Project Proposal

1. Title Page

- **Project Title**: Image Classification for Plant Disease Detection Using Deep Learning
- **Project Manager/Lead**: Shahriar Ratul
- **Organization**: N/A
- **Date**: 10th April, 2025

---

2. Executive Summary

The goal of this research is to create an image classification system based on deep learning that can identify plant illnesses from photos of leaves. The system classifies photos into predetermined disease categories using a Convolutional Neural Network (CNN) that has been implemented in PyTorch. Agricultural output can be greatly increased by addressing the requirement for automated and precise plant disease detection. High classification accuracy, precision, and recall are important objectives, as are visual insights offered by performance metrics like ROC curves and confusion matrices.

---

3. Project Background

- **Context and Rationale**: Global food security is seriously threatened by plant diseases. Farmers can reduce losses by taking prompt action with the aid of early and accurate detection. Conventional techniques rely on labour - intensive and error-prone manual inspection. This project offers a scalable and effective solution by automating the process using deep learning.
- **Problem Statement**: Plant disease identification by hand is ineffective and frequently incorrect. An automated system that can reliably identify plant diseases from photos is required.
- **Previous Work or Research**: CNNs have proven to be successful in picture classification jobs in the past, including applications in agriculture and medical diagnosis. Building on previous developments, this study focusses on the identification of plant diseases.

---

4. Project Objectives

- Objective 1: Create a CNN model that can identify plant illnesses from photos of leaves with at least 90% accuracy.
- Objective 2: Use metrics like precision, recall, F1-score, and AUC-ROC to assess the model.
- Objective 3: Provide visual aids for interpreting model performance, such as ROC curves and confusion matrices.

5. Scope of the Project

- **Inclusions**:
    1. Preprocessing and augmenting data.
    2. Validation and training of models.
    3. Implementation of an essential prediction function for evaluation.
    4. Basic prediction based on the built model.
    5. Visualization tools integration (confusion matrices, ROC-AUC curve).
    6. Training workflow implementation.
- **Exclusions**:
    1. Deployment in real time on edge or mobile devices.
    2. integration with the Internet of Things technologies or agricultural drones.
    3. Cross-platform application development.
    4. Cloud service deployment
    5. Multi-model data fusion (image data + sensor data).
- **Boundaries:**
    1. Restricted to the given dataset (e.g., certain plant diseases)
    2. Computational limitations are due to the available GPU/CPU hardware.
    3. Maximum batch size fixed by GPU memory.
    4. Training time is different based on machine/pc specs.
    5. Computational resources limit training time.

---

6. Design Methodology/Approach

- **Overview of Approach**: Data loading, preprocessing, model training, assessment, and visualization are all steps in a typical deep learning pipeline.
- **Project Phases**:
  - Phase 1: Plan all the tools, required library, GPU configuration, dataset collection etc.
  - Phase 2: Data preparation, preprocessing and exploration.
  - Phase 3: Model development and training
  - Phase 4: Model evaluation and performance visualization
  - Phase 5: Plan to build a UI (optional).
- **Tools and Resources**: 1. **PyTorch** for creating model. 2. To calculate matrices, **Scikit- learn**. 3. **Seaborn** and **Marplotlib** for visualizations 4. **GPU** for training speed. 5. **CudaToolKit12.6** for GPU use.

---

7. Testing Methodology

- **How Success Will Be Measured**:
  - F1-score, recall, accuracy, and precision of the model.
  - ROC curves and confusion matrices are visually examined.
  - Comparison with performance measures for the baseline.
- **Monitoring and Reporting**:
  1. During training, progress is monitored using loss/accuracy metrics.
  2. Final report with all metrics and visualisations included.

---

8. Project Timeline

- **Phase 1**: March 18 – March 20
- **Phase 2**: March 21- march 23
- **Phase 2**: March 24 – March 31
- **Phase 3**: April 7 – April 11
- **Phase 5**: [optional]
- **Final Deliverable**: Same as Project submission deadline

---

9. Budget

- **Personnel**: N/A
- **Materials/Equipment**: Cost for GPU/cloud computing resources: 0tk
- **Software/Tools**: Open-source tools: 0tk
- **Miscellaneous**: N/A
- **Total Budget**: 0tk

---

10. Team Members and Roles

- **Project Manager**: [Shahriar] – Responsible for overseeing the project's execution and timeline.
- **Team Members**: [Soleman& Durjoy] – Responsible for data preprocessing and model training
- **Team Members**: [Shahriar& Nafis] – Responsible for performance evaluation and visualisation.
- **Team Members**: [All members] – Responsible for documentation and report generation etc.

---

11. Risks and Mitigation Strategies

- **Risk 1:** Hardware limitation
  - Mitigation: Utilize the batch size. Use cloud GPU if needed
- **Risk 2**: Dataset limitation
  - Mitigation Strategy: Augmenting the data (Flip, rotation) and synthetic data generation
- **Risk 3**: Model Overfitting and Underfitting
  - Mitigation Strategy: For overfitting, add dropout layers (50%) and L2 regularisation; implement early stopping; use data augmentation and for underfitting, increase model complexity; reduce regularisation; extend training epochs; optimise preprocessing

12. Conclusion and Limitations

Using PyTorch, this research effectively created a deep learning-based picture categorisation system for the identification of plant diseases. With the use of strategies including data augmentation, dropout regularisation, and meticulous hyperparameter tuning, the developed CNN model showed excellent performance in recognising disease patterns from leaf pictures, attaining high accuracy and robust generalisation. Important insights into model behaviour and class-specific performance were obtained from the thorough evaluation that included metrics such as confusion matrices, ROC analysis, and precision-recall curves.

Nonetheless, it is necessary to recognise certain limitations. The quality and diversity of training data have a significant impact on the model's efficacy, and performance may suffer when applied to plant species or disease variations that are not covered in the dataset. Furthermore, additional optimisation for different climatic conditions and picture collection scenarios would be necessary for real-world implementation. Transfer learning with larger structures, integration with field-use mobile applications, and extension to a wider variety of crops and illnesses could all be investigated in future research.

---

14. References

1. PyTorch Documentation: https://pytorch.org/docs/stable/index.html
2. Scikit-learn Metrics: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics
3. Deep Learning for Plant Disease Detection: https://www.sciencedirect.com/science/article/pii/S0168169918310320
4. CNN Optimization Techniques: https://arxiv.org/abs/1803.09820.