

SEGMENTATION D'IMAGES



Future Vision Transport

Contexte

Systèmes embarqués de vision par ordinateur pour les véhicules autonomes

1. Acquisition des images en temps réel
2. Traitement des images
- 3. Segmentation des images**
4. Système de décision

Objectifs

1. Concevoir un premier modèle de segmentation d'images qui devra s'intégrer facilement dans la chaîne complète du système embarqué.
2. API Flask déployée grâce au service Azure qui recevra en entrée l'identifiant d'une image et retournera l'image avec les segments identifiés par votre modèle et l'image avec les segments identifiés annotés dans le jeu de données

Jeux de données

- 2750 images issues du jeux de données CityScape pour l'entraînement et 500 images pour les tests
- Utilisation de 8 catégories

Group	Classes
flat	road, sidewalk, parking, rail track
human	person, rider
vehicle	car, truck, bus, on rails, motorcycle, bicycle, caravan, trailer
construction	building, wall, fence, guard rail, bridge, tunnel
object	pole, pole group, traffic sign, traffic light
nature	vegetation, terrain
sky	sky
void	ground, dynamic, static

- Palette de couleurs



- Réduction des images en 128x256 pixels

Calcul de la perte

Entropie croisée catégorielle (Categorical Cross Entropy)

Mesure la similarité entre les données prédites et les données vraies.

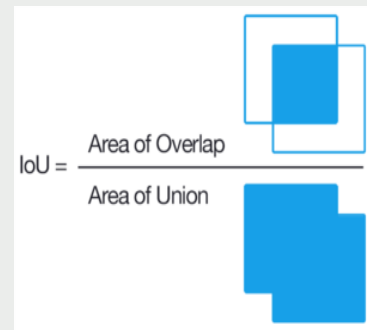
Métriques

Précision

Proportion des prédictions pertinentes parmi l'ensemble des données proposées

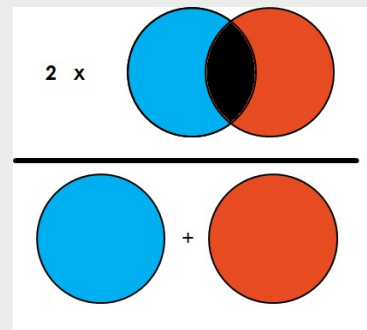
Coefficient IoU ou de indice de Jaquard

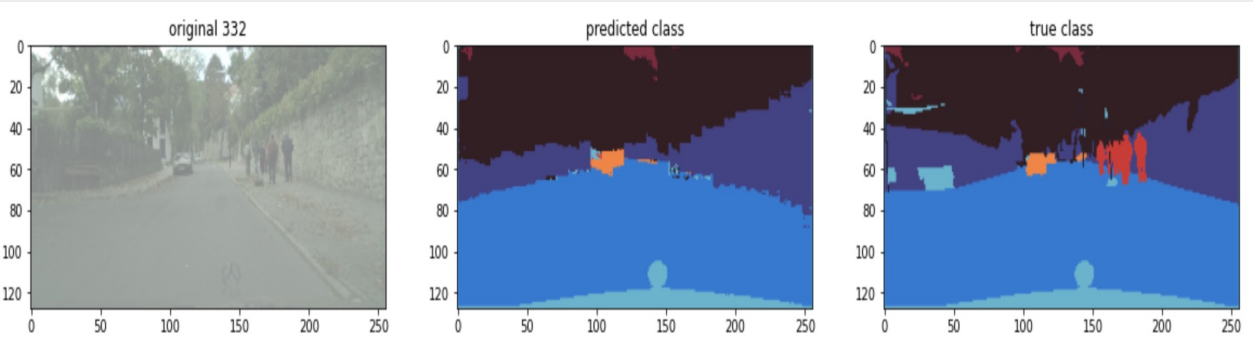
Zone de chevauchement entre la segmentation prédite et la vérité terrain divisée par la zone d'union entre la segmentation prédite et la vérité terrain



Coefficient de Dice ou coefficient de Sørensen-Dice

Double de la surface de l'intersection de la segmentation prédite et la vérité de terrain, divisé par la somme des surfaces des deux surfaces



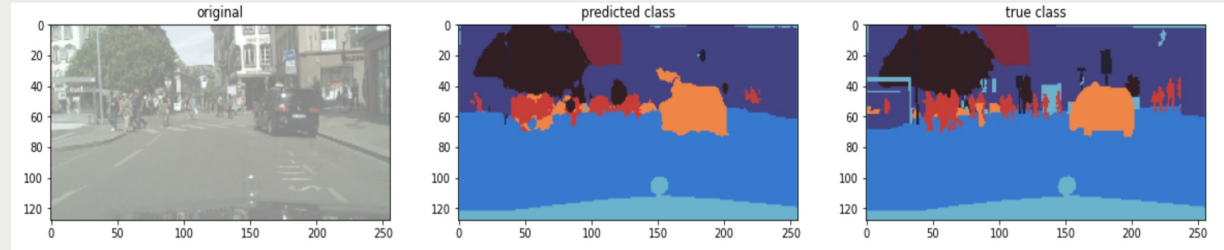


Temps d'entrainement	Perte	Précision	IoU	Dice
00:31:04	0.584	0.849	0.680	0.805

FCN-8

- Temps d'entrainement court
- Bonnes performances avec la catégorie « Flat »
- Mauvaises performances avec la catégorie « Human »

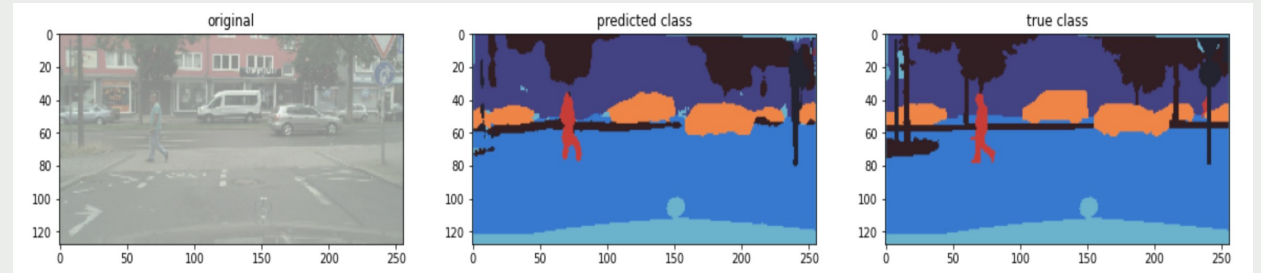
U-Net



Temps d'entrainement	Perte	Précision	IoU	Dice
00:11:27	0.452	0.885	0.297	0.457

- Temps d'entrainement très court
- Bonnes performances en général

PSPNet



Temps d'entrainement	Perte	Précision	IoU	Dice
01:09:07	0.610	0.904	0.817	0.896

- Temps d'entrainement très long
- Bonnes performances sur l'ensemble des catégories

Synthèse

Modèle	Temps d'entraînement	Perte	Précision	IoU	Dice
FCN-8	00:31:04	0.584	0.849	0.680	0.805
U-Net	00:11:27	0.452	0.885	0.297	0.457
PSP-Net	01:09:07	0.610	0.904	0.817	0.896

Sélection du modèle U-Net pour sa rapidité d'exécution et sa relative performance

Optimisations

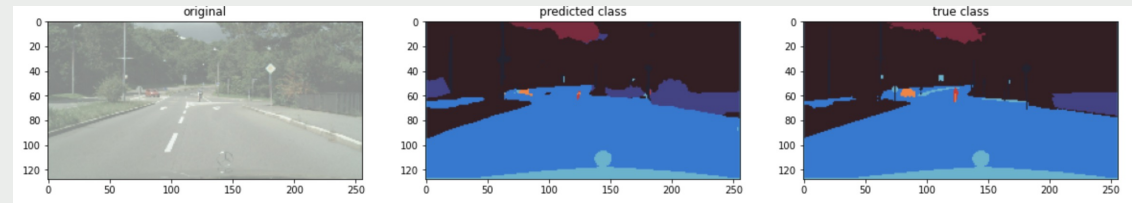
Dice Loss

Meilleure prise en compte des catégories qui sont par définition non uniformes au sein d'une même image.

Ajout de données

- Retournement horizontal
- Luminosité et contraste
- Flou
- Rotation (max 45°)

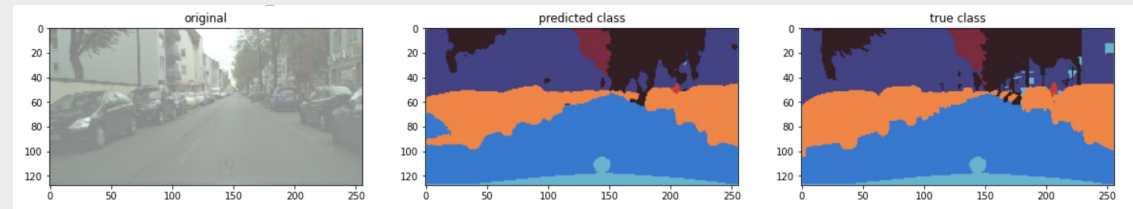
Optimisation Dice Loss



	Temps d'entrainement	Perte	Précision	IoU	Dice
Avant	00:11:27	0.452	0.885	0.297	0.457
Après	00:11:44	0.067	0.887	0.770	0.876

- Meilleurs scores IoU et Dice

Optimisation Data Augmentation



	Temps d'entrainem ent	Perte	Précision	IoU	Dice
Avant	00:11:27	0.452	0.885	0.297	0.457
Après	00:10:06	0.063	0.889	0.735	0.844

- Meilleurs scores IoU et Dice
- Meilleure précision

Conclusion

- Modèle U-Net
- Bonnes performances
- Utilisation de Dice Loss pour la calcul de la perte
- Utilisation de l'augmentation de données

Déploiement Azure

- Déploiement du modèle choisi
- Création du jeu de données
- Création de l'API
- Développement de l'API
- Présentation de l'API

Déploiement Azure

Déploiement du modèle choisi

```
model = Model.register(workspace = ws,  
model_path= os.path.join(app_folder, 'save_model'),  
model_name = 'future-vision-transport',  
description = 'Semantic segmentation')
```

Accueil > Modèles > future-vision-transport:9

future-vision-transport:9

Détails Versions Artefacts Points de terminaison Explications (préversion) Justesse (préversion) Jeux de données

Actualiser Déployer Tout télécharger

Attributs

Version
9

ID
future-vision-transport:9

Date d'inscription
26/08/2021 à 09:55:17

Infrastructure
Custom

Version du Framework
--

Nom de l'expérience
--

ID de série
--

Créé par
Pierre Augereau

Étiquettes

Aucune étiquettes

Propriétés

Aucune propriétés

Description


Semantic segmentation






Déploiement Azure


Création du jeu de données


Espace de stockage

Accueil > ocwrk7336028223 >

 **azureml-filestore-81a6ebff-7004-4f0d-9981-894656...**
Partage de fichiers

»  Connecter  Charger  Ajouter un répertoire  Actualiser  Supprimer le partage

 Rechercher les fichiers par préfixe

Nom	Type	Taille
 val	Répertoire	

Déploiement Azure

Création de l'API

Détails de l'instance

Vous avez besoin d'une base de données ?

[Essayez la nouvelle expérience web + base de données.](#)

Nom *

ftt81

.azurewebsites.net

Publier *

☒ Code

☐ Conteneur Docker

Pile d'exécution *

Python 3.8

Système d'exploitation *

☒ Linux

☐ Windows

Région *

Central US

Vous ne trouvez pas votre plan App Service ? Essayez une autre région.

Plan App Service

Le niveau tarifaire du plan App Service détermine l'emplacement, les fonctionnalités, le coût et les ressources de calcul associés à votre application.

[En savoir plus](#)

Plan Linux (Central US) * ⓘ

appsvc_linux_centralus_basic (B1)

Créer

Référence et taille *

De base B1

100 ACU au total , 1.75 Go de mémoire

Intégration du modèle

Nom
APPINSIGHTS_INSTRUMENTATIONKEY
APPLICATIONINSIGHTSAGENT_EXTENS
AZUREML_PASSWORD

Mappage des chemins d'accès


Nom	Chemin de montage	Type	Nom du compte	Nom du partage
dataset	/dataset	AzureFiles	ocwrk7336028223	azureml-filestore-81a6eb

18

Déploiement Azure

- Utilisation des extensions Azure pour VS Code
- Code sous GitHub
- Utilisation de ServicePrincipalAuthentication pour l'accès au modèle
- Copie du modèle depuis l'espace Azure
- Accès aux images et aux masques par le point de montage
- Prédiction faite dans l'API

Déploiement Azure

Future Vision Transport




Image d'origine

Sélection d'une image dans le jeu de données

Lancer la prédiction

Au hasard Balthazar

Prédiction de la segmentation

Accuracy : 0.937%

Mean IoU : 0.295

Construction	0.473
Appartement	0.488
Humain	0.275
Nature	0.192
Objet	0.000
Ciel	0.000
Véhicule	0.440
Autre	0.491




Image annotée

Questions