Implementation of Paper: DFN-PSAN: Multi-level deep information feature fusion extraction network for interpretable plant disease classification

Introduction

Keywords: Deep learning, Image processing, Feature fusion, Multilevel features, Pixel attention, Disease classification

Accurate identification of crop diseases is an effective way to promote the development of intelligent and modernized agricultural production, as well as to reduce the use of pesticides and improve crop yield and quality. Deep learning methods have achieved better performance in classifying input plant disease images. However, many plant disease datasets are often constructed from controlled scenarios, and these deep learning models may not perform well when tested in real-world agricultural environments, highlighting the challenges of transitioning to natural farm environments under the new demand paradigm of Agri 4.0.

Based on the above reasons, this work proposes using a multi-level deep information feature fusion extraction network (DFN-PSAN) to achieve plant disease classification in natural field environments. DFN-PSAN adopts the YOLOv5 Backbone and Neck network as the base structure DFN and uses pyramidal squeezed attention (PSA) combined with multiple convolutional layers to design a novel classification network PSAN, which fuses and processes the multi-level depth information features output from DFN and highlights the critical regions of plant disease images with the help of pixel-level attention provided by PSA, thus realizing effective classification of multiple fine-grained plant diseases.

The proposed DFN-PSAN was trained and tested on three plant disease datasets. The average accuracy and F1-score exceeded 95.27%. The PSA attention mechanism saved 26% of model parameters, achieving a competitive performance among existing related methods. In addition, this work effectively enhances the transparency and interpretability of the model.

Motivation

Agriculture is a cornerstone of national development, playing a critical role in the economy and food security. The following facts underscore the importance and urgency of improving crop disease identification:

- According to the Food and Agriculture Organization (FAO), agricultural productivity is essential for ensuring food security for a growing global population.
- Crop diseases can reduce yields by up to 30% annually, leading to substantial economic losses for farmers (Jayagopal et al., 2022).
- Traditional methods of disease identification, such as manual inspection, are time-consuming and often ineffective, resulting in delayed responses to disease outbreaks.
- Early and accurate detection of diseases can significantly mitigate yield losses and improve the quality of crops, thus enhancing overall food security (Legrand, 2023).
- The rise of advanced technologies, such as machine learning and computer vision, offers new opportunities to transform crop disease detection by providing rapid, precise, and scalable solutions.

By leveraging these technologies, this project aims to address the limitations of traditional methods, enabling more efficient disease identification and timely interventions. This approach promises to safeguard agricultural production, reduce economic impacts, and support national food security.

Novel Lightweight Deep Learning Model

Based on the YOLOv5 network, we designed a novel lightweight deep learning model to identify plant diseases. The model, named DFN-PSAN, improves upon the original YOLOv5 network by:

- Keeping the feature extraction network (Backbone) and the feature fusion network (Neck)
- Removing the Head structure
- Designing a novel PSAN classification network structure for plant disease classification
- Implementing Pyramid Squeeze Attention (PSA) for PSAN, allowing the network to focus on important features and ignore unimportant ones

Main Contributions

- The DFN structure obtains information on plant disease characteristics at different scales through image feature fusion techniques.
- The PSAN structure utilizes rich information on important semantic features, with the embedded PSA attention mechanism reinforcing important

information and suppressing non-important information.

- A two-stage weather data augmentation technique is used for plant disease datasets in three real agricultural scenarios, improving model generalization and suppressing overfitting.
- The t-SNE method is used to interpret the feature layer data of the DFN-PSAN model through visualization of two-dimensional clustering distribution.
- The SHAP interpretable AI (XAI) visualization method explains whether DFN-PSAN correctly focuses on plant disease features or pattern information.

Dataset Acquisition

Three plant disease datasets constructed from actual scene collections were selected for this study, as they more accurately reflect the complex plant disease symptoms found in natural field environments.

Key Dataset: Katra-Twelve, a public dataset of leaf images provided by the University of Shri Mata Vaishno Devi in Katra, consisting of healthy and diseased leaf samples.

Note: The PlantVillage open-source plant disease dataset was not used due to its uniform background, which doesn't accurately represent real-world conditions. [Read More]

DFN-PSAN Model

DFN-PSAN: Multi-level deep information feature fusion extraction network for interpretable plant disease classification

Key highlights of the paper include:

---

1. Model Architecture: DFN-PSAN is built upon the YOLOv5 Backbone and Neck network, which serves as the base structure (DFN). It integrates pyramid squeeze attention (PSA) with multiple convolutional layers to design a novel classification network (PSAN). **Read More**

Read Paper

Plant Disease Dataset Information

2.1. Acquisition of datasets

Three plant disease datasets constructed from actual scene collections were selected as validation objects for this study:

1. **Katra-Twelve:** A public dataset of leaf images provided by the University of Shri Mata Vaishno Devi in Katra. It consists of 4503 images (2278 healthy, 2225 diseased) from 12 plants with 22 leaf types.
2. **BARI-Sunflower:** Constructed from the demonstration farm collection of Bangladesh Agricultural Research Institute (BARI), Gazipur. Contains 467 raw images of delicate leaves and infected sunflower leaves and flowers.
3. **FGVC8:** Constructed by Cornell Initiative for Digital Agriculture (CIDA). Contains 18,632 images with 12 categories of apple leaf diseases.

Table 1: Dataset Descriptions

	Classes	Numbers diseased	Numbers healthy

Katra-Twelve	265	254
Mango	345	276
Alstonia Scholaris	272	120
Jamun	142	232
Pongamia Pinnata	124	77
Pomegranate	118	—
Chinar	170	179
Gauva	279	322
Arjun	287	103
Jatropha	277	220
Lemon	133	159
Bael	—	148
Basil	470	509
BARI-Sunflower	398	—
Downy mildew	—	—
Leaf scars	—	—
Gray mold	512	1602
Fresh leaf	3181	165
FGVC8	1184	87
complex	1860	97
frog_eye_leaf_spot	120	4826
frog_eye_leaf_spot complex	686	200
healthy	4624	—
powdery_mildew	—	—
powdery_mildew complex	—	—
rust	—	—
rust complex	—	—
rust frog_eye_leaf_spot	—	—
scab	—	—
scab frog_eye_leaf_spot	—	—
scab frog_eye_leaf_spot complex	—	—

Data Preprocessing

Data preprocessing in this study is divided into two main stages, each with specific techniques to ensure high-quality data for model training:

1. ### Preliminary basic image preprocessing:

   - **Image read-in and format conversion:** Initially, images are read using the OpenCV `imread` method, which by default reads images in BGR format. These images are then converted to RGB format using the `cvtColor` method to ensure consistency in color representation.

- **Adaptive scaling:** To standardize image sizes and prevent loss of important image information, each image is scaled to a resolution of 256 × 256 pixels. This is achieved through adaptive scaling techniques that maintain image quality.
- **Gaussian filtering:** The `GaussianBlur` method is applied to smooth the image and reduce noise. A Gaussian kernel of size 3 is used, which replaces each pixel's value with the average value of its surrounding pixels, effectively minimizing noise and fine details.
- **Non-local mean noise reduction:** To further reduce noise while preserving details, the `fastNlMeansDenoisingColored` method is employed. This technique uses parameters `h` and `hColor` set to 3 to maintain the texture and edges of the image while denoising.
- **Brightness and contrast adjustment:** The `convertScaleAbs` method is used to adjust the image's brightness and contrast. Parameters `α = 1.00` and `β = 0` are applied to control these properties, ensuring that the image has optimal visibility and contrast for feature extraction.

2.

Weather Data Augmentation

- FGVC8: Six data augmentations per image for categories with sample volumes below 500
- Katra-Twelve and BARI-Sunflower: Two data augmentations per image
- **Solar illumination transformation:** To simulate natural lighting conditions, the `RandomSunFlare` technique is used. Solar flares are added to the image with the appearance region determined by parameters (x_min, y_min, x_max, y_max). For this study, the upper right corner (0.9, 0, 1, 0.5) and upper left corner (0.0, 0.0, 1.0, 0.1) were chosen, and the aperture radius parameter was set to 300.
- **Raindrop transformation:** The `RandomRain` technique simulates the effect of rain on images. This technique adds raindrops with a size of 1.0, using a drizzle type with a brightness coefficient of 0.6 to mimic realistic rain effects.
- **Shadow transformation:** To account for natural shadow effects, the `RandomShadow` technique is employed. Shadows are added randomly, with the number of shadows varying between 1 and 5 and the side parameter of the shadow polygon set to 6 to represent realistic shadow patterns.
- **Fog transformation:** To simulate foggy conditions, the `RandomFog` technique adds fog to different parts of the image. Fog intensity (fog coef) and fog circle transparency (alpha coef) are set between 0.25 and 0.8, with a fog parameter value of 0.3, to blur the background and simulate natural fog effects.

The image processing and augmentation were performed using VSCode and Python 3.10, with PyTorch 1.13.1 + cu117 and the OpenCV library, utilizing GPU acceleration for efficient processing. Due to dataset imbalances in the FGVC8 dataset, six data augmentations per image were applied to categories with fewer than 500 samples. The Katra-Twelve and BARI-Sunflower datasets underwent only two augmentations per image. The effectiveness of the two-stage image processing, including both preliminary preprocessing and weather data augmentation, is illustrated in Figures 2 and 3, showing the impact of these methods on image quality and dataset robustness.

Technical Details

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

Dataset Division

After preprocessing, the three new datasets were randomly divided into three parts:

- Training set: 80%
- Validation set: 10%
- Test set: 10%

DFN-PSAN Architecture with PSA

The DFN-PSAN (Deep Fusion Network with Pyramid Squeeze Attention Network) architecture is an improvement of YOLOv5, designed for more accurate plant disease detection. It focuses on enhanced feature extraction and classification through the use of Pyramid Squeeze Attention (PSA), which boosts the model's ability to focus on important features in plant images.

Key Components

1. YOLOv5 for Feature Extraction

YOLOv5, a real-time object detection model, handles feature extraction. While YOLOv5n offers speed and low weight, it lacks deep feature extraction capability. To address this, DFN-PSAN introduces modifications to improve feature extraction, fusion, and convergence speed.

2. YOLOv5 Architecture

- **Backbone:** Utilizes CSPDarkNet with a 6 × 6 convolutional layer replacing the older Focus structure.
- **SPP Module:** Expands the receptive field, extracting both local and global features through max-pooling at various scales. The operation can be represented as:

$$\mathrm{SPP}(FM) = \mathrm{Concat}(\mathrm{MaxPool}(FM, k_1), \mathrm{MaxPool}(FM, k_2), \mathrm{MaxPool}(FM, k_3))$$

where FM is the input feature map and k_i are the pooling kernel sizes.

3. Neck (DFN)

The Neck combines the Feature Pyramid Network (FPN) and Path Aggregation Network (PAN). The FPN upscales feature maps from lower levels to capture high-level semantic information, while the PAN downscales feature maps from higher levels to improve localization accuracy. This can be described mathematically as:

$$\mathrm{FPN}(x) = \mathrm{Upsample}(x) + \mathrm{SkipConnection}(x)$$

$$\mathrm{PAN}(x) = \mathrm{Downsample}(x) + \mathrm{SkipConnection}(x)$$

where x represents the feature maps at various levels of the network.

4. PSAN Classification Layer

The PSAN classification layer replaces YOLOv5's Head to enhance classification performance. The Pyramid Squeeze Attention mechanism refines the focus on important features using:

$$\mathrm{PSA}(FM) = \mathrm{GAP}(\mathrm{Attention}(FM))$$

where GAP represents Global Average Pooling and $\mathrm{Attention}$ denotes the attention mechanism applied to the feature map FM.

5. Feature Fusion and Attention

The Neck structure integrates features from various layers, improving the network's ability to handle objects at different scales. The attention mechanism, which can be expressed as:

$$\mathrm{Attention}(FM) = \sigma(W \cdot FM + b)$$

where σ is the activation function, W is the weight matrix, and b is the bias, enhances the focus on relevant features. Classification is performed using the Softmax function:

$$\mathrm{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

which converts the output logits into probabilities for each class.

6. Training

Training involves updating the model parameters using a deep neural network with 30 hyperparameters. The optimization process minimizes the cross-entropy loss function with label smoothing, which can be expressed as:

$$\text{Loss} = -\sum_{i=1}^{N} \left(y_i \log(p_i) \right)$$

where N is the total number of categories, y_i is the prediction result for category i, p_i is the confidence score of the network output for category i, and ϵ is the label smoothing hyperparameter. Label smoothing modifies y_i as follows:

$$y_i = \begin{cases} 1 - \epsilon & \text{if } i \text{ is the target category} \\ \frac{\epsilon}{N} & \text{if } i \text{ is not the target category} \end{cases}$$

The loss function with label smoothing helps improve the model's generalization by preventing it from becoming too confident about its predictions.

Conclusion

The DFN-PSAN architecture, through its enhancements and mathematical formulations, achieves superior plant disease detection by integrating advanced feature extraction, attention mechanisms, and effective classification methods.