Détection et classification automatique des documents pour l'application KYC

Culture Sciences de l'Ingénieur

William KETCHANTANG – Dimitrios TSOLAKIDIS - Kevin MEETOOA

Édité le 22/07/2022

école normale supérieure paris—saclay

Cette ressource est issue d'une publication du numéro 109 de La Revue 3EI de juillet 2022 et fait partie du « Dossier Intelligence Artificielle » [7] sur Culture Sciences de l'Ingénieur. William Ketchantang est Team leader machine learning and computer vision, Dimitrios Tsolakidis et Kevin Meetooa sont machine learning and computer vision Engineer chez Vialink.

Cette ressource présente la détection et la classification automatique de documents scannés ou pris en photos pour des fins de digitalisation de l'entrée en relation clients (KYC : Know Your Customer). Nous présentons un modèle de Deep Learning de détection et de classification automatique des documents capable de détecter des documents sur une image avec une efficacité de 90 % et un temps moyen de l'ordre de la seconde sur des machines équipées de cpu i7 standard.

1 – Introduction

1.1 - Application KYC

Notre société se digitalise de plus en plus, et à ce titre, nous souhaitons aller plus loin en permettant l'ouverture des comptes bancaires, la souscription de prêts de consommation, immobiliers 100 % en ligne par exemple. Pour cela, nous avons besoin d'identifier le souscripteur et de contrôler automatiquement ses documents, afin de s'assurer que tout le processus ne souffre d'aucune irrégularité. Ce processus s'appelle couramment KYC (« *Know Your Customer* »), consistant à digitaliser l'entrée en relation. Il peut être schématisé comme l'illustre la Figure 1. Ce processus 100 % numérique permet un gain d'efficacité chez les opérationnels, améliore la qualité de vie des particuliers, et permet la conformité de la réglementation bancaire sur la lutte contre les fraudes et le blanchiment d'argent.

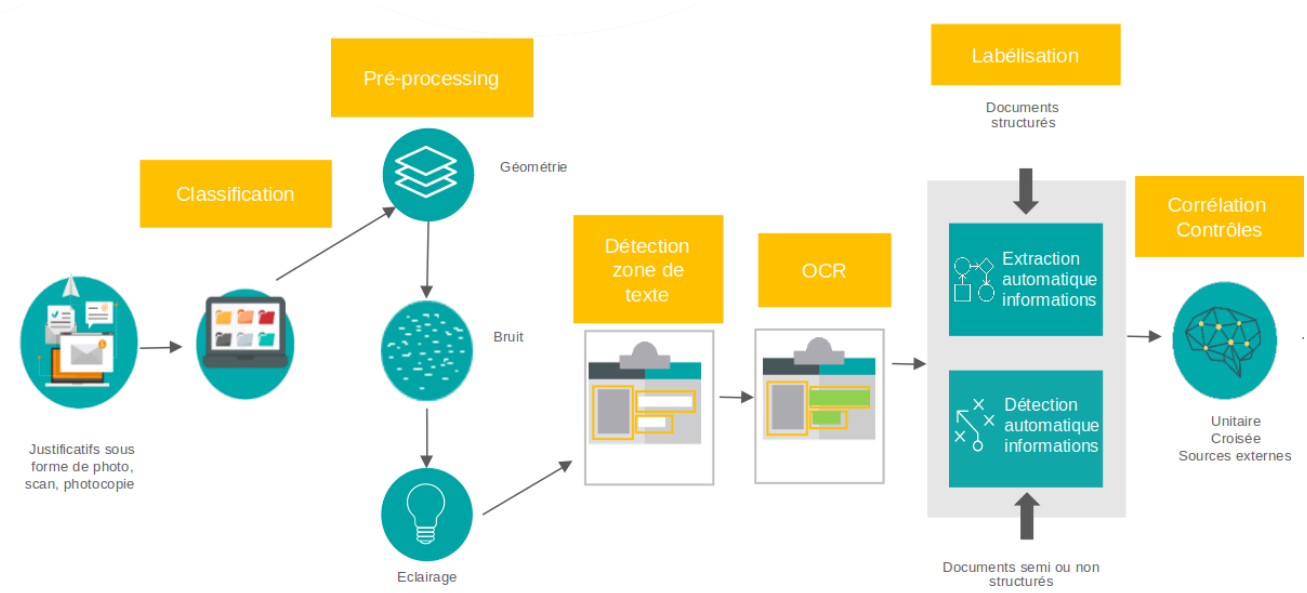


Figure 1 : Aperçu d'un processus KYC

Le processus KYC est un processus complexe. En début de processus, il est important de détecter et classifier automatiquement les documents transmis par l'utilisateur. Ces documents sont variables selon les contrôles souhaités par les clients, à l'instar des pièces d'identité, justificatifs de domicile, avis d'imposition, bulletins de salaire, Kbis, etc. Dans cette ressource, nous décrirons le module de détection/classification automatique des documents, en charge de localiser précisément tout type de documents supportés, y compris leurs faces avant et arrière, indépendamment des conditions d'acquisition (camera, scanner, luminosité ambiante non contrôlée, déformations géométriques des objets dans l'image, etc.). Comme dans l'état de l'art, nous allons utiliser le mot Détection au lieu de Détection et Classification.

1.2 - Détection des documents

La détection d'objets est l'une des tâches les plus importantes et les plus difficiles de la vision par ordinateur, utile pour la vidéosurveillance, la conduite autonome, l'identité numérique, etc. L'objectif est de localiser tous les objets et leurs catégories respectivement avec des scores de confiance. Cette tâche est bien connue et bien traitée dans l'état de l'art. Nous identifions deux groupes d'approches : d'une part, les premières approches sont basées sur des **méthodes traditionnelles** qui nécessitent beaucoup d'efforts de « *Feature Engineering* » ; d'autre part, les nouvelles approches dites modernes se concentrent sur les **méthodes d'apprentissage profond**.

Les méthodes traditionnelles peuvent être découpées en deux groupes. Tout d'abord, certaines approches sont basées sur l'extraction de points clés des objets que nous voulons détecter. L'auteur [4] utilise SURF (« *Speed Up Robust Features* », [2]) et BoW (« *Bag of Words* ») pour extraire les caractéristiques des points clés des objets. Ces caractéristiques sont ensuite utilisées pour entraîner un classifieur SVM (Machine à vecteurs de support) afin de détecter les catégories d'objets. Les auteurs obtiennent de bons résultats (environ 70 % d'efficacité sur des bases d'images publiques Caltech [5]). Cependant, extraire les caractéristiques des nouvelles catégories d'objets nécessite de revisiter la méthode d'extraction, d'où un temps de développement plus important.

De plus, certains auteurs utilisent d'autres approches d'extraction de caractéristiques n'utilisant pas les points clés, comme le LBP (« *Local Binary Pattern* », [20]) utilisé par [21] pour extraire les caractéristiques des visages, à des fins de détection. L'histogramme des gradients est également utilisé pour les caractéristiques des objets, comme décrit par [17], à des fins de détection.

Toutes les approches basées sur l'extraction de caractéristiques de type « *Feature Engineering* » nécessitent un effort important de la part des scientifiques pour construire les algorithmes ad-hoc permettant d'extraire les caractéristiques pertinentes des objets à détecter. Si nous voulons détecter une nouvelle catégorie d'objets, nous devons revisiter l'algorithme d'extraction de caractéristiques pertinentes, et y passer beaucoup de temps, sans garantie de résultats à la hauteur des attentes. Il s'agit d'une limite importante des approches dites de « *Feature Engineering* ». C'est pourquoi, de plus en plus de scientifiques utilisent maintenant des approches d'apprentissage profond pour espérer obtenir des caractéristiques pertinentes en peu de temps et avoir une très bonne généralisation sur les nouveaux types d'objets.

Plusieurs auteurs [8] utilisent des approches d'apprentissage profond pour détecter des objets. Deux types de détecteurs sont détaillés : le premier détecteur est basé sur les réseaux de proposition de régions sous forme de boîte rectangulaire, avec la catégorie associée (R-CNN « *Region Convolutional Neural Network* »). La précision de tels réseaux est de l'ordre de 75 % sur la base d'images publique Pascal VOC 2007 [22].

Le second détecteur proche du précédent mais plus léger (YOLO « *You Only Look Once* ») est également basé sur une proposition de régions sous forme de boîte englobante et la catégorie

associée de l'objet [12]. Le temps de traitement est plus faible que celui du premier détecteur, de plus ce modèle tourne facilement sur les plateformes embarquées de type téléphone mobile. Cependant, la précision de la localisation est inférieure (63%) à celle de la première approche sur la base d'images publiques Pascal VOC 2007.

Avec les approches d'apprentissage profond, il est très important d'avoir une base d'images représentative des objets et de laisser le modèle apprendre les caractéristiques pertinentes des objets, tout en étant capable de généraliser la caractérisation des objets qui ne sont pas présents dans notre jeu de données d'apprentissage. Et cela, sans aucun effort supplémentaire de la part des scientifiques. C'est l'avantage d'utiliser des approches d'apprentissage profond pour la détection d'objets. Les auteurs [11] résument également les avantages des approches d'apprentissage profond par rapport aux approches traditionnelles. Dans notre cas, où nous avons plusieurs documents de différents types, structurés et non structurés, les approches d'apprentissage profond sont préférées.

Cette ressource est structurée comme suit : dans la section suivante, nous décrivons dans un premier temps l'architecture du modèle que nous utilisons. Nous présentons dans un second temps les expériences et les résultats. Enfin, nous donnons la conclusion et les perspectives.

2 – Modèle de Détection des documents

2.1 - Architecture

Nous utilisons une architecture U-Net, introduite pour la première fois par [13]. Le modèle de segmentation U-Net a montré de meilleures performances que les réseaux purement convolutifs. Il s'agit d'un réseau en forme de U composé d'un encodeur et d'un décodeur, comme l'illustre la Figure 2.

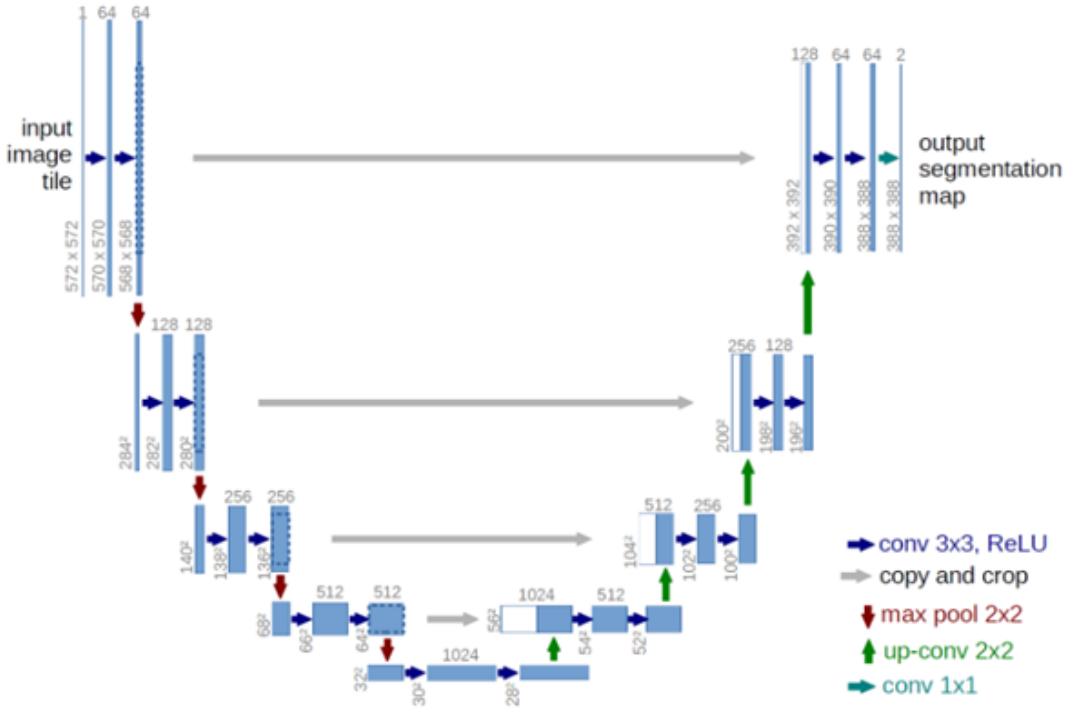


Figure 2 : Architecture U-Net.

Chaque boîte bleue correspond à une carte de caractéristiques multicanaux. Le nombre de canaux est indiqué en haut de la boîte et la taille x-y de la boîte est indiquée sur le bord inférieur gauche de celle-ci. Les boîtes blanches représentent des cartes de caractéristiques copiées issues de la partie encodeur. Les flèches indiquent les différentes opérations [13].

Intuitivement, l'encodeur apprend à extraire les éléments caractéristiques discriminants des objets. L'encodeur, quant à lui, réduit les dimensions spatiales en sortie de chaque couche tout en augmentant le nombre de filtres. Le tenseur obtenu en sortie de l'encodeur est généralement appelé goulot d'étranglement ou espace latent. On parle également de vecteur de caractéristiques.

Le décodeur apprend à localiser le document. Le point d'entrée du décodeur est le vecteur de caractéristiques de l'encodeur. Le décodeur augmente les dimensions spatiales en sortie de chaque couche tout en réduisant le nombre de filtres, jusqu'à obtenir la résolution spatiale de l'image d'entrée.

La « *skip-connection* » permet d'ajouter la sortie de l'encodeur à une profondeur donnée du réseau à l'entrée du décodeur à la même profondeur. Ainsi, le sur-échantillonnage dans la branche du décodeur garde les détails de l'encodeur. Par conséquent, la sortie du décodeur à chaque profondeur du réseau est de meilleure qualité.

En sortie du modèle, les dimensions spatiales du masque prédit sont les mêmes que les dimensions de l'image d'entrée, ce qui permet de faire une prédiction de la catégorie de chaque pixel de l'image d'entrée.

Nous utilisons un modèle de type EfficientNet-B3 dérivé de la famille EfficientNet [18] comme encodeur de la branche U-Net.

2.2 - Fonction de coût

Nous utilisons la fonction de coût « *Dice* » [16] et la fonction de coût dite « *focale* » [10] pour l'entraînement du modèle. L'idée principale est d'assigner à chaque pixel une classe d'appartenance. Ainsi, nous avons un masque prédit pour chaque classe de documents. La fonction de coût *Dice* mesure donc le chevauchement relatif entre un masque de classe de document prédit et sa vérité terrain (i.e le masque labellisé pour ce document) comme l'illustre la Figure 3 où sont présentés les masques prédits et de vérité de terrain pour un document. Le masque de vérité de terrain est construit à partir d'un rectangle englobant chaque face du document dessiné manuellement par nos soins grâce à un outil d'annotation dédié [24].

La formule de la fonction de coût de *Dice* est présentée ci-dessous :

$$D_{loss} = 2 \frac{\sum_{1}^{N} p_i g_i}{\sum_{1}^{N} p_i^2 + \sum_{1}^{N} g_i^2}$$

où p_i est la valeur d'un pixel prédit pour une classe de document donnée (0 ou 1) et g_i est la valeur du même pixel issu de la vérité de terrain et N le nombre total d'images dans la base de documents.

La fonction de coût *Dice* n'est pas adaptée au déséquilibre de classes. Pour cette raison, nous utilisons une fonction de coût dite « focale » en plus de la fonction de coût *Dice*. La fonction de coût focale est exprimée par la formule suivante :

$$F_{loss}(p_t) = -\alpha_t (1 - p_t)^{\gamma} log(p_t)$$

où p_t correspond à la probabilité d'appartenance d'un pixel à la bonne classe. γ contrôle la forme de la courbe. Plus la valeur de γ est élevée, plus les pixels mal classés sont mis en évidence dans

le calcul de la fonction de coût. Ainsi, cette fonction de coût est plus adaptée lorsque les classes sont déséquilibrées.

Nous obtenons la fonction de coût finale en additionnant les deux fonctions de coût :

$$Total_{loss} = D_{loss} + F_{loss}$$

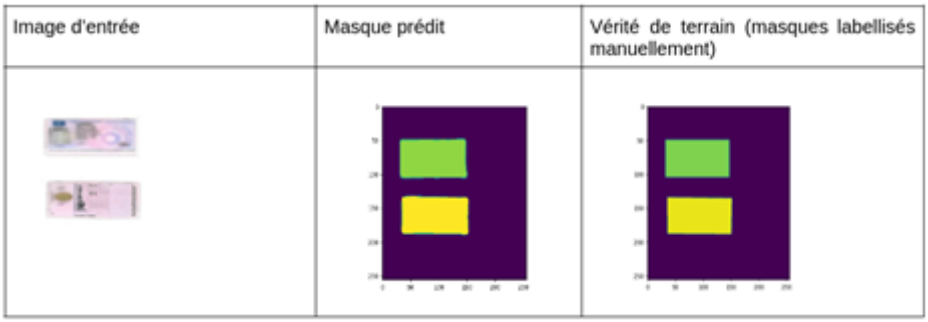


Figure 3 : Illustration d'un document, de son masque prédit et de sa vérité de terrain. Vert : Face recto du permis de conduire français. Jaune : Face verso du permis de conduire français.

2.3 - Entraînement

Les images d'entrée et leurs masques de vérité de terrain correspondantes sont utilisés pour entraîner le réseau. Les images d'entrée et les masques sont redimensionnés en taille 256 x 256 pixels. Notre base interne de documents se compose de plusieurs classes, avec 12370 images de documents au total. Dans cette base de documents, nous avons également des documents augmentés artificiellement (rotation, changement de luminosité, distorsion, etc.).

Le réseau a été entraîné :

- Pendant 40 époques ;
- Avec une taille de batch de 4 ;
- En utilisant l'optimiseur Adam [9] ;
- Avec un taux d'apprentissage de 0,001 ;
- Avec des poids initiaux issus du pré-entraînement du modèle sur la base d'images publiques ImageNet [15].

Pour la fonction de coût focale, nous utilisons les meilleures valeurs mentionnées par les auteurs [10] avec $\alpha = 0.25$et $\gamma = 2$.

Les résultats de l'entraînement sont illustrés par les fonctions de coût de la figure 4.

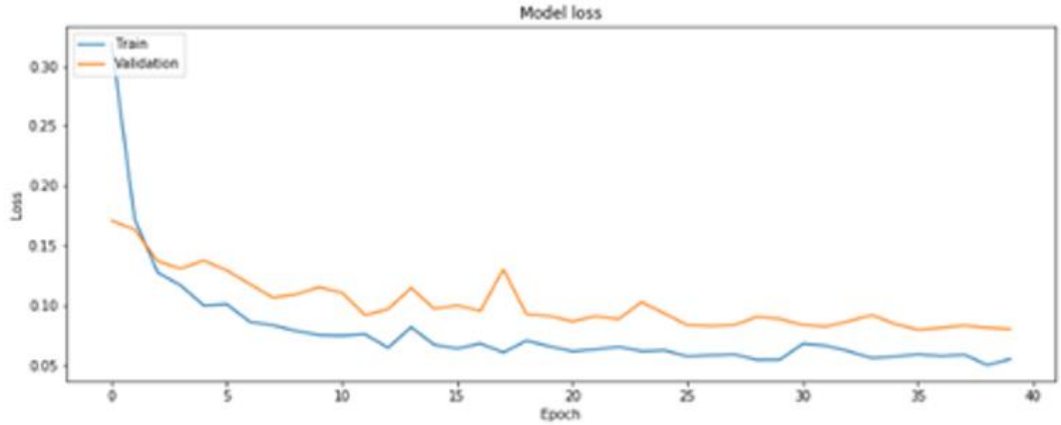


Figure 4 : Évolution de la fonction de coût D_loss durant l'entraînement

Nous observons que la fonction de coût converge aussi bien sur la base de documents d'entraînement que sur la base de documents de validation. Ainsi, notre modèle est bien entraîné.

3 – Résultats expérimentaux

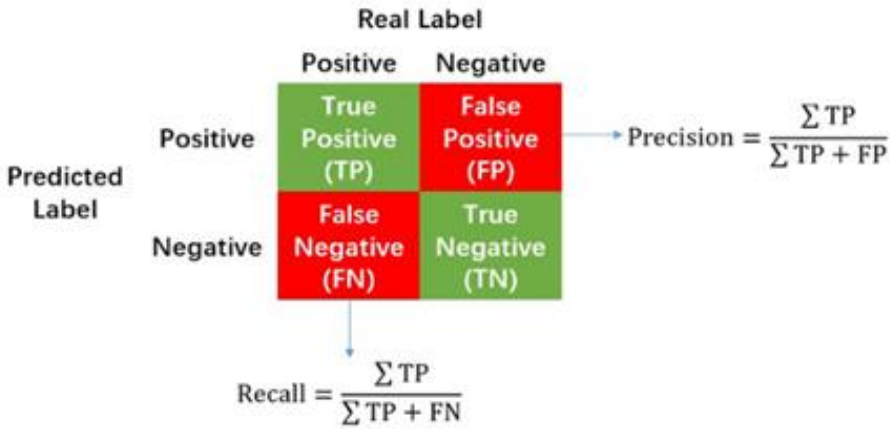
Nous utilisons trois métriques pour évaluer l'efficacité de notre modèle : Précision, Rappel et IoU.

Pour un document donné et deux classes binaires, nous définissons :

- **TP** comme le nombre de vrais positifs : c'est le nombre de pixels de la classe positive qui ont été correctement prédits ;
- **FP** comme le nombre de faux positifs : c'est le nombre de pixels de la classe négative qui ont été prédits à tort comme positifs ;
- **FN** est le nombre de faux négatifs : il s'agit du nombre de pixels de la classe positive qui ont été prédits à tort comme négatifs.

Par conséquent,

- La **précision** quantifie le nombre de prédictions correctes de pixels pour la classe positive ;
- Le **rappel** quantifie la proportion de prédictions de pixels positifs par rapport au nombre total de pixels positifs dans le document.



Cette définition peut ensuite être étendue à plus de deux classes. Pour un document donné et deux classes binaires, nous définissons IoU comme suit :

$$IOU = \frac{P \cap GT}{P \cup GT}$$

Où

- P correspond au masque prédit pour la classe positive ;
- GT correspond au masque de vérité pour la classe positive.

Au numérateur, nous calculons la zone de chevauchement entre le masque prédit et le masque de vérité.

Le dénominateur est l'union entre le masque prédit et le masque de vérité.

En divisant la zone de chevauchement par la zone d'union, nous obtenons le score final d'Intersection sur Union (IoU).

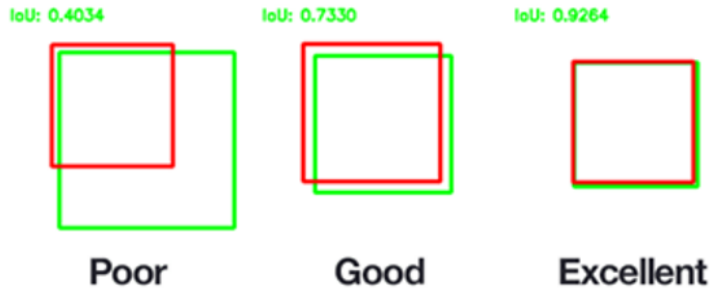


Figure 5 : Vert : vérité de terrain du masque. Rouge : masque prédit.

Cette définition peut également être étendue à plus de deux classes.

Les mesures de précision, de rappel, et de IoU score obtenues sur notre base de tests sont présentées dans le tableau 1.

Précision	Rappel	IoU
92.71%	94.14%	**88.23%**

Tableau 1 : Efficacité du modèle sur notre base de test de documents privés.

Avec 92 % de précision, 94 % de rappel, et 88 % de IoU score, nous considérons que le modèle utilisé donne des résultats satisfaisants.

De plus, notre modèle est capable d'atteindre un temps d'inférence de 1,25 secondes en moyenne sur une machine équipée d'un CPU i7, ce qui est satisfaisant pour nos besoins.

Nous présentons quelques résultats qualitatifs obtenus sur quelques documents de notre jeu de données de test, à la figure 6.

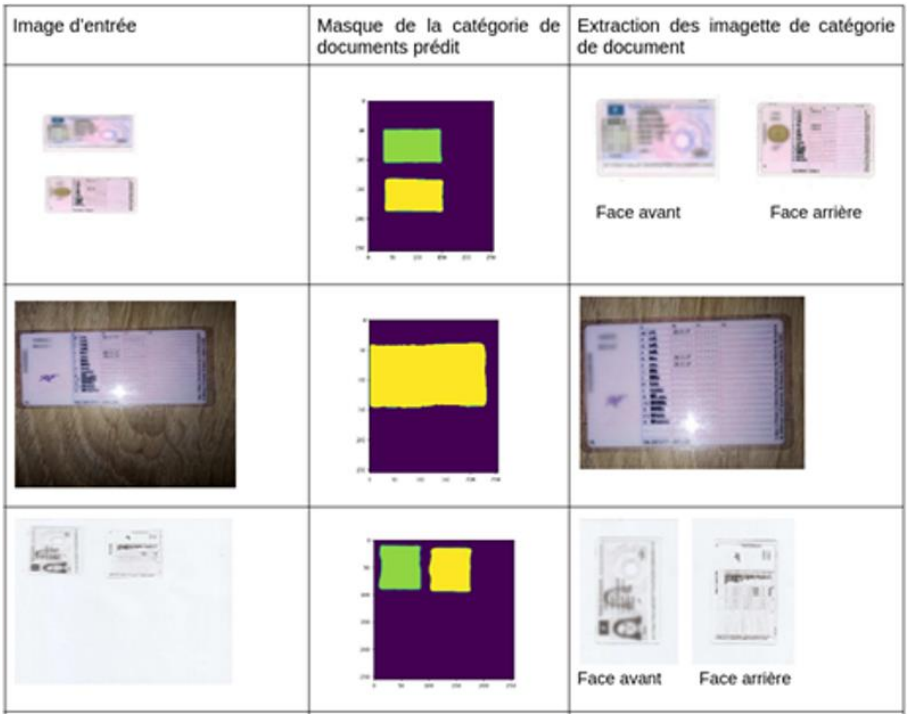


Figure 6 : Illustration de la détection et classification des documents. Vert : Face recto du permis de conduire français. Jaune : Face verso du permis de conduire français.

Comme nous pouvons le voir ci-dessus, le modèle est robuste et permet de localiser avec précision les documents, quelle que soit leur qualité. Cela s'explique par le fait que nous avons entraîné le

modèle sur un grand ensemble de documents comprenant des documents de bonne et de mauvaise qualité.

4 – Conclusion

Nous avons utilisé un modèle de détection de documents dans une image basée sur l'architecture U-Net. L'efficacité (IoU score) du modèle sur notre base de 12370 documents internes de diverses catégories (Cartes d'identités, passeports de différents pays, etc.) est de 90 % environ. Le temps d'inférence sur une infrastructure classique équipée de CPUs i7 standards, est de l'ordre de la seconde pour 1 image.

Pour la suite, nous augmenterons la taille de notre base avec plus de catégories de documents. Ainsi, notre modèle supportera encore plus de catégories en production, et généralisera encore mieux. De plus, nous envisageons changer le l'encodeur « *backbone* » par un encodeur de type ResNet par exemple, plus efficace mais plus lourd.

Références :

[1]: Agarap, Abien Fred. *Deep Learning using Rectified Linear Units (ReLU)*. 2018, https://arxiv.org/pdf/1803.08375.pdf

[2]: Bay, Herbert, et al. "SURF: Speed Up Robust Features." *ECCV*, 2006.

[3]: Chollet, Francois. "Xception: Deep Learning with Depthwise Separable Convolutions}." 2017, https://arxiv.org/pdf/1610.02357.pdf

[4]: Farooq, Javeria. "Object Detection and Identification using SURF and BoW Model." *ResearchGate*, April 2016.

[5]: Fei-Fei, L., et al. "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories." *IEEE. CVPR*, 2004.

[6]: G. Lowe, David. "Distinctive Image Features from Scale-Invariant Keypoints." *IJCV*, 2004.

[7]: Dossier Intelligence Artificielle, juin 2022, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/dossier-intelligence-artificielle

[8]: Jiao, Licheng, et al. "A Survey of Deep Learning-based Object Detection." *IEEE*, 11 July 2019.

[9]: Kingma, Diederik, and Jimmy Ba. "Adam: A Method for Stochastic Optimization." *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[10]: Lin, Tsung-Yi, et al. "Focal Loss for Dense Object Detection." *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999-3007.

[11]: Mahomy, Niall O. "Deep Learning vs. Traditional Computer Vision." *Advances in Computer Vision Proceedings of the 2019 Computer Vision Conference (CVC)*, 30 october 2019.

[12]: Redmon, Joseph, et al. "You Only Look Once: Unified, Real-Time Object Detection." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13]: Ronneberger, Olaf, et al. "U-Net: Convolutional Networks for Biomedical Image Segmentation." *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015, pp. 234–241.

[14]: Rublee, Ethan, and Gary Bradski. "ORB: an efficient alternative to SIFT or SURF." *IEEE International Conference on Computer Vision. IEEE International Conference on Computer Vision*, 2011.

[15]: Russakovsky, Olga, et al. “ImageNet Large Scale Visual Recognition Challenge.” *Int. J. Comput. Vis.*, vol. 115, 2015, pp. 211-252.

[16] Sudre, Carole, et al. “Generalized Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations.” *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, 2017.

[17]: Sultana, Marjia, et al. “Object Detection using Template and HOG Feature Matching.” *International Journal of Advanced Computer Science and Applications*, July 2020

[18]: Tan, Mingxing, and Quoc Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, 2019.

[19]: Zhang, Zhilu, and Mert Sabuncu. “Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels.” *NeurIPS (2018)*, 2018.

[20]: Lizé, Jade, et al. “Local binary pattern and its variants: application to face analysis.” *HAL Id: hal-02924534*, 28 August 2020.

[21]: Ahonen, T., Hadid, A., Pietikäinen, M. (2004). Face Recognition with Local Binary Patterns. In: Pajdla, T., Matas, J. (eds) Computer Vision - ECCV 2004. ECCV 2004. Lecture Notes in Computer Science, vol 3021. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-24670-1_36

[22]: Mark Everingham and Luc Van Gool and C. K. I. Williams and J. Winn and Andrew Zisserman. “The PASCAL Visual Object Classes (VOC) challenge.”

[23]: https://paperswithcode.com/sota/object-detection-on-pascal-voc-2007

[24]: Dutta, A. and Gupta, A. and Zissermann, A. “VGG Image Annotator (VIA)”