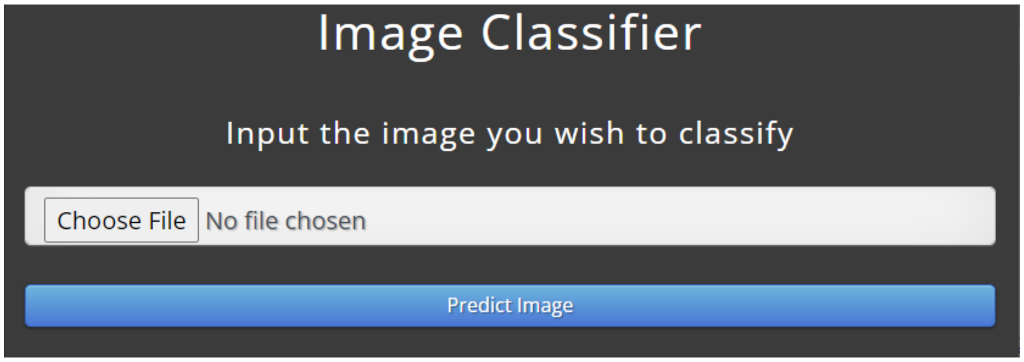
Name: *Jimmy Gaspard Jean*
Internship Batch: *LISUM04*
Submission date: 2021-10-30
Submitted to: DataGlacier-internship/lisum04-w4-Flask at master · jimmygjean/DataGlacier-internship (github.com)

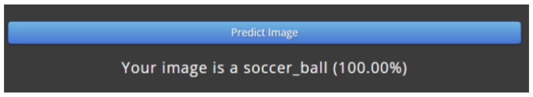
Deployment on Flask

1. The user is provided with an interface where he/she can upload an image.

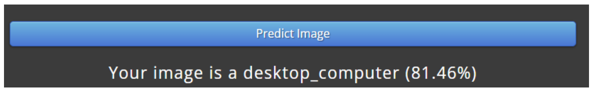


2. The user might for example choose to upload a picture of this car or of this dog:

3. After submission, a deep learning model processes the image and makes a prediction of what the image could be (a MobileNet model was used). The result is then displayed to the user.

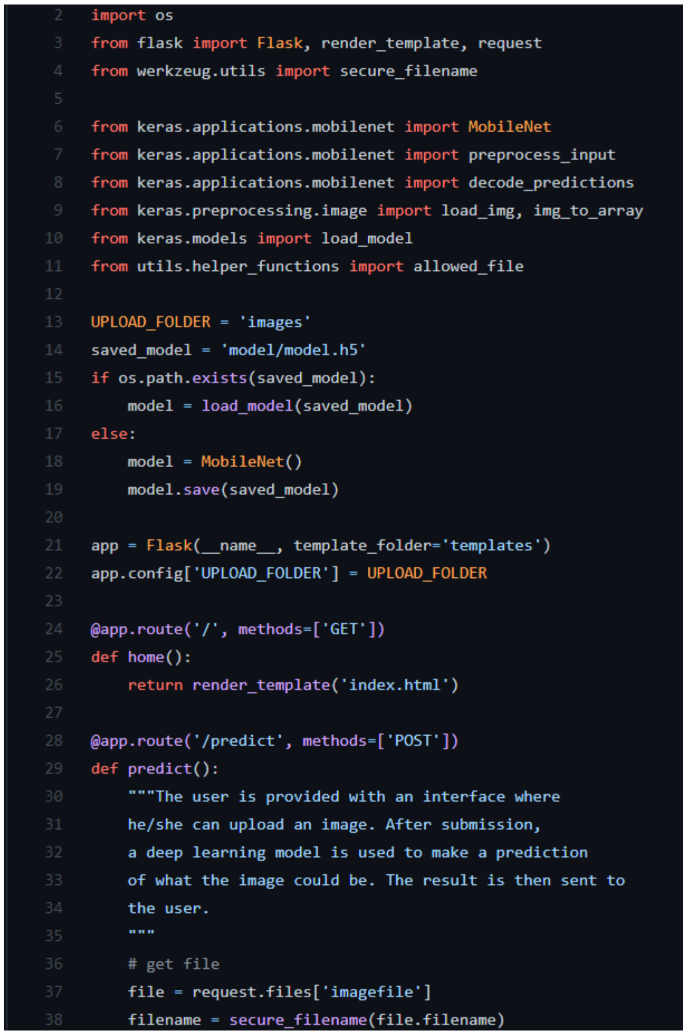


Your image is a soccer_ball (100.00%)

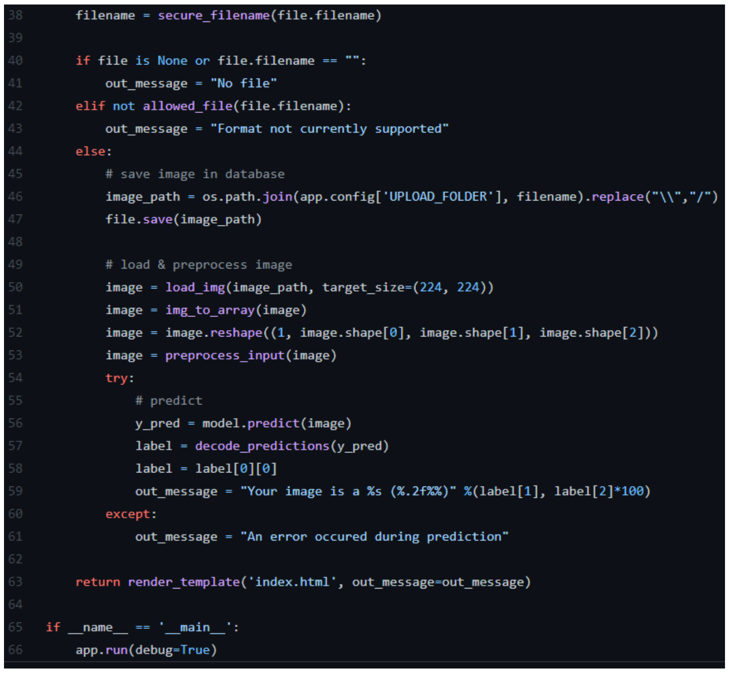


Your image is a desktop_computer (81.46%)

A snapshot of the main code is provided below. The complete code is available on GitHub. Feel free to interact with it.



```python
import os
from flask import Flask, render_template, request
from werkzeug.utils import secure_filename

from keras.applications.mobilenet import MobileNet
from keras.applications.mobilenet import preprocess_input
from keras.applications.mobilenet import decode_predictions
from keras.preprocessing.image import load_img, img_to_array
from keras.models import load_model
from utils.helper_functions import allowed_file

UPLOAD_FOLDER = 'images'
saved_model = 'model/model.h5'
if os.path.exists(saved_model):
    model = load_model(saved_model)
else:
    model = MobileNet()
    model.save(saved_model)

app = Flask(__name__, template_folder='templates')
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

@app.route('/', methods=['GET'])
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    """The user is provided with an interface where
    he/she can upload an image. After submission,
    a deep learning model is used to make a prediction
    of what the image could be. The result is then sent to
    the user.
    """
    # get file
    file = request.files['imagefile']
    filename = secure_filename(file.filename)
```



```python
    filename = secure_filename(file.filename)
```

```python

    if file is None or file.filename == "":
        out_message = "No file"
    elif not allowed_file(file.filename):
        out_message = "Format not currently supported"
    else:
        # save image in database
        image_path = os.path.join(app.config['UPLOAD_FOLDER'], filename).replace("\\","/")
        file.save(image_path)

        # load & preprocess image
        image = load_img(image_path, target_size=(224, 224))
        image = img_to_array(image)
        image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
        image = preprocess_input(image)
        try:
            # predict
            y_pred = model.predict(image)
            label = decode_predictions(y_pred)
            label = label[0][0]
            out_message = "Your image is a %s (%.2f%%)" %(label[1], label[2]*100)
        except:
            out_message = "An error occured during prediction"

    return render_template('index.html', out_message=out_message)

if __name__ == '__main__':
    app.run(debug=True)
```