Standard Operating Procedure (SOP) for DFN-PSAN Project

Ayush Kumar Mishra, Ayush Patel, Divyanshu Praksah, Shivam, Ujjwal Raj

8 sep 2024

1. Project Overview

1.1 Purpose

This SOP outlines the procedures for developing, implementing, and maintaining the DFN-PSAN (Deep Fusion Network with Pyramid Squeeze Attention Network) for plant disease classification.

1.2 Scope

This SOP covers all stages of the machine learning project lifecycle, from data collection to model deployment and maintenance.

2. Data Management

2.1 Data Collection

- Identify and document all data sources (Katra-Twelve, BARI-Sunflower, FGVC8)
- Ensure proper permissions and licenses are obtained for all datasets
- Document the process of data acquisition, including any API calls or web scraping techniques

2.2 Data Preprocessing

- Implement image preprocessing steps:
  - Image read-in and format conversion
  - Adaptive scaling
  - Gaussian filtering
  - Non-local mean noise reduction
  - Brightness and contrast adjustment
- Implement weather data augmentation:

  - Solar illumination transformation
  - Raindrop transformation
  - Shadow transformation
  - Fog transformation
- Ensure all preprocessing steps are reproducible and documented

2.3 Data Splitting

- Split data into training (80%), validation (10%), and test (10%) sets
- Ensure stratification to maintain class distribution across splits
- Document the random seed used for reproducibility

2.4 Data Version Control

- Use a data version control system (e.g., DVC) to track changes in datasets
- Document all versions of the dataset used in experiments

3. Model Development

3.1 Environment Setup

- Use virtual environments for isolation (i am using conda)
- Document all dependencies in a requirements.txt file(req.txt)
- Specify exact versions of all libraries used(in req.txt)

3.2 Model Architecture

- Implement YOLOv5 backbone and neck (DFN)
- Implement Pyramidal Squeezed Attention (PSA) module
- Implement Classification network (PSAN)
- Implement Multi-level feature fusion mechanism
- Document the architecture with diagrams and explanations

  - **Image processing: Vs code and Python 3.10**
  - **Deep learning framework: Pytorch 1.13.1 + cu117**
  - **Image processing library: OpenCV**
  - **Hardware acceleration: GPU**

3.3 Training Procedure

- Set and document all hyperparameters (learning rate, batch size, etc.)
- Implement training loop with proper logging
- Use early stopping to prevent overfitting
- Implement checkpointing to save model states

3.4 Evaluation Metrics

- Implement and document all evaluation metrics:
  - Accuracy
  - F1-score
  - Confusion matrix
  - ROC-AUC (if applicable)

3.5 Experiment Tracking

- Use an experiment tracking tool (i will use Weights & Biases)
- Log all hyperparameters, metrics, and artifacts for each run

5. Deployment

5.1 Deployment Platform

- The project is deployed on Render at `https://krishi-ai.onrender.com/`
- Ensure all team members have access to the Render dashboard for monitoring and updates

5.2 Deployment Process

- Document the step-by-step process for deploying updates to the Render platform
- Include any necessary environment variables or configuration settings
- Implement a staging environment for testing before production deployment

5.3 Monitoring and Maintenance

- Set up monitoring tools on Render to track application performance and uptime
- Establish a routine for checking logs and addressing any issues
- Document the process for rolling back to a previous version if needed

5.4 Documentation

- Maintain comprehensive documentation on GitHub at `https://github.com/shivamlth27/Krishi_ai`
- Include a detailed README.md file with setup instructions, usage guidelines, and API documentation
- Generate and update a PDF version of the documentation using LaTeX for formal documentation needs

9. Collaboration and Review

9.1 Team Structure

The project team consists of five members:

- Ayush Kumar Mishra
- Ayush Patel
- Shivam
- Ujjwal Raj
- Divyanshu Prakash

Each team member's roles and responsibilities should be clearly defined and documented.

9.2 Collaboration Tools

- Use GitHub for version control and code collaboration
- Implement a branching strategy (e.g., Git Flow) and document it in the repository
- Utilize GitHub Issues for task tracking and bug reporting
- Set up a team communication platform (e.g., Slack, Microsoft Teams) for daily interactions

9.3 Code Review Process

- Implement a pull request (PR) process for all code changes
- Require at least one team member to review and approve each PR before merging
- Use GitHub's code review features to provide inline comments and suggestions
- Establish and document code review guidelines, including code style, documentation requirements, and performance considerations

9.4 Regular Team Meetings

- Schedule weekly team meetings to discuss progress, challenges, and upcoming tasks
- Rotate the responsibility of leading the meeting among team members
- Maintain a shared document for meeting agendas and minutes
- Use these meetings to review project timelines and adjust priorities as needed

9.5 Knowledge Sharing

- Encourage team members to document their work and share insights regularly
- Set up a shared wiki or knowledge base for the team
- Organize periodic technical sessions where team members can present on specific topics or challenges they've overcome

10. Compliance

10.1 Regulatory Compliance

- Ensure compliance with relevant regulations and standards
- Document all compliance-related procedures and checks

10.2 Licensing

- Clearly document the licensing of the project and all its components
- Ensure compliance with licenses of all used libraries and datasets

By following this SOP, the DFN-PSAN project team can ensure consistent, high-quality development practices, reproducible results, and ethical considerations throughout the project lifecycle.