

```

from flask import Flask, request, jsonify
import torch
from torchvision import transforms
from PIL import Image
import io

app = Flask(__name__)

# Load model
model = torch.load("plant_disease_model.pt", map_location=torch.device('cpu'))
model.eval()

# Load classes
with open("classes.txt", "r") as f:
    classes = [line.strip() for line in f.readlines()]

# Prediction route
@app.route('/predict', methods=['POST'])
def predict():
    if 'image' not in request.files:
        return jsonify({'error': 'No image uploaded'}), 400

    image = request.files['image']
    img_bytes = image.read()
    img = Image.open(io.BytesIO(img_bytes)).convert('RGB')

    transform = transforms.Compose([
        transforms.Resize((224, 224)),
        transforms.ToTensor()
    ])
    img_tensor = transform(img).unsqueeze(0)

    with torch.no_grad():
        outputs = model(img_tensor)
        _, predicted = torch.max(outputs.data, 1)
        label = classes[predicted.item()]

    return jsonify({'prediction': label})

```