



DataScientest • com

Rapport Technique d'évaluation Classification de documents

Promotion : Novembre 2023 Data Scientist (continu)

ASRI Achraf,
BENKOU Ayoub,
FERNANDEZ Bryan,
HRYNIEWSKI Daniel

Année: 2024

Table des matières

1. Préambule :	3
2. Les données brutes	3
3. Analyse exploratoire des données physiques des images	4
3.1. Premier coup d'œil sur les images	4
3.2. Préparation du fichier .CSV	5
3.3. Répartition des images par catégorie dans les jeux de données	5
3.4. Analyse de la définition des images	6
3.4.1. Analyse de la définition des images par jeu de données	6
3.4.2. Analyse de la définition des images par catégorie par jeu de données	7
3.5. Analyse du flou des images par jeu de données	8
3.6. Analyse des histogrammes des couleurs des jeux de données	9
4. Analyse exploratoire des données textuelles dans les images	9
4.1. Distribution des langues dans l'ensemble des jeux de données	10
4.2. Distribution du nombre de mots par page par catégorie	11
4.3. Quels mots caractérisent chaque catégorie ?	11
5. Synthèse de l'analyse exploratoire des données	13
6. Première modélisation : Classification à partir des données textuelles.	14
6.1. Prétraitement des données textuelles.	14
6.2. Choix des modèles (Lazypredict)	16
6.3. Résultats d'entraînements	17
7. Deuxième modélisation : Classification à partir des images.	19
7.1. Normalisation du dataset PRADO	19
7.2. Pipeline de chargement et de transformation d'images	20
7.3. Entraînement de modèles de Deep Learning.	21
8. Résultats et comparaison de modèles de deep learning.	23
8.1. Courbes d'entraînement	23
8.2. Matrices de Confusion	24
8.3. Précisions des Modèles par Catégorie	24
8.4. Évaluation du Temps de Prédiction	25
8.5. Zones d'intérêts : GradCAM CNN et ResNet :	27
8.6. Synthèse des résultats	28
9. Difficultés rencontrées lors du projet.	28
9.1. Prétraitement des Données Textuelles	28
9.2. Classification des données textuelles en machine learning	29

9.3. Développement des modèles de deep learning : Classification d'images	29
9.4. Évaluation et Interprétation des Modèles :	30
9.5. Prendre de l'avance sur la formation :	30
10. Conclusion.	30
10.1. Bilan	30
10.2. Suite du projet	31
11. Bibliographie	32
12. Diagramme de gantt	33

1. Préambule :

Ce projet fil rouge a pour but d'introduire les notions de base de classification de documents vues en cours et de les appliquer. Pour chacun des membres de ce projet, nous avons des contextes différents, pour certains, ce projet s'inscrit dans le cadre professionnel actuel en entreprise, et pour d'autres, une diversification des compétences pour une potentielle reconversion de carrière.

Dans un contexte où la quantité de données augmente exponentiellement, il devient important de pouvoir extraire les informations et de les analyser de façon efficace et précise. Ce projet répond à un besoin où les documents se retrouvent de plus en plus sous format numérique (images). Ce projet vise à développer des modèles intelligents pour catégoriser efficacement une variété de documents visuels (Facture, carte nationale d'identité, passeport, etc.). Ces modèles permettent, dans le cas d'une entreprise, d'organiser ses données mais aussi d'automatiser des tâches de traitement et d'analyse.

Ce projet consiste en deux axes principaux. Dans un premier temps nous allons modéliser un modèle de classification en utilisant seulement les données textuelles. Puis nous allons élaborer un modèle de classification d'image type CNN. Afin d'atteindre les objectifs de ce projet de classification de documents numérisés, une phase en amont d'analyse exploratoire des données, de prétraitement et de choix des modèles ainsi que de la métrique adaptée sera réalisée.

2. Les données brutes

Les données initiales à notre disposition pour ce projet sont les suivantes :

- **Data_01**, Text extraction for OCR : L'ensemble de données est constitué de fichiers XML et d'images. Les fichiers XML contiennent les données textuelles extraites des images. Au total 519 images de factures.
- **Data_02**, Projet OCR classification. Le dataset contient 1607 images de tout type de documents (23 classes).
- **Data_03**, RVL-CDIP Dataset : L'ensemble de données RVL-CDIP (Ryerson Vision Lab Complex Document Information Processing) se compose de 400 000 images en niveaux de gris réparties en 16 classes, avec 25 000 images par classe. Il existe 320 000 images d'entraînement, 40 000 images de validation et 40 000 images de test. Les images sont dimensionnées de manière que leur plus grande dimension ne dépasse pas 1 000 pixels.

- **Data_04**, PRADO : Nous avons ajouté ces données, de notre propre initiative, provenant du Conseil européen. Ce jeu de données recense des spécimens de passeports et de cartes nationales d'identité provenant de divers pays du monde.

3. Analyse exploratoire des données physiques des images

3.1. Premier coup d'œil sur les images

Avant même d'analyser quoi que ce soit, nous avons affiché des échantillons d'images de certaines catégories. Voici un exemple ci-dessous de la catégorie « passeport » du jeu de données **data_02**. Cette étape a pour but de déjà comprendre quels types d'images nous avons à disposition, l'homogénéité des images et la justesse d'annotation. Pour voir tous les échantillons de chaque catégorie par jeu de données, veuillez-vous reporter au notebook *00-hryniewski-download-datasets.ipynb*.

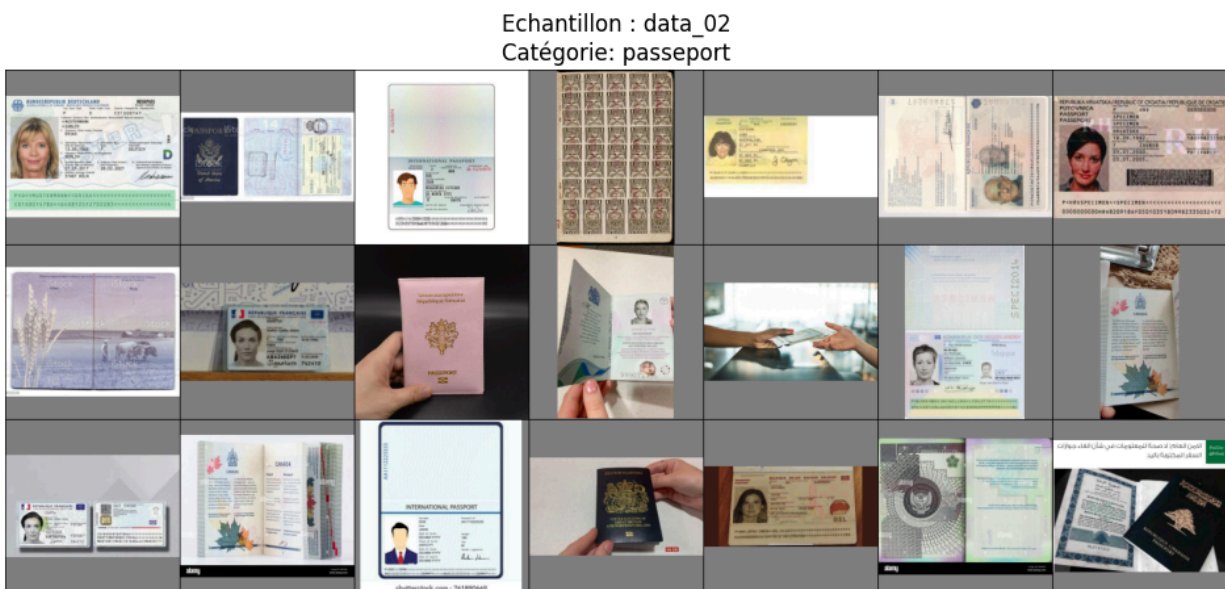


FIGURE 1 EXEMPLE ÉCHANTILLONS CATÉGORIE "PASSEPORT" DU JEU DE DONNÉES DATA_02

Nous avons pu déjà relever quelques incohérences dans le jeu de données data_02 :

- Présence de CNI dans la catégorie « passeport »
- Présence de factures dans la catégorie « justif_domicile »

3.2. Préparation du fichier .CSV

Sachant que nous avons des données de types images, nous avons d'abord identifié des variables qui nous seront utiles dans la compréhension de notre jeu de données et qui nous permettront de réaliser nos fonctions de prétraitement à l'issue de l'analyse exploratoire des données (cf : 01-aasri-extract-img-info.ipynb). Voici un extrait du fichier CSV sur lequel nous allons travailler comprenant les variables retenues :

	image_name	colorspace	height	width	bluriness	path	dataset	label
0	image_0000000.fit	gray	1000	767	1656	data/raw/data_01/images/image_0000000.tif	data_01	facture
1	image_0000001.fit	gray	1000	754	3841	data/raw/data_01/images/image_0000001.tif	data_01	facture
2	image_0000002.fit	gray	1000	771	2121	data/raw/data_01/images/image_0000002.tif	data_01	facture
3	image_0000003.fit	gray	1000	781	4507	data/raw/data_01/images/image_0000003.tif	data_01	facture
4	image_0000004.fit	gray	1000	783	3481	data/raw/data_01/images/image_0000004.tif	data_01	facture

FIGURE 2 EXTRAIT DU FICHIER CSV

Remarque :

Compte tenu de la taille assez conséquente du jeu de données **data_3**, nous avons réduit sa taille en récupérant 5% d'images par catégorie.

3.3. Répartition des images par catégorie dans les jeux de données

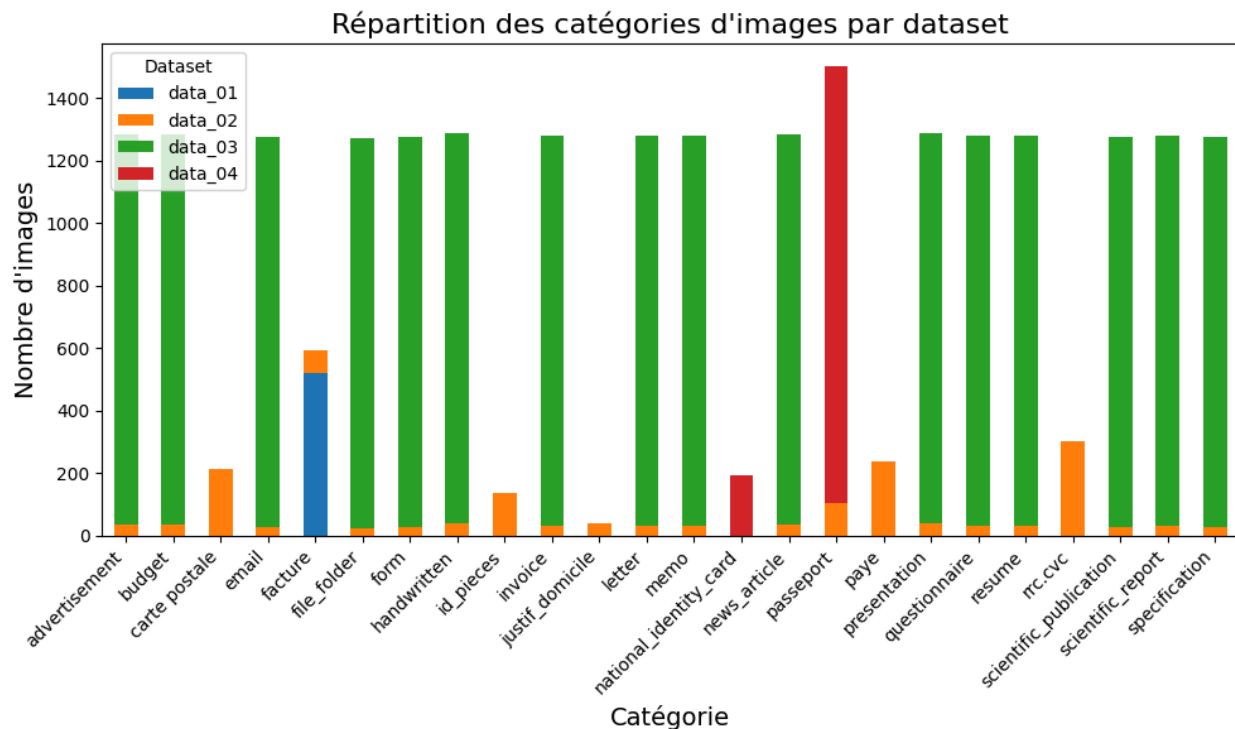


FIGURE 3 RÉPARTITION DES CATÉGORIES PAR DATASET

On peut observer plusieurs points intéressants :

- **data_01** : Il est normal d'avoir que des factures car ce jeu de données n'est constitué que de cette catégorie.
- **data_02** : On remarque un déséquilibre entre les catégories.
- **data_03** : Nous avons évidemment un jeu de données équilibré étant donné que nous l'avons réduit à 5% d'images par catégorie
- **data_04** : Ce jeu de données ne contient que des documents d'identité. On remarque que nous avons plus d'image dans la catégorie passeport que national_identity_card. Dans l'aperçu des échantillons, on s'aperçoit que les pages intermédiaires de passeport sont aussi présentes.

3.4. Analyse de la définition des images

3.4.1. Analyse de la définition des images par jeu de données

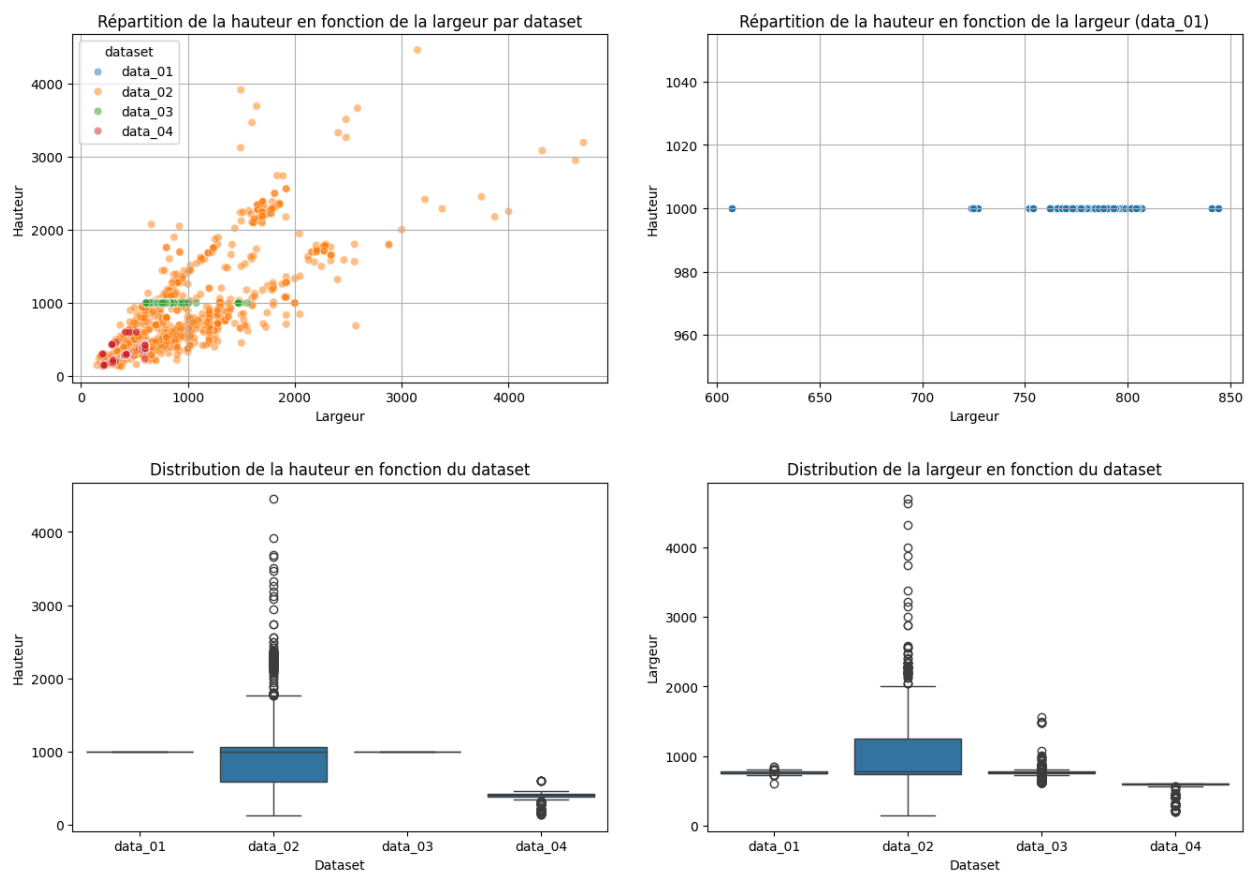


FIGURE 4 GRAPHES PERMETTANT D'ANALYSER LA DÉFINITION DES IMAGES PAR JEU DE DONNÉES

Ces graphes nous permettent de voir l'uniformité de la résolution des images dans chaque jeu de données. On remarque les points suivants :

- Les jeux de données **data_01** et **data_03** ont des images relativement de même résolution. Seule la largeur varie mais relativement peu par rapport au jeu de données **data_02**.
- Les points du jeu de données **data_04** sont fidèles (regroupés ensemble) mais de basse résolution.
- Au contraire, les points du jeu de données **data_02** sont plus dispersés.

3.4.2. Analyse de la définition des images par catégorie par jeu de données

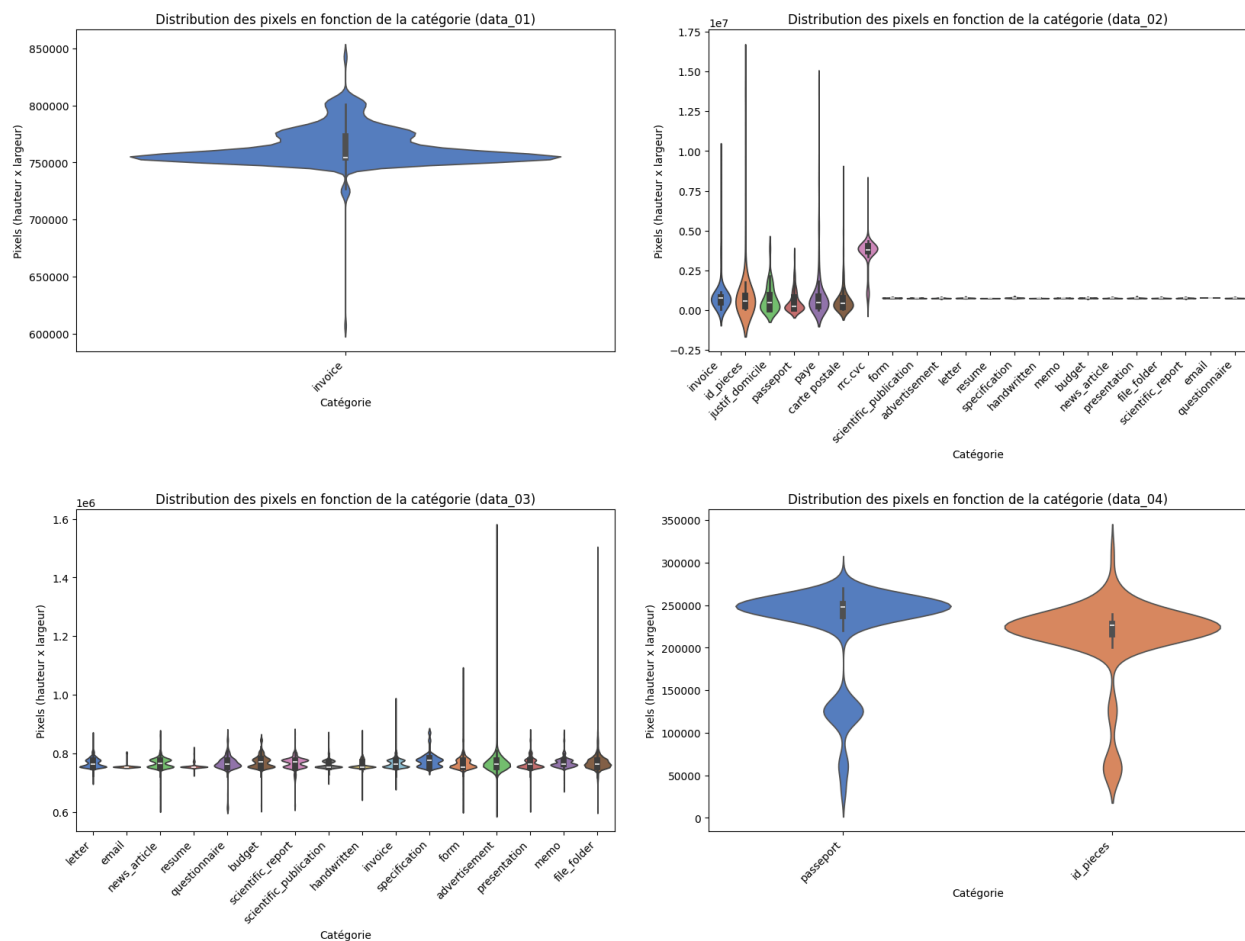


FIGURE 5 GRAPHES PERMETTANT D'ANALYSER LA DÉFINITION DES IMAGES PAR CATÉGORIE PAR JEU DE DONNÉES

Ces graphes nous permettent de faire les observations suivantes :

- Dans le jeu de données **data_01**, la distribution est moins étendue.

- Dans le jeu de données **data_03**, nous avons deux catégories présentant des valeurs aberrantes : "advertisement" et "file_folder"
- Dans le jeu de données **data_02**, certaines catégories présentent des images de taille uniforme, telles que 'form', 'scientific_publication', etc.
- Dans le jeu de données **data_02** nous avons des valeurs aberrantes comme dans les catégories "invoice", "id_pieces", "paye" et "carte_postale".

3.5. Analyse du flou des images par jeu de données

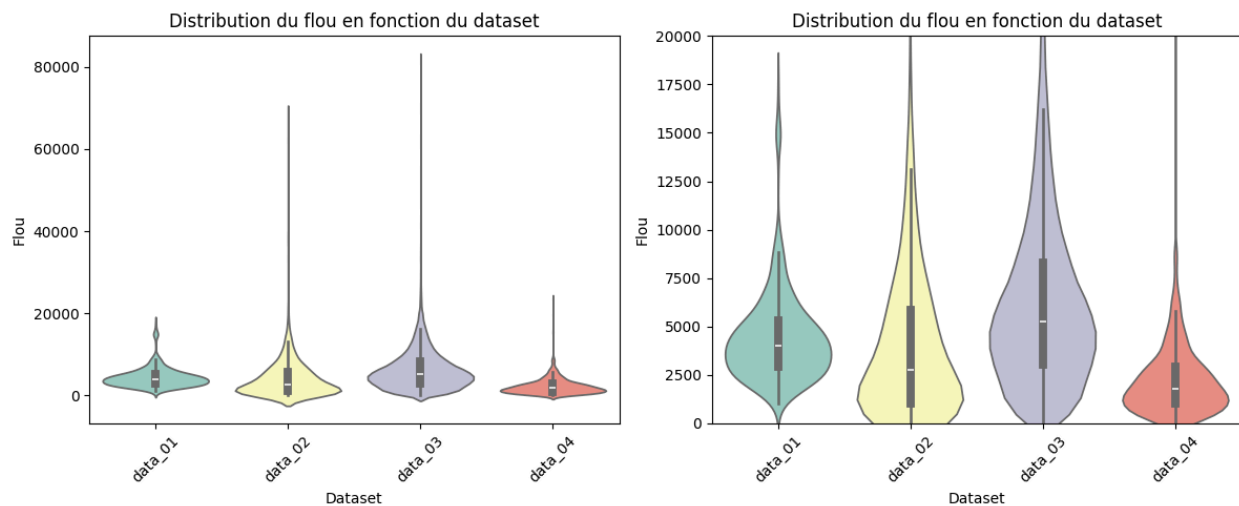


FIGURE 6 DISTRIBUTION DU FLOU PAR DATASET

Cette partie nous montre des images floues. Plus le score est bas, plus l'image est floue. Ici nous avons fixé un seuil à 60. Après quelques observations, les images ayant un score au-delà de ce seuil sont assez nettes et lisibles. Il y a 12 images ayant un score en dessous du seuil de 60 (*cf : 07-fernandez-synthese-aed.ipynb*)

3.6. Analyse des histogrammes des couleurs des jeux de données

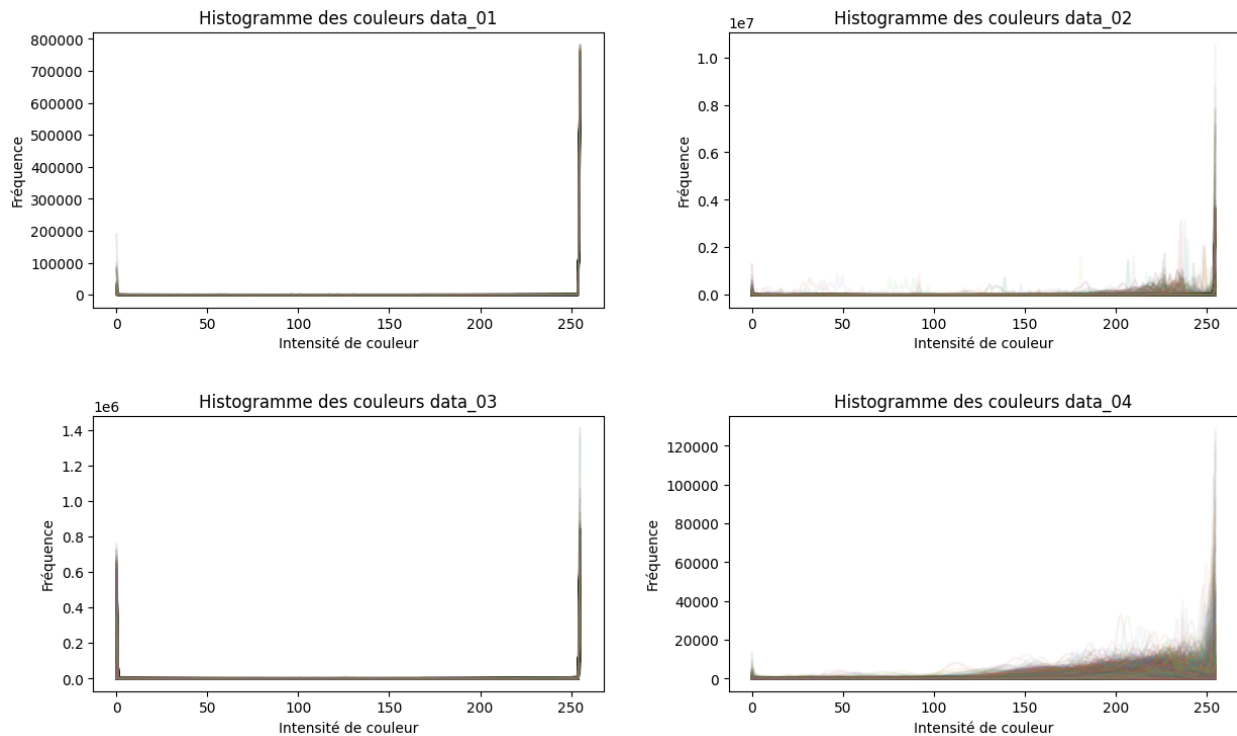


FIGURE 7 GRAPHES DES HISTOGRAMMES DES COULEURS PAR JEU DE DONNÉES

- Les jeux de données **data_01** et **data_03** sembleraient être composés exclusivement d'images en noir et blanc, car les histogrammes des couleurs ne montrent aucune variation significative de couleurs entre 0 et 255.
- Les jeux de données **data_02** et **data_04** comportent des images en couleur, comme on peut le voir dans la variation des courbes dans les histogrammes des couleurs.

4. Analyse exploratoire des données textuelles dans les images

Dans cette partie, nous allons utiliser le fichier csv que nous avons réalisé sur le jeu de données global (pas de distinction entre les jeux de données)

4.1. Distribution des langues dans l'ensemble des jeux de données

A l'aide du module de « Texte mining » vu en cours, nous avons vu qu'il était nécessaire d'extraire la langue dans les images afin de réaliser un filtrage efficace. Après avoir réalisé une océration à l'aide de l'outil Tesseract (*cf : 04-hryniewski-tesseract.ipynb*), nous avons défini une fonction permettant d'extraire la langue dans les images. On obtient le graphe suivant :

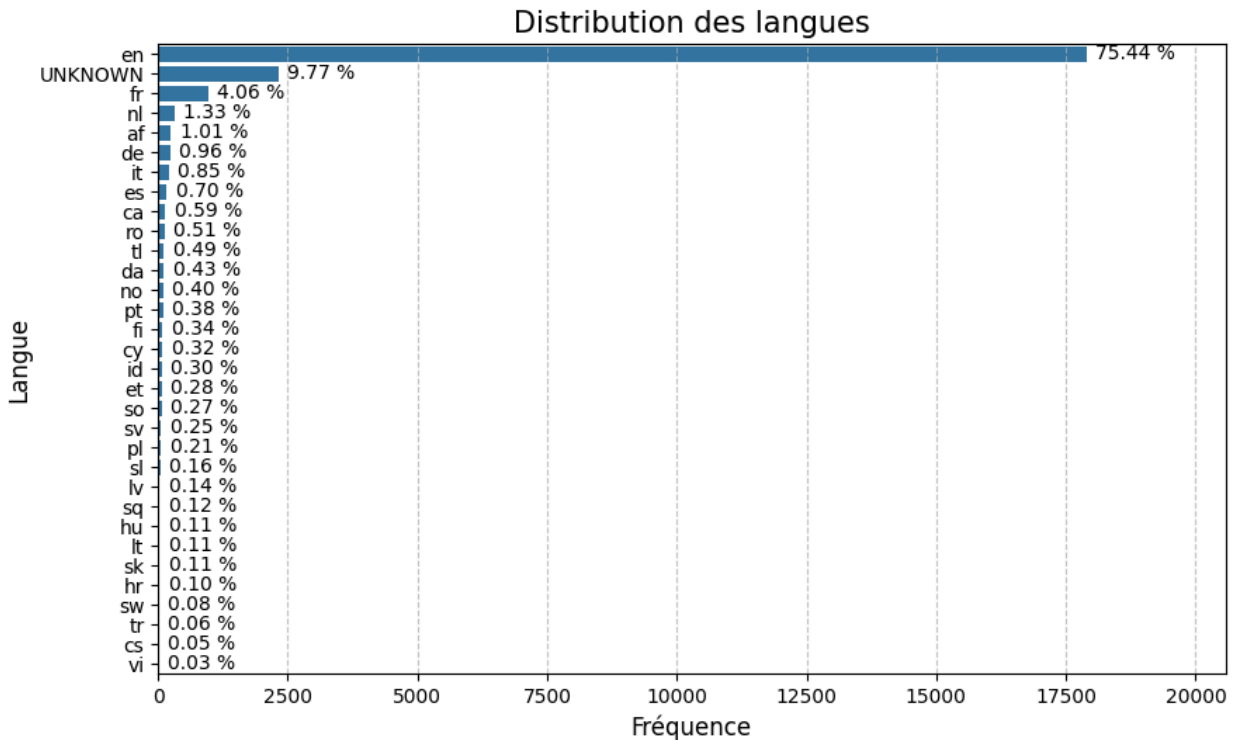


FIGURE 8 DISTRIBUTION DES LANGUES DANS LE JEU DE DONNÉES GLOBAL

On peut rapidement remarquer que la langue anglaise est la plus présente dans notre jeu de données global représentant 75.44%. La deuxième langue connue est le français. Les autres langues représentent chacune d'elle qu'une infime partie du jeu de données global. La diversité des langues s'explique avec la base de données **data_04** constituée entièrement de passeport et des cartes nationales d'identités de différents pays.

4.2. Distribution du nombre de mots par page par catégorie

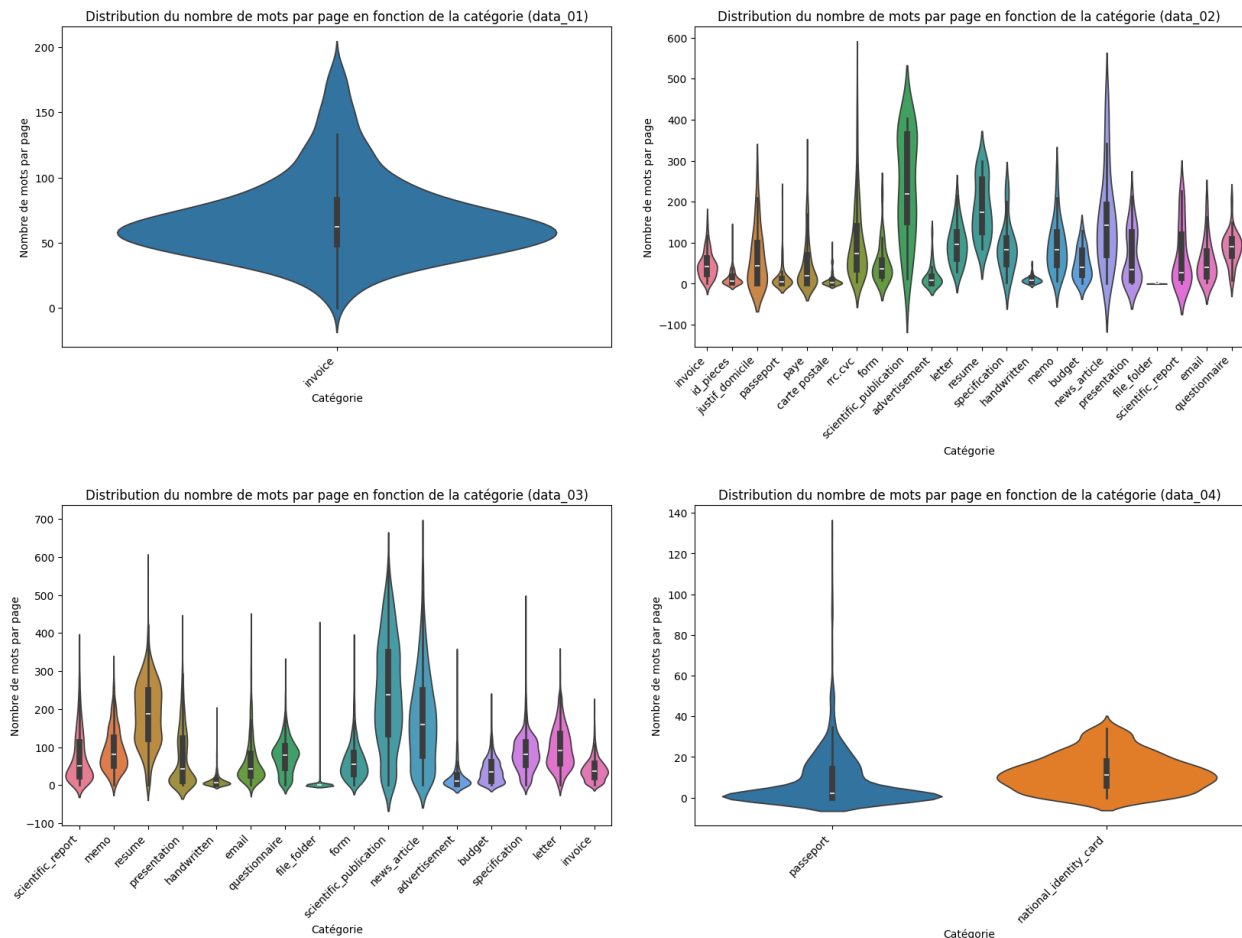


FIGURE 9 DISTRIBUTION DES MOTS PAR PAGES PAR CATÉGORIE

Le but de ces graphes était de voir si le nombre de mots par page était uniforme à travers les différentes catégories (petite étendue en y). Cependant on remarque que de nombreuses catégories ont des densités différentes notamment les catégories supposées comporter énormément de textes telles que "scientific_publication" et "news_article". Néanmoins, on peut voir que certaines catégories ont la densité la plus large au niveau de la médiane telles que "passeport", "invoice" et "avertissement".

4.3. Quels mots caractérisent chaque catégorie ?

Nous avons affiché les nuages de mots de chaque catégorie pour voir quels sont les mots que l'on retrouve le plus souvent. Un travail de filtrage des *stop words* et des mots d'une longueur inférieure à 3 caractères a été fait.

[illegible]

Par la suite, nous avons effectué le graphe représentant la fréquence des 5, 10, 25 et 50 mots les plus présents dans chaque catégorie.



Ce graphique est très intéressant si l'on souhaite faire de la classification à partir des données textuelles. Il nous permet de voir quel pourcentage on obtient en fonction du nombre de mots les plus fréquents par rapport à la totalité des mots par catégorie.

Prenons par exemple la catégorie "invoice" : Les 10 mots les plus fréquents de cette catégorie ne présentent même pas 10% de la totalité des mots de la catégorie. Mais les 50 mots les plus fréquents représentent plus de 20% des mots de la catégorie.

5. Synthèse de l'analyse exploratoire des données

L'analyse exploratoire des données a révélé plusieurs points importants. L'affichage des échantillons a montré que certaines images du jeu de données **data_02** n'étaient pas correctement catégorisées (par exemple, des cartes d'identité se trouvaient dans la catégorie "passeport"). Ceci était un point à prendre en compte dans la suite de nos analyses car nous faisons plusieurs divisions par catégorie au sein des jeux de données.

L'analyse des caractéristiques numériques des images a révélé que les définitions des images dans les jeux de données **data_01**, **data_03** et **data_04** étaient relativement uniformes, contrairement au jeu de données **data_02** qui présentait une plus grande dispersion. La quantité d'images floues était globalement faible (12 images en dessous du seuil de netteté). Enfin, nous avons vu les différents espaces colorimétriques par jeu de données.

L'analyse des textes extraits des images a permis d'identifier les langues présentes dans les images. Ensuite, une analyse des fréquences des mots a été réalisée pour déterminer quels mots caractérisent chaque catégorie et à partir de quel nombre de mots les plus fréquents un seuil représentatif des catégories était atteint.

Les analyses ont conclu que le jeu de données **data_02** n'était pas suffisamment exploitable pour la suite du projet en raison du trop grand nombre d'images mal catégorisées. Il a été constaté que certaines images, bien que correctement catégorisées, ne présentaient pas beaucoup d'intérêt (par exemple, une image avec uniquement 'Bulletin de paie' écrit en gros). De plus, les tailles des images varient trop dans chaque catégorie de ce jeu de données. Par conséquent, il a été décidé d'abandonner le jeu de données **data_02**.

Ces analyses serviront également à définir les fonctions de prétraitement nécessaires pour uniformiser les images et les textes, facilitant ainsi la construction des pipelines d'images et de texte.

Enfin, nous avons décidé de garder les catégories suivantes :

- Passeport
- National ID card
- Email
- Invoice
- Scientific publication
- Handwritten

Ces classes sont les plus pertinentes pour une application concrète et disposent d'un jeu de données large et équilibré en termes de quantité entre les classes. De plus, les images sont suffisamment nettes (scannées).

6. Première modélisation : Classification à partir des données textuelles.

Dans cette partie, la variable cible est "category". L'objectif est de modéliser un ou plusieurs classifieurs permettant d'attribuer le bon label. Seules les données textuelles présentes dans les images seront utilisées. L'extraction du corpus de mots a été effectuée avec Tesseract.

6.1. Prétraitement des données textuelles.

Trois approches possibles pour la modélisation de modèles à partir des données textuelles :

- Utiliser uniquement le corpus des pages.
- Déterminer et utiliser les données structurelles.
- Combiner les deux premières approches.

Pour l'approche n°1, un TF-IDF a été réalisé. TF-IDF est une mesure statistique utilisée pour évaluer l'importance d'un mot dans un document par rapport au corpus.

Pour l'approche n°2, qui se base sur les données de "structure", ce qui est sous-entendu derrière le mot "structure" inclut le nombre de mots, la diversité lexicale et la densité des mots-clés pour chaque catégorie et pour 5, 10, 25 et 50 mots les plus communs (cf : *06-fernandez-concat-text_process.ipynb*).

Définitions :

- **Nombre de mots** : Nombre total de mots dans le document.
- **Diversité lexicale** : Mesure de la diversité du vocabulaire au sein d'un texte donné. Elle est calculée comme le rapport entre le nombre de mots uniques et le nombre total de mots.
- **Densité des mots-clefs** : Mesure de la fréquence des mots-clés spécifiés dans un texte. Elle est calculée comme le rapport du nombre d'occurrences des mots-clés au nombre total de mots.

La densité des mots-clés ne dépend pas de la catégorie de la page. Ce calcul est effectué pour chaque catégorie.

doc_id	categorie	count	lexical_diversity
0	passport	22	0,909091
1	email	48	0,854167
2	invoice	80	0,862500

FIGURE 13 EXEMPLE D'ANNOTATION DE NOMBRE DES MOTS ET DE LA DIVERSITÉ LEXICALE PAR DOCUMENT

doc_id	categorie	keyword_passport_5	keyword_passport_10	keyword_passport_25	keyword_passport_50
0	passport	0,136364	0,136364	0,136364	0,181818
1	email	0,020833	0,041667	0,041667	0,041667
2	invoice	0,012500	0,012500	0,050000	0,050000

doc_id	categorie	keyword_email_5	keyword_email_10	keyword_email_25	keyword_email_50
0	passport	0,000000	0,000000	0,000000	0,000000
1	email	0,041667	0,104167	0,166667	0,187500
2	invoice	0,050000	0,062500	0,062500	0,137500

FIGURE 14 EXEMPLE D'ANNOTATION DE DENSITÉ DES MOTS-CLEFS

Pour mieux comprendre ce point, prenons l'exemple du document avec l'ID 1. Dans la figure 14 on peut voir que la catégorie de cette page est "email". Cependant, on remarque que parmi les 5 mots-clés les plus fréquents de la catégorie "passport", il y a un rapport d'environ 0.02. Cela veut dire que dans ce document, il y a des mots-clés parmi les 5 les plus fréquents de la catégorie "passport".

Dans la figure 14, toutes les colonnes ne sont pas toutes visibles. Chaque document a bien une densité de mots-clés pour chaque catégorie et pour 5, 10, 25 et 50 mots-clés les plus fréquents. La base de données de "structure" est composée de vingt-quatre colonnes avec la densité des mots-clés, une colonne avec la densité lexicale et une colonne comportant le nombre des mots.

6.2. Choix des modèles (Lazypredict)

Pour le choix des modèles de classification, la librairie Python Lazypredict a été utilisée. Cette librairie permet de former et de comparer plusieurs modèles avec un minimum de code, offrant un moyen pratique de comparer différents algorithmes sur un ensemble de données. Cela est particulièrement utile pour obtenir rapidement une idée des modèles susceptibles de bien fonctionner sur un problème spécifique (*cf : 08-fernandez-choice-models-text.ipynb*).

Model	Words				Structure				Words & Structure			
	Accuracy	Balanced Accuracy	F1 Score	Time Taken	Accuracy	Balanced Accuracy	F1 Score	Time Taken	Accuracy	Balanced Accuracy	F1 Score	Time Taken
LogisticRegression	0.86	0.82	0.86	231.71	0.77	0.71	0.77	0.11	0.86	0.82	0.86	102.80
ExtraTreesClassifier	0.86	0.81	0.86	358.41	0.81	0.76	0.81	0.39	0.85	0.80	0.86	92.40
RandomForestClassifier	0.84	0.79	0.84	369.69	0.81	0.76	0.82	0.91	0.86	0.81	0.86	223.19
XGBClassifier	0.82	0.77	0.82	656.11	0.82	0.77	0.82	0.89	0.85	0.81	0.85	514.32
LGBMClassifier	0.82	0.76	0.82	265.25	0.82	0.77	0.82	0.87	0.84	0.79	0.84	217.54
NearestCentroid	0.85	0.81	0.85	261.89	0.66	0.64	0.67	0.01	0.85	0.82	0.86	191.00
PassiveAggressiveClassifier	0.84	0.80	0.84	429.68	0.73	0.68	0.70	0.04	0.84	0.80	0.84	247.65
RidgeClassifier	0.82	0.78	0.83	380.51	0.74	0.67	0.72	0.01	0.83	0.78	0.83	284.48
RidgeClassifierCV	0.82	0.78	0.83	431.81	0.74	0.67	0.72	0.03	0.83	0.78	0.83	334.38
LinearSVC	0.85	0.82	0.85	5284.04	0.76	0.70	0.75	1.64	0.85	0.82	0.85	5461.39

FIGURE 15 LES MEILLEURS RÉSULTATS OBTENUS AVEC LAZYPREDICT.

Après l'exécution de Lazypredict selon les trois approches, six modèles de classification ont été identifiés en fonction des différentes métriques :

- Logistic Regression
- Random Forest Classifier
- Nearest Centroid
- Extra Trees Classifier
- XGBClassifier
- LGBMClassifier

Un comparatif des six modèles a été réalisé pour chaque approche avec différents paramètres afin d'identifier le modèle optimal pour la classification des documents à partir des données textuelles.

Pour chaque approche, le protocole suivant a été appliqué :

- Chargement des jeux de données.
- Prétraitement en fonction des données d'entrées (standardisation, TF-IDF, etc.).
- Définition de la grille de paramètres.
- Recherche d'hyperparamètres avec GridSearchCV.
- Analyse des résultats pour chaque paramètre du GridSearchCV.
- Analyse des résultats des meilleurs paramètres pour chaque modèle.
- Entraînement d'un Voting classifier.

Pour plus de détails sur les résultats de chaque hyperparamètres de chaque modèle veuillez-vous reporter aux notebooks: *"09-benkou-models_standarscaler.ipynb"*, *"10-aasri-tfidf-modeling-text.ipynb"* et *"11-hryniewski-GridSearchCV-words_structure.ipynb"*.

6.3. Résultats d'entraînements

L'accuracy a été calculé sur chaque modèle pour les trois approches, ainsi que localement pour chaque catégorie. Les catégories les mieux détectées sont : "email", "invoice", "passeport" et "scientific publication". Les catégories les moins bien détectées sont "handwritten" et "national identity card". Pour les documents de la catégorie "handwritten", cela est compréhensible, car Tesseract n'est pas entraîné pour détecter et reconnaître les caractères manuscrits. Nous ne savons pas pourquoi la catégorie "national identity card" est la moins bien classée.

Voici la représentation graphique des résultats des trois approches:

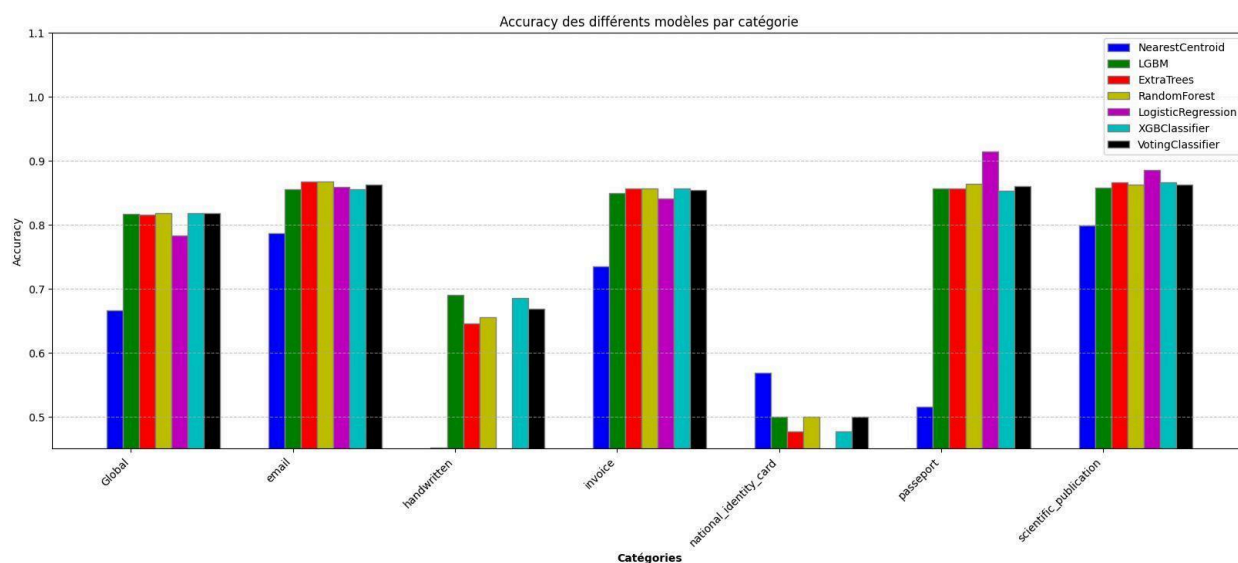


FIGURE 16 RÉSULTATS D'ACCURACY SUR LES DONNÉES DE TESTS SUR L'APPROCHE DE LA STRUCTURE DES MOTS.

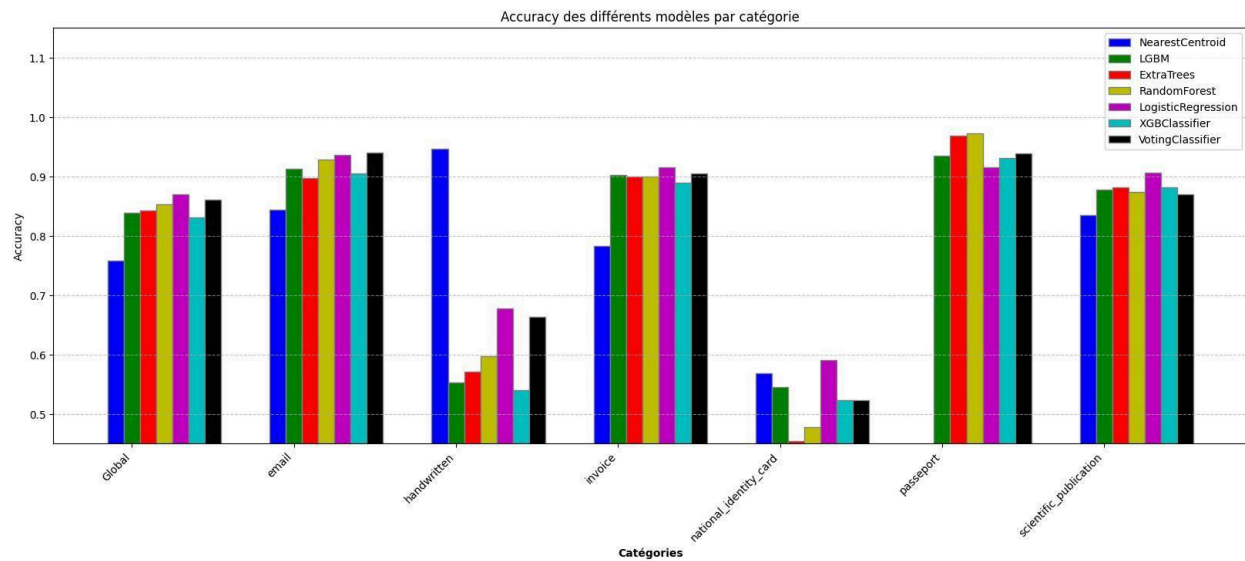


FIGURE 17 RÉSULTATS D'ACCURACY SUR LES DONNÉES DE TESTS SUR L'APPROCHE UNIQUE DE MOTS.

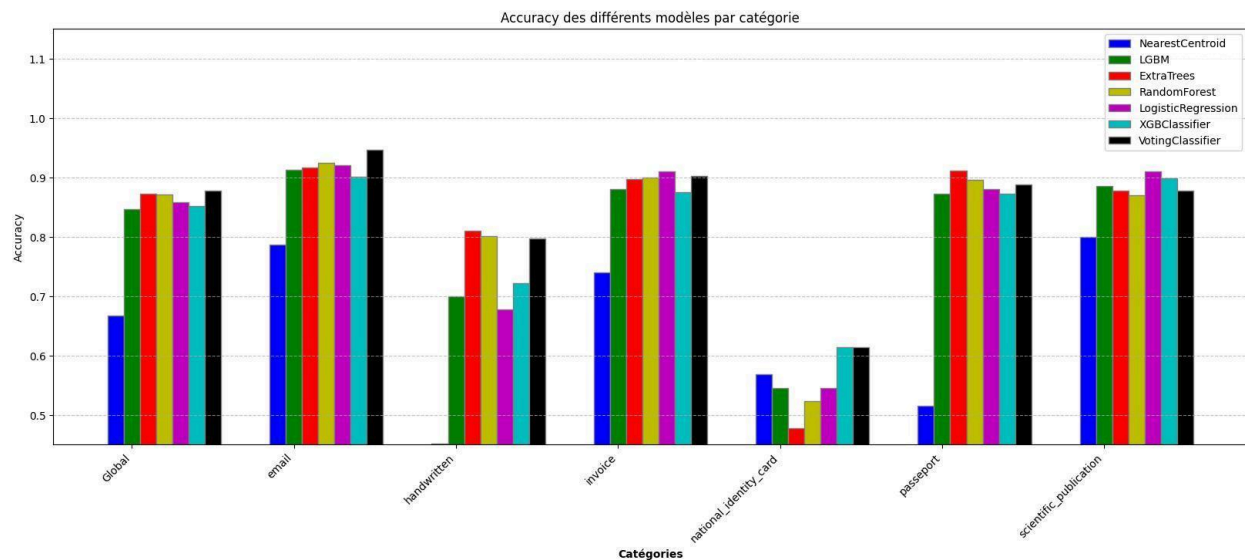


FIGURE 18 RÉSULTATS D'ACCURACY SUR LES DONNÉES DE TESTS SUR L'APPROCHE DE LA COMBINAISON DES DEUX AUTRES APPROCHES.

Globalement, les modèles ont appris à classer les documents, indépendamment de l'approche d'entraînement, avec des résultats d'accuracy avoisinant les 85%. Un point important à noter est que le seul modèle qui se distingue en classant correctement la catégorie "handwritten" est le modèle Nearest Centroid utilisant uniquement les mots, avec une accuracy de 0.95.

Voici le meilleur résultat d'accuracy des trois approches :

- **Structure des mots :**
 - Meilleur modèle : **XGBClassifier**
 - Accuracy : 0.8185
- **Uniquement les mots :**
 - Meilleur modèle : **LogisticRegression**
 - Accuracy : 0.8691
- **La combinaison des deux approches :**
 - Meilleur modèle : **VotingClassifier**
 - Accuracy : 0.8776

Le modèle ayant l'accuracy la plus élevée, à savoir le Voting Classifier avec une accuracy de 87.8%, a été retenu.

Malgré les excellents résultats obtenus lors des tests, il est important de noter que les données sont biaisées. L'analyse des échantillons et des nuages de mots a montré que de nombreuses images contiennent des informations du domaine du tabac pour les catégories "email", "invoice" et "scientific publication". Cela signifie que les modèles développés performant très bien dans ce domaine spécifique, mais peuvent être moins efficaces dans des contextes différents et plus globaux.

7. Deuxième modélisation : Classification à partir des images.

7.1. Normalisation du dataset PRADO

Parmi les bases de données utilisées, **data_04 (PRADO)** présente des images différentes des autres, ne suivant pas le format standard de page A4 scannée. Pour faciliter l'apprentissage des modèles, il a été décidé d'entraîner ces derniers que sur des documents scannés. Ainsi, une normalisation du dataset **data_04** s'est avérée nécessaire.

L'objectif est de générer une base de données conséquente avec des documents d'identité positionnés aléatoirement sur une page A4. La base de données PRADO contient également la

taille réelle des documents en millimètres, ce qui permet de calculer la taille idéale de l'image par rapport au DPI (dots per inch).

Le calcul est le suivant :

$$\text{taille_en_pixels} = (\text{taille_en_mm} / 25.4) * \text{DPI}$$

La taille est convertie en pouces, puis multipliée par le DPI choisi en amont. Pour la génération des images, un ou plusieurs documents de la même catégorie sont positionnés aléatoirement avec une rotation aléatoire sur une image vierge de format A4.

Un total de 1500 images A4 avec des cartes nationales d'identité et 1500 images A4 avec des passeports ont été générés.



FIGURE 19 EXEMPLES D'IMAGES EN FORMAT A4 DU DATASET PRADO.

7.2. Pipeline de chargement et de transformation d'images

La base de données est composée de 6816 images d'entraînement et de 1704 images de test. Pour unifier les trois datasets, un fichier train et un fichier test au format CSV ont été créés, contenant les chemins de chaque image ainsi que leur catégorie. L'encodage de chaque catégorie est le suivant :

- **Email** : 0
- **Handwritten** : 1
- **Invoice** : 2
- **National Identity Card** : 3
- **Passport** : 4
- **Scientific Publication** : 5

La taille des images d'entrée des modèles est de 224 x 224 pixels. Étant donné que les images originales sont rectangulaires, une transformation est nécessaire.

Deux solutions sont possibles :

- Redimensionnement : Redimensionner les images à la taille exacte de 224 x 224 pixels, ce qui change le ratio et provoque une déformation des documents.
- Padding : Ajouter du padding de chaque côté des images pour conserver le même ratio et la structure des documents. Cette méthode a été choisie.

Les images doivent être normalisées, il est plus efficace d'entraîner les modèles avec des valeurs proches de zéro plutôt qu'avec des valeurs allant de 0 à 255. La moyenne et l'écart-type de chaque couche RGB des images d'entraînement ont été calculés pour une normalisation plus précise.

Avec ces transformations, il est possible d'entraîner les modèles. Cependant, pour obtenir des modèles plus robustes, un pipeline d'augmentation de données est nécessaire. Voici le pipeline mis en place :

- Chargement des images en couleur ou en niveaux de gris.
- Application aléatoire de la compression JPEG.
- Application aléatoire de flip horizontal, vertical ou diagonal.
- Ajout de "salt & pepper noise".
- Ajout de "gaussian noise".
- Mise à l'échelle (224 x 224 x 3).
- Normalisation.

Chaque dégradation ou transformation d'image est appliquée avec des paramètres aléatoires, sans excès, afin de préserver la reconnaissabilité de l'image d'origine.

7.3. Entraînement de modèles de Deep Learning.

Pour l'entraînement des modèles de Deep Learning, deux approches ont été adoptées.

Premièrement, création d'un modèle de réseau de neurones convolutif (CNN) personnalisé et entraîné à partir de zéro.

Deuxièmement, l'utilisation de modèles pré entraînés avec la méthode du transfert learning. Plusieurs modèles de CNN ont été utilisés, notamment ResNet50, SqueezeNet, EfficientNetB1 et MobileNetV2. Cette méthode permet de tirer parti de modèles ayant déjà appris à extraire des caractéristiques utiles à partir de vastes bases de données comme ImageNet. Les modèles sélectionnés se distinguent par leur efficacité et leur petite taille, ce qui les rend adaptés pour une utilisation sur un CPU avec un temps de traitement raisonnable.

Modèle de convolution	Description	Nombre total de paramètres	Notes supplémentaires
CNN (modèle from scratch)	Réseau classique de convolution, conçu spécifiquement pour la classification de documents, avec une architecture personnalisée pour cette tâche.	44 399 174	Architecture personnalisée selon spécification
ResNet50	ResNet50 est un réseau profond qui utilise des connexions résiduelles pour faciliter l'apprentissage de réseaux très profonds.	24 113 798	Comprend 50 couches dans les blocs résiduels
MobileNetV2	MobileNetV2 est optimisé pour les environnements mobiles grâce à sa faible consommation de paramètres tout en maintenant des performances élevées.	3 575 878	Conçu pour l'efficacité des calculs
EfficientNetB1	EfficientNetB1 ajuste uniformément la profondeur, la largeur, et la résolution du réseau, trouvant un équilibre efficace entre précision et calcul.	6 904 717	Optimisé pour un bon équilibre précision/efficacité
SqueezeNet	SqueezeNet réduit le nombre de paramètres par l'utilisation de blocs "Fire", offrant une précision comparable à AlexNet tout en utilisant beaucoup moins de ressources.	738,502	Conçu pour une taille de modèle réduite

FIGURE 25 MODÈLES DE CLASSIFICATION DEEP LEARNING

Pour chaque modèle le processus d'entraînement était identique, contrôlé avec la même seed afin de garantir une comparaison équitable et de sélectionner le meilleur modèle.

8. Résultats et comparaison de modèles de deep learning.

8.1. Courbes d'entraînement

Les courbes d'entraînement montrent la précision et la perte de chaque modèle sur les données de validation. Elles permettent de comparer la manière dont les modèles convergent et s'améliorent au fil des epochs.

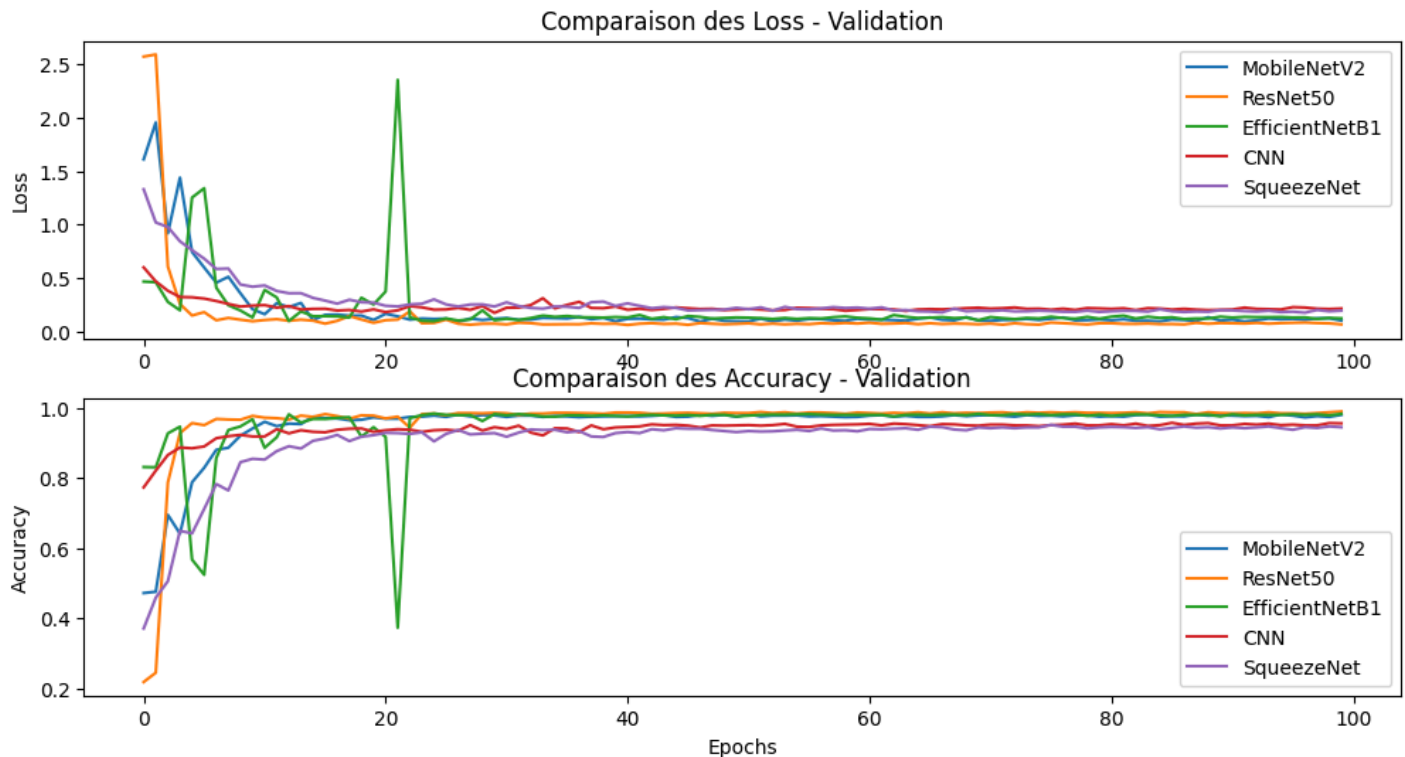


FIGURE 26 COURBES DE LOSS ET DE VALIDATION DES MODÈLES DE DEEP LEARNING

Ces courbes nous montrent que :

- ResNet50 se distingue par sa stabilité et sa précision élevée dès le début, faisant de lui le meilleur choix pour des tâches exigeant une haute performance.
- MobileNetV2 présente une instabilité initiale dans la perte, mais atteint une précision comparable en fin d'entraînement.
- EfficientNetB1, CNN, et SqueezeNet offrent une convergence régulière et stable, ce qui en fait des options solides pour des tâches générales.

8.2. Matrices de Confusion

Les matrices de confusion révèlent la performance de chaque modèle en termes de classification correcte et incorrecte pour chaque catégorie de document.

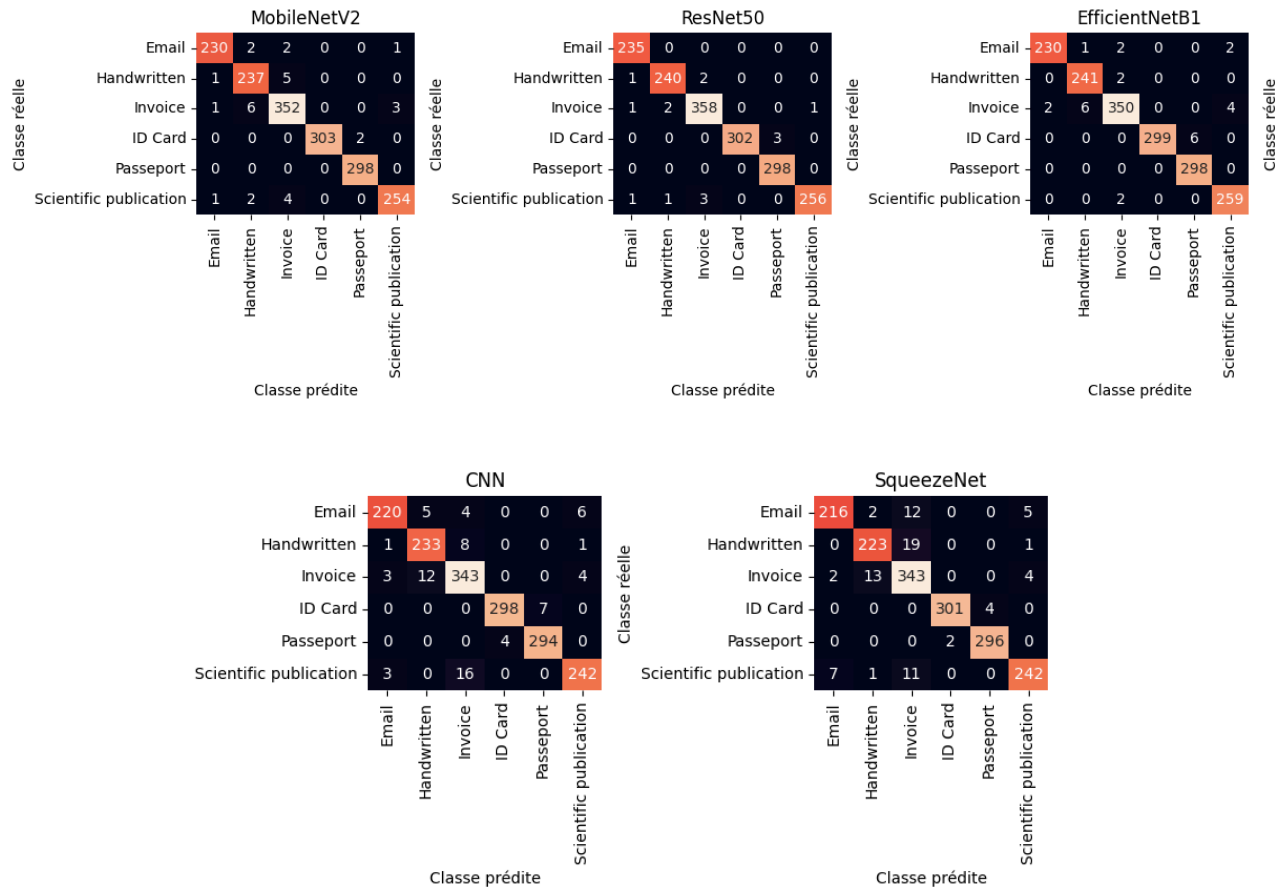


FIGURE 27 MATRICES DE CONFUSION DES MODELES DE DEEP LEARNING

L'analyse des matrices de confusions montre que le ResNet50 et EfficientB1 sont les plus performants, offrant un bon équilibre entre précision et rappel, avec très peu de faux positifs et faux négatifs.

8.3. Précisions des Modèles par Catégorie

Voici les précisions obtenues pour chaque modèle par catégorie de document :

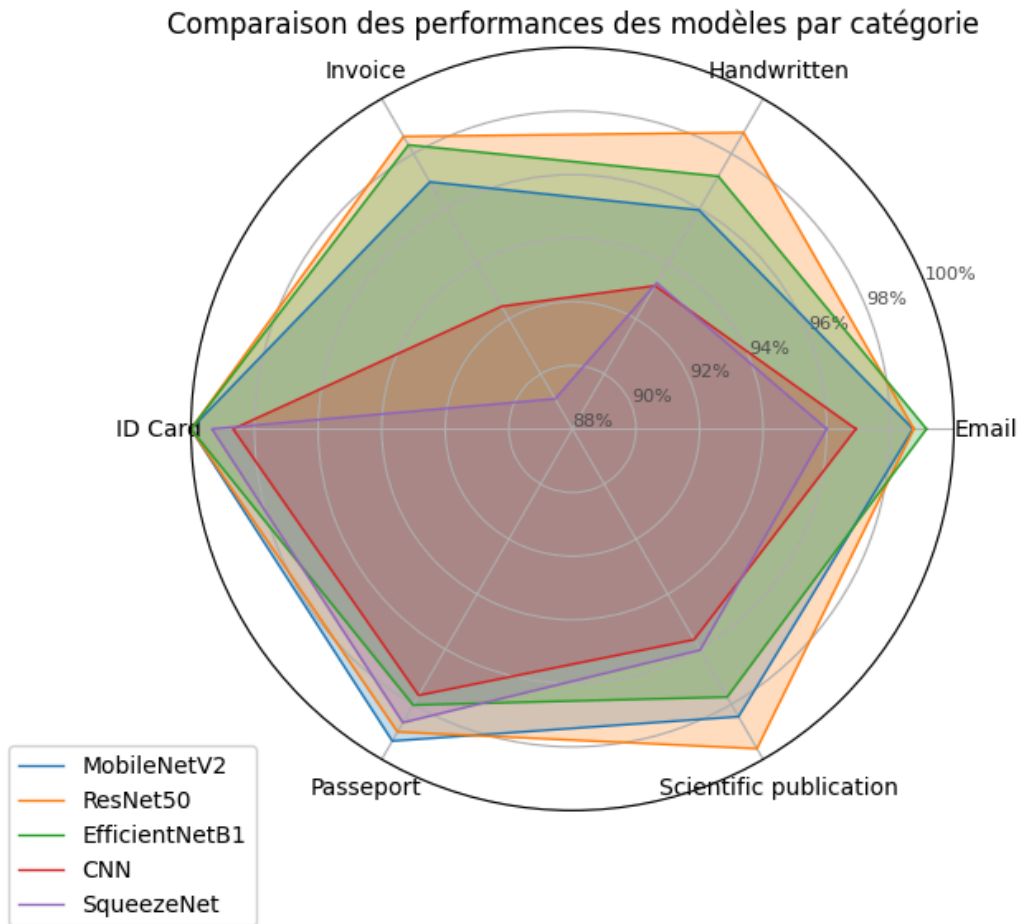


FIGURE 28 GRAPH RADAR DES PERFORMANCES DES MODÈLES EN FONCTION DES CATÉGORIES

Le radar plot ci-dessus nous permet de voir que le modèle ResNet50 offre les meilleures performances globales, avec une précision particulièrement élevée dans presque toutes les catégories. EfficientNetB1 et MobileNetV2 suivent de près avec des performances solides.

8.4. Évaluation du Temps de Prédiction

Dans cette section, une autre métrique a été analysée afin de juger les performances des modèles, à savoir le temps moyen d'inférence sur CPU et GPU pour chaque modèle.

Il est important de noter que les performances de ces modèles peuvent varier en fonction des spécificités de l'ordinateur utilisé, ci-dessous les caractéristiques de l'ordinateur utilisé pour cette partie.

- Carte graphique : NVIDIA GeForce RTX 2070 with Max-Q
- Processeur : Intel® Core™ i7-10750H CPU @ 2.60 GHz
- Capacité mémoire de la carte graphique : 8192 MiB
- Utilisation de la mémoire pendant les tests : 365 MiB
- Version du pilote NVIDIA : 522.06
- Version CUDA : 11.8

Modèle	Temps moyen d'inférence - CPU (s)	Temps moyen d'inférence - GPU (s)	Précision Moyenne (%)	Taille d'image
MobileNetV2	0.229	0.226	98.2	224x224
ResNet50	0.239	0.241	99.1	224x224
EfficientNetB1	0.246	0.244	98.4	224x224
CNN	0.182	0.184	95.7	224x224
SqueezeNet	0.213	0.207	95.1	224x224

FIGURE 29 TABLE DES PERFORMANCES DE VITESSE DES MODÈLES

On remarque que ResNet50 et EfficientNetB1 présentent des temps d'inférence légèrement plus élevés mais restent compétitifs en termes de précision. MobileNetV2 est rapide mais légèrement moins précis. Le CNN a le temps d'inférence le plus court mais offre une précision inférieure.

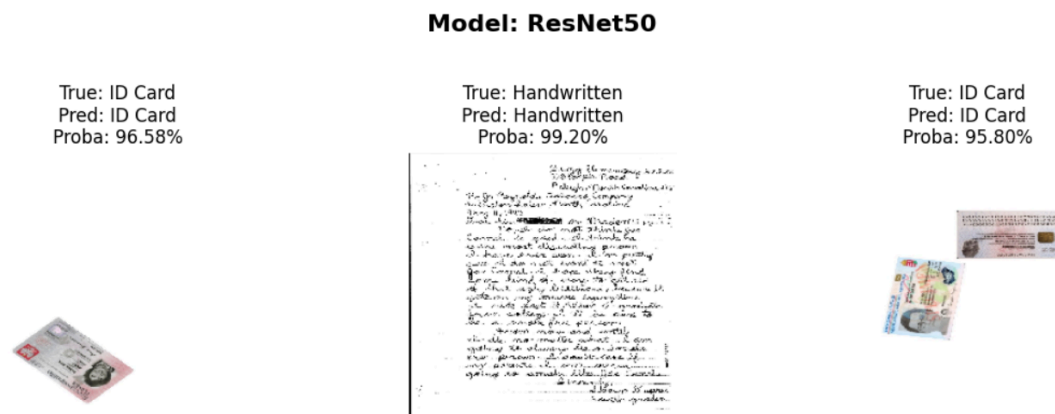


FIGURE 30 EXEMPLE DE PRÉDICTION RESNET50

8.5. Synthèse des résultats

ResNet50 se révèle être le modèle le plus performant. Sa profondeur et ses connexions résiduelles lui permettent de capturer des caractéristiques complexes tout en évitant les problèmes d'entraînement associés aux réseaux profonds. Le transfert learning depuis ImageNet renforce encore ses capacités de classification.

9. Difficultés rencontrées lors du projet.

Pour une personne débutante dans un projet de classification de documents utilisant des modèles de machine learning et de deep learning, les principaux verrous scientifiques rencontrés pourrait souvent être lié à l'un des éléments suivants :

9.1. Prétraitement des Données Textuelles

- **Sélection des bases de données pour le projet** : L'analyse des données que nous avons réalisée nous a permis de sélectionner avec soin les jeux de données à conserver. Nous avons décidé de retirer le jeu de données `data_02` en raison de son manque d'exploitabilité. Pour améliorer la précision des prédictions pour cette classe de documents, nous avons renforcé notre jeu de données avec le jeu de données `data_04`, qui contient principalement des cartes d'identité et des passeports.
- **Complexité du traitement du langage naturel (NLP)** : Le prétraitement des textes, incluant la tokenisation, la lemmatisation et la suppression des stopwords, peut être complexe. Les choix faits lors de ces étapes influencent directement les performances du modèle. Maîtriser ces techniques exige une compréhension approfondie des concepts linguistiques et des outils NLP.
- **Représentation des données** : Transformer les données textuelles en formats exploitables par les modèles, tels que les vecteurs d'embedding, est essentiel mais peut être complexe. Comprendre et utiliser des techniques comme TF-IDF ou Word2Vec, nécessitant de prendre de l'avance sur les modules de la formation

9.2. Classification des données textuelles en machine learning

- **Choix des modèles pour la comparaison :**

Pour nous, sélectionner les modèles appropriés était un challenge. Grâce à l'encadrement du projet, nous avons découvert l'outil 'Lazypredict', qui nous a grandement facilité la tâche en identifiant les meilleurs modèles.

- **Temps de calcul :**

Toutefois, l'utilisation de GridSearchCV pour le réglage des hyperparamètres a entraîné des temps de calcul très longs. Par exemple, nous n'avons pas pu exécuter le notebook "10-aasri-tfidf-modeling-text" sur notre machine en raison de sa lourdeur. Heureusement, notre collègue Daniel disposait d'un ordinateur suffisamment puissant pour le faire fonctionner.

9.3. Développement des modèles de deep learning : Classification d'images

Le principal défi de cette partie a été de réduire les temps de calcul à un niveau acceptable. Nous avons rencontré des difficultés avec les scripts, notamment pour configurer correctement les GPU afin de les exploiter efficacement. Cette configuration était essentielle, car exécuter nos modèles sur un CPU aurait nécessité plusieurs jours de calcul.

Par exemple, un processeur comme le Intel Core i7-10510U, avec une fréquence de base de 1.80 GHz et une fréquence turbo de 2.30 GHz, est conçu pour des performances équilibrées dans les appareils mobiles mais n'est pas optimisé pour les tâches de calcul intensif telles que l'entraînement de modèles de deep learning.

Cette situation a causé un certain retard dans nos calculs, mais nous avons néanmoins réussi à respecter les délais fixés.

9.4. Évaluation et Interprétation des Modèles :

Interprétabilité des résultats : Comprendre les raisons pour lesquelles un modèle fonctionne ou non est crucial pour optimiser ses performances. Cependant, les modèles de deep learning sont souvent considérés comme des "boîtes noires", ce qui complique leur interprétation. Pour surmonter ce défi, nous avons utilisé la technique Grad-CAM. Cette méthode permet de visualiser et d'interpréter les réseaux de neurones convolutifs (CNN), particulièrement pour les tâches de classification d'images. En générant des cartes de chaleur (heatmaps) pour les images traitées, Grad-CAM nous a aidés à comprendre les zones spécifiques sur lesquelles notre modèle se base pour effectuer ses prédictions

9.5. Prendre de l'avance sur la formation :

Pour réussir ce projet, il a été crucial de se préparer en avance sur les modules de cours et d'acquérir des compétences complémentaires. Nous avons pris l'initiative d'explorer des outils et des bibliothèques essentielles, telles que la bibliothèque `os` en Python et `pytesseract` pour l'extraction de texte. Cette préparation nous a permis de maîtriser les aspects techniques nécessaires et de mener à bien notre projet de classification de documents.:

10. Conclusion.

10.1. Bilan

Ce projet de classification de documents nous a permis d'appliquer un large éventail de connaissances et de compétences acquises tout au long de la formation. Le contexte de ce projet consistait à réaliser un modèle permettant de classifier des documents que l'on retrouve communément en entreprise.

Dans ce projet, nous avons d'abord abordé le sujet par une analyse exploratoire des données. À partir de ces analyses, nous avons ensuite réalisé du nettoyage de données et pu définir des fonctions nous permettant de standardiser celles qui ont été retenues. Puis vient la partie définition du modèle ou plutôt des modèles. Nous avons abordé cette problématique suivant deux approches différentes de classification :

- Classification à travers les méthodes de machine learning classique à partir des données textuelles.
- Classification à l'aide de deep learning.

Nous avons dans ces deux approches des subdivisions. En effet, nous n'avons pas juste choisi un modèle par approche. Nous avons appliqué et testé différents modèles afin de retenir ceux qui répondent le mieux à notre problématique.

Voici les grandes étapes pour les deux approches :

- Choix des modèles (machine learning: lazypredict | deep learning: recherche web)
- Choix des métriques
- Entraînement des différents modèles
- Comparaison des modèles

Le modèle basé sur les données textuelles (corpus et structure du texte) et répondant le mieux à notre problème est le VotingClassifier se basant sur les 5 meilleurs modèles issus de lazypredict avec une précision de 0.8776.

Pour la partie deep learning, nous avons réalisé un modèle depuis zéro puis 4 autres en transfer learning. Le modèle le plus performant est le ResNet50 suivant différentes métriques (voir partie 7.4).

Suivant le contexte de ce projet, bien que les performances des modèles soient très bonnes, voire excellentes, les objectifs n'ont pas été atteints pour différentes raisons :

- **Données biaisées** : La majorité des documents proviennent du domaine du tabac. Dès que l'on souhaite classer un document en dehors de ce domaine, on a de grandes chances d'avoir un résultat non satisfaisant.
- **Documents obsolètes** : Oui, la plupart des documents suivent un format datant de plusieurs décennies en arrière. On n'arrive pas à classer un email avec le format actuel.

10.2. Suite du projet

Plusieurs axes d'amélioration sont possibles dans notre cas pour une classification des documents couvrant un plus large champ de domaines et de périodes :

- **Ajouter des données plus récentes** : Intégrer des documents récents et variés dans le jeu de données afin de réduire le biais actuel et améliorer la généralisation du modèle.
- **Diversification des sources de données** : Collecter des documents provenant de différents domaines pour créer un dataset plus représentatif de la variété des documents rencontrés en entreprise.
- **Amélioration du prétraitement des données** : Améliorer les techniques de nettoyage et de standardisation des données pour mieux gérer les documents obsolètes et les différents formats de texte.
- **Augmentation des données** : Utiliser des techniques d'augmentation des données pour augmenter la diversité des documents d'entraînement, telles que la génération de documents synthétiques.

- **Exploration de modèles plus avancés** : Tester des architectures de deep learning plus récentes et plus complexes, telles que les modèles basés sur Transformers, qui ont montré des performances exceptionnelles dans diverses tâches de traitement de texte et d'images.
- **Développement d'une interface utilisateur** : Créer une interface utilisateur conviviale pour permettre aux utilisateurs finaux de classer facilement de nouveaux documents et de visualiser les résultats de classification.

En appliquant ces améliorations, nous pouvons espérer obtenir une classification plus précise et robuste des documents d'entreprise, couvrant un plus large éventail de domaines et de périodes, répondant ainsi mieux aux besoins des utilisateurs finaux.

11. Bibliographie

- Image Classification on ImageNet (2023). <https://paperswithcode.com/sota/image-classification-on-imagenet>
- Image Classification (2024). <https://paperswithcode.com/task/image-classification>
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (2015). Deep Residual Learning for Image Recognition. <https://arxiv.org/pdf/1512.03385>
- Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, (2019). MobileNetV2: Inverted Residuals and Linear Bottlenecks. <https://arxiv.org/pdf/1801.04381>
- Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer, (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. <https://arxiv.org/pdf/1602.07360>
- PRADO - Registre public en ligne de documents authentiques d'identité et de voyage. <https://www.consilium.europa.eu/prado/fr/prado-start-page.html>
- Text extraction for OCR. <https://www.kaggle.com/datasets/manishthem/text-extraction-for-ocr>
- Adam W. Harley, Alex Ufkes, and Konstantinos G. Derpanis (2015). The RVL-CDIP Dataset. <https://adamharley.com/rvl-cdip/>
- <https://arxiv.org/abs/1905.11946> : EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (ICML 2019)
- DataScientest courses : <https://learn.data-scientest.com/>

12. Diagramme de gantt

	Janvier				Février					Mars				Avril			
	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15	s16	s17
Exploration des bases de données																	
Analyses et data visualisation sur les Images																	
Extraction de textes avec Tesseract																	
Analyses et data visualisation de tesseract																	
Préparation de données textuelles d'entraînement et de tests																	
Entraînement avec lazypredict																	
Recherche d'hyperparametres avec GridSearchCV																	
Analyses et comparaison de résultats.																	
Normalisation d'images PRADO																	
Développement de fonctions de transformations.																	
Développement de Dataset d'entraînement et de test																	
Entraînement de modèles de DeepLearning																	
Tests et comparaisons de modèles																	
Rédaction du Rapport																	
Développement de la démo Streamlit																	

	Mai					Juin				Juillet				Août				
	s18	s19	s20	s21	s22	s23	s24	s25	s26	s27	s28	s29	s30	s31	s32	s33	s34	s35
Exploration des bases de données																		
Analyses et data visualisation sur les Images																		
Extraction de textes avec Tesseract																		
Analyses et data visualisation de tesseract																		
Préparation de données textuelles d'entraînement et de tests																		
Entraînement avec lazypredict																		
Recherche d'hyperparametres avec GridSearchCV																		
Analyses et comparaison de résultats.																		
Normalisation d'images PRADO																		
Développement de fonctions de transformations.																		
Développement de Dataset d'entraînement et de test																		
Entraînement de modèles de DeepLearning																		
Tests et comparaisons de modèles																		
Rédaction du Rapport																		
Développement de la démo Streamlit																		