



MASTER INFORMATIQUE

Neural Network and Learning Facial recognition: the mask/no mask case

Auteurs:

Antoine Cousson
Dylann Batisse
Margaux Schmied

Professeur:

Enrico Formenti

2021/2022

Table des matières

1	Introduction	1
2	Partie I	1
2.1	Data	1
2.2	Choix de conception	1
2.3	Répartition des tâches	2
2.4	Comment utiliser	2
2.5	Amélioration possible du logiciel	2
3	Conclusion	3

1 Introduction

Le but de cette première partie est de réaliser un logiciel d'annotations d'images en python pour fournir un jeu de données à l'IA de la seconde partie du projet.

Nous avons dû donner la possibilité à l'utilisateur d'encadrer une ou plusieurs parties de l'image puis de l'assigner à une catégorie. Chaque cadre doit pouvoir être éditable, pour changer la catégorie ou supprimer le cadre. L'utilisateur doit pouvoir sauvegarder toutes ses annotations. En ce qu'il concerne les catégories il est nécessaire de pouvoir en importer, en ajouter ou en supprimer.

2 Partie I

2.1 Data

Pour récolter les données nécessaires à l'entraînement de l'IA nous nous sommes rendu sur [kaggle.com](https://www.kaggle.com) où nous avons recherché un jeu de données varié d'humains et autres. Nos critères étaient d'y trouver des personnes dans diverses situations, de différents âges, ethnies et en nombre variés pour avoir une diversité suffisante, afin d'éviter les problèmes liés à des jeux de données trop réduits. Cependant les images nous semblent petites, nous verrons dans la seconde partie du projet si elles conviennent bien à nos besoins.

Pour la partie annotations, nous avons stocké, un identifiant unique à chaque image, les coordonnées des zones annotées associées au titre d'une catégorie, la taille de l'image ainsi que le chemin vers l'image dans un JSON. En revanche la sauvegarde n'est pas automatique, il faut enregistrer à la fin de chaque session d'annotations dans un fichier JSON.

2.2 Choix de conception

Méthodologie : Pour mieux s'organiser au cours du projet nous avons utilisé l'outil de versioning git. Cet outil nous a permis de travailler à plusieurs sur différentes branches en simultané et enfin de fusionner le tout pour avoir un produit tout le temps fonctionnel.

Technologies : **Python3.9.9**

Pour réaliser l'interface graphique de ce projet nous nous sommes plutôt rapproché d'une librairie externe nommée Qt, développée en C++ qui selon nous est mieux que le GUI classique de Python «**tkinter**».

Qt est une librairie très puissante, moderne et récente c'est pourquoi nous voulions en apprendre plus dessus.

Les différentes librairies utilisées au cours du projet :

- PySide6==6.2.1 (librairie Qt adaptée pour python)
- Pillow==8.4.0 (redimensionner des images)
- Shapely==1.8.0 (intersections entre les rectangles)

2.3 Répartition des tâches

Pour la répartition des tâches nous nous sommes organisé via GitHub et son système d'issues, mais aussi via discord où nous recensons les bugs à réparer ainsi que les fonctionnalités à implémenter, nous passons par des phases de test où nous essayons de mettre le logiciel à l'épreuve pour détecter les bugs potentiels.

Antoine s'est occupé de l'import et de l'export des données et des catégories, Dylann de toute la gestion des zones d'annotations, leur chevauchement, leur affichage et Margaux de la gestion des catégories, l'ajout, la suppression et la modification.

2.4 Comment utiliser

NOTE : Pensez à mettre les droits d'exécution sur le fichier **build.sh** pour installer les dépendances et lancer l'application.

Pour lancer :

```
#!/bin/bash
chmod u+x build.sh
./build.sh
```

- Pour **supprimer une annotation** faites cliquer droit sur la zone annotée de l'image.
- Pour **éditer une annotation** double-cliquez dessus, la fenêtre des choix de catégorie va apparaître et vous pourrez éditer votre annotation.
- Pour **éditer une catégorie** double-cliquez dessus pour qu'elle devienne éditable, une fois la modification faite, appuyez sur "Change Category" pour valider le changement, qui se reportera dans le fichier JSON d'annotations.
- CTRL + S : pour sauvegarder.
- CTRL + O : pour ouvrir un fichier.
- CTRL + SHIFT + O : ouvrir un dossier.
- CTRL + W : pour quitter une image.
- CTRL + H : pour ouvrir ce README.

2.5 Amélioration possible du logiciel

Nous avons pensé à implémenter un système de Ctrl-Z Ctrl-Y à l'avenir sur l'édition des zones d'annotations, c'est une fonctionnalité qui paraît simple en apparence mais nous demandent beaucoup de travail en arrière-plan, invisible pour l'utilisateur. Nous avons testé le logiciel à la recherche de bug, ce qui a permis d'en régler plusieurs lors de cette période. Nous n'en avons plus trouvé après les corrections de ceux recensés, cela ne veut pas dire que le logiciel est désormais sans bug, il est impossible de garantir une telle chose mais lors d'une utilisation normale il est peu probable que l'utilisateur en rencontre un.

Le projet étant très intéressant nous avons imaginé continuer le logiciel et peut-être faire en sorte d'annoter des vidéos pour que l'on puisse à partir d'une séquence obtenir une multitude d'images et entraîner l'IA sur celle-ci voir même sur des films.

3 Conclusion

Lors de cette première partie nous avons donc réalisé ce logiciel de façon qu'il nous convienne pour travailler pour la seconde partie. Nous avons implémenté les fonctionnalités qui nous semblaient utiles, pour pouvoir annoter efficacement les images. En effet si nous avons un logiciel ergonomique, nous faciliterons la phase de création du jeu de données d'images, pour nourrir notre algorithme.

Cette partie nous a permis de prendre en main les librairies graphiques en python et de faire attention à l'efficacité du logiciel.