# Plant Disease Detection Project

## Introduction

Plant disease detection is an essential application in the agricultural sector. This project utilizes image processing and machine learning techniques to analyze leaf images and determine whether the plant is healthy or diseased.

## Literature Survey

Several approaches exist for plant disease detection, including traditional visual inspections, deep learning models, and image processing techniques. Our project focuses on using OpenCV and machine learning algorithms for analysis.

## Technical Requirements

- Python 3.x  
- Flask  
- OpenCV  
- NumPy  
- skimage  
- Vue.js  
- Bootstrap

## Project Description

The project involves uploading an image of a plant leaf, processing it using OpenCV, and determining its health status based on disease index calculations.

## System Design

1. \*\*Frontend\*\*: Vue.js-based interface for image upload and displaying results.  
2. \*\*Backend\*\*: Flask API for processing images and returning results.  
3. \*\*Processing\*\*: OpenCV and NumPy for image processing.

## Source Code

### Backend (Flask API)

from flask import Flask, request, jsonify  
import cv2  
import numpy as np  
from skimage import filters  
  
app = Flask(\_\_name\_\_)  
  
@app.route('/upload', methods=['POST'])  
def process\_image():  
 file = request.files['image']  
 if not file:  
 return jsonify({'error': 'No file uploaded'})  
  
 image = cv2.imdecode(np.fromstring(file.read(), np.uint8), cv2.IMREAD\_COLOR)  
 gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  
 blurred = cv2.GaussianBlur(gray, (5, 5), 0)  
 edges = cv2.Canny(blurred, 50, 150)  
 ret, thresh = cv2.threshold(blurred, 0, 255, cv2.THRESH\_BINARY + cv2.THRESH\_OTSU)  
 disease\_index = np.mean(thresh) / 255.0  
 status = 'Diseased' if disease\_index < 0.5 else 'Healthy'  
 affected\_area\_percentage = np.mean(thresh) \* 100 / 255  
  
 return jsonify({  
 'affected\_area\_percentage': affected\_area\_percentage,  
 'disease\_index': disease\_index,  
 'num\_diseased\_regions': np.sum(edges > 0),  
 'status': status  
 })  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)