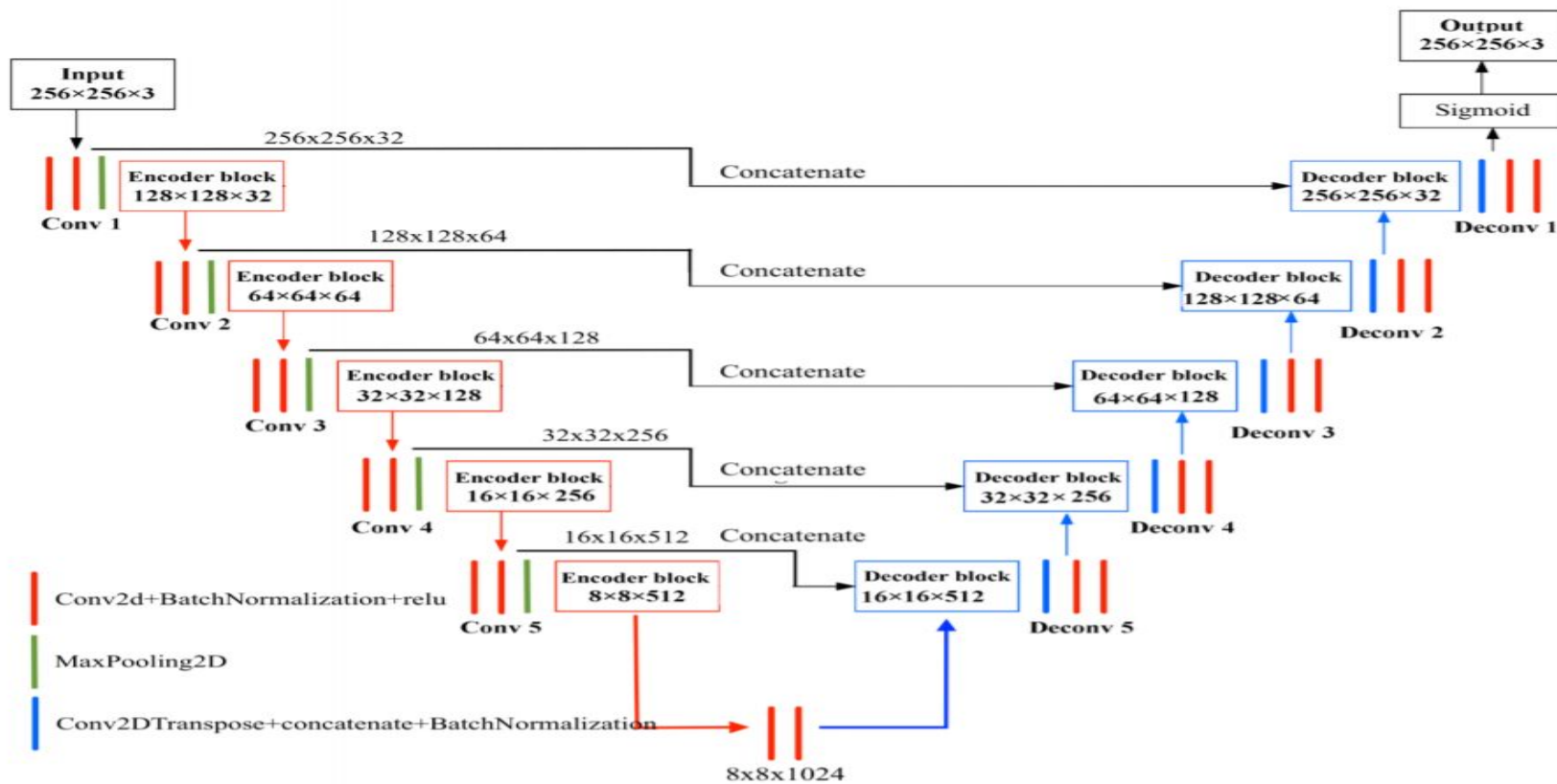


Exigence et context du projet

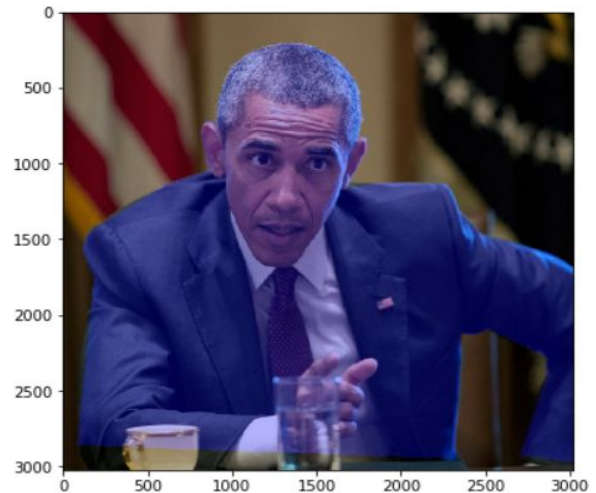
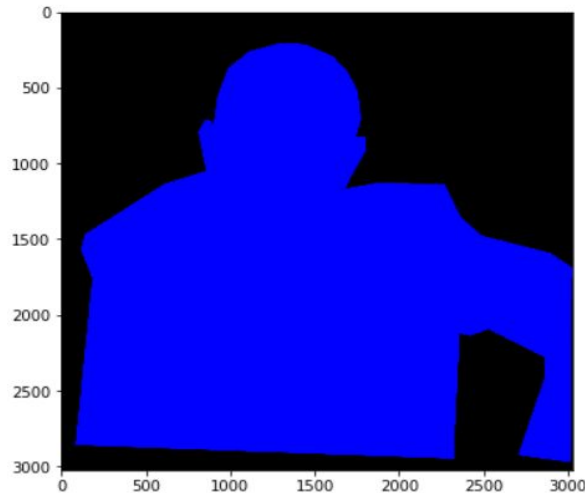
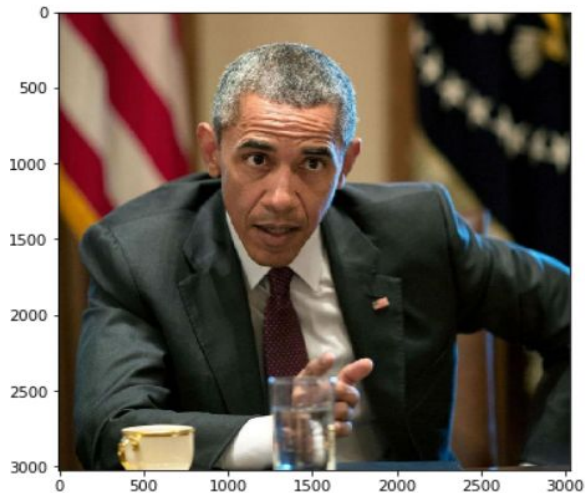
- Le but de ce projet est d'entraîner un modèle auto-encoder pour la segmentation des présidents de USA (Obama et Trump)
- La segmentation sémantique consiste à classifier chaque pixel et d'une image et lui attribuer un label
- Le modèle classifie deux classes Obama et Trump
- Le modèle est basé sur l'architecture U-Net

Architecture U-Net



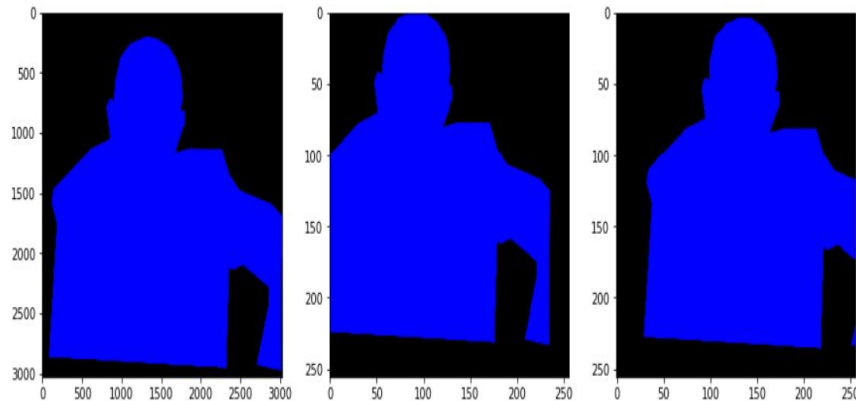
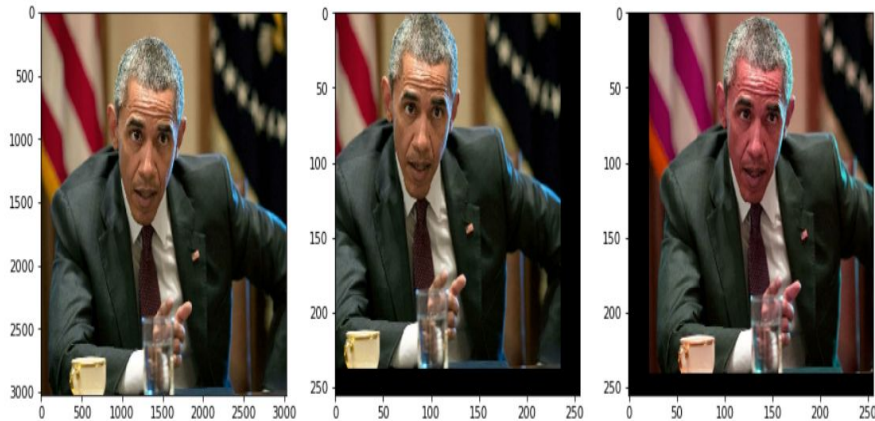
Réalisation

- J'ai scrappé 132 images par internet qui correspond aux deux classes Obama et Trump
- J'ai utilisé l'outil VGG Image Annotator pour segmenter les photos manuellement, cet outil génère un fichier csv qui contient les coordonnées des sommets des polygones
- J'ai créé les masks en utilisant OpenCV en se basant sur le fichier csv



Réalisation

- Le but du data pipeline est d'appliquer plusieurs transformations (redimensionnement, translation horizontale/ verticale, effet miroir, hue) sur les images et les masks pour augmenter la dataset mais aussi pour réduire les effets du overfitting
- L'avantage du data pipeline c'est que chaque epoch, on applique des transformations aléatoire sur les données ce qui fait que le modèle ne revoit jamais la même image

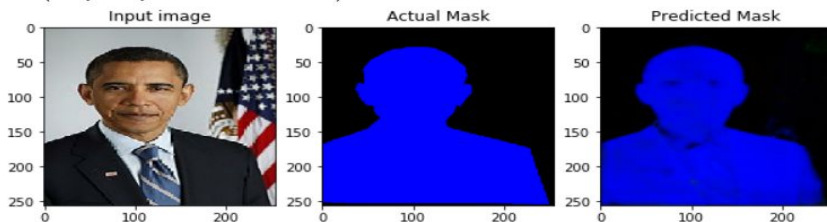


Réalisation

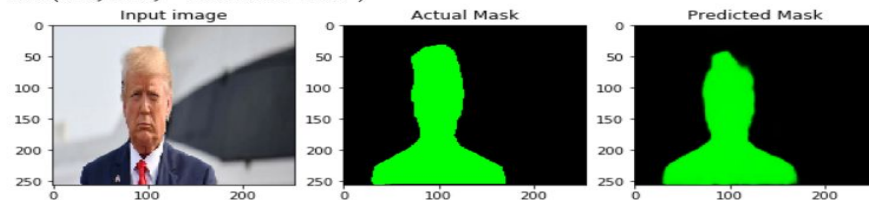
- Pour la creation du modele on a utiliser Keras functional API qui est un hight level API qui offre une abstraction sur les détails des poids et des biais
- On a utilisé le dice_coefficient comme métrique, c'est une mesure qui représente l'intersection entre deux masks
- Pour la fonction d'erreur j'ai utilisé MSE
- Learning rate: 0.005, batch size: 3
- Nombre d'epoch: 200

Réalisation

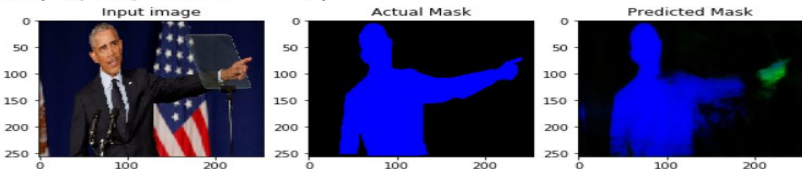
Text(0.5, 1.0, 'Predicted Mask')



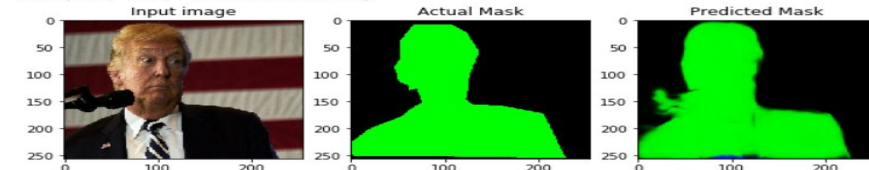
Text(0.5, 1.0, 'Predicted Mask')



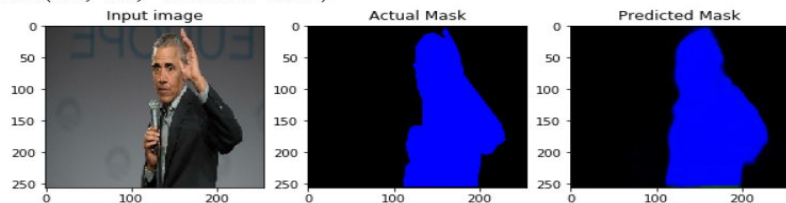
Text(0.5, 1.0, 'Predicted Mask')



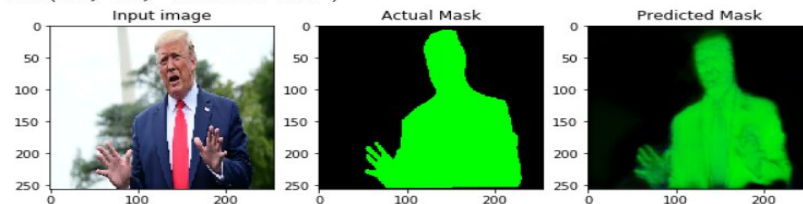
Text(0.5, 1.0, 'Predicted Mask')



Text(0.5, 1.0, 'Predicted Mask')

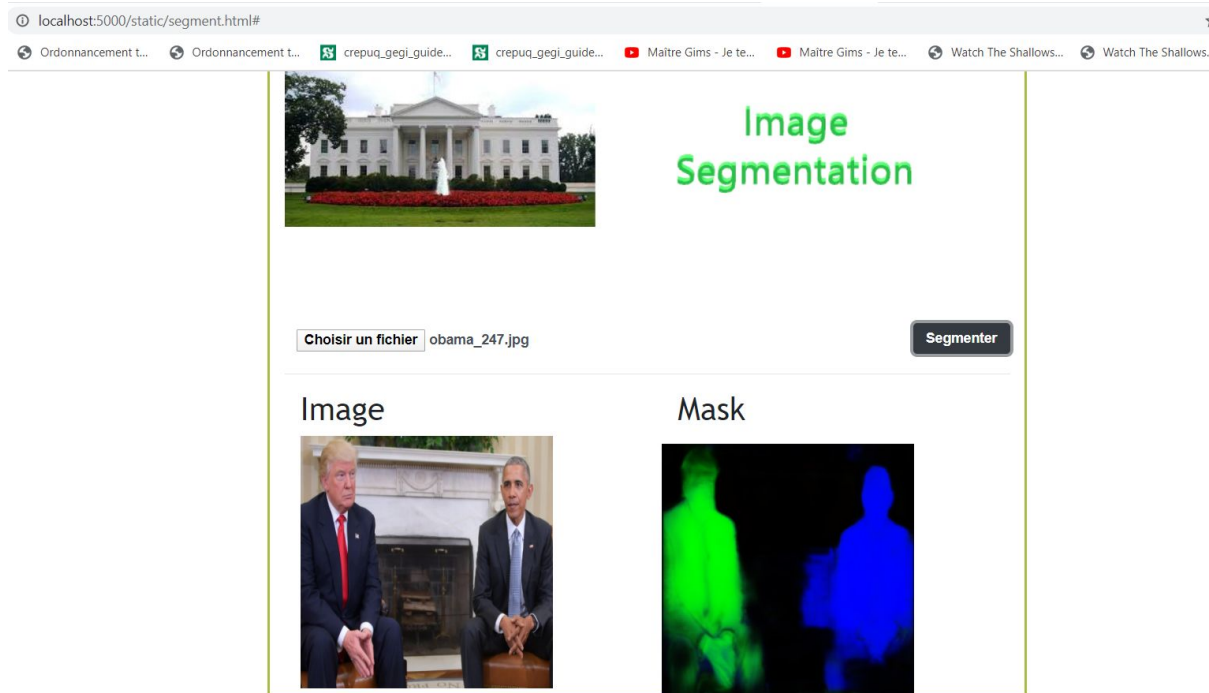


Text(0.5, 1.0, 'Predicted Mask')



Réalisation

- Création d'une application web qui permet de charger une image et de la segmenter dans le navigateur en déployant le modèle grâce à Flask



Conclusion

- Le modèle créée arrive à segmenter les deux présidents avec un taux de reconnaissance de 83% en lui entraînant sur 200 epoches
- Le modèle a une taille de 356,58 MB ce qui n'est pas pratique de l'envoyer à chaque vers le client, en plus il prend un temps considérable pour s'exécuter sur le navigateur, ce qui suggère la possibilité d'utiliser un serveur web pour exécuter le modèle sur une machine distante