

Machine Learning phase-2 Submission

Ayush Kumar Mishra (12240340)

Ayush Patel (12240340)

Divyanshu Prakash (12240540)

Shivam (12241710)

Ujjwal Raj (12241920)

October 16, 2024

Dashboard

For checking the working of the
model, Please go to

Website

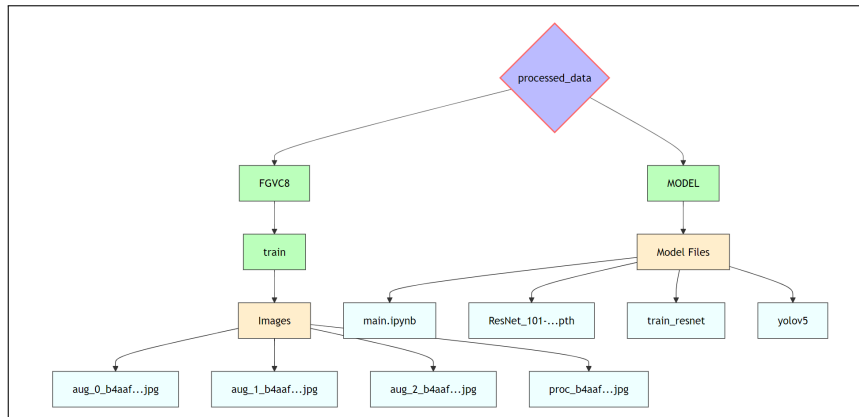
<https://krishi-ai.onrender.com/>

Github Reposotory [Github website](#)

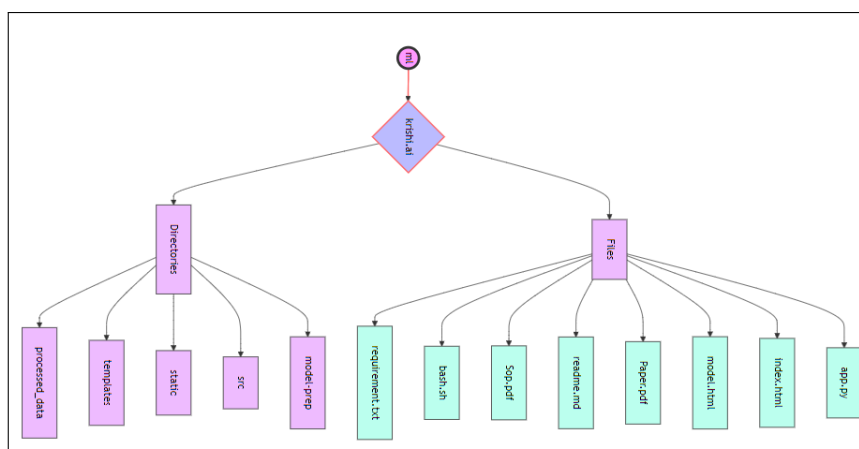
1 Project Directory

To begin with the project, we first created a directory structure to organize our work. The directory structure is as follows:

- **Data:** This directory contains the datasets used for training and validation. It includes subdirectories for each dataset, along with the necessary annotations and labels.



This is flowchart for model and data directory.



This is flowchart for flask server and website directory.

2 Project Workflow

The project workflow was divided into several phases, each focusing on a specific aspect of the project. The workflow included the following steps:

1. Data Collection:

Datasets

As stated in the paper, we used three different datasets for training and validation. These datasets contained images of plant leaves affected by various diseases.

2. Data Preprocessing and Augmentation: In this step, we performed the following operations:

a **Basic Preprocessing:**

- Converted images from BGR to RGB color space.
- Resized all images to 256x256 pixels for consistency.
- Applied Gaussian blur with a 3x3 kernel for noise reduction.
- Used Non-Local Means Denoising for further noise reduction.
- Adjusted image contrast and brightness.

b **Data Augmentation:**

We applied various weather-based augmentations with a probability of 0.6, including:

- Random sun flare effects.
- Simulated rainfall.
- Random shadow generation.
- Fog simulation.



As we can see in the image, the weather-based augmentations have been applied to the original image, simulating different weather conditions.

- **Normal Image:** The original image without any modifications.
- **Image with Shadow:** We added a random shadow effect for a dynamic look.
- **Image with Fog:** A layer of fog was introduced to create a misty atmosphere.
- **Image with Rain:** We applied a rain effect to simulate a rainy environment.

c **Dataset Split:**

Dataset Split

- **Training Set:** 80% of the data.
- **Validation Set:** 10% of the data.
- **Test Set:** 10% of the data.

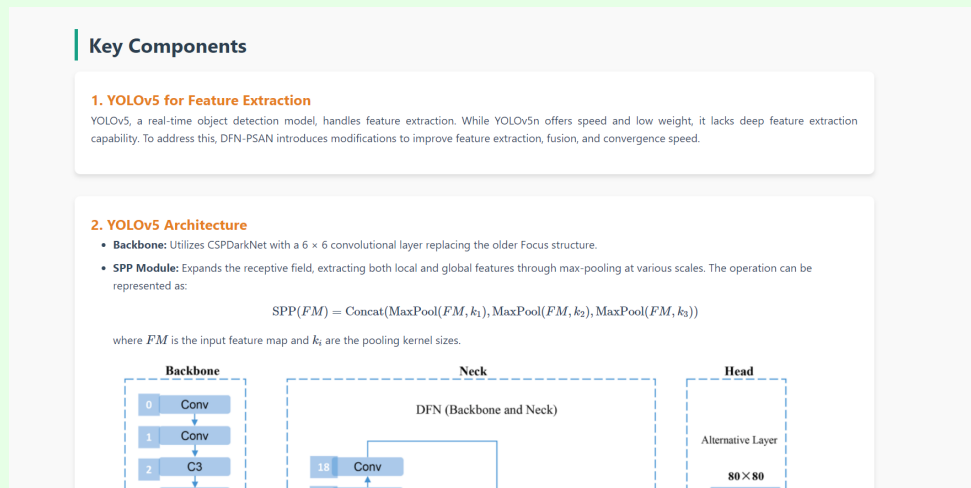
For this, I created a bash file to randomly split the dataset into training, validation, and test sets.

The script ensured that each set had an equal distribution of images across different classes, maintaining the original dataset's class balance.

d Website Development:

We developed a user-friendly website interface for users to upload images of plant leaves and receive disease classification results. The website included the following features:

- A home page with an intuitive interface.
- Model page for knowing about the structure and training of the model.



- Dataset page for knowing about the dataset used and to download it for personal use.
- An image upload form for users to submit leaf images.
- A results page displaying the predicted disease class.

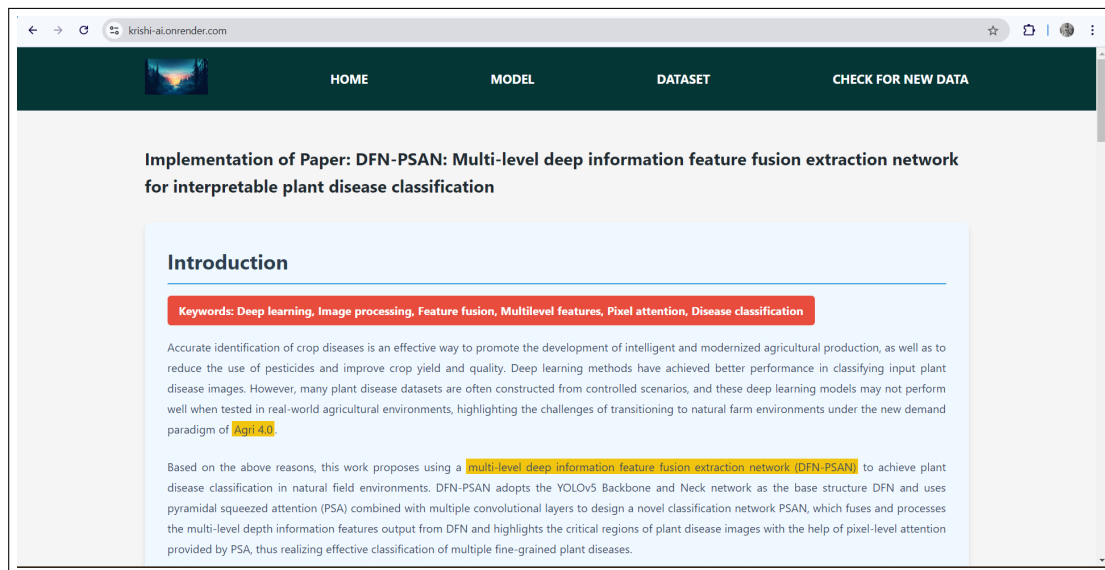
3. Achievements accomplished in this phase.

Website Part:

Home interface.

Home page of our website looks like this.

It has a simple and intuitive interface for users to navigate through different sections of the website.



Process of Loading Image by the user

Any user can upload an image using the following interface:

The screenshot shows the "Upload New Data" form on the same website. The header is identical. The form has a title "Upload New Data" and a label "Select an image or file:". Below the label is a file selection input field with a "Choose File" button and the text "No file chosen". Below the input field is a green "Upload File" button.

Form Interface to upload the image of a plant.

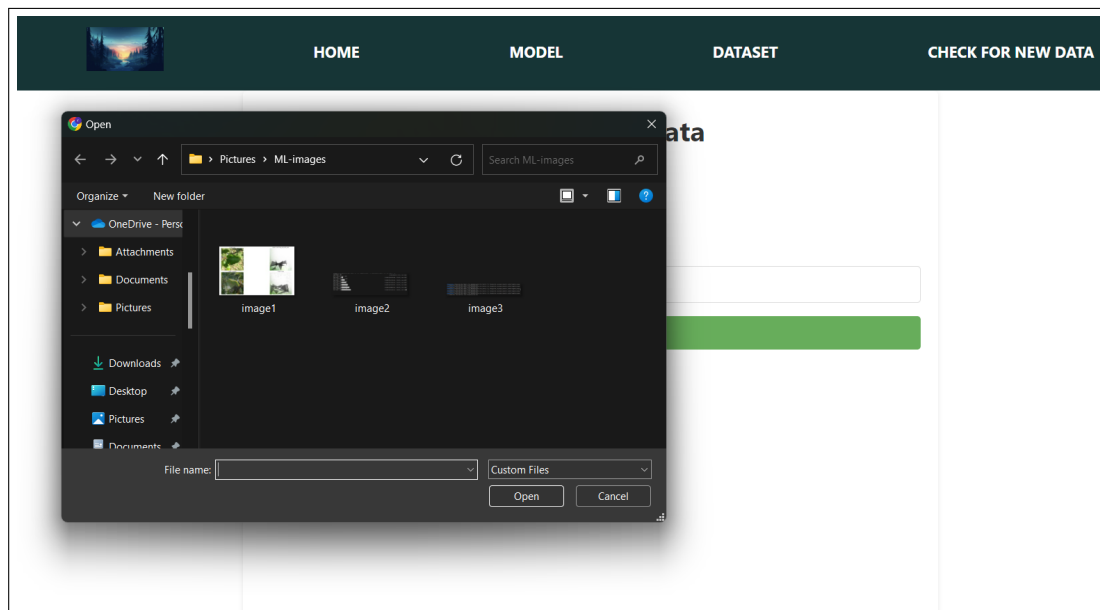
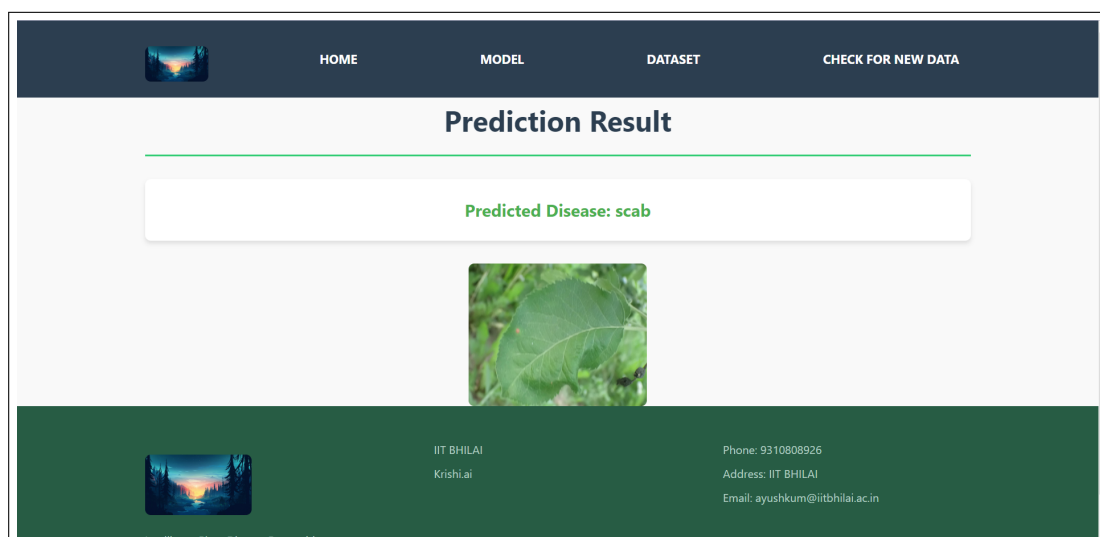


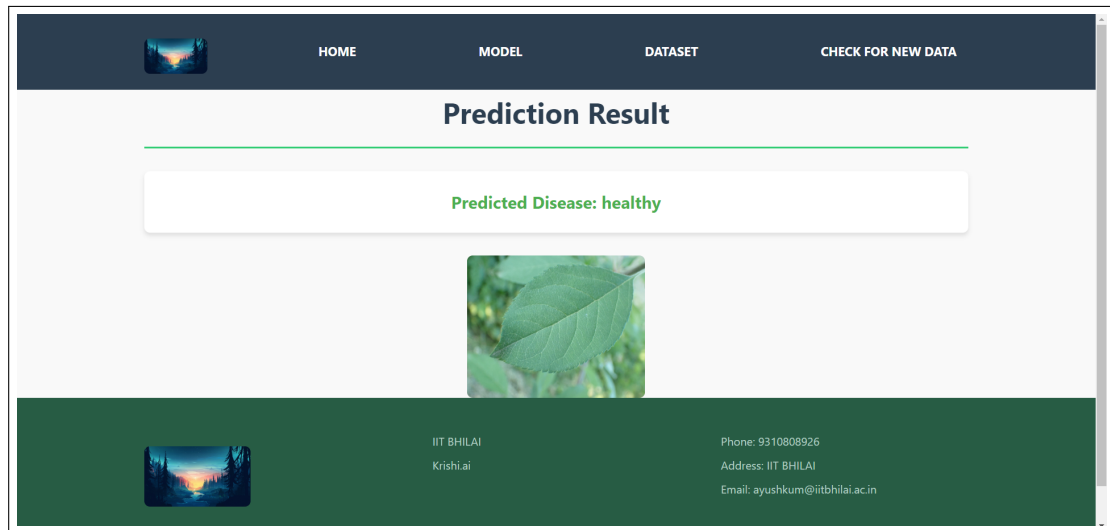
Image uploaded by the user.

Results Page

The results page displays the predicted disease class along with the confidence score.



The model predicted the disease as **Scab**.



The confidence score for this prediction is **healthy**.

4. Tasks to be Done

Hyperparameter Tuning

Objective: *Hyperparameter tuning is a critical step to enhance the model's performance. We will explore different combinations of learning rates, batch sizes, and optimization techniques to find the most effective configuration.*

Steps:

- Explore different hyperparameter combinations using **grid search** and **random search**.
- Apply cross-validation to ensure consistent model performance across configurations.

PSAN Parameter Complexity

Objective: *The current configuration of the PSAN (Pyramid Split Attention Network) parameters is overly simplistic. Our goal is to introduce more complexity to better capture high-level features, improving model robustness and accuracy.*

Steps:

- Experiment with deeper layers and refine attention mechanisms.
- Adjust split strategies and attention modules to optimize feature extraction.

Model Expansion to Other Datasets

Objective: *The current model has been trained on only one dataset. We aim to extend the training process to the other two available datasets to ensure generalization across diverse data sources.*

Steps:

- Preprocess and augment the additional datasets.
- Train the model on new datasets and validate it across all datasets to ensure consistent and accurate results.

Generalization for Future Datasets

Objective: *To ensure the model can handle new datasets in the future, improving flexibility and adaptability to various data types.*

Steps:

- Design a flexible model architecture that can generalize with minimal retraining.
- Utilize **transfer learning** and fine-tuning strategies for incorporating new datasets efficiently.

5. Members Contribution

Ayush Patel (12240350)

- **Data Augmentation & Preprocessing:** Made data pipeline for performing preprocessing, including image resizing(256 * 256 pixels), applying **GaussianBlur**, **fastNIMeansDenoisingColored**, **convertScaleAbs** and augmentation using **RandomFog**, **RandomRain**, **RandomShadow**, **RandomSunFlare** to ensure high-quality data for training.
- **Data Management:** Managed dataset versioning (managed all the git related stuffs) and ensured smooth data handling for training and validation.
- **Code Optimization:** Optimized the DFN model's efficiency through mixed-precision training and minimizing redundant computations.

Ayush Kumar Mishra (12240340)

- **Model Architecture Implementation (DFN):** Played a key role in implementing the Deep Fusion Network (DFN) backbone, utilizing the YOLOv5 architecture and designing the multi-level feature fusion module.
- **Model Training & Evaluation:** Managed the model training process, including loss function optimization, early stopping, and model evaluation on validation data.
- **Splited the data** Made bash script to split the data into training, validation, and test set randomly acc to config stated in research paper.
- **Integrating model into flask:** Integrated the trained model into the Flask server, enabling seamless interaction between the website and the model.

Divyanshu Prakash (12240540)

- **Pyramid Squeezed Attention Module (PSA):** Contributed to integrating and fine-tuning the Pyramid Squeezed Attention Network (PSAN) module for better feature extraction.
- **Website Development:** Helped in developing the website interface, designing the homepage, and ensuring the image upload feature for disease classification.

Shivam (12241710)

- **Project Workflow & Coordination:** Led the overall project workflow, assigning tasks, managing deadlines, and ensuring that team members were aligned with their responsibilities.
- **Pyramid Squeezed Attention Module (PSA):** Worked alongside Divyanshu to fine-tune the attention layers in the PSAN module.
- **Helped in Data Augmentation:** Helped in implementing weather-based augmentations, including sun flare, rain, shadow, and fog effects.

Ujjwal Raj (12241920)

- **Website Development:** Spearheaded the development of the website interface, ensuring proper design and functionality for the disease classification system.

- **Deployment:** Integrated the trained model into the website and managed the deployment pipeline on the Render platform, ensuring user-friendly interaction.
- **Model Tuning:** Focused on hyperparameter tuning to achieve better accuracy by adjusting learning rates, batch sizes, and regularization techniques.
