|  |
| --- |
| import numpy as np |
|  | import os |
|  | from PIL import Image |
|  | from flask import Flask, request, render\_template, url\_for |
|  | from werkzeug.utils import secure\_filename, redirect |
|  | from gevent.pywsgi import WSGIServer |
|  | from keras.models import load\_model |
|  | from keras.preprocessing import image |
|  | from flask import send\_from\_directory |
|  |  |
|  | UPLOAD\_FOLDER = 'C:/Users/Dell/PycharmProjects/A-novel-method-for-digit-recognition-system/flask\_app/uploads' |
|  |  |
|  |  |
|  | app = Flask(\_\_name\_\_) |
|  | app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER |
|  |  |
|  | model = load\_model("mnistCNN.h5") |
|  |  |
|  |  |
|  | @app.route('/') |
|  | def index(): |
|  | return render\_template('index.html') |
|  |  |
|  |  |
|  | @app.route('/predict', methods=['GET', 'POST']) |
|  | def upload(): |
|  | if request.method == "POST": |
|  | f = request.files["image"] |
|  | filepath = secure\_filename(f.filename) |
|  | f.save(os.path.join(app.config['UPLOAD\_FOLDER'], filepath)) |
|  |  |
|  | upload\_img = os.path.join(UPLOAD\_FOLDER, filepath) |
|  | img = Image.open(upload\_img).convert("L") # convert image to monochrome |
|  | img = img.resize((28, 28)) # resizing of input image |
|  |  |
|  | im2arr = np.array(img) # converting to image |
|  | im2arr = im2arr.reshape(1, 28, 28, 1) # reshaping according to our requirement |
|  |  |
|  | pred = model.predict(im2arr) |
|  |  |
|  | num = np.argmax(pred, axis=1) # printing our Labels |
|  |  |
|  | return render\_template('predict.html', num=str(num[0])) |
|  |  |
|  |  |
|  | if \_\_name\_\_ == '\_\_main\_\_': |
|  | app.run(debug=True, threaded=False) |