
Détection d'Infections Pulmonaires par Intelligence Artificielle, et Utilisation de Marqueurs ArUco pour la Modélisation 3D des Poumons

Auteur :
Elias Mouaheb

2 décembre 2023

Résumé

Ce projet, entièrement réalisé utilisant Python, avec Blender pour la simplification de modèle, se concentre sur l'amélioration des méthodes de détection des infections pulmonaires en utilisant deux technologies : les marqueurs ArUco pour la modélisation tridimensionnelle des poumons et les algorithmes d'intelligence artificielle pour l'analyse d'images radiographiques.

Les marqueurs ArUco sont des images semblables à des codes QR avec des motifs carrés noirs et blancs. Un programme utilisant une caméra permet de modéliser un objet 3D sur le marqueur reconnu, de le faire se mouvoir ainsi que l'agrandir en rapprochant le marqueur de la caméra et inversement. Dans la première phase, ils sont déployés pour créer une représentation 3D détaillée des poumons, offrant ainsi une visualisation précise de la structure pulmonaire.

Simultanément, un modèle d'intelligence artificielle entraîné détecte et cartographie les zones infectées à partir d'images radiographiques. L'utilisation de ces deux approches permet non seulement une identification rapide et précise des infections, mais offre également une représentation visuelle immersive des résultats diagnostiques.

Ce projet repousse les frontières de la convergence entre la vision par ordinateur et l'intelligence artificielle, ouvrant de nouvelles perspectives pour l'analyse des maladies pulmonaires. Les implications potentielles de cette recherche sont significatives, allant de diagnostics améliorés à une meilleure compréhension des processus pathologiques pulmonaires, et pourrait apporter un soutien de taille à la cardiologie notamment.

Table des matières

1	Mise en place de marqueurs ArUco	2
2	Carte d'Attention	3
3	Modélisation 3D	3
4	Difficultés et Limites	5

1 Mise en place de marqueurs ArUco

- (i) **Initialisation** : Au début du processus, les marqueurs ArUco 1 doivent être générés avec les paramètres désirés, notamment l'identifiant et la dimension du marqueur, pour fournir un support à reconnaître au programme. Cette étape établit la fondation nécessaire pour des reconstructions 3D fiables.

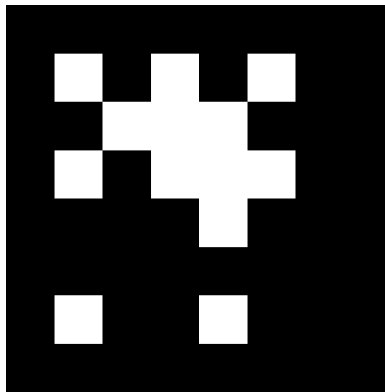


FIGURE 1 – Exemple de marqueur ArUco 6X6, d'id 24.

- (ii) **Calibration** : La phase de calibration ajuste finement le système de marqueurs, permettant aux marqueurs d'avoir une bien meilleure précision spatiale. Cette étape est essentielle pour aligner parfaitement les marqueurs avec la structure anatomique des poumons. A ce titre, il est requis de fournir au programme une certaine quantité d'images montrant le marqueur ArUco sous différents angles, au minimum 20. Cette étape permettra d'obtenir les différents paramètres de la caméra utilisée, qui seront essentiels pour toute modélisation. Une fois la calibration faite, les marqueurs ArUco peuvent être utilisés pour modéliser un objet 3D dans l'espace.

2 Carte d'Attention

Une carte d'attention (attention map) est une représentation visuelle qui met en évidence les zones spécifiques d'une image ou d'un ensemble de données qui ont été particulièrement "notées" ou "soulignées" par un modèle, souvent basé sur des réseaux neuronaux. Ces zones sont celles où le modèle a concentré son "attention" lors de l'analyse. Il s'agit d'un procédé très utilisé dans le domaine de la vision par ordinateur notamment.

Pour ce projet, le modèle d'intelligence artificielle qui a été utilisé pour la création de carte d'attention, et a préalablement été entraîné sur un dataset d'images radiographiques, a été récupéré d'un dépôt Github [1], tandis que les différentes images radiographiques comme sur la Figure 2 proviennent d'un autre dépôt Github [2] récupéré depuis un dataset Kaggle.

L'utilisation du modèle de génération de carte d'attention s'est fait via un environnement de travail Python avec Conda, afin d'avoir accès aux versions spécifiques des dépendances requises. Enfin, un programme Python permettant de placer des cercles autour de zones rouges sur une image passée en paramètre permet d'obtenir les coordonnées et la taille des différentes infections visibles sur la carte d'attention. Ce sont ses données, stockées dans un fichier généré par le programme, qui seront utilisées pour ajouter des sphères au modèle de poumon 3D.



FIGURE 2 – Image Radiographique.

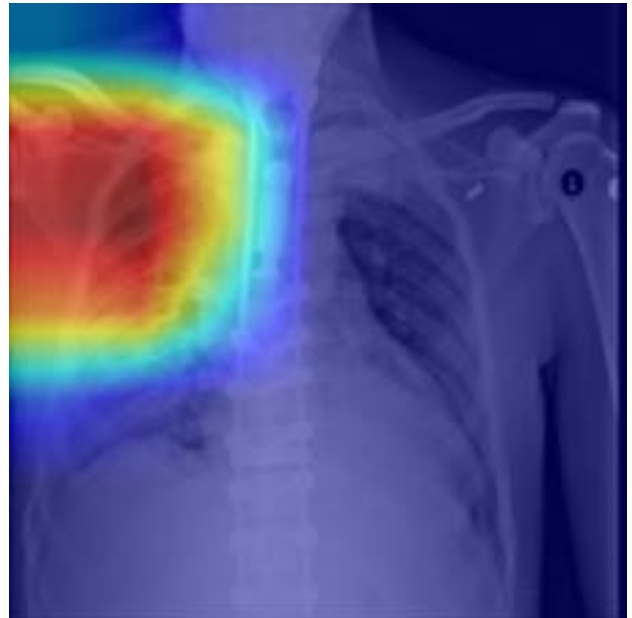


FIGURE 3 – Carte d'Attention.

3 Modélisation 3D

Les modèles 3D gratuits de véritables poumons s'avérant compliqué à trouver sur internet, il a fallu utiliser des modèles moins réalistes, comme ce modèle provenant de Sketchfab avec un moindre niveau de détail, mais possédant des textures permettant une meilleure visualisation. C'est ce modèle qui sera utilisé tout au long de ce projet.

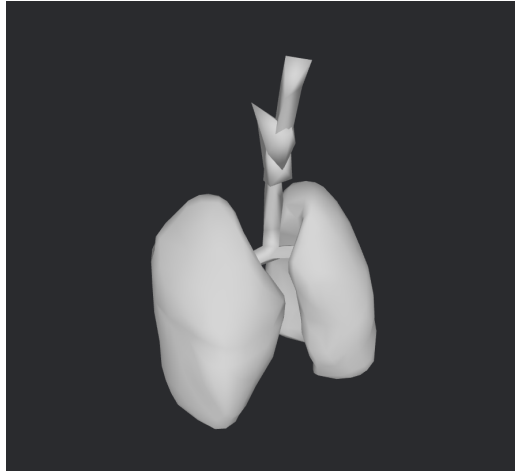


FIGURE 4 – Modèle 3D de poumon de synthèse.

OpenGL permet de l'utiliser facilement dans un programme Python, en y mêlant d'autres dépendances telle numpy. La modélisation du poumon sur le marqueur ArUco a nécessité plusieurs dépôts Github qui ont permis de s'en inspirer mais aussi de mieux comprendre le fonctionnement de cette technologie[3] **Aruco'Python**. Il aura toutefois fallu modifier les programmes permettant d'afficher le modèle sur le marqueur ArUco pour prendre en compte les caractéristiques du modèle. L'opacité de ce dernier a été divisée par 2 pour permettre de voir les sphères représentant les infections à l'intérieur, ainsi que son nombre de triangles par 10 pour ne pas ralentir le programme de modélisation.

Les sphères sont ajoutées à la scène affichée en récupérant les coordonnées depuis le fichier Python précédemment créé 2. Les coordonnées récupérées ont ensuite été normalisées afin que les sphères apparaissent au même endroit sur le modèle que les infections sur la carte d'attention dont ils sont issus. En plus de la manière classique pour zoomer mise en place par les marqueurs ArUco, où faire apparaître le marqueur plus grand agrandit aussi le modèle qu'il affiche, et inversement, une fonction a été rajoutée pour permettre de zoomer et dézoomer facilement. Cette fonction compte le nombre de pixels rouges (resp. vert), et si le nombre détecté est au-dessus du seuil prévu, le modèle de poumon 3D ainsi que les différentes sphères en son sein apparaîtront comme plus grand (resp. petit).



FIGURE 5 – Modélisation du poumon avec sphères.

4 Difficultés et Limites

Plusieurs parties de ce projet ont nécessité une quantité importante de recherche, tant dans les technologies utilisées que dans la compréhension du contexte. Pour l'utilisation de modules comme torch et OpenGL par exemple, la documentation ainsi que les différents dépôts Github auront été des outils essentiels. Dans les références de ce rapport sont différentes sources m'ayant grandement aidé à appréhender cette étape, et envers lesquels je suis reconnaissant.

L'intégration des différents modules aura aussi été un obstacle conséquent, chaque dépôt Github ayant besoin de versions spécifiques de dépendances. Trouver un modèle de poumon gratuit a aussi été difficile.

Malgré les différents obstacles, ce projet a pu aboutir à un résultat utilisable. Subsistent des axes d'amélioration, conséquences de limites atteintes. La première limite est la complexité du modèle de poumon. Le modèle utilisé dans ce projet se veut simple, un modèle à peine plus précis entraînant un temps de chargement du programme de 20 secondes pour 100 000 triangles environ, le rendant peu pratique à l'utilisation. Un modèle réaliste n'est donc pas envisageable pour ce projet.

La seconde limite est liée à la représentation des infections sur le modèle. En effet, le choix de privilégier la modélisation de sphères plutôt que d'utiliser spécifiquement la carte d'attention entraîne une perte de précision, étant donné que les infections ne présentent généralement pas une forme sphérique parfaite. Dans le contexte médical, où la précision est cruciale, une marge d'imprécision aussi importante ne peut être tolérée, compromettant ainsi l'utilité potentielle du projet dans le domaine de la médecine.

Références

- [1] P. MISRA. "Xrays and GradCAM". GitHub repository. (2021), adresse : <https://github.com/priyavrat-misra/xrays-and-gradcam>.
- [2] IEEE8023. "covid-chestxray-dataset". GitHub repository. (2021), adresse : <https://github.com/ieee8023/covid-chestxray-dataset>.
- [3] BRYCEQING. "OPENCV_AR". GitHub repository. (année $_{OPENCV_A R}$), adresse : https://github.com/BryceQing/OPENCV_AR.
- [4] RAUBCAMAIONI. "OpenCV_Position". GitHub repository. (2017), adresse : https://github.com/RaubCamaioni/OpenCV_Position.
- [5] GEEKLIB. "AR-BXT-AR4Python". GitHub repository. (2016), adresse : <https://github.com/GeekLiB/AR-BXT-AR4Python>.
- [6] AJAYMIN28. "Aruco_python". GitHub repository. (2017), adresse : https://github.com/ajaymin28/Aruco_python.