

Rapport de débogage sur le projet Visio car

Introduction

Le projet Visio car permet de repérer une voiture sur une image. Il est créé avec le micro-framework Flask et du html pour la page web.

Corrections de bugs

Cette application ne contient qu'un fichier python "run.py". Le champ d'investigation est donc bien restreint. Je me suis contenté d'essayer d'interpréter le fichier et de tenter de me rendre sur la route GET "/" pour corriger les bugs au moment où je les ai rencontrés.

Bug #1 :

Le template "home.html" n'existe pas. Il s'agit de "index.html".

```
@app.route('/')
def index():
    return render_template('home.html')
```

Correction :

```
@app.route('/')
def index():
    return render_template('index.html')
```

Bug #2 :

Une branche "if" dans la route "/upload_image" est mal placée. Elle doit être indentée.

```

if request.method == 'POST':

    image = request.files['file']

    if image.filename == '':
        return "Nom de fichier invalide"

    img = Image.open(image)

```

Correction :

```

if request.method == 'POST':

    image = request.files['file']

    if image.filename == '':
        return "Nom de fichier invalide"

    img = Image.open(image)

```

Bug #3 :

Un problème de chargement du modèle keras.

```

c:\engine\base_layer.py", line 3531, in load_own_variables
    raise ValueError(
ValueError: Layer 'block1_conv1' expected 2 variables, but received 0 variables during loading. Expected: ['block1_conv1/kernel:0', 'block1_conv1/bias:0']

```

Correction :

Lancer le projet dans un environnement Linux.

Ce problème était plutôt complexe à résoudre heureusement la solution a été trouvée par la classe.

Bug #4:

Le chemin du modèle est incorrect.

```

File "/home/henry/.local/lib/python3.10/site-packages/keras/src/saving/legacy/save.py", line 234, in load_model
    raise IOError(
OSError: No file or directory found at unet_vgg16_categorical_crossentropy_raw_data.keras

```

Correct:

```
model = tf.keras.models.load_model("models/unet_vgg16_categorical_crossentropy_raw_data.keras", compile=False) # CORRECTION : Wrong path (missing 'models' folder)
```

Bug #5 :

La prédiction n'est pas écrite dans le buffer d'octets.

Vision Car POC

Choisir un fichier

frankfurt_...g8bit.png

Uploader



Image RGB



Prediction

Correction :

J'ai d'abord vérifié la source de l'image : elle était vide.

J'ai ensuite vérifié le contenu du tableau qui était censé être assigné à la source : il était vide.

Dans le code python de la route, on voit que des buffer d'octets sont utilisés pour retourner le contenu.

```
buffered_img = BytesIO()
img.save(buffered_img, format="PNG")
base64_img = base64.b64encode(buffered_img.getvalue()).decode("utf-8")

buffered_mask = BytesIO()
predict_image = Image.fromarray(predict_mask)
predict_image.save(buffered_mask, format="PNG") # CORRECTION : Prediction was not written
base64_mask = base64.b64encode(buffered_mask.getvalue()).decode("utf-8")
```

Le résultat :

Vision Car POC

Choisir un fichier

frankfurt_...g8bit.png

Uploader



Image RGB



Prediction

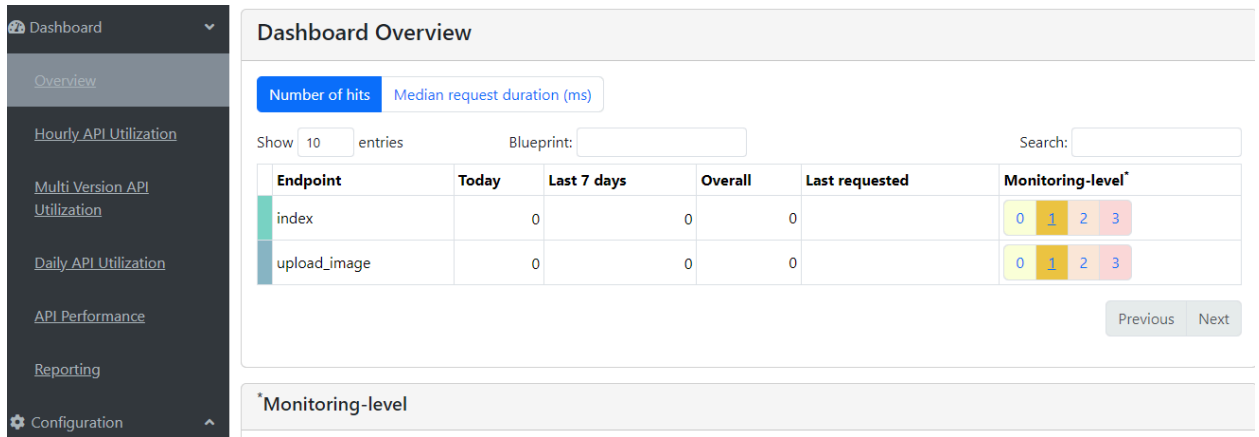
Monitoring

Pour assurer un monitoring de cette petite API, des modifications ont été ajoutées :

- logging

```
2024-08-28 11:45:53,530 : csc[31mcsc[1mWARNING: This is a development server. Do not use it in a production deployment
* Running on http://127.0.0.1:5000
2024-08-28 11:45:53,530 : csc[33mPress CTRL+C to quitcsc[0m
2024-08-28 11:45:53,534 : * Restarting with stat
2024-08-28 11:45:57,809 : * Debugger is active!
2024-08-28 11:45:57,810 : * Debugger PIN: 562-054-038
2024-08-28 11:45:58,324 : 127.0.0.1 - - [28/Aug/2024 11:45:58] "GET / HTTP/1.1" 200 -
2024-08-28 11:45:58,502 : 127.0.0.1 - - [28/Aug/2024 11:45:58] "csc[36mGET /static/style.css HTTP/1.1csc[0m" 304 -
```

- Flask monitoring dashboard



Elles permettront de tracker les erreurs et d'obtenir des statistiques sur l'utilisation de l'API.