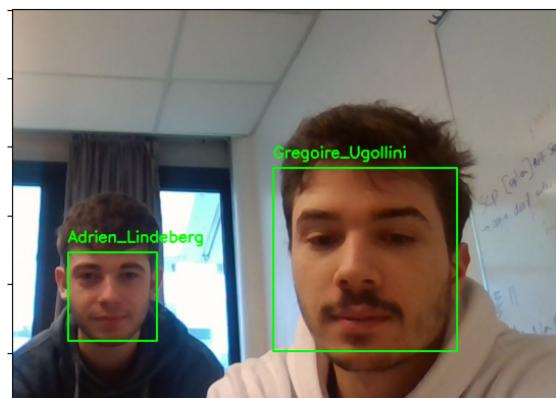


Projet deep learning:

LINDEBERG Adrien
IA et Deep Learning
ESIEE - 2023
France
adrien.lindeberg@edu.esiee.fr

LABOU Rémi
IA et Deep Learning
ESIEE - 2023
France
remi.labou@edu.esiee.fr



Abstract: Pendant ce projet, on a programmé une reconnaissance faciale. On est passé par plusieurs étapes rendant nos modèles de plus en plus performants pour au final atteindre une précision aux alentours des 95%. Pour cela, on a commencé par tester notre modèle sur des données normalisées où l'on a extrait le visages des personnes. Puis dans un second temps, on a utilisé des données où les visages étaient recadrés, alignés et normalisés. Ensuite, on a pris nos visages recadrés et on les a encodés, c'est-à-dire que notre ordinateur à fait des mesures sur chaque visages que l'on pourra comparer avec notre modèle. On a testé plusieurs modèles (réseaux de neurones, SVM, knn...) pour savoir lequel était le plus performant sur nos images encodées et l'utiliser comme modèle final. Pour finir, on a créé notre propre dataset avec les visages de dix personnes différentes et environ soixante photos pour chacun d'eux.

Index Terms: Deep learning, convolutional network, tests, modèles, face, encodage de visage.

Introduction: Une reconnaissance faciale est une technologie qui permet d'identifier et de vérifier l'identité de personnes à partir de leurs traits faciaux. Cette technologie peut être utilisée dans de nombreux secteurs, tels que la sécurité, le contrôle d'accès, la surveillance, la reconnaissance de personnes recherchées... Elle répond surtout à une problématique de sécurité.

Elle peut permettre d'automatiser celle-ci en permettant de vérifier l'identité de quelqu'un instantanément. Ainsi, à travers notre projet, on va programmer une reconnaissance faciale capable de reconnaître 10 personnes différentes. Ainsi, quand on demandera à notre reconnaissance faciale de nous donner la personne sur une photo entre dix personnes, elle sera capable de nous prédire la bonne personne avec un pourcentage de réussite de plus de 95%.

related works:

- "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection" de Belhumeur, Hespanha et Kriegman(1997): Cet article présente une méthode de reconnaissance des visages qui n'utilise pas le deep learning. Pour cette reconnaissance faciale, deux techniques de projection linéaire ont été utilisées, les Eigenfaces et les Fisherfaces, pour réduire la dimension des images de visage et extraire les caractéristiques importantes.

Les résultats ont montré que la technique Fisherfaces est plus efficace que la technique Eigenfaces en termes de précision de reconnaissance. Le principal défaut de cette méthode est qu'elle ne fonctionne pas bien avec des images de visage qui ont des variations de pose et d'éclairage importantes.

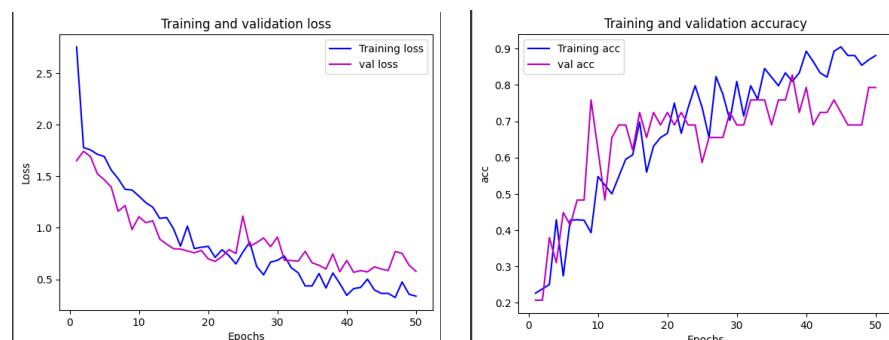
- "DeepFace: Closing the Gap to Human-Level Performance in Face Verification" de Taigman et al. (2014): Cet article présente une méthode de reconnaissance des visages qui utilise le deep learning. Dans ce projet un réseau de neurones convolutionnels est utilisé pour extraire les caractéristiques des images de visage. Ils ont utilisé une grande base de données d'images de visage, comprenant plus de 4 millions d'images, pour entraîner leur modèle. Les résultats ont montré que leur modèle a atteint une précision de reconnaissance supérieure à celle des êtres humains. Le principal défaut de cette méthode est qu'elle nécessite une grande quantité de données et de puissance de calcul.
- "ArcFace: Additive Angular Margin Loss for Deep Face Recognition" de Deng et al. (2019): Cet article présente une méthode de reconnaissance des visages qui utilise le deep learning. Deng et al ont utilisé un réseau de neurones convolutionnels pour extraire les caractéristiques des images de visage, qui ont ensuite été utilisées pour classer les images. Ils ont proposé une nouvelle fonction de perte, appelée la perte de marge angulaire additive, qui permet d'améliorer la séparation entre les classes et la généralisation du modèle. Les résultats ont montré que leur modèle est très performant, dépassant de nombreux autres modèles de reconnaissance des visages. Le principal défaut de cette méthode est qu'elle est relativement complexe et nécessite une optimisation minutieuse des paramètres pour atteindre des performances optimales.

PROPOSITION:

A. Face detection:

La détection des visages est une étape clé dans la reconnaissance faciale qui consiste à localiser et extraire les visages dans une image ou une vidéo. L'article: "Understanding Face Detection with the Viola-Jones Object Detection Framework" explique bien le fonctionnement de la détection de visages en prenant comme exemple la méthode de Viola-Jones.

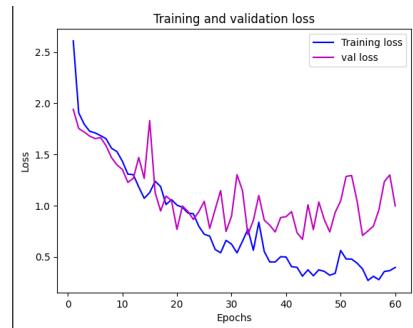
Pour commencer notre reconnaissance faciale, on commence avec un dataset contenant 218 photos de 6 personnes différentes: Ian Malcolm, Claire Dearing, Ellie Sattler, John Hammond, Owen Grady et Alan Grant. On possède des photos d'eux de taille, luminosité, et d'angles différents. On a un nombre différent de photos pour chacun d'eux comme avec Claire Dearing qui a plus de photos d'elle que Ian Malcolm. Puis, on a eu besoin d'extraire le visage de chaque photo car cela permet de détecter l'emplacement du visage sur la photo pour cadrer et redimensionner la photo sur le visage afin que toutes les photos aient les mêmes dimensions et puissent être traitées par un modèle plus tard. Avant le traitement des images, les techniques de prétraitement que l'on a utilisées sont la normalisation des images pour les rendre plus uniformes et faciles à traiter. et un codage one-hot des labels. Ensuite, on a divisé notre dataset d'images en deux packet. un paquet d'entraînement qui prend 70% de images et un paquet de test qui prend les images restantes. Pour notre modèle, on a utilisé un réseau de convolution de quatre couches et prenant comme entrées des images de dimensions 128x128x3. Puis, on utilise un réseau de neurones de quatre couches: la première servant à transformer le résultat du réseau de convolution en un tableau 1D, puis un dropout pour éviter l'overfitting et une couche avec une activation relu et pour finir une couche avec une activation softmax qui nous retourne la probabilité de la personne qui est sur l'image. Notre modèle a de bons résultats car il a très peu d'overfitting et a une bonne précision même s'il a pas mal de pertes de données .



Lorsqu'on lui demande de prédire le label d'une image avec notre paquet test, on a une précision aux alentours de 73% qu'il nous retourne la bonne personne.

résultat du modèle après face detection:

```
2/2 [=====] - 0s 17ms/step - loss: 0.7746 - acc: 0.7460
Test accuracy: 74.60317611694336 %
```



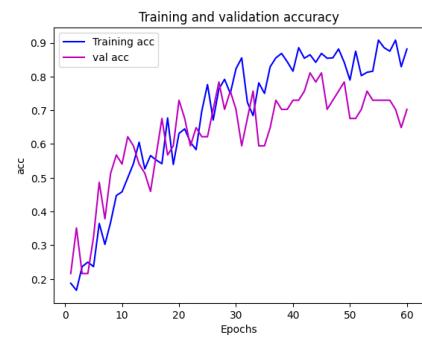
Lorsque l'on fait varier le nombre de couches, le nombre d'unités cachées, le taux du dropout ou encore le taux d'apprentissage, on obtient souvent un modèle qui fait de l'overfitting ou à une précision plus faible et plus de pertes de données.

B. Pose estimation:

L'estimation de la pose est une technique de vision par ordinateur qui consiste à détecter et à localiser la position et l'orientation d'un visage ou d'une personne dans une image puis de la déformer de manière à ce que les yeux et les lèvres se trouvent toujours au même endroit dans l'image comme le décrit l'article de Wei et al. (2016), "3D Human Pose Estimation from a Single Image via Distance Matrix Regression" avec l'exemple de la régression de la matrice de distance.

Après avoir effectué l'estimation de la pose sur nos images, on a maintenant un dataset de visages recadrés, alignés et normalisés. Les images ont les mêmes dimensions que pendant la face detection soit: 128x128x3. On a besoin de faire la pose estimation car après avoir extrait le visage sur chaque photo, l'emplacement des différentes parties de celui-ci restent différentes pour chaque photo ce qui rend l'apprentissage du modèle compliqué. Ainsi, on utilise la pose estimation pour détecter les différentes parties du visage et les replacer afin que pour chaque photo, elles se trouvent au même endroit. Pour faire notre pose estimation, on utilise un algorithme appelé "estimation des points de repère du visage". Cet algorithme localise 68 points sur chaque visage, toujours les mêmes. Ensuite, on modifie l'image pour que les yeux et la bouche soient centrés le mieux possible.

Pour cette partie, on a divisé de la même manière notre dataset que lors de la première partie. On utilise le même modèle que l'on a utilisé sur nos images avec visages extraits et on remarque que la précision s'améliorent un peu, on tourne autour des 80% lors de l'évaluation du modèle au lieu de 73%. Les résultats sont bons car il y a moins de perte de données et une meilleure précision lors de l'évaluation du modèle.



résultat modèle après pose estimation:

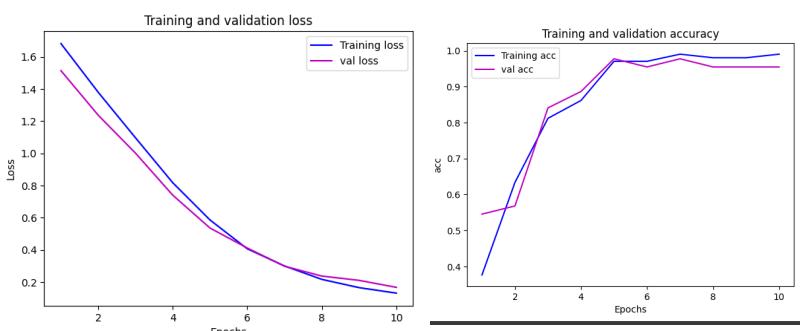
```
2/2 [=====] - 0s 13ms/step - loss: 0.5789 - acc: 0.8254
Test accuracy: 82.53968358039856 %
```

Il est logique que les résultats s'améliorent car le modèle s'entraîne sur des images où les parties importantes pour reconnaître le visage se trouvent sensiblement au même endroit pour chaque personne contrairement à la partie précédente.

C. Face encoding:

L'encodage de visage est le processus de conversion d'une image de visage en une représentation numérique. Cette représentation numérique est généralement une série de calculs fait par notre ordinateur qui représentent les caractéristiques faciales importantes d'une personne, telles que la distance entre les yeux, la largeur du nez, la forme de la mâchoire... Un article de référence sur l'encodage des visages est "FaceNet: A Unified Embedding for Face Recognition and Clustering" de Schroff et al., publié en 2015. Cet article présente la méthode FaceNet pour l'encodage de visages.

Le principal avantage de l'encodage de visage est que pour décider quelles mesures il va faire sur chaque visage il examine trois images de visage à la fois : deux images de la même personne et l'image d'une personne totalement différente. Ensuite, l'algorithme examine les mesures générées pour chacune de ces trois images et modifie le réseau neuronal pour s'assurer que les mesures générées pour une même personne sont plus proches des mesures pour des personnes différentes. Ainsi, il sera plus facile pour notre modèle d'identifier les données correspondant à la même personne. On donne à l'algorithme les images des visages extraits et alignées et elles seront transformées en un tableau de 128 mesures. Pour cette partie, on n'a pas utilisé un réseau convolutionnel mais plutôt un réseau de neurones. On obtient d'excellents résultats: 98.5% de précision après l'évaluation du modèle. De plus, on n'a pas d'overfitting et une nette baisse des pertes de données (moins de 0.15).



résultat modèle après face encoding:

```
2/2 [=====] - 0s 12ms/step - loss: 0.1888 - acc: 0.9841
Test accuracy: 98.41269850730896 %
```

Il est logique que nos résultats s'améliorent ainsi car il est plus simple pour notre modèle de reconnaître des tableaux de mesures que des images et l'algorithme fait des mesures ou pour chaque personnes les résultats sont éloignés.

D. Face recognition:

La reconnaissance faciale est le processus qui permet de trouver la personne dans notre base de données de personnes qui a les mesures les plus proches d'une image test. Un article qui explique bien comment fonctionne la reconnaissance faciale est "DeepFace: Closing the Gap to Human-Level Performance in Face Verification" de Taigman et al., publié en 2014.

La différence entre la reconnaissance des visages et la détection des visages est que la détection des visages consiste simplement à localiser la position et la taille d'un visage dans une image, tandis que la reconnaissance des visages implique l'identification ou la vérification de l'identité d'une personne.

Nous avons décidé pour cette tâche d'utiliser quatre classificateurs différents: régression logistique, knn, SVM et réseaux de neurones. J'ai décidé de prendre ces quatre classificateurs car ils sont reconnus pour avoir de très bon résultats pour la reconnaissance faciale. Chacun a ses avantages et limites ce qui rend intéressant de comparer leurs résultats. Chaque classificateur donne de très bons résultats pour notre dataset mais là où la différence se fait réellement est sur la vitesse d'exécution.

résultat des classifieurs:

```
Logistic Regression: Validation Accuracy: 98.41% -> Time: 0.0230s
SVM: Validation Accuracy: 96.83% -> Time: 0.0038s
kNN: Validation Accuracy: 100.00% -> Time: 0.0053s
Neural Network: Validation Accuracy: 98.41% -> Time: 0.5155s
```

Le knn est bien plus rapide avec une vitesse d'exécution qui varie autour 0.006s que le réseau de neurones et la régression logistique et a une précision supérieure à tous les autres classificateurs. Ainsi, même s'il est plus lent que le SVM, le knn est plus intéressant car il allie une meilleure précision que le svm et à tout de même une très bonne vitesse d'exécution ce qui pour moi en fait le meilleur classifieur des quatre.

E. Personal dataset:

Notre dataset contient 60 à 70 photos de 12 personnes différentes soit 753 photos différentes. On remarque qu'avec notre dataset marche vraiment mieux. Il y a une bien meilleure précision (on passe à une précision de plus de 92.5% pour l'étape d'extraction de visages), plus aucun overfitting et beaucoup moins de pertes de données (moins de 0.2 pour l'étape d'extraction de visages) pour les modèles de face detection et pose estimation. Mais, il y a plus vraiment d'écart entre les étapes d'extraction de visages et d'estimation de la pose ce qui est logique car on a déjà une très forte précision après la première étape. Il y a toujours une baisse de la perte des données entre les deux étapes (passe de 0.2 à 0.15). Cependant, pour les modèles de face recognition, si les résultats sont un peu supérieurs, c'est surtout la vitesse d'exécution qui augmente ce qui est normal car notre dataset est largement plus grand que le dataset précédent. Ainsi, ce n'est plus le classifieur knn le plus intéressant mais le SVM qui a des résultats similairement identiques aux autres classifieurs mais est bien plus rapide, dix fois plus rapide que le knn qui est le deuxième plus rapide.

Résultat des classifieurs avec notre dataset:

```
Logistic Regression: Validation Accuracy: 98.23% -> Time: 0.0473s
SVM: Validation Accuracy: 98.23% -> Time: 0.0158s
kNN: Validation Accuracy: 98.23% -> Time: 0.1086s
Neural Network: Validation Accuracy: 98.23% -> Time: 1.0702s
```

Ceci est dû au fonctionnement de l'algorithme, plus il y aura de données plus il sera facile de séparer les différentes classes grâce aux ressemblances apprises par le modèle.

F. Extra - Bias analysis:

Un biais dans l'apprentissage automatique est un déséquilibre dans les données d'apprentissage qui peuvent conduire à des prédictions incorrectes ou injustes. Les biais est donc un problème dans l'apprentissage automatique car ils peuvent entraîner des décisions injustes ou discriminatoires. La reconnaissance faciale est un bon exemple de situation où les biais peuvent être un problème. Une étude a montré que les systèmes de reconnaissance faciale ont une précision inférieure pour les femmes et les personnes de couleur, en partie en raison de la sous-représentation de ces groupes dans les ensembles de données d'apprentissage. Dans mon dataset il n'y a effectivement pas beaucoup de femmes ou de gens de couleurs de peau donc je ne peux pas dire si mon modèle a de moins bon résultats pour une certaine ethnique.

CONCLUSION:

Finalement, avec la dataset initial, on a pu voir après chaque tâche que notre modèle s'améliorait. On a après chaque étape une meilleure précision avec moins d'overfitting et de pertes de données. On obtient au final une reconnaissance faciale avec une précision de plus de 97%, aucun overfitting et peu de pertes de données.

En passant à notre dataset on a une nette amélioration de la précision et une baisse des pertes de données pour l'extraction de visages et l'estimation de la pose même si les résultats deviennent ressemblant ensuite. Le classifieur le plus efficace change aussi entre les deux dataset passant du knn pour le premier au SVM pour notre dataset.

On pourrait rendre notre reconnaissance faciale encore plus complète en augmentant le nombre de personnes dans le dataset et ajoutant plus de personnes de différentes ethnies.

RÉFÉRENCES:

- <https://ieeexplore.ieee.org/document/598228>
- <https://ieeexplore.ieee.org/document/6909616>
- <https://ieeexplore.ieee.org/document/8953658>
- <https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14>
- <https://ieeexplore.ieee.org/document/8099653>
- <https://ieeexplore.ieee.org/document/7298682>