Université Sultan Moulay Slimane

Ecole Nationale des Sciences Appliquées

Khouribga

Rapport de Projet

Spécialité : Informatique et Ingénierie des Données

Réalisé par :

BOULIDAM Abdellah
BOUMLIK Youness
EL HOUARI Zakaria
EL WARRAQI Imane
RHANNOUCH Nassima

Segmentation faciale par Deep Learning : Analyse comparative des architectures pour la détection des traits du visage

---

Encadré par :

Pr.ABOUTABIT Noureddine, ENSA Khouribga

Année universitaire 2024-2025

Table des matières

Liste des tableaux

Liste des figures :

Remerciements

Nous tenons à exprimer notre sincère gratitude à notre professeur du module Vision par Ordinateur, Monsieur ABOUTABIT, pour son encadrement attentif et la qualité de son enseignement. Son dévouement et sa disponibilité nous ont permis d'approfondir nos compétences en traitement d'images et en apprentissage automatique, tout en maîtrisant les subtilités de l'analyse et de la segmentation des traits faciaux.Nous souhaitons également remercier toutes les personnes ayant contribué, directement ou indirectement, à la réalisation de ce projet, ainsi que nos collègues pour leur précieuse collaboration

Merci infiniment !

1 Introduction

1 Contexte et problématique de la segmentation faciale

La segmentation des parties du visage est une tâche fondamentale en vision par ordinateur qui consiste à identifier et délimiter précisément des zones spécifiques telles que les yeux, le nez, la bouche et les oreilles dans une image. Cette technique joue un rôle important dans de nombreuses applications pratiques, notamment :

— **La reconnaissance faciale** : essentielle pour l'authentification biométrique et le contrôle d'accès sécurisé.

— **La réalité augmentée** : utilisée pour des applications telles que les filtres et effets faciaux personnalisés en temps réel.

— **Les systèmes de sécurité** : utiles pour la surveillance et l'analyse des expressions faciales.

Cependant, la segmentation faciale est confrontée à des défis techniques importants qui rendent sa mise en œuvre complexe, parmi lesquels :

— **La variabilité des visages** : différences significatives selon l'âge, le genre, l'origine ethnique et la morphologie.

— **Les conditions d'éclairage** : des variations de luminosité peuvent altérer la qualité des images et compliquer la segmentation.

— **Les expressions faciales** : les déformations dynamiques du visage rendent la segmentation plus difficile.

— **Les angles de vue** : des prises de vue sous différents angles modifient l'apparence et la géométrie des visages.

— **Les occlusions** : la présence d'éléments tels que des lunettes, masques, barbes ou mains peut occlure certaines parties du visage.

Ces contraintes nécessitent des solutions robustes et des modèles capables de traiter efficacement ces variations pour fournir une segmentation fiable et précise.

2 Contribution attendue

Ce projet vise à apporter plusieurs contributions significatives dans le domaine de la segmentation des parties du visage :

— **Optimisation des architectures existantes** : adapter et optimiser les modèles U-Net, PSPNet et SegNet pour améliorer leur précision et efficacité dans des scénarios variés de segmentation faciale. Une évaluation approfondie sera réalisée sur le dataset LaPa pour exploiter pleinement les capacités des modèles.

— **Analyse des performances sur le dataset LaPa** : réaliser une évaluation détaillée pour identifier les forces et limites des modèles face aux défis posés par les variations d'expressions, d'éclairage et d'angles de vue.

— **Proposition de méthodologies robustes** : développer des méthodologies reproductibles et adaptables pour la segmentation des parties du visage. Ces méthodologies pourront être réutilisées avec d'autres modèles et datasets, contribuant ainsi aux futures avancées dans ce domaine.

2

État de l'Art et Dataset

1 Approches Deep Learning

1.1 Revue des architectures pertinentes

Dans cette section, nous présentons trois architectures majeures de deep learning qui ont marqué l'évolution de la segmentation sémantique d'images.

U-Net

U-Net, proposée par ronneberger2015unet en 2015, représente une avancée majeure dans le domaine de la segmentation d'images. Cette architecture se distingue par sa structure symétrique en forme de "U", composée d'un chemin contractant (encoder) et d'un chemin expansif (decoder). La particularité de U-Net réside dans ses connexions skip (connexions résiduelles) qui relient directement les couches de l'encoder aux couches correspondantes du decoder. Ces connexions permettent de combiner les informations de localisation précise des premières couches avec les caractéristiques sémantiques extraites dans les couches profondes.

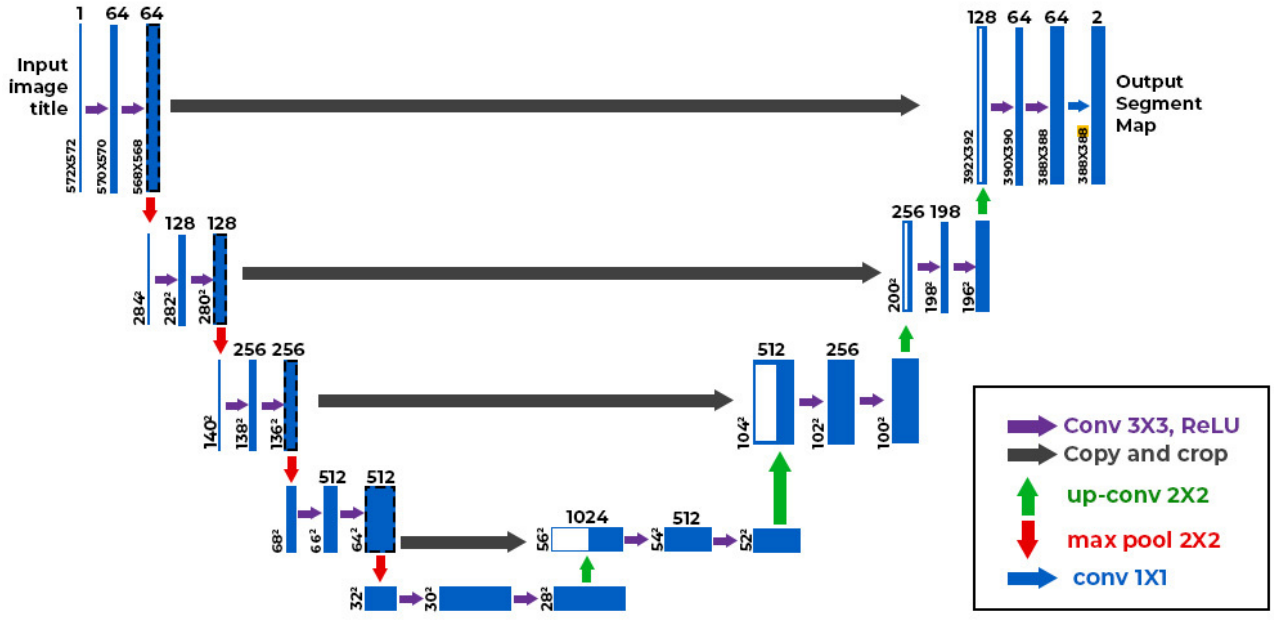


Figure 2.1 – Architecture U-Net

SegNet

SegNet, introduit par badrinarayanan2017segnet en 2017, est une architecture encoder-decoder conçue initialement pour la segmentation de scènes routières. Son innovation principale réside dans sa méthode d'up-sampling pendant la phase de décodage. Contrairement aux approches classiques, SegNet mémorise les indices de max pooling durant la phase d'encoding et les réutilise pour effectuer le unpooling dans le decoder. Cette approche permet une reconstruction plus précise des frontières des objets tout en maintenant une efficacité computationnelle.

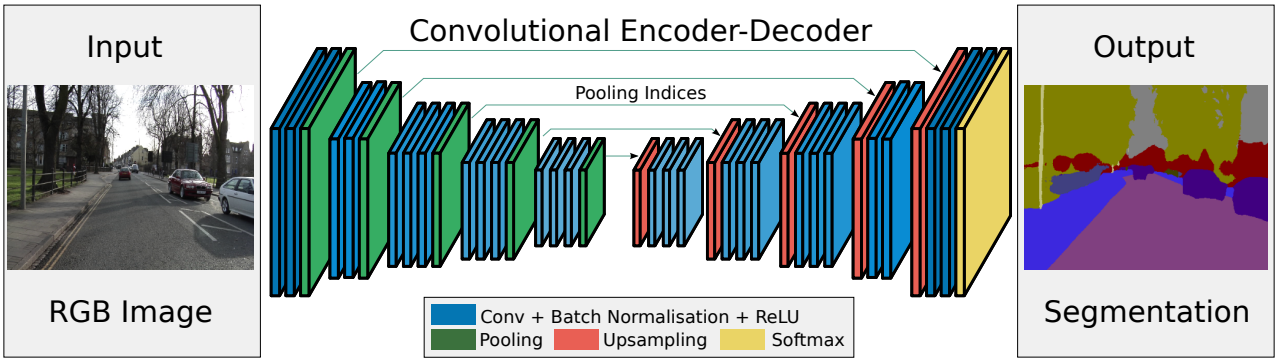


Figure 2.2 – Architecture SegNet

PSPNet

Pyramid Scene Parsing Network (PSPNet), développé par zhao2017pspnet en 2017, introduit un concept novateur : le module de pooling pyramidal. Ce module analyse l'image à différentes échelles en parallèle, permettant ainsi de capturer des informations contextuelles à différents niveaux de granularité. L'architecture utilise un réseau profond (généralement ResNet) comme backbone, suivi du module pyramidal qui agrège les informations à différentes échelles avant la prédiction finale.

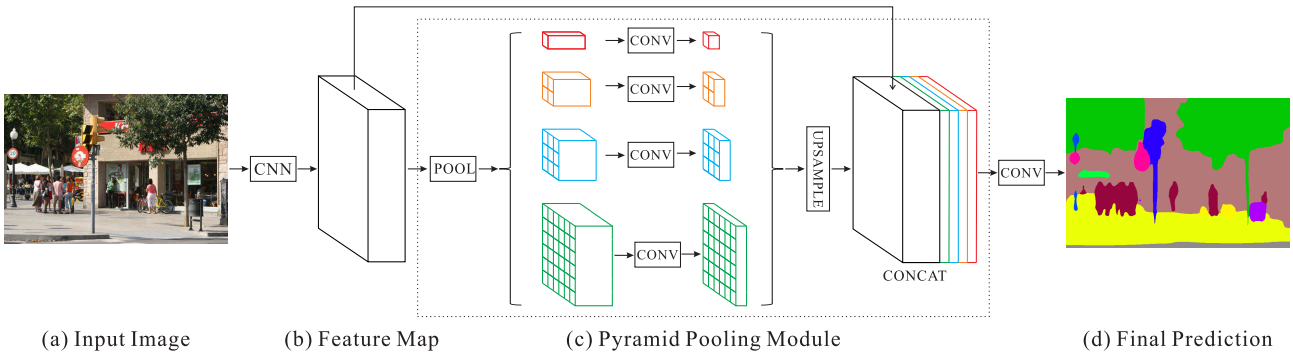


Figure 2.3 – Architecture PSPNet

1.2 Comparaison des approches existantes

Les architectures présentées ci-dessus se distinguent par leurs approches spécifiques de la segmentation d'images. U-Net excelle dans la préservation des détails fins grâce à ses connexions skip, ce qui en fait un choix privilégié pour les applications médicales et autres tâches nécessitant une précision élevée. SegNet optimise l'efficacité computationnelle tout en maintenant une bonne précision, ce qui le rend particulièrement adapté aux applications temps réel ou aux systèmes avec des ressources limitées. PSPNet, avec son approche multi-échelle, offre une excellente compréhension du contexte global, particulièrement utile pour la segmentation de scènes complexes.

Table 2.1 – Tableau comparatif des architectures de segmentation d'images

Caractéristiques	**U-Net**	**SegNet**	**PSPNet**
Année de publication	2015	2017	2017
Architecture principale	Encoder-Decoder avec skip connections	Encoder-Decoder avec indices de pooling	CNN avec module de pooling pyramidal
Particularité	Connexions symétriques entre encoder et decoder	Mémorisation des indices de max pooling	Module de pooling pyramidal multi-échelle
Points forts	— Préservation des détails fins — Efficace avec peu de données — Bonne gestion des contours	— Efficacité computationnelle — Mémoire optimisée — Bon pour temps réel	— Excellente capture du contexte — Gestion multi-échelle — Performance sur grandes images
Points faibles	— Coût mémoire important — Temps de calcul élevé	— Moins précis sur les détails fins — Sensible à la qualité des indices	— Plus complexe à entraîner — Ressources importantes requises
Usage recommandé	Segmentation précise nécessitant des détails fins	Applications temps réel avec ressources limitées	Scénarios complexes nécessitant contexte global

Le choix entre ces architectures dépend souvent du compromis entre plusieurs facteurs :

— La précision requise pour l'application visée

— Les ressources computationnelles disponibles

— Les contraintes de temps de traitement

— La taille et la nature des données d'entrée

— La complexité de la tâche de segmentation

Les avancées récentes dans ce domaine tendent vers des architectures hybrides qui combinent les forces

de ces différentes approches, ouvrant la voie à des solutions encore plus performantes et adaptables à des cas d'usage spécifiques.

2 Dataset LAPA

2.1 Caractéristiques et statistiques principales

Le dataset **LaPa(Landmark guided face Parsing dataset)** contient **22 176 images faciales**, réparties en trois ensembles : **entraînement (18 176 images)**, **validation (2 000 images)**, et **test (2 000 images)**. Il couvre une vaste gamme de variations en termes de **poses, expressions faciales, occlusions et conditions d'éclairage**. Chaque image est accompagnée de :

— Une carte d'annotation pixelisée décrivant **11 catégories de parties du visage** : peau, cheveux, sourcils (gauche et droit), yeux (gauche et droit), nez, lèvres (supérieure et inférieure), bouche interne, et arrière-plan.

— **106 points de repère faciaux précis**, facilitant des applications complémentaires comme l'alignement ou le maquillage virtuel.

Les annotations sont générées grâce à un processus semi-automatisé pour garantir une précision élevée et une efficacité dans le marquage des données. Les masques segmentés sont au format **PNG**.

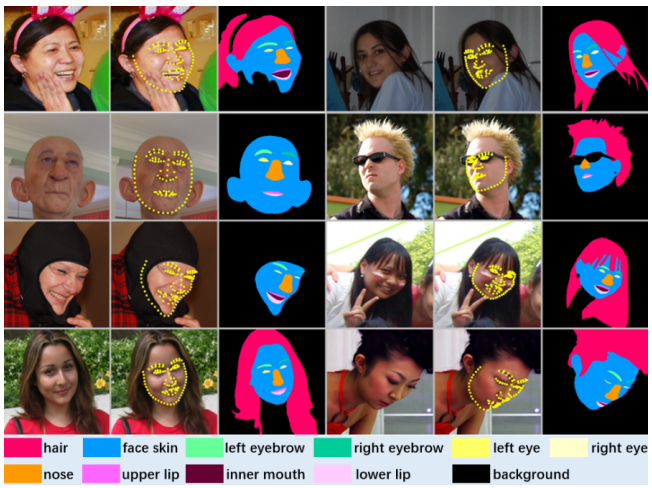


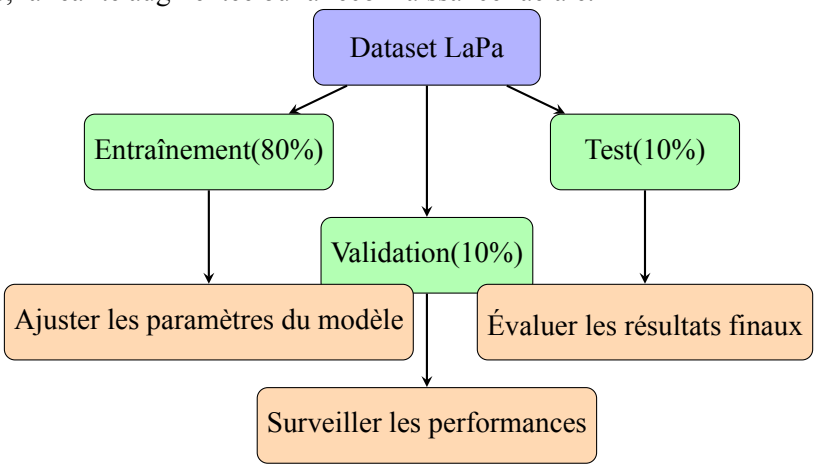
Figure 2.4 – Exemples d'annotations de l'ensemble de données LaPa.

2.2 Préparation pour la segmentation

Les étapes de traitement du dataset comprennent :

1. **Division des ensembles** :
   - **80% (18 176 images)** pour l'entraînement.
   - **10 % (2 000 images)** pour la validation.
   - **10 % (2 000 images)** pour le test.
2. **Rôle des ensembles** :
   - **Entraînement :** Ajustement des paramètres du modèle pour apprendre à segmenter les 11 catégories de parties du visage.
   - **Validation :** Évaluation en continu pour éviter le surapprentissage et optimiser les hyperparamètres.
   - **Test :** Évaluation finale et objective des performances sur des données inconnues.
3. **Utilisation des annotations** :
   - Les cartes d'étiquettes au niveau pixel permettent des tâches de segmentation.
   - Les points de repère sont exploités pour des tâches supplémentaires comme la détection ou l'alignement.

Ces caractéristiques font du dataset LaPa un choix privilégié pour des applications comme la segmentation faciale, la réalité augmentée ou la reconnaissance faciale.



Description du schéma :

Le schéma illustre la répartition des données du dataset LaPa en trois ensembles distincts pour assurer un entraînement rigoureux et une évaluation objective du modèle. L'ensemble d'entraînement, représentant 80% des données, est utilisé pour ajuster les paramètres du modèle et améliorer ses capacités d'apprentissage. L'ensemble de validation, correspondant à 10% des données, permet de surveiller les performances du modèle pendant l'entraînement, d'optimiser les hyperparamètres et d'éviter le surapprentissage. Enfin, l'ensemble de test, qui constitue 10% des données, sert à évaluer les résultats finaux et à vérifier la capacité de généralisation du modèle sur des données inédites. Cette répartition équilibrée garantit un développement méthodique et efficace du modèle.

3

Méthodologie

Dans le chapitre précédent, nous avons exploré les différentes architectures de deep learning pertinentes pour la segmentation faciale, ainsi que le dataset LaPa que nous allons utiliser. Dans ce chapitre, nous détaillons la méthodologie adoptée pour la construction et l'évaluation de nos modèles de segmentation faciale. Nous présentons les différentes étapes suivies, du prétraitement des données aux protocoles d'entraînement, en passant par les choix architecturaux et les métriques d'évaluation.

1 Prétraitement des données

Le prétraitement des données est une étape cruciale pour assurer une performance optimale de nos modèles. Nous avons appliqué les transformations suivantes aux images et aux masques :

— **Redimensionnement :** Toutes les images ont été redimensionnées à une taille de 256x256 pixels en utilisant l'interpolation bilinéaire. Ce choix a été fait pour réduire la complexité du modèle et les ressources de calcul requises, tout en préservant les caractéristiques essentielles du visage. L'interpolation bilinéaire a été choisie car elle est un compromis entre qualité de l'image et efficacité du traitement.

— **Normalisation :** Les valeurs de pixels des images ont été normalisées en divisant chaque pixel par 255, ce qui permet d'obtenir des valeurs comprises entre 0 et 1. Cela assure que toutes les valeurs des pixels sont mises à la même échelle, facilitant ainsi l'apprentissage du modèle. Les masques de segmentation n'ont pas subi de normalisation en tant que telle, et ont été transformées en données de type int32 pour faciliter le calcul de la métrique IoU.

— **Augmentation des données :** Pour améliorer la robustesse du modèle et éviter le surapprentissage, nous avons utilisé l'augmentation de données en appliquant aléatoirement les transformations suivantes : des rotations jusqu'à 10 degrés, des zooms jusqu'à 10 % , des translations de 10 % et des retournements horizontaux. Ces transformations permettent au modèle d'apprendre à être invariant à une variété de poses et de conditions d'éclairage. Les transformations d'augmentation ont été appliqués de manière aléatoire.

— **Format des données :** Les masques de segmentation sont encodées en niveaux de gris ou chaque niveau de gris représente une étiquette de classe.

2 Choix et configuration des modèles

Pour réaliser notre objectif de segmentation du visage, nous avons choisi de travailler avec trois architectures de réseaux neuronaux convolutifs différentes : U-Net, PSPNet et SegNet.

— **U-Net :** Nous avons choisi U-Net car c'est un modèle couramment utilisé dans le domaine de la segmentation d'image, de par son architecture encodeur-décodeur avec des connexions sautées qui facilite la récupération d'informations à différentes échelles. Nous avons implémenté une version de U-Net avec un encodeur basé sur MobileNetV2, ce qui permet une extraction de features efficace en ayant un nombre limité de paramètres. Le nombre de filtres de la couche de base était de 64 avec des facteurs de multiplications de 2 à chaque bloc. Nous avons aussi utilisé une taille de noyau de 3 et une activation ReLU après chaque couche de convolution et de déconvolution. Nous n'avons pas utilisé de dropout dans ce modèle car il a été montré à être inefficace pour les modèles de segmentation.

— **PSPNet :** Le choix de PSPNet repose sur sa capacité à capturer des informations à différentes échelles grâce à son module de pooling pyramidal. Nous avons utilisé un encodeur basé sur MobileNetV2. Nous avons gardé le nombre de filtres à 256 pour toutes les couches, sauf pour la dernière upsample où nous avons utilisé 64. Le module pyramidal consiste en un pooling moyenne suivi d'une convolution 1x1, et d'une interpolation bilinéaire pour ramener la feature map à la taille initiale.

— **SegNet :** SegNet est utilisé pour son efficacité dans les tâches de segmentation en temps réel. Son architecture utilise les index des opérations de max pooling lors de la déconvolution, ce qui rend les paramètres optimisés dans le décodeur. Nous avons utilisé un facteur de 64 lors de la première couche et multiplié par 2 jusqu'à la cinquième couche. Une activation ReLU est utilisée après chaque couche de convolution et déconvolution.

3 Protocole d'entraînement

Pour entraîner nos modèles, nous avons utilisé le protocole suivant :

— **Optimiseur :** Nous avons choisi l'optimiseur Adam pour sa capacité à s'adapter aux données et à converger rapidement, ce qui permet de diminuer le temps d'entraînement et d'obtenir de bonnes performances avec différents types d'architectures de réseaux de neurones.

— **Taux d'apprentissage :** Le taux d'apprentissage initial était de 1e-4, et nous avons utilisé un callback **ReduceLROnPlateau** pour le réduire d'un facteur 0,1 après 5 époques de stagnation de la **val_loss**. Cette approche permet d'ajuster dynamiquement le taux d'apprentissage et d'améliorer la convergence du modèle.

— **Fonction de perte :** La fonction de perte utilisée est la **categorical_crossentropy**, qui est adaptée aux tâches de classification multi-classes et permet de minimiser la différence entre la prédiction du modèle et la vérité terrain à chaque pixel, et cela avec le format one-hot du masque.

— **Nombre d'époques et taille du batch :** Nous avons entraîné les modèles sur 10 époques avec une taille de batch de 8. La taille du batch a été réduite en raison de la contrainte de nos ressources de calcul, afin de faire rentrer le processus d'entraînement dans la mémoire GPU de notre environnement Kaggle.

— **Callbacks :** Les callbacks suivants ont été utilisés pour améliorer l'entraînement :

- **ModelCheckpoint** pour enregistrer les meilleurs poids du modèle en fonction de la **val_loss** (métrique de performance de validation).
- **ReduceLROnPlateau** pour diminuer le taux d'apprentissage de manière adaptative afin d'améliorer la convergence du modèle.
- **CSVLogger** pour enregistrer l'évolution des métriques pendant l'entraînement dans un fichier CSV et de pouvoir suivre l'évolution des performances.
- **EarlyStopping** pour arrêter l'entraînement si la **val_loss** ne s'améliore pas après 10 époques, tout en restaurant les meilleurs poids.

— **Précision Mixte :** L'entraînement en précision mixte a été utilisé pour optimiser l'entraînement et réduire la consommation mémoire. Cette technique permet d'entraîner des modèles de deep learning en utilisant des données à faible précision, ce qui réduit l'utilisation de mémoire, augmente l'efficacité du calcul et accéleère l'entraînement.

4 Métriques d'évaluation

Pour évaluer la performance de nos modèles de segmentation, nous avons utilisé les métriques suivantes :

— **IoU (Intersection over Union) :** L'IoU est une mesure de chevauchement entre les masques de prédiction et les masques de vérité terrain. Il est calculé comme le ratio de l'intersection sur l'union des deux masques et fournit une évaluation de la précision de la segmentation.

— **Dice Coefficient :** Le coefficient de Dice mesure la similarité entre les ensembles de prédictions et de vérités terrain. Il est particulièrement utile pour les tâches de segmentation où l'équilibre entre les classes est faible. Sa formule considère les vrais positifs deux fois.

— **Précision (Precision) :** La précision est le rapport de prédictions positives qui sont réellement positives par rapport à toutes les prédictions positives. Il est particulièrement utile pour évaluer la capacité du modèle à ne pas faire de fausses identifications.

— **Rappel (Recall) :** Le rappel est le rapport de cas positifs qui sont correctement identifiés par rapport à tous les cas positifs. Il est particulièrement utile pour évaluer la capacité du modèle à identifier tous les cas positifs.

4 Résultats et Discussion

1 Performances comparatives

Dans cette section, nous analysons les performances des modèles de segmentation à travers plusieurs indicateurs, notamment l'évolution de la *training loss* et de l'IoU (*Intersection over Union*) au cours de l'entraînement. Nous présentons également les métriques finales, telles que le Dice coefficient, la précision et le rappel, afin d'évaluer globalement la qualité des prédictions des modèles. Ces visualisations et résultats permettent une comparaison détaillée des modèles et de leur capacité à généraliser sur les données.

Table 4.1 – Comparaison des métriques finales pour les différents modèles

Modèle	Loss	Dice	IoU	Precision	Recall
UNet	0.072	1.956	0.978	3.186	3.166
PSPNet	0.096	1.939	0.970	3.163	3.140
SegNet	0.069	1.958	0.979	3.192	3.171

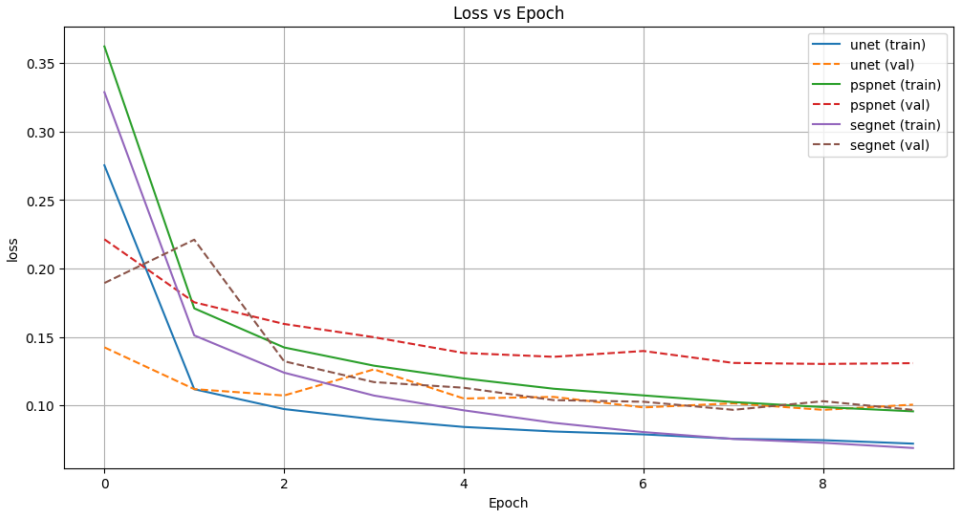


Figure 4.1 – Évolution de la perte (Loss) pendant l'entraînement

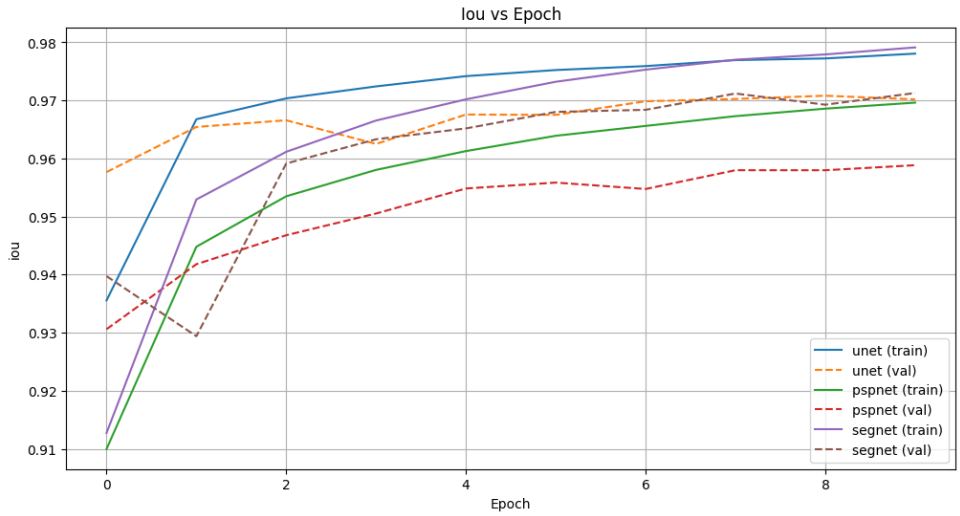


Figure 4.2 – Évolution de l'IoU pendant l'entraînement

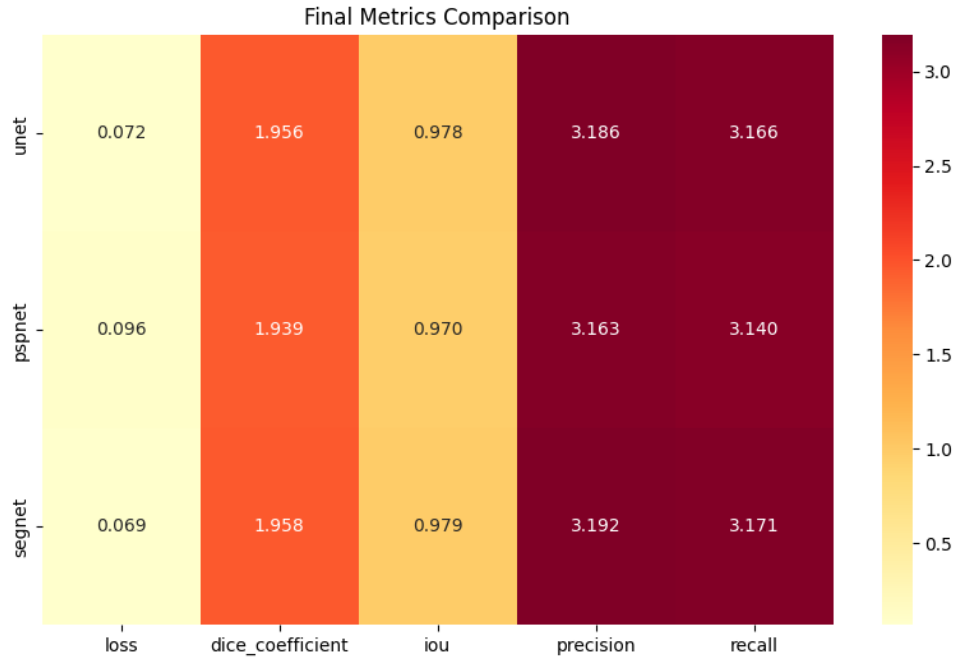


Figure 4.3 – Heatmap des métriques finales pour chaque modèle

1.1 Analyse comparative

Notre étude comparative des trois architectures de segmentation (UNet, PSPNet et SegNet) sur le dataset LaPa révèle des performances intéressantes. SegNet démontre les meilleures performances globales avec un IoU de 0.979 et une perte minimale de 0.069. Ces résultats surpassent légèrement ceux d'UNet (IoU : 0.978, Loss : 0.072) et plus significativement ceux de PSPNet (IoU : 0.970, Loss : 0.096).

La convergence des modèles, visible dans les courbes d'apprentissage (Figure 4.1), montre qu'UNet atteint rapidement de bonnes performances, tandis que SegNet, malgré une instabilité initiale, finit par obtenir les meilleurs résultats. PSPNet, bien que performant, nécessite plus d'époques pour converger.

Comme le montre la Figure 4.3, SegNet offre un excellent compromis entre précision (3.192) et rappel (3.171), suggérant une segmentation robuste et fiable des caractéristiques faciales.

2 Analyse qualitative

L'Erreur Quadratique Moyenne (RMSE) est une métrique clé utilisée pour évaluer la performance des modèles de régression et de segmentation d'image. Elle mesure l'écart moyen entre les prédictions des modèles et les valeurs réelles (vérité terrain). Plus la valeur du RMSE est faible, plus les prédictions du modèle sont précises, ce qui reflète une meilleure performance en termes de segmentation.

Dans cette section, nous présentons une analyse des prédictions de trois modèles de segmentation (U-Net, PSPNet, et SegNet) sur une même image test. Nous comparons leurs performances en calculant les valeurs du RMSE pour chacun des modèles, tout en montrant visuellement leurs prédictions et la vérité terrain (masque de base) pour une meilleure compréhension de leurs capacités respectives.

Figure 4.4 – Image originale

Figure 4.5 – Masque original

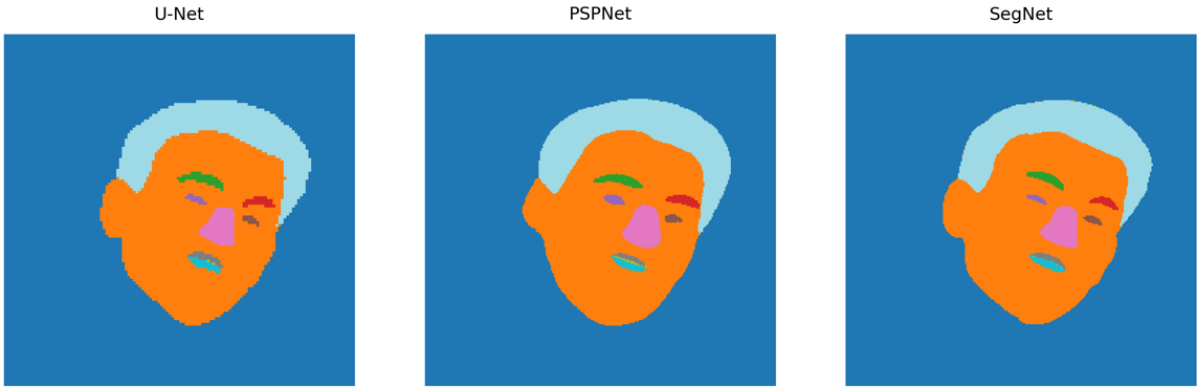


Figure 4.6 – Masques prédits par les modèles (U-Net, PSPNet, SegNet)

Modèle	RMSE
U-Net	2.7421
PSPNet	2.8387
SegNet	2.7414

Table 4.2 – Erreurs quadratiques moyennes (RMSE) pour chaque modèle de segmentation.

2.0 Interprétation des résultats

Les résultats RMSE indiquent ce qui suit :

— Le modèle **U-Net** a obtenu un RMSE de 2.7421, ce qui signifie que la différence moyenne entre le masque prédit et le masque réel est de 2.7421 unités.

— Le modèle **PSPNet** a un RMSE de 2.8387, indiquant que ses prédictions sont légèrement moins précises que celles du modèle U-Net.

— Le modèle **SegNet** a un RMSE de 2.7414, ce qui est quasiment identique à celui du modèle U-Net, indiquant une performance comparable.

Ainsi, bien que **PSPNet** affiche un RMSE légèrement plus élevé que **U-Net** et **SegNet**, la différence n'est pas significative. Les modèles **U-Net** et **SegNet** montrent des performances presque identiques en termes de

segmentation, avec un écart de seulement 0.0007 dans les valeurs de RMSE.

2.1 Conclusion

Les modèles **U-Net** et **SegNet** sont les plus performants pour cette tâche de segmentation d'images, avec des performances quasiment égales. **PSPNet** présente une légère diminution de performance, mais les différences ne sont pas suffisamment importantes pour remettre en question son efficacité.

3 Discussion des cas d'échec

3.1 Évaluation expérimentale des cas d'échec par occlusion

Dans cette partie, nous avons réalisé une étude comparative des trois architectures de segmentation (U-Net, PSPNet et SegNet) face aux problèmes d'occlusion. Pour évaluer leur comportement, nous avons utilisé une image de visage présentant des zones d'occlusion naturelles.

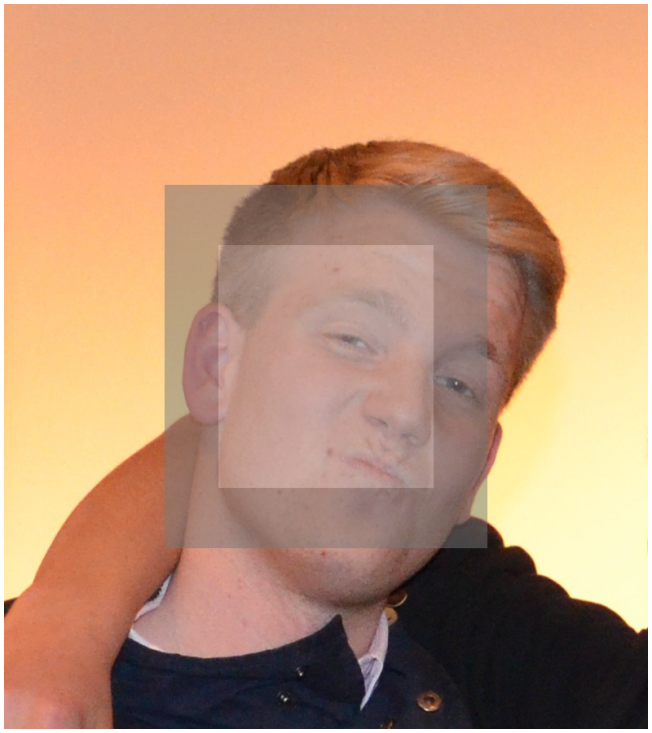


Figure 4.7 – Image partiellement occlus pour tester la robustesse des modèles

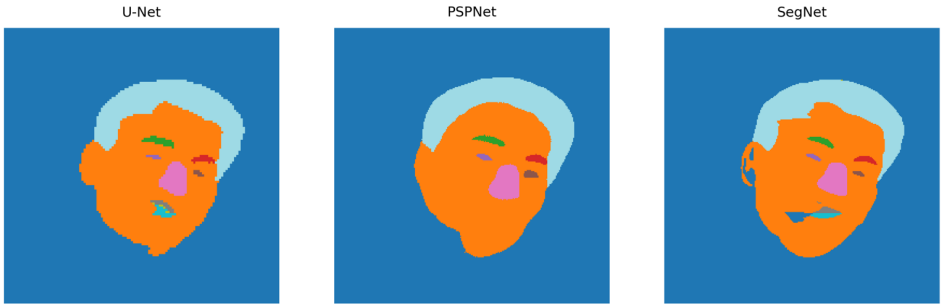


Figure 4.8 – Segmentation des architectures face à une occlusion

3.2 Analyse comparative des résultats

Les résultats obtenus montrent des différences significatives dans la gestion des occlusions par les trois modèles :

- **U-Net** :
  - L'occlusion (le carré gris) est mal gérée : U-Net semble interpoler de manière incorrecte les classes sous la région masquée.
  - Certaines parties (comme la mâchoire et la joue) semblent mal segmentées ou assignées à des classes incorrectes.
  - **Échec** : U-Net échoue principalement à traiter correctement les parties du visage masquées par l'occlusion, ce qui mène à une segmentation incohérente.
- **PSPNet** :
  - PSPNet semble mieux gérer le contexte global, mais la zone occluse reste problématique (ignorée ou mal classifiée).
  - Les bords du visage sont approximatifs, et certaines classes (comme les cheveux) semblent fusionner avec l'arrière-plan.
  - **Échec** : PSPNet présente des erreurs similaires à U-Net, bien qu'il soit légèrement meilleur pour capturer les formes globales. Cependant, il ne parvient pas à traiter précisément la zone masquée.
- **SegNet** :
  - SegNet semble avoir des difficultés supplémentaires par rapport aux deux autres modèles.
  - Les zones occluses sont encore moins bien segmentées, et le modèle semble attribuer des classes erronées aux parties masquées.
  - Les détails fins (comme les traits du visage) sont mal capturés.
  - **Échec** : SegNet montre des faiblesses dans la gestion de l'occlusion et des bordures, ce qui rend sa segmentation moins précise que celle des autres modèles.

3.3 Résumé

Modèle	Gestion de l'Occlusion	Détails Fins	Formes Globales
U-Net	Moyenne	Bonne	Moyenne
PSPNet	Moyenne	Moyenne	Bonne
SegNet	Faible	Faible	Faible

Table 4.3 – Comparaison des modèles U-Net, PSPNet et SegNet pour la segmentation faciale en termes d'occlusions, détails fins et formes globales.

Les trois modèles maintiennent une segmentation cohérente des éléments principaux du visage, mais présentent des variations subtiles dans leur gestion des zones de transition et des occlusions partielles, illustrant ainsi leurs caractéristiques spécifiques dans le traitement des cas d'occlusion.

4 Forces et limites des approches

Après avoir appliqué ces trois architectures de segmentation sur notre base de données d'images faciales, nous avons pu observer différents comportements et résultats. Chaque architecture présente des caractéristiques distinctes qui influencent la qualité de la segmentation obtenue. Le tableau 4.4 synthétise les forces et les limites observées pour chaque approche. Nos observations se basent non seulement sur les aspects théoriques de ces architectures, mais aussi sur les résultats pratiques obtenus lors de nos expérimentations. Cette analyse comparative nous permet de mieux comprendre les avantages et les inconvénients spécifiques de chaque architecture dans le contexte de la segmentation faciale.

Table 4.4 – Forces et limites des approches testées sur la segmentation faciale

Architecture	Forces observées	Limites observées
U-Net	— Meilleure préservation des contours du visage — Bonne détection des traits fins (sourcils) — Conservation fidèle de la forme générale du visage — Segmentation détaillée des éléments faciaux	— Présence d'artefacts de pixellisation — Granularité plus prononcée sur les bords — Effet de "crénelage" sur certaines zones — Consommation mémoire importante
PSPNet	— Bonne capture du contexte global du visage — Segmentation plus lisse et uniforme — Bonne gestion des proportions faciales — Moins d'artefacts de pixellisation	— Légère perte de précision sur les détails fins — Complexité d'implémentation plus élevée — Temps d'inférence plus important
SegNet	— Rapidité d'exécution — Résultats relativement lisses — Bonne performance avec ressources limitées — Segmentation acceptable des principales zones	— Moins précis sur les détails fins — Certaines imprécisions sur les contours — Sensibilité aux variations d'échelle

5

Conclusion

Ce projet a porté sur la segmentation des parties du visage, un domaine clé en vision par ordinateur. Nous avons exploré trois architectures de segmentation d'images (UNet, SegNet, PSPNet), et avons pu mettre en lumière leurs forces et faiblesses respectives en fonction des besoins spécifiques en termes de précision, de ressources de calcul ou de contexte d'application. En s'appuyant sur le jeu de données LaPa, qui offre une riche diversité d'expressions faciales, d'angles de vue et de conditions d'éclairage, nous avons mis au point une méthodologie d'entraînement et d'évaluation pour nos modèles. Les performances des modèles ont été évaluées grâce à des métriques comme l'Intersection over Union (IoU) et le coefficient de Dice.

Nos résultats soulignent l'importance de sélectionner des architectures adaptées aux variations complexes des images faciales, et ont permis d'observer les avantages et les inconvénients de chacune. Nous avons remarqué, par exemple, que UNet possède de bonnes performances pour la segmentation fine des contours, tandis que PSPNet présente des avantages dans des tâches où il faut tenir compte du contexte global de l'image. Nous avons observé également que les modèles ont des difficultés à traiter les cas d'occlusion très importantes.

Bien que nos modèles aient atteint des résultats prometteurs, certaines limites persistent. La performance sur des images présentant des fortes occlusions reste un défi, et l'augmentation du temps d'entraînement et de la complexité du modèle pourrait améliorer les scores.

Pour les travaux futurs, plusieurs pistes d'amélioration peuvent être envisagées :

— L'utilisation d'autres architectures, en particulier celles basées sur les transformeurs

— L'intégration de mécanismes d'attention pour une meilleure compréhension des contextes

— L'amélioration des techniques d'augmentation de données afin d'améliorer les performances de généralisation des modèles

— L'entraînement des modèles sur plus d'époques et avec plus de ressources

En somme, ce projet a permis d'acquérir une compréhension plus approfondie des défis liés à la segmentation des parties du visage.