

# Projet 8 : Future Vision Transport

Challenge : Participez à la conception d'une voiture autonome

---

[https://github.com/blanchonnicolas/IA\\_Project8\\_Openclassrooms\\_IA\\_VoitureAutonome](https://github.com/blanchonnicolas/IA_Project8_Openclassrooms_IA_VoitureAutonome)

# Agenda

01

## Introduction

Contexte et Objectifs  
Présentation du jeu de données

02

## Architecture Réseau de Neurones

L'architecture UNET  
Le choix des métriques  
La fonction de perte  
L'apport de l'augmentation des images  
Les autres architectures  
Backbones

03

## Déploiement de l'outil

L'architecture client / serveur  
Les plateformes Cloud  
Heroku et Streamlit Share  
La Démo

04

## Note Technique

le livrable PDF

# Introduction

---

# Contexte et Objectifs

# Systèmes de vision par ordinateur

Les véhicules autonomes doivent être en mesure de détecter les obstacles capturés par les caméras, et classer ceux-ci en différentes catégories.

---

# Segmentation des images

La segmentation d'image permet de répondre à ce besoin, afin de reconstituer la situation routière par reconnaissance de formes (voies, véhicules, obstacles, panneaux, limites de chaussées)

---

# Objectifs de la mission



Concevoir un premier modèle de segmentation d'images, qui devra s'intégrer facilement dans la chaîne complète du système embarqué.



Fournir un API simple à utiliser, qui prendra une image en entrée, et renverra la segmentation de l'image en sortie.



Jeu de données CityScapes, sur lequel nous travaillons à l'augmentation des données d'origines.



Technique de Computer Vision, utilisant les réseaux de Neurones.  
Outil déployé en production.  
Documentation technique explicative.

# Présentation du jeu de données

# Compréhension du jeu de données



## Cityscapes

### Phase

<input type="checkbox"/>	train	2975
<input type="checkbox"/>	test	1525
<input type="checkbox"/>	val	500

### City

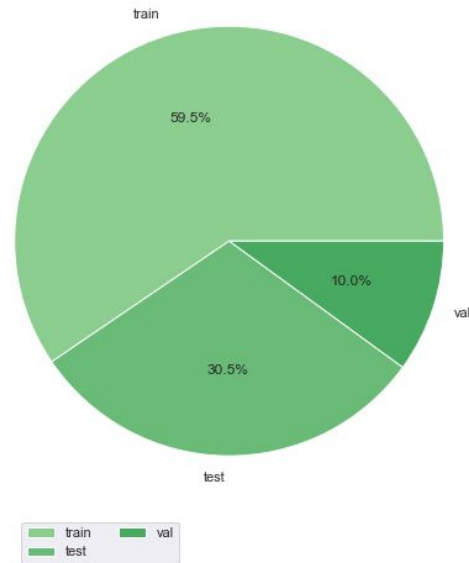
<input type="checkbox"/>	strasbourg	365
<input type="checkbox"/>	bremen	316
<input type="checkbox"/>	hamburg	248
<input type="checkbox"/>	dusseldorf	221
<input type="checkbox"/>	hanover	196
<input type="checkbox"/>	stuttgart	196
<input type="checkbox"/>	aachen	174
<input type="checkbox"/>	cologne	154
<input type="checkbox"/>	others	1105

12Go de données stockées,  
contenant une structure de  
dossiers avec image PNG et  
fichiers JSON.

8 Catégories présentes  
dans les masques, à  
consolider à partir des  
fichiers

*`_gtFine_labelIds.png`*

3 most presents values identified in column Phase :  
TOTAL unique = 3





# Architecture Réseau de Neurones

---

# Architecture UNET

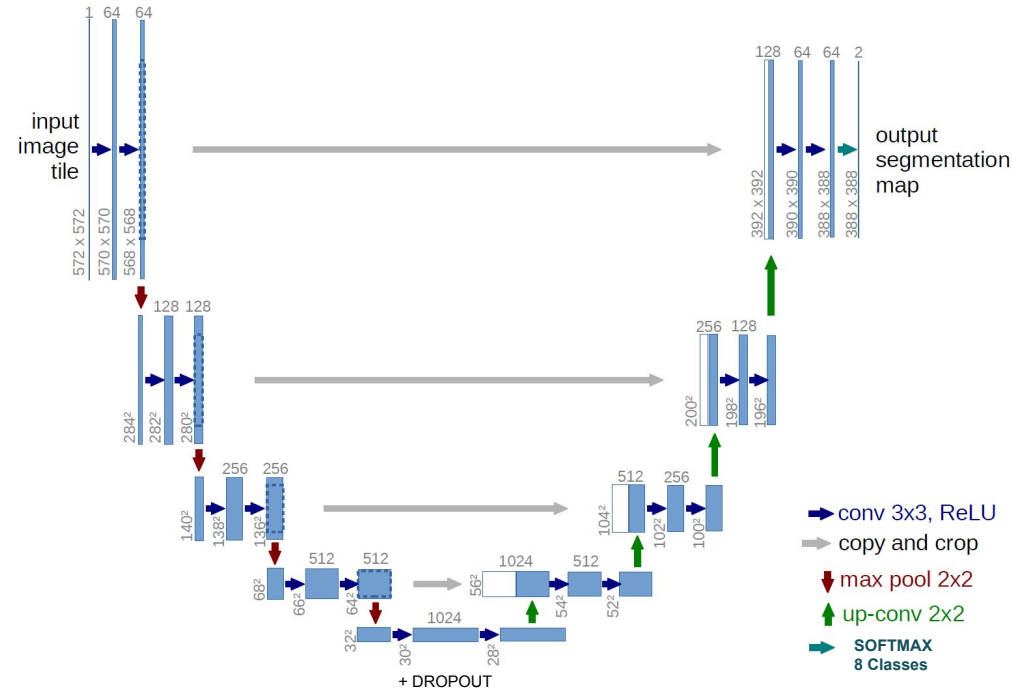
# Architecture UNET

U-NET est un modèle de réseau de neurones performant pour les tâches de “Computer Vision”. L'objectif est donc de générer à partir d'une image, un masque sur lequel chaque pixel est classifié (==> Segmentation)

L'architecture de U-NET est composée de deux chemins (architecture en U):

- Contraction: Capturer le contexte d'une image.
- Expansion: Localise grâce à la convolution transposée.

Cette architecture préserve la taille initiale = la taille de sortie est égale à la taille d'entrée.



# Modèle UNET: Recherche des meilleures métriques

## Metric: Accuracy

Baseline: Non adapté au dataset déséquilibré, et donc peu pertinent pour la segmentation.

01

## Metric: Dice

Adapté aux classes déséquilibrées. Pénalise légèrement les vrais négatifs

02

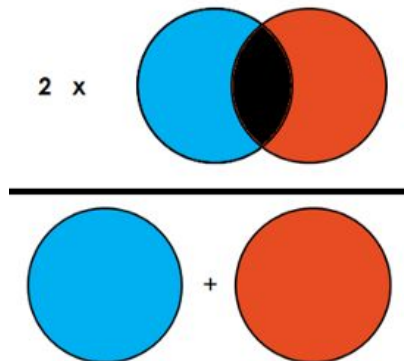
## Metric: IoU

Adapté aux classes déséquilibrées. Pénalise durement les vrais négatifs

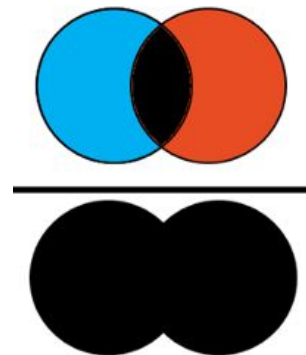
03

Images & Masks non augmentées & redimensionnées: Largeur = 256 ; Hauteur = 128

Dice  
Coefficient  
(F1 Score)



Métriques adaptées  
au classes  
déséquilibrées (Ici 8  
Classes mesurées  
par la moyenne  
IOU)

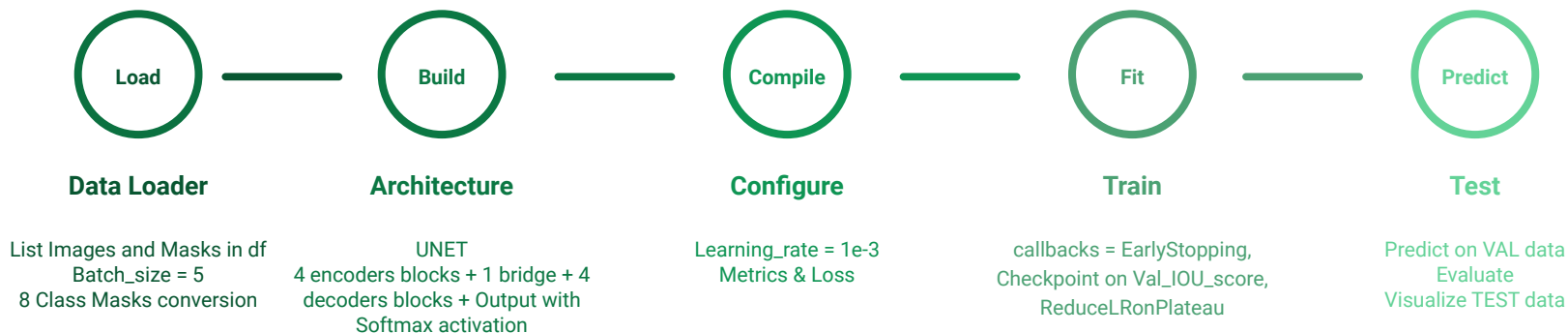


Intersection-O  
ver-Union  
(Jaccard  
Index)

# Modèle UNET: Recherche de la fonction de Coût

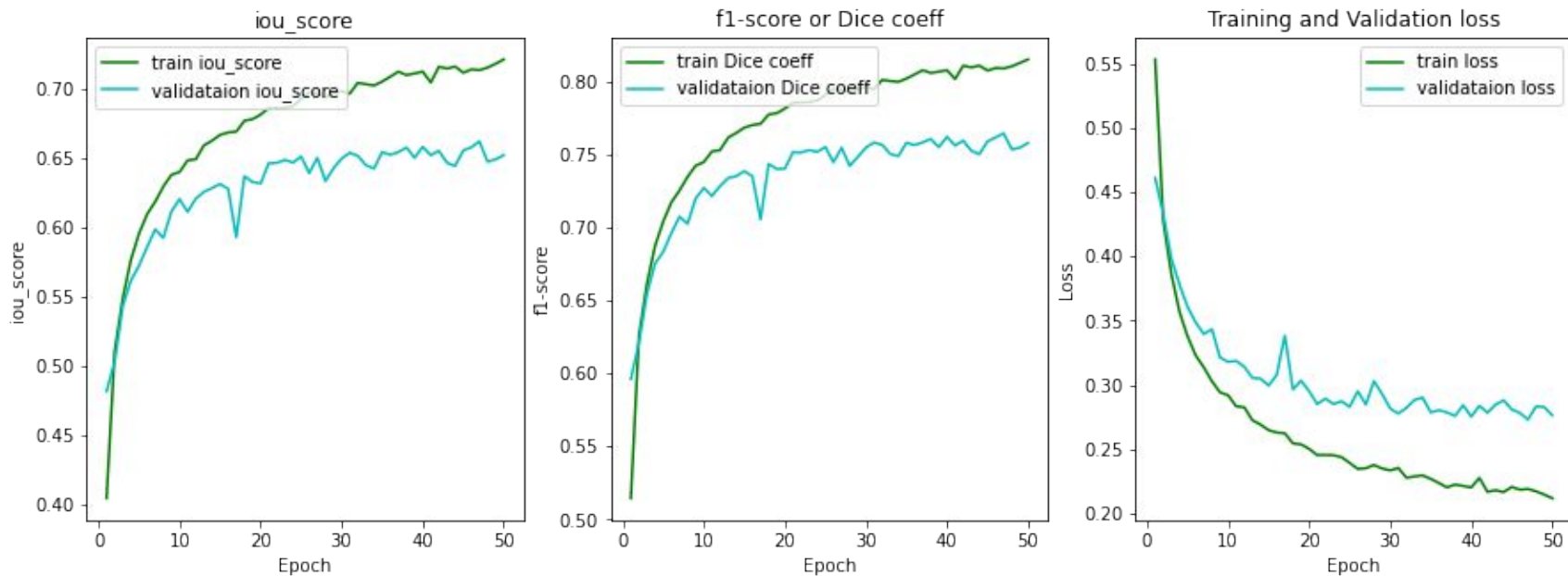


Utilisation des librairies segmentation\_models



# UNET: Courbes d'apprentissages du Meilleur Modèle

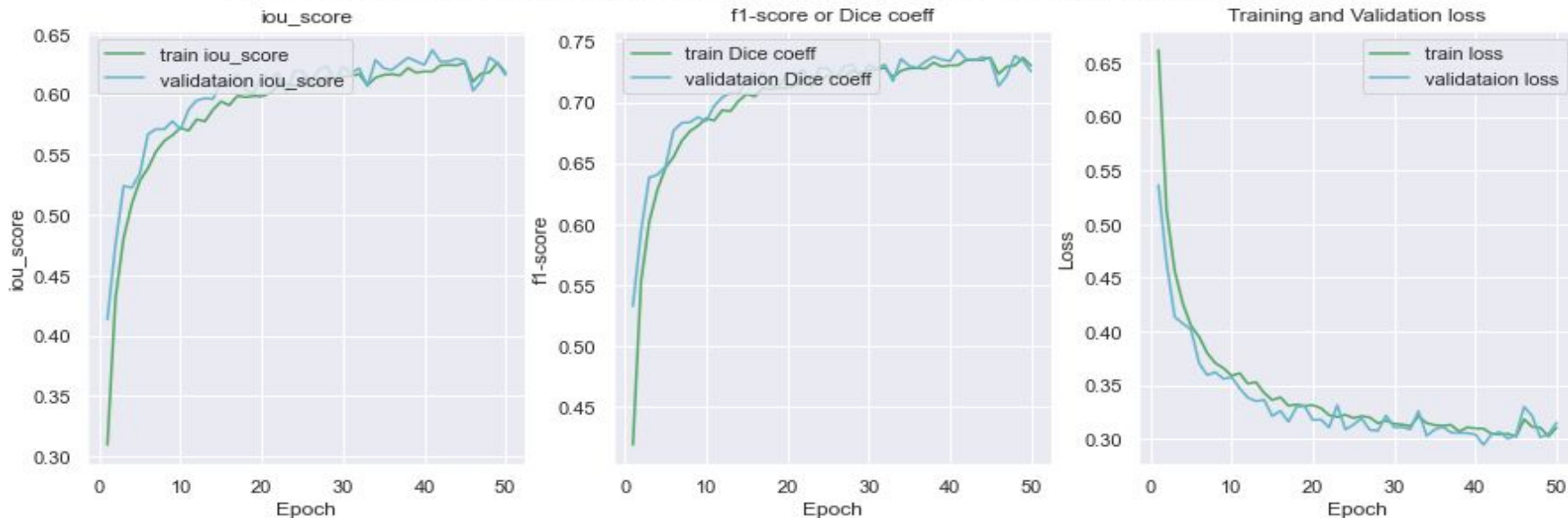
IoU Score, Dice Coeff and Loss evolution during U-Net training, (stopped by callback at epochs : 50)



Courbes démontrant un apprentissage du modèle (fonction de coût décroissante), et un sur-apprentissage identifié au fil des epochs par la divergence des courbes train/validation.

# UNET: Courbes avec augmentations des données

IoU Score, Dice Coeff and Loss evolution during U-Net training, (stopped by callback at epochs : 50)

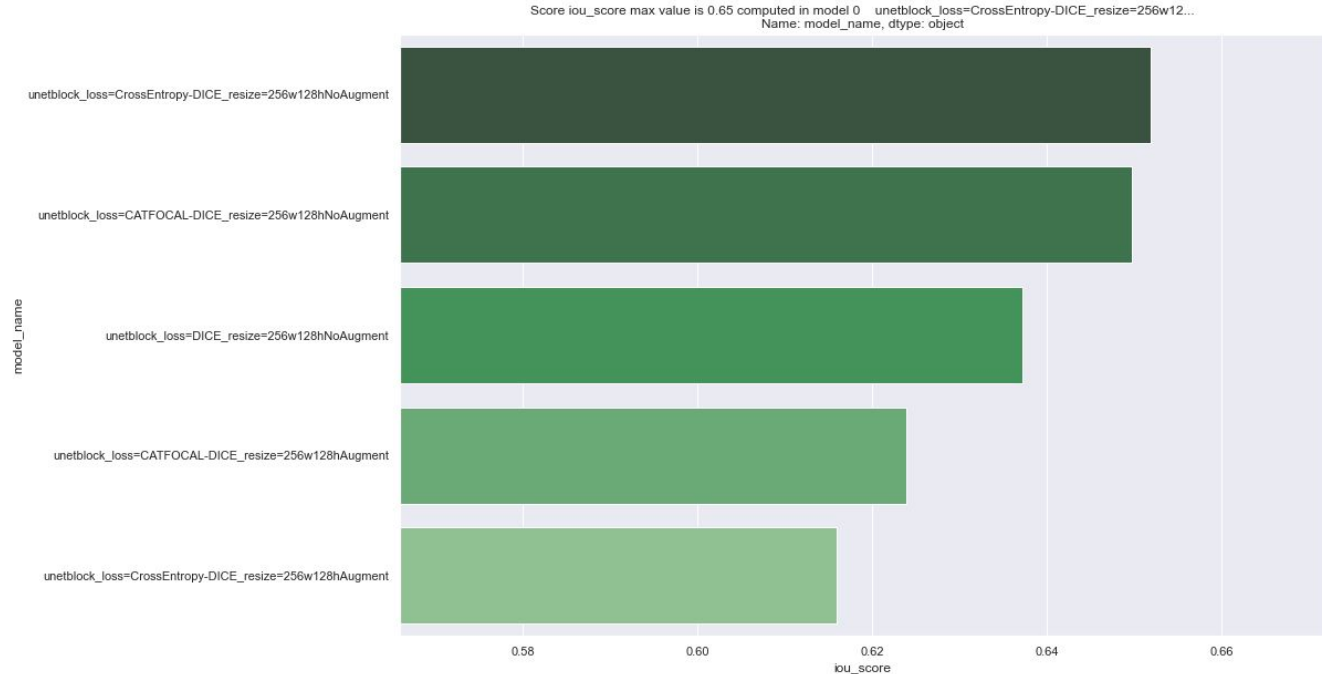


L'Augmentation permet de réduire le risque de sur-apprentissage, en appliquant plusieurs transformations sur les données d'entraînement (Rotation, Shift, Saturation RGB, Contrast, ...)

# Modèle UNET: Top 5 IOU scores

Meilleur IOU  
score de 0.6518

Score de la  
fonction de coût  
de 0.2762

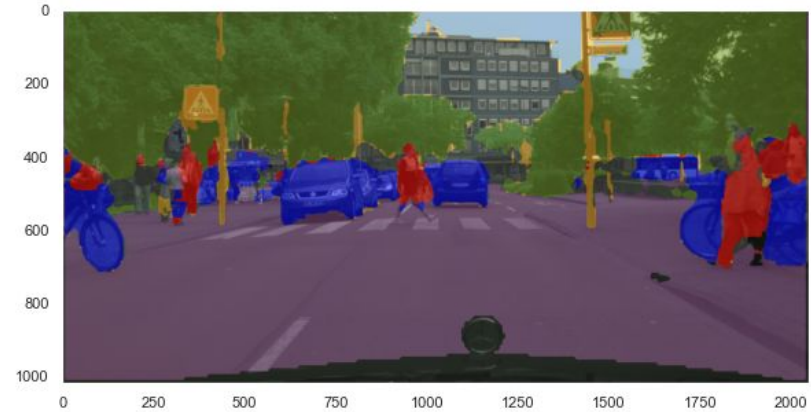
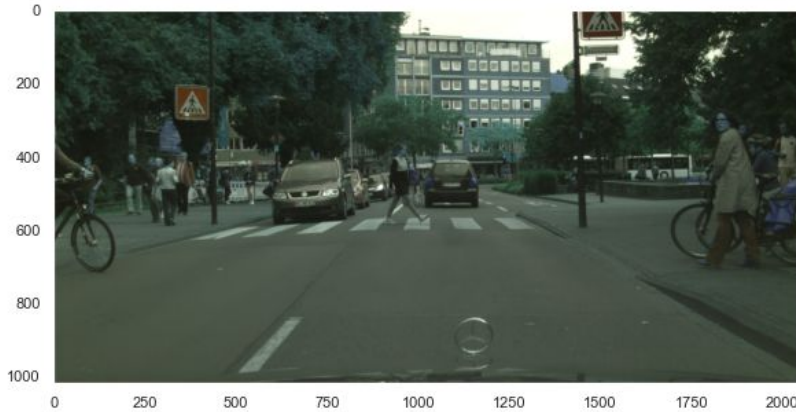


Meilleur modèle entraîné avec une fonction de coût mutualisant la Cross Entropy et le coefficient de Dice ([Loss scénario 4](#))



# Visualisation d'un résultat de segmentation

Native Image, and 8 classes predicted Mask with best model

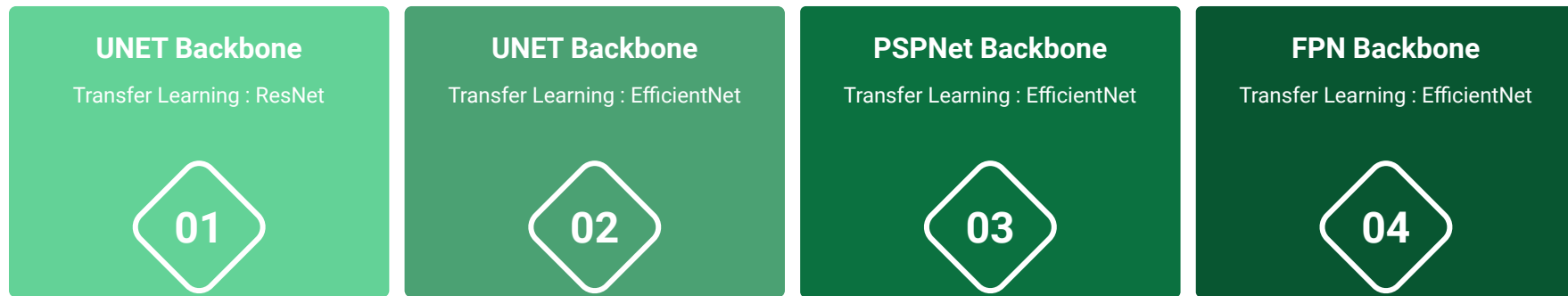


## Plusieurs défauts notables:

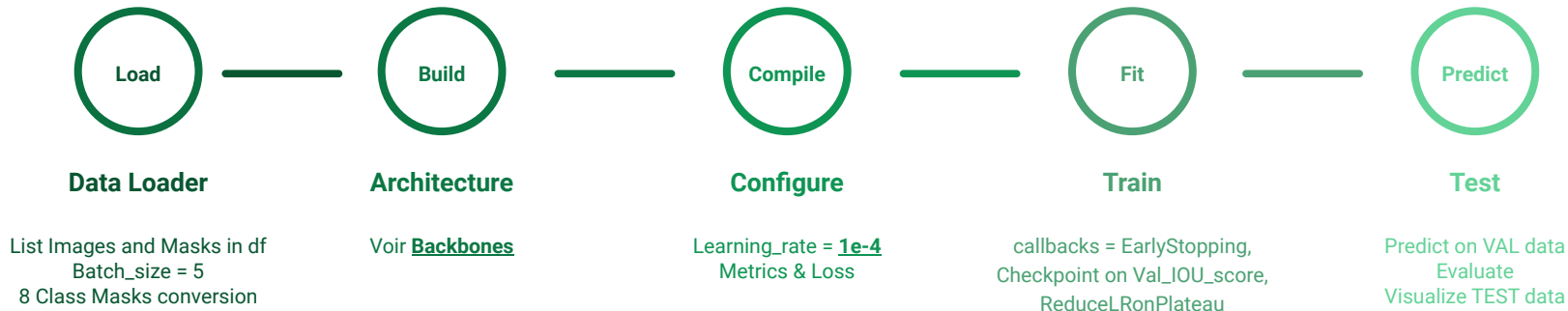
- Segmentation Véhicule vs Humain (Cas du cycliste à droite)
- Segmentation Construction vs Véhicule (Cas du food truck à Gauche)
- Segmentation Nature vs Objet (Cas du panneau à Droite)

# Autres Architectures de réseau de neurones

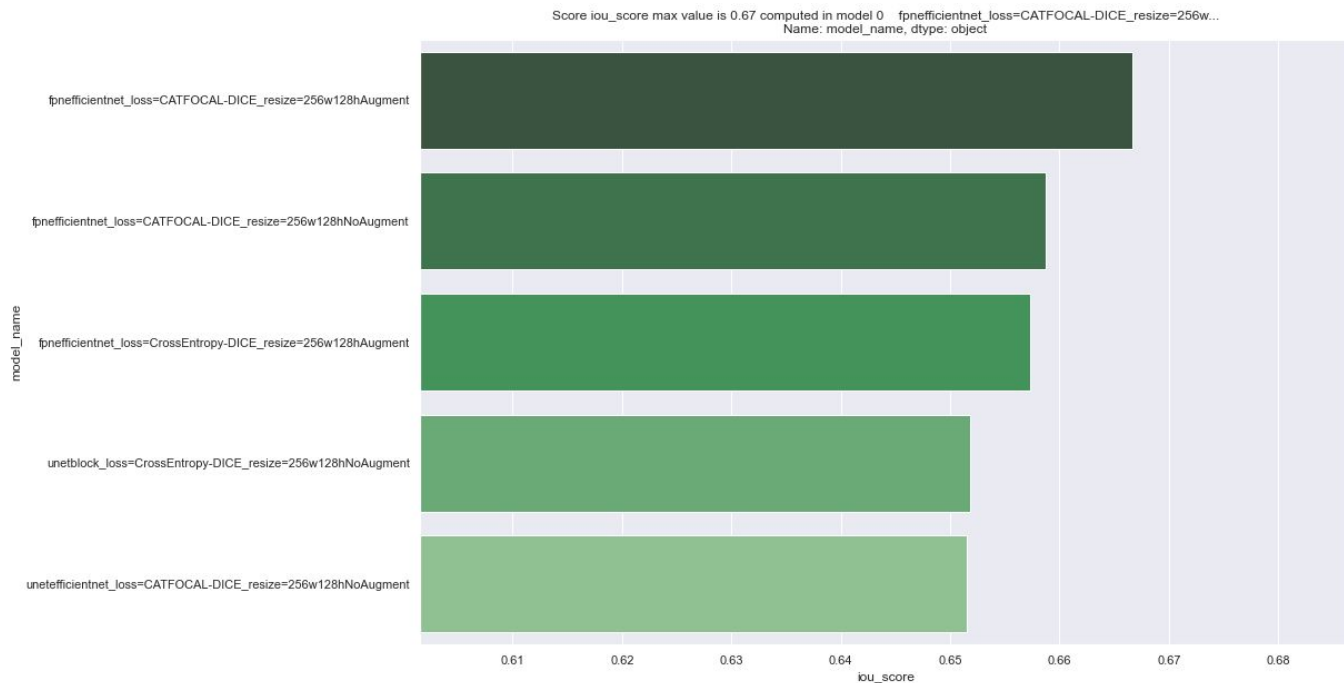
# Diverses Architectures Backbone, avec Transfer Learning



Nous utilisons ici des architectures Backbones, contenant des poids pré-entraînés sur le dataset ImageNet 2012 ILSVRC (Voir [Librairie GitHub](#))



# Résultat des Modèles : Top 5 IOU scores



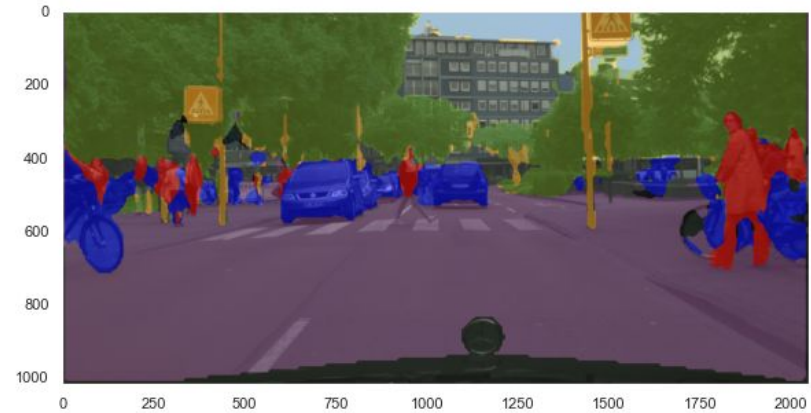
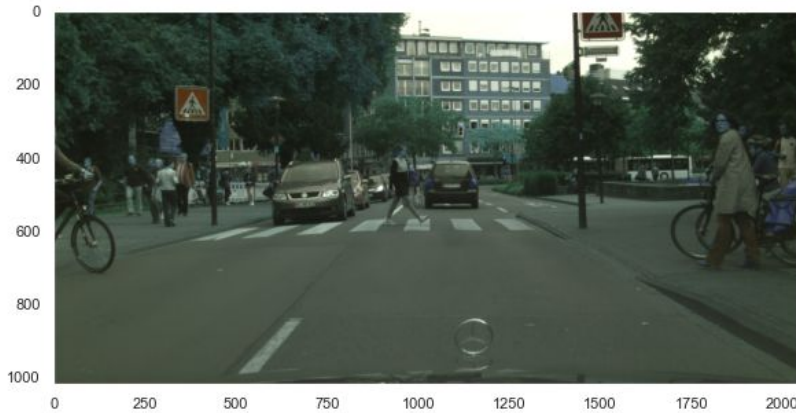
Meilleur IOU  
score de  
0.6666

Score de la  
fonction de  
coût de 0.245

Meilleur modèle Backbone entraîné **avec l'augmentation des données**, et une fonction de coût mutualisant la Cross Entropy + Focal et le coefficient de Dice (Loss scénario 5)

# Visualisation d'un résultat de segmentation

Native Image, and 8 classes predicted Mask with best model

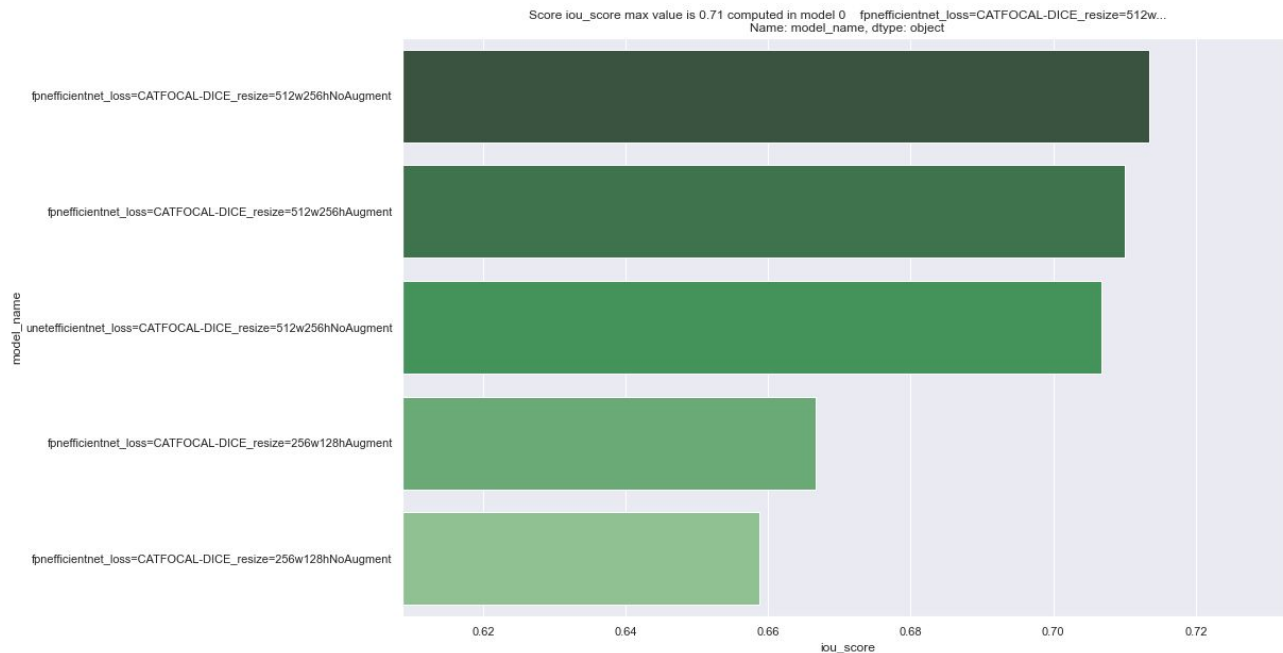


## Plusieurs défauts persistants avec quelques améliorations:

- Segmentation Véhicule vs Humain (Cas du cycliste à droite + Cas du piéton au centre)
- Segmentation Construction vs Véhicule (Cas du food truck à Gauche) → Amélioré
- Segmentation Nature vs Objet (Cas du panneau à Droite) → Amélioré

# Tailles des Images et Apprentissage

# Résultat des Modèles : Top 5 IOU scores



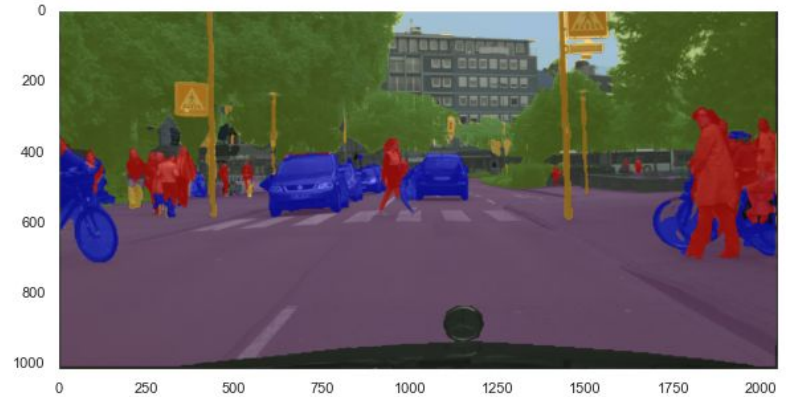
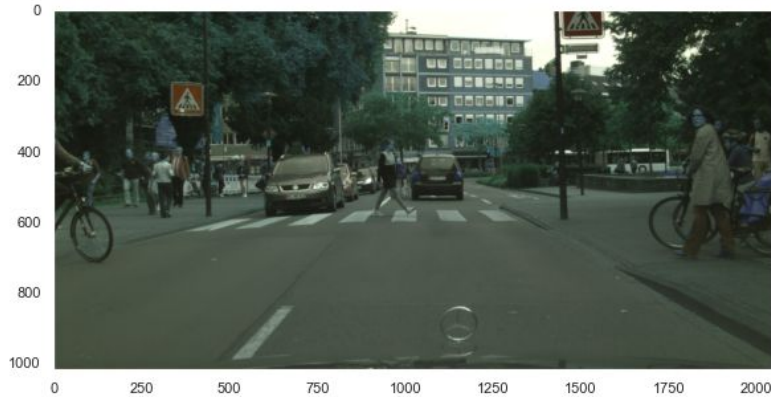
Meilleur IOU  
score de  
0.7134

Score de la  
fonction de  
coût de 0.209

Meilleur modèle Backbone entraîné sur des **données de plus grandes tailles**, avec l'**augmentation des données**, et une fonction de coût mutualisant la Cross Entropy + Focal et le coefficient de Dice (Loss scénario 5)

# Visualisation d'un résultat de segmentation

Native Image, and 8 classes predicted Mask with best model



## Plusieurs améliorations notables !

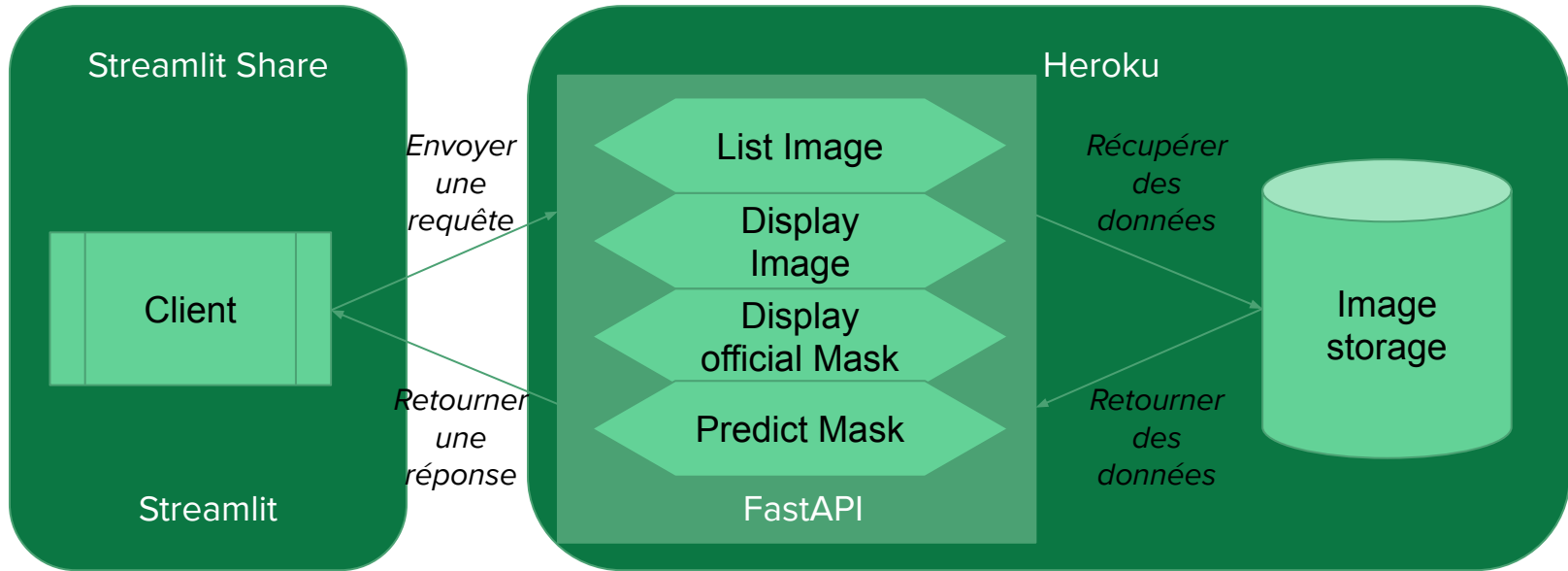
- Segmentation Route vs Humain (Cas du piéton au centre) → Amélioré
- Segmentation Construction vs Véhicule (Cas du food truck à Gauche) → Amélioré
  - Attention, le Bus à droite n'est lui plus aussi bien détecté !
- Segmentation Nature vs Objet (Cas du panneau à Droite) → Amélioré



# Déploiement de l'outil

---

# Architecture Client / Server

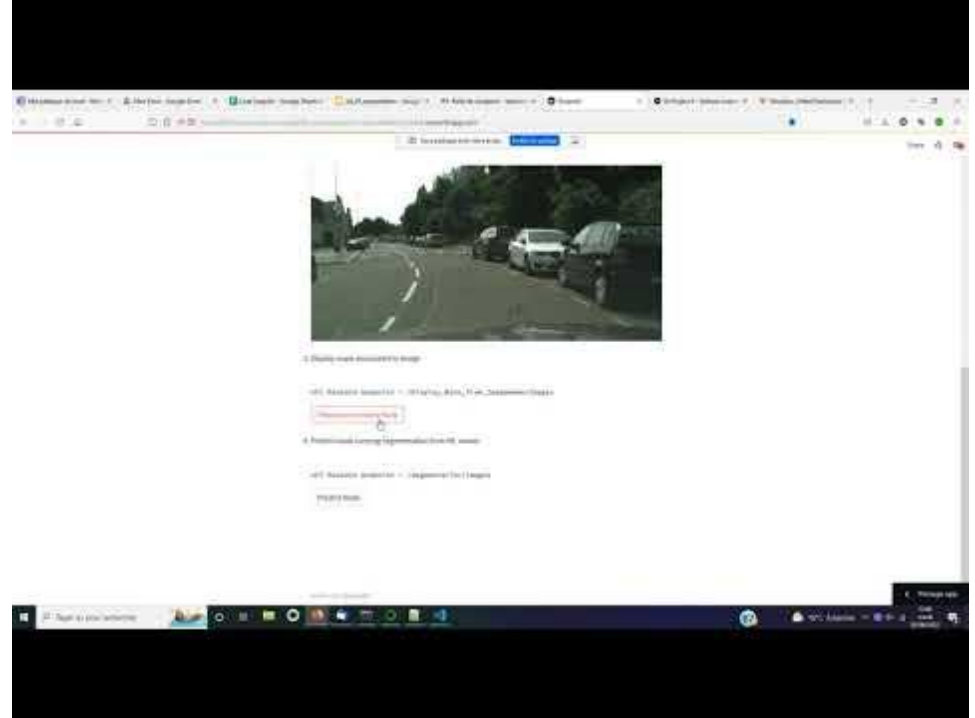


- Interface Client accessible : <https://blanchonnicolas-ia-project8-openclassroom-user-interface-5iw4te.streamlitapp.com/>
- API URL = '<https://vast-sea-04286.herokuapp.com>'; 4 APIs endpoint :
  - List\_images = '/List\_Images/Images'
  - Display\_image = '/Display\_Image\_from\_ImageName/Images'
  - Display\_mask = '/Display\_Mask\_from\_ImageName/Images'
  - Predict\_mask = '/Segmentation/Images'

# Démonstration

## 4 fonctionnalités:

- Affichage de la liste des id des images disponibles
- Affichage de l'image réelle
- Affichage du mask associé à l'image réelle
- Affichage du mask prédit, à partir de l'image réelle



# Note Technique (voir PDF)

---

# Conclusions



Plusieurs modèles ont été évalués.  
La stratégie Backbone est pertinente.  
L'utilisation des poids pré-entraînés  
offre un gain de performance.



Le générateur de données est un outil  
indispensable pour traiter de larges volumes.  
La mise à disposition des APIs sur le cloud  
permet une utilisation efficace pour les  
équipes en charge du système de décision.



Le choix des métriques adaptées à la  
problématique métier.  
Le choix de la fonction de perte  
permet d'optimiser l'apprentissage  
du modèle.



Les techniques d'augmentation des  
données peuvent être poussées.  
La taille des images doit considérer le  
compromis entre la ressource consommée  
et la performance obtenue.