app.py

```python
from flask import Flask, request, render_template, url_for
from tensorflow.keras.models import load_model
import numpy as np
import cv2
import os
import tensorflow as tf

app = Flask(__name__)

# Load the model
model = load_model('models/imageclassifier.keras')

# Ensure the 'static/uploads' directory exists
UPLOAD_FOLDER = 'static/uploads'
os.makedirs(UPLOAD_FOLDER, exist_ok=True)

# Home route for the file upload form
@app.route('/')
def home():
    return render_template('index.html')  # 'index.html' contains the file upload form

# Predict route to classify the uploaded image
@app.route('/predict', methods=['POST'])
def predict():
    if 'file' not in request.files:
        return render_template('results.html', result='No file part in request', filename=None)

    file = request.files['file']
    if file.filename == '':
        return render_template('results.html', result='No file selected', filename=None)

    # Save the uploaded image to the 'static/uploads' directory
    filepath = os.path.join(UPLOAD_FOLDER, file.filename)
    file.save(filepath)

    # Read the image and preprocess it
    img = cv2.imread(filepath)
    if img is None:
        return render_template('results.html', result='Error reading image', filename=None)

    # Preprocess the image
    img = cv2.resize(img, (256, 256))
    img = img / 255.0  # Normalize
    img = np.expand_dims(img, axis=0)  # Add batch dimension

    # Make prediction
    yhat = model.predict(img)
    label = 'maize' if yhat > 0.5 else 'chilli'
```

```
49
50      # Render result.html with prediction and filename
51      return render_template('results.html', result=label, filename=file.filename)
52
53  if __name__ == '__main__':
54      app.run(debug=True)
55
```