CSE 344: Computer Vision Project

Plant Disease Detection and Their Treatment Recommendations

kshitij
kshitij22256@iiitd.ac.in
tarandeep
tarandeep22536@iiitd.ac.in
roshan
roshan22418@iiitd.ac.in
Himanshu Kumar
himanshu22215@iiitd.ac.in

Abstract

Plant diseases threaten global agriculture, leading to economic losses and food insecurity. This project develops an AI-powered plant disease detection system using Convolutional Neural Networks (CNNs) trained on images from 45 plant class disesases. The system identifies diseases and provides treatment recommendations via a Natural Language Processing (NLP)-based advisory module.

Deployed as a web application using Streamlit and also build a mobile application, it enables users to upload images, receive real-time predictions, and access AI-driven treatment suggestions. By integrating computer vision and NLP, this system enhances accessibility to precision agriculture, reducing reliance on manual inspections and improving disease management.

1. Problem Statement

Plant diseases pose a significant threat to global agriculture, leading to substantial economic losses and food security challenges. Early detection and accurate diagnosis of plant diseases are crucial for effective intervention and improved crop yield. Traditional methods of disease identification rely on expert knowledge and manual inspection, which can be time-consuming, error-prone, and inaccessible to many farmers.

This project aims to develop an AI-powered plant disease detection system that leverages Convolutional Neural Networks (CNNs) trained on a diverse dataset covering over 45 plant class diseases, including wheat, rice, sugarcane, apple, strawberry, corn, grapes, tomato, squash, and potato. The system will identify diseases from plant images and provide recommendations using a Natural Language Processing (NLP)-based advisory module.

1.1. Scope of the Problem

The system will accept images of plant leaves as input, which will be analyzed using deep learning models to classify the disease. The output will include:

- The identified disease, if present.
- Possible causes and symptoms.
- Recommended treatment options, including pesticide suggestions and preventive measures.
- Climate-based precautions for disease management.

1.2. Target Users

This application is designed for farmers, agricultural experts, and researchers who need an efficient and accurate tool for disease detection and treatment recommendations. The system will benefit users in remote locations where access to agronomists is limited.

1.3. User Interface

The application is deployed on a web-based platform, developed using Streamlit for rapid prototyping and also built as a mobile app. The interface will enable users to:

- Upload images of plant leaves via a user-friendly web interface and a mobile app.
- Receive real-time disease predictions based on AI analysis.
- Access AI-generated treatment suggestions tailored to the identified disease and environmental conditions.

By integrating computer vision with NLP-based advisory services, this system aims to empower farmers with accessible, data-driven insights for disease management and agricultural productivity enhancement.

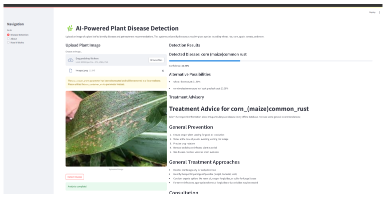


Figure 1. this is our web interface

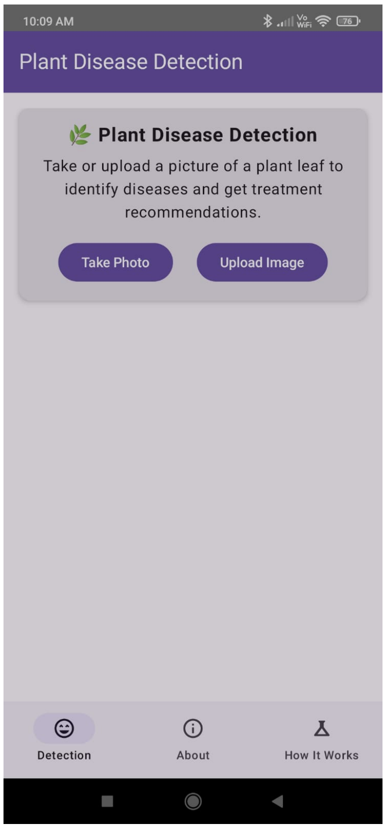


Figure 2. this is our App fronted

2. Related Work

Several research studies have explored plant disease detection using machine learning and deep learning techniques. In this section, we highlight notable approaches from the literature and outline baseline methods relevant to our study.

2.1. Machine Learning-Based Approaches

Initial approaches to automated plant disease detection utilized classical machine learning algorithms such as Support Vector Machines (SVM), Random Forests, and k-Nearest Neighbors (k-NN). These models typically relied on handcrafted feature extraction techniques, including color histograms, shape descriptors, and texture features. For example, SVMs have been successfully applied to distinguish among common leaf diseases using features derived from image preprocessing techniques [3]. However, these methods were often limited by their dependence on feature engineering and their inability to generalize across diverse plant species.

2.2. Deep Learning-Based Approaches

With the advent of deep learning, Convolutional Neural Networks (CNNs) have become the dominant approach in plant disease classification. Studies have shown the effectiveness of deep models like ResNet, AlexNet, and VGG in achieving high accuracy by learning hierarchical image features directly from raw data [1]. In particular, the work in [2] introduced an optimized CNN architecture that outperformed traditional classifiers on the PlantVillage dataset. Moreover, hybrid architectures combining CNNs with SVMs have been proposed to improve interpretability and performance, as discussed in [3].

2.3. Lightweight and Mobile-Friendly Models

Recent work has also focused on developing lightweight CNN models like MobileNetV2 to enable on-device disease detection. These models are optimized for deployment in real-world agricultural scenarios, especially in resource-constrained environments. The study in [4] demonstrated the viability of using such architectures for real-time plant disease detection with minimal computational overhead, maintaining competitive classification performance.

2.4. Baseline Methods and Evaluation Criteria

Based on the insights from the above studies, we identify the following baseline methods for comparison:

- ResNet-50 trained on the PlantVillage dataset for high-accuracy disease classification [1].
- MobileNetV2 for efficient, real-time disease detection on mobile devices [4].
- A hybrid CNN-SVM model that combines deep feature extraction with classical classification methods [3].

These baseline methods serve as critical reference points to evaluate the performance of our proposed system in terms of classification accuracy, efficiency, and usability.

3. Datasets & Evaluation Metrics

3.1. Datasets

For our experiments, we exclusively use two publicly available datasets from Kaggle related to plant disease detection. These datasets contain labeled images of healthy and diseased plant leaves across multiple species.

3.1.1 New Plant Diseases Dataset (Augmented)

The **New Plant Diseases Dataset (Augmented)** is a large dataset containing images of 14 different plant species and their corresponding disease classifications. The dataset has been augmented with variations in lighting, rotation, and zoom to improve model generalization. The dataset is publicly available on Kaggle [5].

3.1.2 Five Crop Disease Dataset

The **Five Crop Disease Dataset** consists of labeled images for five different plant species affected by various diseases. This dataset helps in evaluating model performance across diverse crop types. The dataset is publicly available on Kaggle [6].

Apart from this dataset, we also augmented the dataset for the robustness the model. After merging all the file and we get total images as following

- **Training Set:** 58,026 images
- **Validation Set:** 12,434 images
- **Test Set:** 12,435 images

3.2. Evaluation Metrics

To assess the performance of our models, we use standard classification evaluation metrics:

- **Accuracy**: Measures the percentage of correctly classified images.
- **Precision, Recall, and F1-Score**: Evaluates class-wise performance, especially for imbalanced datasets.
- **Confusion Matrix**: Provides insight into misclassified images and overall model performance.

4. NLP-based Advisory System

The NLP-based advisory system provides plant disease treatment recommendations using advanced language models (LLMs). The system integrates multiple LLM options and offers a fallback system to ensure continuous service. It is embedded into both web and mobile applications for ease of use.

4.1. LLM Configuration Description

The `LLM_CONFIG` dictionary defines the available configurations for the language models used to generate plant disease treatment advice. The following configurations are available:

- **Default Configuration:** The `"default"` configuration uses Meta's Llama-2-7b-chat-hf model, which is loaded locally with 8-bit quantization to optimize memory usage while maintaining good performance. This configuration is balanced, providing detailed agricultural advice with a temperature setting of 0.7 to introduce some creative variation in the responses.
- **Lightweight Configuration:** The `"lightweight"` configuration utilizes Microsoft's Phi-2 model, which has only 2.7 billion parameters, making it much smaller than Llama-2. Despite its size, it still offers high-quality agricultural advice, making it ideal for systems with limited computational resources or when faster inference is required.
- **API Configuration:** The `"api"` configuration connects to OpenAI's GPT-3.5-Turbo via their REST API, which requires an API key stored in environment variables. This configuration offers high-quality responses but requires internet connectivity and incurs usage costs.

5. Analysis of Results

5.1. Performance Analysis

Module-wise performance is evaluated using metrics such as runtime, memory usage, network bandwidth, and accuracy for visual recognition tasks.

Model	Runtime (s)	Memory Usage (MB)	Bandwidth (Mbps)
Custom CNN (Trained from Scratch)	160.4	460.6	2.84
ResNet-50	164.96	483.25	2.93
Support Vector Machine (SVM)	61.72	81.55	2.23
Random Forest	216.41	1034.38	0.64

Table 1. Performance analysis of different models (excluding accuracy)

5.2. Performance Metrics

Below is a performance matrix of individual models. The table below summarises the results.

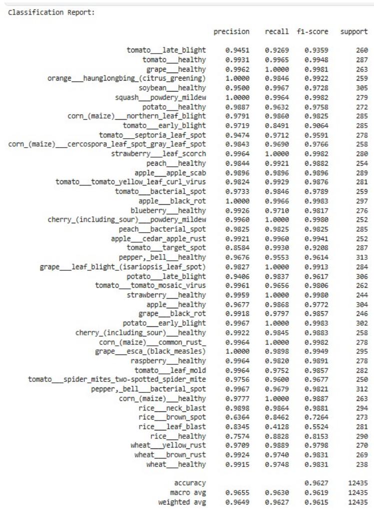
Module	Accuracy (%)	Precision (Macro) (%)	Recall (Macro) (%)	F1-Score (Macro) (%)
Random Forest	86.00	86.00	86.00	86.00
SVM	93.00	93.00	93.00	93.00
ResNet-50	94.73	95.17	94.77	94.77
CNN	96.27	96.53	96.30	96.19

Table 2. Performance comparison of different classification models on plant disease detection.

5.3. Observations and Insights

From the performance metrics, we observe the following:

- **Accuracy:** cnn achieved the highest accuracy, while random forest had the lowest.
- **Runtime:** SVM had the shortest runtime, making it the most efficient execution speed.



Classification Report:

	precision	recall	f1-score	support
tomato___late_blight	0.9451	0.9269	0.9359	260
tomato___healthy	0.9931	0.9965	0.9948	287
grape___healthy	0.9962	1.0000	0.9981	263
orange___haunglongbing_(citrus_greening)	1.0000	0.9846	0.9922	259
soybean___healthy	0.9500	0.9967	0.9728	305
squash___powdery_mildew	1.0000	0.9964	0.9982	279
potato___healthy	0.9887	0.9632	0.9758	272
corn_(maize)___northern_leaf_blight	0.9791	0.9860	0.9825	285
tomato___early_blight	0.9719	0.8491	0.9064	285
tomato___septoria_leaf_spot	0.9474	0.9712	0.9591	278
corn_(maize)___cercospora_leaf_spot_gray_leaf_spot	0.9843	0.9690	0.9766	258
strawberry___leaf_scorch	0.9964	1.0000	0.9982	280
peach___healthy	0.9844	0.9921	0.9882	254
apple___apple_scab	0.9896	0.9896	0.9896	289
tomato___tomato_yellow_leaf_curl_virus	0.9824	0.9929	0.9876	281
tomato___bacterial_spot	0.9733	0.9846	0.9789	259
apple___black_rot	1.0000	0.9966	0.9983	297
blueberry___healthy	0.9926	0.9710	0.9817	276
cherry_(including_sour)___powdery_mildew	0.9960	1.0000	0.9980	252
peach___bacterial_spot	0.9825	0.9825	0.9825	285
apple___cedar_apple_rust	0.9921	0.9960	0.9941	252
tomato___target_spot	0.8584	0.9930	0.9208	287
pepper,_bell___healthy	0.9676	0.9553	0.9614	313
grape___leaf_blight_(isariopsis_leaf_spot)	0.9827	1.0000	0.9913	284
potato___late_blight	0.9406	0.9837	0.9617	306
tomato___tomato_mosaic_virus	0.9961	0.9656	0.9806	262
strawberry___healthy	0.9959	1.0000	0.9980	244
apple___healthy	0.9677	0.9868	0.9772	304
grape___black_rot	0.9918	0.9797	0.9857	246
potato___early_blight	0.9967	1.0000	0.9983	302
cherry_(including_sour)___healthy	0.9922	0.9845	0.9883	258
corn_(maize)___common_rust_	0.9964	1.0000	0.9982	278
grape___esca_(black_measles)	1.0000	0.9898	0.9949	295
raspberry___healthy	0.9964	0.9820	0.9891	278
tomato___leaf_mold	0.9964	0.9752	0.9857	282
tomato___spider_mites_two-spotted_spider_mite	0.9756	0.9600	0.9677	250
pepper,_bell___bacterial_spot	0.9967	0.9679	0.9821	312
corn_(maize)___healthy	0.9777	1.0000	0.9887	263
rice___neck_blast	0.9898	0.9864	0.9881	294
rice___brown_spot	0.6364	0.8462	0.7264	273
rice___leaf_blast	0.8345	0.4128	0.5524	281
rice___healthy	0.7574	0.8828	0.8153	290
wheat___yellow_rust	0.9709	0.9889	0.9798	270
wheat___brown_rust	0.9924	0.9740	0.9831	269
wheat___healthy	0.9915	0.9748	0.9831	238
accuracy			0.9627	12435
macro avg	0.9655	0.9630	0.9619	12435
weighted avg	0.9649	0.9627	0.9615	12435

Figure 3. Confusion Matrix of the Best Performing Model

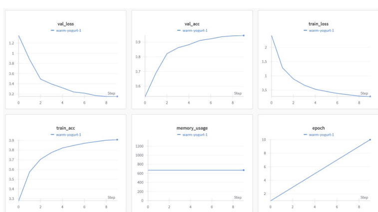


Figure 4. Graph of Val, train loss and accuracy CNN Model

- **Memory Usage:** Random forest consumed the most memory, due to its complex processing and multiple decision trees.

- **Network Bandwidth:** The highest bandwidth usage was recorded for Resnet50, indicating a higher data transfer requirement.

Further optimizations can be applied to balance accuracy and resource consumption.

6. Compute Requirements

The dataset requires significant computational resources for training deep learning models:

- **GPU Memory**: for the project, we used 16GB (GPU p100), which is required for efficient training, and we use kaggle resources for the training.

- **RAM**: 16GB is utilized when we do training

- **Storage**: The datasets require 10-15GB of disk space.

- **Energy Consumption**: Training deep models over multiple epochs leads to high power consumption, due to large datasets.

By optimizing batch sizes and utilizing pre-trained models, we aim to balance computational efficiency with model accuracy.

7. Individual Tasks

Member	Assigned Task(s)
Kshitij	NLP task, DataLoader, Mobile web
Tarandeep	Preprocessing, Model Robustness, Web App
Himanshu	Web Application, Mobile App, Model Training
Roshan	Training CNN, ResNet, SVM, Random Forest, web app

Table 3. Individual Task Assignments

8. About My Previous Project

My previous project was also focused on plant disease detection, but it was built using a basic approach and simpler machine learning techniques. At that time, we used Statistical Machine Learning (SML) methods such as random forest, along with a basic convolutional neural network (CNN) architecture. The model was trained on a limited number of classes, which restricted its ability to generalise across various plant diseases, and this project was done by Roshan and Tarandeep, not by other members.

However, the current project—Plant Disease Detection and Treatment Recommendation—is significantly more advanced and comprehensive. Below are the major improvements:

- ✓**Expanded Scope:** We now classify 45 different plant disease classes, making the model much more robust and applicable to real-world agricultural scenarios.

- ✓**Advanced CNN Architecture:** We have integrated deep CNN architectures, including ResNet50, to extract more complex features from leaf images.

- ✓**Treatment Recommendation:** Unlike the previous version, this system also provides treatment suggestions based on the detected disease.

- ✓**User Interface Development:** A fully functional web and mobile application has been developed to enable easy interaction. Users can upload leaf images, receive disease identification, location of the disease on the leaf, and treatment guidance.

- ✓**Regularization Techniques:** We implemented dropout layers to reduce overfitting and improve model generalization.

In summary, while my earlier work served as a foundational prototype, the current project is a complete end-to-end solution, integrating advanced computer vision techniques, deep learning, and user-friendly interfaces, making it fundamentally different and more impactful.

9. Future Work

Currently, we have implemented a CNN, Resnet50, Random Forest, and SVM models that predict the disease affecting plants. Our future work consists of the following steps:

- **Detection Model and NLP Integration:** We have already trained a model that localizes the disease on the leaf for focused treatment. The next step is integrating this model into the mobile and web applications, allowing users to upload images and visualise the affected regions.
- **Mobile App Enhancement:** The mobile app will be updated to enable users to take photos directly through the app. This will automatically detect the disease and provide recommendations for treatment.
- **Robust Data Generation:** We aim to develop a robust data generation model that works under varying conditions, including low light, different angles, and environmental factors, ensuring accurate disease detection in diverse real-world scenarios.
- **Clinical Expert Validation:** We will incorporate a clinical expert interface into both the web and mobile applications. The expert system will handle farmer queries, validate model predictions, and provide guidance, especially when there is a risk of severe disease spread or plant destruction.
- **NLP Fine-Tuning:** To improve the accuracy and relevance of the information provided, we aim to enhance our NLP model using techniques such as Retrieval-Augmented Generation (RAG) and Agentic Chunking.
- **GPT API Integration:** We plan to integrate a GPT-based API to predict diseases more accurately, offering reliable results for farmers. This will also serve to validate the models and refine their predictions.

References

[1] Fasila P. and Asha Gowda Karegowda, "Plant disease detection using deep learning: A review and comparative analysis", *Journal of Big Data*, vol. 10, no. 1, pp. 1–21, 2023. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00863-9

[2] Ali Hussain, Aqleem Abbas et al., "A comprehensive survey on plant disease detection using deep learning", *IEEE Access*, vol. 9, pp. 56606–56628, 2021. Available: https://ieeexplore.ieee.org/document/9399342

[3] A. S. M. M. Rahman et al., "An Overview of the Research on Plant Leaves Disease Detection using Image Processing Techniques", *ResearchGate*, 2017. Available: https://www.researchgate.net/publication/314436486

[4] Yassine Belhaj et al., "Lightweight CNN architectures for mobile-based plant disease detection", *Scientific Reports*, vol. 13, pp. 1–10, 2023. Available: https://www.nature.com/articles/s41598-023-34549-2

[5] Vipul Sharma, "New Plant Diseases Dataset (Augmented)", Kaggle, 2020. Available: https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset

[6] Shubham Pawar, "Five Crop Diseases Dataset", Kaggle, 2022. Available: https://www.kaggle.com/datasets/shubham2703/five-crop-diseases-dataset

Model Output

For the web and app application, we work on model prediction. For the web application, the image is uploaded from the system, and the disease recommendation is generated. In the mobile app, the photo is taken directly and the output is generated instantly. The disease spot identification is integrated into both platforms.Below is web application working ob corn rust and their recommendation of disease detection.

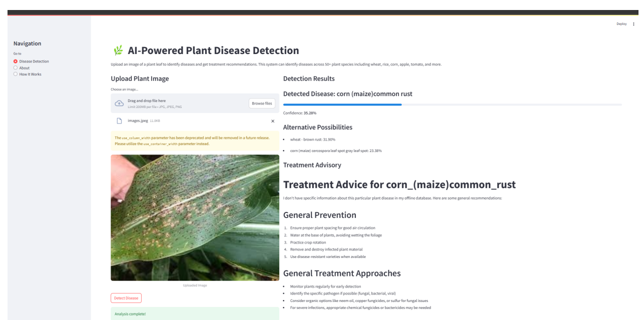


Figure 5. Corn rust detected using the web app

Disease Spot Identification

The disease spot is identified accurately. The integration helps in highlighting the affected area on the plant, and this is something to integrate into the model in future work.

EDA and Model Graphs

We analyse the dataset and understand a balanced or unbalanced dataset and we balance it. And also analyse the how much dataset of each class.

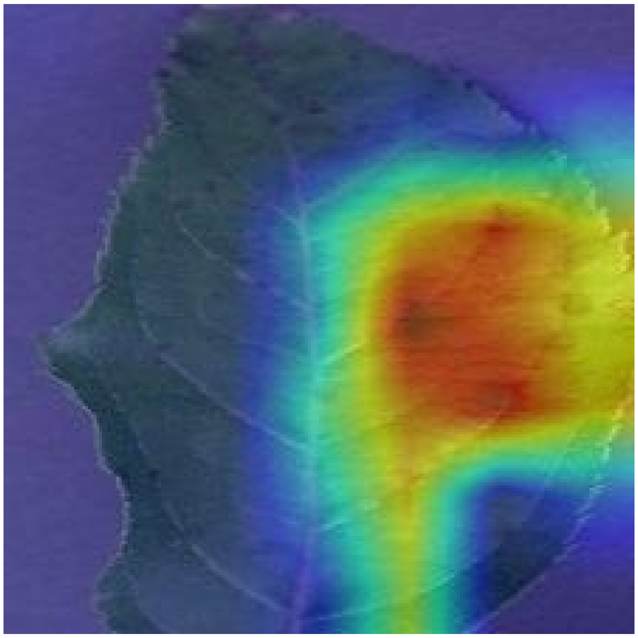


Figure 6. Disease detection spot

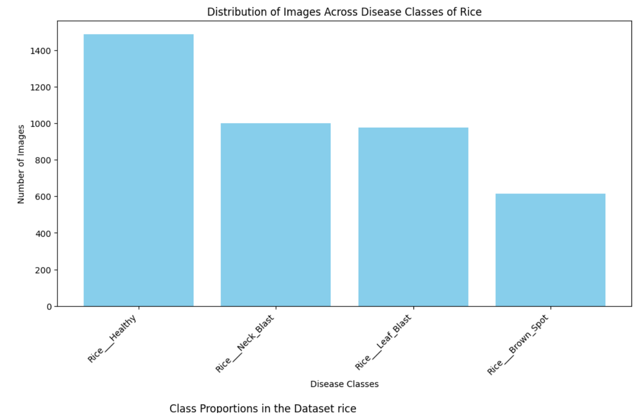


Figure 8. Rice plant dataset distribution

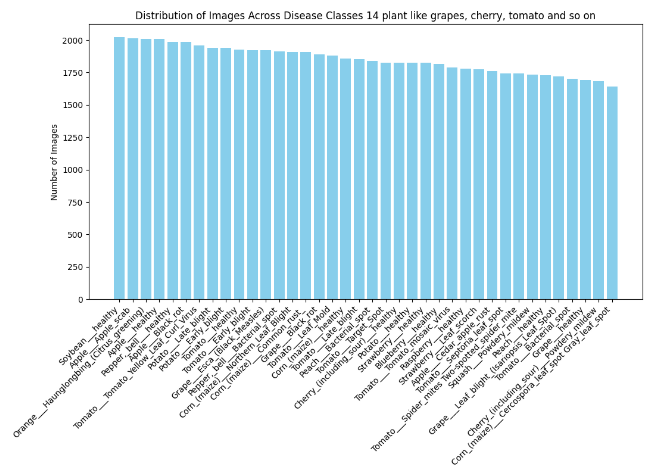


Figure 9. Distribution of 14 plant types and their diseases

Figure 7. Disease spot identified on the plant

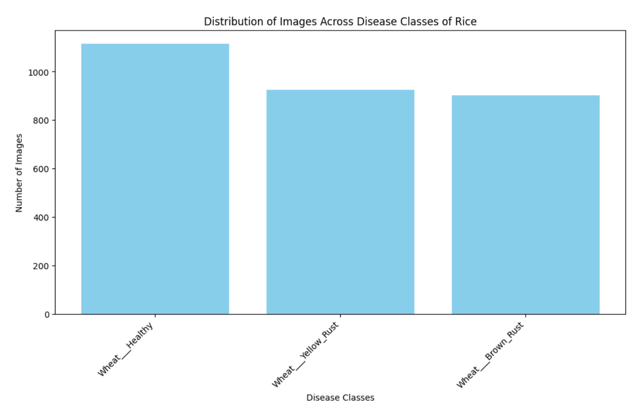


Figure 10. Wheat plant dataset distribution

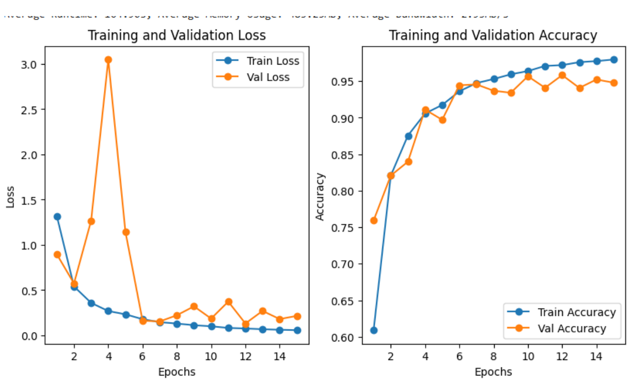


Figure 11. val and train loss and accuracy of Resnet model