detection methods to support sustainable agriculture practices.

2.2.Proposed Solution :

Our project aims at developing a deep learning model in order to predict the three rice crop diseases Bacterial Blight , Leaf Smut and White Tip at any location.We have used different python libraries like Tensorflow , Keras , Numpy and so on.The model is trained using Convolution Neural Network layers (CNN) and finally a yolo model has been generated in order to predict the disease of given rice crop image.The model predicted the disease of rice crop images present in the testing dataset accurately using YOLO algorithm.This deep learning model completely works based on yolo algorithm.

3.Theoretical Analysis

3.1.Block Diagram :

Diagrammatic Overview of the project :

In this phase we present a comprehensive understanding of the deep learning techniques and methodologies used in this project for predicting the rice crop diseases.We explore the underlying concepts ,algorithms involved in creating the best rice crop disease detector which will predict the rice crop disease accurately.

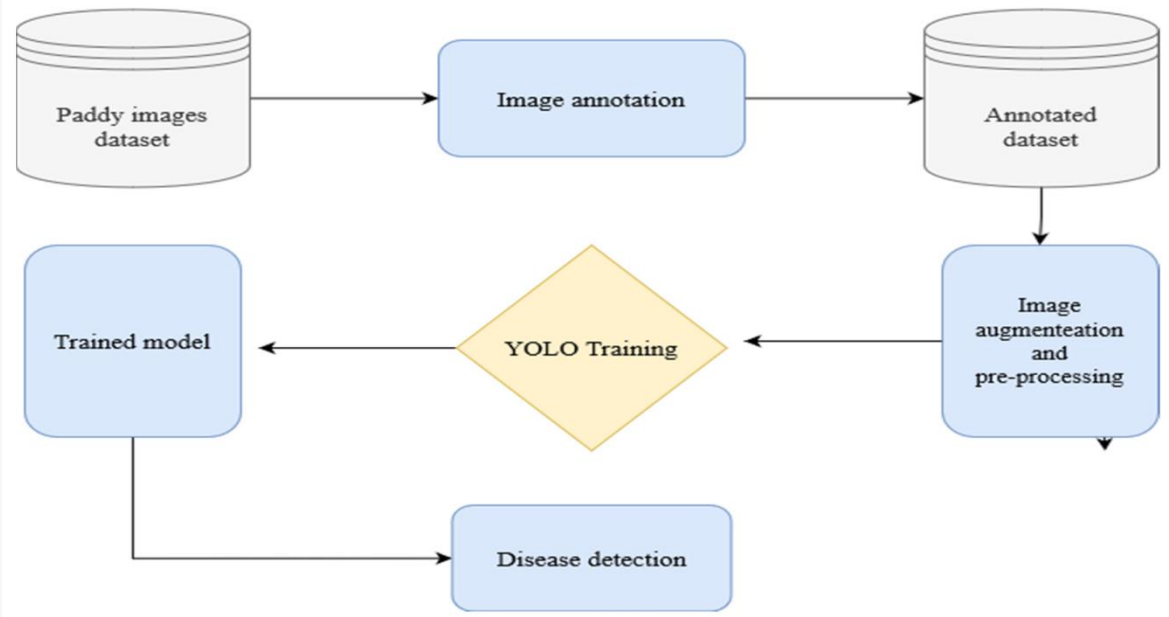

Fig-4 : Block Diagram of Rice Crop Disease Detection using YOLO

3.2. Hardware / Software designing :

Hardware Requirements:

GPU (Graphics Processing Unit) : YOLO's real-time performance benefits significantly from a powerful GPU. GPUs accelerate the computations required for deep learning tasks, making training and inference faster. NVIDIA GPUs are commonly used for deep learning tasks due to their compatibility with popular deep learning frameworks like TensorFlow and PyTorch.

CPU (Central Processing Unit) : While a powerful CPU isn't as critical as a GPU for inference, it's still important for managing overall system performance and training processes.

Memory (RAM) : Deep learning models like YOLO can be memory-intensive, especially during training. A sufficient amount of RAM is important to avoid memory-related bottlenecks.

Storage : You'll need enough storage space to store your dataset, pre-trained models, and the results of your training processes.

Software Requirements:

Deep Learning Framework : YOLO implementations are available in popular deep learning frameworks like TensorFlow and PyTorch. Choose the framework you are comfortable with. YOLO versions such as YOLOv3 and YOLOv4 have been widely used for object detection tasks.

Tensor flow : TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers can easily build and deploy ML-powered applications.

Keras : Keras leverages various optimization techniques to make high-level neural network API easier and more performance. It supports multiple platforms and backends.It is a user-friendly framework that runs on both CPU and GPU.

Python : YOLO implementations are typically written in Python, so you'll need to have Python installed on your system.

Anaconda Installation : Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system.

Annotation Tools : You'll need tools to annotate your dataset with bounding boxes around the disease symptoms. LabelImg, VoTT (Visual Object Tagging Tool), or RectLabel are examples of annotation tools.

Install Microsoft's Visual Object Tagging Tool (VoTT) :
Head to VoTT [releases](https://github.com/Microsoft/VoTT/releases) and download and install the version for your operating system.'vott-2.x.x-win32.exe' for Windows users

Dataset : A well-labeled dataset containing images of rice plants with annotated disease symptoms is essential for training the YOLO model.

Pre-trained YOLO Model : You can start with a pre-trained YOLO model as a base for transfer learning. These models are often trained on large datasets for general object detection tasks and can be fine-tuned for rice crop disease detection.

Text Editor or IDE : A text editor or integrated development environment (IDE) is necessary for writing, editing, and running your Python scripts for training and inference.

Fig-5

Fig-6

Fig-7

Fig-8

4.Experimental Investigations

Experimental investigations involve the process of training and evaluating the deep learning model to predict the rice crop disease using the annotated rice crop training images.After the model gets trained then the model will be able to predict the disease of the rice crop image given for the testing.

(i) Project Structure :

We need to create the project structure the same as below to build our deep learning model.

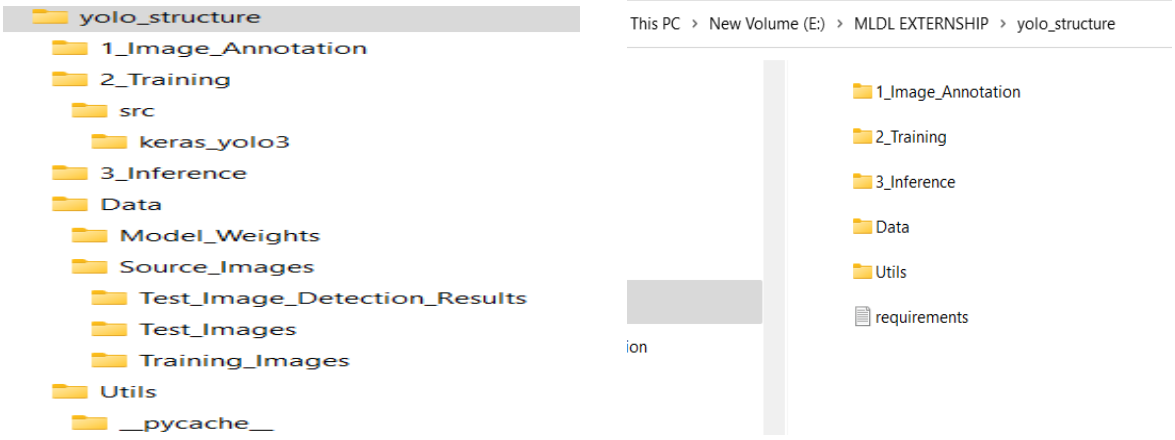

Fig-9,10 : Project Structure

(ii) Create Dataset :

- Download the Rice Crop Disease Dataset from Kaggle.
- To train the YOLO object detector on our dataset, copy the training images to [`yolostructure/Data/Source_Images/Training_Images`](/Data/Source_Images/Training_Images/).

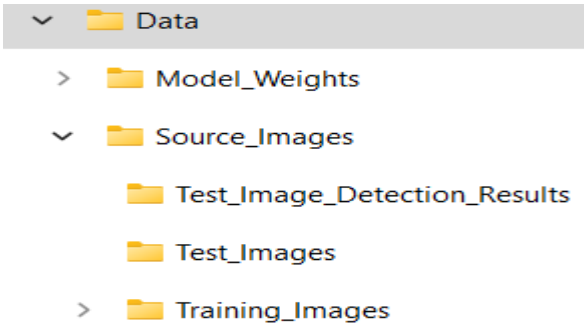

Fig-11: Training_Images folder

(iii) Annotate Images :

To make our detector learn, we first need to feed it some good training examples. We use Microsoft's Visual Object Tagging Tool (VoTT) to manually label images in our training folder [`yolostructure/Data/Source_Images/Training_Images`](/Data/Source_Images/Training_Images/).

Create A New Project in VoTT :

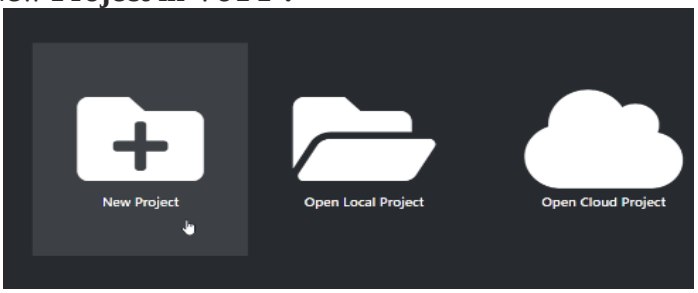

Fig-12 : Creating new project in VoTT

- Create a **New Project** and call it `Annotations`. It is highly recommended to use `Annotations` as your project name. If you like to use a different name for your project, you will have to modify the command line arguments of subsequent scripts accordingly.
- Under **Source Connection** choose **Add Connection** and put `Images` as **Display Name**. Under **Provider** choose **Local File System** and select [`yolostructure/Data/Source_Images/Training_Images`](/Data/Source_Images/Training_Images) and then **Save Connection**. For **Target Connection** choose the same folder as for **Source Connection**. And Tags of diseases.Hit **Save Project** to finish project creation.

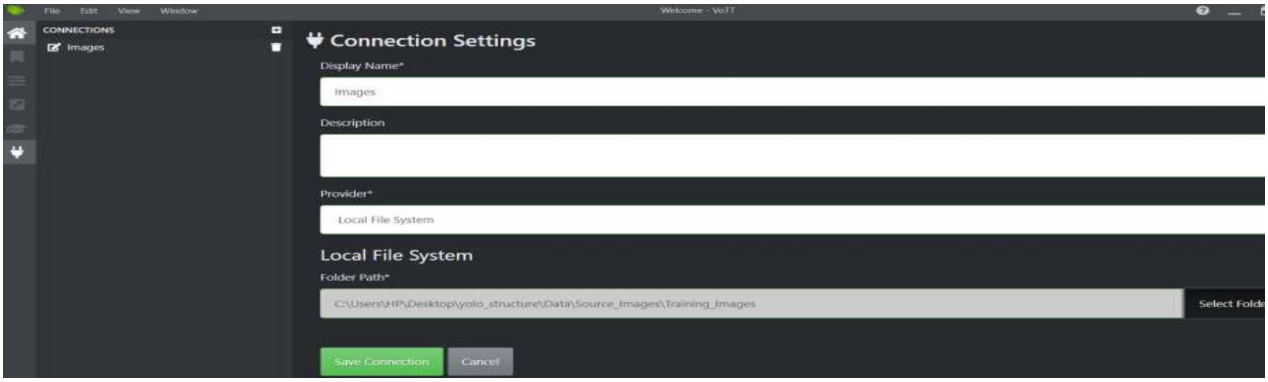

Fig-13 : Connection Settings

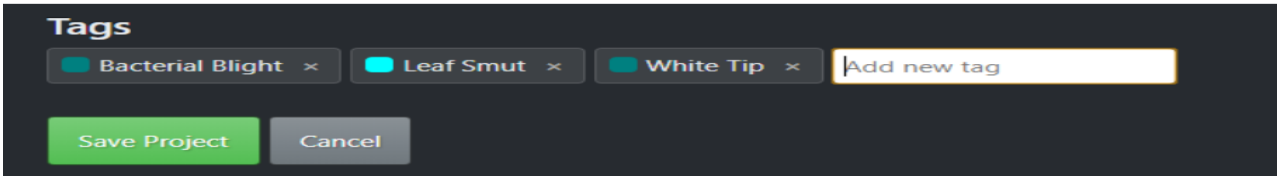

Fig-14 : Tags with which the images have to be labelled

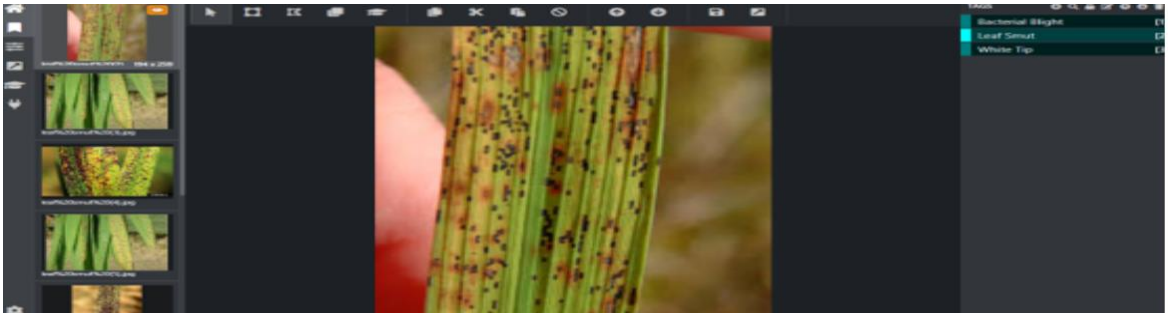

Fig-15 : Labeling the images with the tags

- Navigate to **Export Settings** in the sidebar and then change the **Provider** to **`Comma Separated Values (CSV)`,** then hit **Save Export Settings**.

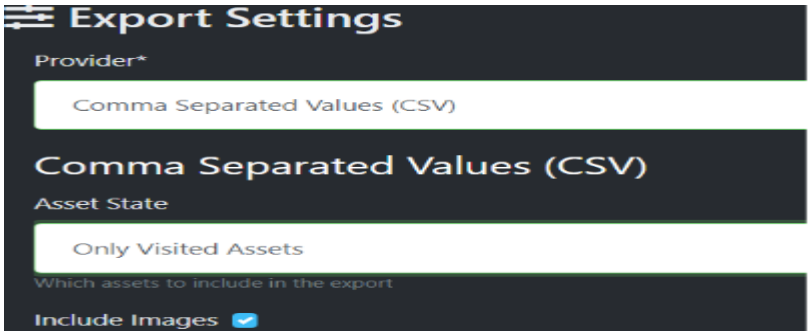

Fig-16 : Export Settings

- Once you have labeled enough images press **CRTL+E** to export the project. You should now see a folder called [`vott-csv-export`](/Data/Source_Images/Training_Images/vott-csv-export) in the [`Training_Images`](/Data/Source_Images/Training_Images) directory. Within that folder, you should see a `*.csv` file called [`Annotations-export.csv`](/Data/Source_Images/Training_Images/vott-csv-export/Annotations-export.csv) which contains file names and bounding box coordinates.

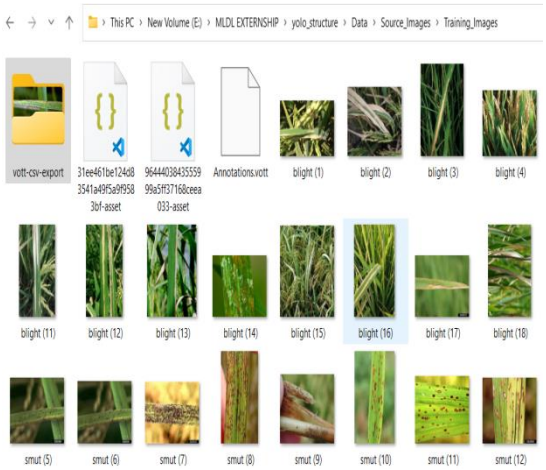

Fig-17 : vott-csv-export folder

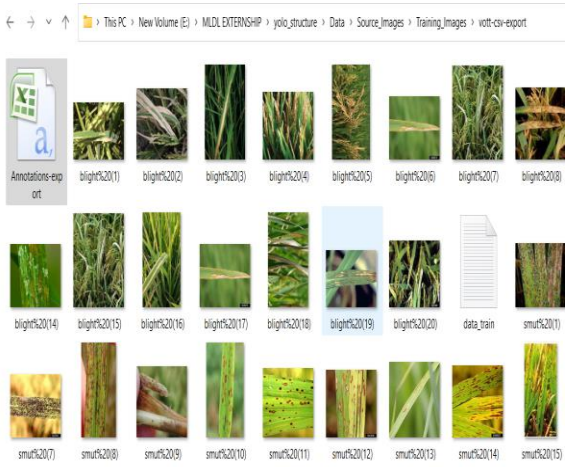

Fig-18 : Annotation-export.csv

- As a final step, convert the VoTT CSV format to the YOLOv3 format. To do so, run the conversion script from within the [`yolostructure/1_Image_Annotation`] folder.

- To run the file open the "**Anaconda prompt"** navigate to yolo_structure/1_Image_Annotation and run **Convert_to_YOLO_format.py**

```
(base) C:\Users\YASASWI ADAPA>e:

(base) E:\>cd MLDL EXTERNSHIP

(base) E:\MLDL EXTERNSHIP>cd yolo_structure

(base) E:\MLDL EXTERNSHIP\yolo_structure>cd 1_Image_Annotation

(base) E:\MLDL EXTERNSHIP\yolo_structure\1_Image_Annotation>python Convert_to_YOLO_format.py
```

Fig-19 : Executing Convert_to_YOLO_format.py

- The script generates two output files : [`data_train.txt`] (/Data/Source_Images / Training _ Images /vott-csv-export/data_train.txt) located in the [`yolostructure /Data /Source _Images /Training_Images/vott-csv-export`](/Data/Source_Images/Training_Images/vott-csv-export) folder and [`data_classes.txt`](/Data/Model_Weights/**data_classes.txt**) located in the [`yolostructure/Data/Model_Weights`](/Data/Model_Weights/) folder.

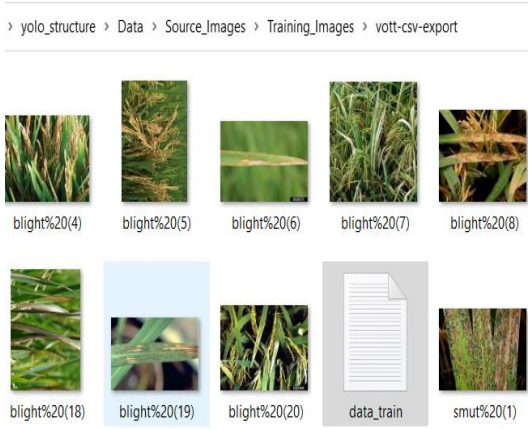

Fig-20 : data_train.txt

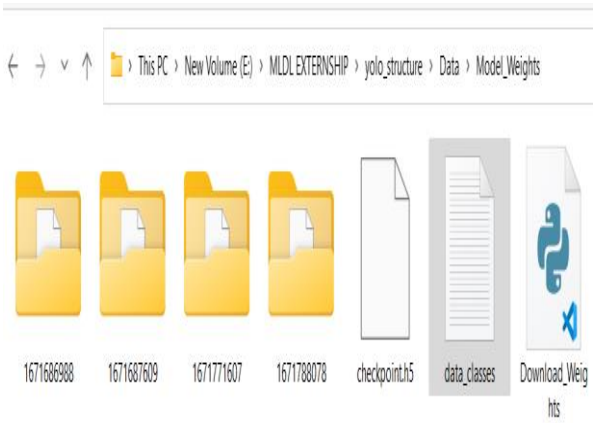

Fig-21 : data_classes.txt

(iv) Training YOLO :

(a) Download And Convert Pre-Trained Weights:

- Download the pre-trained YOLOv3 weights and convert them to the Keras format before training the model using the training images and the annotation file created in previous step.
- To run both steps run the download and conversion script from within the [`yolostructure / 2_Training`](/2_Training/) directory."**python Download_and_Convert_YOLO_weights .py"**

```
(base) E:\MLDL EXTERNSHIP\yolo_structure>cd 2_Training

(base) E:\MLDL EXTERNSHIP\yolo_structure\2_Training>python Download_and_Convert_YOLO_weights.py
```

Fig-22 : Executing Download_and_Convert_YOLO_weights.py

- This will download yolov3 weights and convert them into keras format. These files will be downloaded in[yolo_structure \2_Training\src\keras_yolo3].

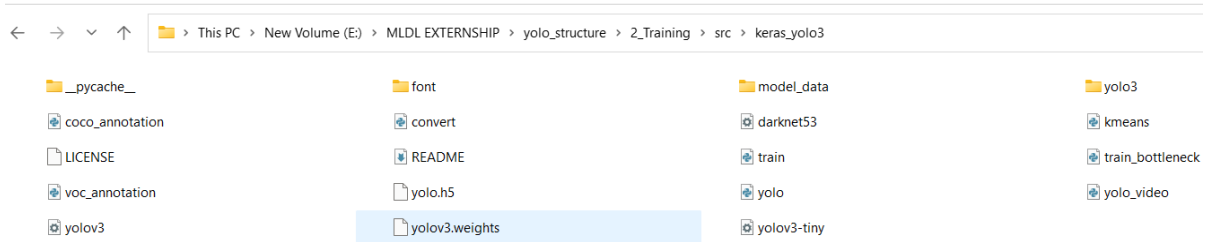

Fig-23 : keras_yolo3 folder

(b) Train YOLOv3 detector :

To start the training, run the training script from within the [`yolostructure /2_Training`](/2_Training/) directory : " **python Train_YOLO.py "**

```
(base) E:\MLDL EXTERNSHIP\yolo_structure>cd 2_Training

(base) E:\MLDL EXTERNSHIP\yolo_structure\2_Training>python Train_YOLO.py
```

Fig-24 : Executing Train_YOLO.py to train the model

- Depending on our set-up, this process can take a few minutes to a few hours. The final weights are saved in [`yolostructure/Data/Model_weights`](/Data/Model_weights).

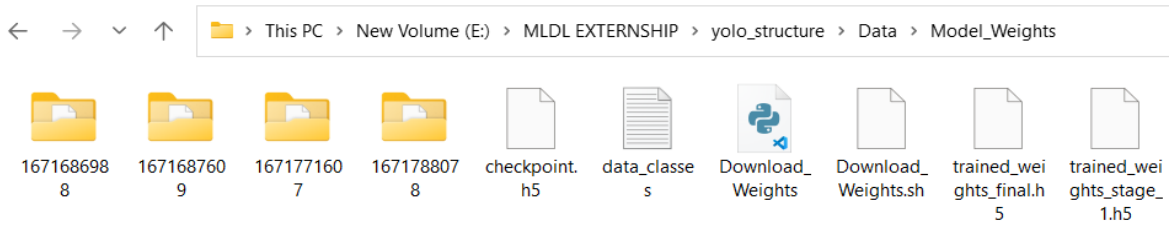

Fig-25 : Final trained weights in Model_Weights

(v) Testing The Model :

- In this step, we test our detector on infected crop images located in [`yolostructure/Data/Source_Images/Test_Images`](/Data/Source_Images/Test_Images).
- Place all the test images which can be downloaded from google and place them in [`yolostructure/Data/Source_Images/Test_Images`]

5.Flowchart

A flowchart is a picture of the separate steps of a process in sequential order.

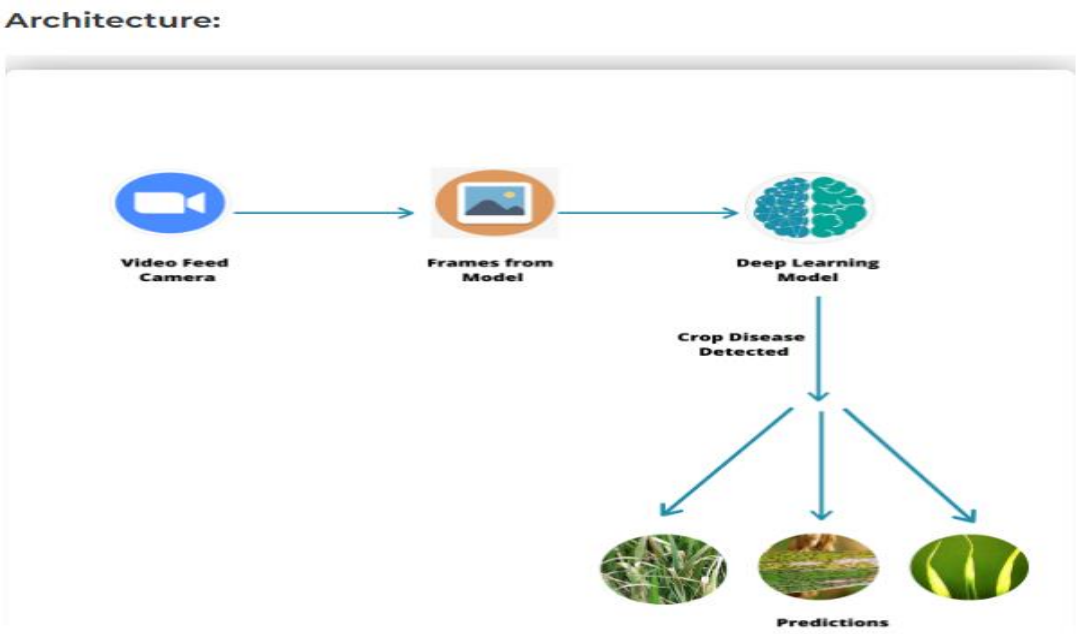

Fig-26 : Flowchart for Rice Crop Disease Detection using YOLO

6.Results

- To detect objects run the detector script from within the [`yolostructure/3_Inference`](/3_Inference/) directory: " **python Detector.py** "

```
(base) E:\MLDL EXTERNSHIP\yolo_structure>cd 3_Inference

(base) E:\MLDL EXTERNSHIP\yolo_structure\3_Inference>python Detector.py
```

Fig-27 : Executing Detector.py

Output:

The result of test images is stored in the Test_Image_Detection_Results folder.

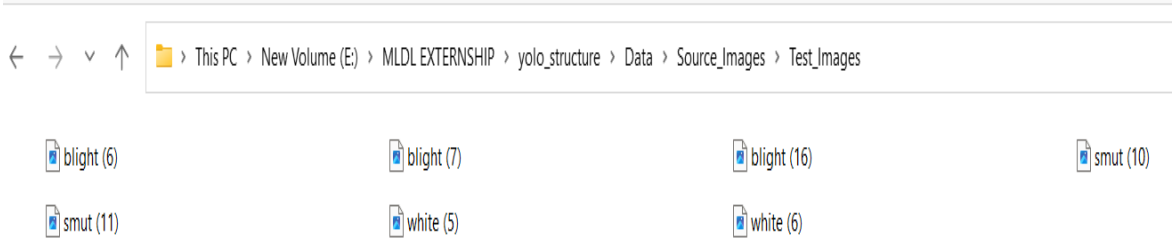

Fig-28 : Contents in Test_Images Folder

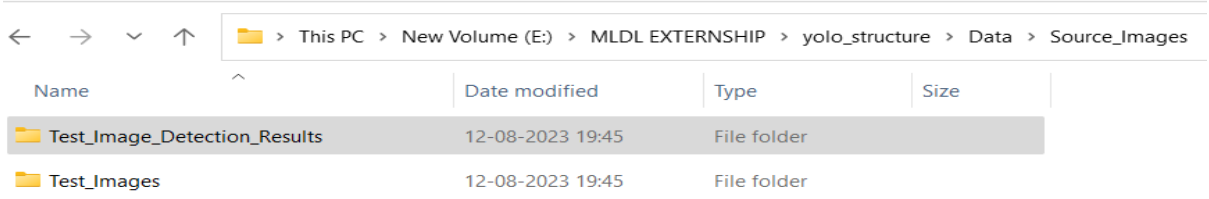

Fig-29 : Output Folder

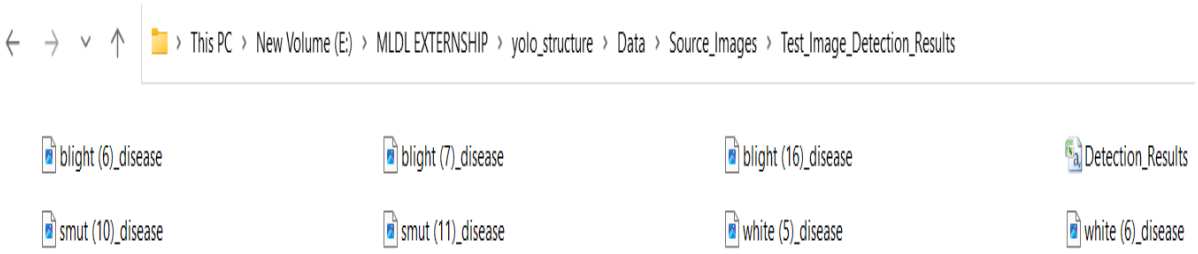

Fig-30 : Contents in Output Folder

7.Adavantages & Disadvantages

Advantages :

Real-time Detection : YOLO is known for its real-time processing capabilities, allowing for quick and efficient detection of rice crop diseases. This is crucial in agriculture, as early detection of diseases can prevent their spread and minimize crop losses.

Efficiency : YOLO's single-pass architecture processes the entire image in one go, making it faster compared to other object detection methods that use sliding windows. This efficiency is important in large-scale agricultural operations where timely detection can significantly impact yields.

Accuracy : YOLO has been shown to achieve competitive accuracy in object detection tasks. While accuracy can vary based on factors like dataset quality and model tuning, YOLO's detection performance can help identify diseases accurately, leading to better decision-making for farmers.

Multi-class Detection : YOLO is capable of detecting multiple classes (types) of diseases in a single image simultaneously. In rice crop disease detection, where there can be various disease types affecting the crop, this capability is valuable as it provides a comprehensive view of the health status of the field.

Integration with Other Technologies : YOLO can be integrated with Geographic Information Systems (GIS) and other remote sensing technologies, enabling farmers to have a holistic view of their fields and make informed decisions based on spatial data.

Research and Development : Implementing YOLO for rice crop disease detection contributes to ongoing research in the field of precision agriculture and computer vision. The insights gained from using such technologies can lead to advancements in disease management and agricultural practices.

Disadvantages :

While using the YOLO (You Only Look Once) project for rice crop disease detection offers several advantages, there are also some potential disadvantages to consider:

Limited Robustness to Variability: YOLO's performance can degrade when faced with significant variations in lighting, weather conditions, camera angles, and other environmental factors.

Model Overfitting: There's a risk of overfitting, where the model becomes too specialized in the training data and doesn't generalize well to new, unseen data. This can lead to poor performance when applied to real-world conditions.

Small and Rare Classes: If certain disease types or instances are rare in the training data, YOLO might struggle to accurately detect them. The model may not learn to recognize these rare classes effectively, leading to limited utility in real-world scenarios.

False Positives and Negatives: YOLO might generate false positives (detecting diseases that are not present) and false negatives (failing to detect actual diseases). These errors could lead to incorrect management decisions, potentially harming crop yields and economic outcomes.

To mitigate these disadvantages, careful planning, model validation, and ongoing monitoring are necessary. It's essential to consider the specific context of rice crop disease detection and evaluate whether YOLO is the best fit given the available resources and challenges.

8.Applications

The YOLO (You Only Look Once) project's object detection capabilities can be applied to rice crop disease detection in various ways. Some of the applications include:

Early Disease Detection : YOLO can be employed to monitor rice fields and detect diseases at an early stage. Early detection allows farmers to take timely actions, such as implementing targeted treatments or adjusting irrigation and fertilization strategies, to mitigate the spread of diseases and minimize crop losses.

Precision Pest Management : YOLO's real-time detection can aid in the precise application of pesticides and fungicides. Instead of blanket treatments, farmers can target specific areas with disease outbreaks, reducing the overall use of agrochemicals and minimizing environmental impact.

Automated Surveillance : Drones equipped with cameras and YOLO can fly over large rice fields to automatically identify disease hotspots. This automated surveillance saves time and resources compared to manual scouting, enabling efficient monitoring of extensive agricultural areas.

Quality Control in Seed Production : Before distributing rice seeds, they need to be free from diseases. YOLO can be used to inspect seed quality by identifying any disease symptoms, ensuring that healthy seeds are provided to farmers for planting.

Monitoring Disease Spread : YOLO can track the progression of disease outbreaks across a field or even multiple fields. This information helps in understanding the patterns of disease spread, which can be useful for regional disease management strategies.

Research and Disease Analysis : Researchers can use YOLO to analyze the prevalence and distribution of various rice diseases over time. This data can contribute to understanding disease dynamics and developing more effective disease-resistant rice varieties.

Integrated Pest Management : YOLO's data can be integrated with other sources, such as weather data and pest life cycles, to develop comprehensive integrated pest management strategies. This approach considers multiple factors to determine the best time for disease control interventions.

Customized Advisories : YOLO can be integrated into mobile apps or web platforms that provide real-time disease detection information to farmers. These apps can offer personalized recommendations for disease management based on the detected diseases and the specific field conditions.

Education and Extension Services : YOLO-based systems can be used for educational purposes, helping farmers and agricultural extension workers to identify diseases accurately. This can improve farmers' knowledge and awareness about rice diseases and their management.

Sustainable Agriculture : By enabling targeted disease management, YOLO contributes to more sustainable agricultural practices. Farmers can reduce the use of chemical inputs, conserve resources, and adopt environmentally friendly approaches to disease control.

Scientific Research and Development : The data collected through YOLO-based systems can be used by researchers to study disease trends, develop new detection methods, and enhance disease-resistant rice varieties.

These applications demonstrate the potential of using YOLO for rice crop disease detection to improve agricultural productivity, reduce losses, and promote sustainable farming practices.

9.Conclusion

In conclusion, the application of the YOLO (You Only Look Once) project for rice crop disease prediction holds significant promise in revolutionizing the way we manage agricultural health. By harnessing the power of real-time object detection, YOLO offers a range of advantages that can address critical challenges faced by rice farmers and contribute to sustainable agriculture.

The ability of YOLO to provide early disease detection, precise pest managementcan greatly enhance crop yield and quality. Its efficiency in detecting multiple disease types simultaneously empowers farmers with comprehensive insights into the health status of their fields. This, in turn, enables informed decision-making, leading to timely interventions that mitigate the spread of diseases and minimize economic losses.

While the advantages are evident, it's crucial to acknowledge the potential limitations. The quality and diversity of training data, adaptation to new diseases, and the need for expert fine-tuning underscore the importance of careful implementation.

Ultimately, the YOLO project's impact extends beyond the agricultural field. Its applications hold the potential to reshape how we approach food security, sustainable practices, and technological advancements in modern agriculture. By leveraging YOLO's capabilities for rice crop disease prediction, we embark on a journey towards healthier crops, increased agricultural productivity, and a more resilient and prosperous farming future.

10.Future Scope

In the coming years, the intersection of agriculture, computer vision, and AI will likely lead to transformative advancements in rice crop disease detection. By addressing challenges, embracing innovation, and collaborating across disciplines, the YOLO project has the potential to contribute significantly to the resilience and sustainability of global rice production.

Autonomous robotic systems equipped with YOLO could perform tasks like targeted spraying, precise sample collection, and even removal of infected plants, contributing to efficient disease control practices. Implementing YOLO on edge devices, such as drones, autonomous vehicles, and farm equipment, could enable real-time disease detection directly in the field, reducing the need for data transmission to centralized systems.

Future iterations of YOLO models could be designed to handle a wider range of environmental conditions, such as varying lighting, weather, and field terrains. This would make the technology more adaptable to real-world agricultural settings.

Collaborations between agronomists, plant pathologists, computer vision researchers, and data scientists will become increasingly important. These collaborations can lead to more accurate models, better integration with existing agricultural practices, and more effective disease management strategies.

YOLO-based rice disease detection can be extended to diverse geographical regions, allowing for localized adaptations to different rice varieties and disease prevalence. This scalability can contribute to addressing global food security challenges. The YOLO project can continue evolving through continuous learning and improvement cycles. As more data is collected and new insights are gained, models can be iteratively refined to better address the dynamic challenges of rice crop disease detection.

11.Bibiliography

1. International Research Journal of Modernization in Engineering Technology and Science (Peer-Reviewed, Open Access, Fully Refereed International Journal) Volume:04,Issue:03/March-2022 www.irjmets.com @International Research Journal of Modernization in Engineering, Technology and Science DETECTION OF RICE CROP DISEASE USING THE YOLO ALGORITHM R Srinivasan, Lakkireddy Bhanu Prakash Reddy, Dr. Syed Jahangir Badashah.
2. Muhammad Hammad Saleem et al, “Plant disease detection and classification by deep learning”, Plants, 31 October 2019.
3. Yang Lu et al, “Identification of rice diseases using deep convolutional neural network”, Elsevier, 6 February 2017.
4. Konstantinos P. Ferentinos, “Deep learning models for plant disease detection and diagnosis”, Elsevier, September 2017.
5. Sharada P. Mohanty, “Using Deep Learning For Image-Based Plant Disease Detection”, Frontiers in plant science, volume 7, September 2016.

Appendix

Source Code :

https://drive.google.com/file/d/1Lofw_kFUUTLuPSmC8i6AMMqMNLNqC4HL/view?usp=drive_link