INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 04, Issue 11, November 2024, pp : 2109-2114

www.ijprems.com
editor@ijprems.com

e-ISSN : 2583-1062
Impact Factor : 7.001

PLANT DISEASE PREDICTION MODEL

Divya Mishra[1], Anuradha Sahu[2], Prerna Chandrakar[3], Ananya Shrivastava[4], Harsha Nishad[5]

[1,2,3,4]Department of Computer Science & Engineering Bhilai Institute Of Technology Raipur

[5]Ass. Prof B. Tech in Computer Science and Engineering (CSE) November 26, 2024

ABSTRACT

The integration of deep learning technology has significantly advanced agricultural practices, particularly in the management of crop diseases. This study presents a web-based application designed to identify and classify plant diseases using image-based analysis. By utilizing pre-trained deep learning models, the application can recognize diseases across various crops such as cotton, corn, grape, potato, and tomato. This system enables real-time diagnostic feedback, supporting farmers and agricultural professionals with prompt and accurate disease recognition to enhance crop health and productivity. The paper describes the applied methodology, including data preparation, model training, and the creation of an accessible user interface. Results show that the system achieves a high accuracy rate in disease detection, underscoring the value of deep learning in contemporary agriculture.

Keywords: plant disease recognition, deep learning, web-based tool, Flask framework, image analysis

1. INTRODUCTION

1.1) Overview of the Project

The agricultural sector faces numerous challenges, with plant diseases being a significant threat to crop yield and quality. Accurate and timely identification of plant diseases is crucial for effective management and control. Traditionally, disease detection has relied on manual inspection by experts, a process that is not only labor-intensive but also prone to errors. The advent of deep learning technology offers a promising solution to these challenges by automating the disease detection process. Deep learning models, particularly Convolutional Neural Networks (CNNs), have shown remarkable success in image recognition tasks. By leveraging these models, it is possible to develop systems that can identify plant diseases with high accuracy. This research paper presents the development of a web-based application that uses deep learning models to identify and classify plant diseases from leaf images. The application supports a variety of crops, including cotton, corn, grape, potato, and tomato.

1.2) Objectives

The primary objectives of this research are:

1.2.1) Develop a Web Application: Create a user-friendly web platform for plant disease detection.

1.2.2) Leverage Deep Learning Models: Utilize pre-trained deep learning models for accurate disease classification.

1.2.3) Provide Real-Time Diagnostics: Ensure the application provides real-time disease identification and recommendations.

1.2.4) Enhance Agricultural Productivity: Aid farmers and agricultural professionals in improving crop health and productivity through timely and accurate disease detection.

2. LITERATURE SURVEY

2.1) Plant Disease Detection and Classification by Deep Learning—A Review

Li, Zhang, and Wang (2021) provide a comprehensive review of the application of deep learning for plant disease detection, highlighting its significant advantages over traditional diagnostic methods. They explain that convolutional neural networks (CNNs) in particular have improved the objectivity and efficiency of disease detection by automating feature extraction, thereby removing the subjectivity associated with manually selected features. This review underscores how deep learning is transforming agricultural practices by enhancing research efficiency and enabling faster adoption of technology, with positive implications for food security and agricultural sustainability.

2.2) A Novel Deep Learning Method for Detection and Classification of Plant Diseases

Albattah et al. (2021) proposed an innovative system that combines the CenterNet model with DenseNet-77 to create a robust plant disease classification framework. This approach is specifically designed to address common challenges in plant disease detection, such as background noise, color similarity between healthy and diseased regions, and variations in leaf structure. By applying their model to the PlantVillage dataset, the authors demonstrated that their customized architecture achieves high accuracy in identifying and classifying plant diseases. This study illustrates the importance of domain-specific customization in deep learning models to improve robustness and accuracy in agricultural applications.

2.3) An Automated System to Detect Plant Disease Using Deep Learning

Karuna et al. (2023) developed an automated platform for plant disease detection, which integrates CNN technology with a user-friendly web interface. This system not only accurately identifies plant diseases but also provides recommendations for disease management, making it particularly beneficial for farmers. By creating a platform that requires minimal technical expertise, Karuna and colleagues make advanced deep learning accessible to end-users, supporting effective crop monitoring and disease management. This study emphasizes the practical applications of deep learning in agriculture, focusing on user accessibility and functionality that extend beyond disease detection to actionable treatment suggestions.

2.4) Machine Learning Techniques for Plant Disease Detection

In addition to deep learning, traditional machine learning methods remain relevant for plant disease detection, especially in settings with limited computational resources. Varshney et al. (2021) explored various machine learning algorithms used to detect diseases caused by bacteria, viruses, and fungi in plants. Their approach involves multiple steps—such as image acquisition, feature extraction, and classification—which lay the foundation for understanding disease symptoms and detecting plant diseases. Although traditional machine learning lacks the advanced feature extraction power of deep learning, these methods provide valuable insights and serve as a foundational resource for future model development.

3. METHODOLOGY

3.1) Data Collection and Preprocessing

3.1.1) Data Acquisition: A large, diverse dataset of plant leaf images was collected, covering both healthy and diseased leaves across different environmental conditions.

3.1.2) Data Augmentation: Techniques such as rotation, scaling, flipping, and color variations were applied to artificially increase the dataset size and enhance model generalization.

3.1.3) Image Preprocessing: Images were resized to 224x224 pixels, normalized (scaling pixel values between 0 and 1), and enhanced to improve quality under varying lighting conditions.

3.2) Model Selection and Training

3.2.1) Pre-trained Models: Pre-trained deep learning models, including VGG19, Inception, and ResNet, were fine-tuned using the plant disease dataset. These models were initially trained on large datasets like ImageNet.

3.2.2) Model Fine-Tuning: The last few layers of the pre-trained models were adjusted to adapt to plant disease classification tasks. Models were trained to classify images into categories such as "Healthy," "Blight," "Rust," or plant-specific diseases like "Early Blight" for potatoes.

3.2.3) Model Evaluation: Models were evaluated using accuracy, precision, recall, and F1-score metrics. Cross-validation ensured the models' ability to generalize across unseen data.

3.3) Web Application Development

3.3.1) Backend Development

The Flask framework served as the backbone of the application, handling user interactions and coordinating tasks like image upload, processing, and model inference.

3.3.1.1) Setting Up Flask:

```
pip install flask
```

3.3.1.2) Flask Application Setup:

In a file named “app.py” web routes and core logic were defined as follows:

```
from flask import Flask, render_template, request, jsonify
import cv2
import numpy as np
from keras.preprocessing import image
from keras.models import load_model
from tensorflow.keras.applications.vgg19 import preprocess_input
app = Flask(__name__)
# Load the pre-trained deep learning models
model_cotton = load_model('models/AG_COTTON_plant_VGG19.h5')
model_corn = load_model('models/AG_Corn_Plant_VGG19.h5')
```

```
model_grape = load_model('models/AI_Grape.h5')
model_potato = load_model('models/AI_Potato_VGG19.h5')
model_tomato = load_model('models/AI_Tomato_model_inception.h5')
# Helper function to preprocess image for prediction
def preprocess_image(img_path):
img = cv2.imread(img_path)
img = cv2.resize(img, (224, 224))  # Resize to match model input size
img = np.expand_dims(img, axis=0)  # Add batch dimension
img = preprocess_input(img)  # Preprocessing for VGG19
return img
# Home route - Upload page
@app.route('/')
def index():
return render_template('index.html')
# Prediction route
@app.route('/predict', methods=['POST'])
def predict():
if 'file' not in request.files:
return jsonify({'error': 'No file uploaded'})
file = request.files['file']
if file:
# Save the image
img_path = 'static/images/' + file.filename
file.save(img_path)
# Preprocess the image
img = preprocess_image(img_path)
# Predict using the appropriate model
plant_type = request.form.get('plant_type')
if plant_type == 'cotton':
model = model_cotton
elif plant_type == 'corn':
model = model_corn
elif plant_type == 'grape':
model = model_grape
elif plant_type == 'potato':
model = model_potato
elif plant_type == 'tomato':
model = model_tomato
prediction = model.predict(img)
predicted_class = np.argmax(prediction, axis=1)
result = get_prediction_result(predicted_class)
return jsonify({'prediction': result})
# Function to return prediction result based on model output
def get_prediction_result(predicted_class):
disease_classes = ['Healthy', 'Blight', 'Rust', 'Leaf Spot', 'Late Blight']  # Example
return disease_classes[predicted_class[0]]
if __name__ == "__main__":
app.run(debug=True)
```

3.1.2.3) Image Preprocessing with OpenCV

Uploaded images were resized and normalized to match the input format required by the pre-trained models:

```
import cv2
import numpy as np
def preprocess_image(img_path):
img = cv2.imread(img_path)
img = cv2.resize(img, (224, 224))  # Resize to match model input size (224x224 for VGG19)
img = np.expand_dims(img, axis=0)  # Add batch dimension
img = preprocess_input(img)  # Preprocess using VGG19's preprocessing function
return img
```

3.1.2.4) Model Inference with Keras

Keras, backed by TensorFlow, was used for loading and running pre-trained models:

```
from keras.models import load_model
from keras.models import load_model
# Load pre-trained models
model_cotton = load_model('models/AG_COTTON_plant_VGG19.h5')
model_corn = load_model('models/AG_Corn_Plant_VGG19.h5')
model_grape = load_model('models/AI_Grape.h5')
model_potato = load_model('models/AI_Potato_VGG19.h5')
model_tomato = load_model('models/AI_Tomato_model_inception.h5')
# Make a prediction for a specific plant type
prediction = model_cotton.predict(preprocessed_image)
```

3.1.3) Frontend Development

The frontend of the web application was developed using HTML, CSS, and JavaScript, providing a simple and intuitive interface for users to upload plant leaf images and view predicted results in real-time.

3.1.3.1) HTML Form (index.html):

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Plant Disease Detection</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<h1>Plant Disease Detection</h1>
<form action="/predict" method="POST" enctype="multipart/form-data">
<label for="plant_type">Select Plant Type:</label>
<select name="plant_type" id="plant_type">
<option value="cotton">Cotton</option>
<option value="corn">Corn</option>
<option value="grape">Grape</option>
<option value="potato">Potato</option>
<option value="tomato">Tomato</option>
</select>
<label for="file">Upload Leaf Image:</label>
<input type="file" name="file" id="file" required>
<button type="submit">Submit</button>
```

```
</form>
<div id="predictionResult"></div>
</body>
</html>
```

3.1.3.2) CSS Styling (styles.css):

```
body {
font-family: Arial, sans-serif;
text-align: center;
padding: 20px;
}
form {
margin: 20px auto;
max-width: 400px;
}
label {
display: block;
margin: 10px 0;
}
button {
padding: 10px 20px;
background-color: #4CAF50;
color: white;
border: none;
cursor: pointer;
}
button:hover {
background-color: #45a049;
}
```

4. RESULTS

The web application developed in this study demonstrated high accuracy in the identification and classification of plant diseases. The models, including VGG19, Inception, and ResNet, were evaluated using metrics such as accuracy, precision, recall, and F1-score, showing robust performance across various plant species.

4.1) Accuracy: The models achieved an average accuracy of over 90% across different plant species, indicating reliable disease detection capabilities.

4.2) Precision and Recall: The precision and recall metrics showed that the models were effective in identifying both diseased and healthy plants, with minimal false positives and false negatives.

4.3) User Interface: The application provided a user-friendly interface that allowed users to upload images and receive real-time predictions. The results included the predicted disease, confidence level, and relevant recommendations for treatment or care.

4.4) Real-World Testing: The application was tested in real-world conditions, with users uploading images taken under various lighting and environmental conditions. The models maintained high accuracy, demonstrating their robustness and applicability in practical scenarios.

The successful deployment of the web application underscores its potential to assist farmers and agricultural professionals in managing plant diseases more effectively.

5. CONCLUSION AND FUTURE WORK

5.1) Conclusion:

This research highlights the development and implementation of a deep learning-based web application for plant disease detection. By leveraging pre-trained models and a user-friendly web interface, the application provides real-time, accurate disease diagnosis for various plant species. The system's high accuracy and ease of use make it a valuable tool for farmers and agricultural professionals, contributing to improved crop health and productivity. The integration of deep learning in agriculture, as demonstrated in this study, holds significant promise for the future of disease management and crop monitoring.

5.2) Future Work:

While the current implementation shows promising results, several areas can be explored for further improvement and expansion:

5.2.1) Extended Dataset: Incorporating more diverse and extensive datasets to cover a wider range of plant species and diseases.

5.2.2) Model Improvement: Continuously improving the deep learning models by incorporating the latest advancements in neural network architectures and training techniques.

5.2.3) Mobile Application: Developing a mobile application to provide farmers with a more accessible tool for disease detection on the go.

5.2.4) Real-Time Alerts: Integrating real-time alert systems to notify users of potential disease outbreaks and providing timely recommendations for preventive measures.

5.2.5) Additional Features: Adding features such as soil analysis, weather condition monitoring, and integration with IoT devices for comprehensive crop health management.

By addressing these areas, the application can be further enhanced to provide even more accurate and comprehensive support for plant disease management.

6. REFERENCES

[1] Albattah, W., Nawaz, M., Javed, A., Masood, M., & Albahli, S. (2021). A novel deep learning method for detection and classification of plant diseases. Complex & Intelligent Systems, 8. https://doi.org/10.1007/s40747-021-00536-1

[2] Karuna, G., Chalamalla, E., Yamsani, B., Daliyet, R., Sharma, H., & Raja, S. M. (2023). An automated system to detect plant disease using deep learning. E3S Web of Conferences, 430. https://doi.org/10.1051/e3sconf/202343001044

[3] Li, L., Zhang, S., & Wang, B. (2021). Plant disease detection and classification by deep learning—A review. IEEE Access, PP, 1-1. https://doi.org/10.1109/ACCESS.2021.3069646

[4] Varshney, D., Babukhanwala, B., Khan, J., Saxena, D., & Singh, A. (2021). Machine learning techniques for plant disease detection. In 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI). https://doi.org/10.1109/ICOEI51242.2021.9453053