Future Vision Transport

Conception d'une voiture autonome

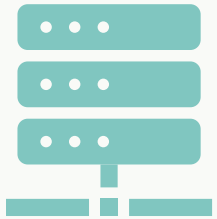
Segmentation sémantique

CITYSCAPES DATASET

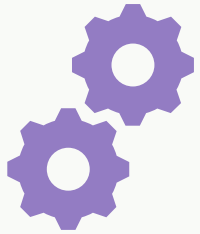
Sommaire

Contexte

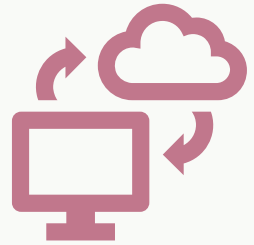
Récupération des masques



Générateur de données



Modélisation



Azure

Future Vision Transport

Contexte

Future Vision Transport

Contexte

Conception d’une voiture autonome

Segmentation des images

Jeu de données CityScapes

Future Vision Transport

CityScapes Dataset

5000 images

34 classes

8 classes finales

Démarche

Récupération des masques (CityScapes-dataset)

Générateur de données

Modélisation

Entrainement / Déploiement Azure (MLOps)

Future Vision Transport

CITYSCAPES DATASET

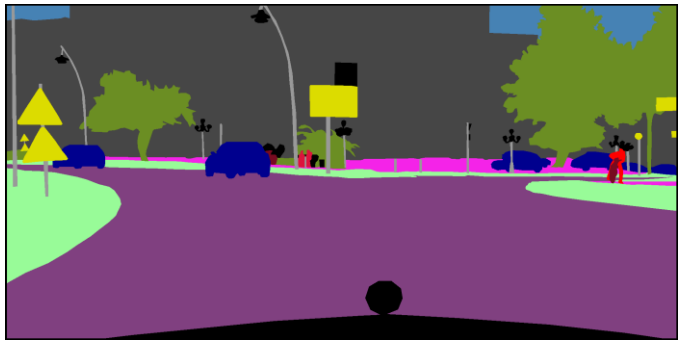
Récupération des masques

Future Vision Transport

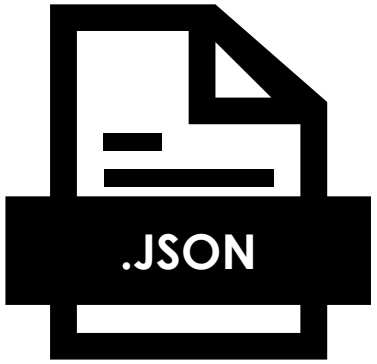
CITYSCAPES DATASET

Masques du jeu de données

Nuance gris (1channel) => 34 classes



RGB (3 channel) => 34 classes



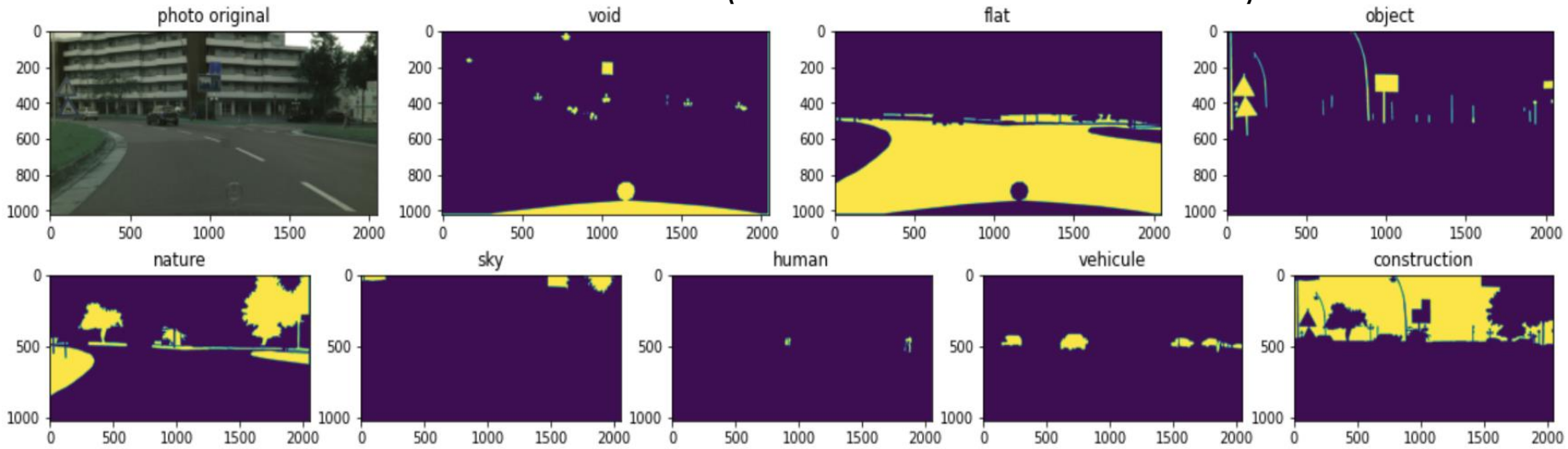
Polygone => 34 classes

Future Vision Transport

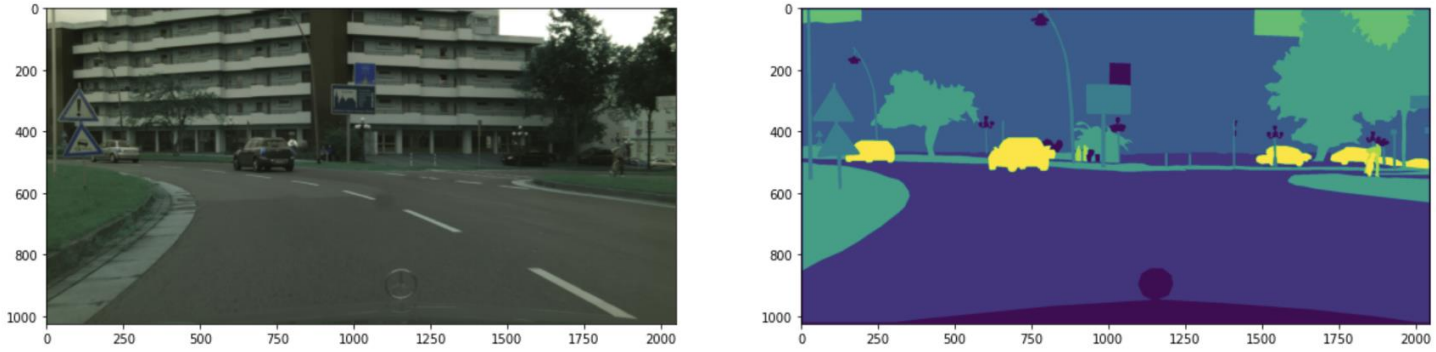
CITYSCAPES DATASET

Masque final (8 classes)

One hot encoder (8 channel 2 classes => 0/1)



Label encoder (1 channel 8 classes)



Future Vision Transport

CITYSCAPES DATASET

Benchmark des différentes solutions

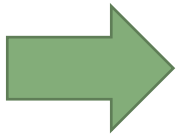
	label_ids	json_file	gain_seconde_labelid_vs_json
label_encoder	0.1946	0.8548	0.6602
one_hot_encoder	0.1876	4.5758	4.3882

Générateur de données

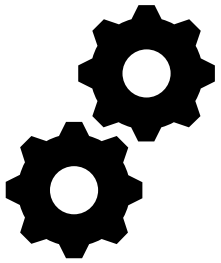
Générateur de données

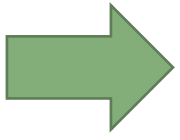
./data/inputs/xxxx.png
...
./data/inputs/xxxx.png

./data/outputs/xxxx.png
...
./data/outputs/xxxx.png

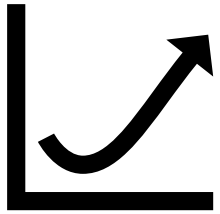


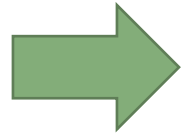
Générateur



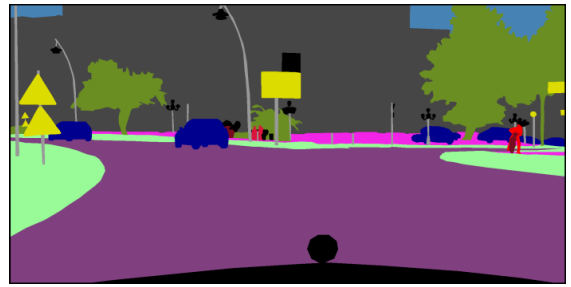


Modèle





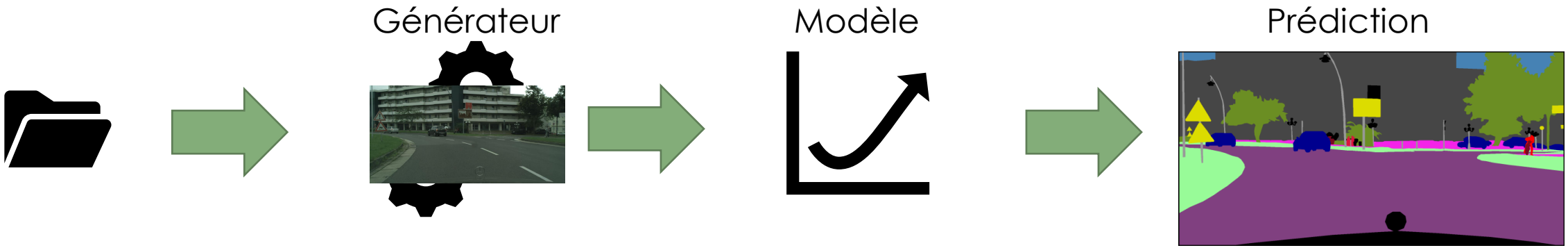
Prédiction



Future Vision Transport

CITYSCAPES DATASET

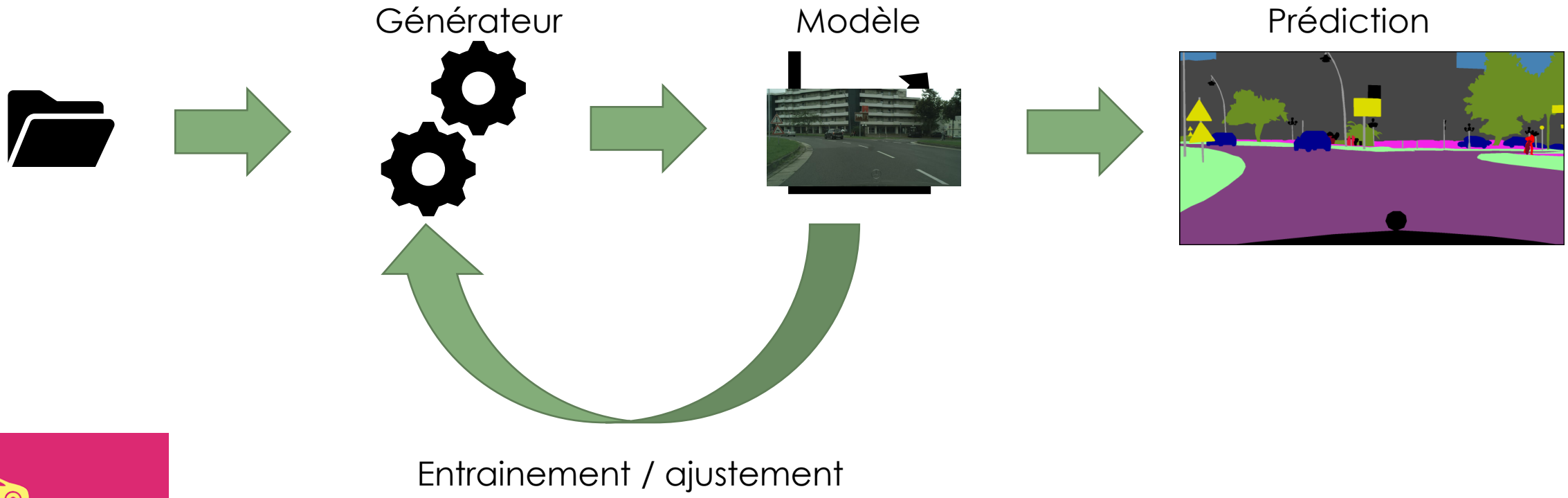
Générateur de données



Future Vision Transport

CITYSCAPES DATASET

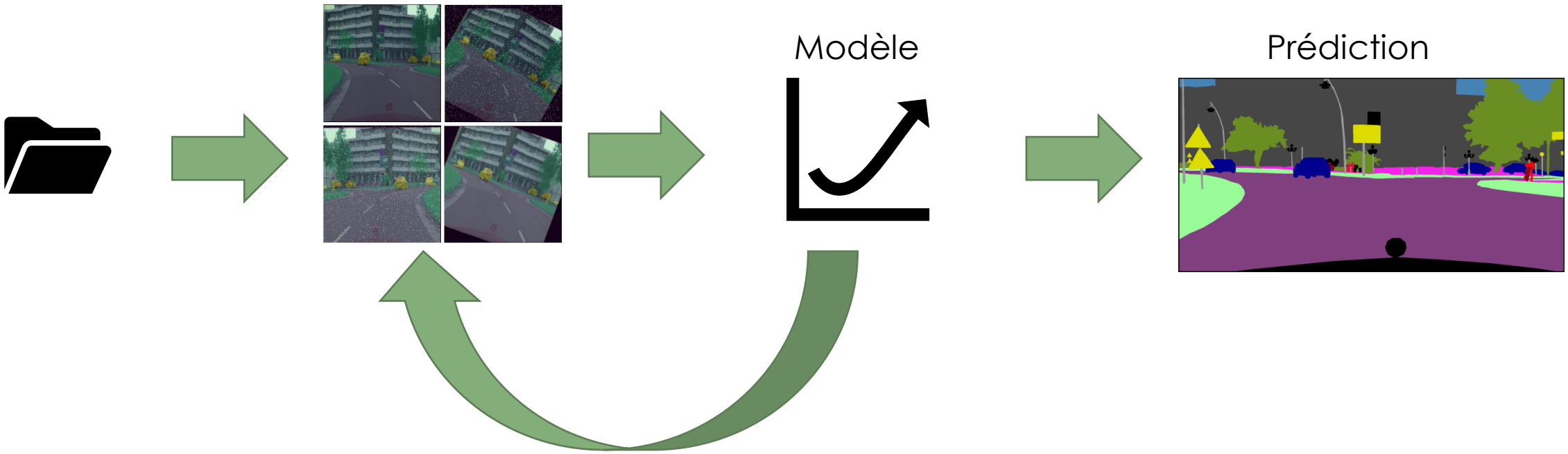
Générateur de données



Future Vision Transport

CITYSCAPES DATASET

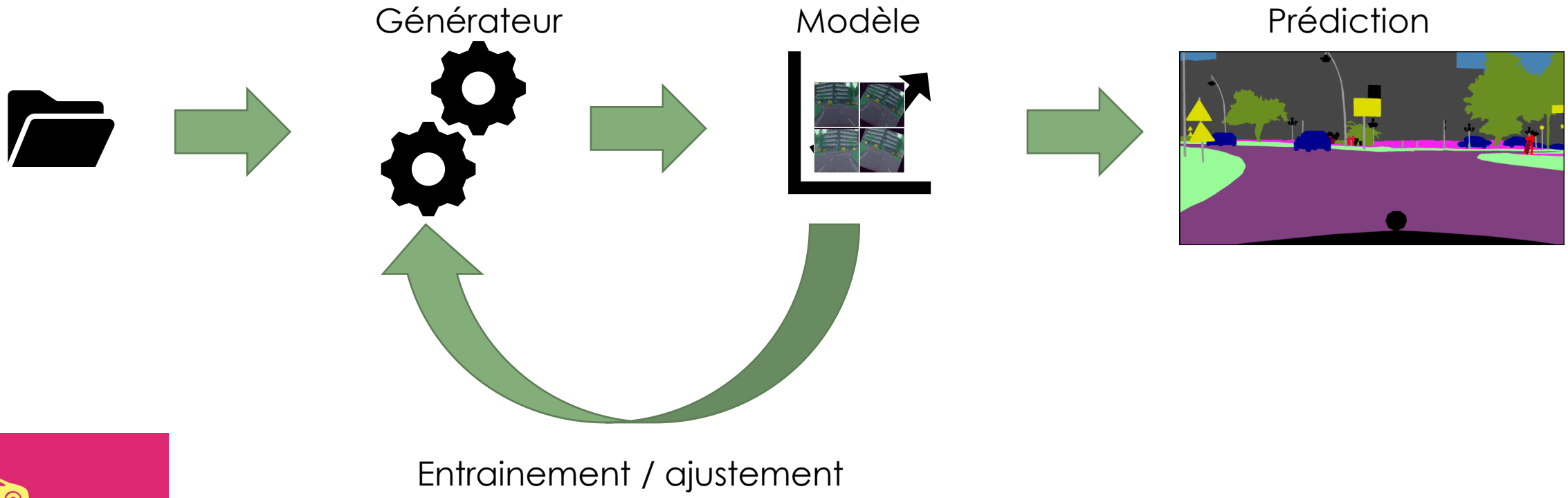
Générateur de données (augmentation données)



Future Vision Transport

CITYSCAPES DATASET

Générateur de données (augmentation données)



Future Vision Transport

CITYSCAPES DATASET

Modélisation

Future Vision Transport

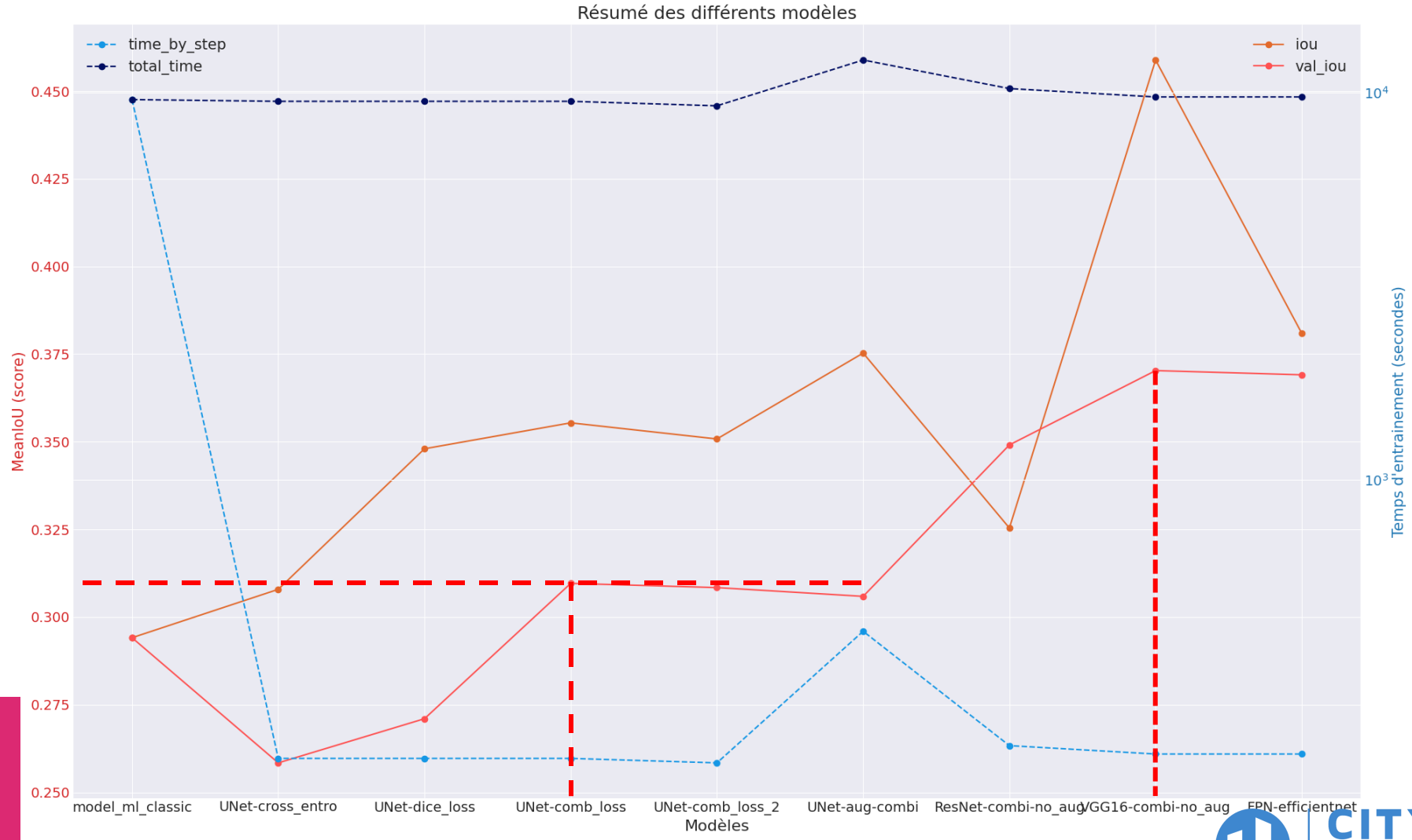
Différentes modélisations réalisé

- Machine learning (modélisation linéaire)
- UNet
- UNet avec Augmentation des données
- Transfert Learning Unet (ResNet50 / VGG16)
- FPN transfert learning (EfficientNetb3)

Future Vision Transport

CITYSCAPES DATASET

Synthèse des modélisations



Future Vision Transport

CITYSCAPES DATASET

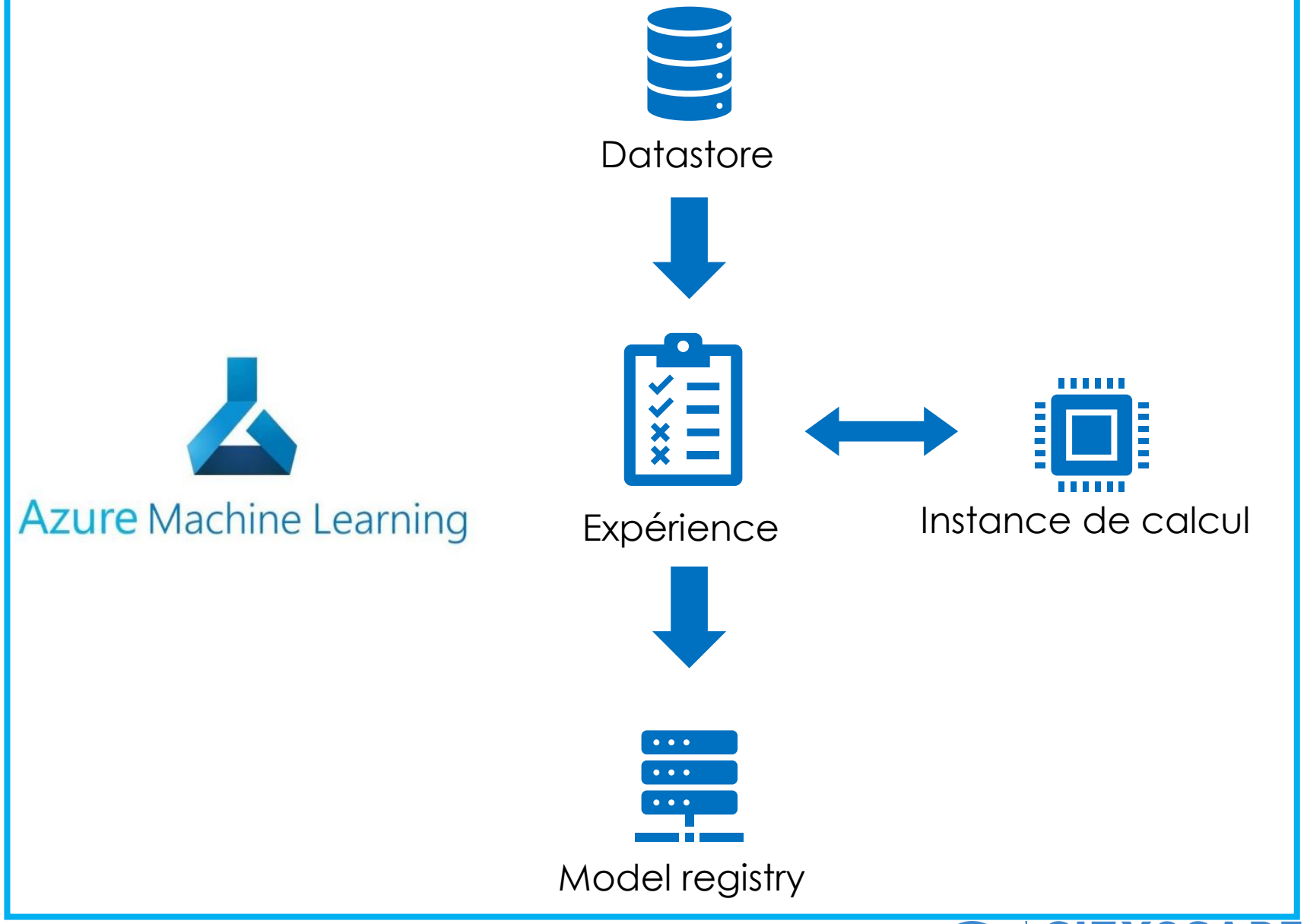
Azure

Future Vision Transport

CITYSCAPES DATASET

Entrainement

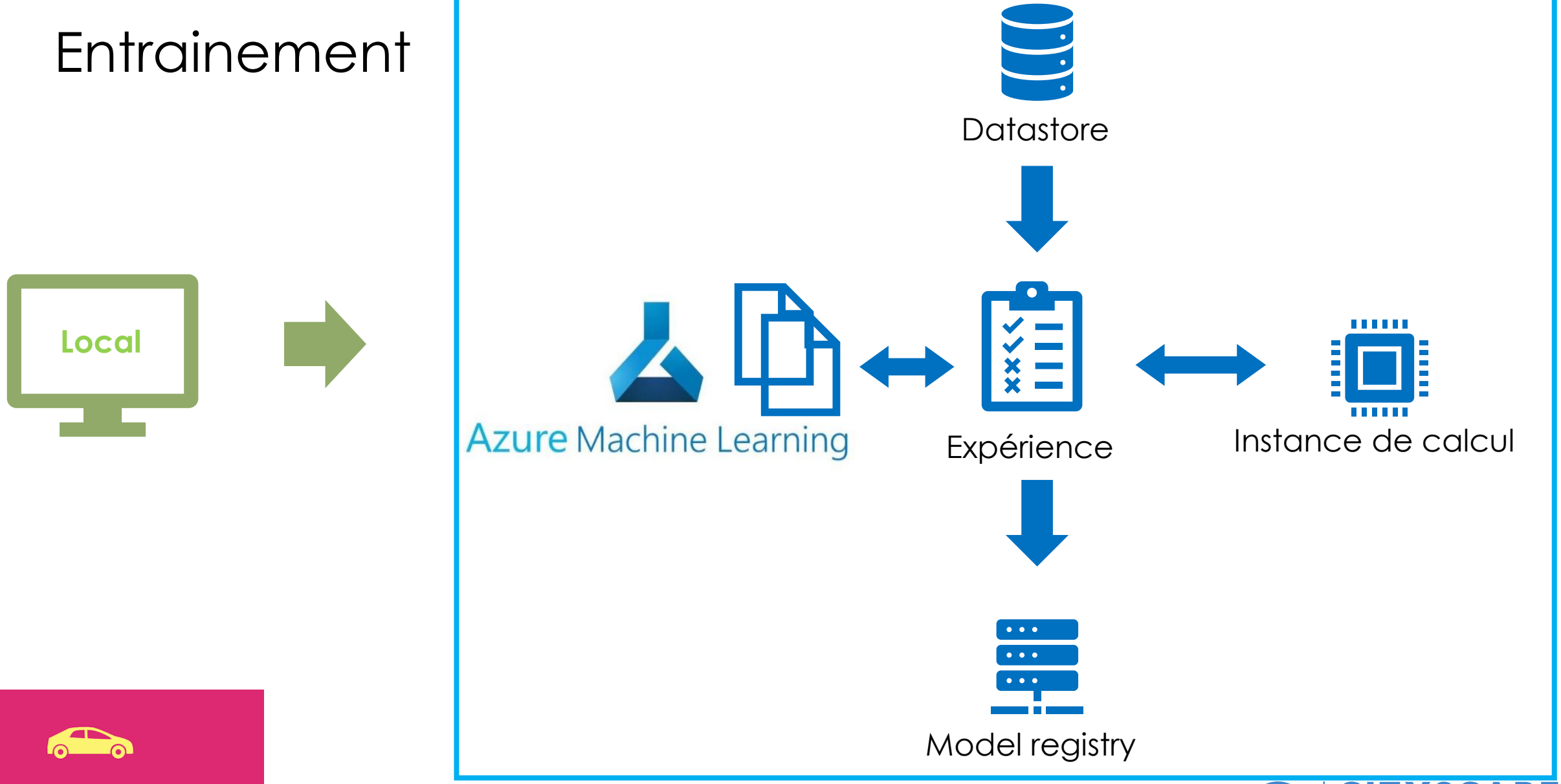
Script d'entrainement



Future Vision Transport

CITYSCAPES DATASET

Entrainement



Future Vision Transport

CITYSCAPES DATASET

Entrainement

```
from azureml.core import Workspace, Experiment, Environment, ScriptRunConfig

experiment = Experiment(workspace=ws, name='experiment_train_tensorflow')

config = ScriptRunConfig(source_directory='./src',
                         script='train.py',
                         compute_target='train-model-fvt')

env_tf = ws.environments['AzureML-tensorflow-2.4-ubuntu18.04-py37-cuda11-gpu']

config.run_config.environment = env_tf

run = experiment.submit(config)

run.wait_for_completion(show_output=True)
```

Déploiement (ACI)

```
from azureml.core.environment import Environment
from azureml.core.model import InferenceConfig, Model
from azureml.core.webservice import AciWebservice, Webservice

env_tf = ws.environments['AzureML-tensorflow-2.4-ubuntu18.04-py37-cpu-inference']


inference_config = InferenceConfig(source_directory=source_directory,
                                   entry_script="x/y/score.py",
                                   environment=env_tf)

deployment_config = AciWebservice.deploy_configuration(cpu_cores = 2, memory_gb = 4,auth_enabled=True)

service = Model.deploy(
    workspace = ws,
    name = "semantic-segmentation",
    models = [final_model],
    inference_config = inference_config,
    deployment_config = deployment_config)

service.wait_for_deployment(show_output = True)
```

Future Vision Transport

CITYSCAPES DATASET

Déploiement (ACI)

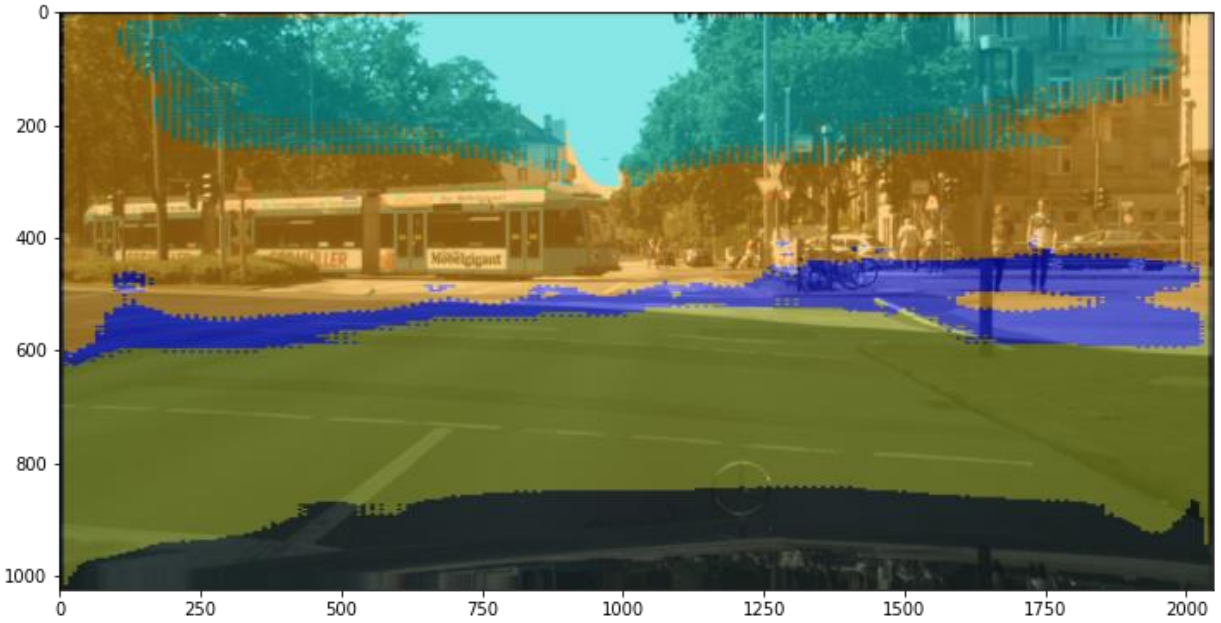
```
input_payload = json.dumps({
    'id':1
})

output = service.run(input_payload)
output = np.array(output)

============ OUTPUT ==============
array([[[ 59,  60,  57],
        [ 57,  59,  55],
        [ 55,  57,  53],
        ...,
        [ 38,  43,  40],
        [ 38,  43,  40],
        [ 38,  43,  40]]])
```

Future Vision Transport

CITYSCAPES DATASET

Déploiement (ACI)

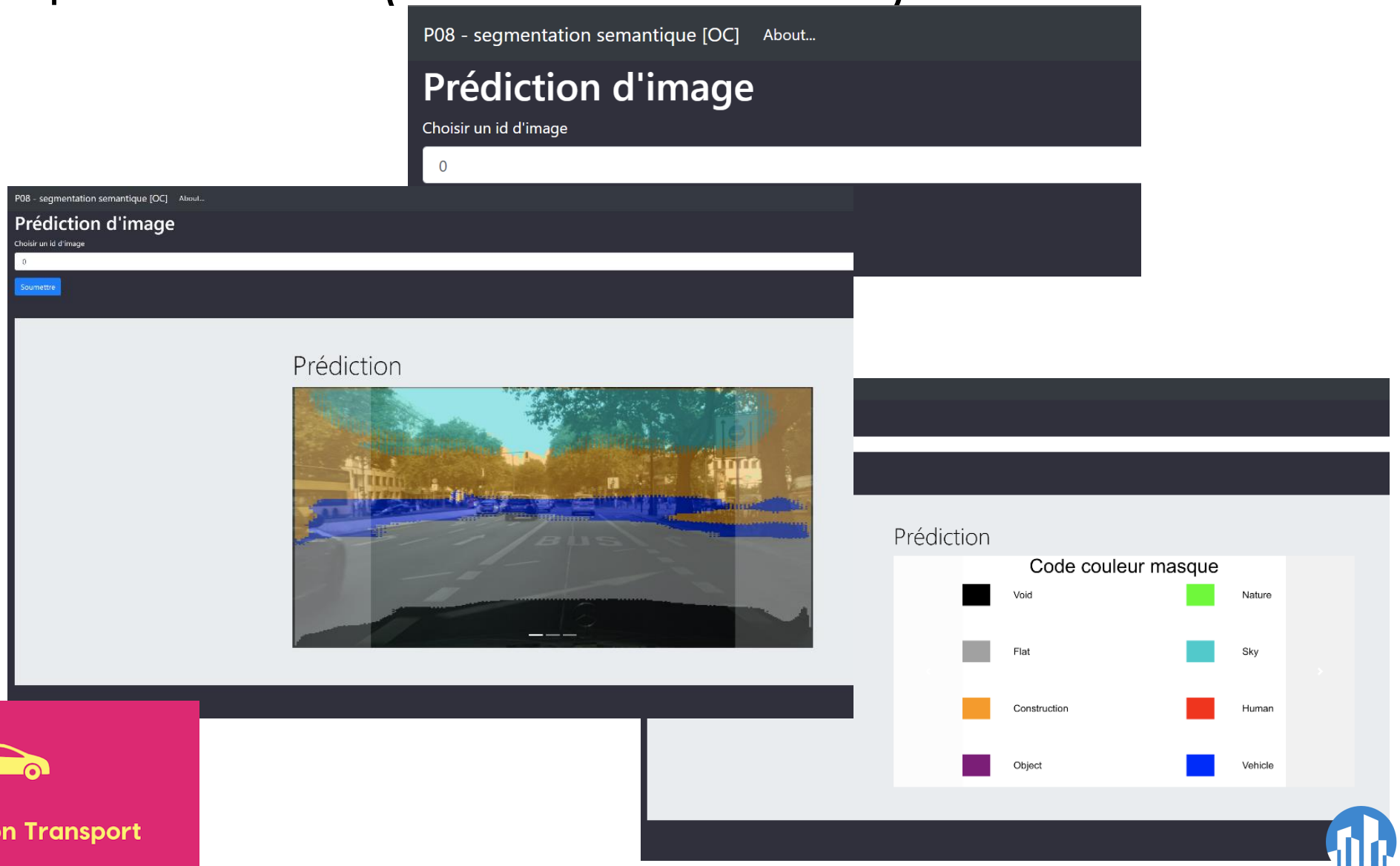
```
plt.figure(figsize=(12,10))
plt.imshow(output)
```



Déploiement (Web APP / Flask)

Azure Web App

Déploiement (Web APP / Flask)



Future Vision Transport

CITYSCAPES DATASET

Démonstration Flask

Future Vision Transport

CITYSCAPES DATASET

Conclusion

Future Vision Transport

CITYSCAPES DATASET

Conclusion

Performance du modèle

Optimisation des paramètres
Augmenter la taille des images
Augmenter le nombre d'images initial
Modification du transformateur imgaug

MLOps à automatiser

Mise a jour Datastore
Déploiement / entrainement

Merci de votre attention, avez-vous des questions ?