Note Méthodologique

DATASET RETENU

Le dataset Cityscapes a été choisi pour ce projet. Il s'agit d'un ensemble de données d'images urbaines haute résolution annotées pour la segmentation sémantique, d'instance et panoptique. Ce dataset a été conçu pour l'analyse de scènes routières dans des environnements urbains variés, avec des annotations précises pour des objets tels que les routes, les véhicules, les piétons et les feux de signalisation.

Cityscapes est composé de 5 000 images finement annotées (train, validation, test) et de 20 000 images grossièrement annotées, capturées dans différentes villes sous divers angles et conditions d'éclairage. Il est largement utilisé comme benchmark pour évaluer les performances des modèles de segmentation.

** exemple de masque segmenté*

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+

** les différentes classes*

LES CONCEPTS DE CONVNEXT

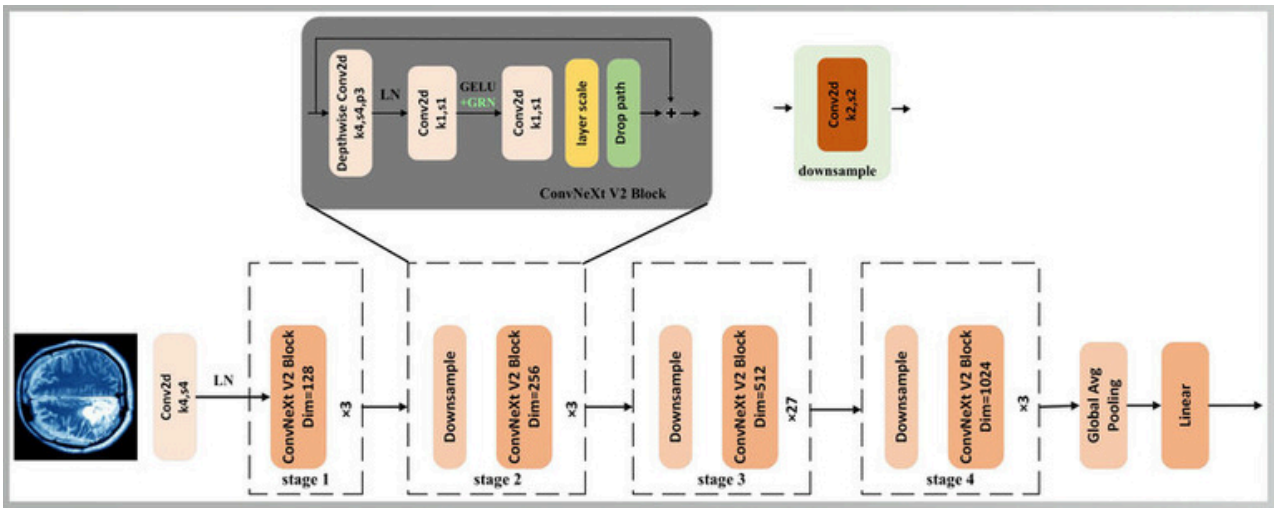
ConvNeXt est une architecture convolutionnelle récente conçue pour moderniser les CNNs et rivaliser avec les transformers dans les tâches de vision par ordinateur. Proposée par Zhuang Liu et al. (2022), elle conserve les principes fondamentaux des réseaux convolutifs tout en intégrant des optimisations inspirées des transformers.
L'objectif de ConvNeXt est d'améliorer l'efficacité et la précision des CNNs en repensant la structure des blocs convolutifs, tout en maintenant une complexité raisonnable pour une utilisation pratique.

Bien que ConvNeXt ait apporté des améliorations significatives aux CNNs traditionnels, certaines limites subsistaient, notamment en matière de régularisation et d'adaptabilité au pré-entraînement auto-supervisé.

Pourquoi une évolution vers ConvNeXt V2 ?
ConvNeXt V2, introduit en 2023, affine l'architecture en renforçant la robustesse et la stabilité du modèle grâce aux améliorations suivantes :

- Layer Scale → Permet de mieux ajuster l'amplitude des activations pour stabiliser l'apprentissage.
- DropPath → Introduit une régularisation avancée pour limiter l'overfitting.
- Optimisation du pré-entraînement → Exploite mieux l'auto-supervision, réduisant le besoin d'annotations massives.

Ces modifications font de ConvNeXt V2 un modèle plus performant et plus robuste, en particulier pour des tâches exigeantes comme la segmentation d'images urbaines.



** ConvNext V2 Architecture*

ConvNeXt V2 repose sur une structure hiérarchique en 4 étapes successives, avec des blocs de convolution modernisés qui optimisent le passage de l'information et la stabilité de l'apprentissage.

Architecture globale

- Des convolutions depthwise 7×7 (au lieu de 3×3 comme dans ResNet).
- LayerNorm avant les convolutions (au lieu de BatchNorm).
- Activation GELU pour améliorer la convergence.
- Réorganisation des blocs convolutionnels pour optimiser la propagation du gradient.

Améliorations spécifiques à ConvNeXt V2

- Layer Scale → Permet de mieux ajuster l'amplitude des activations dans chaque bloc, réduisant l'instabilité.
- DropPath → Introduit un mécanisme avancé de régularisation, améliorant la robustesse du modèle face au sur-ajustement (overfitting).
- Meilleure efficacité en pré-entraînement → ConvNeXt V2 est conçu pour exploiter les modèles pré-entraînés de manière plus efficace, permettant de meilleurs résultats avec moins d'annotations.

Résumé du fonctionnement des blocs ConvNeXt V2

Chaque ConvNeXt V2 Block suit la séquence suivante :

1. LayerNorm (normalisation avant la convolution).
2. Convolution depthwise 7×7 (extraction des features locales et globales).
3. Activation GELU (favorise une meilleure convergence).
4. Layer Scale (ajustement fin des activations).
5. DropPath (mécanisme de régularisation).

LA MODÉLISATION

Méthodologie de modélisation

Nous avons sélectionné ConvNeXt V2 comme backbone intégré dans FPN (Feature Pyramid Network) afin d'exploiter les capacités multi-échelles du modèle et améliorer la segmentation d'images urbaines.

Pipeline de modélisation

Notre approche suit les étapes suivantes :

- **Prétraitement des images**:
  - Redimensionnement des images à 512×512 pour l'entraînement.
  - Normalisation des pixels entre [0, 1].
  - Application d'une data augmentation modérée (rotation, flip horizontal) pour améliorer la généralisation.

- **Entraînement du modèle** :
  - Initialisation du modèle avec ConvNeXt V2 pré-entraîné sur ImageNet.
  - Ajout d'un FPN pour une meilleure exploitation des features à différentes résolutions.
  - Optimisation via Adam avec une scheduler de learning rate pour stabiliser l'apprentissage.

```python
class FPN_ConvNeXtV2_Segmenter(nn.Module):
    def __init__(self, num_classes=8):
        super(FPN_ConvNeXtV2_Segmenter, self).__init__()

        # ✅ Charger ConvNeXt V2-Large pré-entraîné
        self.convnext_backbone = timm.create_model("convnextv2_large", pretrained=True, features_only=True)

        # ✅ Convolutions latérales 1x1 pour aligner les features avec FPN
        self.lateral_convs = nn.ModuleList([
            nn.Conv2d(192, 256, kernel_size=1),   # Feature 0 (128x128)
            nn.Conv2d(384, 256, kernel_size=1),   # Feature 1 (64x64)
            nn.Conv2d(768, 256, kernel_size=1),   # Feature 2 (32x32)
            nn.Conv2d(1536, 256, kernel_size=1),  # Feature 3 (16x16)
        ])

        # ✅ Convolutions 3x3 après fusion des features
        self.fpn_convs = nn.ModuleList([
            nn.Conv2d(256, 256, kernel_size=3, padding=1),
            nn.Conv2d(256, 256, kernel_size=3, padding=1),
            nn.Conv2d(256, 256, kernel_size=3, padding=1),
            nn.Conv2d(256, 256, kernel_size=3, padding=1),
        ])

        # ✅ Convolution finale pour segmentation
        self.final_conv = nn.Conv2d(256, num_classes, kernel_size=1)
```

** Code pour ConvNext V2*

Hyperparamètres clés

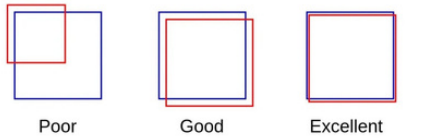
- Optimizer : Adam (lr et weight_decay)
- Batch size : Ajusté pour équilibrer performance et mémoire.
- Régularisation :
  - EarlyStopping : Stoppe l'entraînement si la performance en validation ne s'améliore plus.
  - ReduceLROnPlateau : Réduit dynamiquement le learning rate en cas de stagnation.

PIPELINE DE MODELISATION SIMPLIFIÉ

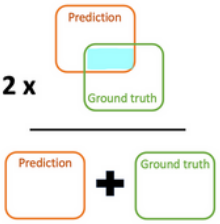
Image d'entrée → *prétraitement* → *Entrainement* → *Prédiction des masques*

- **Évaluation des performances**
  Métriques utilisées pour comparer Resnet 50 et ConvNext V2 :
  - IoU (Intersection over Union) : Mesure la superposition entre les prédictions et les masques réels.
  - Dice Score (F1) : Donne une indication sur la qualité de la segmentation en pondérant les faux négatifs et faux positifs.
  - Loss utilisée : Total Loss (Dice Loss + CrossEntropy Loss)
    - La Dice Loss compense les classes déséquilibrées.
    - La CrossEntropy Loss affine la classification des pixels en probabilités.

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



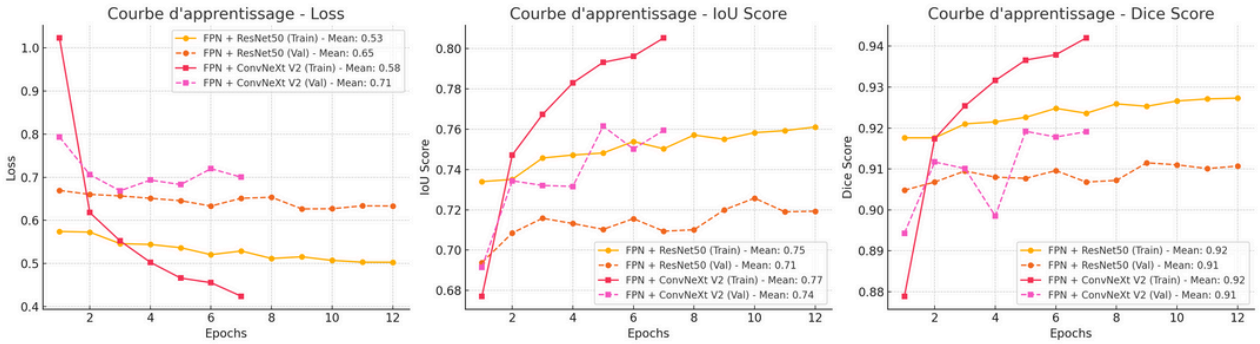
$$Dice = \frac{2\,X\,Area\,of\,overlap}{Total\,area} = \frac{2 \times}{}$$



SYNTHÈSE DES RÉSULTATS

1. Comparaison des performances globales

L'entraînement a montré une convergence plus rapide pour FPN + ConvNeXt V2, stoppant après 7 epochs contre 12 epochs pour FPN + ResNet50.



** Comparatif des courbes d'apprentissage*

Observations principales sur les courbes d'apprentissage

- *Loss*
  - ConvNeXt V2 commence avec une loss plus élevée, mais diminue plus rapidement.
  - La loss finale en validation est légèrement plus haute (0.71 vs 0.65 pour ResNet50), indiquant une régularisation plus forte.

- *IoU Score*
  - ConvNeXt V2 atteint un IoU plus élevé (0.74 vs 0.71), indiquant une meilleure segmentation globale.
  - Le modèle atteint rapidement son plateau, contrairement à ResNet50 qui stagne plus tôt.

- *Dice Score*
  - Très similaire pour les deux modèles (~0.91), indiquant que les contours des objets sont bien détectés dans les deux cas.

Interprétation

- ConvNeXt V2 converge plus vite et généralise mieux.
- La loss plus haute en validation suggère un modèle plus régularisé, avec moins de sur-apprentissage.

2.Analyse des prédictions par classe

L'analyse des pixels prédits montre que ConvNeXt V2 améliore la segmentation des petites classes et textures complexes.

Classe	ResNet50 (%)	ConvNeXt V2 (%)	Différence (%)
Route	69.18	73.03	3.85
Trottoir	98.28	96.27	-2.01
Bâtiment	91.7	89.61	-2.09
Végétation	38.29	76.77	38.48
Ciel	94.4	92.31	-2.09
Véhicule	95.84	98.1	2.26
Personne	76.84	93.6	16.76
Mobilier urbain	92.22	94.16	1.94

** Pourcentage des pixels correctement prédits*

Observations principales

- *Gains significatifs sur les objets détaillés :*
  - *Végétation : +38.5% → Meilleure reconnaissance des textures complexes.*
  - *Personnes : +16.7% → Meilleure segmentation des silhouettes.*
- *Classes avec peu de variations :*
  - *Mobilier urbain : +2% → Meilleure distinction des objets urbains.*
  - *ConvNeXt V2 surpasse légèrement ResNet50 sur les routes (+3.85%) mais perd quelques points sur trottoirs et bâtiments (-2%), probablement en raison d'un équilibre différent des classes dans l'entraînement.*

Interprétation

- ConvNeXt V2 segmente bien mieux les classes difficiles, avec une énorme amélioration sur la végétation et les piétons.
- Les grandes classes restent bien segmentées, mais ResNet50 fait légèrement mieux sur les trottoirs et bâtiments.

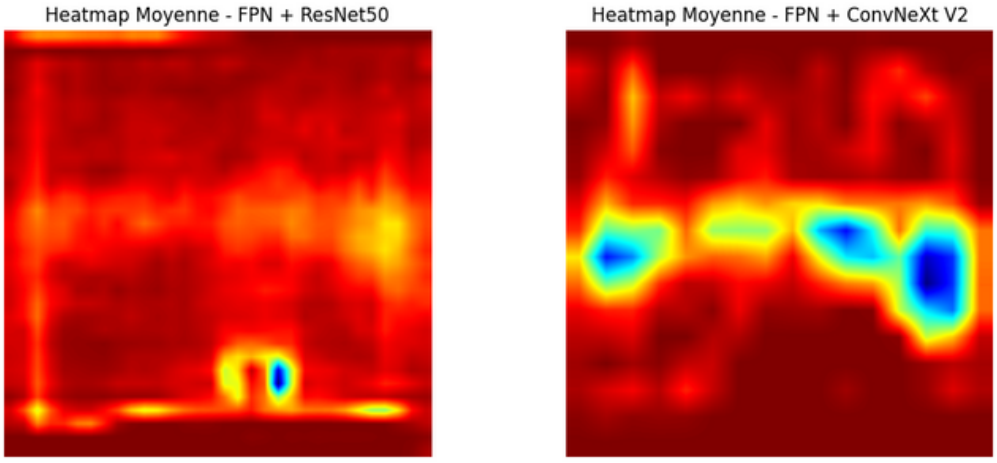
Conclusion

Les résultats montrent que FPN + ConvNeXt V2 surpasse FPN + ResNet50, en particulier sur la rapidité d'apprentissage et la précision des objets complexes
ConvNeXt V2 offre un gain significatif sur la segmentation des objets fins et difficiles, tout en maintenant une bonne robustesse sur les classes principales.
Il remplace avantageusement ResNet50 pour une segmentation plus fine, sans coût computationnel excessif.

ANALYSE DE LA FEATURE IMPORTANCE GLOBALE ET LOCALE

1. Importance globale des features (Heatmaps moyennes)

L'analyse de l'importance des features globales permet de comprendre où le modèle focalise son attention en moyenne lors des prédictions. J'ai utilisé Grad-CAM sur un ensemble d'images pour générer une heatmap moyenne des activations de FPN + ResNet50 et FPN + ConvNeXt V2.



** Heatmap moyenne des activations des modèles*

Résultats des heatmaps globales

- *FPN + ResNet50*
  - Activation plus diffuse et désorganisée, sans focus clair sur des objets spécifiques.
  - Certains éléments (routes, bâtiments) sont faiblement détectés.
  - Résultats suggérant une moindre capacité à segmenter avec précision.

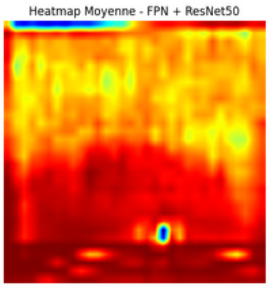
- *FPN + ConvNeXt V2*
  - Activations plus nettes et structurées, avec une attention portée sur les objets clés.
  - Détection claire des zones importantes comme les routes, véhicules et bâtiments.
  - Meilleure segmentation suggérée par cette répartition plus précise de l'attention.

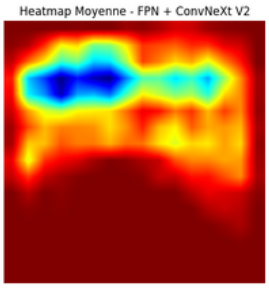
- Conclusion : ConvNeXt V2 montre une capacité à mieux cibler les structures et objets utiles à la segmentation, ce qui explique ses meilleurs scores sur les métriques d'évaluation.

2. Importance locale des features (Analyse par classe spécifique)

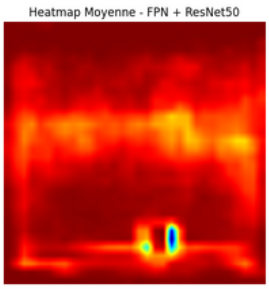
Nous avons approfondi l'analyse en appliquant Grad-CAM par classe, pour comparer la capacité de chaque modèle à détecter des catégories spécifiques. Deux classes ont été sélectionnées en raison de leurs fortes différences de précision dans les pixels prédits :

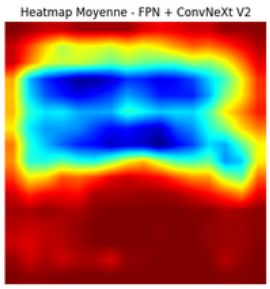
- 🌿 Végétation (Classe 3) : ConvNeXt V2 avait une précision supérieure de +38,48%.
- 🚶 Personnes (Classe 6) : ConvNeXt V2 avait une précision supérieure de +16,76%.





** Heatmap class Végétation*





** Heatmap class Personne*

Résultats des heatmaps par classe

- *Végétation :*
  - *FPN + ResNet50 : Activations diffuses, la végétation n'est pas bien séparée des autres classes.*
  - *FPN + ConvNeXt V2 : Détection claire et bien définie des zones végétales (arbres, buissons), en accord avec la précision des pixels prédits.*
- *Personnes :*
  - *FPN + ResNet50 : Quasiment aucune activation, confirmant ses faibles performances sur cette classe.*
  - *FPN + ConvNeXt V2 : Détection cohérente des piétons, avec un focus clair sur les silhouettes humaines.*

- Conclusion : ConvNeXt V2 apporte une meilleure discrimination entre les classes et capte mieux les structures complexes comme les végétaux et les piétons, contrairement à ResNet50 qui peine à les identifier.

Synthèse et impact sur la segmentation

L'analyse des heatmaps Grad-CAM confirme les différences observées dans les pixels prédits.

- ConvNeXt V2 se montre plus performant pour capter les objets et textures complexes, ce qui justifie son meilleur IoU.
- ResNet50 manque de précision dans ses activations, ce qui peut expliquer ses performances plus faibles sur certaines classes.

LIMITES ET AMÉLIORATIONS POSSIBLES

Limites identifiées

- Temps d'inférence plus long que ResNet50 ⏳
  - ConvNeXt V2 est ≈1.66x plus lent que ResNet50 (34.96 ms vs 21.10 ms).
  - La segmentation est plus précise, mais le temps d'inférence augmente légèrement.

- Sensibilité aux classes sous-représentées ⚖️
  - Malgré la pondération des classes dans la loss function, certaines restent moins bien segmentées, notamment les classes rares ou à faible contraste.
  - Ex : les objets petits ou éloignés (mobilier urbain, piétons en arrière-plan) pourraient encore être mieux détectés.

```
Comparaison des temps d'inférence :
➡️ FPN + ResNet50 : 21.10 ms
➡️ FPN + ConvNeXt V2 : 34.96 ms
```

** Temps d'inférence des modèles*

Améliorations envisageables

- *Optimisation du modèle pour réduire le temps d'inférence*
  - *Tester ConvNeXt V2-Base ou Tiny, qui sont plus légers tout en conservant les bénéfices des améliorations structurelles.*
  - *Appliquer des techniques de quantization ou pruning pour alléger le modèle sans perte significative de performance.*

- *Utilisation d'un modèle pré-entraîné plus robuste*
  - *Tester une version plus avancée (ConvNeXt V2-Huge) pour voir si elle apporte un gain significatif en précision.*

- Amélioration de la segmentation des classes minoritaires
  - Tester une augmentation ciblée des données, en générant artificiellement des images contenant davantage d'objets sous-représentés (ex : augmenter artificiellement les piétons dans les scènes).

Ces résultats confirment que ConvNeXt V2 constitue une amélioration notable sur ResNet50, tout en ouvrant des pistes d'optimisation futures pour une intégration en production.