PROJET IA - Reconnaissance faciale



**Ce rapport aborde notre solution proposée pour le projet final de l’unité IA et deep learning de E3. Ce projet consiste à mettre en place un programme utilisant des réseaux de neurones dans l’optique de reconnaître les visages de différentes personnes.**

Index :

Deep learning : L’utilisation des réseaux de neurones afin de résoudre des tâches complexes.

Réseaux de neurones artificiels : réseau artificiel fonctionnant de manière similaire au cerveau humain. Des neurones communiquent tous entre eux sous forme de layers.

biais algorithmique : est un terme utilisé lorsque les résultats d’un algorithme ne sont pas neutre et sont donc influencés par d’autres facteurs non prévu par les concepteurs (ce qui est très commun dans les domaines de l’intelligence artificielle).

I. INTRODUCTION

Dans la suite de ce document nous allons ensemble découvrir les problèmes auxquels nous avons été confrontés afin de réaliser un modèle détectant des visages de personnes qu’il connaît .

Nous commencerons par aborder le premier problème qui est la détection des visages, dans cette partie nous verrons l’importance d’avoir des données “propres” et de comprendre celle-ci afin d’optimiser l'entraînement.

Nous aborderons ensuite d’autres problèmes tout aussi complexes qui ont un point commun : obtenir un résultat propre et cohérent.

II. RELATED WORKS

Pour ce projet nous avons effectué plusieurs recherches sur le sujet, Ainsi il nous semble important d’aborder rapidement les documents qui nous ont le plus aidés.   
Teoh, K. *et al.* (2021) *IOPscience*, *Journal of Physics: Conference Series*. Available at: https://iopscience.iop.org/article/10.1088/1742-6596/1755/1/012006/meta (Accessed: 22 April 2024).

Cet article aborde la reconnaissance faciale en utilisant des algorithmes de Deep Learning similaire à la façon dont nous avons réalisé notre projet. En effet, les auteurs de l’article utilisent deux outils qui sont Python (ce que nous utilisons pour notre modèle) et la bibliothèque OpenCV. L’objectif des auteurs est la reconnaissance faciale en temps réel grâce à un flux vidéo dans différents environnements. A la fin de l'étude on apprend que le modèle a atteint une précision de 91.7% pour les images et 86.7% pour les vidéos en temps réel.

* Zhi, H. and Liu, S. (2018) *Face recognition based on genetic algorithm*, *Journal of Visual Communication and Image Representation*. Available at: https://www.sciencedirect.com/science/article/abs/pii/S1047320318303389 (Accessed: 22 April 2024).

Dans cet article, La reconnaissance faciale est basée sur un algorithme génétique et les Support Vector Machine (SVM). L’objectif des auteurs est d’atteindre un résultat bien plus élevé que les méthodes d'entraînements classiques de reconnaissance faciale comme le deep learning. Au final, cet objectif est atteint en atteignant une précision supérieure à 99% prouvant l’efficacité de cette méthode.

* Perkowitz, S. (2021) *The bias in the machine: Facial recognition technology and racial disparities*, *MIT Case Studies in Social and Ethical Responsibilities of Computing*. Available at: https://mit-serc.pubpub.org/pub/bias-in-machine/release/1?readingCollection=34db8026 (Accessed: 22 April 2024).

Dans cet article, les biais algorithmiques liés à la reconnaissance faciale sont abordés de manière éthique notamment au sujet de l’ethnicité. On apprend que de nos jours de nombreux modèles présentent des lacunes importantes en manière de décision et précision en particulier lorsque ceux-ci sont appliqués à des groupes ethniques minoritaires.

Personnellement nous trouvons cet article controversé et discutable mais permet de montrer malgré tout un souci important qu’est le biais algorithmique et qui existe réellement dans ce domaine.

III. PROPOSITION

*A. Face detection*

Notre première étape dans notre projet est d’identifier un visage dans une image (Face detection en Anglais). Cela consiste à identifier le visage présent dans une image et de retourner une image comportant uniquement le visage.

Dans un premier temps, nous avons étudié un Dataset issu du film Jurassic Park". On a un total de 208 images au format (128 x 128 x 3). Et dans notre propre dataset nous avons 500 images.  
  
Nous avons une quantité conséquente d'images, et sur une image il peut y avoir plusieurs visages donc d’extraire les visages est nécessaire. Mais il faut également des données claires, et sans élément en arrière plan qui pourraient tromper notre algorithme. (bruit de fond)

Après avoir extrait les visages, nous les ajoutons dans une liste. On récupère également les labels associés, grâce aux noms de répertoires où se trouvent les images.   
Après cela nous normalisons les données, les visages avec une division par 255 (valeurs de pixels). Et les étiquettes en les transformant en valeurs numériques (0,1,2,3….)

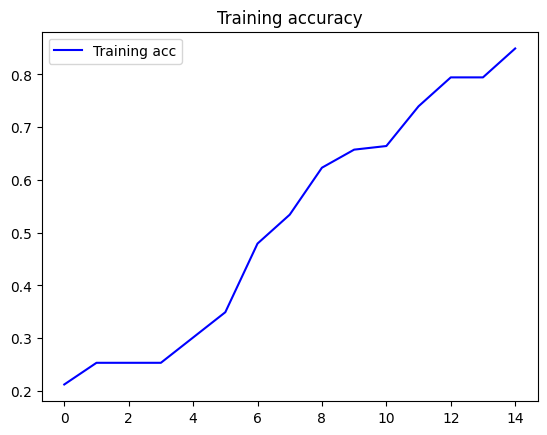
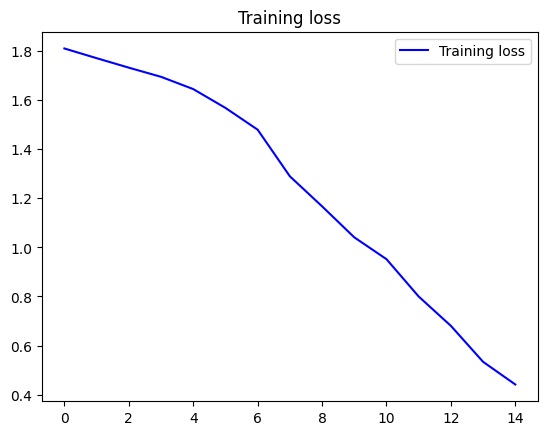
Nous utilisons la fonction train\_test\_split de scikit-learn pour séparer l'ensemble de données en deux ensembles d'entraînement (X\_train, Y\_train) et de test (X\_test, Y\_test). Nous choisissons une taille de test de 30% (0.30) ce qui implique que 70% des données sont utilisées pour l'entraînement et 30% pour le test.

Notre convnet est construit de manière à optimiser la classification d’images. Nous avons des couches de convolution avec des filtres de tailles croissante (32, 64, 128 et 256). Chaque couche est suivie d'un MaxPooling afin de réduire l’overfitting en réduisant le nombre de paramètres. Ensuite on flatten() notre modèle pour afin d'utiliser des layers Dense afin de mettre les différents neurones en relation. Et biensur la fonction softmax en finale en précisant le nombre de sujets étudiés. Nous avions des performances initiales pas très élevés, mais en ajustant nos paramètres nous arrivons à obtenir :

Une précision de 0.59 sur les données de validation contre 0.85 sur les données d'entraînement. Notre modèle surprend légèrement, cela pourrait se résoudre en diminuant la complexité ou en augmentant le nombre de données.

|  |  |
| --- | --- |

Figure 1 - Before vs after face detection

  
Figure 2 - Performance du modèle

*B. Pose estimation*

Dans cette deuxième étape de notre projet, nous appliquons une méthode de pose estimation, qui consiste à aligner les vissages dans un référentiel similaire. Pour cela nous utilisons un modèle comprenant 68 points de repère. Ces points permettent d’estimer la pose du visage.  
Cette méthode est très pertinente, elle permet d’aligner les visages ce qui facilite la comparaison et l’apprentissage de notre modèle.  
  
Notre jeu de données reste similaire, nous avons 208 images avec une shape de (128 x 128 x 3). Les labels sont identiques. A la suite de l’application de la méthode, nous obtenons une liste “aligned” comprenant tous les visages alignés.

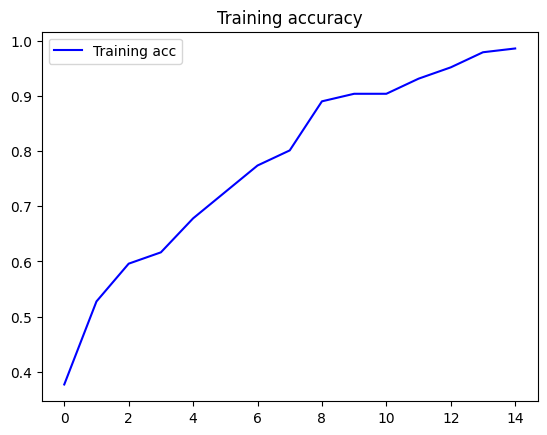
Pour la répartition des training\_set et test\_set, la méthode utilisée est la même que dans la première partie, on sépare l'ensemble de données en deux ensembles d'entraînement (X\_train, Y\_train) et de test (X\_test, Y\_test). Nous choisissons une taille de test de 30% (0.30) ce qui implique que 70% des données sont utilisées pour l'entraînement et 30% pour le test.

Les résultats sont de 0,76 sur les données de test contre 0,59 précisément ce qui est une nette amélioration. La perte est passée de 0,98 à 0,85.   
Cette amélioration n'est pas étonnante, en effet l’application de la pose estimation rend les données plus homogènes ce qui permet au modèle d’apprendre de manière plus efficace.

|  |  |
| --- | --- |

Figure 1 - Before vs after pose estimation



  
Figure 2 - Performance du modèle

*C. Face encoding*

Dans cette troisième étape, nous changeons de méthode, au lieu d'entraîner nos convnet a classer des visages, on entraîne ( ici utilise des modèles pré entraîné) à générer un ensemble de mesures, 128, pour chaque visage. Cette méthode est beaucoup plus efficace, en effet le modèle une fois entraîné on peut l’appliquer en quelqu’un seconde sur une photo non connue. Mais également une réduction des erreurs dues à des facteurs tels que l'éclairage et l'angle de la caméra, ainsi qu'une plus grande efficacité dans la recherche et l'analyse de grandes bases de données d'images faciales.

Nous utilisons 209 images, sur lesquels ont fait 128 mesures donc (209 x 128)

Le format des layers Dense sera utilisé pour ce modèle.

données analysées. Le nombre d'unités nécessaires pour que notre modèle soit applicable sera de 512-256-256 et disposera de fonctions de type "Relu". La fin sera une fonction 'softmax'.

Le nombre de sorties dépend du nombre de labels. Il est logique, en effet, d'établir les chances de chaque individu potentiel sur les images.

Les résultats sont de 0,95 sur les données de test contre 0,76 précisément ce qui est une nette amélioration. La perte est passée de 0,85 à 0,16.



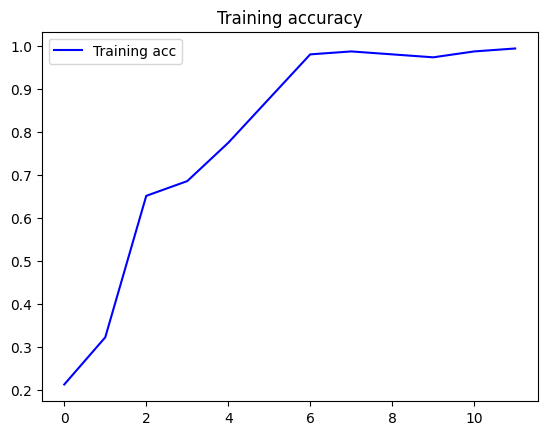


Figure 1 - Performance du modèle

*D. Face recognition*

Le 'Face Recognition' est un processus de détection faciale permettant notamment de vérifier la similitude entre différents visages. Diverses utilisations sont aujourd'hui présentes au quotidien notamment dans la sécurité, la surveillance, la vérification d'identité... La 'Face Recognition' se démarque de la 'Face Detection' en reconnaissant la personne (ou tout du moins tenter de prédire la personne) quand la 'Face Detection' se limite à simplement détecter un visage et non le reconnaître.

Nous aurons recours aux algorithmes suivants : "KNN", "SVM", "Logistic Regression" et enfin "CNN". Ces algorithmes sont différents et permettent donc de déterminer quelle approche serait la plus pertinente parmi cette variété d'algorithmes.

Les performances sont globalement toutes très bonnes. En revanche, CNN semble clairement se démarquer*.*

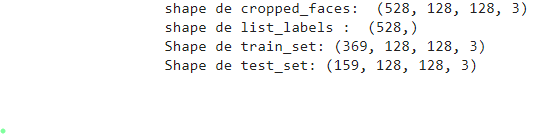
*E. Personal dataset*

Pour construire notre dataset, nous avons pris entre 50 et 100 photos par personnes avec la fonctionnalité rafale de nos téléphones avec 2 secondes de délai afin d’avoir le temps de changer de position, d'angle et d’expression du visage. Nous avons ensuite rangé ces photos dans les dossiers des personnes concernées servant de labels pour notre modèle.

A la fin de l'entraînement notre modèle marche très bien avec les personnes de la dataset. Ce qui signifie que lorsque un inconnu est testé on va obtenir un résultat incohérent mais ce résultat est attendu puisque cette personne n’est pas la dataset.

Notre dataset est plus grande que l’originale avec un nombre de photos s'élevant àenviron 500 images réparties à travers 8 personnes.

En outre, nous avons testé notre modèle avec une photo extérieure à notre base de données, et il a correctement identifié les personnes présentes. Les meilleurs résultats obtenus avec notre propre dataset s'expliquent par plusieurs facteurs : une augmentation de la quantité de données (environ 67 images par personne, contre environ 35 dans le dataset "Jurassic Park"), ainsi que par une meilleure clarté et lisibilité des images.



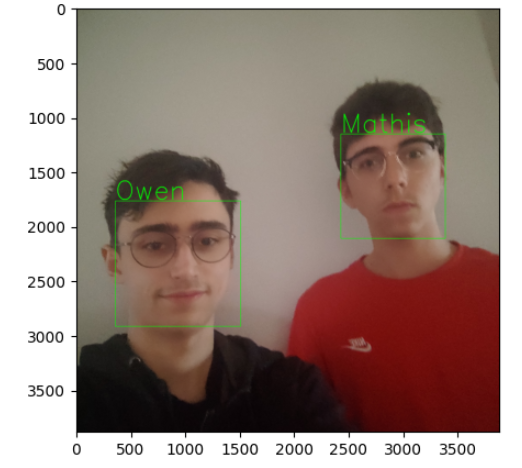


Figure 1 - Test reconnaissance sur image

*F. Extra - Bias analysis*

Dans cette partie nous allons faire quelques critiques sur notre modèle notamment par le biais algorithmique (voir l’index pour la définition).

Ce biais algorithmique est un souci dans le monde de l’informatique puisque cela peut dans certaines situations être difficile à détecter et ensuite entraîner d’énorme erreurs/anomalies avec comme risque par exemple de retarder un projet. Un exemple simple de biais algorithmique dans notre vie de tous les jours est dans les recommandations d’un catalogue de vidéos à la demande comme proposé par Amazon. Dans ce genre de catalogue, si un utilisateur regarde quelques documentaires alors l’algorithme risque d’uniquement recommander des documentaires même si l’utilisateur est également fan d’autres types de films qui ne seront alors plus recommandés.

Pour en revenir à nos données, le biais de notre modèle doit se trouver dans nos photos qui n’ont pas été prises dans des environnements variés. Cependant pour limiter le biais nous avons pris soin de changer d'angle et d’expression du visage dans nos photos

CONCLUSION

Durant ce projet ayant comme objectif la reconnaissance faciale, nous avons pu comprendre et appliquer différents algorithmes. Dans un premier temps pour extraire les visages de nos photos pour ensuite les classifier.

Finalement, ce projet permet de valider nos connaissances sur les réseaux de neurones obtenus lors des enseignements du professeur Laurent NAJMAN.

REFERENCES

* Teoh, K. *et al.* (2021) *IOPscience*, *Journal of Physics: Conference Series*. Available at: https://iopscience.iop.org/article/10.1088/1742-6596/1755/1/012006/meta (Accessed: 22 April 2024).
* Zhi, H. and Liu, S. (2018) *Face recognition based on genetic algorithm*, *Journal of Visual Communication and Image Representation*. Available at: https://www.sciencedirect.com/science/article/abs/pii/S1047320318303389 (Accessed: 22 April 2024)
* <https://www.ibm.com/fr-fr/topics/machine-learning>
* <https://fr.wikipedia.org/wiki/Biais_algorithmique>
* (2020) *Facial-recognition algorithms: A literature review - paramjit kaur, Kewal Krishan, Suresh K. Sharma, Tanuj Kanchan, 2020*. Available at: https://journals.sagepub.com/doi/abs/10.1177/0025802419893168 (Accessed: 22 April 2024).
* Perkowitz, S. (2021) *The bias in the machine: Facial recognition technology and racial disparities*, *MIT Case Studies in Social and Ethical Responsibilities of Computing*. Available at: https://mit-serc.pubpub.org/pub/bias-in-machine/release/1?readingCollection=34db8026 (Accessed: 22 April 2024).
* <https://link.springer.com/chapter/10.1007/978-981-15-0372-6_30>
* <https://www.cv-foundation.org/openaccess/content_iccv_2015_workshops/w11/html/Hu_When_Face_Recognition_ICCV_2015_paper.html>