

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

RAPPORT TECHNIQUE PRÉSENTÉ À MONSIEUR LUC DUONG DANS LE CADRE
DU COURS COMPRÉHENSION DE l'IMAGE

PROJET SYNTHÈSE

PAR
KATIA KACI
ALEJANDRO OLIVARES HERNADEZ

MONTRÉAL, LE 30 AVRIL 2025

Table des matières

Introduction	3
Approche proposée.....	4
Résultats et discussion	9
Conclusion	19
Bibliographie	20

Introduction

Dans le milieu hospitalier, la complexité et l'urgence des interventions médicales, combinée à la pénurie de personnel, augmente le risque d'erreurs humaines. En 2023, près de 9 900 postes d'infirmiers étaient vacants au Québec, forçant ainsi les établissements de la santé à faire appel à du personnel temporaire ou moins expérimenté (IEDM, 2023). Cette surcharge de travail et la fatigue qui peut l'accompagner augmentent le risque d'erreurs, notamment lors de l'administration de médicaments. Parmi ces erreurs, les erreurs médicamenteuses (EM), c'est-à-dire toute erreur évitable survenant lors du processus de médication (préparation, administration, etc.), représentent un problème majeur. Aux États-Unis, elles seraient responsables de pertes s'élevant à environ 42 milliards de dollars par an et causeraient entre 44 000 et 98 000 décès annuellement (AMCP, 2019). Les erreurs LASA (Look-Alike, Sound-Alike) sont des erreurs particulièrement préoccupantes. Elles surviennent lorsqu'un professionnel confond deux médicaments dont leur nom ou leur emballage sont similaires. Ce type d'erreur est d'autant plus fréquent lorsque plusieurs seringues sont préparées à l'avance pour un même patient, ce qui rend leur identification plus difficile.

Ainsi, pour assurer un suivi rigoureux et réduire les risques d'erreur, il est important de bien identifier chaque médicament avant de l'administrer à un patient. Grâce aux avancées en vision par ordinateur et en intelligence artificielle, il est désormais possible d'automatiser une partie de cette tâche, souvent effectuée manuellement par des infirmières ou des médecins, afin d'améliorer la rapidité du processus d'administration et de réduire le risque d'erreur humain. Par conséquent, face à cette problématique, notre projet propose une application capable de détecter automatiquement la couleur de l'étiquette d'un médicament pris en photo. Cette détection permet d'identifier la famille du médicament et d'enregistrer l'heure et la date de son administration au patient. En plus de la reconnaissance de couleur, nous intégrons également un modèle de reconnaissance de texte afin d'extraire le nom du médicament de l'étiquette, ainsi que les autres informations importantes s'il y a lieu, telles que le dosage, le numéro de lot ou la date d'expiration. L'objectif est donc de proposer un système rapide, performant et fiable, capable d'aider les professionnels de la santé à minimiser les erreurs d'administration et à optimiser la rapidité du processus d'administration des médicaments.

Ce rapport présentera d'abord l'approche technique utilisée pour développer notre système, en détaillant notamment les méthodes d'annotation, d'entraînement des modèles de détection et de segmentation avec YOLOv8, ainsi que l'algorithme de détection de couleur et de reconnaissance de texte (OCR). Par la suite, les résultats obtenus seront analysés et discutés. Enfin, nous analyserons les contributions de notre travail, ses limites actuelles, les défis rencontrés ainsi que les pistes d'amélioration envisagées pour de futurs développements.

Approche proposée

Notre application a pour objectif d'améliorer le processus d'administration des médicaments en détectant automatiquement la couleur de l'étiquette d'un médicament à l'aide de la caméra d'un appareil mobile. Grâce à cette détection, l'application identifie la famille du médicament et enregistre l'heure et la date de son administration au patient. Ce projet se penchera sur la recherche et le développement d'une façon simple pour identifier l'étiquette placée sur la seringue et identifier sa couleur. Pour ce faire, nous avons utilisé YOLOv8 (*You Only Look Once* version 8) afin de nous permettre d'identifier les composantes importantes d'une image prise par notre téléphone cellulaire.

YOLOv8 (<https://yolov8.com/>) est un modèle d'intelligence artificielle conçu principalement pour la détection et la segmentation d'objets sur des images. Contrairement aux méthodes traditionnelles qui traitaient la détection en plusieurs étapes distinctes (comme l'extraction de régions d'intérêt suivie d'une classification), YOLO aborde le problème de manière directe. Lorsqu'une image est fournie en entrée, YOLO la divise en une grille, et pour chaque cellule de cette grille, prédit simultanément la présence d'un objet, la classe de l'objet détecté et les coordonnées de sa boîte englobante. Dans le cas particulier de la segmentation, les contours précis de l'objet sous forme de masque sont également prédits. Tout cela est réalisé en un seul passage dans le réseau neuronal, d'où le nom *You Only Look Once*. Ce seul passage rend ce modèle très rapide et adapté aux applications en temps réel. YOLOv8 utilise également une structure de réseau plus légère et optimisée, capable de mieux généraliser même avec des ensembles de données plus petits, ce qui est très important dans notre cas. Le modèle ajuste ses prédictions en apprenant à minimiser plusieurs fonctions de perte spécifiques, ce qui lui permet de devenir progressivement meilleur pour localiser et reconnaître les objets présents dans l'image. C'est pour toutes ces raisons que nous avons décidé d'utiliser YOLOv8 comme modèle d'apprentissage machine dans notre projet.

Nous posons comme hypothèse que ce modèle nous permettra de détecter la seringue et l'étiquette de la seringue directement dans une image et de cibler précisément la zone à analyser pour en déterminer la couleur.



Figure 1. Exemples d'étiquettes pour identifier la famille de médicaments

Dans la figure ci-dessus, on observe différentes seringues étiquetées selon un code couleur standardisé dans le domaine de la santé. Chaque couleur est associée à une classe précise (ex. : hypnotiques, curarisants, vasopresseurs, etc.). En détectant automatiquement la couleur de l'étiquette, nous cherchons ainsi à en déduire la famille du médicament et à conserver automatiquement cette information.

Comme un modèle YOLOv8 ne semble pas être publiquement disponible pour effectuer la tâche de détection des seringues, nous devrons alors l'entraîner avec nos propres images de seringues annotées. Pour ce faire, nous avons capturé et utilisé 233 photos de seringues. Ces images ont été capturées sous différents angles, avec différents arrières-plans, à différentes échelles et sous différentes luminosités. De plus, différentes étiquettes ont été utilisées afin que le système n'associe pas la détection à une couleur, mais bien à la seringue ou l'étiquette elle-même. Afin d'annoter les images pour les différentes méthodes, nous avons utilisé deux solutions Open Source.

La première, *LabelImg* (<https://pypi.org/project/labelImg/>), nous a permis d'annoter la seringue dans l'image à l'aide d'un rectangle de taille variable. Cette solution génère directement des fichiers .txt contenant la position des quatre points du rectangle. Chaque fichier .txt, qui correspond à une image spécifique, peut ensuite être utilisé directement par YOLOv8 pour l'entraînement du modèle.

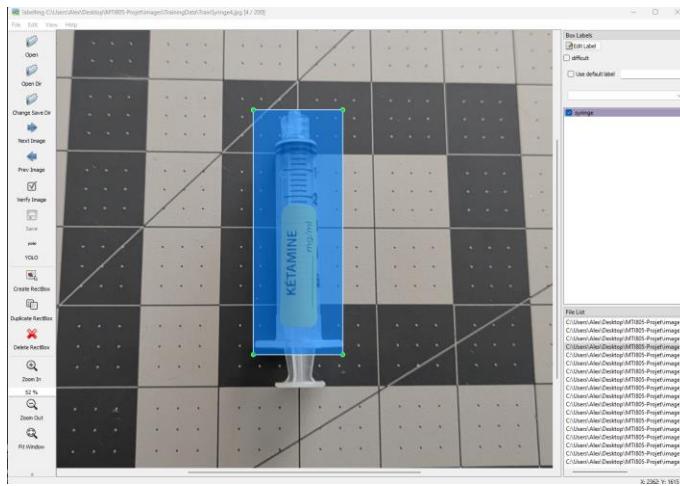


Figure 2. Annotation de la seringue avec *LabelImg*

La deuxième solution, utilisée spécifiquement pour la segmentation, était *LabelMe* (<https://github.com/wkentaro/labelme>). Cette solution permet de segmenter directement l'étiquette de la seringue dans l'image en traçant des points autour de celle-ci. Une fois le traçage fait, les données de l'annotation sont sauvegardées dans un fichier JSON correspondant à l'image. Pour transformer les données de l'annotation en format accepté par YOLO, nous avons utilisé une librairie python *labelme2yolo*. Cette librairie nous a permis de transformer les fichiers d'annotation JSON pour chaque image en fichier texte sous un format nécessaire pour YOLOv8.

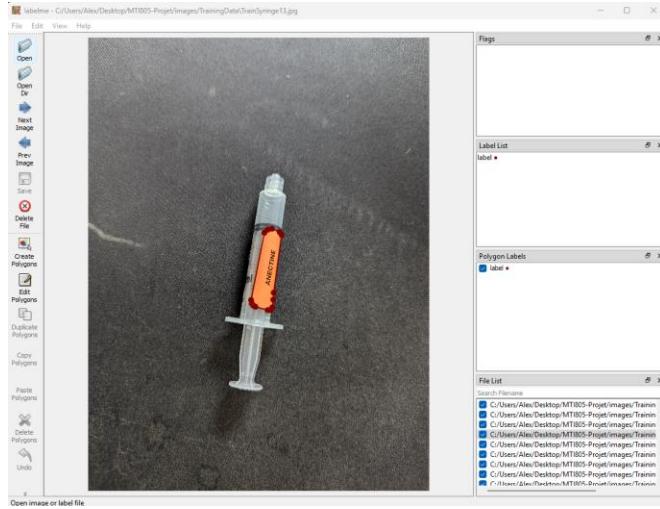


Figure 3. Segmentation et annotation de l'étiquette sur la seringue avec LabelMe

En tout, 200 images ont été utilisées pour l'entraînement de tous nos modèles (détection et segmentation), et 33 images supplémentaires ont été utilisées comme données de validation.

Pour l'entraînement de la détection de la seringue dans l'image, nous avons entraîné et comparé tous les modèles YOLOv8 disponibles avec 50 époques. Ces modèles étaient : YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l et YOLOv8x. Comme on peut le constater sur la figure 3 ci-dessous, la série des modèles YOLOv8 (ligne bleue) surpassé systématiquement les anciennes versions (YOLOv5, v6, v7) en termes de précision (mAP50–95). De plus, les modèles plus petits comme YOLOv8n et YOLOv8s sont les plus rapides et offrent un poids réduit, ce qui les rend particulièrement intéressants pour une utilisation en temps réel sur des appareils avec peu de ressources. À l'inverse, les modèles plus lourds comme YOLOv8x offrent des performances supérieures en précision, mais avec un entraînement beaucoup plus lent.

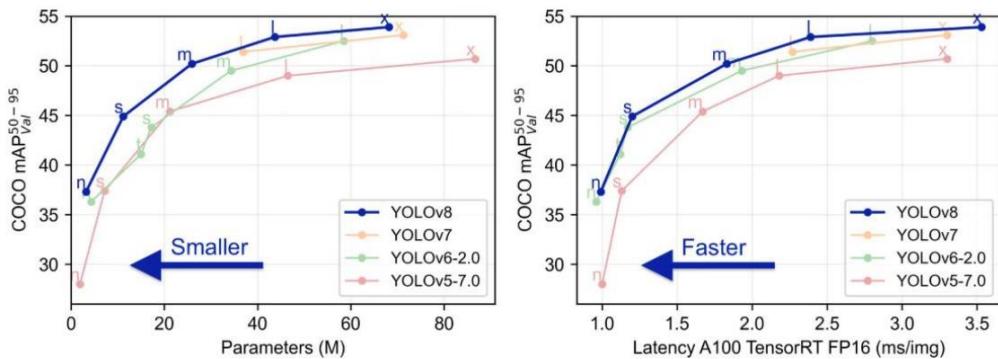


Figure 4. Comparaison des performances des modèles YOLOv5 à YOLOv8. Tirée de <https://docs.ultralytics.com/models/yolov8/>

Notre objectif était de comparer les performances de ces différents modèles dans notre contexte afin de déterminer lequel offrait le meilleur compromis entre précision et coût d'entraînement. Les résultats de cette comparaison seront présentés dans la section suivante de ce rapport.

Concernant les modèles de segmentation, nous avons entraîné uniquement les modèles YOLOv8n-seg, YOLOv8s-seg, YOLOv8m-seg et YOLOv8x-seg. Le modèle YOLOv8l-seg n'a pas été entraîné, car, lors des premiers essais, nous avons constaté que le modèle YOLOv8m-seg était déjà un peu moins performant que le modèle YOLOv8s-seg pour notre cas d'utilisation et nos conditions d'entraînement. Plus de détails à ce sujet seront également présentés dans la section suivante.

Lors de l'entraînement avec YOLOv8, plusieurs techniques spécifiques sont utilisées pour améliorer la robustesse et la performance du modèle. Notamment, YOLO effectue également une augmentation automatique des données (*autoaugmentation*) en appliquant des transformations aléatoires, telles que la rotation, le redimensionnement, le changement de luminosité et le *flipping* horizontal pour améliorer la capacité de généralisation du modèle. Il est important de noter que nous avons augmenté la rotation par défaut de 10 degrés maximum à 90 degrés maximum pour nous assurer que les étiquettes puissent être reconnues sous n'importe quelle rotation. De plus, YOLO utilise une méthode appelée Mosaic Augmentation, où plusieurs images d'entraînement sont automatiquement assemblées ensemble en une seule image. Cela permet au modèle d'apprendre à détecter des objets dans des contextes visuels variés, avec différentes échelles et positions, ce qui augmente sa capacité de généralisation. Ces approches permettent de maximiser l'efficacité de l'entraînement, et ce, même avec le relativement petit nombre d'images de départ qu'on avait.



Figure 5. Méthode Mosaic Augmentation utilisée pour l'entraînement avec YOLOv8

Finalement, pour la détection de la couleur de l'étiquette, nous avons utilisé une méthode inspirée de l'article par Fleyeh (2004). La méthode est simple, après la segmentation de l'étiquette, nous convertissons l'image en format HSB à l'aide d'OpenCV et analysons uniquement les pixels situés à l'intérieur de la région segmentée, qui correspond à l'étiquette de la seringue. Nous calculons séparément la moyenne des valeurs de teinte (H), de saturation (S) et de luminosité (B). Il est important de noter qu'en OpenCV, la valeur de la teinte varie de 0 à 179, ce qui représente l'intervalle complet de 360 degrés qu'aurait normalement une couleur sous format HSB. À partir de ces moyennes, nous déterminons ensuite la couleur de l'étiquette en les comparant aux intervalles prédéfinis, ce qui nous permet d'identifier avec exactitude la couleur de l'étiquette. Lors de la création des intervalles de classification des couleurs, nous avons dû porter attention à bien délimiter les limites de chaque couleur afin d'assurer une distinction précise entre elles.

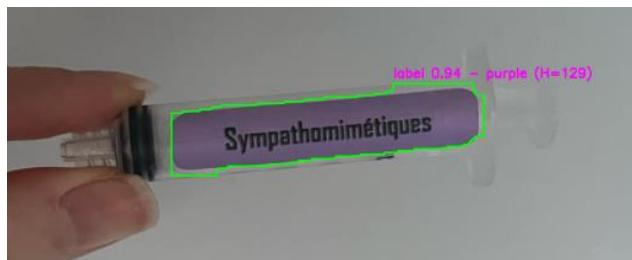


Figure 6. Exemple de la segmentation et de la détection de couleurs sur une seringue étiquetée.

En plus de la détection de la couleur, nous avons également extrait automatiquement le texte inscrit sur l'étiquette de la seringue, comme le nom du médicament. Pour ce faire, nous avons utilisé *Tesseract*, un moteur de reconnaissance optique de caractères (OCR) *open source* développée initialement par HP et maintenant maintenu par Google.

Notre approche consistait à appliquer une série d'étapes de prétraitement sur l'image, et ce, afin d'améliorer la lisibilité par Tesseract. Plus précisément, nous avons converti l'image en niveaux de gris, puis appliqué un seuillage de type *Otsu* (*cv2.THRESH_OTSU*) combiné à une inversion binaire pour faire ressortir le texte clair sur le fond foncé. Ensuite, une opération de dilatation a été appliquée à l'aide d'un noyau rectangulaire de taille (18, 18), ce qui permet de regrouper les caractères proches en blocs, ce qui facilite ainsi leur détection sous forme de contours.

Chaque contour détecté a été interprété comme une zone textuelle potentielle. Ces zones ont été découpées une par une, puis passées à Tesseract pour l'extraction du texte. Le texte extrait était ensuite nettoyé (suppression des sauts de ligne, des espaces de trop, etc.) et enregistré dans un fichier texte dont le nom correspondait à celui de l'image analysée.

Résultats et discussion

Lors de nos essais, nous avons commencé par essayer de détecter la seringue à l'aide de notre modèle YOLO entraîné sur 200 images. Tous les modèles de YOLO, du YOLOv8n au YOLOv8x ont été essayés. Les résultats de la *Box Loss*, Précision et la valeur mAP50-95 sont présentés lors des prochaines figures. Le site *Explore Ultralytics YOLOv8* contient toute la documentation nécessaire pour comprendre ces métriques. (<https://docs.ultralytics.com/models/yolov8/> et <https://github.com/autogyro/yolo-V8>)

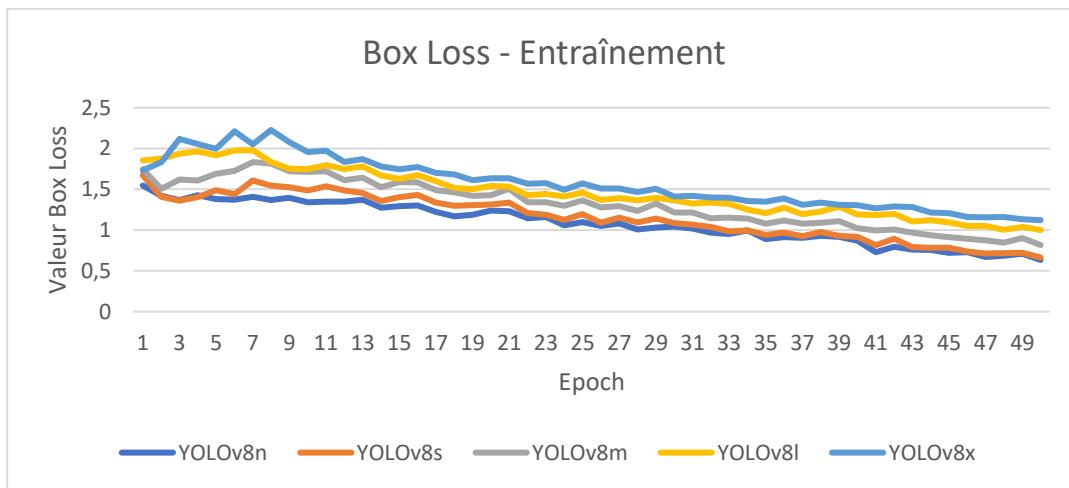


Figure 7. Résultats de la *Box Loss* pour les données d’entraînement à chaque *Epoch* pour tous les modèles de détection entraînés

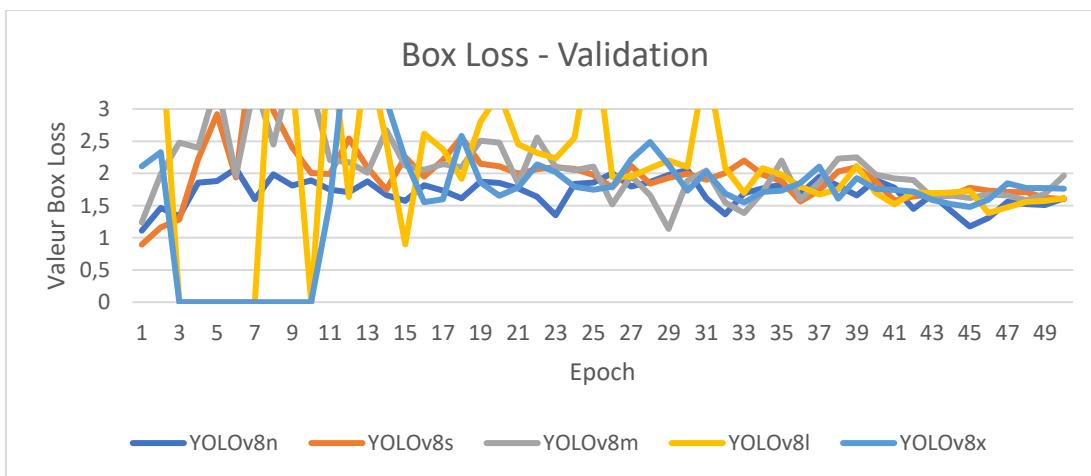


Figure 8. Résultats de la *Segmentation Loss* pour les données de validation à chaque *Epoch* pour tous les modèles de segmentation entraînés

Le *Box Loss* permet d'analyser la qualité de la prédiction des boîtes englobantes tout au long de l'entraînement (au fil des différents *Epoch*). L'objectif est de réduire cette valeur autant que possible. En principe, cette valeur devrait diminuer de manière continue avec les *Epoch*. Nous présentons lors des résultats deux graphiques, l'un pour les données d'entraînement (200 images) et l'autre pour les données de validation (33 images). La valeur du *Box Loss* sur les données de validation permet de déterminer à quel moment le système cesse de progresser avec les images actuelles. Autrement dit, lorsque la détection n'est plus améliorée malgré l'augmentation du nombre d'*Epoch*.

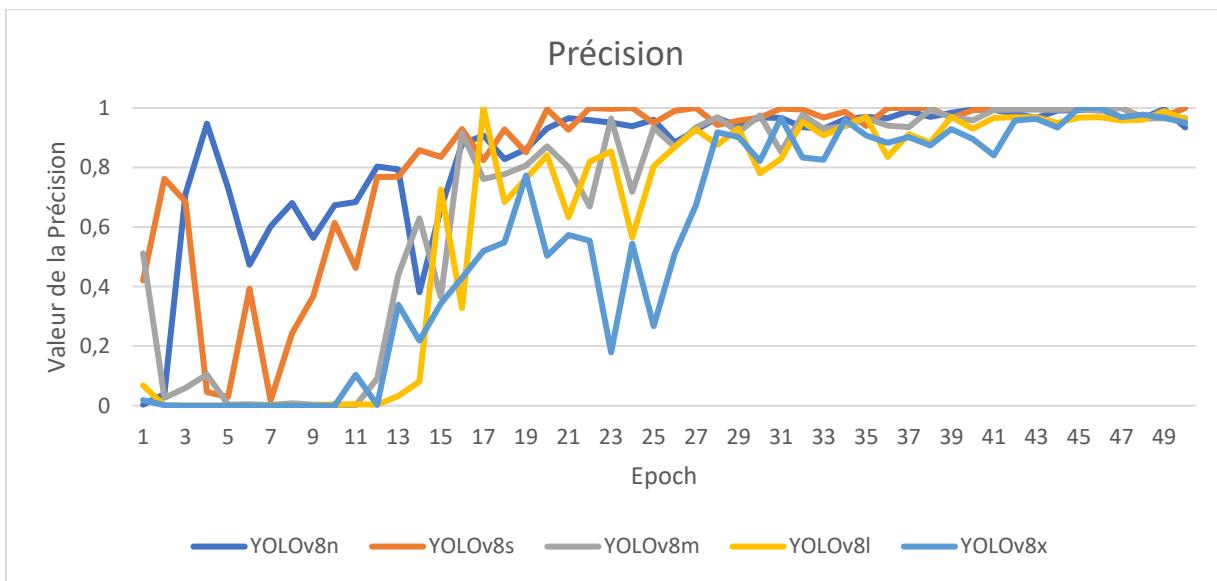


Figure 9. Résultats de la précision à chaque *Epoch* pour tous les modèles de détection entraînés

La précision est une mesure importante, car elle indique la proportion des détections correctes parmi toutes celles effectuées par le modèle sur une image. En d'autres mots, elle permet de savoir si les détections faites sont majoritairement des vrais positifs. L'objectif est donc d'être le plus proche de 100% de précision.

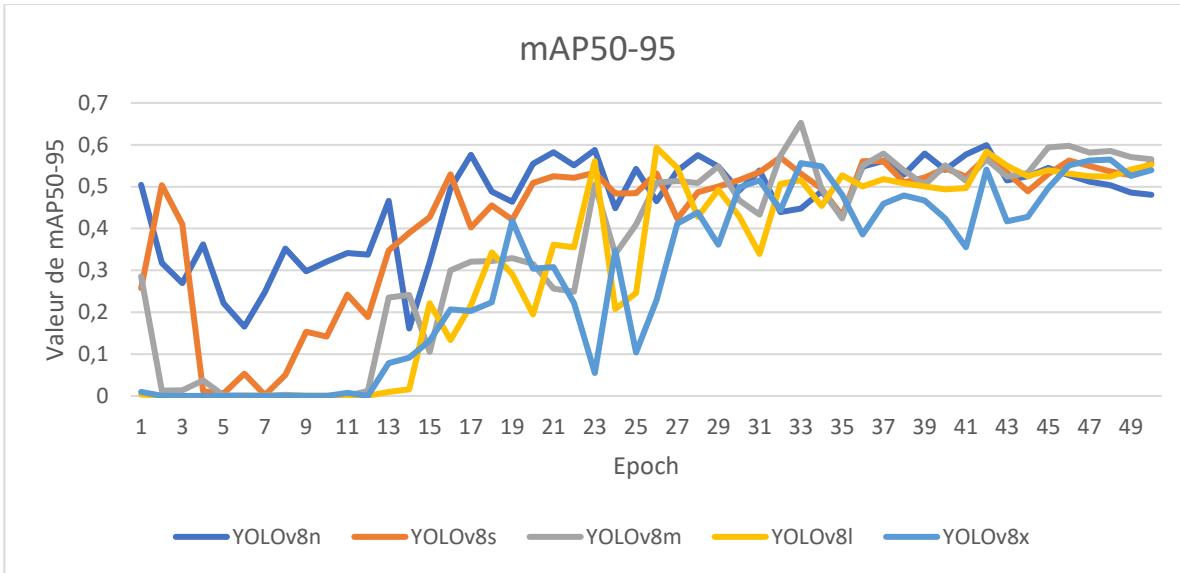


Figure 10. Résultats de la valeur *mAP50-95* à chaque *Epoch* pour tous les modèles de détection entraînés

Le *mAP50-95* est une mesure standard qui évalue la précision moyenne de la segmentation en calculant la moyenne des précisions obtenues à différents seuils allant de 50% à 95%. Cette métrique est très stricte, car elle récompense la précision avec laquelle les objets sont segmentés.

En analysant les résultats du *Box Loss*, on remarque que la valeur sur les données d'entraînement (Figure 7) diminue de façon continue, ce qui est très bon signe puisqu'elle indique que le système est en train d'apprendre. En revanche, sur les données de validation (Figure 8), cette valeur tend à se stabiliser autour de 40 *Epoch*, suggérant que le système n'apprend plus grand-chose de nouveau à partir de ses données d'entraînement. Nous avons donc conclu que les modèles n'avaient plus suffisamment d'information à extraire des images d'entraînement pour améliorer leur performance sur la validation. Il est toutefois possible que nous ayons arrêté l'entraînement un peu trop tôt. Pour confirmer cette hypothèse, un entraînement prolongé jusqu'à 100 *Epoch* aurait permis de vérifier si les modèles atteignent effectivement une stabilité. Cela dit, nous ne pouvons pas nous baser uniquement sur le *Box Loss* pour déterminer le meilleur modèle, car les résultats obtenus sont très similaires entre eux sur cette métrique. Cependant, les modèles YOLOv8n, YOLOv8s et YOLOv8m se démarquent par leur rapidité d'entraînement. En comparaison, YOLOv8l a pris environ 3 heures, tandis que YOLOv8x a nécessité près de 5 heures. Ces durées nous amènent à privilégier les modèles plus légers, qui sont nettement plus efficaces à entraîner.

Les valeurs de précision (Figure 9) et de *mAP50-95* (Figure 10) sont particulièrement intéressantes, car elles permettent d'avoir une meilleure idée de la performance réelle des différents modèles. Après environ 40 *Epoch*, la précision se stabilise près de 100% pour l'ensemble des modèles. Il est toutefois important de noter que YOLOv8s semble légèrement surpasser les autres en termes de précision. En ce qui concerne la valeur *mAP50-95*, les modèles

restent très proches les uns des autres au cours des 10 derniers *Epoch*. Les modèles YOLOv8s, YOLOv8m et YOLOv8l présentent ici des performances très similaires.

Il est très difficile de déterminer avec certitude quel modèle est le meilleur dans notre cas, car leurs performances sont très similaires. Il est clair que tous les modèles auraient probablement bénéficié d'un plus grand nombre d'images d'entraînement, et surtout d'un plus grand ensemble d'images de validation, afin d'améliorer la *Box Loss* sur les données de validation. Un jeu de données plus grand aurait potentiellement permis au système d'apprendre davantage de *features* et ainsi mieux détecter les objets. En ce qui concerne les métriques de précision et de *mAP50-95*, il n'est pas évident d'identifier un modèle clairement supérieur. C'est pourquoi nous avons choisi le modèle YOLOv8s, qui se distingue non seulement par sa rapidité d'entraînement, mais aussi parce qu'il figure parmi ceux ayant obtenu les meilleurs résultats. De plus, en le comparant au modèle YOLOv8x, censé être le plus performant, nous avons constaté que YOLOv8s semble nettement mieux performer sur des images jamais vues (voir exemple à la Figure 11). Il est possible que YOLOv8x ait été désavantage par le nombre relativement restreint d'images d'entraînement. Il serait donc pertinent d'explorer si un plus grand volume de données pourrait améliorer sa performance.

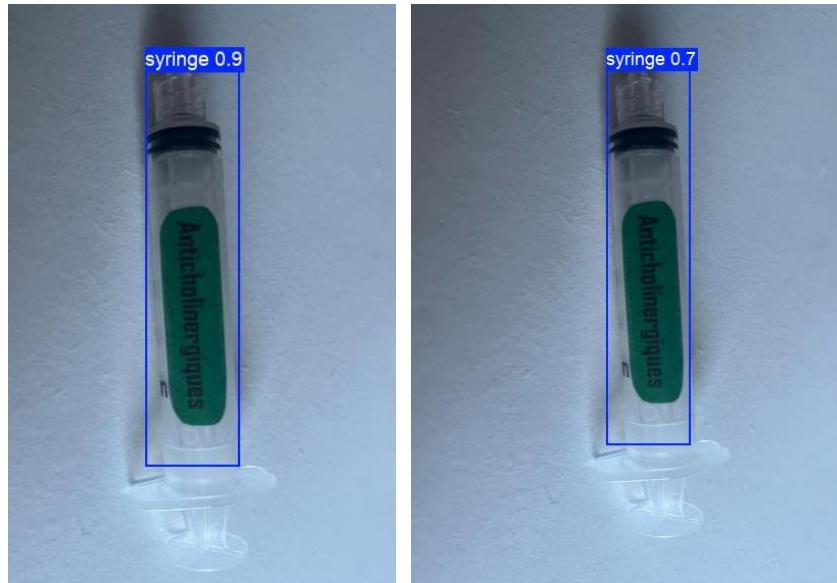


Figure 11. Résultats de la détection sur la même image entre les modèles YOLOv8s (image de gauche) et Yolov8x (image de droite).

Après avoir entraîné le système à détecter une seringue, nous avons pensé que la détection de l'étiquette par une segmentation pourrait être encore plus efficace et qu'en même temps, ceci pourrait nous permettre de détecter la seringue. C'est donc la prochaine étape que nous avons essayée.

Une fois la seringue détectée dans l'image, nous avons dû passer à la détection de l'étiquette dans celle-ci. Les modèles YOLOv8n-seg, YOLOv8s-seg, YOLOv8m-seg et YOLOv8x-seg ont été essayés et leurs résultats ont été comparés. Nous nous sommes surtout penchés sur l'analyse des résultats de la Segmentation Loss pour l'entraînement (200 images) et pour la validation (33 images), puis sur l'analyse de la précision et de la valeur mAP50-95. Voici les résultats obtenus :

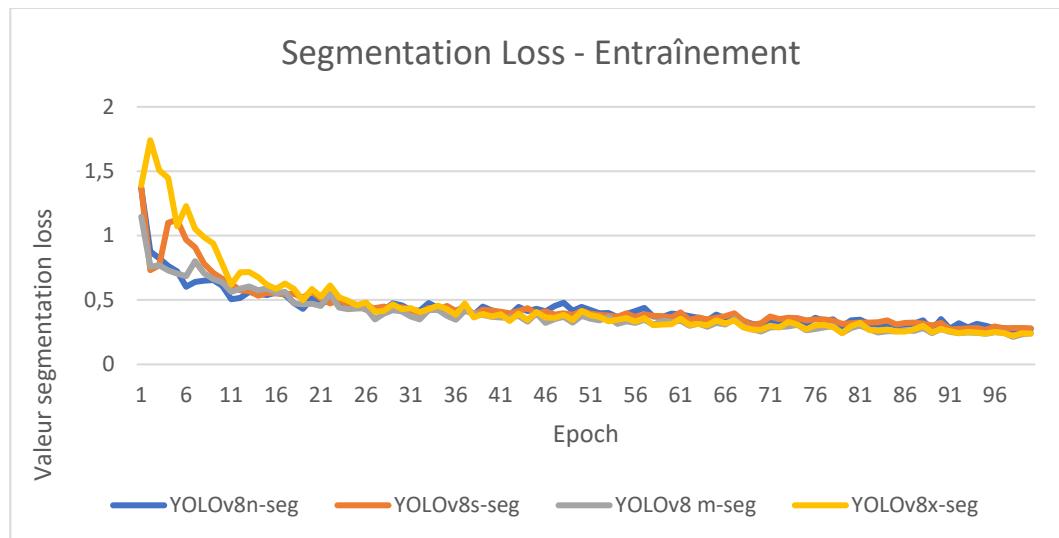


Figure 12. Résultats de la *Segmentation Loss* les données d'entraînement à chaque *Epoch* pour tous les modèles de segmentation entraînés

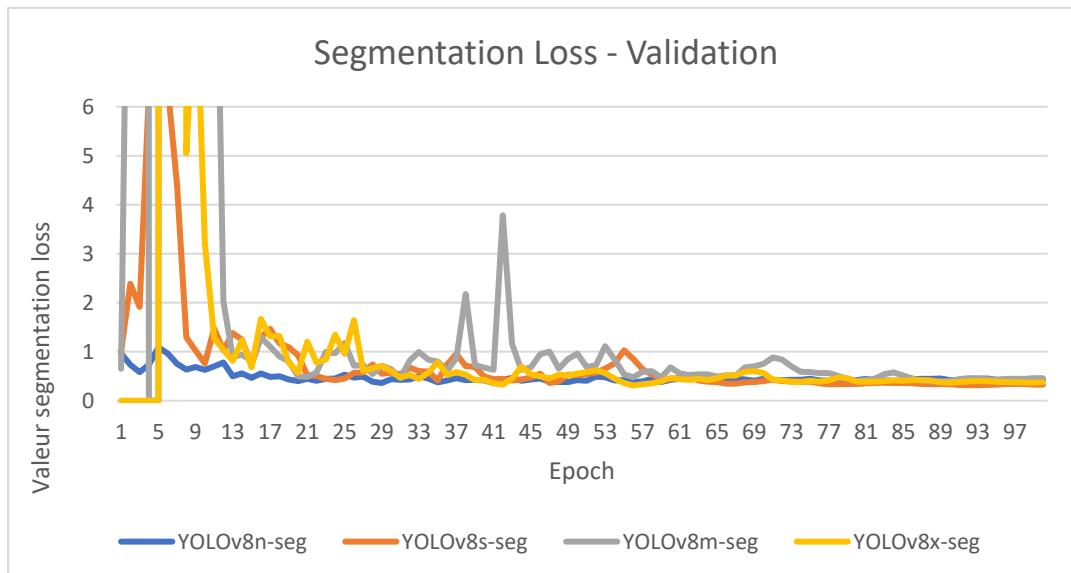


Figure 13. Résultats de la *Segmentation Loss* pour les données de validation à chaque *Epoch* pour tous les modèles de segmentation entraînés

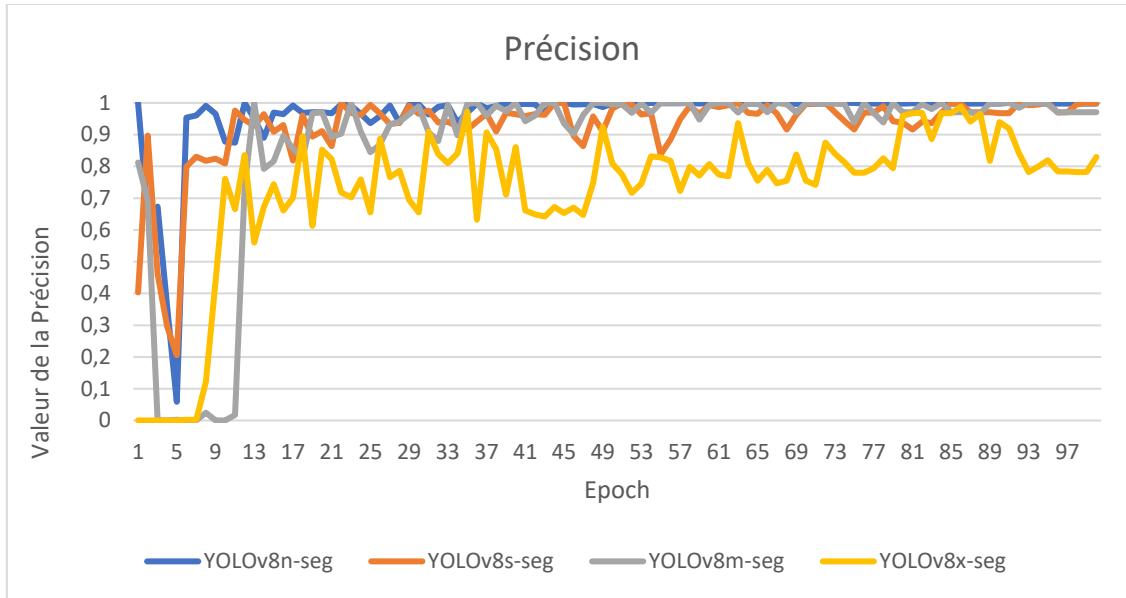


Figure 14. Résultats de la précision à chaque *Epoch* pour tous les modèles de segmentation entraînés

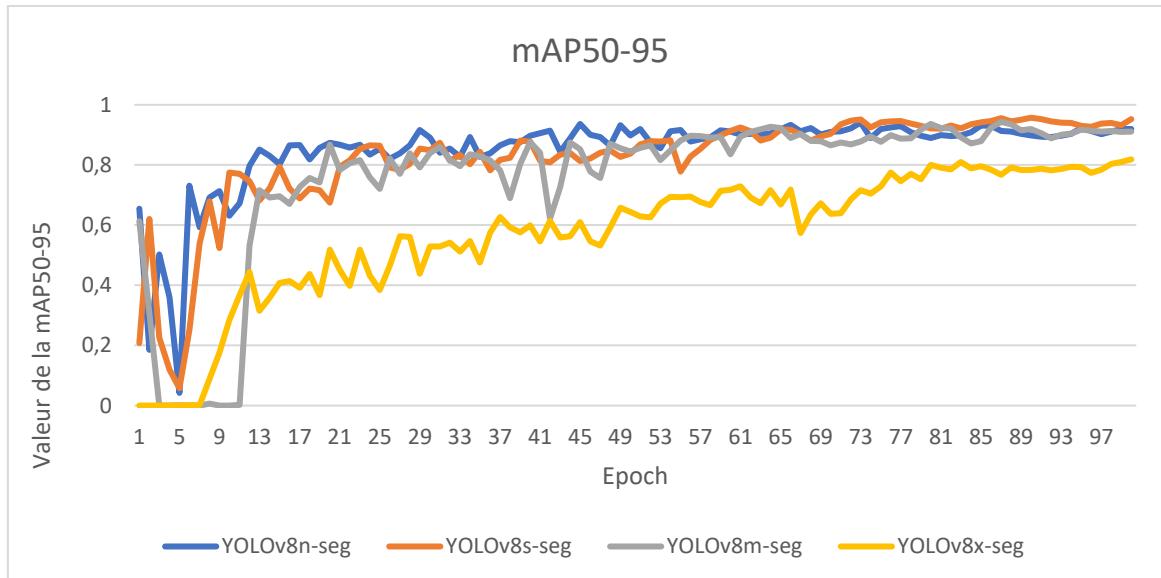


Figure 15. Résultats de la valeur *mAP50-95* à chaque *Epoch* pour tous les modèles de segmentation entraînés

Comme on peut le voir à la Figure 12 pour la *Segmentation Loss* lors de l'entraînement, la majorité des modèles commencent à se stabiliser autour de 80 *Epoch*. Nous pouvons toutefois remarquer une très légère descente dans ces courbes, ce qui pourrait indiquer que les modèles bénéficiaient possiblement de plus d'entraînement. Cependant, lorsqu'on analyse la

Segmentation Loss sur les données de validation (Figure 13), on observe qu'après seulement 60 *Epoch*, tous les systèmes semblent se stabiliser autour d'une valeur de *Loss* de 0.4. Sur ce graphique, on peut également voir que le modèle YOLOv8s semble obtenir la meilleure performance, avec une moyenne de perte autour de 0.3.

Si l'on analyse aussi le graphique de la précision de segmentation (Figure 14), on constate que les modèles n, s et m se stabilisent tous très près de 1 après 90 *Epoch*. À notre grande surprise, le modèle x affiche une précision considérablement plus faible que les autres. Il est important de noter que nous avons même poursuivi l'entraînement du modèle x pour 50 *Epoch* supplémentaires, mais sans amélioration notable des résultats. Nous pensons que cela pourrait être dû au fait que le modèle x, étant plus complexe et plus grand, nécessite davantage d'exemples pour bien apprendre à segmenter les objets. En effet, les modèles plus volumineux, comme YOLOv8x, ont une capacité d'apprentissage plus élevée, mais ont aussi besoin de jeux de données plus importants pour éviter le sous-apprentissage avec des données limitées. Finalement, lorsque nous analysons les résultats du *mAP50-95* à la Figure 15, on remarque que le modèle YOLOv8s semble légèrement plus performant que tous les autres. Encore une fois, nous avons été surpris par les résultats obtenus avec YOLOv8x lors de cette métrique.

Après avoir analysé l'ensemble des résultats, nous estimons que le modèle idéal dans ce contexte est le modèle YOLOv8s. Ce modèle est non seulement l'un des plus rapides à entraîner avec notre banque d'images, mais c'est également celui qui présente la meilleure valeur de *mAP50-95* ainsi qu'une des meilleures précisions, tournant très près de 1. La segmentation pour l'algorithme de détection des couleurs sera donc réalisée à l'aide de ce modèle pour ce projet. Cependant, il est important de noter que tous les modèles pourraient potentiellement atteindre de meilleures performances avec une banque d'images plus étendues et davantage d'*Epoch* d'entraînement. Nous aimerais donc ultérieurement entraîner un nouveau modèle YOLOv8x avec plus de 500 images annotées afin d'évaluer si la précision de la segmentation peut être améliorée comparativement aux résultats que nous avons obtenus au cours de ce projet.



Figure 16. Précision de la segmentation sur des images jamais vues par le système. 0.94 avec YOLOv8s (image de gauche) contre 0.89 avec YOLOv8x (image de droite)

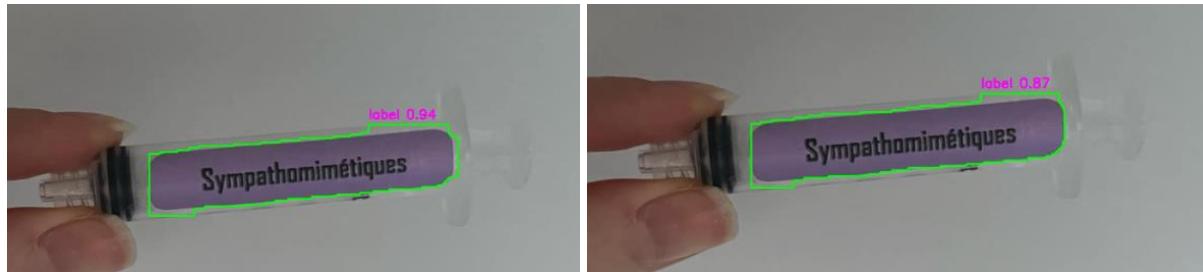


Figure 17. Précision de la segmentation sur des images jamais vues par le système. 0.94 avec YOLOv8s (image de gauche) contre 0.87 avec YOLOv8x (image de droite)

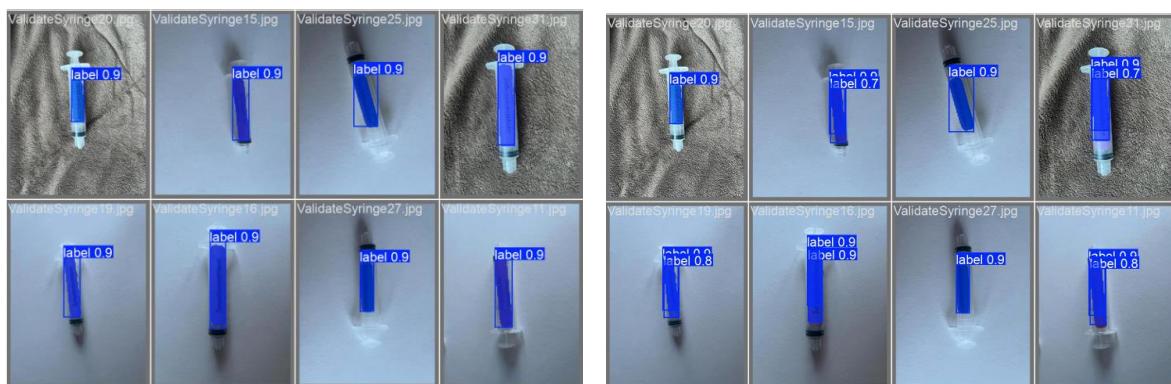


Figure 18. Précision de la segmentation sur les données de validation à la fin de l'entraînement des modèles YOLOv8s (gauche) et YOLOv8x (droite)

Pour ce qui est de la détection des couleurs dans l'étiquette, nous avons testé notre algorithme de détection de couleurs en utilisant la moyenne des pixels segmentée avec 100 images. Chaque image a été passée par l'algorithme et deux choses ont été évaluées. La première était de voir si la segmentation avait été bien faite autour de l'étiquette, puis la deuxième était de vérifier si la couleur détectée était la bonne. Cette évaluation a dû être faite manuellement par un de nos coéquipiers pour s'assurer que la couleur détectée était la bonne. Voici les résultats de cette évaluation :

	Segmentation (> 0.75)	Couleur
Positifs (%)	97	87
Négatifs (%)	3	13

Tableau 1. Résultats de la segmentation et couleur sur 100 images

D'après nos résultats, la détection de couleur fonctionne très bien avec la méthode que nous avons implémentée. Sur 100 images, on a réussi à détecter la bonne couleur sur 87 d'entre elles. Cependant, dans une version future, nous croyons qu'il pourrait être très pertinent de ne pas considérer les pixels noirs lors du calcul de la moyenne de la teinte. Comme l'écriture sur les étiquettes est noire, ces pixels sont actuellement pris en compte, ce qui pourrait affecter la couleur détectée. De plus, il est important de noter qu'un travail considérable a été consacré à bien définir les intervalles de classification pour chacune des couleurs (ajustement des intervalles H). Ces intervalles semblent bien fonctionner avec nos images actuelles, mais ils pourraient nécessiter des ajustements supplémentaires pour essayer d'avoir une meilleure précision sur les couleurs ayant été mal identifiées. Pour l'instant, nous pensons que ce système représente un prototype prometteur pour l'analyse de l'information des étiquettes de seringues.



Figure 19. Exemples de résultats pour la détection de couleur

Finalement, pour ce qui est de la détection du texte dans les images, la méthode que nous avons implémentée s'est montrée très intéressante sur des images de haute qualité, où l'étiquette est bien éclairée et bien centrée. Cependant, nous avons constaté que la reconnaissance du texte rencontrait d'énormes difficultés lorsque les images étaient floues, mal éclairées ou prises à l'aide de webcam d'ordinateur, qui sont de faible qualité. Dans ces cas, Tesseract produit des résultats partiels, incorrects, ou complètement vides. La détection du texte était donc très difficile avec la plupart de nos photos. Nous avons donc décidé de ne pas pousser l'analyse de l'OCR plus loin afin de pouvoir nous concentrer sur d'autres aspects de notre solution, comme la détection de couleurs ou la segmentation.

Conclusion

Ce projet avait pour objectif de concevoir un système capable de détecter automatiquement une seringue médicale, l'étiquette qui lui est collée, la couleur de celle-ci et le texte sur celle-ci, afin d'assister les professionnels de la santé dans le processus d'administration des médicaments aux patients.

Pour ce faire, nous avons mis en œuvre un pipeline complet combinant la détection et la segmentation avec YOLOv8, l'analyse de la couleur par transformation en espace HSB, et l'extraction du texte avec OCR. Après avoir annoté 233 images de seringues, nous avons entraîné plusieurs variantes du modèle YOLOv8 afin de comparer leurs performances. Les résultats ont permis de conclure que le modèle YOLOv8s offrait les meilleures performances, et ce, pour la détection de la seringue et la segmentation de son étiquette.

L'algorithme de détection de couleur basé sur la moyenne des teintes segmentées a montré une précision de 87% sur des images jamais vues, tandis que la segmentation atteignait un score de 97%. Les résultats sont très encourageants, mais il reste malgré tout certaines limites, notamment la sensibilité de l'OCR à la qualité des images. En effet, la reconnaissance de texte a été un peu plus efficace sur des images nettes capturées avec un bon éclairage et avec une bonne orientation, mais très inefficace avec les images de faible qualité. L'OCR reste donc un point à améliorer et explorer davantage.

En conclusion, notre projet présente une solution intéressante pour améliorer l'identification des médicaments. Des pistes d'amélioration incluent l'augmentation du *dataset* pour améliorer la segmentation, l'amélioration du filtrage des pixels lors de la détection de couleur, ainsi qu'un algorithme de l'OCR plus robuste face à des photos de faible qualité. À plus long terme, une intégration complète avec un système de caméras en hôpitaux pourrait permettre une automatisation plus poussée en temps réel et ainsi une meilleure sécurité pour les patients.

Bibliographie

Adrian Kaehler et Gary Bradski (2017), Learning OpenCV 3: Computer Vision in C++, Oreilly Media.

Raphael Gonzalez et Richard Wood (2008), Digital Image Processing, 3ième édition, Prentice Hall.

Richard Hartley et Andrew Zisserman (2004), Multiple View Geometry in Computer Vision, Cambridge University Press.

Richard Szeliski (2021), Computer Vision: Algorithms and Applications, Springer.

David Forsyth et Jean Ponce (2002), Computer Vision: A Modern Approach, Prentice Hall.

R.O. Duda, P.E. Hart, et D.G. Stork(2000), Pattern Classification (2nd Edition), Wiley-Interscience.

Color detection and segmentation for road and traffic signs. (s.d.). Repéré à <https://ieeexplore.ieee.org/document/1460692>

Kundu, R. (s.d.). YOLO Algorithm for Object Detection Explained [+Examples]. Repéré à <https://www.v7labs.com/blog/yolo-object-detection>

Ultralytics. (2023, 12 November). YOLOv8. Ultralytics YOLO Docs. Repéré à <https://docs.ultralytics.com/fr/models/yolov8/#supported-tasks-and-modes>

Agarwal, M., Alameda, J., Audenaert, J., Benoit, W., Beveridge, D., Bhattacharya, M., ... Yeh, S.-W. (2023, 13 June). Applications of Deep Learning to physics workflows. arXiv.org. Repéré à <https://arxiv.org/abs/2306.08106>

GeeksforGeeks. (2020, 13 February). Text Detection and Extraction using OpenCV and OCR. GeeksforGeeks. Repéré à <https://www.geeksforgeeks.org/text-detection-and-extraction-using-opencv-and-ocr/>

Portrait et évolution des postes vacants dans l'industrie des soins de santé et de l'assistance sociale au Québec entre 2019 et 2023. (2024, 5 September). Institut de la statistique du Québec. Repéré à <https://statistique.quebec.ca/fr/communique/postes-vacants-industrie-soins-sante-assistance-sociale-quebec-2019-2023>

IEDM (2023). "La pénurie de personnel infirmier au Québec: améliorer la flexibilité et les conditions de travail“, par Emmanuelle B. Faubert ". Repéré à <https://www.iedm.org/fr/la-penurie-de-personnel-infirmier-au-quebec-ameliorer-la-flexibilite-et-les-conditions-de-travail/>

AMCP (s.d.). "Médication Errors". Repéré à <https://www.amcp.org/concepts-managed-care-pharmacy/medication-errors>

World Health Organization (s.d.). "Medication Without Harm". Repéré à
<https://www.who.int/initiatives/medication-without-harm>

Rayhan A. Tariq, Rishik Vashisht, Ankur Sinha, Yevgeniya Scherbak, "Medication Dispensing Errors and Prevention", StatPearls, 2024.

World Health Organization (s.d.). "Medication safety for look-alike, sound-alike medicines". Repéré à <https://iris.who.int/bitstream/handle/10665/373495/9789240058897-eng.pdf?sequence=1>

GeeksforGeeks. (2020, 13 February). Text Detection and Extraction using OpenCV and OCR. GeeksforGeeks. Repéré à <https://www.geeksforgeeks.org/text-detection-and-extraction-using-opencv-and-ocr/>

Ultralytics. (2023, 12 November). YOLOv8. Ultralytics YOLO Docs. Repéré à
<https://docs.ultralytics.com/fr/models/yolov8/>

Kundu, R. (s.d.). YOLO Algorithm for Object Detection Explained [+Examples]. Repéré à
<https://www.v7labs.com/blog/yolo-object-detection>