```
# coding=utf-8
import numpy as np
# Keras
from keras.applications.imagenet_utils import preprocess_input
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
# Flask utils
from flask import Flask, url_for, redirect, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer
import os
# Define a flask app
app = Flask(__name__)
# Model saved with Keras model.save()
MODEL_PATH = 'vgg16-0014.hdf5'
# Load your trained model
model = load_model(MODEL_PATH)
def model_predict(img_path, model):
img = image.load_img(img_path, target_size=(224, 224))
img = image.img_to_array(img)
img = np.expand_dims(img, axis=0)
img = preprocess_input(img)
preds = model.predict(img)
return preds
```

```
@app.route('/', methods=['GET'])
def index():
# Main page
return render_template('index.html')
@app.route('/predict', methods=['GET', 'POST'])
def upload():
label = {'cardboard': 0,
'glass': 1,
'metal': 2,
'paper': 3,
 'plastic': 4,
 'trash': 5}
  if request.method == 'POST':
  # Get the file from post request
  f = request.files['file']
  # file_path = secure_filename(f.filename)
 # Save the file to ./uploads
 basepath = os.path.dirname(__file__)
 file_path = os.path.join(
 basepath, 'uploads', secure_filename(f.filename))
 f.save(file_path)
# Make prediction
preds = model_predict(file_path, model)
y_classes = preds.argmax()
```

```
result = str(list(label.keys())[list(label.values()).index(y_classes)])
return result
return None
if __name__ == '__main__':
app.run(debug=True)
```