



## EXPOSÉ PROJET 8

---

# PARTICIPEZ À LA CONCEPTION D'UNE VOITURE AUTONOME

---


Le 28 Juillet 2021

Zeineb Guizani

# Plan de la présentation

1. Problématique métier
2. Description du jeu de données
3. Présentation des approches
4. Comparaison des approches
5. Déploiement de API Flask grâce au service Azure
6. Conclusion

# I-Problématique métier

- L'entreprise « Future Vision Transport »  conçoit des systèmes embarqués de vision par ordinateur pour les voitures autonomes.
  - Produire un modèle à base de réseaux de neurones qui permet de segmenter les images.
  - Un problème de classification pixellique où chacun des pixels de l'image doit fournir une catégorie.
- Objectifs:
  - Générateur de données.
  - Keras et Azure Machine Learning pour l'entraînement des modèles.
  - Architectures
  - Augmentations de données.
- Résultats:
  - Déploiement du meilleur modèle sur Azure ML.
  - Faire l'api Flask et l'interface avec streamlit.

## II- Description du jeu de données

### Cityscapes Dataset

- Complexité:
  - 32 catégories à fusionner en 8 : les panneaux, les piétons, les trottoirs, les voitures, la végétation, la construction, le ciel ou le contexte.
- Diversité:
  - 50 villes, plusieurs mois (printemps, été, automne), scènes et contextes.
- Volume:
  - 5000 images annotées avec des annotations fines
  - 20000 images annotées avec des annotations grossières

## II- Description du jeu de données

```
Cityscapes
├── leftImg8bit_trainvaltest
│   ├── leftImg8bit
│   │   ├── train
│   │   │   ├── aachen
│   │   │   │   ├── aachen_000000_000019_leftImg8bit.png
│   │   │   │   ├── aachen_000001_000019_leftImg8bit.png
│   │   │   │   └── ...
│   │   │   ├── bochum
│   │   │   └── ...
│   │   ├── val
│   │   └── test
│   └── gtFine_trainvaltest
│       ├── gtFine
│       │   ├── train
│       │   │   ├── aachen
│       │   │   │   ├── aachen_000000_000019_gtFine_color.png
│       │   │   │   ├── aachen_000000_000019_gtFine_instanceIds.png
│       │   │   │   ├── aachen_000000_000019_gtFine_labelIds.png
│       │   │   │   ├── aachen_000000_000019_gtFine_polygons.json
│       │   │   │   └── ...
│       │   │   ├── bochum
│       │   │   └── ...
│       │   ├── val
│       │   └── test
```

### Data Generator

- Script automatisé et permet le traitement des images sur plusieurs cœurs de calcul.
1. redimensionnement de la taille de l'image
  2. l'application de l'augmentation des données
  3. la normalisation des images
  4. one hot encoding » sur les labels du masque.
  5. l'application du shuffle

# III- Présentation des approches

Architectures

Métriques d'évaluation

Fonctions de coût

Augmentation

## ■ Unet

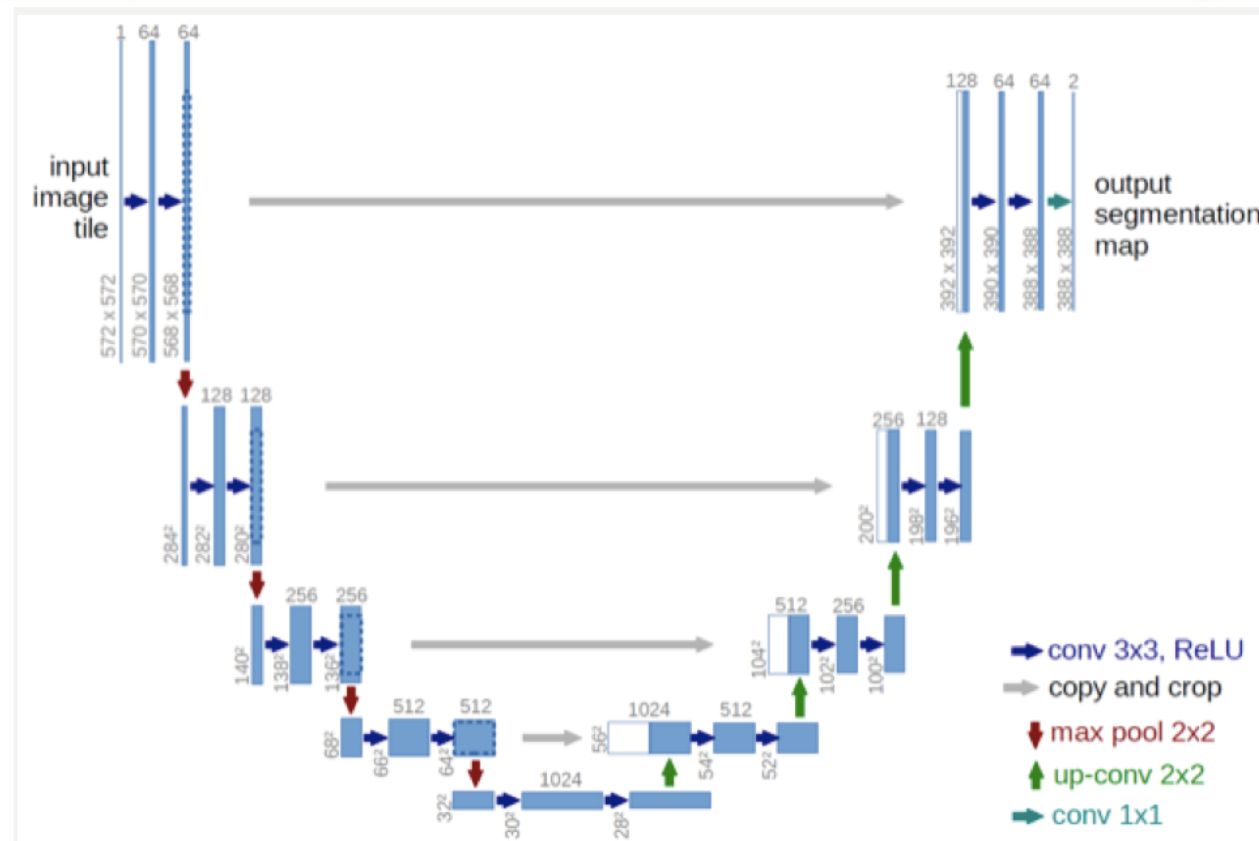


Figure Architecture UNet

# III- Présentation des approches

Architectures

Métriques d'évaluation

Fonctions de coût

Augmentation

## ■ UNet de segmentation models

les librairies fournissent des UNets dont l'encodeur est un réseau de neurone pré-entraîné afin d'améliorer les performances.

**TABLEAU COMPARATIF BACKBONES**

Utilisation	Architecture du backbone	Avantage
La construction d'un modèle pour une utilisation en routine clinique	Xception, MobileNet et <b>EfficientNet</b>	compromis entre consommation des ressources et niveau de précision.
La classification des images où l'information spatiale est fortement discriminante	Squeeze-and-ExcitationNet et Inception	S'adaptent mieux à la forme discriminante.
La classification de plusieurs structures anatomiques	<b>ResNet</b> ou DenseNet	Architecture robuste et performante

# III- Présentation des approches

Architectures


Métriques d'évaluation

Fonctions de coût

Augmentation

## 1) Calcul d'IoU (Intersection over Union)

- une métrique classique en traitement d'image.
- un indicateur permettant de déterminer si la prédiction recouvre pertinemment la ground truth.
- métrique sera calculée pour toutes les images à prédire et sera suivie au cours de l'apprentissage.
- utiliser la librairie segmentation models


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

## 2) Temps d'entraînement



# III- Présentation des approches

Architectures

Métriques d'évaluation

Fonctions de coût

Augmentation

Fonction de perte	Utilisation
Categorical Cross-Entropy	classification multi-class à simple label de sortie.
Dice	<ul style="list-style-type: none"><li>-Inspiré des coefficients Dice, métriques utilisée pour évaluer la performance d'un réseau de segmentation sémantique d'image en mesurant le chevauchement entre deux objets.</li><li>- problématiques de segmentation d'image</li></ul>

## III- Présentation des approches

Architectures

Métriques d'évaluation

Fonctions de cout

Augmentation de données

### ■ Albumentations

appliquer plusieurs transformations d'images pour créer artificiellement de nouvelles données

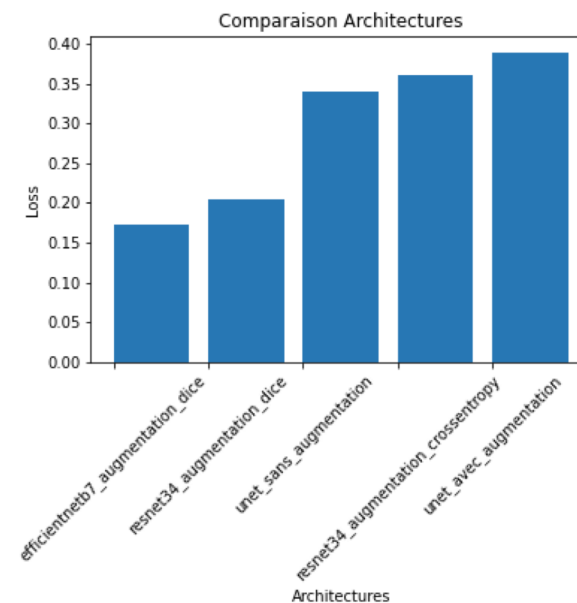
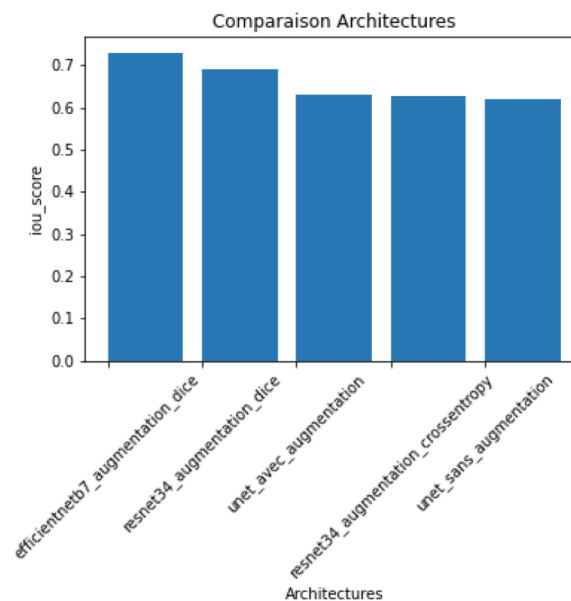
Rotation	Changement d'échelles	Ajout d'effets	Décalage de couleurs	Eclairage	Distorsion
HorizontalFlip	ShiftScaleRotate	Blur	RGBShift	Random Brightness	Grid Distortion
ShiftScaleRotate		GaussNoise		Random Gamma	Optical Distortion
				RandomContrast	

## IV- Comparaison des approches

Augmentation	IoU	Loss	Temps de Traitement	Augmentation	IoU	Loss	Temps de Traitement
Aucune	<b>0.620</b>	0.341	2 h 58 m	RandomGamma(p=0.25), GaussNoise(p=0.5), Blur(p=0.25, blur_limit=7)	0.606	0.349	3 h 23 m
RandomGamma(p=0.25), Blur(p=0.25, blur_limit=7), GaussNoise(p=0.5)	0.599	0.365	3 h 9 m	ShiftScaleRotate(shift_limit=0.0625, scale_limit=0.2, rotate_limit=45, p=0.2), HorizontalFlip(p=0.5), RandomGamma(p=0.25), GaussNoise(p=0.2), Blur(p=0.25, blur_limit=7)#,	0.577	0.369	3 h 24 m
HorizontalFlip(p=0.5), RandomGamma(p=0.25),	<b>0.626</b>	0.362	3 h 26 m	A.HorizontalFlip(p = 0.5), OneOf( [ A.RandomContrast(),A.RandomGamma(), A.RandomBrightness(), ], , OneOf( [ A.ElasticTransform( alpha = 120, sigma = 120 * 0.05, alpha_affine = 120 * 0.03), A.GridDistortion(), A.OpticalDistortion( distort_limit = 2, shift_limit = 0.5 ), ], ) )	0.591	0.330	3 h 25 m
RandomGamma(p=0.25), Blur(p=0.25, blur_limit=7),	<b>0.630</b>	0.390	3 h 25 m				
ShiftScaleRotate(), RGBShift(), Blur(), GaussNoise()	0.536	0.407	3 h 24				
RGBShift(), RandomGamma(p=0.25), Blur(p=0.25, blur_limit=7),	0.588	0.364	3 h 23				

## IV- Comparaison des approches

Architecture	IoU	Loss	Temps de Traitement
UNet simple sans augmentation	<b>0.620</b>	0.341	2 h 58 m
UNet simple avec augmentation	<b>0.630</b>	0.390	3 h 25 m
backbone_resnet34_CE	<b>0.626</b>	0.361	1 h 0 m 35
backbone_resnet34_Dice	0.690	0.205	1 h 15 m
backbone_efficientnetb7_Dice	<b>0.731</b>	<b>0.172</b>	4 h 36 m

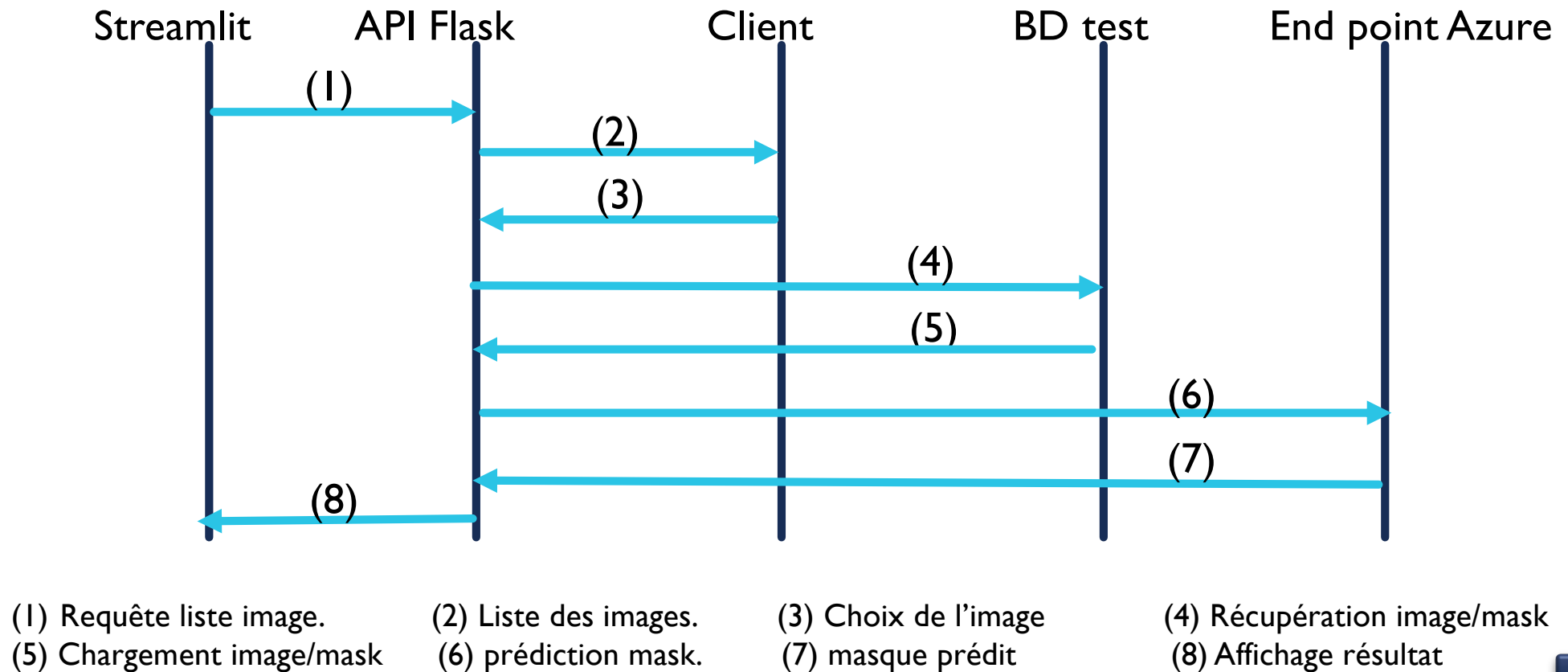


## V- Déploiement du modèle final avec Azure ML

1. Récupération de la configuration du meilleur modèle
2. Enregistrement de la configuration du modèle
3. Déploiement du meilleur modèle
4. Déploiement de l'api Flask sur Azure
5. Interface graphique avec Streamlit

## V- Déploiement du modèle final avec Azure ML

Diagramme de séquence du déploiement :



# Conclusion et axes d'amélioration

## Conclusion & perspectives :

- Production d'un modèle à base de réseaux de neurones pour la segmentation d'images.
- Implémentation d'un générateur de données.
- Entraînement à base d'architectures de réseau de neurones pour la segmentation d'images.
- Utilisation de différentes augmentations de données.
- Déploiement d'une API Flask grâce au service Azure, affichée par Streamlit.
- Test d'architectures du modèle différentes de UNet .
- Traiter le problème de la dégradation de la prédiction suite à la petite résolution d'images.
- Prise en compte du temps de prédiction



MERCI DE VOTRE ATTENTION

