



NGUYEN Thi-Christine - RICHARD Pierre
28/11/2024

Sommaire

1	Mesures	2
1.1	Résultats sur la base de donnée de célébrités	2
1.2	Résultats sur la base de donnée de la promo	4
2	Portage mobile	6
3	Conclusion	6

1 Mesures

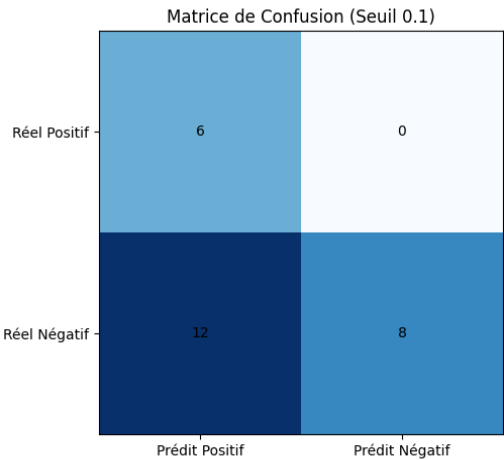
Afin de mesurer l'efficacité de notre application, nous avons réalisé plusieurs mesures. Pour cela, nous nous sommes basés sur un ratio de 80% de données d'entraînement et 20% de données test. Cependant, nous nous sommes rendu compte que notre base de données initiale ne contenait pas le même nombre d'images par personne, ce qui pouvait introduire un biais dans les résultats. De plus, le nombre d'images pour certaines personnes était insuffisant pour constituer un jeu de données test représentatif.

Pour pallier ce problème, nous avons reconstruit une nouvelle base de données composée de 9 célébrités, chaque personne ayant exactement 5 images. Nous avons tout de même réalisé les mesures pour notre base de donnée de la promo.

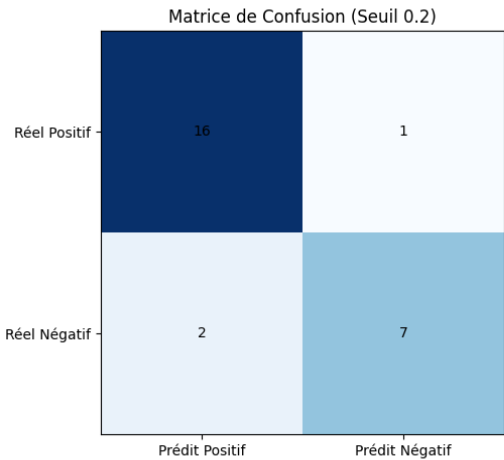
1.1 Résultats sur la base de donnée de célébrités

Nous avons tout d'abord travaillé sur la matrice de confusion qui permet d'analyser les erreurs de classification du modèle.

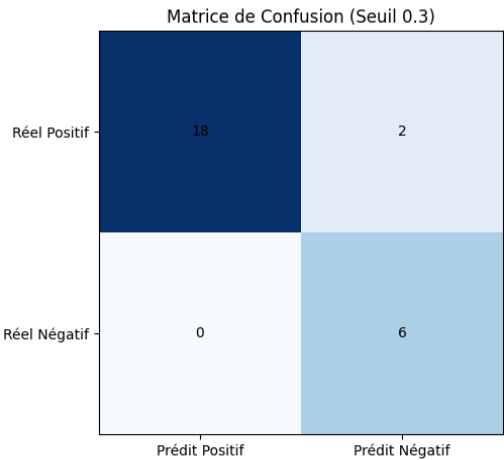
Concernant la matrice de confusion, nous remarquons qu'augmenter le seuil semble réduire les faux négatifs (FN) tout en augmentant les faux positifs (FP). Par exemple, à un seuil de 0.1, le modèle fait beaucoup d'erreurs de faux négatifs (12), tandis qu'à un seuil de 0.5, il y a beaucoup de faux positifs (8) mais aucun faux négatif. Autrement dit, plus le seuil est élevé plus il est permissif. Voici les matrices de confusion pour chaque seuil.



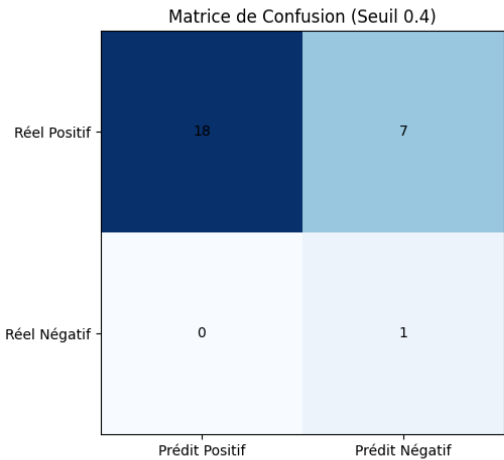
Précision: 1.00
Rappel: 0.33
F1-Score: 0.50



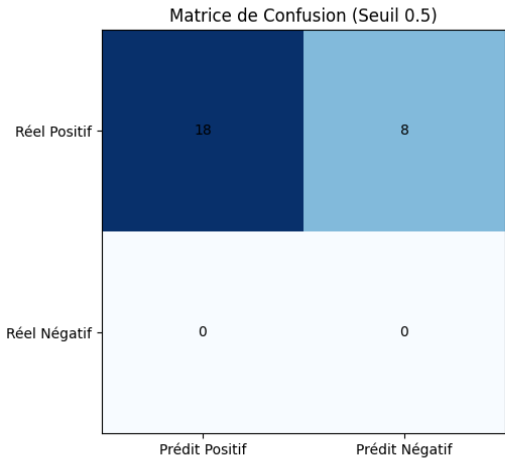
Précision: 0.94
Rappel: 0.89
F1-Score: 0.91



Précision: 0.90
Rappel: 1.00
F1-Score: 0.95



Précision: 0.72
Rappel: 1.00
F1-Score: 0.84



Précision: 0.69
Rappel: 1.00
F1-Score: 0.82

Le F1-score est calculé pour chaque seuil testé et il nous permet d’observer l’impact des seuils sur les performances du modèle. D’après le F1-score de chaque seuil, il semble atteindre son maximum pour le seuil de 0.3. Cela signifie que le modèle atteint un bon compromis entre précision et rappel à ce seuil. On notera tout de même que pour le seuil de 0.2, le F1-score est supérieur à 0.90 ce qui est relativement intéressant.

Nous avons par la suite dessiné la courbe ROC (Receiver Operating Characteristic). Elle permet de visualiser les performances de classification pour tous les seuils. L’AUC (Area Under the Curve) est une métrique qui mesure l’aire sous la courbe ROC. Une valeur d’AUC proche de 1 indique un modèle performant.

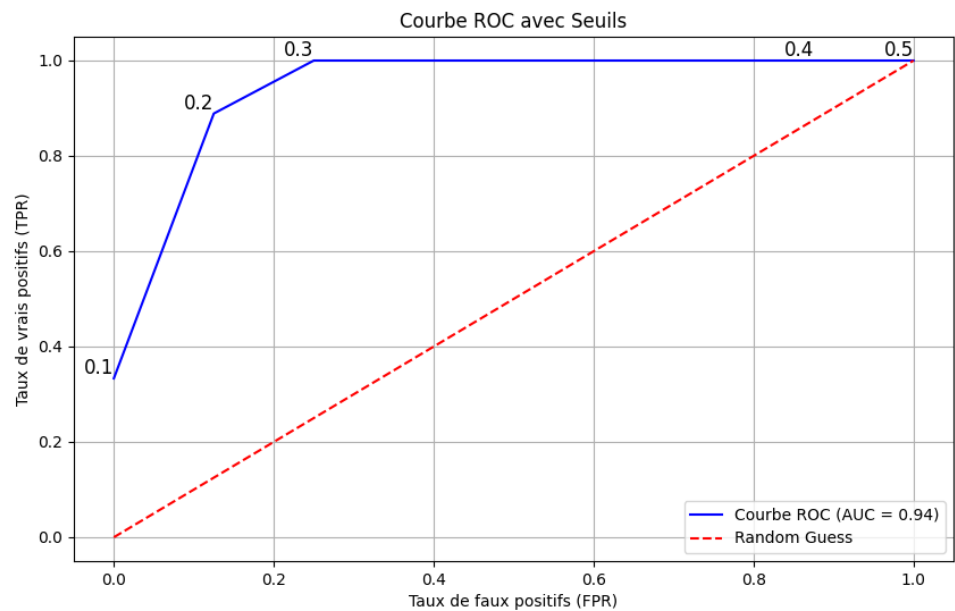


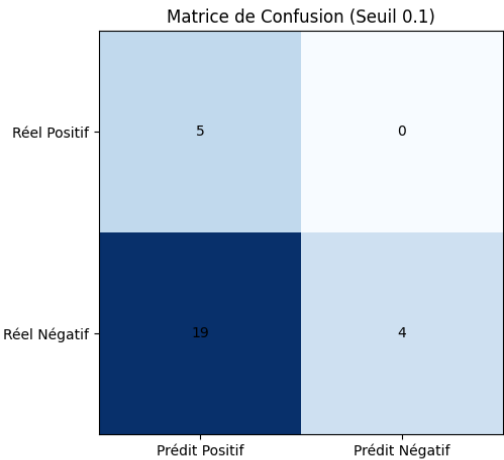
Figure 2: Courbe ROC

Comme vous pouvez le voir, sur la figure ci-dessus, l’AUC est de 0.94 ce qui montre que notre modèle est performant.

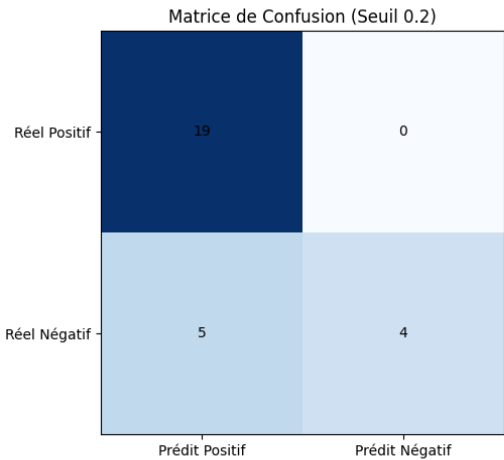
1.2 Résultats sur la base de donnée de la promo

Nous avons effectué les mêmes mesures pour notre base de donnée de la promo malgré son irrégularité et son manque de données. Nous avons fait de notre mieux pour respecter le ratio de 80% d’entraînement et 20% de test. Cette base de donnée est malgré tout celle que nous allons utiliser en temps réel.

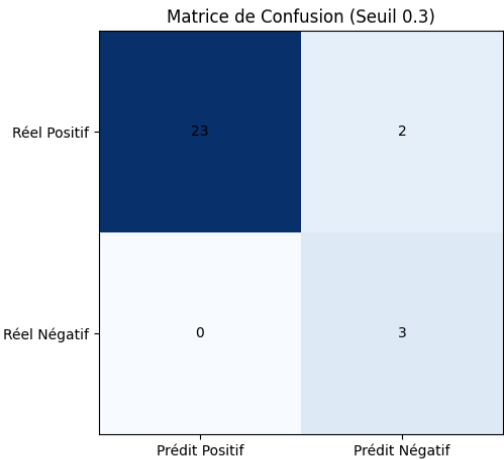
Voici nos matrices de confusion



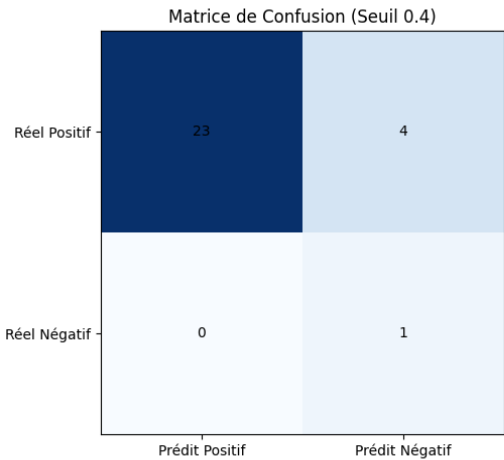
Précision: 1.00
Rappel: 0.21
F1-Score: 0.34



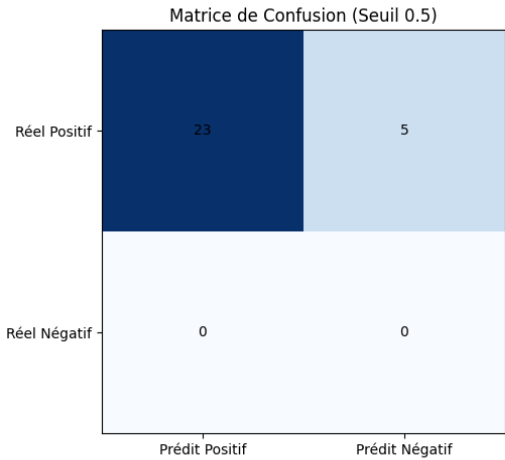
Précision: 1.00
Rappel: 0.79
F1-Score: 0.88



Précision: 0.92
Rappel: 1.00
F1-Score: 0.96



Précision: 0.85
Rappel: 1.00
F1-Score: 0.92



Précision: 0.82
Rappel: 1.00
F1-Score: 0.90

On remarque que le seuil intéressant reste le 0.3 ce qui est en concordance avec nos résultats de la base de donnée de célébrités.

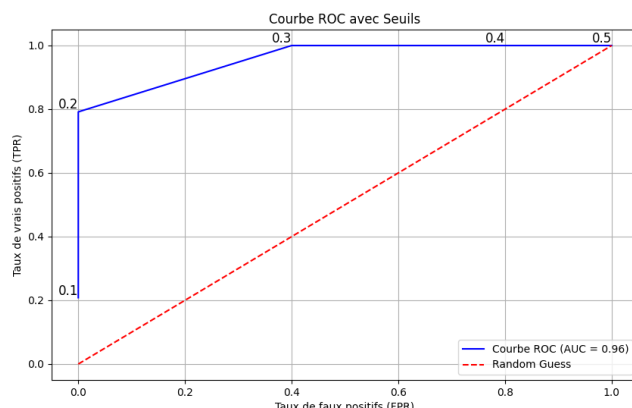


Figure 4: Courbe ROC

Comme vous pouvez le voir, sur la figure ci-dessus, l'AUC est de 0.96 ce qui montre que notre modèle est performant.

2 Portage mobile

Pendant cette semaine, beaucoup d'efforts sur le portage mobile ont été fournis. L'outil buildozer était primordial pour cette partie, mais il prenait également beaucoup de temps à compiler. Buildozer a la particularité de ne prendre en compte que les librairies purement python et quelques rares librairies pour lesquelles des précompilations existent. Également certains type de fichier ne peuvent pas être pris en compte. Pour les fichiers qui ne pouvaient pas être pris en compte, des alternatives ont été trouvées (.pkl à .json, .mat à .npy). Pour les librairies python cela a été plus compliqué, nous avons dû abandonner dlib et utiliser haarcascade. Mais pour le machine learning nous avons besoin de la librairie Keras et bien que buildozer annonce que cette librairie étaient prise en compte ce n'était pas le cas. Nous aurions pu trouver des alternatives à de nombreuses librairies, mais Keras est trop importante pour pouvoir passer outre, ainsi le portage mobile a du être abandonné.

3 Conclusion

D'après les nombreuses mesures que nous avons réalisées. Nous pouvons dire que notre modèle est assez robuste. Il semblerait que le choix d'un seuil de 0.3 soit le plus intéressant. Cependant, selon les diverses utilisations possibles de notre modèle, il sera peut-être préférable d'utiliser un seuil plus restrictif pour éviter d'avoir des faux positifs et ainsi augmenter la sécurité. Dans notre cas, vu que l'application n'a pas de réels besoins sécuritaires, nous avons choisi d'utiliser principalement le seuil de 0.3.

Pour le portage mobile nous pouvons penser à plusieurs solutions, essayer de tout refaire en java, ou mettre en place un serveur qui ferait l'intermédiaire pour la détection et la reconnaissance de visage. Cependant, cela prendrait trop de temps au vu du temps qu'il nous reste jusqu'à la fin du projet.