

PROJET 8 : PARTICIPEZ A LA CONCEPTION D'UNE VOITURE AUTONOME

Présentation “Projet 8” chez “OPENCLASSROOM”
Jaoid KRAIRI
(Mars 2022)

SOMMAIRE



Introduction,



Ma mission



Approche méthodologique,



Synthèse des résultats,



Démonstration de mon API,



Conclusion



Remerciement.

Introduction:



Future Vision Transport

- ✓ Acquisition des images en temps réel,
- ✓ Traitement des images,
- ✓ **Segmentation des images (c'est moi !)**
- ✓ Système de décision

Ma mission:

1/ Global

DONNEES D'ENTREES

- Des images originaux, des images segmentées et annotées de caméras embarquées.
- Le système d'acquisition n'est pas stable.
- Le volume de donnée sera vite important.

DONNEES DE SORTIES

- Une API simple à utiliser.
- En entrée l'identifiant d'une image et renvoie la segmentation de l'image de l'algo, et de l'image réelle.

Ma mission: 2/ Récapitulatif



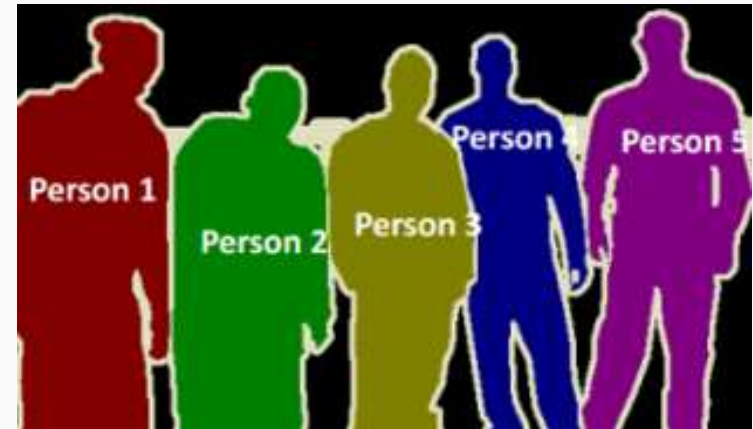
- Entraîner et déployer un modèle de segmentation des images sur les 8 catégories.
- Déployer une API Flask.

Approche méthodologique :

1/ Qu'est-ce que la segmentation d'image ?

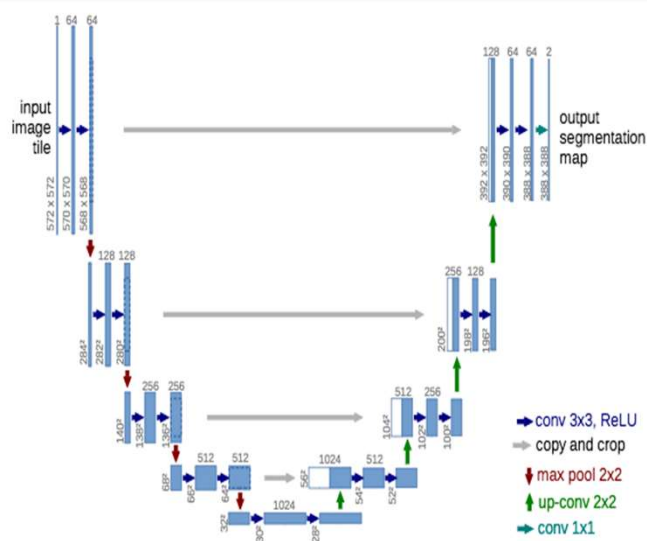


Segmentation Sémantique

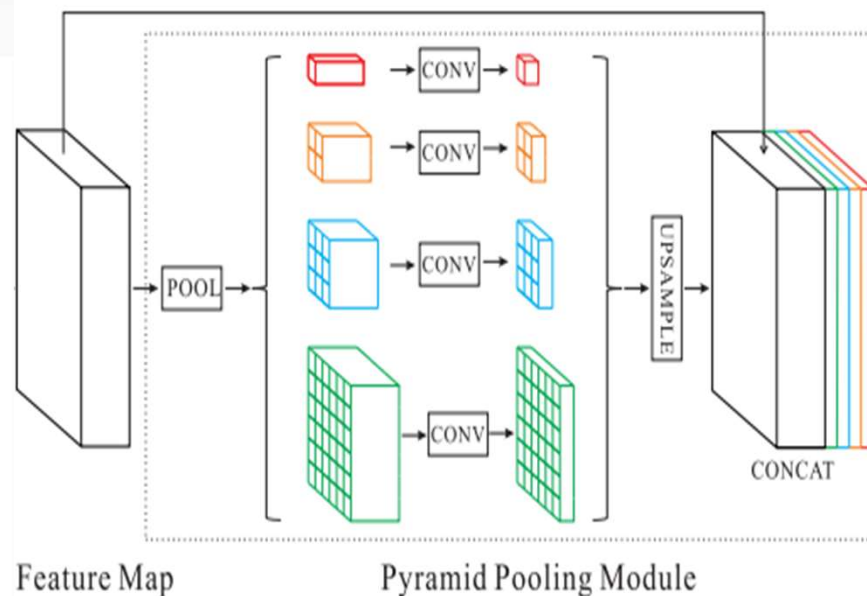


Segmentation d'Instance

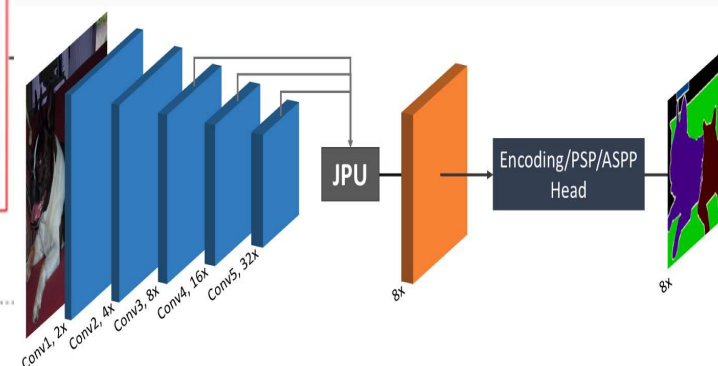
Approche méthodologique : 2/ Les 3 types de segmentation d'image utilisées



Unet



PSPNet

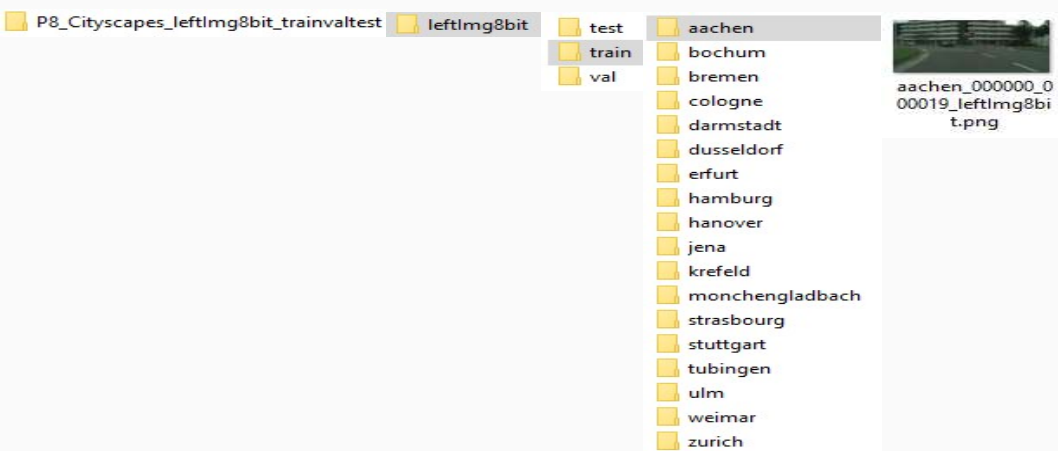


VGG16 FCN8

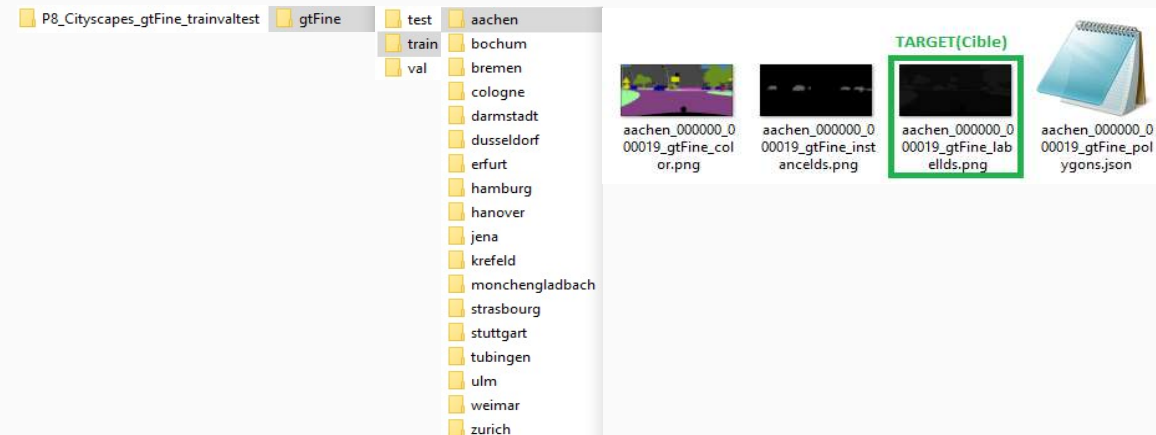
Approche méthodologique :

3/ Préparation des données

Image

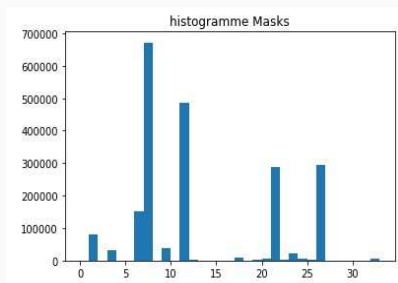


Masques



Approche méthodologique :

4/ Comment faire passer les masques de 35 sous-catégories à 8 catégories principales



```
cats = {  
    'void': [0, 1, 2, 3, 4, 5, 6],  
    'flat': [7, 8, 9, 10],  
    'construction': [11, 12, 13, 14, 15, 16],  
    'object': [17, 18, 19, 20],  
    'nature': [21, 22],  
    'sky': [23],  
    'human': [24, 25],  
    'vehicle': [26, 27, 28, 29, 30, 31, 32, 33, -1]}
```

Identification valeur
pixel et l'associer à
différente catégorie

```
def convertCats(x):  
    if x in cats['void']:  
        return 0  
    elif x in cats['flat']:  
        return 1  
    elif x in cats['construction']:  
        return 2  
    elif x in cats['object']:  
        return 3  
    elif x in cats['nature']:  
        return 4  
    elif x in cats['sky']:  
        return 5  
    elif x in cats['human']:  
        return 6  
    elif x in cats['vehicle']:  
        return 7
```

Fonction permettant
d'attribuer une valeur
de 0 à 7 pour chaque
catégorie

```
convertCats_v = np.vectorize(convertCats)
```

```
def preprocessImg(img):  
    image_matrix = np.expand_dims(img, 2)  
  
    converted_image = convertCats_v(image_matrix)  
    return converted_image
```

Fonction d'appel à la transformation du
masque de 35 sous-catégories à 8
catégories, de plus la fonction cible l'axe
de la matrice du masque à modifier

Valeurs pixels	Catégories
0	void(sol)
1	flat(route)
2	construction
3	object
4	nature
5	sky(ciel)
6	human
7	vehicle

Approche méthodologique :

5/ Générateur de données

- Crée des données par lots et les transmet à mon réseau pour l'entraîner,
- Permettre de se concentrer sur ma conception,
- Keras gère tout le multi-threading (filetage multiple) et la parallélisation,
- Récupérer les lots à l'avance à l'aide de plusieurs cœurs de processeur.

Approche méthodologique :

6/ Etablir les chemins des images et des masques et les charger

```
train_cities = ['aachen', 'bochum', 'bremen', 'cologne', 'darmstadt', 'dusseldorf', 'erfurt', 'hamburg', 'hanover', 'jena', 'krefeld', 'munster', 'paderborn', 'regensburg', 'stuttgart', 'wuppertal']
```

```
train_img_paths = []
train_ann_paths = []

for cities in train_cities:
    train_img_dir = "jk/P8_Cityscapes_leftImg8bit_trainvaltest/leftImg8bit/train/" + cities
    train_ann_dir = "jk/P8_Cityscapes_gtFine_trainvaltest/gtFine/train/" + cities

    train_img_paths = train_img_paths + sorted(
        [
            os.path.join(train_img_dir, fname)
            for fname in os.listdir(train_img_dir)
            if fname.endswith("_leftImg8bit.png")
        ]
    )
    train_ann_paths = train_ann_paths + sorted(
        [
            os.path.join(train_ann_dir, fname)
            for fname in os.listdir(train_ann_dir)
            if fname.endswith("_gtFine_labelIds.png")
        ]
    )

print("Nombre de train images:", len(train_img_paths))
print("Nombre de train annotations:", len(train_ann_paths))

Nombre de train images: 2975
Nombre de train annotations: 2975
```

Approche méthodologique :

7/ Augmentation du nombre d'images et de masque

```
def generateRandomParams(seed):  
    np.random.seed(seed)  
    angle = np.random.randint(26)  
    np.random.seed(seed*2)  
    positive = np.random.randint(2)  
    sigma = np.random.uniform(0, 1)  
  
    if positive == 0:  
        angle = angle * -1  
  
    crop = np.random.randint(3)  
    crop = crop / 10  
  
    #print(angle, crop)  
  
    return angle, crop, sigma
```

```
for mul in range(1, imgaug_multipllier):  
    for i in range(0, self.batch_size):  
  
        angle, crop, sigma = generateRandomParams((1 + i) * mul)  
  
        photo_aug = iaa.Sequential([  
            iaa.Affine(rotate=(angle)),  
            iaa.Crop(percent=(crop)),  
            iaa.GaussianBlur(sigma=(0.0, sigma))  
        ])  
  
        label_aug = iaa.Sequential([  
            iaa.Affine(rotate=(angle)),  
            iaa.Crop(percent=(crop)),  
        ])  
  
        image_aug = photo_aug(image=x[i])  
        x[batch_size * mul + i] = image_aug  
  
        image_aug = label_aug(image=y[i])  
        y[batch_size * mul + i] = image_aug
```

Approche méthodologique :

8/ Réaliser le générateur de données

```
img_size = (128 , 128)
num_classes = 8
batch_size = 2975
imgaug_multiplier = 2
```

```
class Image(Sequence):
    """Itérer sur les données (en tant que matrices Numpy). """

    def __init__(self, batch_size, img_size, input_img_paths, target_img_paths):
        self.batch_size = batch_size
        self.img_size = img_size
        self.input_img_paths = input_img_paths
        self.target_img_paths = target_img_paths

    def __len__(self):
        return len(self.target_img_paths) // self.batch_size

    def __getitem__(self, idx):
        """La ligne de retour (entrée, cible) correspond au batch #idx."""
        i = idx * self.batch_size
        batch_input_img_paths = self.input_img_paths[i : i + self.batch_size]
        batch_target_img_paths = self.target_img_paths[i : i + self.batch_size]

        x = np.zeros((self.batch_size * imgaug_multiplier,) + self.img_size + (3,), dtype="uint8")
        for j, path in enumerate(batch_input_img_paths):
            img = image.load_img(path, target_size=self.img_size)
            x[j] = img
        y = np.zeros((self.batch_size * imgaug_multiplier,) + self.img_size + (1,), dtype="uint8")

        for j, path in enumerate(batch_target_img_paths):
            _img = image.load_img(path, target_size=self.img_size, color_mode="grayscale")
            y[j] = preprocessImg(_img)

        # Augmentation d'image
        for mul in range(1, imgaug_multiplier):
            for i in range(0, self.batch_size):

                angle, crop, sigma = generateRandomParams((1 + i) * mul)

                photo_aug = iaa.Sequential([
                    iaa.Affine(rotate=(angle)),
                    iaa.Crop(percent=(crop)),
                    iaa.GaussianBlur(sigma=(0.0, sigma))
                ])

                label_aug = iaa.Sequential([
                    iaa.Affine(rotate=(angle)),
                    iaa.Crop(percent=(crop)),
                ])

                image_aug = photo_aug(image=x[i])
                x[batch_size * mul + i] = image_aug

                image_aug = label_aug(image=y[i])
                y[batch_size * mul + i] = image_aug

        return x, y
```

Approche méthodologique :

9/ Appeler le générateur de données plus d'autres étapes

```
train_seq = Image(  
    batch_size, img_size, train_img_paths, train_ann_paths  
)
```

1

```
assert train_seq[0][0].shape == (batch_size * imgaug_multiplier, *img_size, 3)  
assert train_seq[0][1].shape == (batch_size * imgaug_multiplier, *img_size, 1)
```

2

```
for x, y in train_seq:  
    break  
x.shape, y.shape
```

3

```
((5950, 128, 128, 3), (5950, 128, 128, 1))
```

```
print("Valeurs max pixels image: ", x.max())
```

4

```
Valeurs max pixels image: 255
```

```
print("Valeurs pixels mask: ", np.unique(y))
```

5

```
Valeurs pixels mask: [0 1 2 3 4 5 6 7]
```

```
image_dataset = x/255.
```

6

```
train_masks_cat = to_categorical(y_train, num_classes=num_classes)  
y_train_cat = train_masks_cat.reshape((y_train.shape[0], y_train.shape[1], y_train.shape[2], num_classes))  
  
test_masks_cat = to_categorical(y_test, num_classes=num_classes)  
y_test_cat = test_masks_cat.reshape((y_test.shape[0], y_test.shape[1], y_test.shape[2], num_classes))  
  
print("Format entrée label entraînement:", y_train_cat.shape)  
print("Format entrée label test:", y_test_cat.shape)  
  
Format entrée label entraînement: (4760, 128, 128, 8)  
Format entrée label test: (1190, 128, 128, 8)
```

8

```
X_train, X_test, y_train, y_test = train_test_split(image_dataset, y, test_size = 0.2, random_state = 42)
```

7

Approche méthodologique :

10/ Création de mon modèles de segmentation(U-net)

```
def conv_block(input, num_filters):
    x = Conv2D(num_filters, 3, padding="same")(input)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)

    x = Conv2D(num_filters, 3, padding="same")(x)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)

    return x

def encoder_block(input, num_filters):
    x = conv_block(input, num_filters)
    p = MaxPool2D((2, 2))(x)
    return x, p

def decoder_block(input, skip_features, num_filters):
    x = Conv2DTranspose(num_filters, (2, 2), strides=2, padding="same")(input)
    x = Concatenate()([x, skip_features])
    x = conv_block(x, num_filters)
    return x

def build_unet(input_shape, n_classes):
    inputs = Input(input_shape)

    s1, p1 = encoder_block(inputs, 64)
    s2, p2 = encoder_block(p1, 128)
    s3, p3 = encoder_block(p2, 256)
    s4, p4 = encoder_block(p3, 512)

    b1 = conv_block(p4, 1024) #Bridge

    d1 = decoder_block(b1, s4, 512)
    d2 = decoder_block(d1, s3, 256)
    d3 = decoder_block(d2, s2, 128)
    d4 = decoder_block(d3, s1, 64)

    if n_classes == 1:
        activation = 'sigmoid'
    else:
        activation = 'softmax'

    outputs = Conv2D(n_classes, 1, padding="same", activation=activation)(d4)
    print(activation)

    model = Model(inputs, outputs, name="U-Net")
    return model

IMG_HEIGHT = X_train.shape[1]
IMG_WIDTH = X_train.shape[2]
IMG_CHANNELS = X_train.shape[3]
input_shape = (IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS)

model = build_unet(input_shape, n_classes=8)
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

1

2

3

4

5

Synthèse des résultats

1/ Résultats après entraînement de mes 4 modèles de segmentations d'images

Modèle	Nbre epochs	Temps d'Entraînement	Perte min(%)	Accuracy max(%)
U-net base	8	11 min et 18 s	34,33%	89,01%
U-net	8	21 min et 22 s	41,28%	86,90%
PSPNet	8	32 min	83,49%	67,23%
VGG16FCN8	8	24 min et 6 s	95,21%	63,02%

Synthèse des résultats

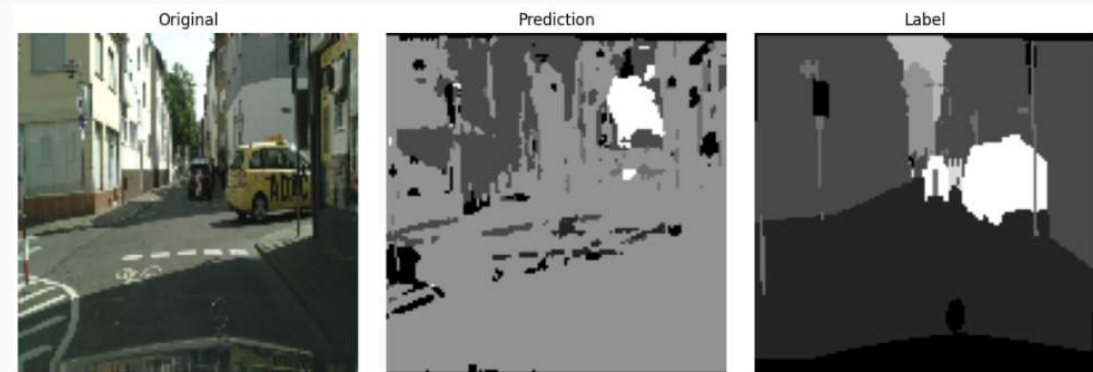
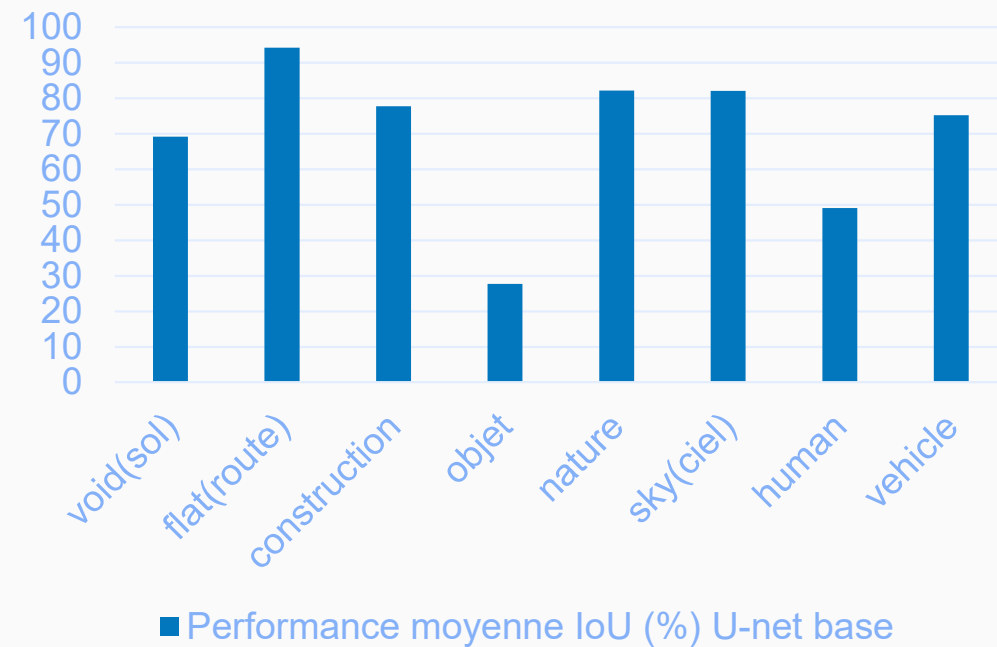
2/ Résultat comparaison U-net base et U-net

Modèle	Nbre epochs	Temps d'Entraînement	Perte min(%)	Accuracy max(%)	Moyenne IoU(%)
U-net base	60	1h 18 min	15,26%	94,72%	69,70%
U-net	60	2h 34 min et 48 s	15,34%	94,35%	65,64%
U-net base plus	100	37 min et 18 s	5,64%	97,90%	76,24%

Synthèse des résultats

3/ Prédiction U-net base sur une image en affichant la métrique moyenne IoU par catégorie sur les données test

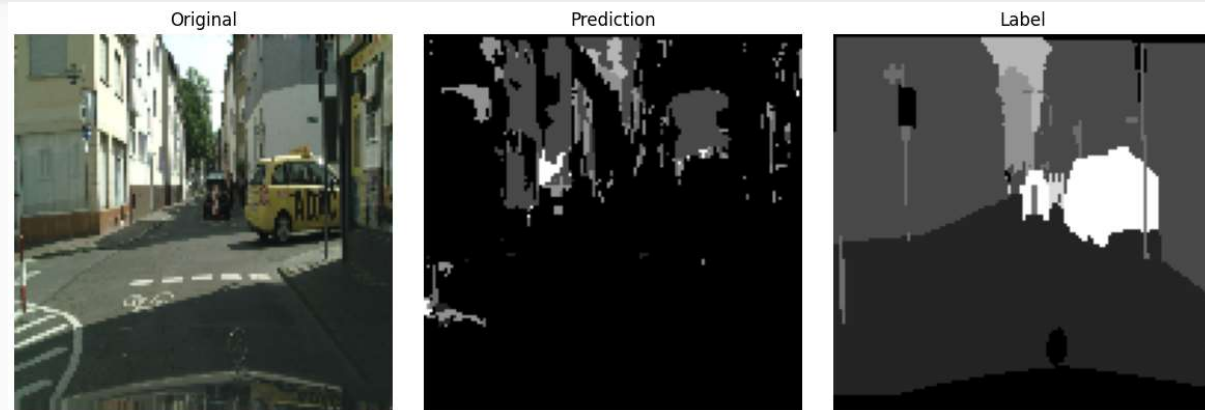
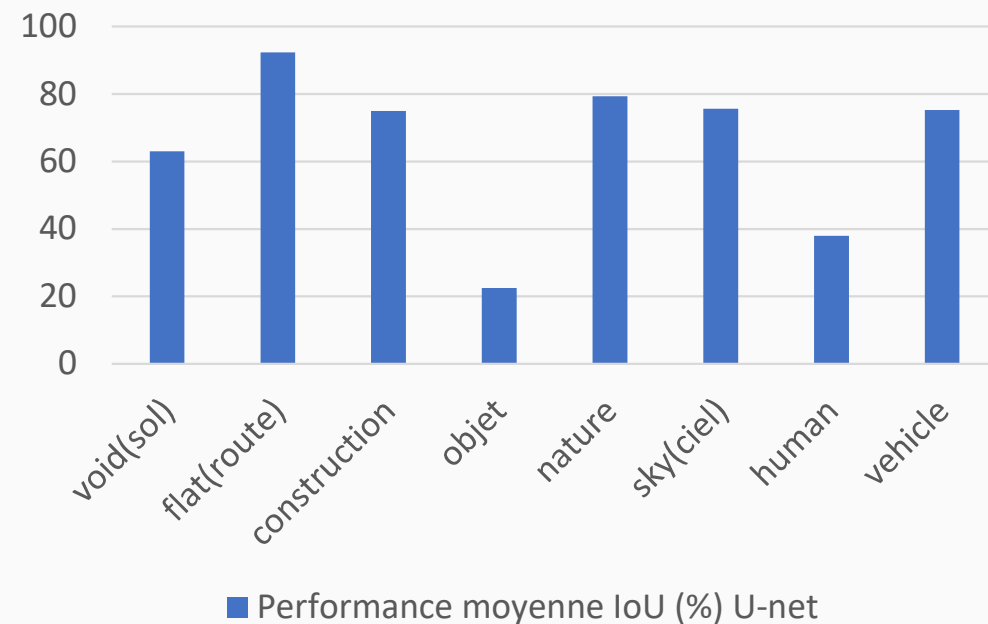
Performance moyenne IoU (%)
U-net base



Synthèse des résultats

4/ Prédiction U-net sur une image en affichant la métrique moyenne IoU par catégorie sur les données test

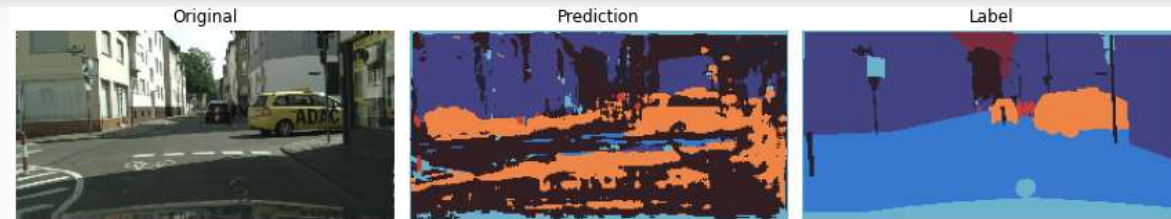
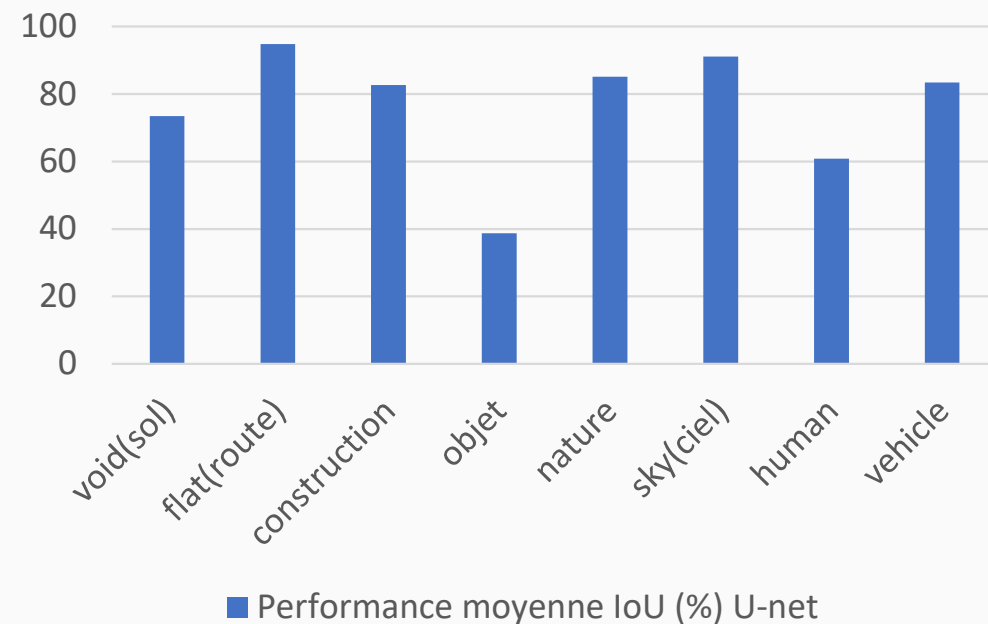
Performance moyenne IoU (%)
U-net



Synthèse des résultats

5/ Prédiction U-net sur une image en affichant la métrique moyenne IoU par catégorie sur les données test

Performance moyenne IoU (%)
U-net base plus



Démonstration de mon API

1/ Enregistrement de mon modèle U-net base plus dans Azure

```
1 from azureml.core import Workspace
2
3
4 ws = Workspace.from_config()
5 print(ws.name, ws.resource_group, ws.location, ws.subscription_id, sep='\n')
```

1

✓

```
projet9
djawed
centralus
cc471f5d-ca48-413b-bb86-0bb26a8001ac
```

```
1 model1 = Model(ws, 'model_base_plus')
```

✓

4

```
1 model1.download(target_dir='.', exist_ok=True)
```

✓

5

```
1 model.save('model_simple_unet_plus.h5')
```

✓

2

```
1 from azureml.core.model import Model
2
3 model_r = Model.register(
4     workspace=ws,
5     model_name='model_base_plus',
6     model_path='model_simple_unet_plus.h5',
7     model_framework=Model.Framework.TENSORFLOW,
8     model_framework_version=tensorflow.__version__)
```

3

```
1 model = load_model('model_simple_unet_plus.h5')
```

✓

6

Démonstration de mon API

2/ Préparation du déploiement du model grâce à une API Flask

7 lignes de code

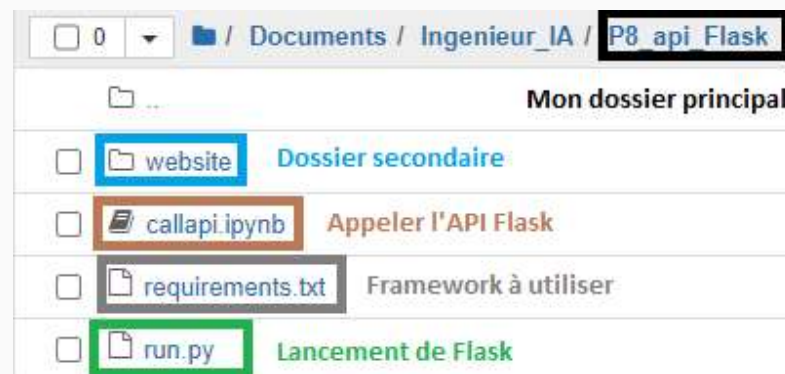
```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return "Hello world !"

if __name__ == "__main__":
    app.run()
```

Dossier principal



Dossier secondaire



Démonstration de mon API

3/ Transfert de mon projet sur GitHub

Créer un nouveau référentiel

Un référentiel contient tous les fichiers du projet, y compris l'historique des révisions. Vous avez déjà un dépôt de projet ailleurs ? [Importer un référentiel](#).

Propriétaire * / Nom du référentiel *

Krairi / p8_api_simple ✓

Les grands noms de l'industrie ? Your new repository will be created as p8_api_simple. Inspiration ? Que diriez-vous de fiction-doodle ?

Descriptif (facultatif)

Segmentation image

☒ Publique
N'importe qui sur Internet peut voir ce référentiel. Vous choisissez qui peut s'engager.

☐ Privé
Vous choisissez qui peut voir et s'engager dans ce référentiel.

Initialisez ce référentiel avec :
Ignorez cette étape si vous importez un référentiel existant.

☐ Ajouter un fichier README
C'est ici que vous pouvez écrire une longue description de votre projet. [Apprendre encore plus.](#)

☐ Ajouter .gitignore
Choisissez les fichiers à ne pas suivre dans une liste de modèles. [Apprendre encore plus.](#)

☐ Choisissez une licence
Une licence indique aux autres ce qu'ils peuvent et ne peuvent pas faire avec votre code. [Apprendre encore plus.](#)

Accordez à vos applications Marketplace l'accès à ce référentiel
Vous êtes abonné à 1 application Marketplace

☒ Pipelines Azure
Créez, testez et déployez en continu sur n'importe quelle plate-forme et cloud

Vous créez un référentiel public dans votre compte personnel.

Créer un référentiel

```
git init
git add .
git commit -m « initialisation »
```

```
git remote add origin https://github.com/Krairi/p8_api_-simple.git
git push origin master
```

Search or jump to... / Pull requests Issues Marketplace Explore

Krairi / p8_api_simple Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

Krairi	Add or update the Azure App Service build and deployment workflow config	e9bcd28 15 hours ago	5 commits
.github/workflows	Add or update the Azure App Service build and deployment workflow c...	15 hours ago	
.ipynb_checkpoints	initialisation	19 hours ago	
__pycache__	initialisation	19 hours ago	
website	Update views.py	15 hours ago	
callapi.ipynb	initialisation	19 hours ago	
config.py	initialisation	19 hours ago	
requirements.txt	Update requirements.txt	19 hours ago	
run.py	initialisation	19 hours ago	

Help people interested in this repository understand your project by adding a README. Add a README

Démonstration de mon API

4/ Déploiement en production de mon api Flask dans Azure avec web app

Créer une application web

App Service Web Apps vous permet de créer, de déployer et de mettre à l'échelle des applications d'API, web et mobiles de classe Entreprise exécutées sur n'importe quelle plateforme rapidement. Respectez les exigences strictes en termes de performances, de scalabilité, de sécurité et de conformité lors de l'utilisation d'une plateforme complètement gérée pour effectuer la maintenance de l'infrastructure. [En savoir plus](#)

Détails du projet

Sélectionnez un abonnement pour gérer les coûts et les ressources déployées. Utilisez les groupes de ressources comme des dossiers pour organiser et gérer toutes vos ressources.

Abonnement *
Groupe de ressources * [Créer nouveau](#)

Détails de l'instance

Vous avez besoin d'une base de données ? [Essayez la nouvelle expérience web + base de données](#)

Nom * [.azurewebsites.net](#)

Publier * ☒ Code ☐ Conteneur Docker ☐ Application web statique

Pile d'exécution *

Système d'exploitation * ☒ Linux ☐ Windows

Région *
Vous ne trouvez pas votre plan App Service ? Essayez une autre région ou sélectionnez votre environnement App Service Environment.

Plan App Service

Le niveau tarifaire du plan App Service détermine l'emplacement, les fonctionnalités, le coût et les ressources de calcul associées à votre application. [En savoir plus](#)

Plan Linux (Central US) * [Créer](#)

Référence et taille * **De base B1**
100 ACU au total, 1.75 Go de mémoire

Redondance de zone

Vous pouvez déployer un plan App Service en tant que service redondant interzone dans des régions le prenant en charge. Vous devez décider de cela au moment du déploiement car vous ne pouvez plus rendre une zone plan App Service redondante après son déploiement. [En savoir plus](#)

Redondance de zone ☐ Activé : Votre plan App Service et les applications qu'il contient seront redondants interzone. Il y aura au minimum trois instances de plan App

[Vérifier > créer](#)[< Précédent](#)[Suivant : Déploiement >](#)

P8jkAppFlask | Centre de déploiement

Rechercher (Ctrl+F)

[Vue d'ensemble](#)

[Journal d'activité](#)

[Contrôle d'accès \(IAM\)](#)

[Étiquettes](#)

[Diagnostic et résoudre les problèmes](#)

[Sécurité](#)

[Événements \(préversion\)](#)

Déploiement

[Démarrage rapide](#)

[Emplacements de déploiement](#)

[Centre de déploiement](#)

Paramètres

[Configuration](#)

[Authentification](#)

[Application Insights](#)

[Identité](#)

[Sauvegardes](#)

[Domaines personnalisés](#)

[Paramètres TLS/SSL](#)

[Paramètres TLS/SSL \(préversion\)](#)

[Réseau](#)

[Enregistrer](#) [Abandonner](#) [Parcourir](#) [Gérer le profil de publication](#) [Synchroniser](#) [Laisser un commentaire](#)

Paramètres

[Journaux](#)

[Informations d'identification FTPS](#)

Vous êtes maintenant dans l'emplacement de production, ce qui n'est pas recommandé pour la configuration de CI/CD. En savoir plus

Déployez et générez du code à partir de votre fournisseur de build et source par défaut. En savoir plus

Source *

Génération avec GitHub Actions. Changez le fournisseur.

GitHub

App Service placera un flux de travail GitHub Actions dans le dépôt choisi pour générer et déployer votre application chaque fois qu'une validation est effectuée sur la branche choisie. Si vous ne trouvez pas d'organisation ou de dépôt, vous devrez peut-être activer des autorisations supplémentaires sur GitHub. En savoir plus

Connecté comme [Krairi](#) [Changer de compte](#)

Organisation *

Dépôt *

Branche *

Option de workflow * ☒ Ajouter un workflow : ajouter un nouveau fichier de workflow 'master_P8jkAppFlask.yml' dans le dépôt et la branche sélectionnés.

☐ Utiliser le workflow disponible : utiliser l'un des fichiers de workflow disponibles dans le dépôt et la branche sélectionnés.

Build

Pile d'exécution

P8jkAppFlask | Centre de déploiement

Rechercher (Ctrl+F)

[Vue d'ensemble](#)

[Journal d'activité](#)

[Contrôle d'accès \(IAM\)](#)

[Étiquettes](#)

[Diagnostic et résoudre les problèmes](#)

[Sécurité](#)

[Événements \(préversion\)](#)

Déploiement

[Paramètres](#)

[Journaux](#)

[Informations d'identification FTPS](#)

[Actualiser](#)

[Heure](#)

[ID de validat...](#)

[Journaux](#)

[Auteur de la validation](#)

[Statut](#)

[Message](#)

[Sunday, April 3, 2022 \(1\)](#)

04/3/2022, 3:52:06 AM +0200

3a614c

Journaux d'application

N/A

Opération réussie (Actif)

["type":"deployment","sha

Krairi / p8_api_simple

[Public](#)

[Code](#)

[Issues](#)

[Pull requests](#)

[Actions](#)

[Projects](#)

[Wiki](#)

[Security](#)

[Insights](#)

[Settings](#)

Summary

[Jobs](#)

[build](#)

[deploy](#)

[Triggered via push 17 minutes ago](#)

[Krairi pushed](#)

[e96cd20](#)

[master](#)

[Status](#)

[Success](#)

[Total duration](#)

[12m 7s](#)

[Artifacts](#)

[1](#)

[master_p8jkappflask.yml](#)

[on push](#)

[build](#)

[54s](#)

[deploy](#)

[10m 55s](#)

[http://p8jkappflask.azurewebsites.net](#)

Démonstration de mon API

5/ Exemple de fonctionnement de mon API Flask en faisant appel au fichier callapi

Fichier callapi

Projet 8 : Appeler l'API Flask pour segmentation des images

Importation de la librairie

```
1 import requests
2 from tensorflow import keras
3 import json
```

Indiquer le chemin d'une image

```
1 image_path = './website/static/test_img_3.png'
```

Lors du lancement de flask un URL port 5000 a été créé pour la page d'accueil

1- Générer le contenu revoyé par la vue URL port 5000 plus predict

2-Ouvrir l'image

3-Effectuer une requête pour poster l'image

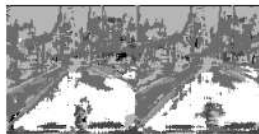
```
1 url = 'http://127.0.0.1:5000/predict'
2 my_img = {'image': open(image_path, 'rb')}
3 r = requests.post(url, files=my_img)
```

4-Charger la prédiction au format Json

```
1 data = json.loads(r.json()['data'])
```

5-Afficher la prédiction du masque

```
1 keras.preprocessing.image.array_to_img(
2     data, data_format=None, scale=True, dtype=None,
3 )
```



Terminal

```
azureuser@krairijl:~/cloudfiles/codes/Users/Krairij/p8_api Flask$ python run.py
2022-03-22 05:27:06.681184: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] failed
detected
2022-03-22 05:27:06.681254: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] ke
1): /proc/driver/nvidia/version does not exist
2022-03-22 05:27:06.699553: I tensorflow/core/platform/cpu_feature_guard.cc:151] This Ten
rary (oneDNN) to use the following CPU instructions in performance-critical operations:
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flag
* Serving Flask app "website.views" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
2022-03-22 05:27:24.054423: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] failed
detected
2022-03-22 05:27:24.054490: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] ke
1): /proc/driver/nvidia/version does not exist
2022-03-22 05:27:24.055080: I tensorflow/core/platform/cpu_feature_guard.cc:151] This Ten
rary (oneDNN) to use the following CPU instructions in performance-critical operations:
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flag
* Debugger is active!
* Debugger PIN: 696-188-572
127.0.0.1 - - [22/Mar/2022 05:27:40] "POST /predict HTTP/1.1" 200 -
```

Conclusion

- ✓ Modèle U-net sans augmentation d'image,
- ✓ Bonnes performance et rapidité d'exécution,
- ✓ Certaines catégories étaient difficiles à détecter,
- ✓ Utiliser une résolution plus importante

REMERCIEMENT

Merci de m'avoir écouté

REPONDRE AUX QUESTIONS