**DEPLOYING MACHINE LEARNING MODELS FOR PLANT DISEASE DETECTION ON WEB PLATFORMS**

Lê Phan Nhật Minh1,2,3, \*\_2252478\_Computer Science, Phan Quang Minh 1,2,3\_2252486\_Computer Science,    
Nguyễn Minh Huy 1,2,3\_2211221\_Computer Engineering, Từ Khánh Minh 1,2,3\_2252494\_Computer Science,

1 Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam

2 Office for International Study Programs, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam

3 Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam

\* Corresponding author: minh.leiguana662@hcmut.edu.vn

**Abstract**

In the current agricultural practice, timely and accurate detection of leaf disease in crops is essential for maintaining crops health and yield leaf diseases, suggesting that AI-powered diagnosis can reduce human workloads, help deliver timely intercept to re However, traditional methods of disease identification are labor-intensive, subjective, and reliant on expert knowledge, which can delay intervention and exacerbate crop damage. This project investigates the application of artificial intelligence (AI), especially deep learning models, to automate detection and intervention of crop diseases, using durian leaf as an example subject. Using convolutional neural networks (CNNs), the proposed method classifies some common durian leaf diseases by analyzing visual patterns captured from the leaves, and performing intervention methods to help control the disease detected. Additionally, we use pre-processing methods to isolate the data from noises, improving disease detection accuracy. In-lab results demonstrate high accuracy and reliability in identifying crop damage, and improve productivity. To further enhance the model's performance and generalization capabilities, future work will involve expanding the dataset with additional disease samples and its intervention system, and incorporating diverse environmental conditions.

***Keywords:*** *AI (Artificial Intelligence), CNNs (Convolutional neural networks), deep learning, prediction, crop diseases.*

**1. Introduction**

In modern agriculture, timely and accurate detection of plant diseases is essential for ensuring crop health, optimizing yield, and minimizing economic losses. Plant disease presents considerable challenges to farmers, particularly high-value crops like durian. According to the Food and Agriculture Organization (FAO), plant diseases cause global crop losses of up to 20–40% annually [1]. In tropical regions of Southeast Asia, this poses a critical threat to durian which is a major economic product. In Vietnam, a leading durian exporter, leaf diseases such as algal leaf spot, leaf blight, and leaf spot can severely impact yields. Traditional detection methods rely heavily on manual visual inspection by agricultural experts, which are labor-intensive, subjective, time consuming, result in late interventions, exacerbating crop damage and reducing productivity. Recent advancements in artificial intelligence (AI) offer promising automated solutions to enhance detection accuracy. Convolutional neural networks (CNNs), which have remarkable success in image-based classification tasks, are widely used in detecting plant diseases in crops such as rice, wheat, and tomatoes. However, few studies have focused on durian, a crop with characteristic disease patterns and often grown on large-scale farms. Existing approaches often require specialized hardware or expertise, limiting their adoption in resource-constrained agricultural settings. To address this, we are interested in integrating statistical texture analysis with deep learning, leveraging web-based applications to improve detection robustness. Specifically, we design and implement a web-based detection mechanism that combines CNNs with Jensen-Shannon Divergence (JSD) in texture analysis, to automate the detection of durian leaf diseases. By employing tile-based analysis and selection, that platform enhances detection accuracy and reduces computation load, making it more resilient to real-world variations in lighting and leaf conditions. The web deployment requires minimal technical expertise, mobility support, thereby empowering farmers and agricultural extension workers to take timely and informed actions.

The main contributions of this work are:

* Develop an AI enabled web-based platform for detecting durian leaf disease in captured image.
* Propose two algorithms for dividing and selecting picture tiles that carry characteristics.

**2. System design**

This section presents the system design for a web-based platform that automates the detection of durian leaf diseases using AI. The proposed system is a web application that allows users to upload high-resolution images of durian leaves through a browser-based interface.

The proposed system has three primary components as in Figure 1. The dataflow processing includes image preprocessing, texture analysis via JSD, and a lightweight convolutional neural network was trained using TensorFlow.js for in-browser inference.

A diagram of a computer network

AI-generated content may be incorrect.

**Figure 1: System architecture**

**2.1 Front end**

**T**he front-end is a web interface built using HTML5, CSS3, and JavaScript, designed for simplicity and accessibility. Key features include:

* Image Upload: Users can upload durian leaf images (e.g., JPEG, PNG) via a drag-and-drop interface or file picker.
* Result Display: Results are presented with disease labels (e.g., "algal," "leafblight," "normal") and confidence scores, optionally accompanied by visualizations highlighting diseased tiles.
* User Experience: The interface includes progress indicators during processing and clear error messages for invalid inputs (e.g., non-image files).

**2.2 Back end**

The back-end is implemented using Node.js with Express.js, handling image processing and model inference. It uses TensorFlow.js to load and execute the pre-trained CNN model. Key responsibilities include:

* Image Handling: Receives uploaded images via HTTP POST requests using Multer for file processing.
* Preprocessing and Analysis: Performs tile extraction, histogram computation, JSD calculation, and CNN classification, as described in the Proposed Algorithm.
* Result Aggregation: Combines tile-level classifications to determine the overall disease status of the leaf.
* Communication: Returns results to the front-end as JSON objects containing disease labels and probabilities.

**2.3 Machine learning model**

The core of the system is a pre-trained CNN model, loaded via TensorFlow.js. The model classifies image tiles into categories such as "algal," "leafblight," "leafspot," "alloca," or "normal." The model is stored on the server and optimized for web-based inference, balancing accuracy and computational efficiency. These images are processed on a server to detect diseases by analyzing visual patterns in leaf tiles

The system employs a two-pronged approach: texture analysis using JSD to assess inter-tile similarities, and semantic classification using a pre-trained CNN to categorize tiles as diseased or normal. Results are aggregated and presented to the user, indicating disease presence and confidence levels, with potential visualizations of affected areas. The system is designed to be user-friendly, requiring minimal technical expertise, and scalable to handle multiple users.

**3. Proposed Algorithm**

This section proposed a technique created to analyze and categorize durian leaf image tiles utilizing a combination of deep learning classification and statistical distribution analysis presented in this section. **Jensen-Shannon Divergence (JSD)** is used to assess texture similarity, and **a convolutional neural network (CNN)** is used to classify each tile semantically.

**3.1 Overview.**

The technique divides a high-resolution image into smaller, fixed-sized tiles (tiles). The tiles are then examined in two complimentary ways:

**Histogram-Based Divergence**: Using grayscale intensity histograms and the Jensen-Shannon Divergence metric, compare local tile texture to a reference.

**CNN-Based Classification**: Feeding the tile into a pre-trained model to determine if the leaf is diseased or normal.

The purpose is to pinpoint illness areas and evaluate inter-tile changes in visual texture.

**3.2 Preprocessing and tile extraction**

To capture rich color‐texture information in each tile (rather than just grayscale), we compute a **per‐channel histogram** over the red, green, and blue pixel values and then concatenate them into one feature vector. Concretely:

* Tile representation: let denote an RGB image tile
* Per-channel Flattening:
* Histogram Binning (Equal-Width):
* Normalization to a Probability Distribution:
* Concatenation into a Single Feature Vector:

Let the original image have dimensions (height × width). The number of tiles generated is:

(1)

Where P is the tile size. Each tile is treated as an independent unit for subsequent analysis steps including featured comparison and classification.

This generalized strategy enables the algorithm to process images of arbitrary resolution while maintaining a consistent approach to texture and semantic analysis.

**3.3 Color histogram calculation**

To characterize the color distribution of each extracted image tile, a normalized intensity histogram is computed.

* **Tile representation**: Let denote an RGB image tile of height H and width W, and we denote .
* **Per-channel flattening**: For each channel, , we extract the single-channel tensor , we then flatten it into a vector of length:
* **Histogram binning (Equal-width)**: We divide the interval into equal bin-width. Then, we define bin-edges: indexed by k: , .

For each channel and bin index we define the histogram of pixels falling into bin

(2)

* **Normalization to a Probability Distribution**: We form a normalized the counter by dividing by the total number of pixels . Hence, for each .
* **Concatenation into a Single Feature Vector**: Finally, we concatenate the three RGB channel vectors , , and into a single 3B-dimensional tensor:

(3)

**3.4 JSD based scoring**

To assess the textural difference between any two image tiles, we use the JSD, a symmetric and bounded measure derived from the Kullback-Leibler divergence. JSD is particularly useful for comparing normalized histograms since it always returns a finite result and considers both distributions equally.

**3.4.1 JSD definition**

Let and be two discrete probability distributions (histograms) over the same set of intensity-bins. Define their **mixture distribution** M by: .

The JSD between P and Q is then:

Where the **Kullback-Leibler divergence**  is given by

**3.4.2 JSD based scoring implementation**

For each tile associate with bin and channel , we first transform the associated set of histogram in equation (2) to which has the same size of in equation (3) then we calculate the

(4)

can be used to rank the tile later.

**Numerical Stability:**

To prevent undefined expressions (e.g, log(0) of division by zero), we add a small constant to all bin probabilities before computing logarithms:

, ,

**Symmetry & Boundedness:**

* .
* (using natural logarithms) in which (i)*Low JSD* : tiles have **very similar** grayscale distributions; and*High JSD*: tiles are **strongly dissimilar** in texture.

**3.4.3 Tile entropy**

In CNN’s prediction, we compute the entropy of the classification probability distribution for each tile. Entropy, defined as , is the probability of class *i,* measures the unpredictability of the model’s output. High entropy indicates low confidence in the prediction, which helps penalize ambiguous tiles and refine the attention score.

This complements the SD-based texture analysis and the CNN-based unhealthy probability , ensuring a balanced assessment of both visual and semantic features.

UnhealthyProb (UP) represents the sum of CNN output probabilities for non-healthy classes (e.g algal, leafblight, leafspot, alloca). Calculated as . quantifies the model’s confidence in detecting a disease in the tile .

**3.5 Tile Selection Algorithm based on Probability and Attention Weights**

**3.5.1 Purpose**

This algorithm is designed to process an image by dividing it into sub-regions (tiles), computing an important score for each tile based on Jensen-Shannon divergence, unhealthy probability from CNN predictions, and entropy, and then selecting N tiles for further work using two mechanisms:

* **Random** **selection** with probability
* **Weighted** **selection** based on the attention map (attentionMap) with probability

This algorithm enhances the original selection mechanism by incorporating the probabilistic selection approach from Algorithm 2, ensuring a balance between exploring new tiles and exploiting important ones, while producing output compatible with subsequent processing steps.

**3.5.2 Algorithm 1**

A screenshot of a computer program

AI-generated content may be incorrect.

**Algorithm 1** describes how to compute and update the attention score for each image tile based on visual and semantic cues. The algorithm operates on a set of tiles extracted from an input image and aims to identify the most informative regions that may indicate disease or abnormality.

For each tile , the algorithm first computes a **color histogram** and compares it to a reference histogram using the **Jensen-Shannon Divergence (JSD)** as in equation (4) to form ). It then feeds the tile into a trained classification model to obtain class probabilities. The **unhealthy probability** (i.e., total probability of all classes excluding “healthy”) is extracted, and the **entropy** of the prediction distribution is computed to capture uncertainty.

These three components—JSD, unhealthy probability, and entropy—are combined linearly to form an **attention score**, with weights emphasizing the semantic component. To ensure temporal smoothness and reduce noise sensitivity, each tile's score is updated using an **exponential moving average** governed by a smoothing factor.

Finally, all tiles are sorted by their updated attention scores, and the top-$k$ tiles are selected as the most significant regions for downstream tasks such as visualization or reporting.

**3.5.3 Algorithm 2**

A screenshot of a computer program

AI-generated content may be incorrect.

After calculating the attention score for each tile (Algorithm 1), Algorithm 2 selects the most informative tiles for focused analysis. The goal is to strike a balance between **exploration** (random sampling) and **exploitation** (attention-guided selection). With probability, the algorithm performs uniform random sampling, enabling discovery of novel or unexpected patterns. Otherwise, it performs **probability-weighted sampling** based on the attention map V, which encodes each tile’s anomaly likelihood (via JSD, CNN uncertainty, and entropy).

This selection mechanism prevents the model from overfitting to only high-scoring tiles, while still prioritizing those most likely to exhibit disease characteristics. The selected tiles are then visualized or stored for further classification or diagnosis.

In our durian leaf disease detection framework, this approach enables **localized and interpretable detection**, helping identify suspicious regions across various leaf conditions and types, especially in large, high-resolution images.

**4 Numerical results**

**4.1 Tile-Based Attention Scoring**

Each tile's score is computed as:

Comparing when choosing random tiles versus chosen tile in terms of the scoring .

Highest scoring tile (A = 1.00) is directly over a leaf with multiple visible infections. Random tiles, although spread evenly, miss these hotspots entirely. This further validates the utility of the attention score in prioritizing meaningful image regions under limited tile budget constraints.

The attention-based tiles (A = 0.95, 0.89, 0.87, 0.86) are concentrated on regions exhibiting clear signs of disease (dark lesions and discoloration). In contrast, most of the random tiles fall on healthy areas or leaf edges with little to no visible symptoms. This supports that the attention mechanism is effectively localizing diagnostically important regions, which are likely to contribute more to accurate classification as shown in figure 2a.

|  |  |
| --- | --- |
|  |  |
|  |  |
| **Figure 2: Tile selection variants on and** | |

**4.2 Attention score factor impact**

Figure 3 visualization shows that the selected tiles closely match the ground-truth bounding box, meaning our attention mechanism is really focusing on the diseased areas. This demonstrates that the method works not just in theory but also aligns well with actual annotations. As a result, our approach provides clearer and more accurate localization of diseased leaves than the baseline.

A close up of leaves

AI-generated content may be incorrect.

**Figure 3: Selected tiles and VGG bounding box**

To investigate the impact of different weighting schemes on the final attention score, we conducted experiments by adjusting the contribution of each component: JSD, Unhealthy Probability, and Entropy—represented as λ₁, λ₂, and λ₃ respectively.

A graph of different colored bars

AI-generated content may be incorrect.

**Figure 4: Attention score across weight configs**

According to the bar chart in figure 4, when the weights are equally distributed (λ₁=λ₂=λ₃=0.33), the final attention scores across different images remain moderate (e.g., algal\_2.jpg yields a score of 0.848), indicating a balanced representation of all contributing factors. However, when we increased the influence of the unhealthy probability (λ₂=0.6) and reduced the others (λ₁=λ₃=0.2), the attention scores consistently increased for all images (e.g., algal\_2.jpg rises to 0.9139). This suggests that unhealthy class probability is a dominant indicator for highlighting salient tiles. Conversely, in configurations where entropy or JSD received more weight, the final attention scores decreased, highlighting their subtler contribution. These observations validate that while all components are useful, the model is most sensitive to direct class probability signals, and tuning the λ-values offers flexible control over the attention policy. This relationship confirms the significance of the attention score as a tunable indicator of visual relevance for tile selection in disease detection.

|  |  |
| --- | --- |
|  | |
| (a) | |
|  |  |
| (b) | (c) |
| **Figure 5: The impact of jsd on attention score** | |

Figure 5 shows the contribution validation of the JSD component in the attention scoring mechanism. We conducted a controlled test across a dataset of annotated leaf images. Each image was divided into overlapping patches of size 150×150 pixels with a stride of 75 pixels. For each patch, we computed its color histogram and compared it to a reference histogram derived from healthy patches using JSD. Alongside, each patch was passed through a classification model to obtain predicted class probabilities and entropy.

We ran multiple configurations with different λ settings and extracted top-k scoring patches. For each configuration, we plotted scatter plots and linear regressions between JSD and attention score to observe correlation patterns and to empirically evaluate whether higher divergence leads to increased attention.

**4.3 Correlation matrix between JSD, entropy, unhealthyProb, and attentionScore.**

In figure 6, the correlation heatmap helps us understand how the three components of the attention score—Jensen-Shannon Divergence (JSD), Unhealthy Probability, and Entropy—are related. Interestingly, JSD and Entropy show a moderately negative correlation (−0.56), meaning that when a tile has a color distribution that looks very different from a healthy reference (high JSD), its classification tends to be more confident (low entropy). On the other hand, Unhealthy Probability doesn't show strong correlation with either JSD or Entropy, indicating it works independently as a signal of how likely a tile is diseased. This balance is important because it suggests that each part of the score captures different information, making the combined attention score more robust and reliable.

A screenshot of a computer screen

AI-generated content may be incorrect.

**Figure 6: The correlation heatmap of multiple factors in attention score**

**4.4. System deployment verification**

In figure 7, we validate the system deployment and demonstrate web portal functionality by uploading images of durian leaves at various resolutions and obtaining disease prediction results within seconds. The portal is built with Node.js, enabling HTTP through TLS (SSL) secure connections. The backend leverages TensorFlow.js [5] to run a trained convolutional neural network (CNN) model for image classification.

For automated processing, the system automatically divides each uploaded image into smaller tiles and applies proposed algorithms to select the most informative regions before passing them to the model. This approach improves efficiency with minimal accuracy trade off and ensures reliable results even for images with varied lighting or background noise.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 7: System web portal deployment**

**6. Conclusion**

This study presents a practical, web-based AI platform that combines CNNs and Jensen-Shannon Divergence (JSD) for an accurate detection of durian leaf diseases. By using tile-based image analysis and a user-friendly interface, the system enables farmers to identify issues like algal leaf spot, leaf blight, and leaf spot in real-time, without requiring technical expertise.

The proposed tile selection algorithms improve performance while keeping the accuracy competitive. Future work will focus on expanding the dataset, adding more disease types and intervention methods, and testing under varied environmental conditions. Integrating advanced deep learning techniques and enhancing the intervention system will further boost accuracy and performance.The designed and implemented system offers a scalable solution to support early disease detection and better crop management in durian farming and possibly other agricultural plants. Due to its native javascript implementation, the system can also be easily deployed in mobile environments that are highly flexible and adaptable, making it very suitable for living conditions in farming areas.

**References**

[1]: FAO report on global crop losses. FAO, “The State of Food and Agriculture”.

[2]: Szabó, M. (2024). Building Ensemble Models with Web Services on Microservice Architecture. Informatica, 48(7). <https://doi.org/10.31449/inf.v48i7.4918>

[3]: Englesson, E., & Azizpour, H. (2021, May 10). Generalized Jensen-Shannon Divergence Loss for Learning with Noisy Labels. arXiv.org. <https://arxiv.org/abs/2105.04522>

[4]: Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org