**Etude des techniques de la reconnaissance faciale :**

**Etude comparatives des différentes méthodes d’extraction de visages à partir des images :**

On va tester différents modèles et méthodes sur différentes images, et on va déduire la méthode la plus performante en termes de précision et de temps d’exécution.

Les méthodes qu’on a sont : 'opencv', 'ssd', 'mtcnn', 'retinaface', 'yolov8', 'yunet', 'fastmtcnn'.

**1ère expérience :**

A l’entrée on va appliquer cette image :

****

Voici dans ce tableau les résultats obtenus :

|  |  |
| --- | --- |
| La méthode | Résultat |
| **opencv** |  |
| **Temps de détection** 1.390409231185913 **seconds** |
| **ssd** |  |
| **Temps de détection** 0.13123464584350586 **seconds** |
| **mtcnn** |  |
| **Temps de détection** 2.428948163986206 **seconds** |
| **retinaface** |  |
| **Temps de détection** 8.324228763580322 **seconds** |
| **Yolov8** |  |
| **Temps de détection** 0.33027172088623047 **seconds** |
| **yunet** |  |
| **Temps de détection** 0.06933712959289551 **seconds** |
| **fastmtcnn** |  |
| **Temps de détection** 1.2057855129241943 **seconds** |

**2eme expérience :**

A l’entrée on va appliquer cette image : 

Voici dans ce tableau les résultats obtenus :

|  |  |
| --- | --- |
| La méthode | Résultat |
| **opencv** |  |
| **Temps de détection** 2.8044612407684326 **seconds** |
| **ssd** |  |
| **Temps de détection** 0.5496735572814941 **seconds** |
| **mtcnn** |  |
| **Temps de détection** 3.4700331687927246 **seconds** |
| **retinaface** |  |
| **Temps de détection** 8.08993124961853 **seconds** |
| **Yolov8** |  |
| **Temps de détection** 0.32061767578125 **seconds** |
| **yunet** |  |
| **Temps de détection** 0.15271806716918945 **seconds** |
| **fastmtcnn** |  |
| **Temps de détection** 2.0015838146209717 **seconds** |

**3eme expérience :**

A l’entrée on va appliquer cette image :



Voici dans ce tableau les résultats obtenus :

|  |  |
| --- | --- |
| La méthode | Résultat |
| **opencv** |  |
| **Temps de détection** 2.8164596557617188 **seconds** |
| **ssd** |  |
| **Temps de détection** 0.8297803401947021 **seconds** |
| **mtcnn** |  |
| **Temps de détection** 3.4041895866394043 **seconds** |
| **retinaface** |  |
| **Temps de détection** 10.146069049835205 **seconds** |
| **Yolov8** |  |
| **Temps de détection** 0.2710270881652832 **seconds** |
| **yunet** |  |
| **Temps de détection** 0.10614466667175293 **seconds** |
| **fastmtcnn** |  |
| **Temps de détection** 1.429173231124878 **seconds** |

**4eme expérience :**



Voici dans ce tableau les résultats obtenus :

|  |  |
| --- | --- |
| La méthode | Résultat |
| **opencv** |  |
| **Temps de détection** 0.48532843589782715 **seconds** |
| **ssd** |  |
| **Temps de détection** 0.14068317413330078 **seconds** |
| **mtcnn** |  |
| **Temps de détection** 1.3943920135498047 **seconds** |
| **retinaface** |  |
| **Temps de détection** 7.003560781478882 **seconds** |
| **Yolov8** |  |
| **Temps de détection** 0.3076813220977783 **seconds** |
| **yunet** |  |
| **Temps de détection** 0.06598830223083496 **seconds** |
| **fastmtcnn** |  |
| **Temps de détection** 0.5654113292694092 **seconds** |

En moyennant les résultats de temps d’exécution qu’on a obtenus et on aura ce graphe qui représente la vitesse d’exécution de chaque méthode :

**Interprétation :**

On peut voir que la méthode "RetinaFace" est la plus longue en temps d'exécution, mais d'après le tableau, on peut constater qu'elle donne de bons résultats. La méthode "SSD" est rapide mais produit de mauvais résultats et ne détecte pas tous les visages dans une image. Nous allons choisir la méthode "Fast MTCNN" car elle est rapide en temps d'exécution et présente de bonnes performances, ayant réussi à détecter tous les visages dans les différents cas expérimentaux.

Après cette étude comparative des différentes méthodes de détection et d'extraction des visages, nous allons maintenant tester différentes méthodes de reconnaissance faciale et évaluer leurs performances, puis réaliser une étude comparative.

Étude comparative des différents modèles de reconnaissance faciale :

Nous allons tester différents modèles et méthodes sur différentes images afin de déterminer la méthode la plus performante en termes de précision et de temps d'exécution le plus court. Les modèles à tester incluent : "VGG-Face", "Facenet", "Facenet512", "OpenFace", "DeepFace", "DeepID", "ArcFace", et "SFace".

Ces modèles sont utilisés pour l'extraction des caractéristiques du visage dans une image sous forme de vecteur.

La méthode consiste à utiliser deux images et à calculer la distance entre les deux vecteurs donnés par chaque modèle. Ensuite, nous pouvons comparer cette distance avec un certain seuil pour déterminer la similarité des deux visages dans les deux images. Dans ces expériences, nous prendrons en considération le temps d'exécution pour chaque méthode sur Google Colab.

**Top of Form**

**1er expérience :**

 

Nous allons utiliser deux photos de deux personnes différentes et calculer la distance entre les vecteurs caractéristiques de chaque modèle.

Si la prédiction est « False », cela indique que nous n'avons pas la même personne. Si la prédiction est « True », cela indique que nous avons la même personne. La valeur du seuil est une valeur définie par défaut et choisie après avoir testé le modèle sur différentes images.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Le modèle** | **Distance** | **Temps d’exécution** | **La prédiction** | **Seuil** |
| VGG-Face | 0.9159331776545481 | 4.150911808013916 | False | 0.68 |
| Facenet | 0.8080704563914897 | 2.833468437194824 | False | 0.4 |
| Facenet512 | 0.8079165673322692 | 3.267817974090576 | False | 0.3 |
| OpenFace | 0.3614301905417918 | 2.417363166809082 | False | 0.1 |
| DeepFace | 0.18514633185158869 | 9.18279218673706 | False | 0.23 |
| DeepID | 0.033132321975335 | 2.3758513927459717 | False | 0.015 |
| ArcFace | 0.9252884665232564 | 2.74701142311096 | False | 0.68 |
| SFace | 0.9739253684255617 | 4.529379844665527 | False | 0.593 |

**Commentaire :** Les différents modèles ont prédit le bon résultat.

**2eme expérience :**



Dans cette deuxième expérience, nous allons tester les modèles sur deux images de la même personne mais avec un éclairage et des expressions faciales différents. Voici les résultats :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Le modèle** | **Distance** | **Temps d’exécution** | **La prédiction** | **Seuil** |
| VGG-Face | 0.6736759059017616 | 2.937410831451416 | True | 0.68 |
| Facenet | 0.6236478723951333 | 2.9040067195892334 | False | 0.4 |
| Facenet512 | 0.4236998679687729 | 2.8820929527282715 | False | 0.3 |
| OpenFace | 0.3712229096442299 | 2.053914785385132 | False | 0.1 |
| DeepFace | 0.23026103484438232 | 5.59130859375 | False | 0.23 |
| DeepID | 0.12204879577616157 | 2.554736375808716 | False | 0.015 |
| ArcFace | 0.37433675889583307 | 2.63346791267395 | True | 0.68 |
| SFace | 0.24810227303972132 | 1.9013197422027588 | True | 0.593 |

**Commentaire :** On peut clairement voir que seuls les modèles VGG-Face, ArcFace et SFace ont donné le bon résultat, tandis que les autres ont échoué.

**3eme expérience :**

 

Maintenant, dans cette partie, nous avons la même personne avec une photo simple et l'autre avec une casquette et des lunettes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Le modele** | **Distance** | **Temps d’execution** | **la prediction** | **Seuil** |
| VGG-Face | 0.6133414794757963 | 3.248588800430298 | True | 0.68 |
| Facenet | 0.38187203233407196 | 1.4804351329803467 | True | 0.4 |
| Facenet512 | 0.37281731632954174 | 1.4907350540161133 | False | 0.3 |
| OpenFace | 0.30112648391917396 | 0.9992396831512451 | False | 0.1 |
| DeepFace | 0.38139448891818506 | 5.371493816375732 | False | 0.23 |
| DeepID | 0.1470279082756102 | 0.7740786075592041 | False | 0.015 |
| ArcFace | 0.6186461820371727 | 1.947218656539917 | True | 0.68 |
| SFace | 0.7803979012523027 | 1.0754146575927734 | False | 0.593 |

**Commentaire :** On peut voir maintenant que les modèles VGG-Face, Facenet et ArcFace ont donné les bonnes prédictions.

**4ème expérience :**

   
Dans cette expérience, nous avons la même personne avec et sans barbe :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Le modèle** | **Distance** | **Temps d’exécution** | **La prédiction** | **Seuil** |
| VGG-Face | 0.6192805123478301 | 2.4109108448028564 | True | 0.68 |
| Facenet | 0.33399397091011995 | 1.6701862812042236 | True | 0.4 |
| Facenet512 | 0.32227831143130603 | 1.640101671218872 | False | 0.3 |
| OpenFace | 0.26733561871553846 | 1.201770305633545 | False | 0.1 |
| DeepFace | 0.21537111581274848 | 5.225113153457642 | True | 0.23 |
| DeepID | 0.047543969387073504 | 1.4541990756988525 | False | 0.015 |
| ArcFace | 0.4843091708996228 | 1.629307746887207 | True | 0.68 |
| SFace | 0.5325555920972895 | 1.0362541675567627 | True | 0.593 |

**Commentaire :** On peut voir que les méthodes VGG-Face, Facenet, DeepFace, ArcFace et SFace ont donné de bons résultats.

**Conclusion :**

Après ces expériences, on peut conclure que les méthodes VGG-Face et ArcFace sont les meilleures, et même la méthode Facenet est bonne. De plus, si l'on prend en considération le temps d'exécution, on peut constater que la méthode ArcFace est plus rapide. Dans notre solution, nous choisirons la méthode Facenet en raison de son efficacité, de sa puissance pour la reconnaissance faciale et même de son adaptation à tous les types de visages.

Top of Form

Dans la partie suivante on va tester le modèle en utilisant deux images d’une même personne comme des images de référence.

**1er expérience :**

On a pris ces deux photos comme des photos de référence :



Et on va tester le modèle sur cette image :



Voici les résultats :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Le modèle** | **Distance** | **Temps d’exécution** | **la prédiction** | **Seuil** |
| VGG-Face | 0.44920762092441663 | 7.741603374481201 | True | 0.68 |
| Facenet | 0.29065156678962356 | 5.476769685745239 | True | 0.4 |
| Facenet512 | 0.229067228395418 | 6.589885711669922 | True | 0.3 |
| OpenFace | 0.18616259303671712 | 4.522589683532715 | False | 0.1 |
| DeepID | 0.11295035396858222 | 4.04485297203064 | False | 0.015 |
| ArcFace | 0.29804117738617 | 6.432822942733765 | True | 0.68 |
| SFace | 0.25465584023378546 | 4.114460468292236 | True | 0.593 |

**2eme expérience :**





Voici les résultats :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Le modèle** | **Distance** | **Temps d’exécution** | **La prédiction** | **Seuil** |
| VGG-Face | 0.8680062967702799 | 7.726759910583496 | False | 0.68 |
| Facenet | 0.8371736378526061 | 5.800991773605347 | False | 0.4 |
| Facenet512 | 0.8371736378526061 | 5.800991773605347 | False | 0.3 |
| OpenFace | 0.2635169168711628 | 4.6398797035217285 | False | 0.1 |
| DeepID | 0.06124834937119389 | 5.070623159408569 | False | 0.015 |
| ArcFace | 0.8395529345218631 | 5.509099245071411 | False | 0.68 |
| SFace | 0.8580545495536092 | 4.730891704559326 | False | 0.593 |

**3 ème expérience :**





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Le modèle** | **Distance** | **Temps d’exécution** | **La prédiction** | **Seuil** |
| VGG-Face | 0.5687141418673098 | 8.868527889251709 | True | 0.68 |
| Facenet | 0.3243648385299075 | 6.263379812240601 | True | 0.4 |
| Facenet512 | 0.3485367013508798 | 7.130490064620972 | False | 0.3 |
| OpenFace | 0.3885802002551872 | 4.791218280792236 | False | 0.1 |
| DeepID | 0.07876269126978791 | 4.270202875137329 | False | 0.015 |
| ArcFace | 0.40230563548017967 | 6.997164249420166 | True | 0.68 |
| SFace | 0.4866726568475417 | 4.195332765579224 | True | 0.593 |

**Commentaire :**

D'après les différentes expériences, les modèles VGG-Face, Facenet, ArcFace et SFace ont donné de bons résultats. On remarque que le modèle SFace a le temps d'exécution le plus court, suivi par le modèle Facenet. Par conséquent, nous choisirons le modèle Facenet pour notre projet.

**Conclusion générale :**

La première partie de l'étude était dédiée à la comparaison des différents modèles de détection et de localisation des visages dans les images. Nous avons sélectionné le modèle Fastmtcnn en raison de son efficacité et de son temps d'exécution.

Dans la deuxième partie, nous avons réalisé une étude comparative entre les différents modèles de reconnaissance faciale, concluant que le modèle Facenet produit de bons résultats. Nous avons également testé le modèle avec une et deux photos de référence, constatant que le système présente de bonnes performances avec deux photos, bien que le temps d'exécution soit plus long par rapport au cas d'une seule photo de référence.

**- La prochaine étape est l’implémentation du programme sur la carte Raspberry Pi 4.**