

Classification et extraction d'information d'un document

1. Introduction

- Présentation du but du document.
- Présentation du projet à nouveau
- Description des classes à classifier

2. Exploration & Visualisation

- Visualisation des jeux de données.
- Analyse des facteurs potentiels d'erreur
- Sélection de correctifs à réaliser lors du preprocessing

3. Méthodologie

- Description de l'approche choisie (axe CNN et axe NLP).
- Explication des intérêts/nécessité de chaque axe.

4. Entraînement du CNN

- CNN testés et résultats obtenus.
- CNN retenu et tweaking.
- Présentation des résultats obtenus.
- Interprétabilité

5. Choix de la librairie OCR

- Présentation de la librairie OCR choisie et justification.
- Présentation des résultats de l'extraction.

6. Entraînement du modèle de NLP

- Modèles testés et résultats obtenus.
- Modèle retenu et tuning des hyperparamètres.
- Présentation des résultats obtenus.
- Interprétabilité

7. Mise en place du modèle de voting

- Explication de l'intérêt.
- Description du modèle de vote retenu.
- Présentation des résultats du modèle (rappel des résultats des modèles seuls).

8. Conclusion

- Résumé des résultats obtenus.
- Comparaison state of the art.
- Axe d'amélioration.

1. Introduction

La conservation et la documentation de tous nos échanges dans le milieu professionnel induit une accumulation d'informations qu'il est difficilement classifiable et repérable. L'enjeu est d'autant plus crucial lorsque les entreprises tentent de numériser leurs archives afin de réduire le stockage physique tout en classifiant en accord avec les nomenclatures modernes de l'entreprise.

L'objectif de ce projet est de développer un ensemble de solutions pour améliorer la gestion et le traitement des documents dans le domaine des assurances.

- 1. Développer un modèle de classification des documents :** Identifier la nature des documents (contrats, justificatifs, offres, etc.) en utilisant des techniques de classification.
- 2. Améliorer l'efficacité du traitement des documents :** Automatiser les tâches manuelles dans le domaine des assurances pour améliorer l'efficacité du traitement des documents.
- 3. Réduire les coûts opérationnels :** Diminuer les coûts associés à la gestion des documents papier par l'automatisation des processus.

Les membres de l'équipe :

- **Quentin Pizenberg**
- **Julien Thielleux**
- **Sofiane Louiba**
- **Chaima Haddoudi**

Le projet vise à classifier près de 400.000 documents scannés d'une base de données d'entreprises du tabac en 16 classes différentes.

Les types de document à classifier sont:

Classification	Type de document
0	letter
1	form
2	email
3	handwritten
4	advertisement
5	scientific report
6	scientific publication
7	specification
8	file folder
9	news article
10	budget
11	invoice
12	presentation
13	questionnaire
14	resume
15	memo

Une approche en deux axes principaux a été retenue:

- L'axe 1 est basé sur la classification visuelle des documents à l'aide d'un réseau de neurones convolutif (CNN).
- L'axe 2 est basé sur la classification du texte extrait des documents à l'aide d'un modèle de traitement du langage naturel (NLP) afin de palier la faible différentiation de certains documents ayant des structures similaires.

Ensuite, nous constituons un modèle de vote pour améliorer les performances des modèles individuels.

Ce document a pour object l'exploration des données à classifier, la préparation des données, la présentation des process et modèles de machines learning sélectionnés ainsi que les résultats obtenus ou les complications observées.

2. Compréhension et manipulation des données

2.1. Cadre

La première étape de la modélisation du problème a été de formaliser et comprendre la structure de données provenant de *The Legacy Tobacco Document Library (LTDL), University of California, San Francisco, 2007*.

Le dataset [RVL-CDIP](#) contient :

- 400.000 images au format .tif
- 400.000 fichiers text avec le même nom que le fichier image, contenant les 16 catégories

Le dataset [text-extraction-for-ocr de Kaggle](#) contient :

- 520 images au format .tif
- 520 fichiers xml gt correspondant aux zones de detection de l'OCR
- 520 fichiers xml ocr correspondant au contenu détecté par l'OCR de référence. Cela sera notre cible pour sélectionner un OCR

Dans la suite de la visualisation, on a pris un ensemble de 1000 fichiers du dataset RVL-CDIP sélectionné aléatoirement avec une équirépartition entre chacune des catégories.

Sources :

- A. W. Harley, A. Ufkes, K. G. Derpanis, "Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval," in ICDAR, 2015
- The Legacy Tobacco Document Library (LTDL), University of California, San Francisco, 2007.

2.2. Cadre Pertinence

Le projet repose sur un jeu de données peu traditionnel, composé d'images de documents plutôt que des ensembles de données textuelles ou numériques couramment analysés en machine learning.

Les variables les plus pertinentes pour ce projet sont les images des documents, les annotations de classification et les champs textuels à extraire. La variable cible est la classe du document pour la classification et le texte extrait pour l'OCR.

Le jeu de données se distingue par la diversité des types de documents et la qualité variable des scans, incluant des manuscrits et des impressions.

Toutefois, des limitations existent, notamment en raison de la qualité des scans et des variations d'écriture manuscrite, qui peuvent poser des défis supplémentaires.

2.3. Pre-processing et feature engineering

Le preprocessing est essentiel pour préparer les images de documents.

En améliorant la qualité des images, en uniformisant les formats et en extrayant des caractéristiques pertinentes, nous optimisons les performances des modèles de machine learning que nous allons mettre en place. Ce pré-traitement permet d'uniformiser les données, de réduire le bruit et de faciliter l'extraction d'informations, conduisant ainsi à une classification plus précise et à une extraction de texte plus fiable.

Pour garantir une reconnaissance optimale, il est crucial que les caractères soient bien distincts du fond, avec des bordures nettes et un contraste élevé. De plus, un bon alignement des caractères et une résolution adéquate sont indispensables. Enfin, réduire le bruit des images améliore encore la qualité du traitement et des résultats obtenus.

Librairies utilisées :

- OpenCV as cv2
- ImageOps from PIL

Format des Images

- **Redimensionnement** (`img_resized = cv2.resize(img, (800, 1000))`) : Uniformisation des tailles des images pour standardiser l'entrée des modèles. Les images sont redimensionnées en 800x1000 pixels.
- **Rotation / Inclinaison** (`cv2.getRotationMatrix2D(center, angle, 1.0) + cv2.warpAffine(image, M, (w, h), flags=cv2.INTER_CUBIC, borderMode=cv2.BORDER_REPLICATE)`) : Correction de l'orientation des images pour aligner les textes horizontalement. Cette opération utilise une matrice de rotation pour redresser les images inclinées, améliorant ainsi la lisibilité et la précision de l'extraction de texte.

Texte des images

Pas besoin de modifier la taille du texte car il est assez homogène et la différence de taille rendra le modèle plus robuste aux futures documents à classifier.

- **Thinning / Skeletonization** (`skeleton = cv2.ximgproc.thinning(img, thinningType=cv2.ximgproc.THINNING_ZHANGSUEN)`) : Réduction de l'épaisseur des traits pour extraire une version simplifiée des éléments visuels, souvent utilisée pour les textes manuscrits. Cette technique permet d'obtenir une représentation minimale tout en conservant les caractéristiques essentielles.

Couleurs

Pas besoin de faire de la binarization / grayscaling car les images sont déjà en noir et blanc avec un contraste et une luminosité correcte. On risquerait de perdre en lisibilité si l'on change notamment le contraste pour les petites lettres.

- **Noise Removal / Reduction** (`img_denoised = cv2.fastNIMeansDenoisingColored(img, None, 10, 10, 7, 21)`) : Suppression des bruits pour obtenir une image plus propre en utilisant des algorithmes de débruitage. Cela est crucial pour éliminer les petits artefacts visuels et améliorer la lisibilité.
- **Black/White inversion** (`PIL.Imageops.invert(img)`) : On inverse le noir et le blanc sur les images majoritairement noires pour avoir une proportion de pixels noirs inférieur à celle des pixels blancs et uniformiser le dataset

2.4. Data Visualisation

Nous allons étudié en parallèle les éléments des deux datasets.

2.4.1. Dataset Kaggle

Préparons une liste principale avec les images ainsi que des tableau numpy d'informations qui seront à analyser : les tailles, résolutions, distribution de couleur et luminosité.

Enfin, regardons le premier élément pour avoir une idée plus précise des types d'éléments considérés.

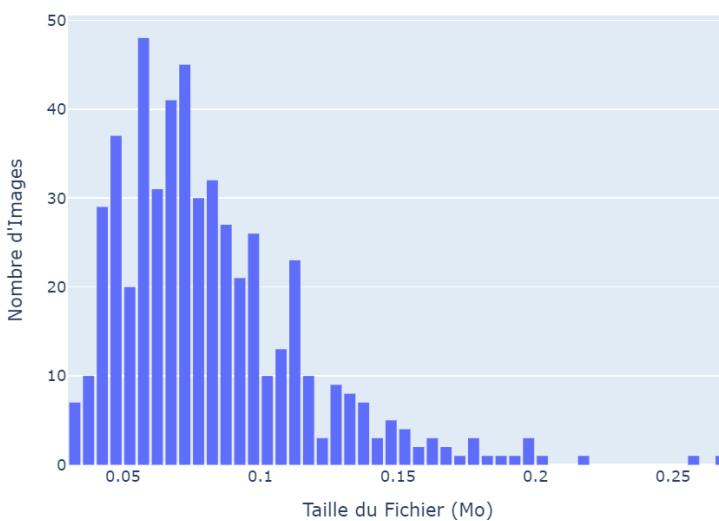
I. Analyse des formats

I.A. Distribution des Tailles d'Images

L'analyse de la distribution des tailles de fichiers permet de comprendre la variation des tailles d'images et de décider si une normalisation de la taille est nécessaire.

En fonction des résultats, nous pouvons choisir de redimensionner les images pour une uniformité.

Distribution des Tailles d'Images

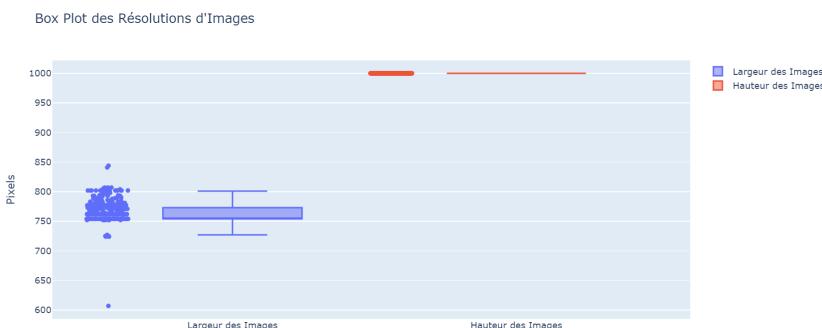


Commentaires & Constat métier

- Observation :** La majorité des images ont une taille de fichier comprise entre 0.05 et 0.1 Mo.
- Recommandation :** Les images semblent être de taille appropriée pour les traitements ultérieurs.

I.B. Résolutions des Images

L'analyse des résolutions des images permet de vérifier l'homogénéité des résolutions. Si les résolutions varient beaucoup, un redimensionnement des images peut être nécessaire pour garantir la cohérence dans le traitement ultérieur.



Les images ont toutes la même hauteur de 1000px. Cependant la largeur n'est pas uniforme.

L'analyse de la distribution des largeurs des images permet de comprendre la variation des tailles d'images et de décider si une normalisation de la dimension est nécessaire.

Distribution des largeurs des images dans le dataset Kaggle

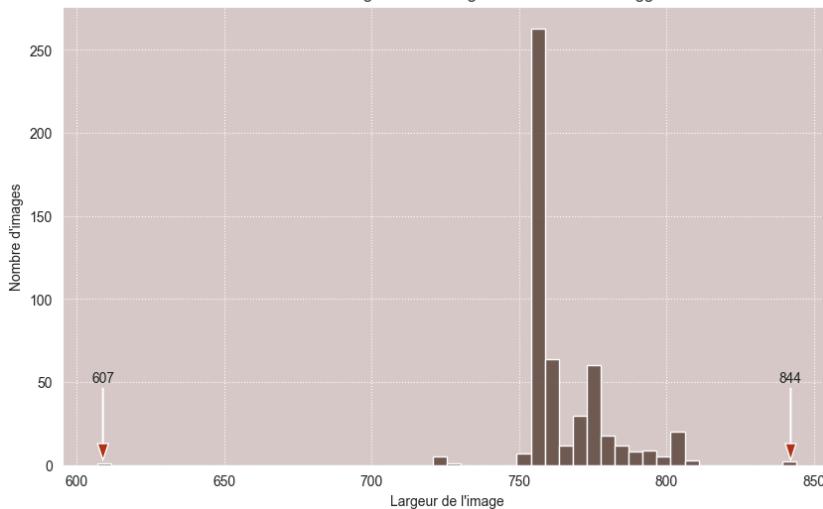


Image la plus petite et la plus grande du dataset Kaggle

Image avec la plus petite longueur

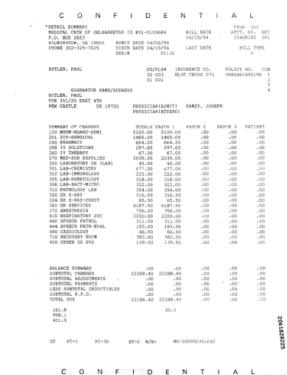
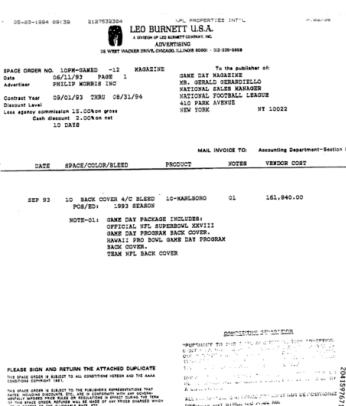


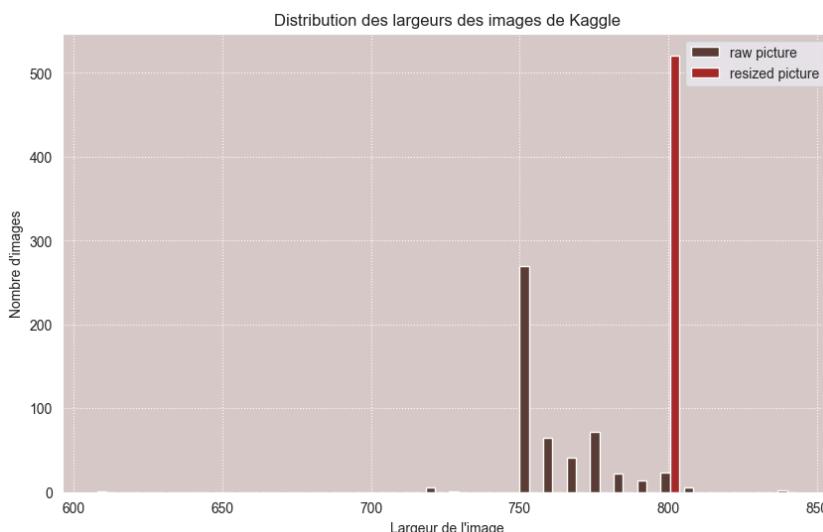
Image avec la plus grande longueur



Commentaires & Constat métier

- Observation :** La largeur des images varie principalement entre 750 et 800 pixels, tandis que la hauteur est uniformément à 1000 pixels.
- Recommendation :** Pour assurer une uniformité dans les dimensions des images, un redimensionnement à une résolution standard (par exemple, 800x1000 pixels) pourrait être appliqué. L'ajout de pixels neutre sur l'un des cotés ne modifirait pas l'information tout en uniformisant la taille des images. Cela faciliterait les traitements et analyses subséquents.

Avec l'uniformisation de la largeur des images à 800px :

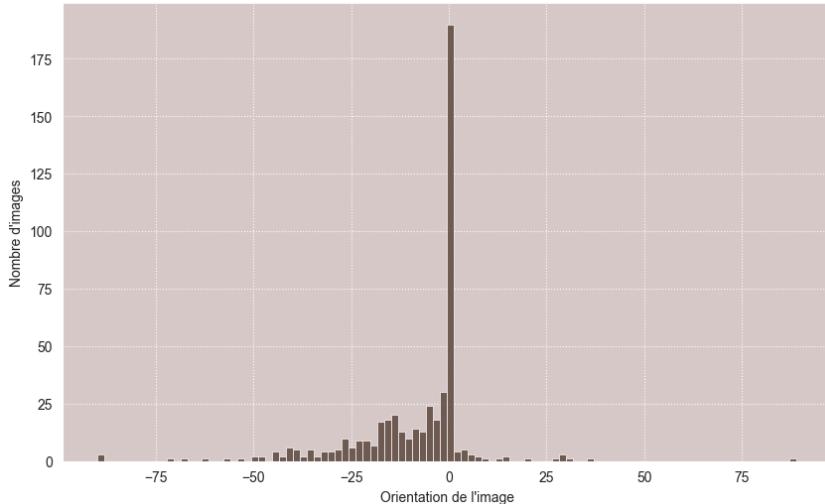


I.C. Analyse de la Rotation des Images

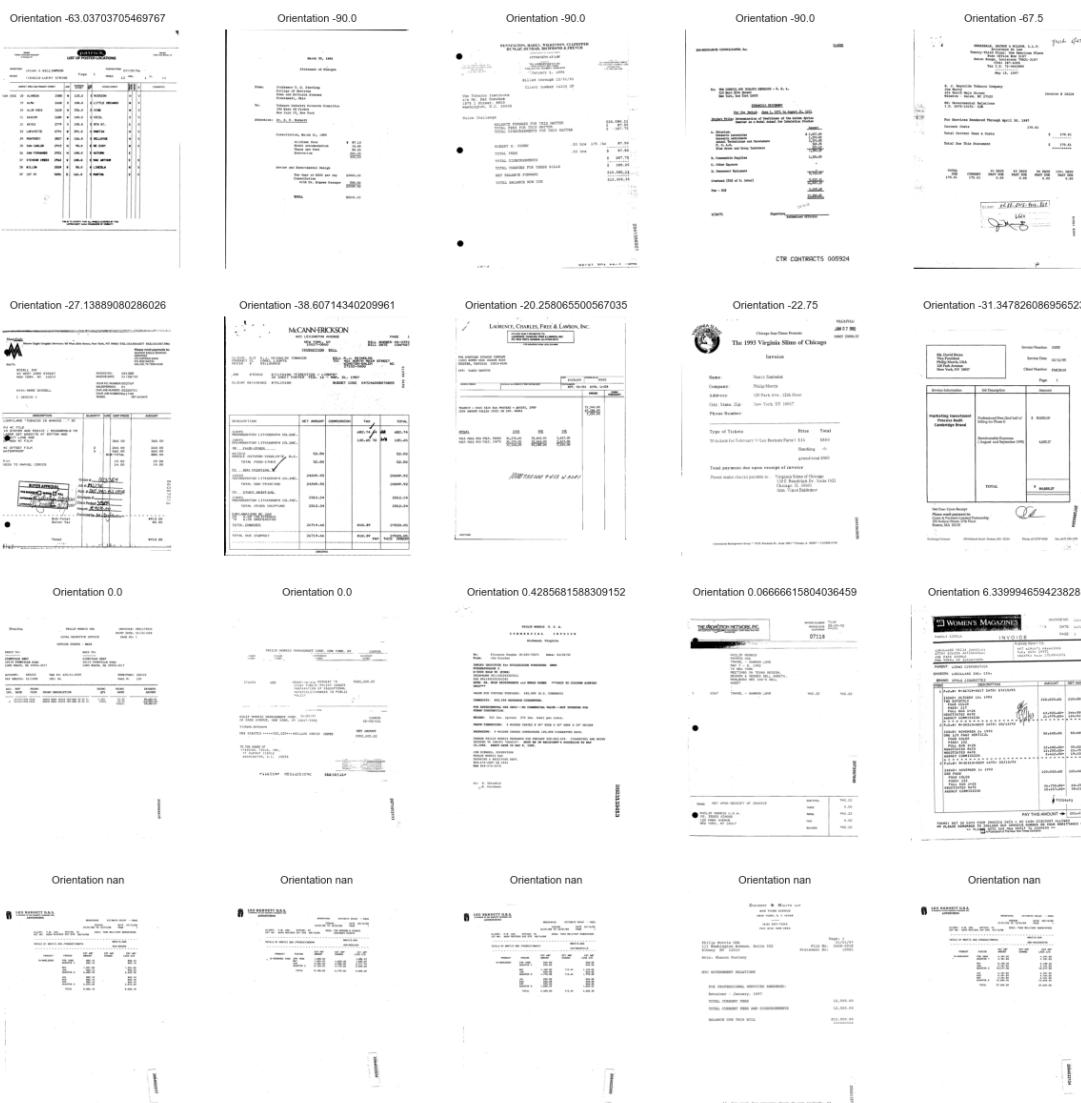
Les documents écrits doivent être correctement orientés pour une reconnaissance optimale. Analyser l'orientation des images permet d'identifier les images nécessitant une correction de rotation.

On récupère l'orientation des images avec la librairie build_features

Distribution des orientations des images dans le dataset Kaggle



Differentes images avec différentes orientations



Commentaires + Constat métier

- **Observation** : Il y a une diversité dans l'orientation des images, avec une concentration autour de 0 degré.
35 documents sur 520, soit 6,7% des images n'ont pas d'orientation.
Cela est vraisemblablement du à la présence de texte en position vertical et horizontal présent sur le document.

La grande majorité des images ont une orientation verticale.
On observe que beaucoup d'image ont une orientation entre 0° et 90°. Cependant, lorsque l'on étudie l'image, on remarque que celle-ci est bien verticale.

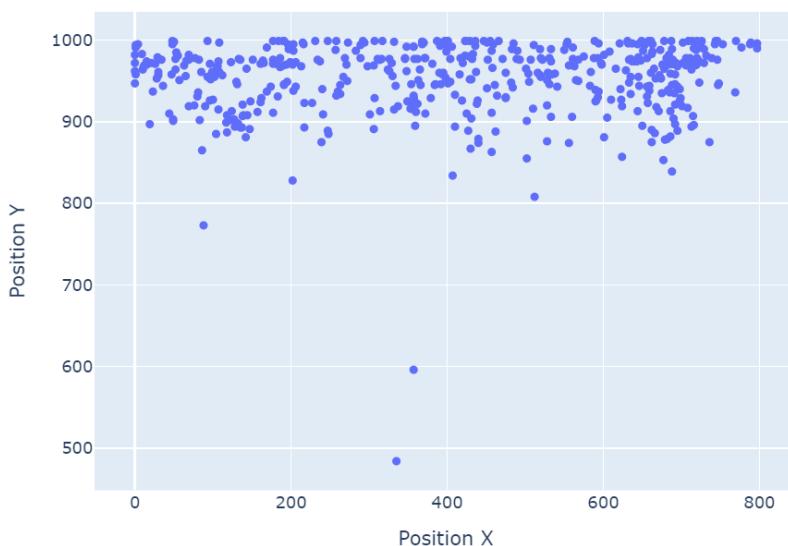
L'orientation provient des autres éléments de l'image, par exemple des marges, des tableaux, des logos, de l'écriture manuscrite.
- **Recommendation** : Une correction d'angle (rotation) pour aligner les textes horizontalement serait bénéfique.
L'utilisation de la détection de l'angle d'orientation et la rotation corrective en conséquence pourrait standardiser l'orientation des documents. On restera attentif à l'incidence de l'orientation sur le choix de l'OCR

II. Analyse du texte

II.A. Analyse du Contenu Texte et Bordures

Pour les documents écrits, il est crucial de s'assurer que le contenu texte est centré et bien contenu dans l'image. Cette analyse aide à déterminer si des ajustements de translation sont nécessaires pour recentrer le contenu.

Position du Contenu Texte dans les Images



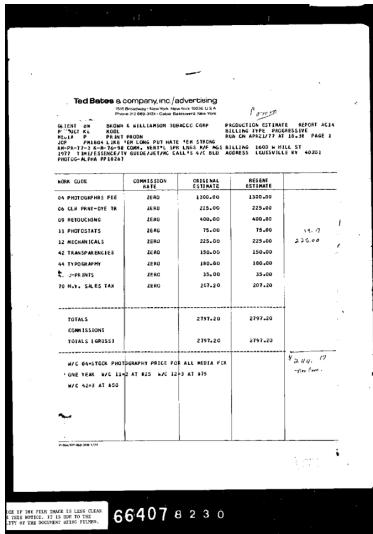
Commentaires & Constat métier

- **Observation** : La position du contenu texte est majoritairement en haut des images.
- **Recommendation** : Pour les analyses nécessitant l'extraction de texte, un recadrage automatique pour cibler les zones où le texte est principalement situé pourrait être mis en place. Cela permettrait de concentrer les efforts de traitement sur les zones d'intérêt.

II.B. Taille Moyenne des Lettres

La taille moyenne des lettres peut révéler des informations sur le type de document (une publicité qui diffère d'un document comptable). Une taille de lettre plus petite peut nécessiter une meilleure résolution pour la reconnaissance de caractères. Selon les résultats, une augmentation de la résolution ou un zoom sur les sections de texte pourrait être envisagé.

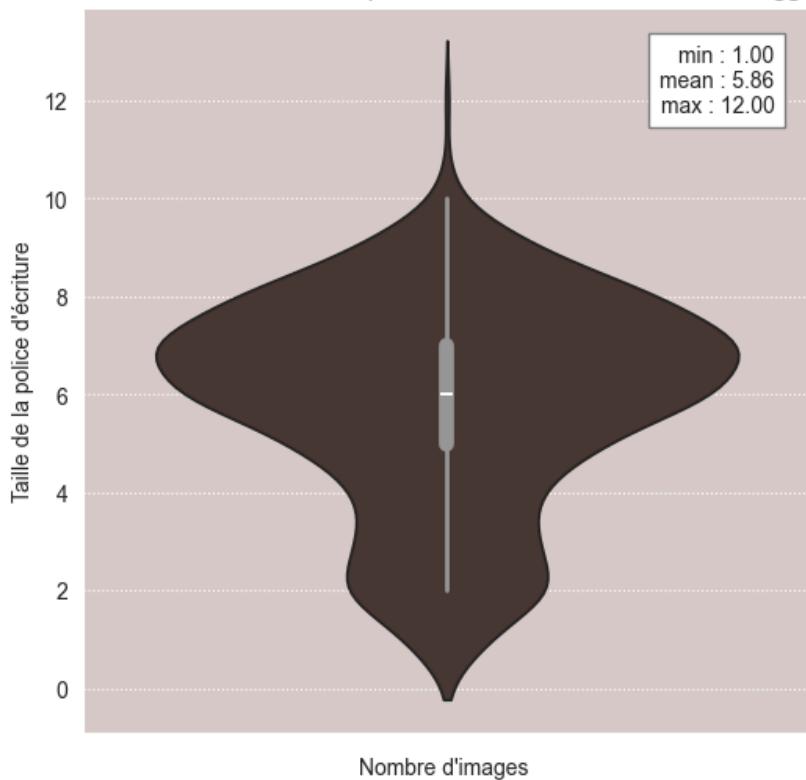
Un document à une taille de police abérante de 968px, soit la taille du document.
Une visualisation rapide permet de voir que la bordure a été détectée comme un caractère



66407 8 2 3 0

On étudie la répartition des tailles de police en excluant l'image abérante

Distribution des tailles de police d'écriture dans le dataset Kaggle



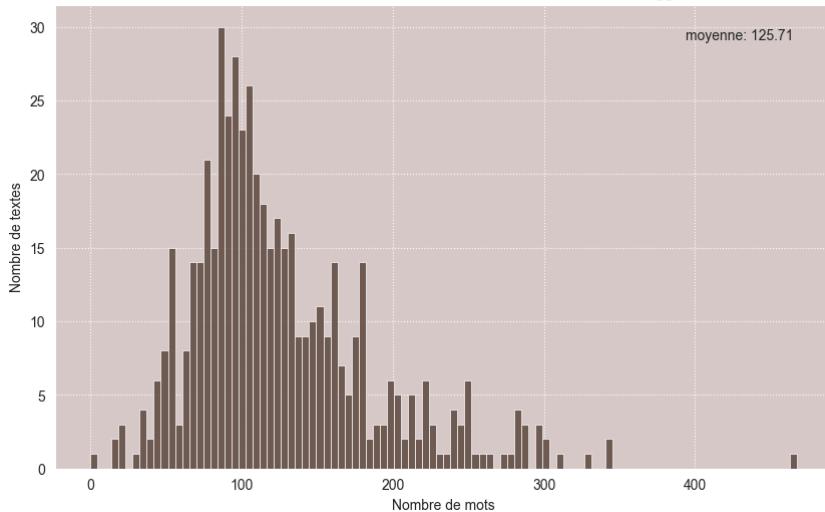
Commentaire & Constat Metier

- Observation :** On remarque que un document possède une bordure qui a été confondue avec une lettre. La taille moyenne des lettres est de 5.86 pixel. Les documents ont tous une résolution identique de 72 dpi
- Recommendation :** Une faible taille de police peut être impactant pour la detection de caractères de l'OCR. De plus, le matériel de numérisation utilisé implique une résolution faible de 72dpi. Un eventuel agrandissement des images pourra être pris en compte si la resolution n'est pas satisfaisante pour l'OCR et ne permet pas la detection de caractères. Les premières évaluations ne montre pas de besoin d'agrandissement pour detecter les lettres.

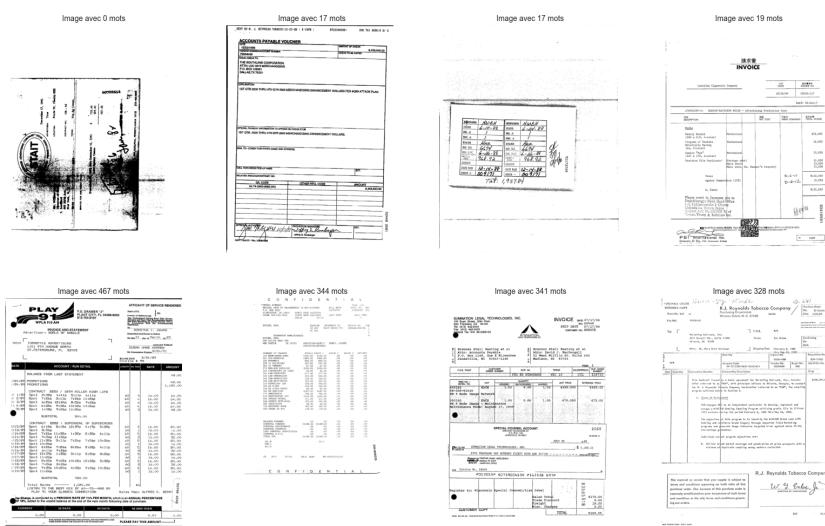
II.C. Nombre Moyen de Mots

Le nombre moyen de mots dans un document peut indiquer la densité d'information. Une densité de mots élevée peut nécessiter des techniques de prétraitement spécifiques comme la segmentation du texte. Selon les résultats, un traitement spécifique pour gérer les documents denses en texte pourrait être appliquée.

Distribution du nombre de mots dans les textes du dataset Kaggle

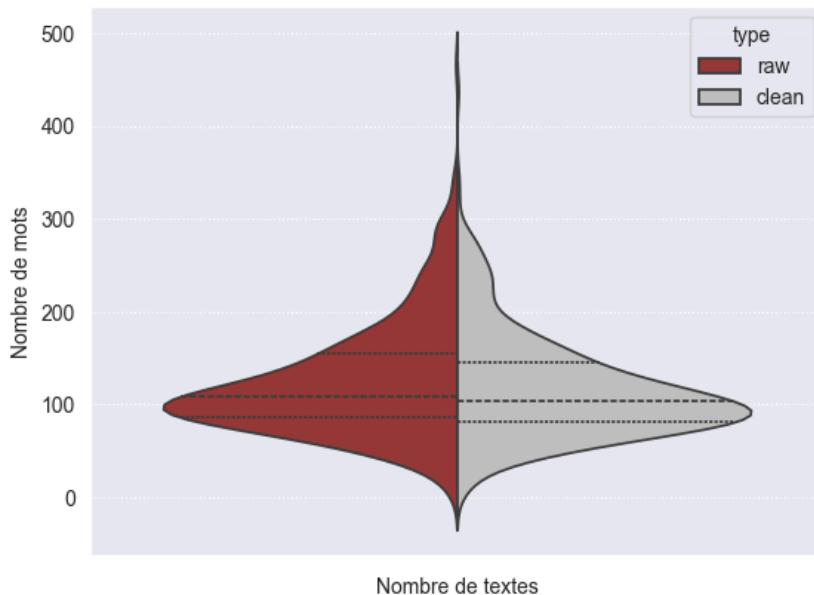


Differentes images avec différentes nombres de mots détectés



On regarde si la présence de stopwords influe sur le nombre de mots détectés

Distribution des textes du dataset Kaggle avant et après nettoyage



Commentaire :

- Observation :** Les documents sont assez uniformément distribués autour de 100 mots. On observe que les valeurs extrêmes sont dues à des petits documents peu remplis ou bien illisibles d'une part et à des documents comportant beaucoup de lignes écrites d'autre

part. Le nombre de mot est cohérent avec ce que l'on observe en regardant les images. On remarque que les données détectées ne sont pas parfaite et que beaucoup de texte peut au final ne pas être détecté.

On observe que la présence de stopwords reste faible et impacte faiblement la taille des textes.

- **Recommendation** : On cherche un OCR qui détecte le texte aussi bien que la cible du dataset

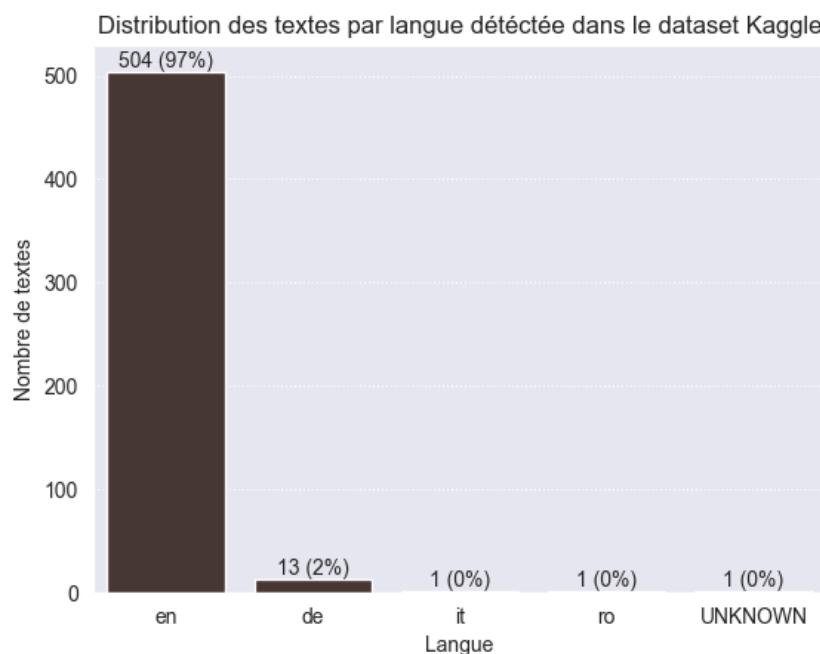
II.D. Langue la plus fréquente

Afin de détecter les bons mots et ne pas supprimer d'informations pour l'entraînement du modèle NLP, il est nécessaire de regarder la langue dominante qui ressort de chaque texte.

Ainsi, on pourra adapter les stopwords pour éviter de supprimer des informations importantes.

En fonction des résultats, on adaptera la liste des mots à exclure pour affiner les analyses de contenu.

On détecte la langue en utilisant LanguageDetector de spacy



Commentaire :

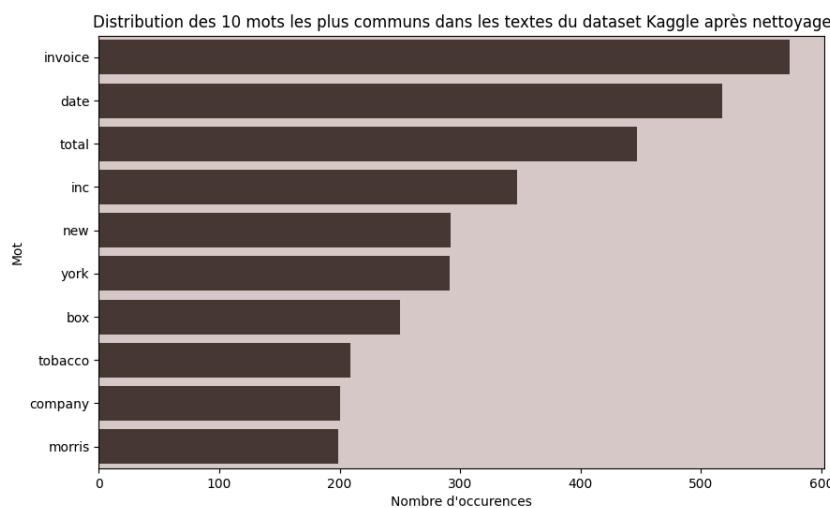
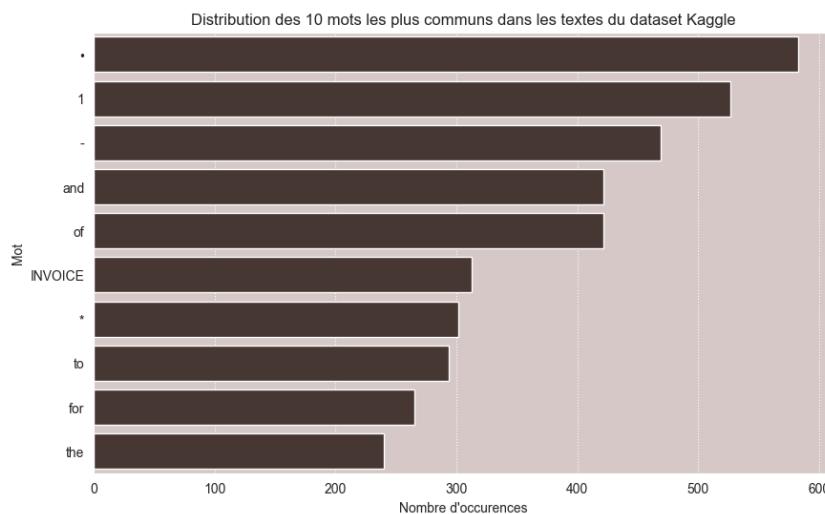
- **Observation** : Une majorité des textes (97%) sont en anglais. Dans les textes dans d'autres langues, seul une partie du texte n'est pas en anglais, souvent des noms propres.
- **Recommendation** : On se concentrera principalement sur la sélection de stopwords en anglais car cela représente la majorité des documents du dataset Kaggle. Les documents ayant une autre langue ont un mélange entre l'anglais et une autre langue. Ces documents auront peu d'incidence sur le modèle NLP.

II.E. Mots les Plus Fréquents

Identifier les mots les plus fréquents permet de comprendre le contenu principal des documents et d'éliminer les mots sans intérêt pour les analyses de texte. En fonction des résultats, une liste de mots à exclure pourrait être générée pour affiner les analyses de contenu.

On observe une forte présence de stopwords et ponctuation qui pollue la détection de mots.

On enlève ces éléments pour faire ressortir les mots les plus fréquents.



Commentaire :

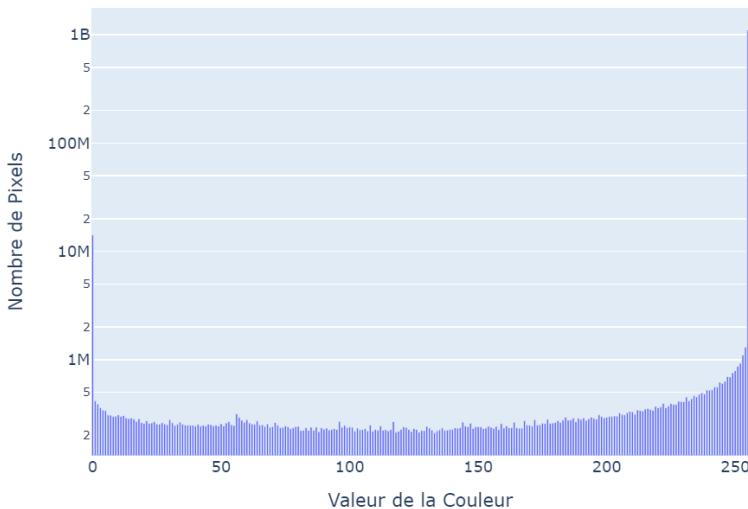
- **Observation :** Les stopwords sont assez fréquents et polluent la lecture des fichiers texte de façon automatique. Après nettoyage, on remarque que le mot le plus fréquent est Invoice ce qui est cohérent avec le dataset kaggle qui ne regroupe que des factures.
- **Recommendation :** On éliminera les stopwords avant d'utiliser un modèle NLP afin de ne conserver que les mots importants et facilitant la classification du document

III. Analyse des couleurs

III.A. Distribution des Intensités de Couleurs

Analyser la distribution des intensités de couleurs permet de comprendre la répartition des couleurs dans les images et d'identifier les éventuelles valeurs aberrantes. Cela aide à décider si un ajustement de la balance des couleurs ou une normalisation est nécessaire.

Distribution des Intensités de Couleurs



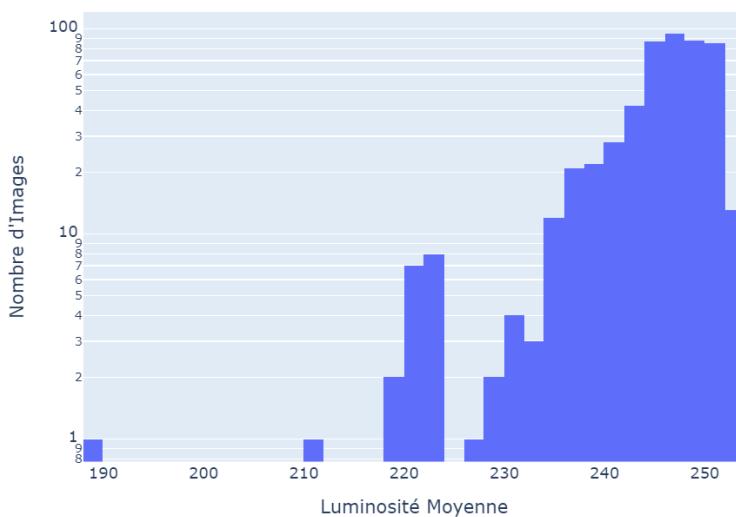
Commentaires & Constat métier

- Observation :** Une majorité des pixels ont des valeurs d'intensité élevées, proche de 250.
Les images sont déjà assez contrastées à la vue des pics autour de 0 et 250
- Recommendation :** Il pourrait être bénéfique d'augmenter le contraste ou d'effectuer une égalisation d'histogramme pour les images où la majorité des pixels ont des valeurs d'intensité intermédiaires, afin d'améliorer la visibilité et la différenciation des caractéristiques.

III.B. Luminosité Moyenne

L'analyse de la luminosité moyenne des images permet de vérifier si les images ont des niveaux de luminosité cohérents. Des ajustements peuvent être nécessaires pour corriger les variations de luminosité afin d'assurer une qualité d'image uniforme.

Distribution de la Luminosité Moyenne



Commentaires & Constat métier

- Observation :** La plupart des images ont une luminosité moyenne élevée, avec un pic autour de 240.
- Recommendation :** Pour les images avec une luminosité moyenne faible, une correction de luminosité pourrait être appliquée pour améliorer la lisibilité. Pour les images très lumineuses, une réduction de la luminosité pourrait être nécessaire pour éviter la saturation.

III.C. Distribution des Couleurs pour Chaque Image

Analyser la distribution des couleurs pour chaque image permet d'identifier les variations de couleur d'une image à l'autre. Cela peut aider à déterminer si des ajustements spécifiques par image sont nécessaires pour la normalisation des couleurs.

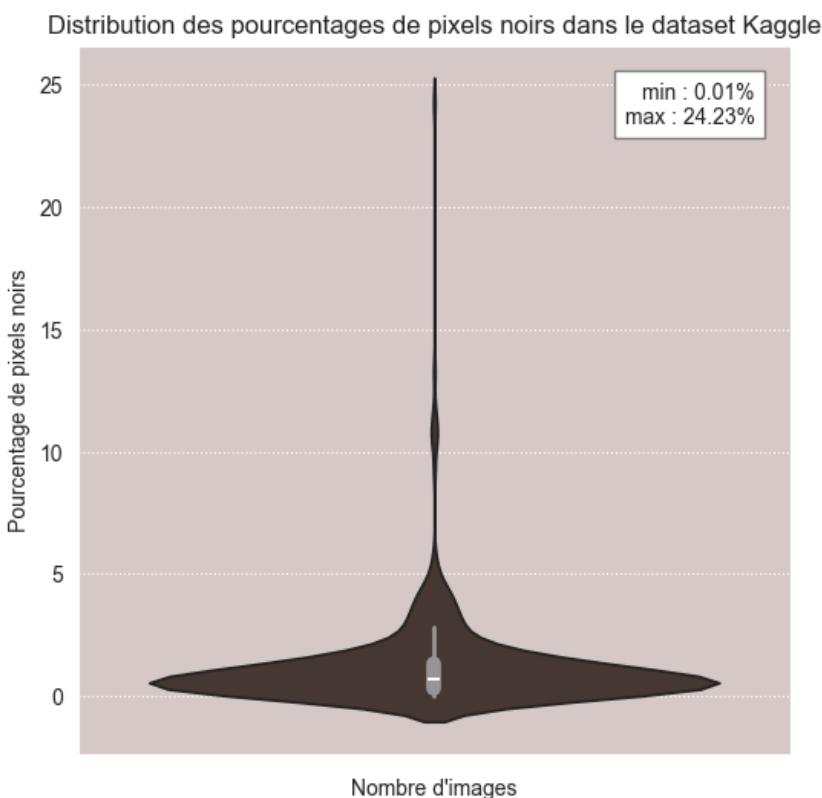
Commentaires & Constat métier

- **Observation :** La plupart des images ont une luminosité moyenne élevée, avec un pic autour de 240.
- **Recommendation :** Pour les images avec une luminosité moyenne faible, une correction de luminosité pourrait être appliquée pour améliorer la lisibilité. Pour les images très lumineuses, une réduction de la luminosité pourrait être nécessaire pour éviter la saturation.

III.D. Pourcentage de Pixels Noirs dans l'Image

Le pourcentage de pixels noirs peut indiquer la densité de numérisation ou la présence d'arrière-plans sombres et d'images importantes. Une forte densité de pixels noirs pourrait nécessiter un prétraitement pour améliorer le contraste.

Selon les résultats, une augmentation du contraste, un filtrage des pixels noirs ou une inversion des images pourrait être nécessaire.



Commentaire :

- **Observation :** La majorité des images possèdent une proportion de 1% à 5% de pixels sombres. Les quelques images possédant une bordure sombre explique le pic à 24% de pixel sombre.
- **Recommendation :** La distribution des pixels sombres sur le dataset de Kaggle est assez uniforme et ne nécessite pas de prétraitement nécessaire pour ajuster les images

2.4.1. KRLV_CDIP DATASET

Le dataset initial comporte plus de 400.000 images.

Afin de simplifier l'analyse et de pouvoir comparer facilement avec le dataset de Kaggle, nous avons sélectionné 992 images, respectivement 62 images prises aléatoirement dans chacune des 16 catégories.

Préparons une liste principale avec les images ainsi que des tableau d'informations qui seront à analyser : les tailles, résolutions, distribution de couleur et luminosité.

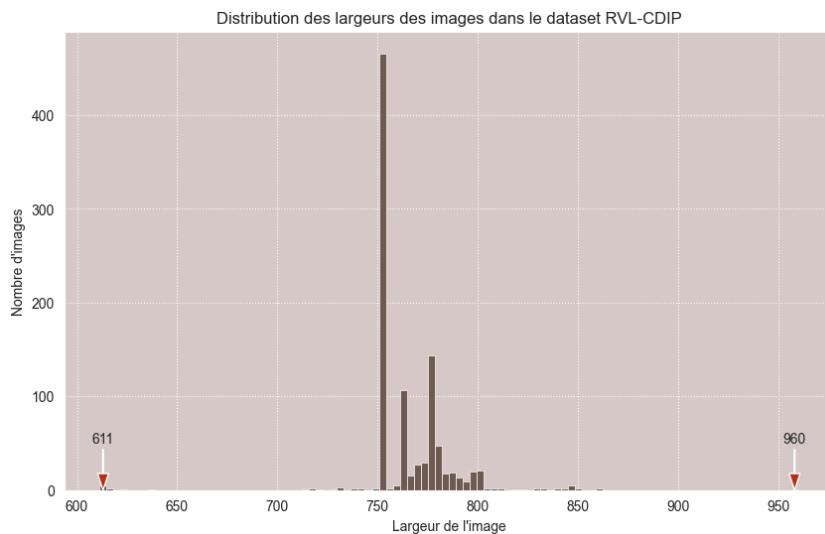
Enfin, regardons le premier élément pour avoir une idée plus précise des types d'éléments considérés.

I. Analyse des formats

I.A. Distribution des Dimensions d'Images

Les images ont toutes la même hauteur de 1000px. Cependant la largeur n'est pas uniforme.

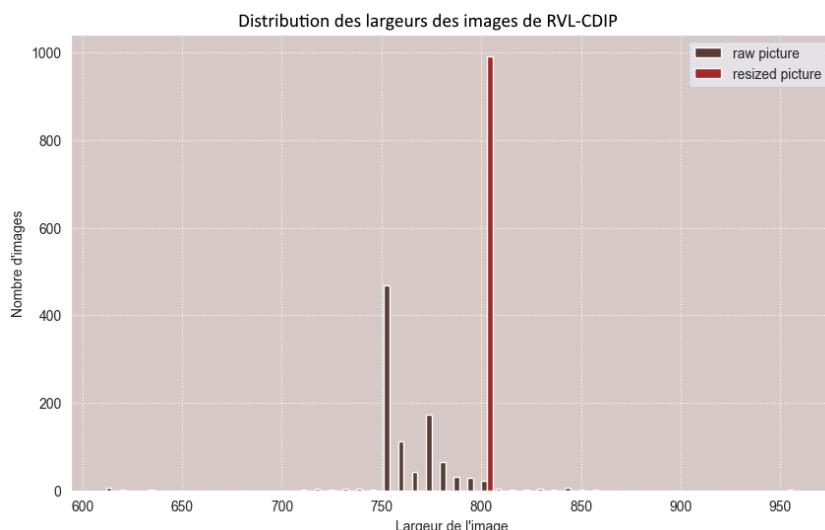
L'analyse de la distribution des largeurs des images permet de comprendre la variation des tailles d'images et de décider si une normalisation de la dimension est nécessaire.



Commentaires

- **Observation :** La largeur des images varie principalement entre 750 et 800 pixels, tandis que la hauteur est uniformément à 1000 pixels. On observe une répartition assez similaire à celle observé sur le dataset de Kaggle
- **Recommandation :** Pour assurer une uniformité dans les dimensions des images, un redimensionnement à une résolution standard (par exemple, 800x1000 pixels) pourrait être appliqué. Le redimensionnement pourra être identique pour l'entraînement du CNN et l'entraînement de l'OCR

En redimensionnant le dataset, on uniformise la taille des images pour l'utilisation du CNN et de l'OCR qui pourraient y être sensibles

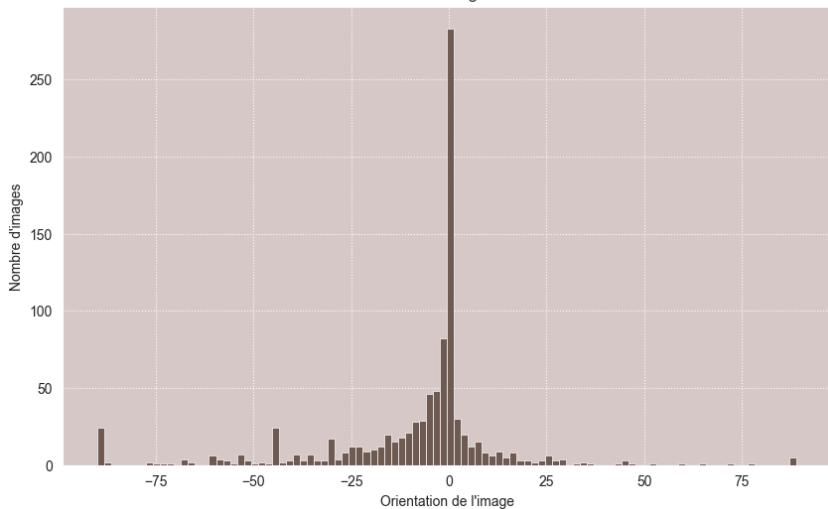


I.B. Analyse de la Rotation des Images

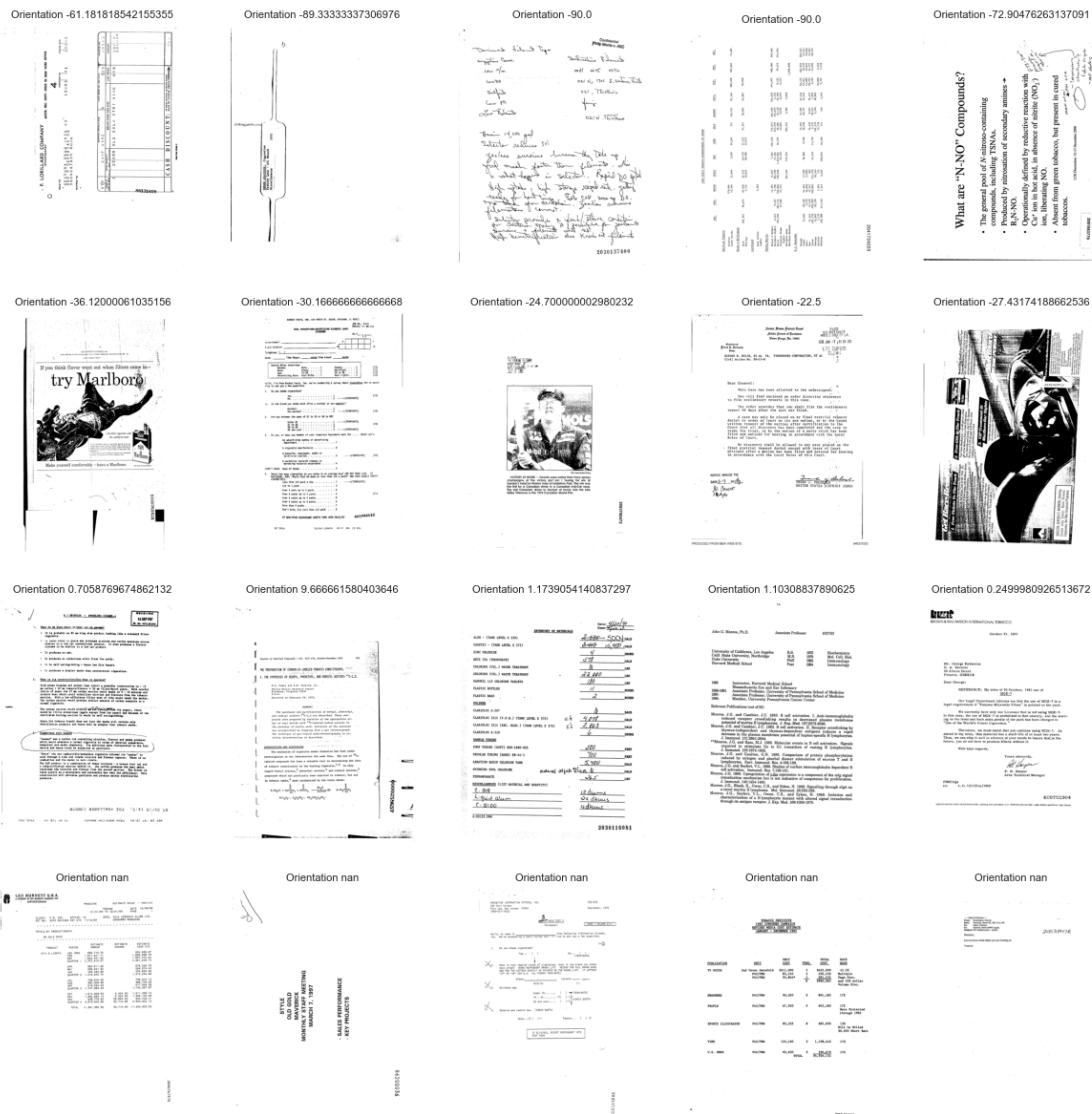
Les documents écrits doivent être correctement orientés pour une reconnaissance optimale. Analyser l'orientation des images permet d'identifier les images nécessitant une correction de rotation.

On récupère l'orientation des images avec la librairie build_features

Distribution des orientations des images dans le dataset RVL-CDIP



Differentes images avec différentes orientations



Commentaires

- Observation :** Il y a une diversité dans l'orientation des images, avec une concentration autour de 0 degré.

Les documents sans orientation sont du à la présence de texte en position vertical et horizontal sur le document.

On observe une plus grande diversité des orientations sur les images, qui n'est pas biaisé par la présence de lignes verticales et de

tableaux.

- **Recommandation :** Une correction d'angle pour aligner les textes horizontalement serait bénéfique pour un choix d'un OCR, cependant elle ne refléterait pas le comportement naturel pour la détection via un CNN.

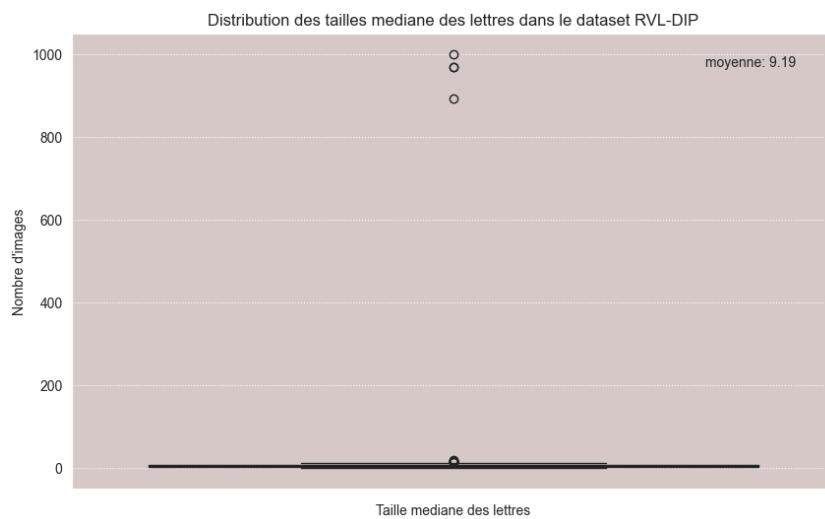
L'utilisation de la détection de l'angle d'orientation et la rotation corrective en conséquence pourrait standardiser l'orientation des documents. On restera attentif à l'incidence de l'orientation sur le choix de l'OCR et on appliquera la même transformation pour l'OCR et le CNN

II. Analyse du texte

II.A. Taille Moyenne des Lettres

La taille moyenne des lettres peut révéler des informations sur le type de document (une publicité en gros caractères contre un document comptable en petite police).

Une taille de lettre plus petite peut nécessiter une meilleure résolution pour la reconnaissance de caractères. Selon les résultats, une augmentation de la résolution ou un zoom sur les sections de texte pourrait être envisagé.

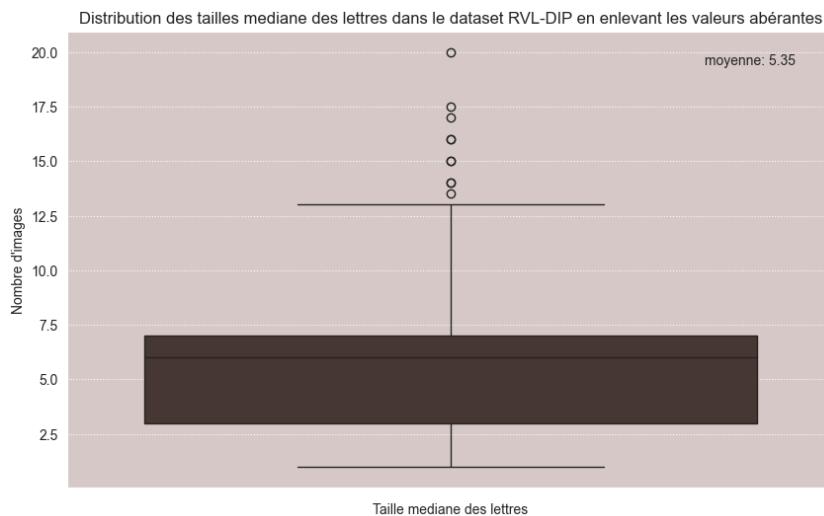


On remarque 3 valeurs abérantes. Ce sont des documents avec des tailles de caractères de plus de 800 px.
Une visualisation rapide permet de voir que la bordure a été détecté comme un caractère.

Differentes images avec différentes orientations



On étudie la répartition des tailles de police en excluant les images abérantes.



Commentaire :

- **Observation :** On remarque que certains documents possède des bordures et des fonds qui ont été confondues avec des lettres. La taille moyenne des lettres est de 5.35 pixel. Elle reste cohérente avec la taille de pixel observé sur le dataset de Kaggle. Les deux datasets sont cohérents sur la résolution et la taille des caractères. Les documents ont tous une résolution identique de 72 dpi
- **Recommendation :** On appliquera le même préprocessing sur la définition des documents sur le dataset d'entraînement de Kaggle et de RVL-CDIP pour l'OCR.

III. Analyse des couleurs

III.A. Luminosité Moyenne

L'analyse de la luminosité moyenne des images permet de vérifier si les images ont des niveaux de luminosité cohérents. Des ajustements peuvent être nécessaires pour corriger les variations de luminosité afin d'assurer une qualité d'image uniforme.

Commentaires

- **Observation :** La plupart des images ont une luminosité moyenne élevée, avec un pic autour de 240.
- **Recommendation :** Pour les images avec une luminosité moyenne faible, une correction de luminosité pourrait être appliquée pour améliorer la lisibilité. Pour les images très lumineuses, une réduction de la luminosité pourrait être nécessaire pour éviter la saturation.

III.B. Distribution des Couleurs pour Chaque Image

Analyser la distribution des couleurs pour chaque image permet d'identifier les variations de couleur d'une image à l'autre. Cela peut aider à déterminer si des ajustements spécifiques par image sont nécessaires pour la normalisation des couleurs.

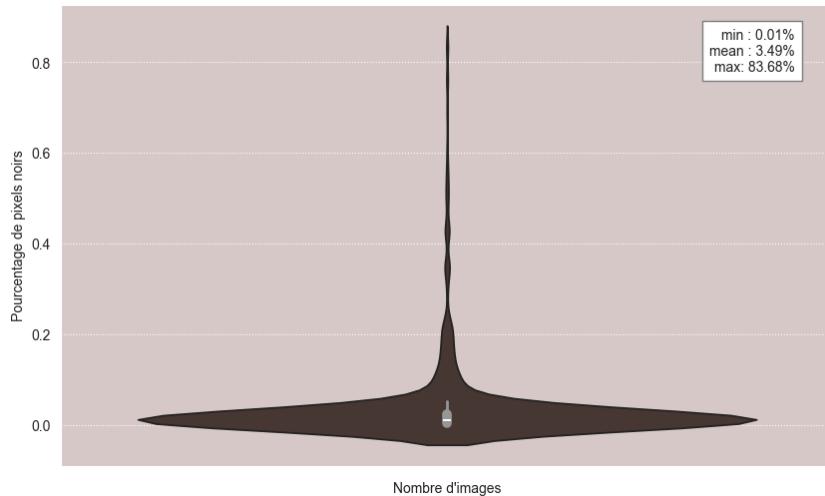
Commentaires

- **Observation :** Les images sont en noir et blanc.
- **Recommendation :** Aucune transformation nécessaire.

III.C. Pourcentage de Pixels Noirs dans l'Image

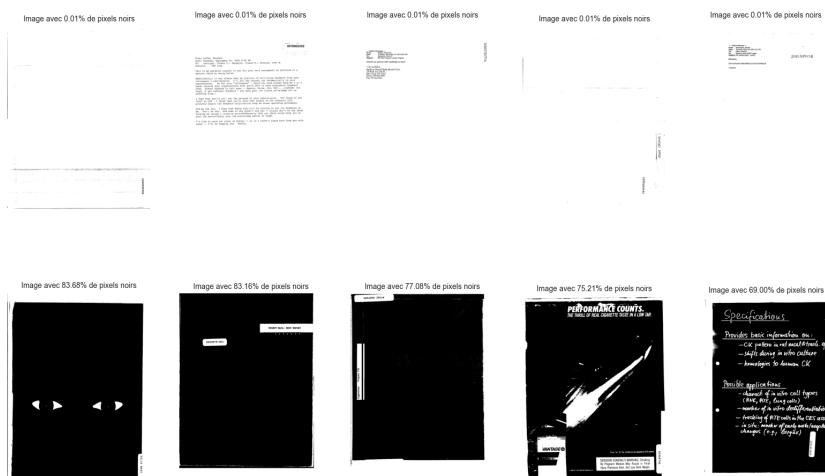
Le pourcentage de pixels noirs peut indiquer la densité de numérisation ou la présence d'arrière-plans sombres et d'images importantes. Une forte densité de pixels noirs pourrait nécessiter un prétraitement pour améliorer le contraste.

Distribution des pourcentages de pixels noirs dans le dataset RVL-CDIP



On remarque une distribution concentré autour d'une faible densité de pixel, avec des valeurs extremes assez distantes de la moyenne.
On observe en détail un extrait des images avec une faible et une forte densité de pixels noirs.

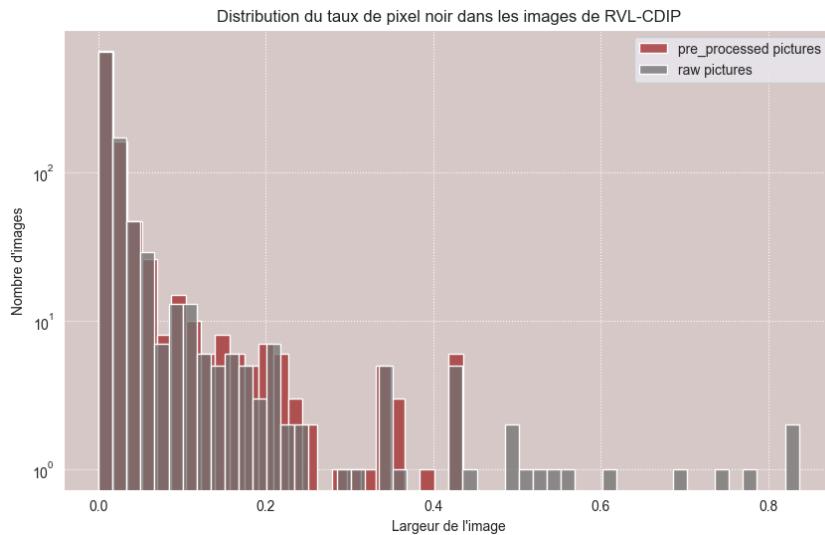
Differentes images avec différentes nombres de mots détectés



Commentaire :

- **Observation :** La majorité des images possèdent une proportion de 1% à 5% de pixels sombres.
Cependant, on observe des documents avec peu d'écriture et donc une faible densité de pixels sombres et à l'inverse des images majoritairement noires.
On observe aussi sur des images sombres que le texte est écrit en blanc sur fond noir.
- **Recommendation :** Afin de rééquilibrer les données et conserver une distribution uniforme, on préconise une inversion du noir et du blanc pour les images ayant plus de 50% de pixels sombres afin de recentrer la distribution.

En inversant le noir et le blanc pour les images ayant plus de 50% de pixels noirs, on uniformise le dataset pour que les valeurs aberrantes aient moins d'impact sur l'entraînement.



3. Méthodologie

L'approche choisie pour ce projet est une combinaison de deux axes principaux : la classification visuelle et la classification textuelle.

La classification visuelle, implique l'utilisation d'un réseau de neurones convolutif (CNN). Les couches de neurones convolutives des CNN les rendent très efficaces pour traiter les images. L'idée ici est d'apprendre à un CNN à reconnaître certaines caractéristiques des documents qui permettraient de les classifier. Par exemple on peut imaginer qu'un document qui présente de nombreuses lignes verticales et horizontales a plus de chances d'être une facture ou un rapport scientifique qu'une publicité. Nos CNN ont été entraînés sur un ensemble de documents scannés étiquetés du dataset RVL-CDIP. Notre espoir est qu'on puisse arriver à un score d'accuracy correct simplement en utilisant des indices visuels avec cette technique.

La classification textuelle, est basé sur le traitement du langage naturel (NLP). Pour cela, nous avons d'abord utilisé une librairie OCR pour extraire le texte des documents scannés. Ensuite, nous avons entraîné un modèle de NLP sur le texte extrait pour classer les documents en fonction de leur contenu textuel. La quantité de mots, la présence ou fréquence de mots particuliers peuvent être de bons indicateurs pour classifier un document. Par exemple l'apparition du mot "invoice" serait un bon indicateur que le document est une facture.

Enfin, pour améliorer les performances de nos modèles individuels, nous avons mis en place un modèle de vote. Ce modèle prend en compte les prédictions de nos deux modèles (CNN et NLP) et décide de la classe finale du document en se basant sur ces prédictions. Cette approche nous permet de prendre en compte les forces et faiblesses de chaque modèle et d'améliorer la précision globale de notre classification.

4. Entraînement du CNN.

4.1 CNN testés et résultats obtenus

Deux architectures de CNN ont été testées dans le cadre de ce projet : un modèle classique et un modèle pré-entraîné. Chaque modèle a été évalué en fonction de son exactitude (accuracy) sur un ensemble d'entraînement de 20 000 documents.

Le modèle de réseau neuronal convolutif (CNN) présenté ici est conçu pour la classification d'images. Il est composé de plusieurs couches de convolutions et de couches entièrement connectées, permettant d'extraire et d'apprendre des caractéristiques complexes des images.

- Modèle 1 : CNN simple

1. Architecture du Modèle

Le modèle CNN est structuré comme suit :

- Couche de Convolution 1 (conv1)
- Couche de Convolution 2 (conv2)
- Couche de Convolution 3 (conv3)
- Couche Entièrement Connectée 1 (fc1)
- Couche Entièrement Connectée 2 (fc2)

2. Flux de Données dans le Modèle

- Entrée : L'image d'entrée est une image RGB avec 3 canaux.
- Conv1 : La première couche de convolution applique 32 filtres de taille 3x3, suivie d'une activation ReLU et d'une opération de pooling pour réduire la dimension.

- Conv2 : La deuxième couche de convolution applique 64 filtres de taille 3x3, suivie d'une activation ReLU et d'une opération de pooling.
- Conv3 : La troisième couche de convolution applique 128 filtres de taille 3x3, suivie d'une activation ReLU et d'une opération de pooling.
- Flatten : Les caractéristiques extraites sont aplatis en un vecteur.
- FC1 : La couche entièrement connectée 1 applique une transformation linéaire, suivie d'une activation ReLU.
- FC2 : La couche entièrement connectée 2 produit les sorties correspondant au nombre de classes de la tâche de classification.



3. Performance du model

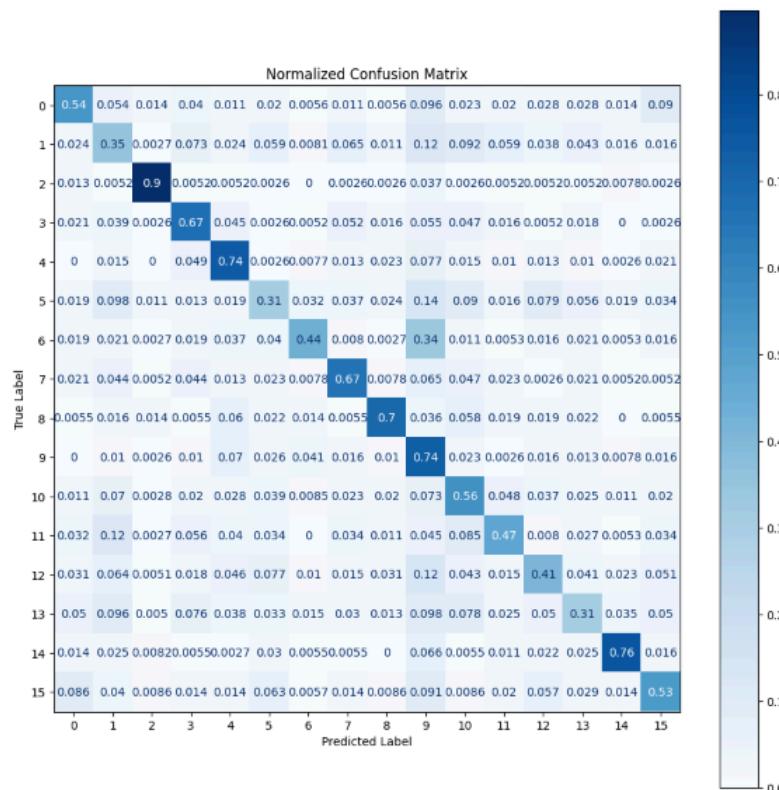
- **Exactitude de la Validation et du Test :**

Les précisions de validation et de test sont d'environ **56%**. Cela indique que le modèle fonctionne de manière relativement cohérente sur les ensembles de validation et de test, mais le pourcentage reste faible.

- **Perte de Validation et de Test :**

Le coût de validation est de 3,4425 et celle de test est de 3,4276. Les valeurs élevées de coût indiquent que le modèle ne détecte pas assez efficacement les caractéristiques des documents ce qui pourrait venir d'une complexité insuffisante ou d'un manque de données.

- **La Matrice de Confusion**



- **Analyse de la Matrice de Confusion :**

- **Performance par Classe :**

Certaines classes sont relativement bien reconnues par le modèle, comme les classes 2, 8 et 14 représentant respectivement les emails, les dossiers et les CV qui ont des valeurs élevées sur la diagonale, indiquant une bonne précision.

Ces images ont un format spécifique qui leur sont propres. Les emails et les CV ont une mise en forme bien spécifique alors que les dossiers présentent peu de texte avec de grandes zones rectangulaires unies.

D'autres classes comme les classes 1, 5, 6, 11 à 13 qui représentent respectivement des questionnaires, des papiers scientifiques, présentations et factures, montrent une performance plus faible avec des valeurs plus basses sur la diagonale et des confusions plus fréquentes avec d'autres classes.

Le manque de détails ressortis par le modèle ne permet pas de distinguer ces documents qui ont des formes similaires et la présence de données équivalentes. Ils contiennent tous des paragraphes avec la présence de chiffres ou de tableaux.

- **Confusions Notables :**

Les classes 1 (formulaire) et 12 (présentation) montrent des confusions significatives avec plusieurs autres classes. La classe 7 (cahier des charges) est souvent confondue avec les classes 0 (lettre) et 3 (document manuscrit). La classe 13 (questionnaire) est fréquemment confondue avec les classes 0, 3 et 7 (lettre, document manuscrit, cahier des charges).

4. Conclusion

Le modèle montre une performance modeste avec une précision stable sur les ensembles de validation et de test. Cependant, il existe des classes spécifiques où la performance pourrait être améliorée en réduisant les confusions.

5. Améliorations Potentielles :

Le modèle ne semble pas faire ressortir suffisamment d'informations pour pouvoir différencier les différents types de documents. Plusieurs possibilités s'offrent à nous :

- **Augmentation des données :** Augmenter la quantité de données d'entraînement pour que le modèle puisse mieux appréhender les différentes classes
- **Augmentation de la période d'apprentissage :** Augmenter le nombre d'époques d'apprentissage afin de réduire le coût et améliorer la précision
- **Complexifier le modèle :** Augmenter le nombre de couches neuronales afin d'accroître le nombre d'informations et de détails pour mieux distinguer les différentes classes

Augmenter le nombre d'époques ne ferait qu'overfitter le modèle et n'améliorerait pas la précision que sur le dataset d'entraînement. De plus, cela prendrait beaucoup de temps pour un résultat incertain. Augmenter le dataset reste peut concluant car il augmenterait le temps d'apprentissage sans apporter d'informations supplémentaires sur le dataset.

On priviliegera l'ajout de couches supplémentaires afin d'accroître le niveau d'information même si cela ajoute du temps de traitement

- Modèle 2 : CNN complexifié

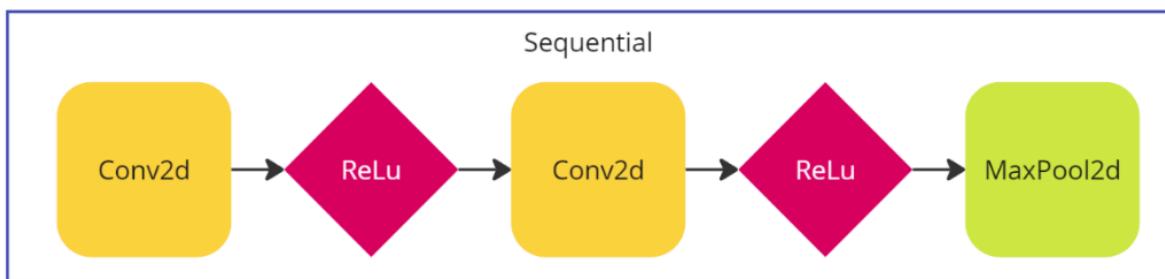
1. Architecture du Modèle

Le modèle CNN est structuré comme suit :

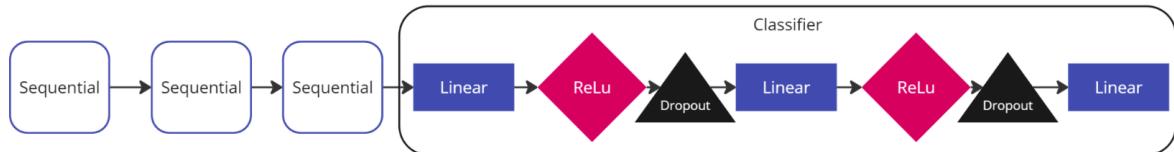
- Couche de Convolution 1 (seq1)
- Couche de Convolution 2 (seq2)
- Couche de Convolution 3 (seq3)
- Couche classificatrice (Classifieur)

2. Flux de Données dans le Modèle

- Entrée : L'image d'entrée est une image en niveau de gris avec 1 canal.
- Seq1 : La première couche de convolution applique 16 filtres de taille 5x5, suivie d'une activation ReLU et d'une opération de pooling pour réduire la dimension.
- Seq2 : La deuxième couche de convolution applique 32 filtres de taille 5x5, suivie d'une activation ReLU et d'une opération de pooling.
- Seq3 : La troisième couche de convolution applique 64 filtres de taille 5x5, suivie d'une activation ReLU et d'une opération de pooling.



- Classifieur : La couche linéaire applique une transformation linéaire, suivie d'une activation ReLU et d'un Dropout, deux fois afin de réduire les dimensions, éliminer les informations non pertinentes et éviter l'overfitting.

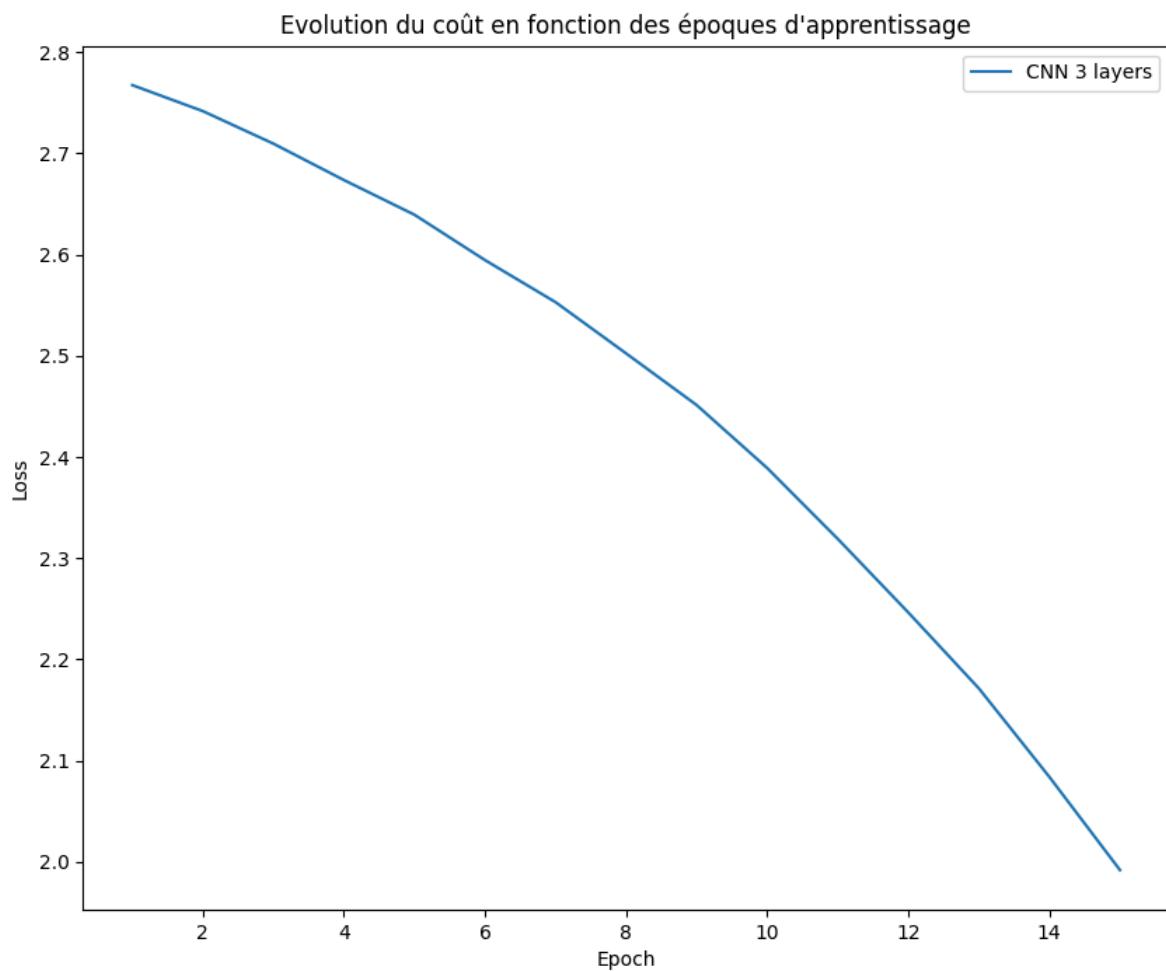


Cette architecture est intéressante car elle permet de facilement augmenter le nombre de couches de convolution jusqu'à 6 si le modèle est concluant.

3. Performance du modèle

Exactitude de la Validation et du Test :

Les précisions de validation et de test sont d'environ **33%** après 15 époques d'apprentissage. Cela indique que le modèle fonctionne de manière relativement cohérente sur les ensembles de validation et de test, mais le pourcentage reste faible. On a du limiter l'apprentissage à 15 époques car il a duré 1085 minutes. Le coût est réduit progressivement ce qui est encourageant avec un optimiseur SGD avec un learning rate de 1e-3. A la vue de la trajectoire, il faudrait augmenter le nombre d'époques afin d'atteindre un minimum local pour la fonction de coût.

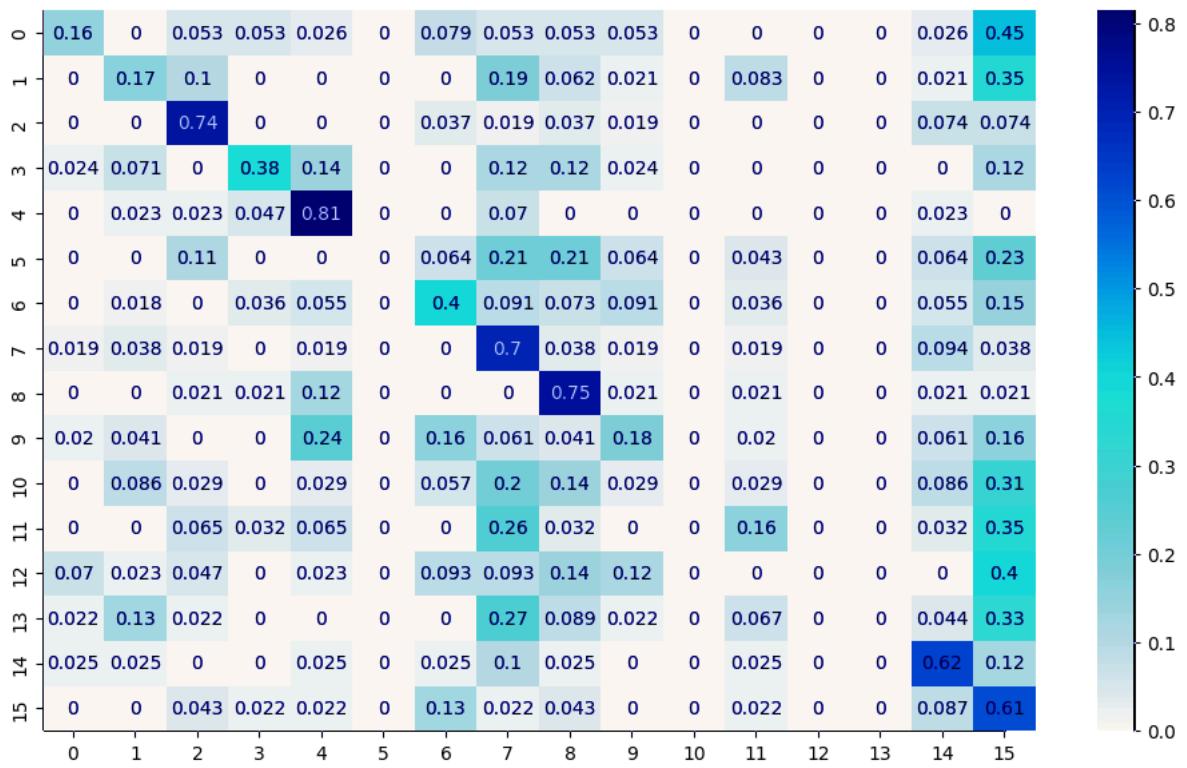


On obtient une précision de 33% ce qui est encourageant pour affiner le modèle.

Perte de Validation et de Test :

Le coût de validation et de test est de 2,01. Les valeurs élevées de coût indiquent que le modèle ne détecte pas assez efficacement les caractéristiques des documents ce qui pourrait venir d'une durée d'apprentissage faible.

La Matrice de Confusion



Analyse de la Matrice de Confusion :

Performance par Classe :

Certaines classes sont relativement bien reconnues par le modèle, comme les classes 2, 4, 7, 8, 14 et 15 représentant respectivement les emails, les dossiers, les CV et des mémos qui ont des valeurs élevées sur la diagonale, indiquant une bonne précision.

Ces images ont un format spécifique qui leur sont propres. Les emails et les CV ont une mise en forme bien spécifique alors que les dossiers présentent peu de texte avec de grandes zones rectangulaires unies. Les publicités sont aussi bien détectées du à la présence d'images.

D'autres classes comme les classes 5, 10, 12 et 13 qui représentent respectivement des rapports, des budgets, présentations et questionnaires, ne sont pas du tout détectées dans leurs classes respectives.

Le manque de détails ressortis par le modèle pour ces documents ne permet pas de les distinguer. Ils contiennent tous des paragraphes avec la présence de chiffres ou de tableaux.

Confusions Notables :

Les classes 15 (mémo) semble récupérer une majorité des documents lorsqu'ils ne sont pas assez bien détectés.

4.Conclusion

Le modèle montre une bonne performance sur quelques classes mais la précision reste instable. Le temps d'exécution reste long face à un coût qui n'a pas atteint son minimum. De plus, le modèle pourrait se voir amélioré par l'ajout de couche supplémentaire du fait de son architecture

5.Améliorations Potentielles :

Le modèle ne semble pas faire ressortir suffisamment d'informations sur certaines classes pour pouvoir différencier les différents types de documents. Plusieurs possibilités s'offrent à nous :

- **Augmentation des données :** Augmenter la quantité de données d'entraînement pour que le modèle puisse mieux appréhender les différentes classes
- **Augmentation de la période d'apprentissage :** Augmenter le nombre d'époques d'apprentissage afin de réduire le coût et améliorer la précision
- **Complexifier le modèle :** Augmenter le nombre de couches neuronales afin d'accroître le nombre d'informations et de détails pour mieux distinguer les différentes classes

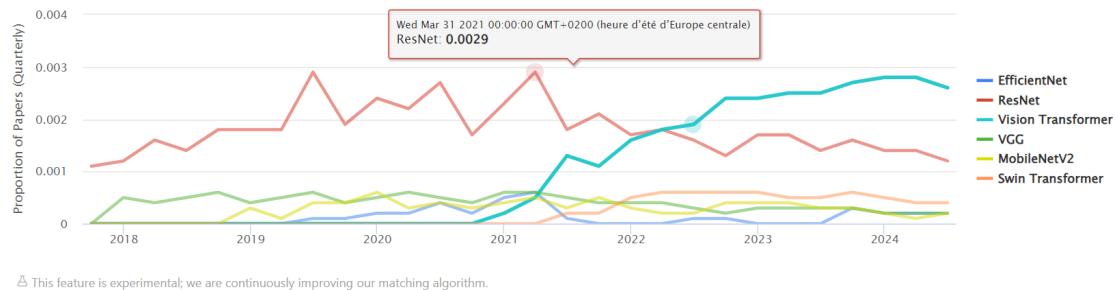
Le modèle est pertinent et avec un plus grand dataset et un plus grand nombre d'époques d'entraînement, le modèle serait plus fiable. Cependant le temps d'entraînement reste beaucoup trop long. Avec un temps plus grand d'entraînement et l'accès à plusieurs GPU pour paralleliser les calculs permettrait d'améliorer ce modèle prometteur.

On souhaite explorer le transfert de connaissance afin de trouver un modèle performant que l'on pourrait améliorer pour notre classification spécifique.

- Modèle par transfert de connaissance

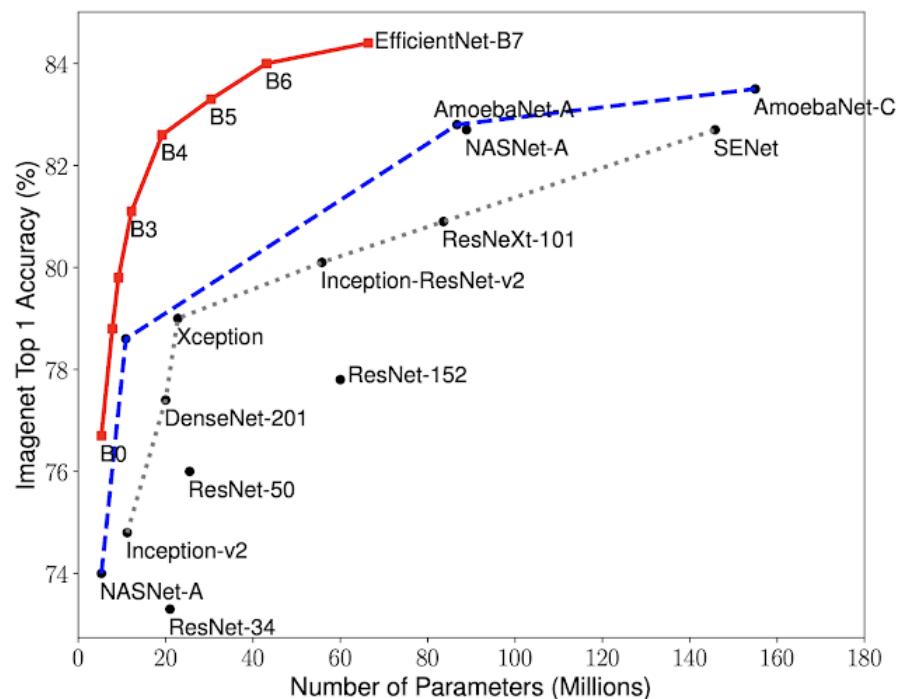
On s'intéresse aux différents modèles de classifications utilisant des réseaux de neurones. On remarque qu'un modèle en particulier se différencie des autres modèles :

Usage Over Time



On remarque EfficientNet comme un modèle devenu prépondérant et majoritaire, suivi de ResNet

Cela est principalement à la forte précision du modèle pour un faible nombre de paramètres en sortie et donc de couches neuronales et de temps d'apprentissage. Le modèle est performant dans la classification d'image mais aussi dans tout autre problème de classification.



EfficientNet :

L'application d'un modèle pré-entraîné tel que EfficientNet a été privilégiée en raison de ses caractéristiques remarquables en termes d'exigences en mémoire et en ressources matérielles. Comparé à d'autres modèles plus complexes, EfficientNet se distingue par sa légèreté, ce qui le rend particulièrement adapté aux environnements contraints en ressources.

En intégrant EfficientNet, il est possible d'améliorer les performances de notre modèle tout en minimisant la consommation de ressources matérielles.

- Modèle 3: Modèle Pré-entraîné EfficientNet B0

1. Introduction:

EfficientNet-B0 est un modèle de réseau neuronal convolutif (CNN) reconnu pour son efficacité et ses performances élevées. Il est pré-entraîné sur ImageNet, un dataset de plus d'un million d'image réparties en 1000 classes, ce qui améliore la rapidité et la précision de l'entraînement sur de nouveaux datasets.

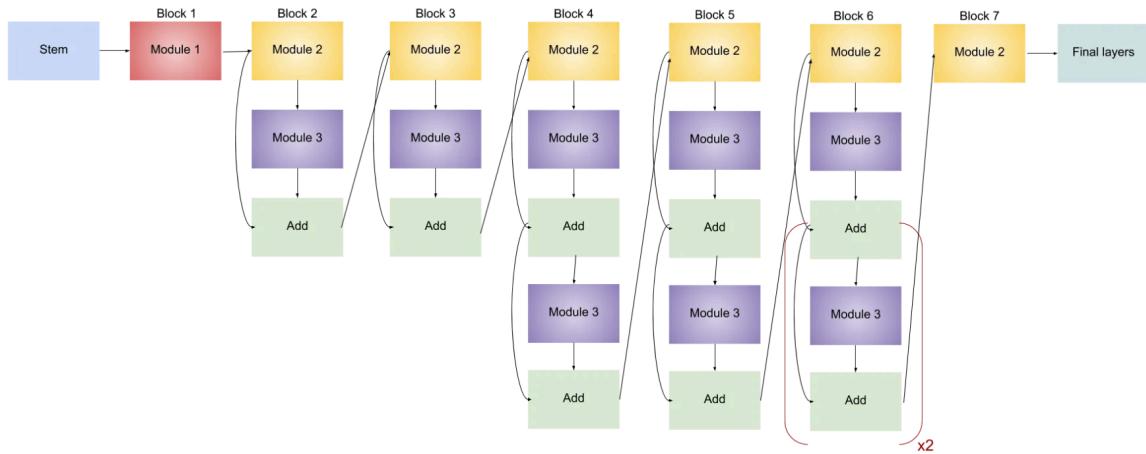
La version B0 est la plus petite et la plus légère de la famille EfficientNet, ce qui la rend appropriée pour des applications nécessitant une utilisation limitée de la RAM et du GPU.

2. Prétraitement et Initialisation

Le modèle que nous utilisons est pré-entraîné sur le dataset ImageNet, ce qui lui permet de bénéficier de connaissances préalables pour des tâches de classification d'images. L'utilisation de ces poids pré-entraînés accélère la convergence lors de l'entraînement sur notre propre dataset et améliore la performance du modèle.

3.Détails Techniques

- **Architecture de Base :** EfficientNet-B0 utilise un ensemble de blocs convolutifs efficaces qui combinent des convolutions standard et des convolutions de profondeur séparables. Cela réduit le nombre de paramètres tout en maintenant une performance élevée.



- **Stratégie d'Initialisation :** Les poids initiaux sont tirés des poids pré-entraînés sur ImageNet, spécifiés par `models.EfficientNet_B0_Weights.DEFAULT`.
- **Modification du Classificateur:** Pour adapter EfficientNet-B0 à notre tâche spécifique de classification, nous avons modifié la dernière couche du classificateur. Par défaut, la couche de sortie du modèle pré-entraîné est conçue pour classer les images en 1000 catégories (classes d'ImageNet). Nous avons remplacé cette couche par une nouvelle couche linéaire adaptée au nombre de classes de notre problème (16 classes)

4.Avantages de l'EfficientNet-B0

Efficacité : L'un des principaux avantages d'EfficientNet-B0 est son efficacité en termes de calcul et de mémoire, grâce à une architecture optimisée qui utilise des convolutions de profondeur séparables.

Précision : Malgré sa taille compacte, EfficientNet-B0 offre une précision comparable à celle de modèles beaucoup plus volumineux.

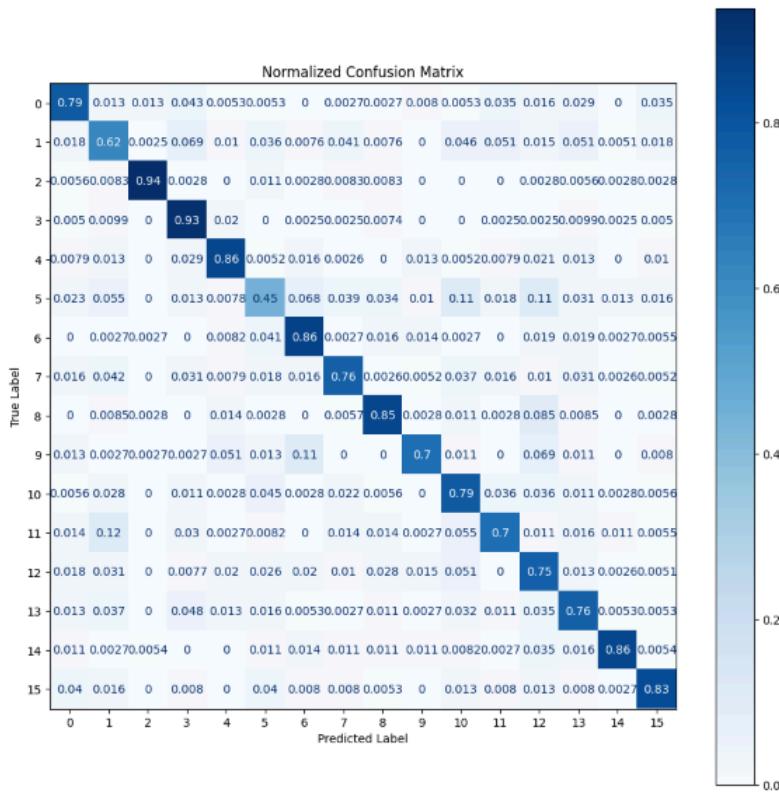
5.Performance du model

Exactitude de la Validation et du Test :

Les précisions de validation et de test sont d'environ **77,7 %**. Cela indique que le modèle fonctionne de manière cohérente sur les ensembles de validation et de test, ce qui suggère que le modèle se généralise bien aux données non vues. Perte de Validation et de Test :

La perte de validation et de test est identique à 1.0063, ce qui renforce l'idée que la performance du modèle est stable.

La Matrice de Confusion



Analyse de la Matrice de Confusion :

Performance par Classe :

Certaines classes sont très bien reconnues par le modèle, comme les classes 0, 2, 3, 6, 8, 11, 12, 13 et 15 qui ont des valeurs élevées sur la diagonale, indiquant une bonne précision.

D'autres classes comme les classes 1 (questionnaire) et 5 (rapport scientifiques) montrent une performance plus faible sûrement lié à la mise en page et à la présence d'éléments similaires à d'autres classes comme les budgets ou les présentations.

Confusions Notables :

Les classes 5 et 10 montrent des confusions significatives avec plusieurs autres classes. La classe 7 est souvent confondue avec les classes 0 et 3. La classe 13 est fréquemment confondue avec les classes 0, 3 et 7.

6.Conclusion

Le modèle montre une bonne performance globale avec une précision stable sur les ensembles de validation et de test. Cependant, il existe deux classes sous performantes pourrait être améliorée en réduisant la confusion.

7.Améliorations Potentielles :

Pour améliorer la performance globale, il serait bénéfique de se concentrer sur les classes avec des taux de confusion élevés. Pour cela on pourrait :

- **Augmentation des données :** Augmenter la quantité de données d'entraînement pour que le modèle puisse mieux appréhender les différentes classes.
- **Augmentation de la période d'apprentissage :** Augmenter le nombre d'époques d'apprentissage afin de réduire le coût et améliorer la précision. Cependant on augmente le risque d'overfitting et il n'est pas sûr que cela augmente la précision sur les classes 1 et 5
- **Complexifier le modèle linéaire :** Ajouter des couches supplémentaires dans la partie linéaire afin de supprimer certains éléments qui défavoriseraient les classes 1 et 5
- **Complexifier le modèle neuronale:** Ajouter une couche neuronale supplémentaire afin de faire ressortir plus d'informations sur les classes avec une prévision en dessous de la précision globale
- **Changer pour un modèle plus complexe :** Passer à des modèles EfficientNet plus complexes et plus coûteux en entraînement qui font ressortir plus d'informations

- Modèle 4: Amélioration du modèle EfficientNet B0

1.Introduction:

EfficientNet-B0 possède de bonnes performances sur le dataset cependant la classe 1 et la classe 5 sont mal détectées et souvent confondues respectivement avec la classe 3 et la classe 10 et 12.

On souhaite iterer sur le modèle B0 en ajoutant une couche neuronale supplémentaire ou bien en améliorant le classificateur.

2.Prétraitement et Initialisation

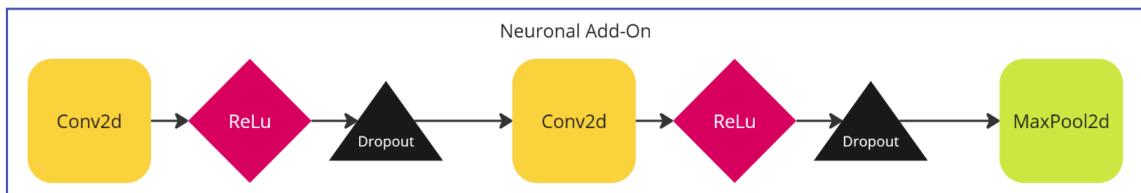
Le modèle que nous utilisons est pré-entraîné sur le dataset ImageNet :

- Neuronal Add_On : On ajoute dans une couche neuronale supplémentaires, tout en conservant les poids initiaux sur les couches précédentes et on entraîne ce nouveau modèle afin d'accroître les détails différentiant pour les classes faiblement reconnues
- Linear Add-On : On modifie le classificateur avec une couche linéaire plus complexe pour mieux faire ressortir les différences entre les différentes classes

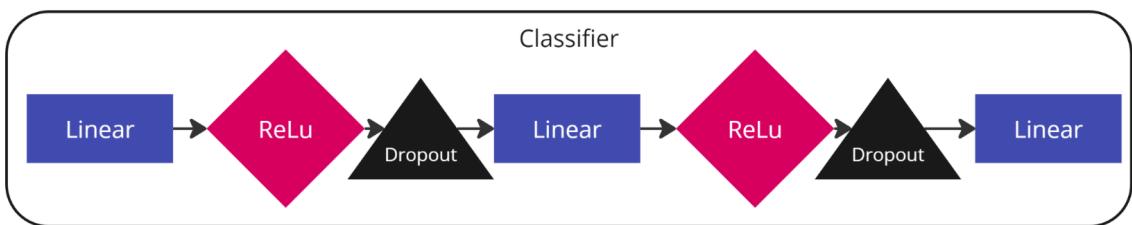
3.Détails Techniques

- **Architecture de Base :**

- Neuronal Add-On : On ajoute une dernière couche neuronale sous cette forme pour accroître les différences et donc le nombre de paramètres en sorties.



- Linear Add-On : On modifie le classificateur afin d'avoir un modèle linéaire plus progressif tout en réduisant les éléments peu différents



- **Stratégie d'Initialisation :** Les poids initiaux sont tirés des poids pré-entraînés sur ImageNet, spécifiés par `models.EfficientNet_B0_Weights.DEFAULT`.

4.Avantages de modifier l'EfficientNet-B0

Efficacité : On conserve la rapidité d'apprentissage du premier que l'on dégrade légèrement en ajoutant ces modifications. Cependant, on reste sur des temps d'apprentissage moins importants que l'entraînement de base d'un modèle similaire.

Précision : L'ajout de ces éléments devrait modifier la précision du modèle dans une moindre mesure. On ne risque pas ainsi de fortement dégrader la précision. Cependant les gains seront moins importants que de repartir d'un modèle de base.

5.Performance du modèle

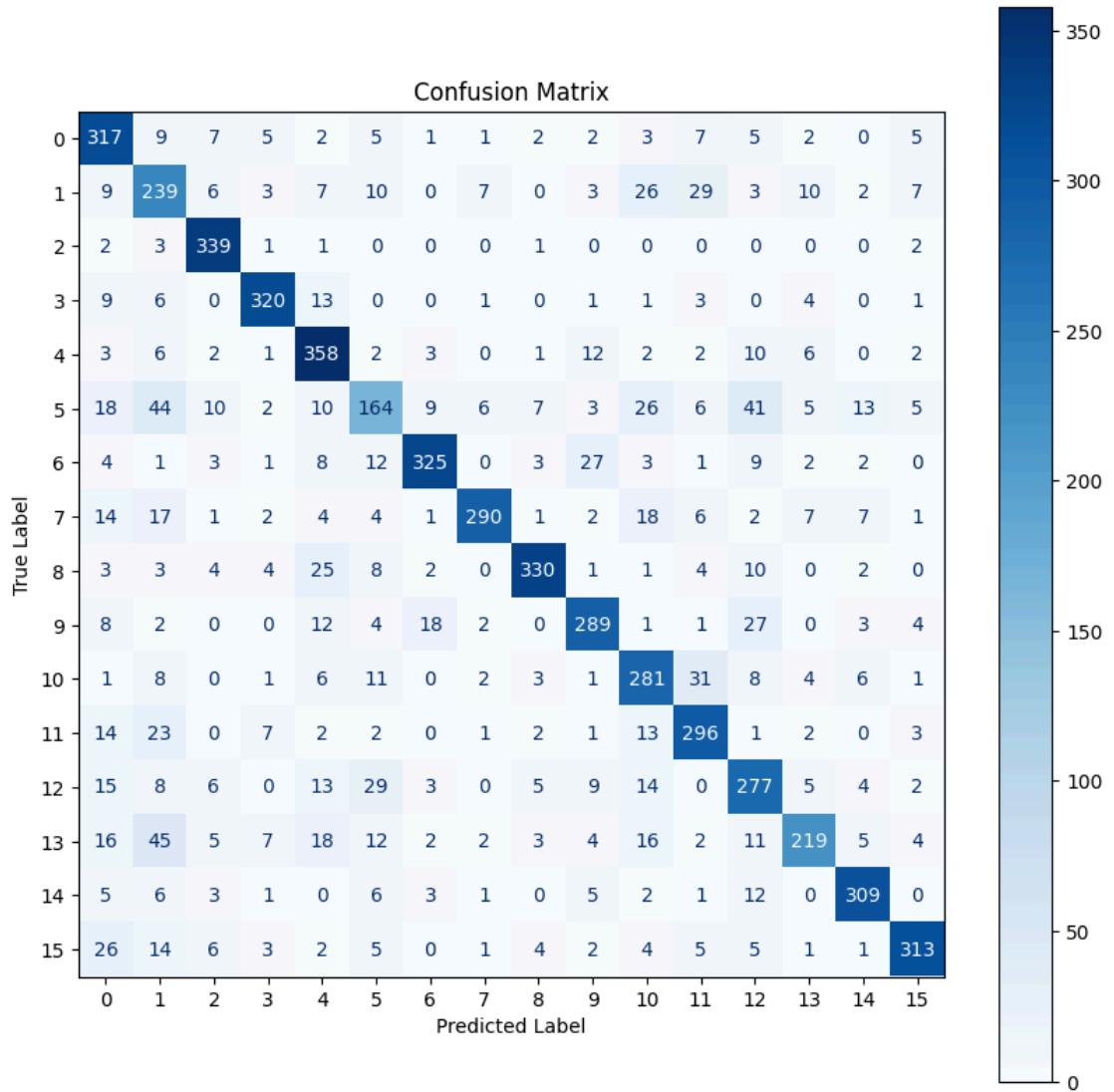
Exactitude de la Validation et du Test :

- Neuronal Add-On : **77,9%** de précision
- Linear Add-On : **78,2%** de précision

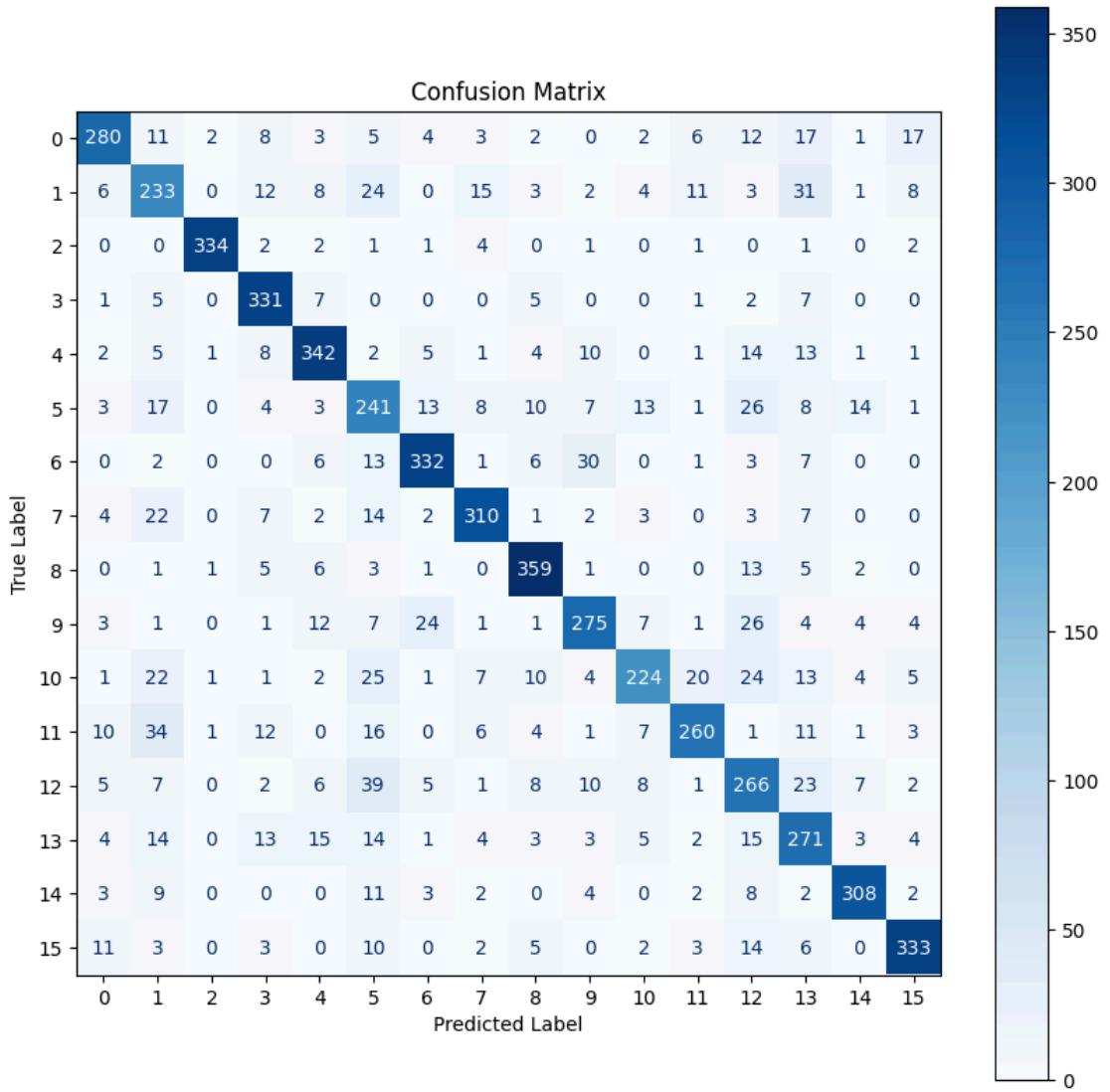
Le modèle Neuronal Add-On ne semble pas accroître significativement la précision malgré un doublement du temps d'apprentissage. De même, le modèle Linear Add-On fait légèrement mieux sans être suffisamment différentiant. Cependant le temps d'apprentissage est similaire au modèle EfficientNet B0.

La Matrice de Confusion

- Neuronal Add-On



- Linear Add-On



Analyse de la Matrice de Confusion :

Performance par Classe :

Le NAO (Neuronal Add-On) accroît la confusion autour de la classe 5 (rapport scientifique) et dans une moindre mesure sur la classe 13 (questionnaires).

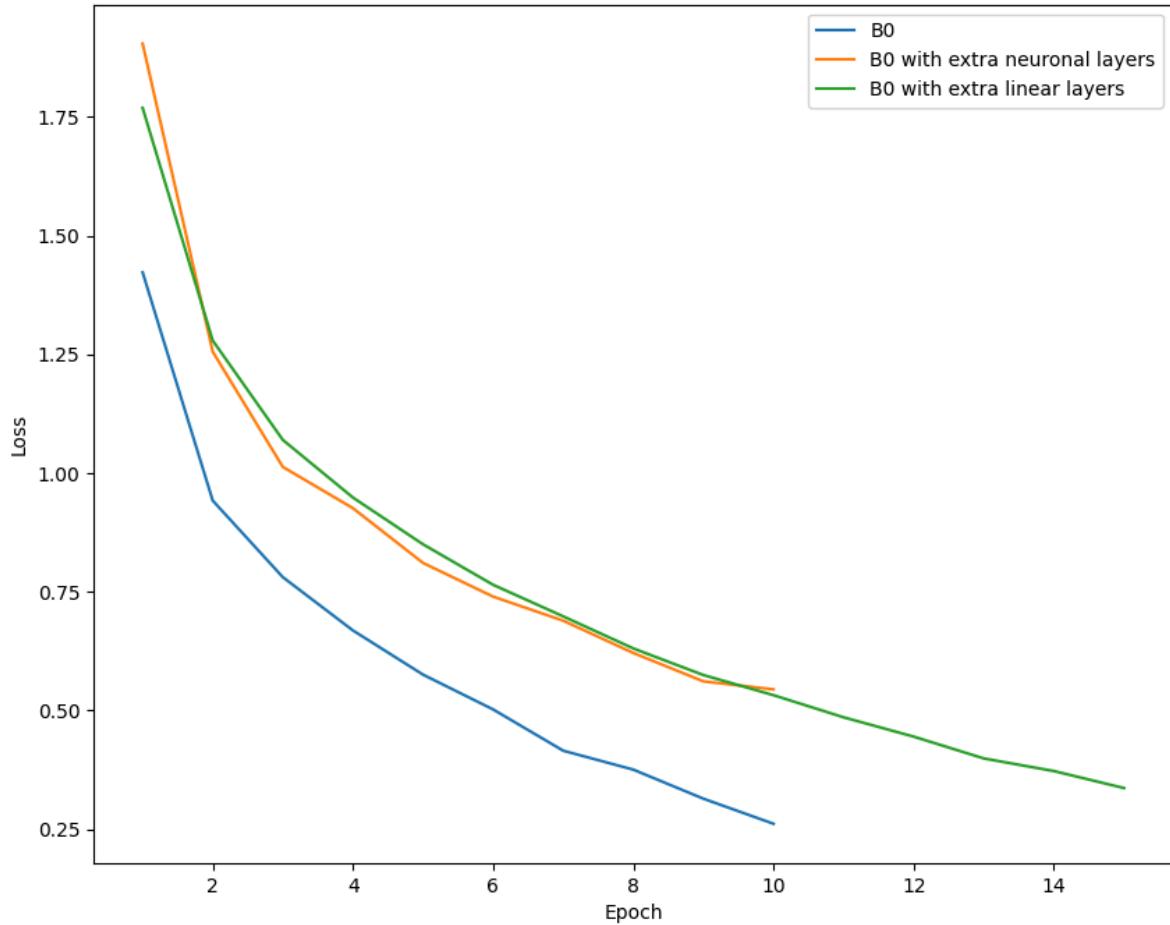
Ainsi le modèle accroît légèrement sa précision au profit de l'abandon de la classe 5 qui ne se différencie pas assez.

Le modèle LAO (Linear Add-On) semble lui au contraire lisser les classes et réduire les disparités présentes entre les classes.

6.Conclusion

Le modèle est légèrement plus performant avec l'ajout d'une couche linéaire. Cependant l'ajout d'une couche neuronale supplémentaire ne renforce que les disparités. De plus ces modèles demandent plus de temps d'entraînement car pour un nombre d'époque équivalente, le modèle n'a pas réduit suffisamment son coût.

Evolution du coût pour les modèles EfficientNet



7. Améliorations Potentielles :

Pour améliorer la performance globale, on peut soit augmenter la durée d'entraînement ce qui pourrait entraîner un overfitting, la taille du dataset en augmentant la part des classes mal évaluées.

On explore par la suite une autre version du modèle EfficientNet plus coûteuse en temps d'apprentissage mais qui semble plus efficace

- Modèle 3: Modèle Pré-entraîné EfficientNet B1

1. Introduction:

EfficientNet-B1 est un modèle de réseau neuronal convolutif (CNN) dérivé d'EfficientNet-B0. Il est aussi pré-entraîné sur ImageNet, un dataset de plus d'un million d'image réparties en 1000 classes.

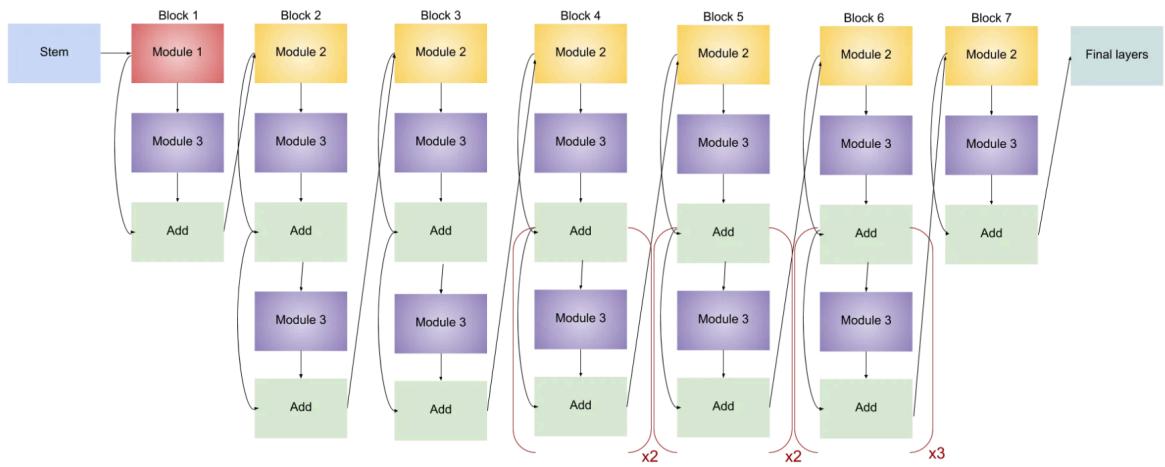
La version B1 est plus complexe et plus grande que la version B0 mais reste suffisamment rapide comparés aux modèles plus performants allant jusqu'à B7.

2. Prétraitement et Initialisation

L'utilisation de ces poids pré-entraînés accélère la convergence lors de l'entraînement sur notre propre dataset et améliore la performance du modèle.

3. Détails Techniques

- **Architecture de Base :** EfficientNet-B1 utilise un ensemble de blocs convolutifs similaire à EfficientNet-B0 avec quelques blocs convolutifs supplémentaires. Ainsi le modèle permet d'accroître le niveau de détail détecté et ainsi il peut prendre en charge des images plus grandes.



- Stratégie d'Initialisation :** Les poids initiaux sont tirés des poids pré-entraînés sur ImageNet, spécifiés par `models.EfficientNet_B0_Weights.DEFAULT`.
- Modification du Classificateur:** Pour adapter EfficientNet-B1 à notre tâche spécifique de classification, nous avons modifié la dernière couche du classificateur. Par défaut, la couche de sortie du modèle pré-entraîné est conçue pour classer les images en 1000 catégories (classes d'ImageNet). Nous avons remplacé cette couche par une nouvelle couche linéaire adaptée au nombre de classes de notre problème (16 classes)

4. Avantages de l'EfficientNet-B1

Efficacité : L'un des principaux avantages d'EfficientNet-B1 est sa plus grande précision par rapport à EfficientNet-B0 sans pour autant augmenter de manière considérable le temps de calcul.

Précision : Une meilleure précision par rapport à EfficientNet-B0.

5. Performance du modèle

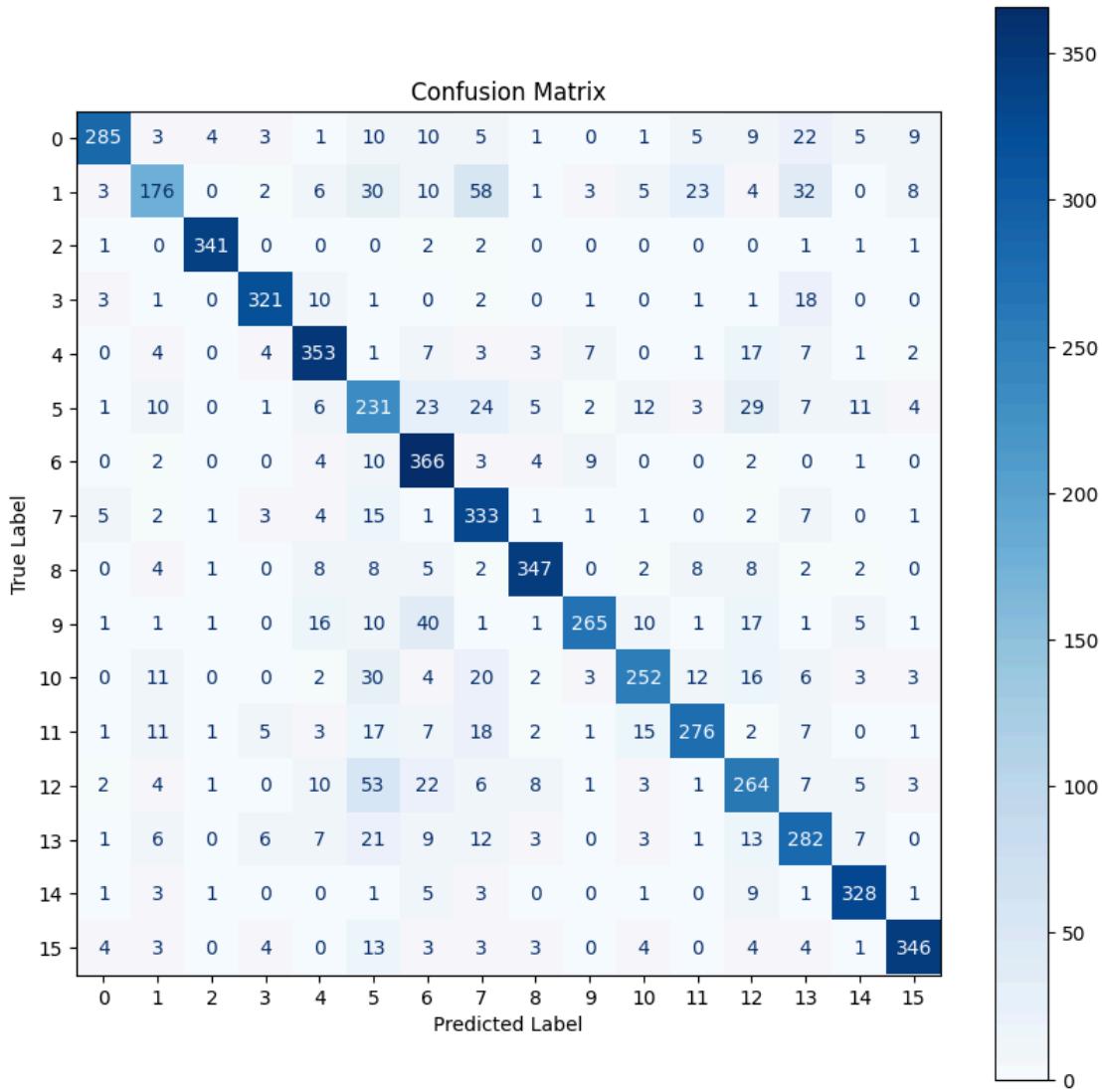
Exactitude de la Validation et du Test :

Les précisions de validation et de test sont d'environ **79,6 %**. Cela indique que le modèle fonctionne de manière cohérente sur les ensembles de validation et de test, ce qui suggère que le modèle se généralise bien aux données non vues.

Perte de Validation et de Test :

La perte de validation et de test sont de 0.93 et 0.91, ce qui renforce l'idée que la performance du modèle est stable. De plus, le coût est réduit par rapport à EfficientNet-B0.

La Matrice de Confusion



Analyse de la Matrice de Confusion :

Performance par Classe :

Les classes 1 (questionnaire) et 5 (rapport scientifiques) montrent une performance plus faible sûrement lié à la mise en page et à la présence d'éléments similaires à d'autres classes comme les budgets ou les présentations. Le modèle rest ainsi plus performant là où EfficientNet-B0 était déjà performant mais il n'arrive pas à compenser les faiblesses de B0 sur la classe 1 et 5.

Confusions Notables :

Les classes 1 et 5 se confondent avec la classe 7. montrent des confusions significatives avec plusieurs autres classes. La classe 6 et 7 sont performantes mais elles réduisent le recall car d'autres classes sont détectées dans celle-ci.

6.Conclusion

Le modèle montre une bonne performance globale avec une précision stable sur les ensembles de validation et de test. Cependant, il percute deux classes sous performantes qui pourraient être améliorées en augmentant le dataset d'entraînement.

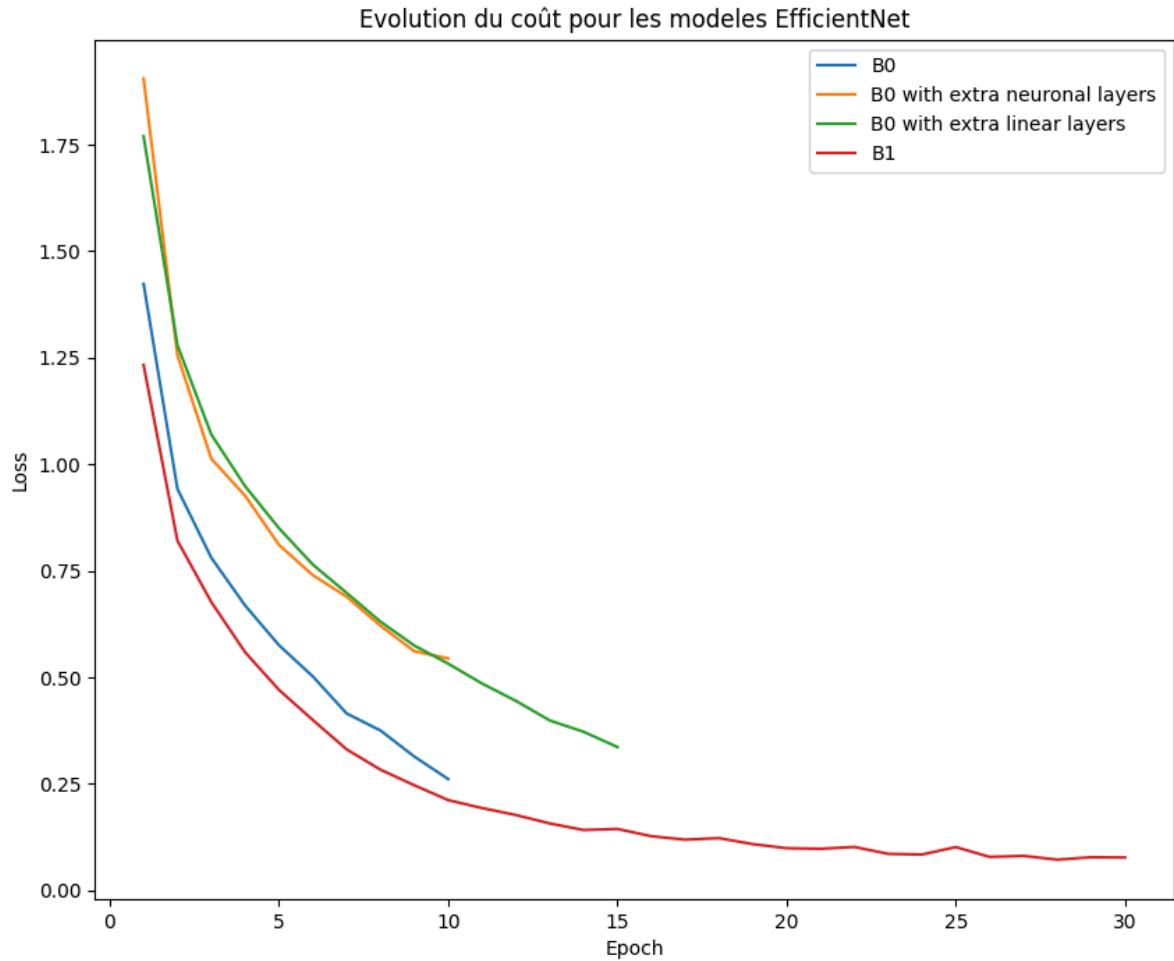
7.Améliorations Potentielles :

Pour améliorer la performance globale, il serait bénéfique de se concentrer sur les classes avec des taux de confusion élevés. Pour cela on pourrait :

- **Augmentation des données :** Augmenter la quantité de données d'entraînement pour que le modèle puisse mieux apprécier les différentes classes.
- **Augmentation de la période d'apprentissage :** Augmenter le nombre d'époques d'apprentissage afin de réduire le coût et améliorer la précision. Cependant on augmente le risque d'overfitting et il n'est pas sûr que cela augmente la précision sur les classes 1 et 5.
- **Changer pour un modèle plus complexe :** Passer sur les modèles B2 à B7. Cependant ils consomment beaucoup de ressources, RAM, GPU et temps d'apprentissage.

4.2 CNN retenu

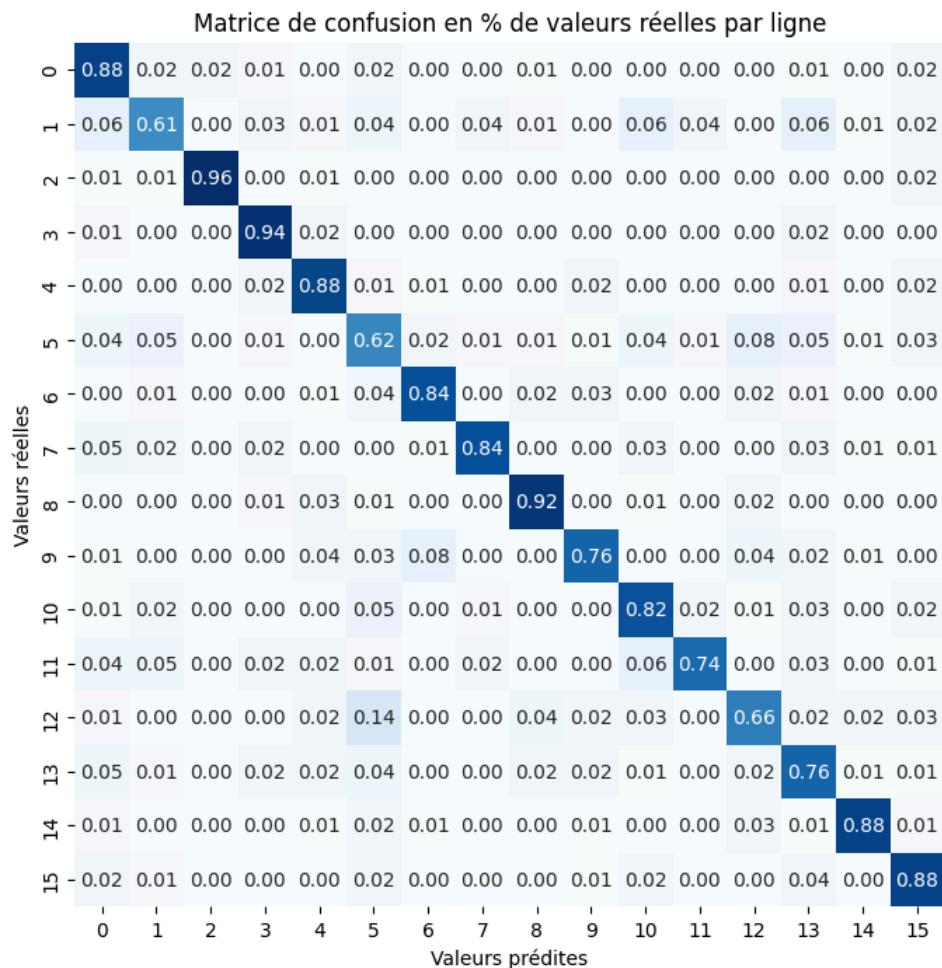
Après avoir comparé les performances des différents modèles, nous avons choisi le Modèle Pré-entraîné EfficientNet-B1 pour notre classification. Ce modèle a montré une bonne performance de sur l'ensemble d'entraînement. De plus, c'est le modèle qui a su réduire le plus efficacement le coût.



4.3 Présentation des résultats obtenus

Le modèle choisi est EfficientNet B1. Il présente une meilleure précision que le modèle B0 tout en possédant un temps d'entraînement raisonnable, avec une augmentation de 20% du temps de calcul.

On entraîne le modèle choisi sur 16000 images reparties uniformément sur les 16 classes



Voici les performances du modèle :

- CNN accuracy: 0.8192
- CNN F1 Score: 0.8196

On conserve des lacunes dans la détection des classes 1, 5 et 12 représentant respectivement des formulaires, des rapports scientifiques et des présentations.

4.4 Interprétabilité

1. Introduction:

Grad-CAM, ou Gradient-weighted Class Activation Mapping, est une technique utilisée pour fournir des explications visuelles des prédictions faites par les réseaux de neurones convolutifs (CNN). Cette méthode met en évidence les régions de l'image d'entrée qui sont les plus pertinentes pour la décision du modèle. Grad-CAM produit une carte de localisation grossière des régions importantes de l'image. Cela aide à comprendre et à interpréter le comportement du modèle, ce qui est crucial dans les applications où la transparence du modèle est nécessaire.

L'objectif est de fournir une interface interactive permettant de sélectionner des images et des couches spécifiques d'un modèle CNN pour observer les heatmaps générées par Grad-CAM, facilitant ainsi l'interprétation des décisions du modèle.

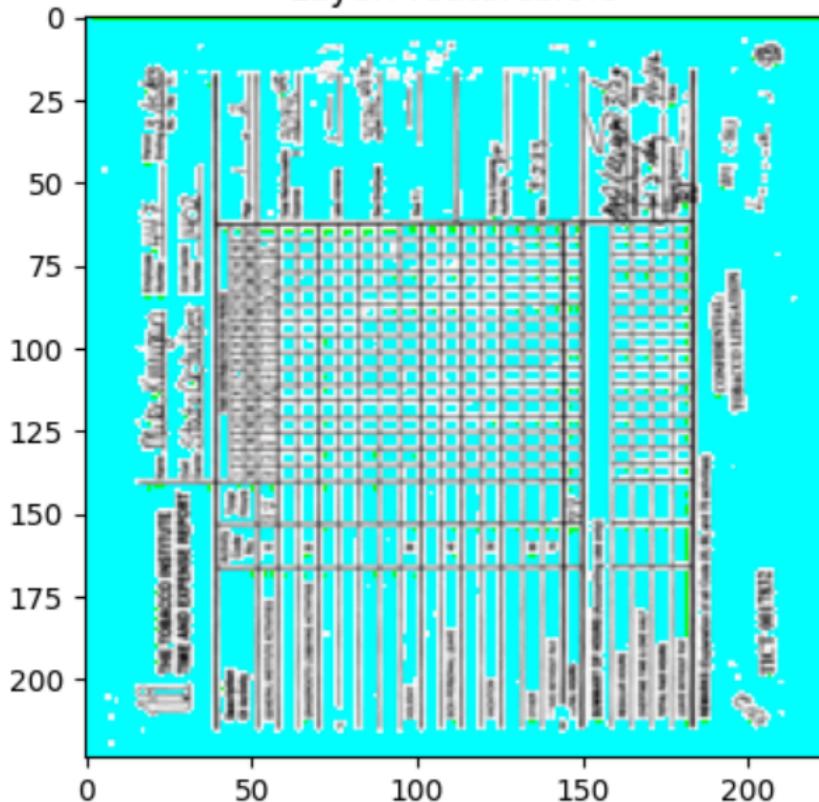
2. Analyse :

Dans cette section, nous présentons les visualisations Grad-CAM obtenues pour une image spécifique, à travers plusieurs couches du modèle EfficientNet. Les visualisations montrent comment les différentes couches du modèle mettent en évidence diverses parties de l'image, indiquant les zones qui influencent le plus les décisions du modèle.

• Visualisation 1 :

La première couche convective **features.0.0** se concentre principalement sur les contours et les bords de l'image. Les régions en surbrillance indiquent les zones où les gradients sont les plus forts, ce qui signifie que ces zones sont importantes pour les premières étapes du traitement de l'image par le modèle. On peut voir que les bords et les lignes principales du document sont bien capturés, ce qui est crucial pour les étapes ultérieures du traitement de l'image.

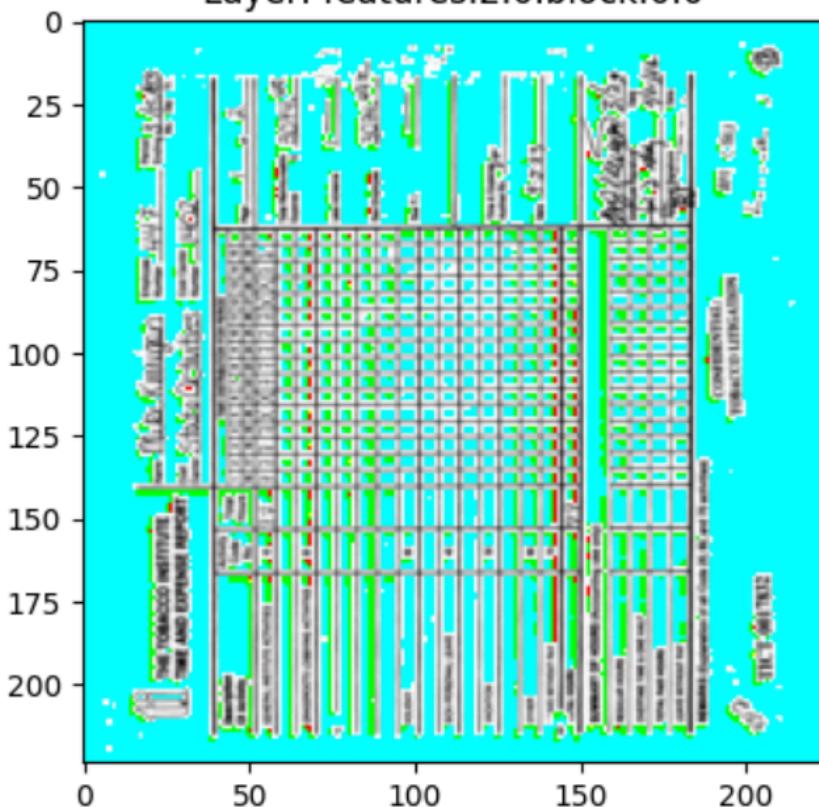
Grad-CAM Layer: features.0.0



- **Visualisation 2 :**

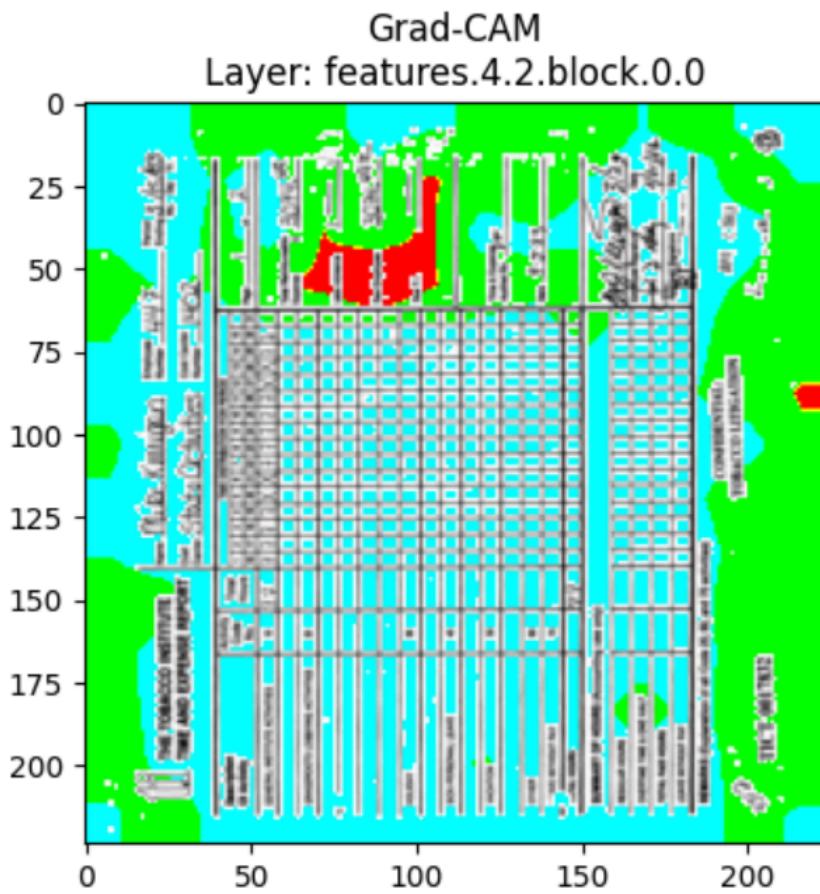
La couche **features.2.0.block.0.0**, le modèle commence à capturer des motifs plus complexes. Les régions en surbrillance incluent non seulement les bords, mais aussi des parties spécifiques du texte et des structures internes du document. Cette couche est capable de détecter des caractéristiques plus élaborées, telles que les lignes de tableau et certains éléments textuels importants.

Grad-CAM Layer: features.2.0.block.0.0



- **Visualisation 3 :**

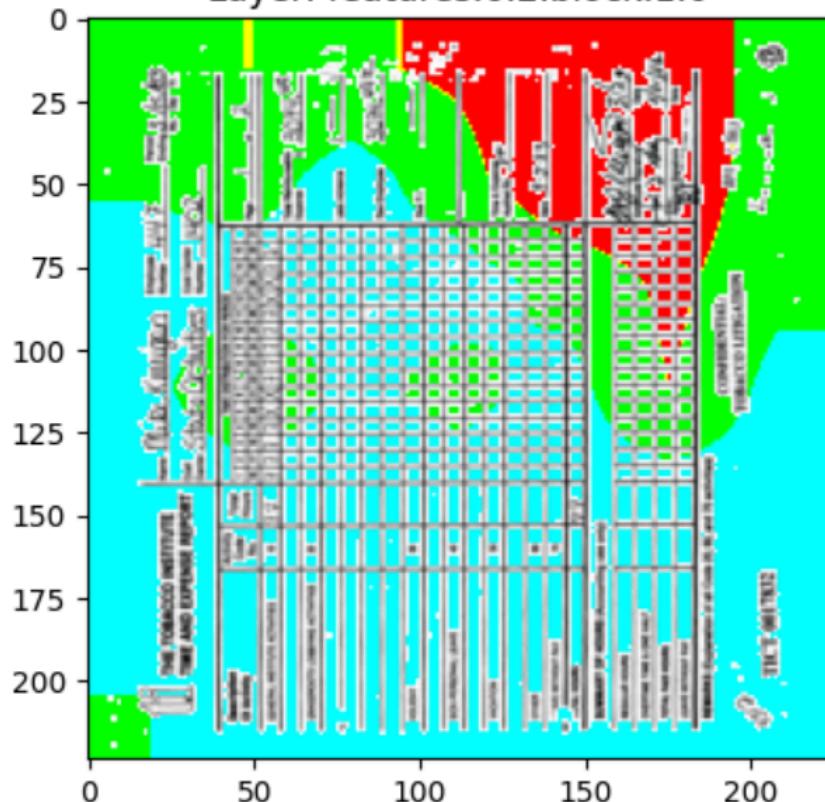
La couche **features.4.2.block.0.0** montre une concentration plus fine sur les détails spécifiques du document. Les surbrillances sont plus dispersées et se concentrent sur des zones textuelles et des parties du tableau. Cela indique que le modèle commence à identifier les structures textuelles et les informations spécifiques contenues dans les cellules du tableau.



- **Visualisation 4 :**

La couche **features.6.2.block.1.0**, le modèle se concentre sur des régions plus spécifiques et significatives du document. Les surbrillances indiquent que le modèle accorde une attention particulière aux informations textuelles importantes et aux lignes du tableau qui peuvent être cruciales pour la classification. Les zones colorées montrent des détails textuels qui sont essentiels pour la prise de décision du modèle.

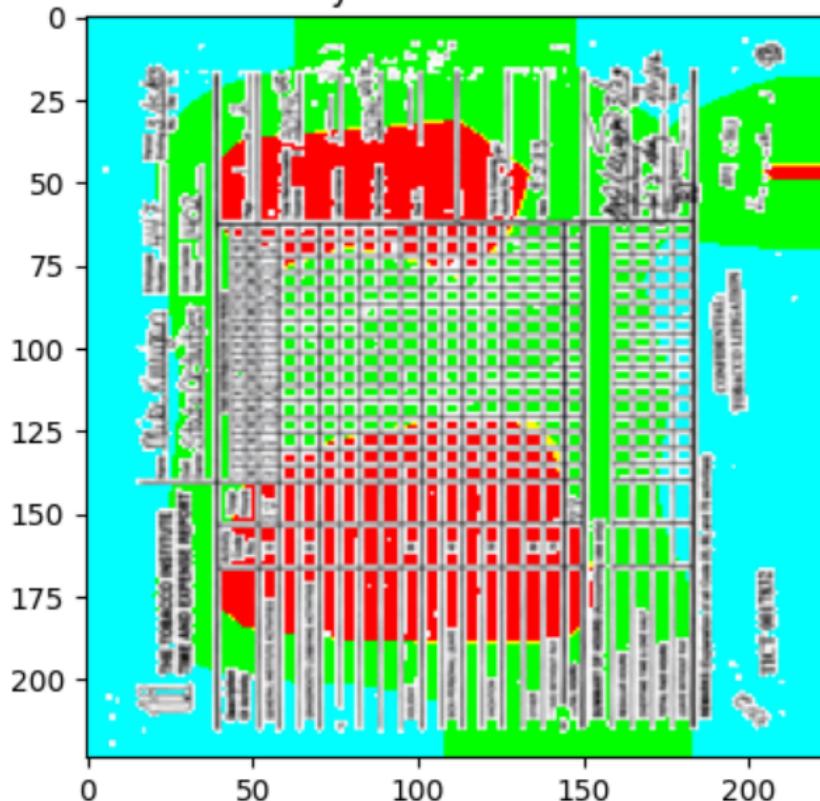
Grad-CAM Layer: features.6.2.block.1.0



- **Visualisation 5 :**

Enfin, la couche **features.8.0** met en évidence les régions les plus importantes de l'image qui influencent directement la prédiction finale du modèle. Les surbrillances intenses sur certaines parties du texte et des structures spécifiques du tableau montrent que ces éléments ont un poids significatif dans la décision finale. Cela indique que le modèle utilise des informations à haut niveau pour effectuer une classification précise.

Grad-CAM Layer: features.8.0

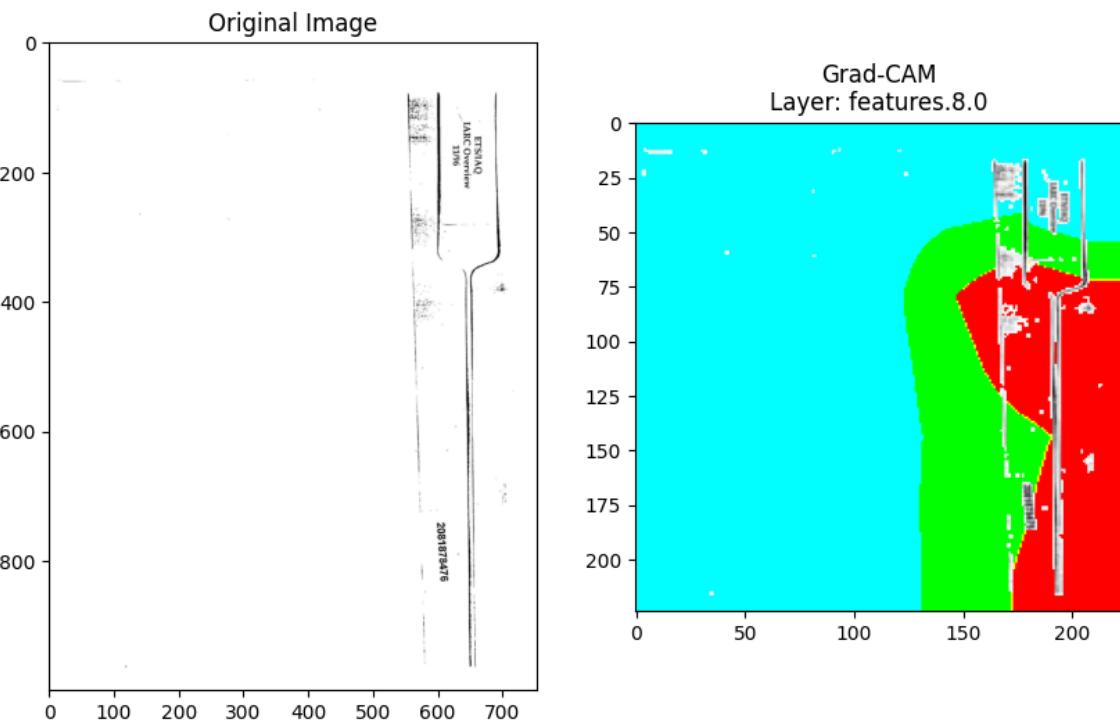


3. Interprétation :

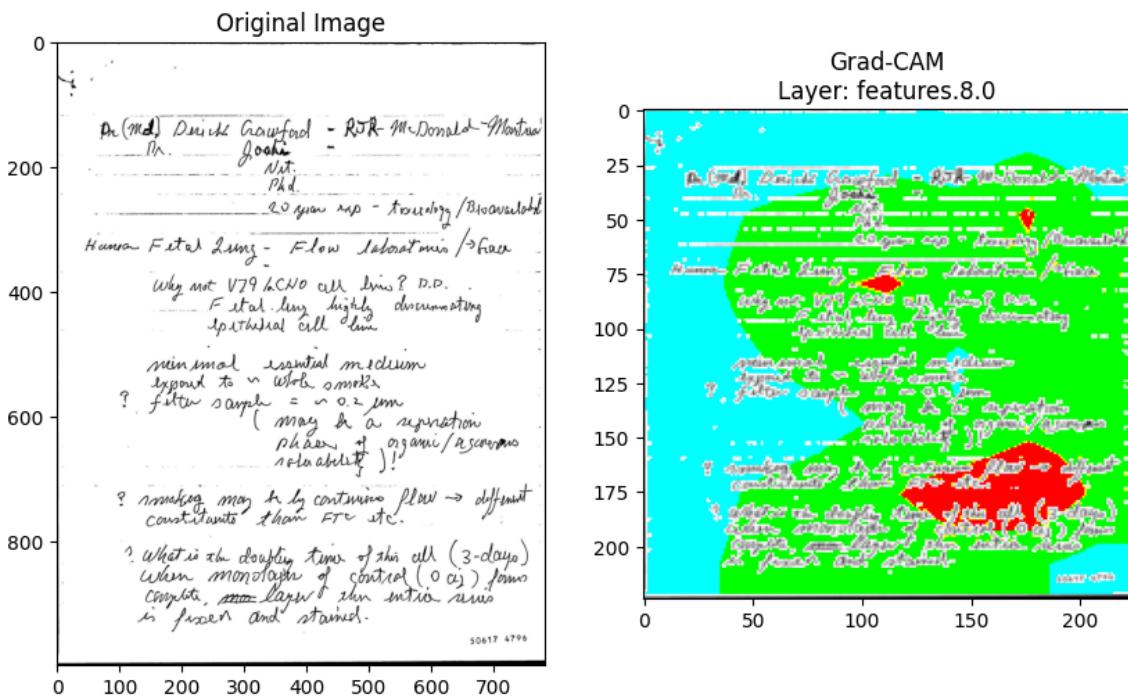
- La visualisation de la carte thermique montre les zones de l'image originale qui sont importantes pour la prédiction du CNN. Les intensités de couleur dans la carte Grad-CAM représentent le niveau d'importance, les couleurs chaudes (rouges et jaunes) indiquant les régions ayant un impact plus élevé sur la prédiction.
- Les régions avec des couleurs chaudes, comme le coin supérieur droit et quelques zones dispersées, sont les parties de l'image sur lesquelles le modèle CNN s'est concentré pour sa prédiction.
- Les régions plus froides (bleus et verts) sont des zones moins significatives pour le processus de décision du modèle.

On remarque que les images bien détectées tels que les mails, les publicités, les dossiers et les notes manuscrites sont assez différenciées des autres.

- Le dossier est détecté par l'absence de texte et la forme du dossier

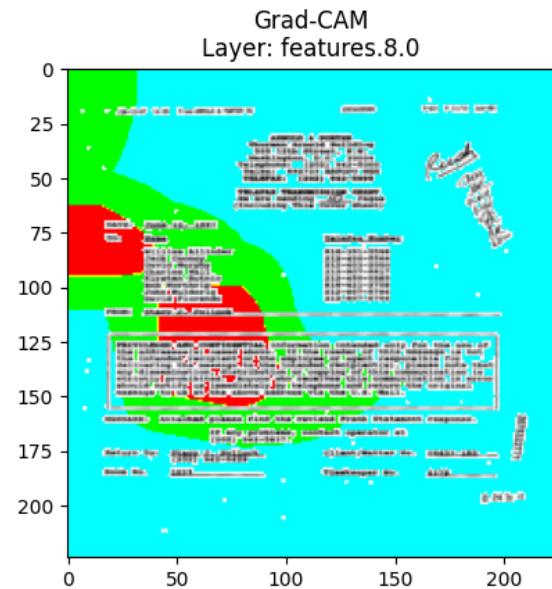
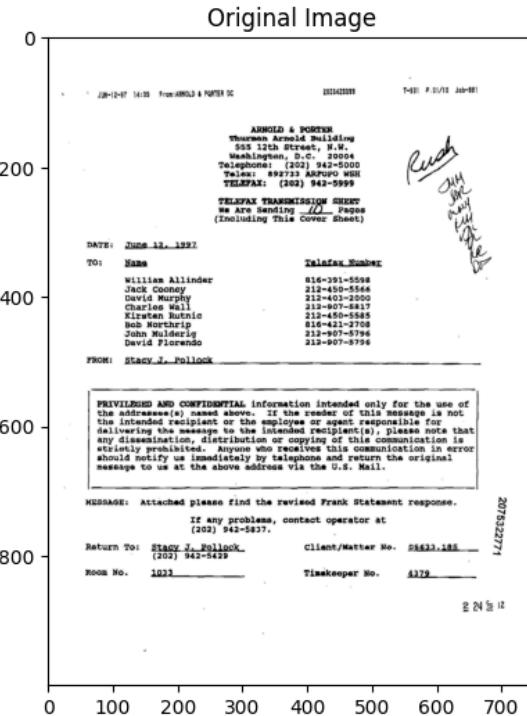


- Les notes manuscrites sont détectées grâce à la présence de texte assez espacé en cursives

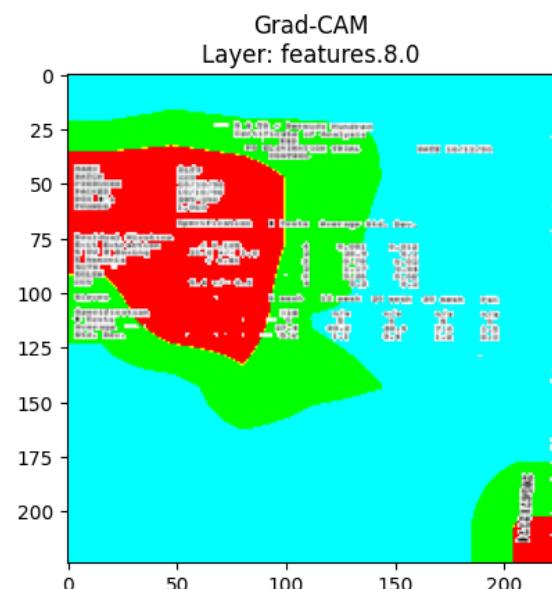
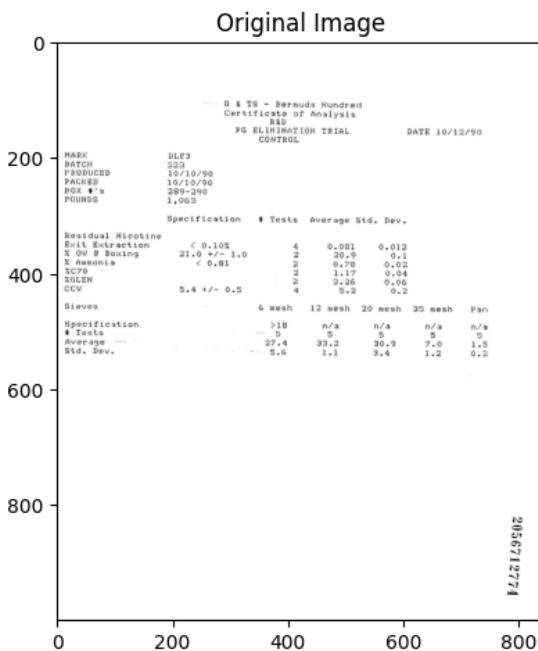


A l'inverse les classes 1 et 5 se confondent avec d'autres classes car leurs structures ne sont pas assez différentiantes

- Un formulaire prédit comme un mémo à cause de la présence de paragraphes conscrits



- Un rapport scientifique predit comme un budget du fait de la présence de blocs de chiffres



Limites des analyses visuelles :

Les analyses visuelles, telles que celles fournies par Grad-CAM, sont très efficaces pour comprendre et interpréter les décisions des modèles de reconnaissance d'images. Cependant, ces méthodes montrent leurs limites lorsque les documents sont visuellement similaires.

- Similarité Visuelle :

Lorsque des documents sont visuellement très similaires, il devient difficile pour les modèles basés uniquement sur l'image de les discriminer. On retrouve le cas avec des documents ayant des structures similaires, notamment avec des paragraphes de textes ou bien des tableaux numériques

- Complexité du Contenu :

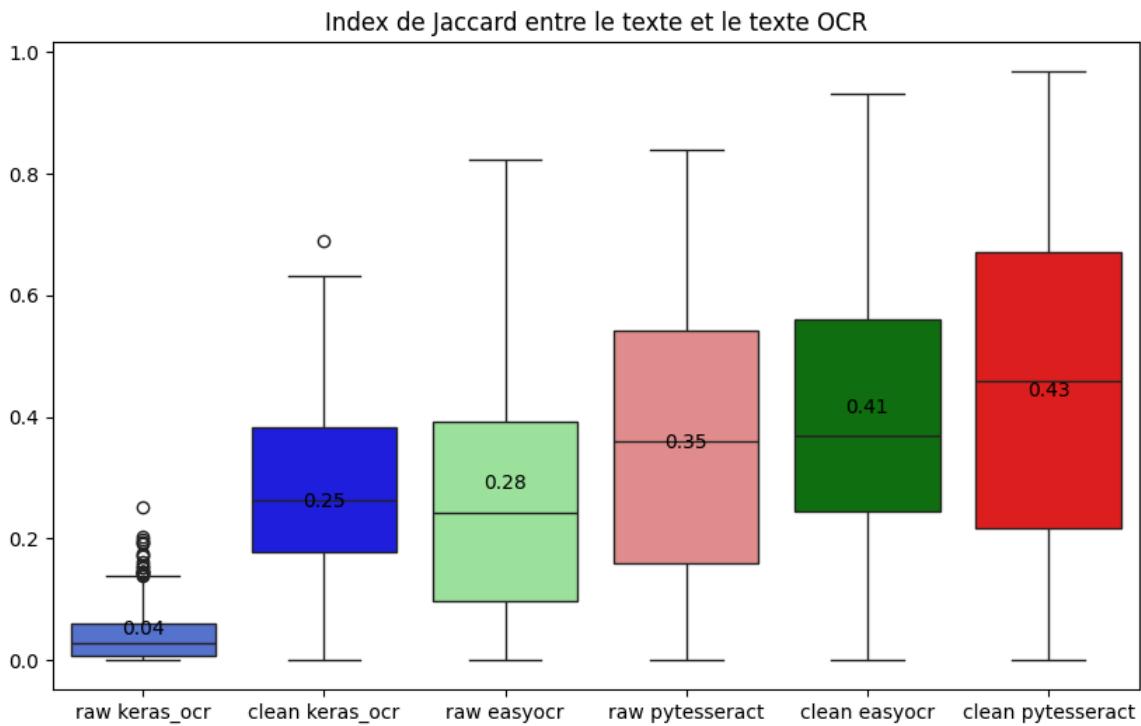
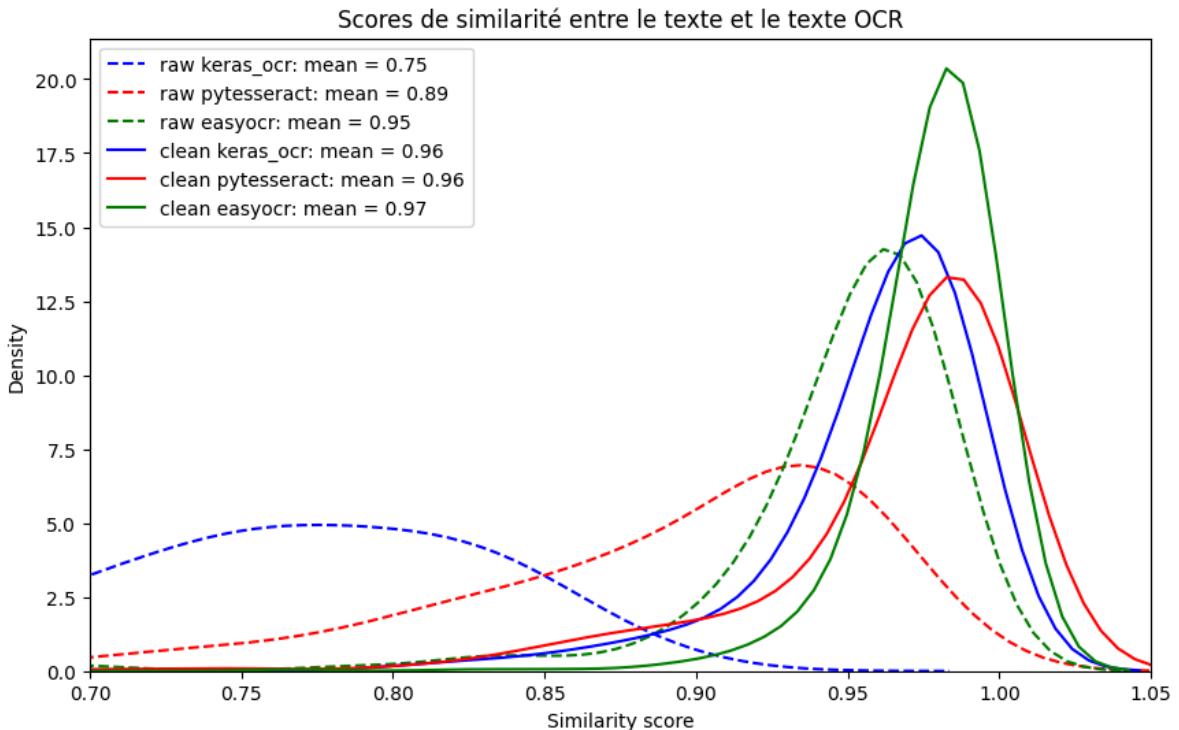
Les documents peuvent contenir des informations complexes et denses en texte qui ne peuvent pas être complètement capturées par des modèles basés uniquement sur des images. Ainsi, le CNN ne peut interpréter le contenu et donc contextualiser le document. Pour renforcer le modèle basé sur la détection d'image, on va le comparer à un modèle basé sur une interprétation textuelle

5. Choix de la librairie OCR

5.1 Présentation de la librairie OCR choisie et justification

Pour l'extraction du texte à partir des documents scannés, nous avons choisi d'utiliser Pytesseract. Cette librairie a été choisie parmi les librairies Pytesseract, Keras-ocr et easyocr pour plusieurs raisons:

- Il s'agit actuellement du standard de l'industrie.
- Une batterie de test a été faite sur un autre jeu de donnée dont le texte avait déjà été extrait et les résultats des différents OCR ont été comparés au texte réel.



A noter cependant plusieurs bémols:

- La librairie d'ocr a été benchmarkée en utilisant un jeu de donnée similaire au jeu de données principal. Ce jeu de données était associé à du texte lui-même issu d'un ocr. Ocr qui était parfois approximatif. Nous avons donc tenté de nous approcher d'un texte de

référence imparfait plutôt que sur le texte réel des documents.

- Après l'ocr et le nettoyage du texte, nous nous retrouvons comme pour le jeu de données du benchmark avec du texte inexacte dans de nombreux cas. Ce fait va forcément impacter les résultats des modèles de NLP.

5.2 Présentation des résultats de l'extraction

Après avoir utilisé Pytesseract sur le jeu de données, nous avons récupéré un ensemble de texte brut. Le préprocessing du texte extrait s'est fait en plusieurs étapes dont les principales sont :

- suppression des caractères spéciaux
- suppression des stop words
- suppression des mots qui n'existent pas
- lemmatization
- suppression des mots qui apparaissent moins de n fois dans tous le corpus de texte

A la fin du preprocessing nous nous retrouvons avec 20000 listes de mots qui serviront de base à l'entraînement des modèles de NLP.

6. Entraînement du modèle de NLP

6.1 Modèles testés et résultats obtenus

Plusieurs modèles de NLP ont été testés dans le cadre de ce projet. Chaque modèle a été évalué en fonction de son accuracy sur un ensemble d'entraînement de 20000 documents. On a décidé de se séparer rapidement des modèles Naive Bayes et Decision Tree qui avaient les moins bons résultats, pour se concentrer sur les modèles Logistic Regression, SVM and Random Forest pour optimiser les hyperparamètres sur un échantillon de 1000 documents cette fois-ci.

De plus des modèles basés sur la sémantique (word2vec, Bert et GPT) ont rapidement été écartés car ils ne produisaient pas de bons résultats. On suppose que cela vient du fait que les tokenizers n'étaient pas capables de travailler correctement avec les mots inconnus présents en grand nombre dans le texte à ce point.

Nous avons utilisés une approche bag of word et une approche tf-idf pour la vectorisation des textes. Ces deux approches sont interchangeables pour tous les modèles testés, la différence dans les métriques étant anecdotique. Nous avons utilisé l'approche tf-idf pour la vectorisation.

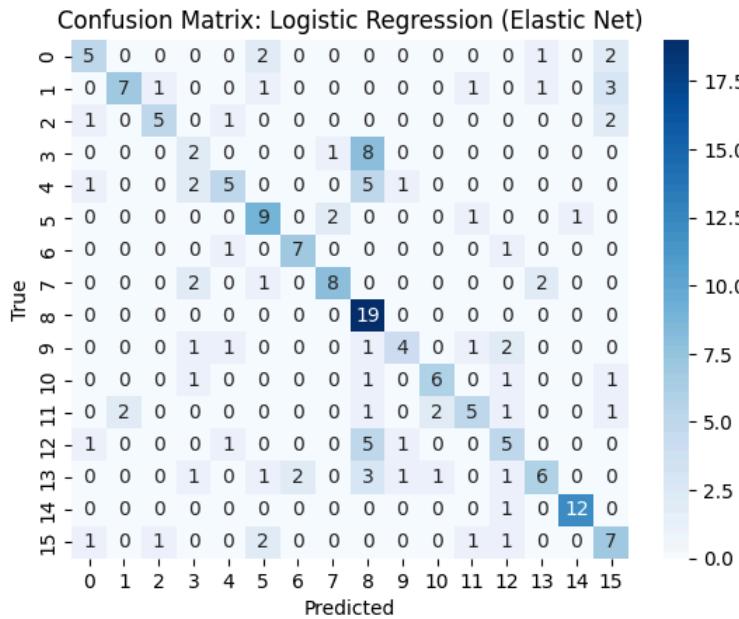
- Modèle 1 : Logistic Regression

On a testé les modèles de Logistic Regression à 3 niveaux : Lasso, Ridge et Elastic Net.

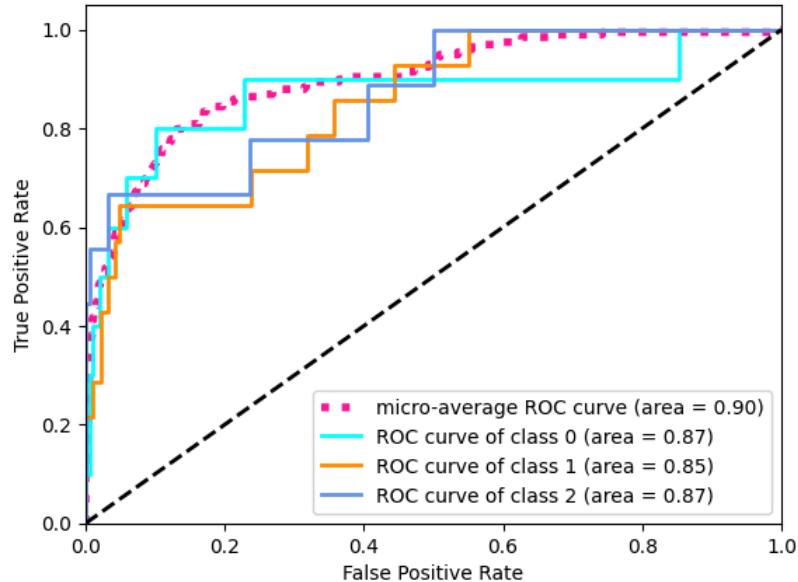
Les hyperparamètres optimaux sont :

- Ridge ("l2" penalty) : accuracy = 0.542714, F1 score = 0.529005, best parameters = {'C': 1, 'max_iter': 5000, 'penalty': 'l2', 'solver': 'saga', 'tol': 0.001}
- Lasso ("l1" penalty) : accuracy = 0.542714, F1 score = 0.53466, best parameters = {'C': 100, 'max_iter': 5000, 'penalty': 'l1', 'solver': 'saga', 'tol': 0.01}
- Elastic Net ("elasticnet" penalty) : accuracy = 0.557789, F1 score = 0.551118, best parameters = {'C': 100, 'l1_ratio': 0.1, 'max_iter': 5000, 'penalty': 'elasticnet', 'solver': 'saga', 'tol': 0.01}

Au vu des résultats, on décide de conserver le modèle Logistic Regression "Elastic Net". Pour les hyperparamètres optimaux, on peut partir de ceux-là "{C: 100, 'l1_ratio': 0.1, 'max_iter': 5000, 'penalty': 'elasticnet', 'solver': 'saga', 'tol': 0.01}" où retester un GridSearch sur un échantillon plus conséquent.

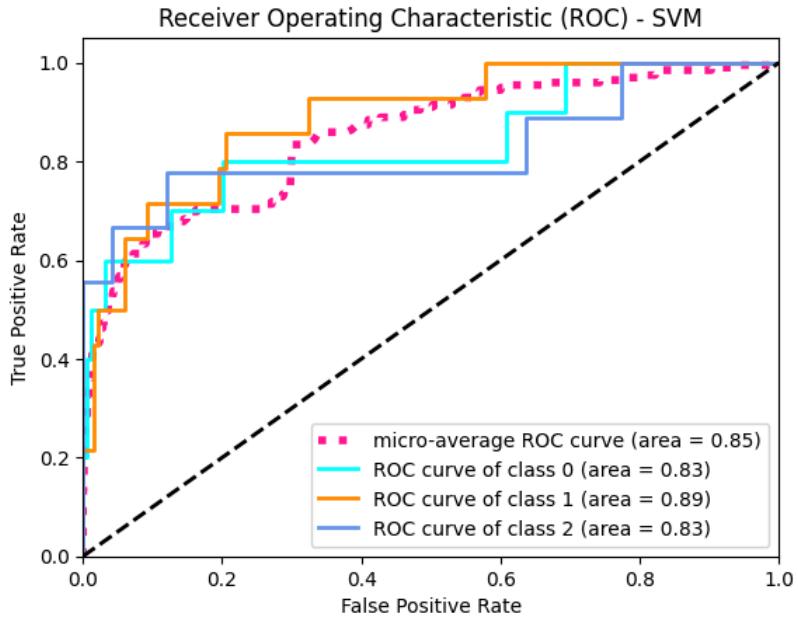
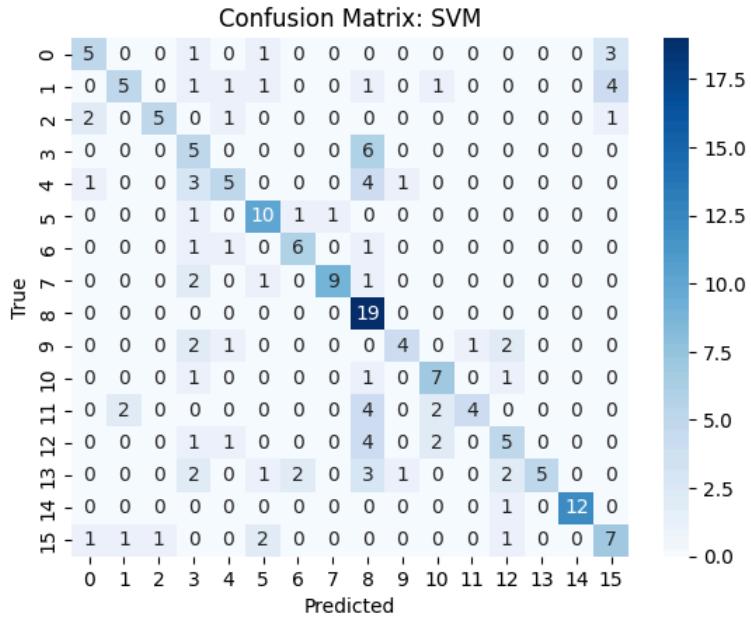


Receiver Operating Characteristic (ROC) - Logistic Regression (Elastic Net)



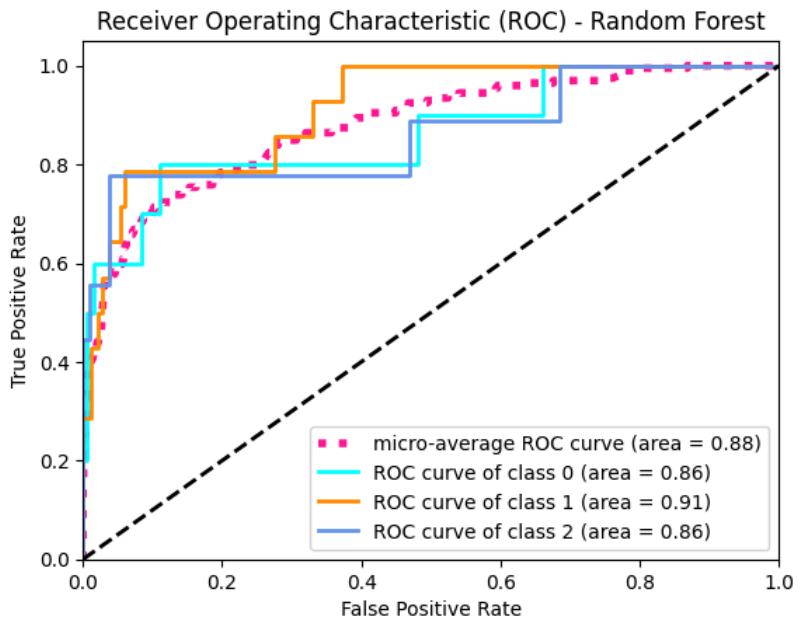
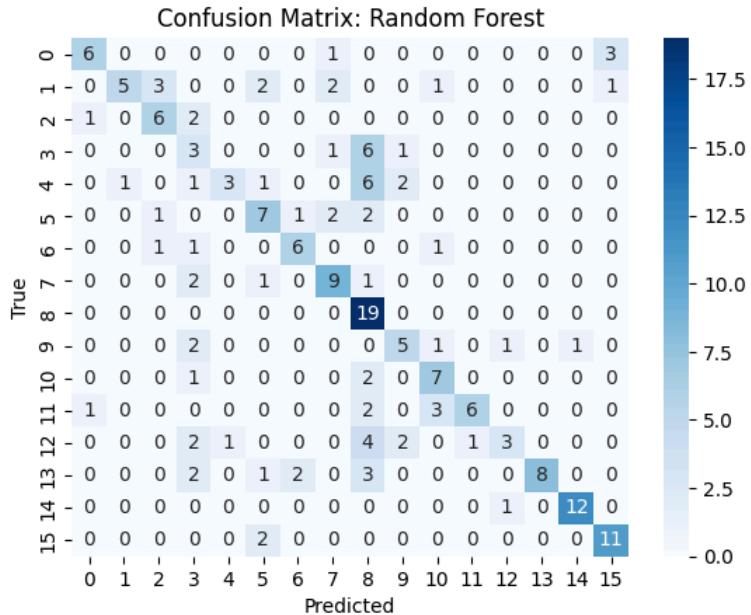
- Modèle 2 : SVM

Les hyperparamètres optimaux sont : accuracy = 0.567839, F1 score = 0.564915, best parameters = {'C': 1, 'degree': 2, 'gamma': 'scale', 'kernel': 'linear'}.

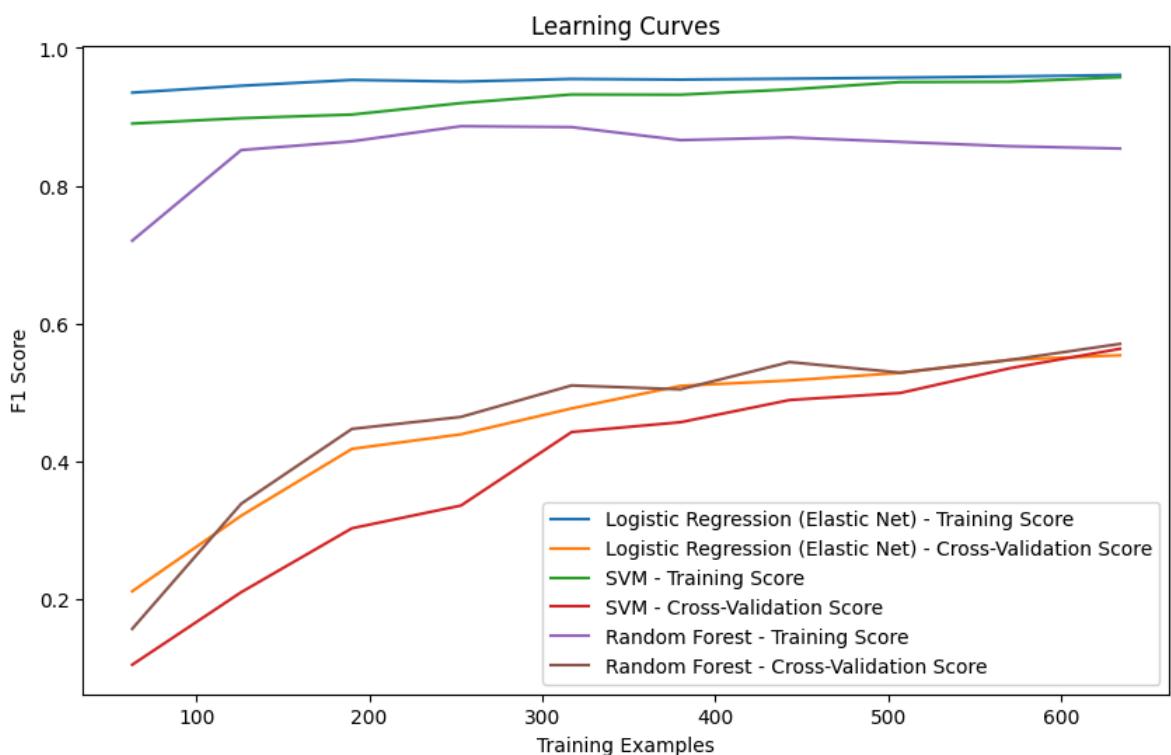
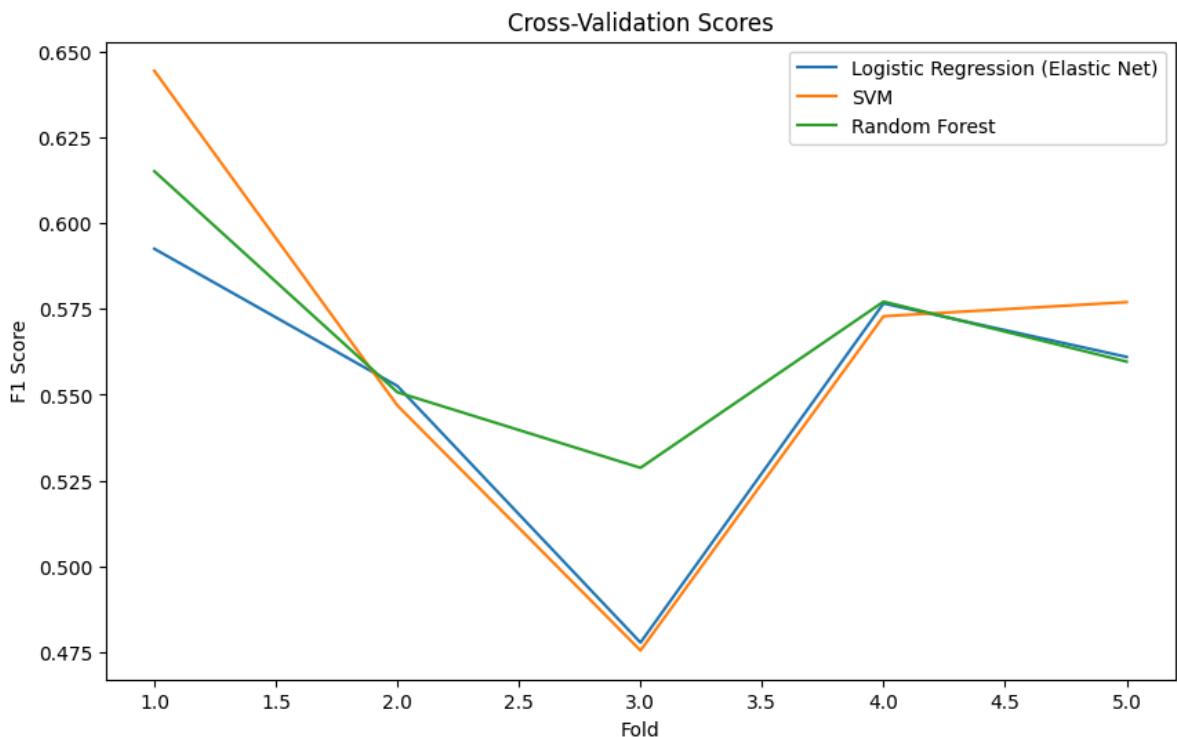


- Modèle 3 : Random Forest

Les hyperparamètres optimaux sont : accuracy = 0.572864, F1 score = 0.564937, best parameters = {'max_depth': 30, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 300}.



6.2 Modèle retenu et tuning des hyperparamètres



Après avoir comparé les performances des différents modèles, nous avons choisi le modèle truc pour notre système de classification. Ce modèle a montré une bonne performance de X sur l'ensemble d'entraînement.

Une fois le modèle sélectionné, nous avons tenté de l'optimiser de cette façon:

6.3 Présentation des résultats obtenus

Nous arrivons à un modèle final dont les résultats sont les suivants pour un jeu d'entraînement de 20000 textes cette fois :

- Accuracy : 0.67
- F1-Score : 0.68

Matrice de confusion en % de valeurs réelles par ligne

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0.61	0.02	0.03	0.03	0.00	0.04	0.02	0.01	0.04	0.02	0.01	0.01	0.03	0.01	0.00	0.12	
1	-0.02	0.53	0.02	0.05	0.01	0.09	0.01	0.03	0.06	0.00	0.04	0.06	0.01	0.02	0.00	0.04	
2	-0.02	0.02	0.78	0.00	0.00	0.00	0.00	0.00	0.04	0.04	0.02	0.00	0.05	0.00	0.00	0.03	
3	-0.01	0.02	0.00	0.56	0.02	0.00	0.00	0.00	0.31	0.00	0.02	0.01	0.00	0.02	0.00	0.00	
4	-0.01	0.00	0.00	0.09	0.46	0.00	0.00	0.00	0.33	0.05	0.01	0.01	0.03	0.00	0.00	0.01	
5	-0.01	0.01	0.00	0.07	0.01	0.56	0.12	0.02	0.09	0.01	0.03	0.02	0.03	0.00	0.01	0.02	
6	-0.00	0.00	0.00	0.02	0.00	0.04	0.78	0.01	0.04	0.06	0.00	0.00	0.03	0.00	0.01	0.00	
7	-0.00	0.03	0.00	0.05	0.02	0.02	0.00	0.84	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.01	
8	-0.00	0.00	0.01	0.04	0.00	0.01	0.00	0.00	0.88	0.00	0.01	0.00	0.01	0.00	0.00	0.03	
9	-0.00	0.00	0.00	0.08	0.03	0.01	0.03	0.00	0.08	0.65	0.02	0.00	0.08	0.01	0.00	0.00	
10	-0.00	0.00	0.00	0.07	0.02	0.03	0.00	0.00	0.16	0.01	0.61	0.05	0.04	0.00	0.00	0.01	
11	-0.01	0.03	0.00	0.06	0.01	0.01	0.00	0.00	0.08	0.00	0.08	0.72	0.00	0.00	0.00	0.00	
12	-0.02	0.02	0.01	0.04	0.02	0.04	0.02	0.01	0.22	0.11	0.02	0.01	0.43	0.01	0.01	0.01	
13	-0.02	0.03	0.00	0.08	0.03	0.01	0.00	0.00	0.05	0.01	0.01	0.02	0.02	0.69	0.00	0.02	
14	-0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.02	0.00	0.92	0.00	
15	-0.06	0.03	0.03	0.02	0.01	0.04	0.00	0.00	0.02	0.00	0.01	0.01	0.03	0.01	0.00	0.71	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

On constate une accuracy de 67% qui n'est pas mal. On constate également que certaines classes de documents sont plus difficile que d'autres à classifier pour ce modèle. Par exemple les classes 3, 4, 10 et 12 (correspondant respectivement à handwritten, advertisement, budget et presentation) sont souvent prédites comme faisant partie de la classe 8 (file folder).

6.4 Interprétabilité

Suite à l'entraînement du modèle de regression logistique, un poids a été associé à chaque mot possible pour chaque classe. Ces poids indiquent l'importance du mot pour la prédiction des classes. Donc en examinant simplement les poids du modèle entraîné, on peut comprendre quels mots sont les plus importants pour la prédiction de chaque classe. Cela rend l'interprétabilité de ce modèle très simple.

Nous avons choisi d'utiliser une représentation visuelle sous forme de wordcloud des mots les plus importants pour chaque classes.

Par exemple, pour les classes Leter et Invoice, les mots ayant le plus grand poids dans la prédiction sont les suivants.



Cette analyse nous permet de visualiser quels sont les mots que notre modèle associe à chaque classe. Cependant, tout comme pour l'analyse visuelle avec le CNN, cette méthode montre ses limites lorsque les documents contiennent des mots similaires mais appartiennent à des classes différentes. C'est pourquoi les prédictions de notre modèle de NLP et notre modèle CNN ont été combinées pour obtenir une prédiction on l'espere plus précise.

7. Mise en place du modèle de vote

7.1 Explication de l'intérêt

L'utilisation d'un modèle de vote a plusieurs avantages.

- Cela permet de combiner les forces du modèle de CNN qui sera plus performant pour classifier un document selon son arrangement global et du modèle de NLP qui sera lui performant pour analyser le texte du document.
- Cela permet également de mitiger les erreurs éventuelles qu'un des deux modèles pourrait faire.

Nous espérons obtenir une meilleure accuracy global en combinant les modèles qu'avec les modèles seuls.

7.2 Description du modèle de vote retenu

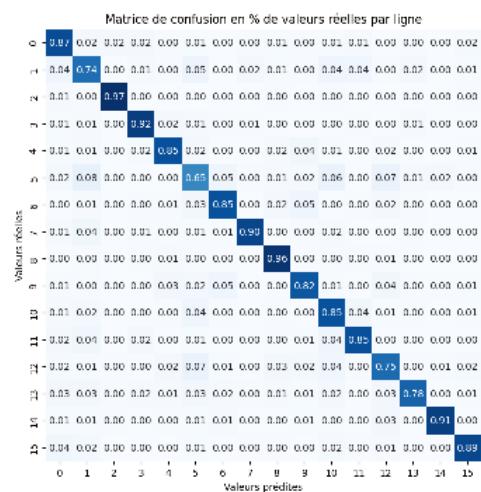
Pour le voting, nous avons choisi d'utiliser le modèle de classification par NLP utilisant la régression logistique. Ce n'est pas le modèle le plus performant, mais il s'est avéré plus complémentaire avec le modèle utilisant un CNN en terme d'accuracy des résultats.

Nous avons choisi d'utiliser un modèle de vote X dit "faible". Ce type de modèle prend en compte les probabilités de chaque classe prédite par nos modèles. Il assemble ces probabilités pour obtenir une nouvelle liste de probabilité pour chaque classe. La classe la plus probable est alors renvoyée comme prediction du modèle. Un système de pondération des probabilités des deux modèles a également été entraîné de façon à ce que les prédictions d'un modèle performant sur une classe en particulier soient avantageées par rapport à l'autre modèle.

7.3 Présentation des résultats du modèle

Après la mise en place du modèle de vote, nous avons évalué ses performances sur un ensemble de test de 4000 documents. Les résultats obtenus sont les suivants :

- Accuracy: 0.8475
- F1 Score: 0.8479



Pour rappel, voici les performances des modèles individuels :

- CNN accuracy: 0.8192
- NLP accuracy: 0.6795
- CNN F1 Score: 0.8196
- NLP F1 Score: 0.6884

On constate donc une amélioration substantielle de l'accuracy suite au voting.

8. Conclusion

8.1 Résultats obtenus

En combinant des modèles de classification visuelle et de classification textuelle, et en utilisant un modèle de vote pour combiner ces deux approches, nous avons réussi à obtenir une accuracy proche de 85%.

Cette approche hybride nous a permis de tirer parti des forces de chaque modèle, maximisant ainsi notre capacité à classer correctement les documents.

Le CNN a permis l'identification des caractéristiques visuelles uniques des documents (formatage, présence de colonnes ...etc), tandis que la NLP a été efficace pour comprendre le contenu du texte. Le modèle de vote a ensuite servi de mécanisme de confirmation/arbitrage, pondérant les probabilités prédictes par chaque modèle en fonction de leur pertinence pour chaque classe.

8.2 Comparaison à l'état de l'art

En comparant nos résultats avec ceux de layoutLM, nous pouvons voir que notre système se comporte mieux que des première itération de CNN non préentraînée, cependant il n'atteint pas l'état de l'art avec des modèles EfficientNet B4 entraînés sur des plus grands datasets.

Author	Year	Pre-trained		Not Pre-trained	
		CNN	CNN + OCR + NLP	CNN	CNN + OCR + NLP
Kumar	2014			43,8	
Kang	2014			65,3	
Afzal	2015			77,6	
Noce	2016				79,8
Afzl	2017	91,13			
Audebert	2019			84,5	87,8
Asim	2019	93,2	95,8		
Proposed wo	2024	81,9	84,8	66	

Nos limites techniques et temporelles n'ont pas permis d'atteindre ces résultats de plus de 95% de précision dans la prediction de classes. Pour autant notre modèle avec une précision de 85% est satisfaisant pour une première approche du problème de classification et est une bonne base robuste pour iterer sur des temps plus importants ou bien avec des ressources supplémentaires.

8.3 Axe d'amélioration

Les limites matériels et temporelles sont très vite apparues comme étant les facteurs limitant d'itérations sur le CNN. Avec un temps plus long et des GPU supplémentaires pour paralleliser le calcul, nous aurions pu iterer sur le CNN multi-couche qui n'avait pas atteint un minimum de coût ou bien tester des modèles plus efficaces mais plus gourmands en ressources comme EfficientNet B2 à B7.

La difficulté principale du côté de l'approche NLP a été le preprocessing des documents en vue d'en obtenir le texte (OCR). La qualité des documents rendait difficile la lecture par la librairie d'ocr. La librairie utilisée n'était pas toujours en mesure de produire le texte exacte du document et produisait très souvent des mots inexacts. Il serait certainement possible de pousser le prétraitement des images pour les rendre plus aptes à être lus par la librairie d'ocr. On pourrait par exemple imaginer de remplacer des mots inexacts par le mot le plus proche dans la langue détectée.

Nous avons également constaté comme on peut s'y attendre qu'augmenter la taille du jeu d'entraînement améliorait les métriques évaluées. On pourrait certainement utiliser un jeu d'entraînement plus grand pour encore améliorer ces métriques. Nous n'avons utilisé que 5% du dataset pour entraîner notre modèle, mais en entraînant sur un plus grand dataset allant jusqu'à 90% du dataset comme le fait EfficientNet sur ImageNet, nous pourrions accroître la précision au prix d'un entraînement beaucoup plus long pouvant dépasser la semaine avec nos ressources actuelles. Il a cependant aussi été observé des rendements décroissants en utilisant cette approche lors des phases d'entraînement.

8.4 Applications

Le modèle combiné que nous avons créé permet de classer un document parmi les 16 catégories sélectionnées mais il pourra facilement être adapté à d'autres problématiques de classification de document, en rajoutant de nouvelles classes pour détecter d'autres types de documents ou encore en changeant la langue du modèle OCR et NLP pour classer des documents dans d'autres langues.

Enfin, ce modèle nous permet aussi d'améliorer l'extraction d'information car on pourra par la suite sélectionner certains types d'OCR en fonction de la classe du document pour améliorer l'extraction d'information.