**Project Requirements:**

**Deep Learning for Plant Disease Detection**

* Understand the **significance of deep learning** in plant disease detection and how it compares to traditional image-processing techniques.
* Recognize the **challenges in plant pathology** related to image-based classification, including dataset variability, environmental conditions, and disease symptoms.

1. **Model Selection and Implementation:**
   * Develop a transformer-based model (Vision Transformer or Swin Transformer).
   * Alternatively, experiment with hybrid architectures combining CNNs and Transformers.
   * Evaluate if the developed model can outperform results reported in the literature.
   * **App Deployment:**
     + Deploy the trained model using **Streamlit** to create a web-based application for plant disease classification.
     + The app should allow users to upload leaf images and receive classification results in real-time.
2. **Dataset:**
   * Use the PlantVillage dataset or a similar publicly available dataset.
   * Perform dataset preprocessing, including resizing, normalization, and augmentation.
3. **Preprocessing Techniques:**
   * Implement techniques like image resizing, normalization, and data augmentation.
   * Document preprocessing choices and their impact on performance.
4. **Training and Fine-Tuning:**
   * Train the selected model with appropriate hyperparameters.
   * Fine-tune pre-trained models if applicable.
   * Optimize learning rates, batch sizes, and dropout rates.
5. **Performance Evaluation:**
   * Evaluate the model using metrics such as accuracy, precision, recall, F1-score, and confusion matrices.
   * Visualize performance through learning curves and attention maps.
6. **Analysis and Discussion:**
   * Analyze results, highlighting strengths, limitations, and areas for improvement.
   * **Compare performance with baseline models such as CNNs.**

In most peer-reviewed papers, you would see:

* + - A table summarizing **previous work with reference** (e.g., "CNN on PlantVillage – 98.2% accuracy", "ViT – 97.5%", etc.)
    - Then show how the new method (e.g., Swin Transformer) performs relative to those.
    - This helps establish whether the new model is **state-of-the-art, competitive, or just acceptable.**

1. **Report Submission:**
   * Submit a comprehensive project report detailing:
     + Introduction and background.
     + Problem statement and goals.
     + Methodology and model architecture.
     + Dataset description and preprocessing steps.
     + Results and performance evaluation.
     + Discussion, conclusions, and future work.
   * The Github link that contains your project (It should include a readme on how to run your codes, the dataset source, etc)
   * A video presentation summarizing your project. The video should highlight the problem addressed, your solution approach (show the streamlit application), key results, and future directions.

**Submission of the GitHub link, a video demo and the final report**

* **Complete Development Work**
  + Ensure the model is fully trained, validated, and optimized for plant disease classification.
* **Prepare the Final Report**
  + Include objectives, methodology, experimental results, key findings, challenges faced, and future directions.
* **Submit a GitHub Repository**
  + Upload all relevant code, documentation, and datasets (if permissible).
  + Ensure the repository is well-organized and includes a README file with setup instructions.
* **Create a Video Presentation**
  + Highlight the problem addressed, solution approach, key results, and future directions.
  + Use clear and engaging visuals to effectively communicate your work.
* **Submit Final Deliverables. The final report link is in week 1 or the assignment menu.**
  + Final report (PDF format).
  + GitHub repository link.
  + Video presentation link or file.

-----------------------------------------------------------------------------------------------------------

**For the GitHub README, it usually includes:**  
Project Overview:  
1–2 sentences about what the project does and why it matters.  
Model Summary:  
Briefly explain which models you used (e.g., CNN, ViT), how they were trained, and which performed best.  
Dataset:  
What dataset you used (e.g., PlantVillage), whether it was full or subset, and any preprocessing applied.  
How to Run the Project:  
Clear steps for how to run the Streamlit app locally. Include commands like pip install -r requirements.txt, streamlit run [app.pyLinks to an external site.](http://app.py/" \t "_blank), etc.  
Key Results:  
Include a short table or bullet list of your best model’s accuracy, precision, or F1-score.  
Credits & Acknowledgments:  
Shout out libraries (like timm, sklearn, etc.) and any collaborators if relevant.  
  
**For the video:**  
Summarize the project (what it is, what model(s) were used)  
Discuss key challenges and solutions you encountered  
Show the working Streamlit app — demo it in action with a few predictions  
Mention final accuracy or insights from evaluation  
  
Keep it around 3–5 minutes, and remember — this isn’t just a demo, it’s your story of building the project.