

NNL REPORT

Reconnaissance faciale : Masque ou non-Masque

Prepared by
Quentin BEAUCHET
Pierre Kouyoumdjian
Raphaël JULIEN

Supervised by
Enrico FORMENTI

Contents

1	Introduction	2
1.1	Présentation du groupe	2
1.2	Présentation du sujet	2
2	L'annotation d'image	2
2.1	Le fonctionnement de l'annotation d'image	2
2.2	La production de donnée	4
3	La prédiction d'images	4
3.1	Nos modèles	4
3.2	La génération des datasets	5
3.3	La classification	5
3.4	La localisation	7
4	Résultat	10
5	Conclusion	11

1 Introduction

1.1 Présentation du groupe

Nous sommes trois étudiants en première année de Master Informatique à l'Université Côte d'Azur : Quentin BEAUCHET, Pierre KOUYOUMDJIAN, Raphaël JULIEN.

1.2 Présentation du sujet

Le but du projet est de pouvoir détecter les visages des personnes et de dire pour chacun d'eux s'ils ont un masque ou non. Nous avons d'abord dû créer un programme d'annotation d'images pour créer nos données avec lesquelles nous avons entraîné notre modèle d'intelligence artificielle.

2 L'annotation d'image

2.1 Le fonctionnement de l'annotation d'image

Lors du lancement de l'application, celle-ci demandera à l'utilisateur de choisir une image à importer. Une fois l'image choisie, la fenêtre principale avec l'image apparaîtra. L'utilisateur peut alors créer ses boxes en faisant un clic gauche. Une fois le clic relâché, une nouvelle fenêtre apparaîtra permettant à l'utilisateur d'associer une catégorie à la box. Il pourra plus tard éditer la box en effectuant un clic droit sur celle-ci.

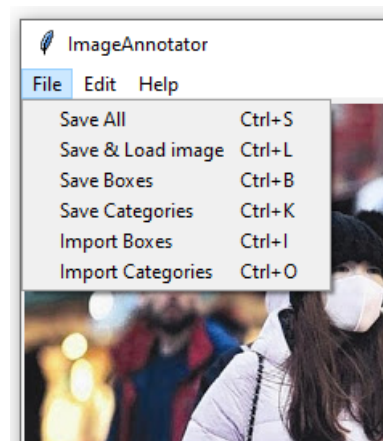


Figure 1: Onglet File

- Save All : Exporte les images des boxes annotés ainsi que les catégories et les coordonnées des boxes.
- Save : Save All et permet à l'utilisateur d'ouvrir une nouvelle image
- Save Boxes : Sauvegarde les coordonnées des boxes dans un fichier .csv et .json
- Import Boxes : Permet à l'utilisateur d'importer des boxes à partir d'un fichier .csv ou .json
- Import Categories : Permet à l'utilisateur d'importer des catégories à partir d'un fichier .csv ou .json

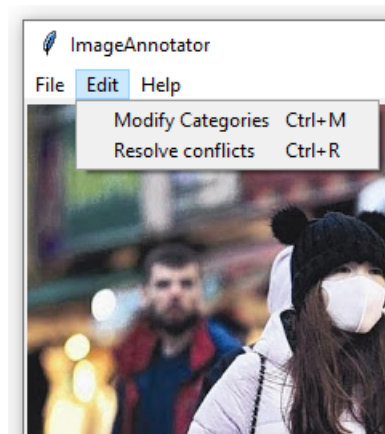


Figure 2: Onglet Edit

- Modify Categories : Permet à l'utilisateur de modifier, ajouter ou supprimer une ou plusieurs catégories.
- Resolve Conflicts : Si une boîte a intersection avec d'autres boîtes pour plus de 20% de sa surface, elle est supprimé.

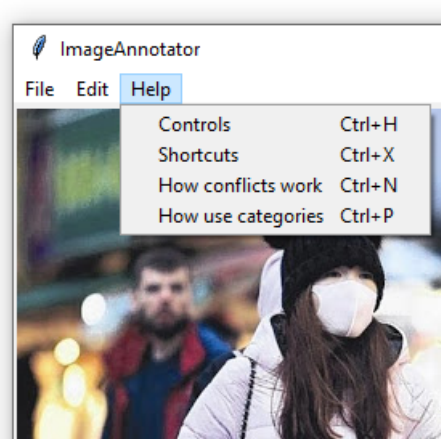


Figure 3: Onglet Help

- Controls : Détails comment créer une box et l'éditer
- Shortcuts : Détails les raccourcis clavier disponible
- How conflits work : Détails comment les conflits fonctionnent
- How use categories : Détails les fonctionnalités en lien avec les catégories

La structure de nos fichiers:

- L'exportation des catégories s'effectuent dans le dossier settings du projet.

- L'exportation des images des boxes s'effectuent dans le dossier output, ils sont mis dans un sous-dossier correspondant à leur catégorie.
- L'exportation des boxes s'effectuent dans le dossier settings du projet.

2.2 La production de donnée

Pour la production de donnée, nous avons récupéré des images du site Kaggle que nous avons ensuite trié pour garder une variété de situation différentes.

Ensuite nous avons importé chacune de ces images dans notre programme d'annotation et nous les avons annoté avec leur box et leur catégorie correspondante. Les données nécessaires sont exporté dans leur dossier correspondante.

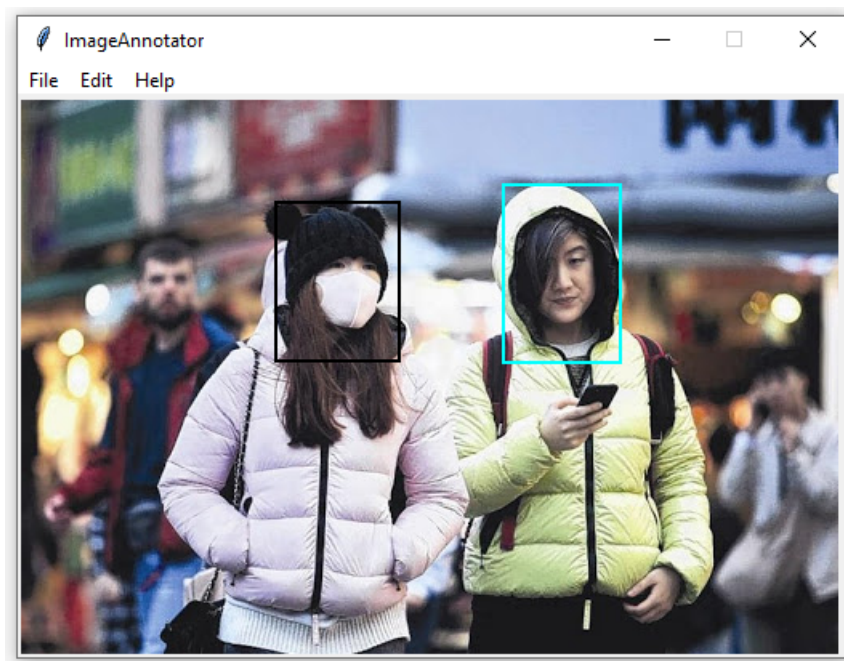


Figure 4: Exemple d'annotation d'une image

3 La prédiction d'images

3.1 Nos modèles

Dans le sujet vous nous avez envoyé vers un tutoriel pour de la classification mais vous n'avez pas précisé comment réaliser la bounding box d'un visage, et nous n'avons pas réussi à installer l'api tensorflow pour de la détection d'objets, c'est pourquoi nous avons choisi de réaliser nos prédictions en deux étapes.

En premier lieu notre modèle de localisation va déterminer la position d'un visage dans une image puis notre modèle de classification va prédire si celui-ci porte un masque ou non.

3.2 La génération des datasets

Notre dataset proviens de [Kaggle](#) et contient 853 images de personnes avec et sans masques. Il contient aussi des annotations de leurs visages et s'ils portent un masque ou non.

Dans le sujet vous nous avez suggéré d'utiliser notre propre annotateur crée précédemment pour récupérer les bounding box de nos images. Au départ c'est ce que nous faisons mais au bout d'un moment on c'est rendu compte que c'était une perte de temps car nous disposions déjà des annotations avec notre dataset.

Aussi le format que vous suggériez, **myimage-bb-XxY-W-H.jpg** est très inconvenient car il aurait fallu découper le nom du fichier pour récupérer la position de sa bounding box puis aller chercher l'image original dans un autre fichier pour la localisation des visages et le plus gros inconvenient est le fait que cela génère des images inutiles car on ne les utilise que quand le programme est en cours donc autrement elles prennent de la place pour rien.

C'est pour ça que nos fichiers ne contiennent que les images de départ et leurs annotations, et c'est lors de l'exécution que l'on va génère nos datasets **X_train**, **y_train**, **X_test** et **y_test** et les sauvegarder sous format pickle et ce pour chaque modèles.

Ces donnés sont sous la forme:

- Classification:
 - **X** [imageDuVisage,...]
 - **y** [[1,0],[0,1],[1,0]...] avec [1,0] s'il porte un masque et [0,1] sinon.
- Localisation:
 - **X** [imageEntiere,...]
 - **y** [[xmin,ymin,xmax,ymax],...] c'est la bounding box du visage.

Pour ne pas avoir de biais nous prenons autant de visages avec masques que sans pour la classification et pour la localisation vu qu'on ne peut détecter qu'une seule personne a la fois, on ne garde que les images avec une seule personne et dont le visage fait plus de 100x100 pixels.

3.3 La classification

Pour la classification nous avons extrait les données de nos XML dans annotations et nous trions les photos par rapport à leur catégorie. Nous sommes obligés de faire ça pour pouvoir après avoir autant de masque que de visage non masqué en tronquant le surplus de photo possédant un masque. A partir d'ici nous séparons notre liste en 2, la zone de test et la zone d'entraînement qu'on sauvegarde au format pickle pour la charger quand nous entraînons le modèle.

Notre modèle possède plusieurs couches de neurones, tout d'abord nous augmentons nos données grâce aux couches **RandomFlip** et **RandomRotation** puis nous avons les couches **Conv2D**, **Activation**, **MaxPool2D**, **Flatten**, **Dense**, **Rescaling**, **RandomFlip** et **RandomRotation**. Nous entraînons notre programme et le sauvegardons dans **Models/ClassificationModel.h5**, puis nous rappelons le modèle après avoir fais la localisation que nous détaillerons plus bas pour qu'il prédicte la catégorie des images passées. Enfin il calcule la probabilité que ce soit un masque ou pas et renvoie la catégorie possédant la plus grande probabilité.

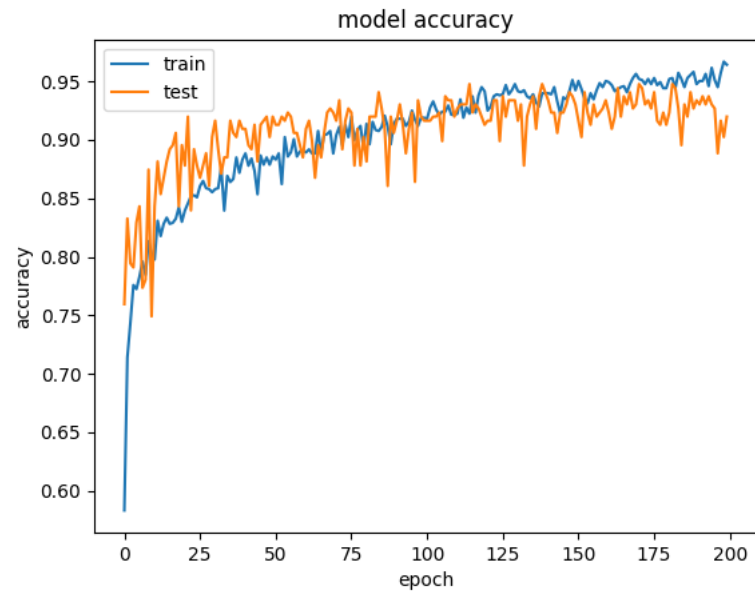


Figure 5: L'évolution de l'accuracy par rapport aux epochs

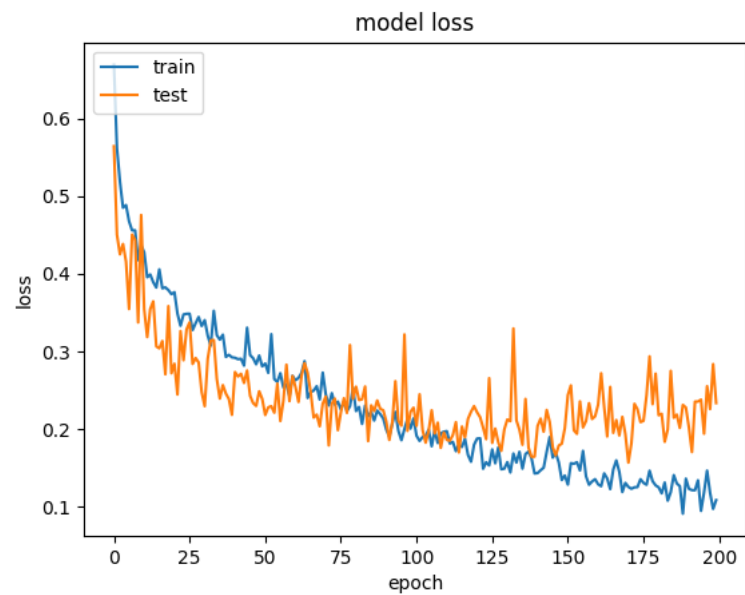


Figure 6: L'évolution du loss par rapport aux epochs

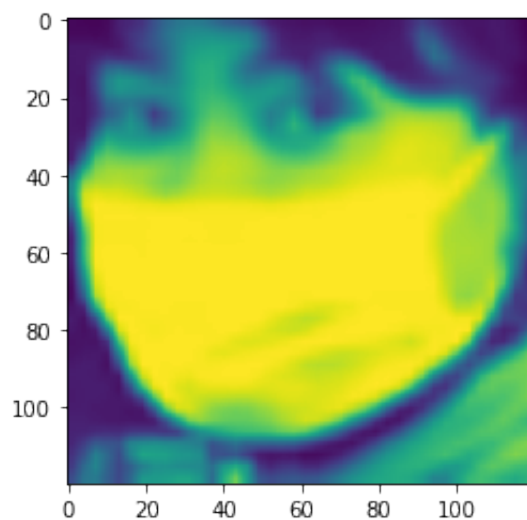


Figure 7: Exemple de prédiction d'un visage avec un masque

Prédiction: ('Mask', array([[1.0000000e+00, 6.0228587e-09]], dtype=float32))

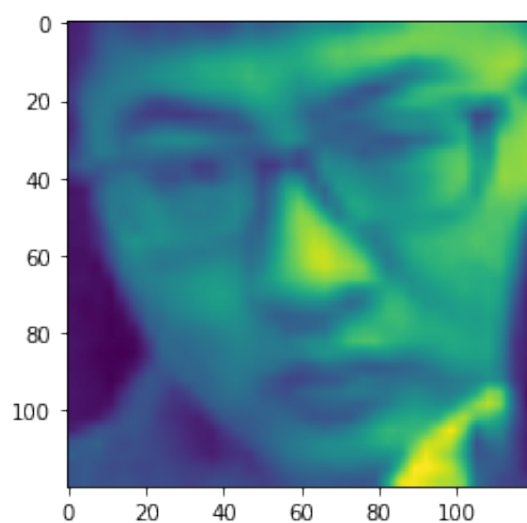


Figure 8: Exemple de prédiction d'un visage sans masque

Prédiction: ('Nude', array([[0.00605881, 0.9939412]], dtype=float32))

3.4 La localisation

Après avoir trié nos données pour ne garder que les visages assez gros et seuls sur l'image, on crée notre modèle CNN selon ce [tutoriel](#) avec une couche **Dense** terminale qui reçoit 4 entiers permettant de définir la bounding box. Enfin on le compile avec une perte calculée sur l'erreur

quadratique moyenne car c'est ce qui exprime le mieux la différence entre des positions. Puis nous entraînons notre modèle avec un callback **ReduceLROnPlateau** ce qui nous permet d'être plus précis après chaque epoch. De plus **EarlyStopping** nous permet d'arrêter prématurément l'entraînement pour éviter le sur apprentissage.

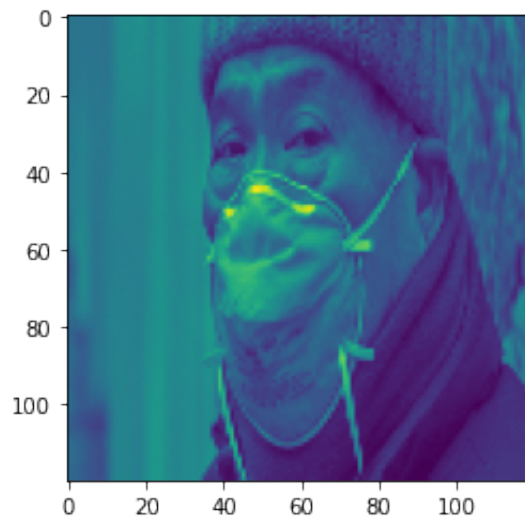


Figure 10: Exemple d'une image avant la prédiction de la localisation du visage

Prédiction: [37, 33, 82, 92]

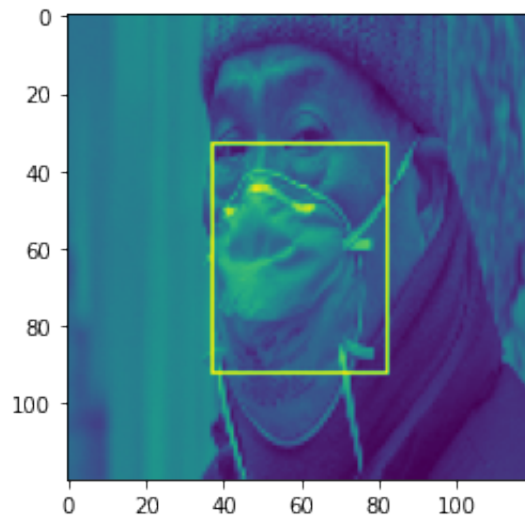


Figure 11: Exemple d'une image après la prédiction de la localisation du visage

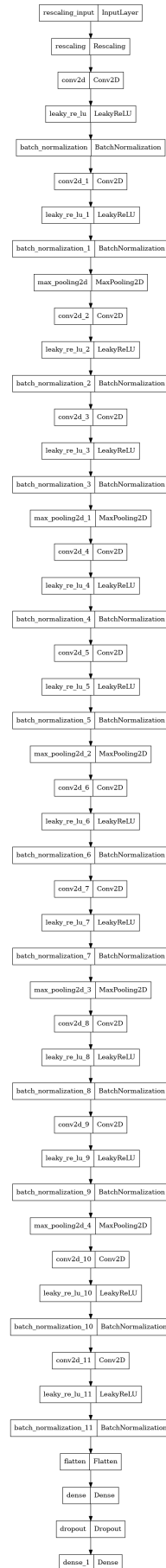


Figure 12: Modèle de notre localisation de bounding box

4 Résultat

Notre modèle de classification est vraiment très précis car il atteint pratiquement 0.91% de précision pour une perte de seulement 0.17% pour nos données de test.

Notre modèle de localisation lui est un peu moins bon car il a encore 87.9 de perte et 0.80% de précision. En combinant les deux on est alors capable de prédire si une personne porte un masque ou non.

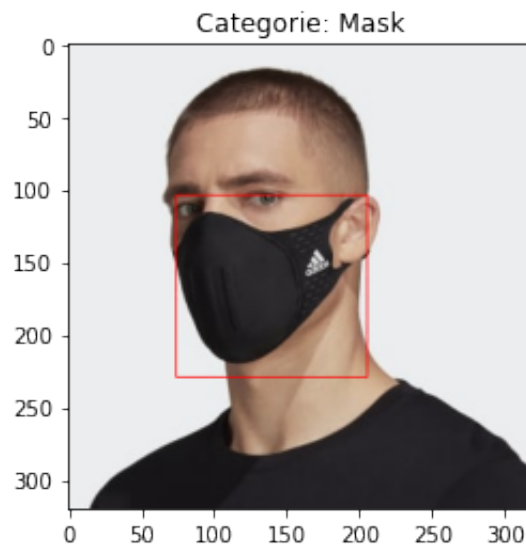


Figure 13: Exemple de prédiction d'image selon la catégorie

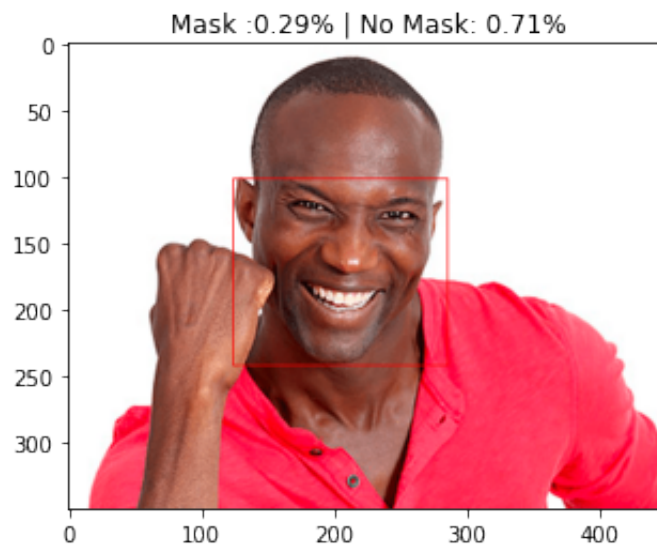


Figure 14: Exemple de prédiction d'image selon la probabilité de chaque catégorie

5 Conclusion

Nous sommes capables de prédire 9 fois sur 10 si une personne porte un masque ou non et de définir les contours de son visage, du coup nous sommes assez fiers de ce que nous avons réalisé. En revanche il y a quand même de sérieux problèmes, notre modèle est incapable de fonctionner sur des images contenant plusieurs personnes et sur des images ne contenant personne. Une piste d'amélioration serait de reconnaître les différents types de masque comme le ffp2, le tissu, etc.

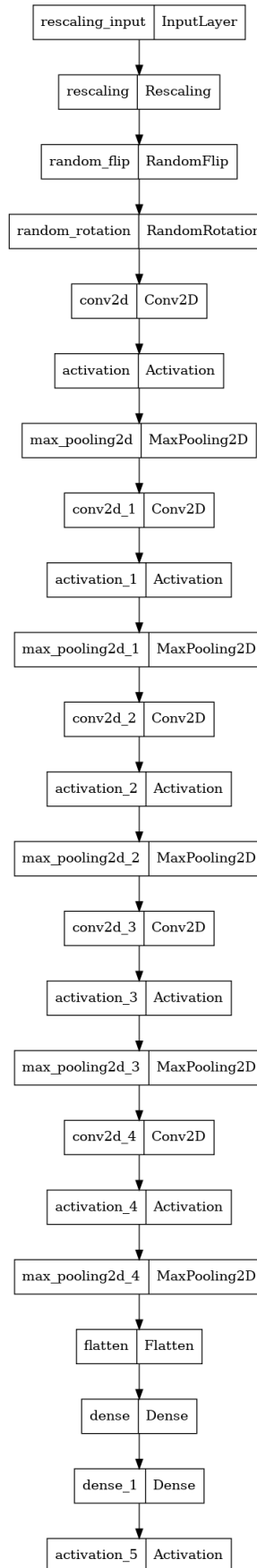


Figure 9: Modèle de notre Classification des visages portant un masque