REDS

Image tagging sur des données provenant de Pixelfed

Nour BOUCHOUCHI, Sofia BORCHANI,
Guillaume FAURE, Garance LUCAS, Lydia ZIDAT

Encadrants :
Olivier Schwander,
Laure Soulier,
Christophe Bouder

Septembre 2023 - Février 2024

Table des matières

I. Extraction des données

1. Objectifs du Dataset

Nous assistons à la croissance sans précédent du nombre de ressources multimédias personnelles, telles que les photos. Cette augmentation massive d'images nécessite des techniques d'accès aux images efficaces. L'association d'étiquettes aux images, également connue sous le nom de "image tagging", tente de décrire une image en utilisant un ou plusieurs concepts textuels pour refléter son contenu visuel, et est une des méthodes les plus efficaces. Cette méthode peut être utilisée pour rechercher une image notamment sur les réseaux sociaux.

Notre objectif est de créer un dataset d'images étiquetées par un tag unique et d'entraîner un modèle supervisé de machine learning à tagger automatiquement des images avec une étiquette textuelle qui reflète le contenu visuel de l'image. Nous effectuerons cette tâche pour des tags correspondant à des hashtags fréquemment utilisés sur les réseaux sociaux.

2. Sources de Données et Techniques Utilisées

Pour construire notre dataset, nous avons exploré des images associées à des hashtags provenant d'un réseau social du Fediverse, en l'occurrence Pixelfed. Notre source d'images provient de l'instance Pixelfed accessible à l'adresse https://pixelfed.social.
Pixelfed est une plateforme libre et décentralisée en alternative à Instagram, axée exclusivement sur le partage d'images, ce qui permet d'espérer une bonne qualité des tags associés aux images.

Étant en possession d'une vaste quantité d'images et de tags (35 millions de publications sur Pixelfed [1]), nous avons dû effectuer une sélection. Pour identifier les hashtags pertinents, nous avons adopté une approche initiale consistant à extraire les tags les plus populaires.

Dans le cadre de cette démarche, nous avons effectué un échantillonnage prospectif correspondant aux 1000 derniers posts publiés en utilisant une méthode de scrapping sur le feed global de Pixelfed à l'aide de Selenium.
Ce processus nous a permis d'acquérir un total de 4301 hashtags, soit une

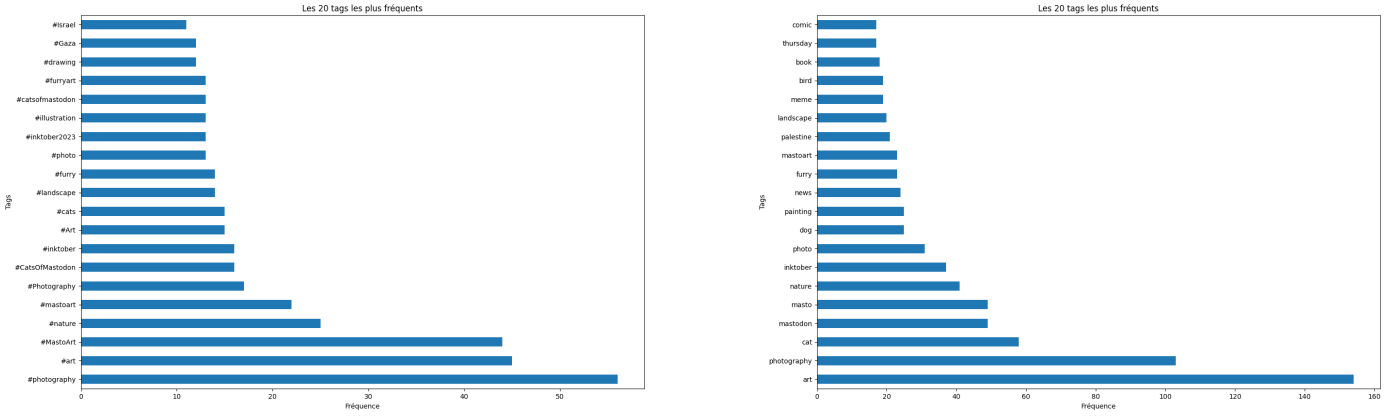
1. https://pixelfed.social/site/about

moyenne de 4,301 hashtags par publication. Au final, nous avons identifié 3106 hashtags uniques.

a. Étude des tags

Nous avons ensuite mis en œuvre une série de techniques de prétraitement sur cette liste dans le but de regrouper des hashtags similaires (par exemple, "cats," "cat," et "CatsOfMastodon") ce qui peut provoquer une modification de la liste de tags les plus fréquents (Figure 1). À l'issue de ces processus de prétraitement, nous avons obtenu un total de 5302 tags, dont 2899 étaient uniques. Les étapes de prétraitement ont été effectuées dans l'ordre suivant :

- Suppression des caractères spéciaux, des chiffres et ponctuations.
- Découpage des tags basé sur les majuscules : "CatsOfMastodon" devient 3 tags "Cats", "Of", "Mastodon".
  Afin de ne pas séparer les acronymes en tag d'une seule lettre, nous avons réalisé la séparation lorsqu'une majuscule était suivie d'une minuscule.
- Conversion en minuscule.
- Suppression des stopwords avec la librairie nltk en anglais.
- Lemmatisation avec nltk wordnet anglais.



(a) Top 20 hashtags

(b) Top 20 hashtags prétraités

FIGURE 1 – Comparaison entre les 20 premiers tags initiaux et les 20 premiers tags prétraités en fonction du nombre de postes

Le prétraitement des tags a engendré des modifications dans l'ordre des tags les plus populaires. À titre d'exemple, le tag 'cats' est passé de la 10ème

position dans le classement avant prétraitement à la 3ème position dans le classement après prétraitement. Cela peut s'expliquer du fait de la fusion de tags tels que 'cat', 'cats', ou 'CatsOfMastodon' en un unique tag 'cat', agrégeant ainsi les occurrences de ces tags initiaux.

De manière similaire, dans le classement des tags les plus populaires après les opérations de prétraitement, on observe l'émergence de tags que l'on ne retrouvaient pas initialement dans le classement, comme par exemple le tag 'palestine'. Cette observation peut s'expliquer par la présence d'un plus grand nombre de mots composés incluant ce terme. Suite au prétraitement, en particulier le découpage des tags composés de plusieurs mots, certains tags vont être regroupés sous un seul terme qui va apparaitre dans le classement après prétraitement. Par exemple, des tags tels que 'FreePalestine', 'Palestine' et 'PalestineWar', qui, pris individuellement, n'avaient pas une fréquence d'occurrence suffisamment élevée pour figurer dans le classement avant le prétraitement, parviennent, une fois combinés, à atteindre une fréquence suffisante pour être inclus dans le classement après le traitement.

b. Constructions des classes

Nous avons extrait les 100 tags les plus populaires (Figure 2) après le prétraitement réalisé sur les tags afin de pouvoir en extraire 5 pour constituer nos classes.

Notre objectif était de choisir un nombre de tags qui serait à la fois suffisant pour englober une diversité de sujets et assez restreint pour garantir un volume adéquat d'images par classe. Cette décision a été motivée par les contraintes en ressources liées à la collecte de données, qui peuvent représenter un coût significatif. Pour parvenir à cet équilibre, nous avons donc décidé de retenir 5 tags.

Cette sélection manuelle a été guidée par plusieurs critères visant à garantir la qualité des données. Nous avons privilégié des termes concrets, évitant ainsi des mots trop abstraits comme "mastodon" ou "travel", et nous avons opté pour des mots ayant des significations claires, en évitant les termes polysémiques. De plus, nous avons exclu les tags liés à des contextes temporels ou ayant une connotation politique tels que "war", "thursday" ou "halloween".

Nous avons choisi les classes parmi les 100 tags les plus fréquents, car de nombreux tags parmi les plus fréquents étaient relativement abstraits ou liés au contexte temporel ou encore avec une connotation politique (prendre dans

le top 20 des tags les plus populaires par exemple n'aurait pas été suffisant).

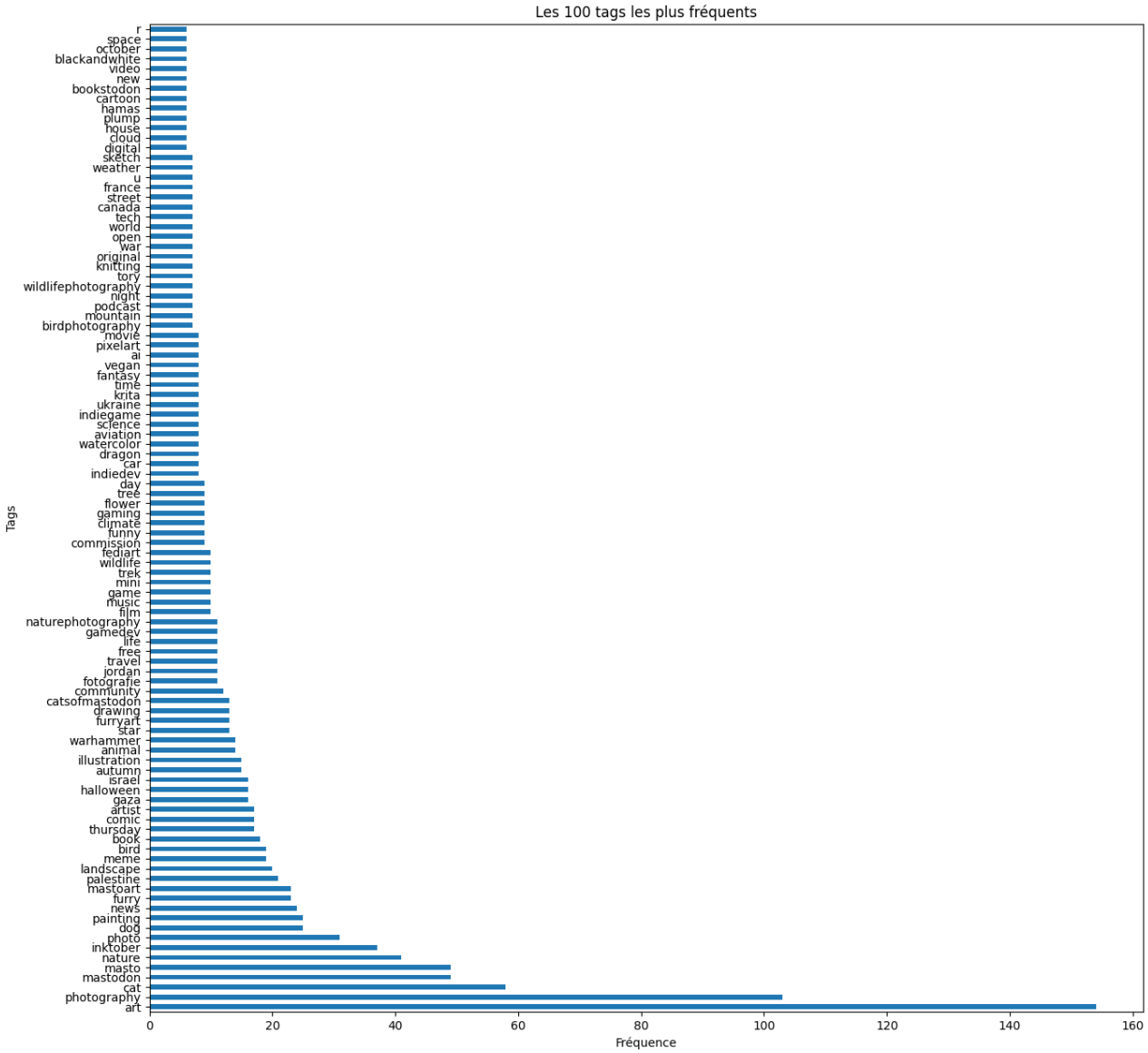


FIGURE 2 – Les 100 tags les plus fréquents

Après avoir écarté les tags comme expliqué précédemment, nous avons arbitrairement décidé de retenir les 5 tags suivants qui forment donc nos classes : cat, dog, flower, bird et car.

c. Scrapping des images en fonction des classes

Pour chacune de ces classes, nous avons choisi de collecter 1000 images. Il est essentiel de noter que nous avons fait en sorte de ne pas inclure d'images provenant de publications marquées comme sensibles, ni d'images de comptes privés qui ne sont pas partagées publiquement. Cette approche a pour but d'assurer le respect de la vie privée des utilisateurs tout en excluant tout contenu potentiellement inapproprié de notre ensemble de données. Par ailleurs, nous avons également écarté les images en noir et blanc, afin d'obtenir un jeu d'images homogène comportant le même nombre de canaux (3 pour RGB). Cette décision avait pour objectif de simplifier la tâche de classification ultérieure.

3. Analyse Descriptive des Données

a. Les tags

Les 5 tags retenus sont les suivants : cat, dog, flower, bird et car.

Ces tags représentent des entités concrètes, aisément identifiables sur le plan visuel, sans ambiguïté pour un utilisateur humain. Nous comptons deux espèces d'animaux domestiques (cat et dog), une catégorie d'oiseaux (bird), une variété de végétation (flower) et un type de véhicule (car).

Dans l'objectif de développer un processus de constitution du dataset capable de s'appliquer à un plus vaste ensemble de labels, nous entreprenons une analyse des relations sémantiques entre les catégories parallèlement à une évaluation de la pureté des classes, de manière à garantir l'absence de relations d'hyponymie entre celles-ci.

Nous avons étudié la similarité de Wu-Palmer entre chaque couple de mots grâce à WordNet afin de capturer les relations sémantiques entre les mots et ainsi avoir une première idée de la difficulté de la tâche de classification. Nous avons obtenu la matrice de similarité représentée par la Figure 3.

Nous obtenons une forte similarité entre "cat" et "dog"."bird" est assez similaire à "cat" et "dog". "flower" a une similarté modérée avec "cat", "dog" et "bird". "car" est le tag avec le moins de similarité vis à vis des autres tags.

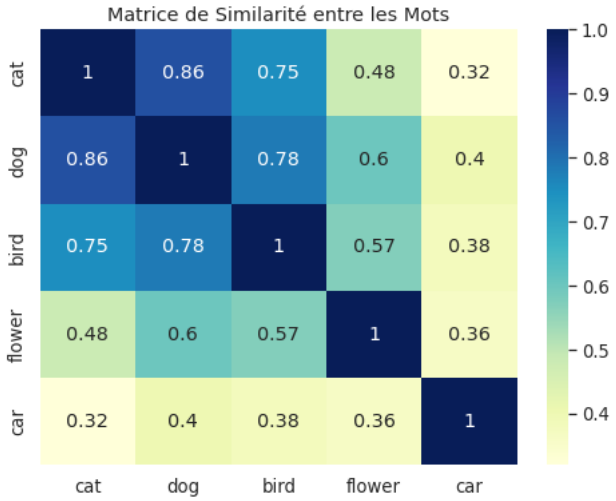


FIGURE 3 – Matrice de similarité sémantique entre les tags

Dans l'objectif d'un processus de construction d'un dataset qui puisse fonctionner sur un ensemble de tags plus large et afin de s'assurer de la pureté de chaque tag, nous avons vérifié grâce à wordnet qu'aucune des tags n'était l'hyponyme d'une autre. En effet, dans le cas où un tag est inclus dans un autre tag, cela présenterait un problème pour la tâche de classification. En observant les classes sélectionnées, la Figure 4 illustre qu'aucun des tags choisis n'est inclus dans un autre tag (il n'y a pas de liens entre les groupes).

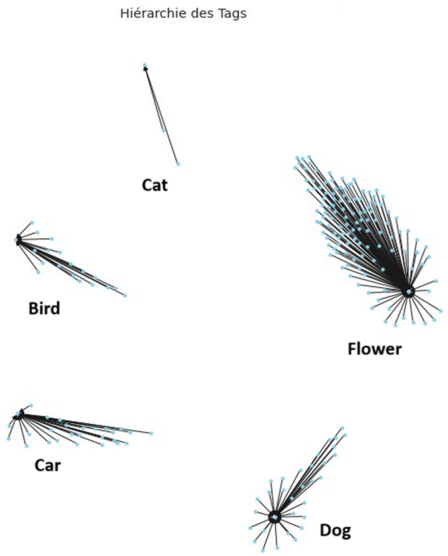


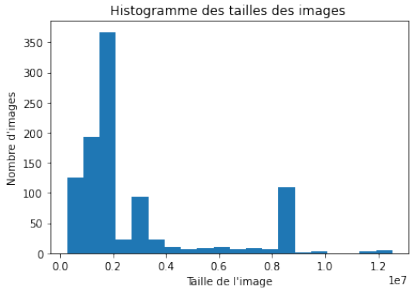
FIGURE 4 – Graphe représentant la hiérarchie entre les tags et leurs hyponymes

b. Les images

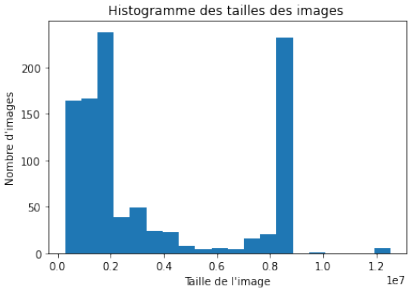
Pour atteindre notre objectif de 1000 images par classe, nous avons initialement extrait entre 1300 et 2000 images pour chaque tag parmis les plus récemment postées sur Pixelfed avec ce tag (nous n'avons pas récupéré les images contenant des *sensitive content*). Cette plage nous a permis de trier et de supprimer les images potentiellement non pertinentes, les doublons ou celles appartenant à plusieurs tags. Ensuite, chaque classe d'images a été soumise à une vérification humaine pour assurer leur homogénéité. Nous avons utilisé une fonction permettant d'afficher 5 images par ligne avec pour titre leur indice, et avons manuellement noté les numéros des images à supprimer. Nous avons ensuite conservé les 1000 premières images restantes.

En fin de compte, notre jeu de données final se compose de 5 000 images en couleur, réparties équitablement en 5 classes, chacune contenant 1 000 images.

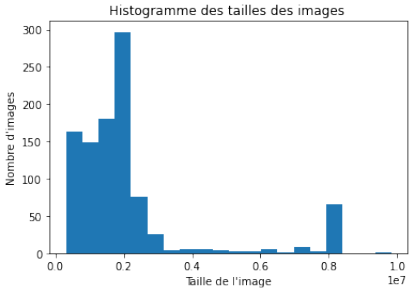
Nous avons généré un histogramme pour visualiser la distribution des tailles d'image en termes de nombre de pixels (Figure 5). Nous constatons que les images récupérées sont de tailles très variable. Afin de faciliter le traitement des données et de les rendre compatibles avec les types de modèles que nous prévoyons d'utiliser, nous avons décidé de nomaliser la taille des images en les redimensionnant à une dimension fixe.



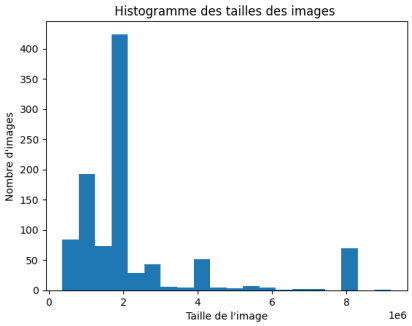
(a) Classe dog



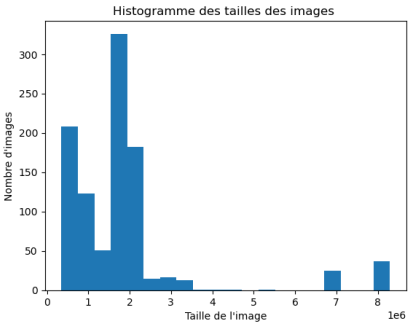
(b) Classe cat



(c) Classe flower



(d) Classe bird



(e) Classe car

FIGURE 5 – Histogrammes des tailles des images

La Table 1 présente des statistiques plus précises sur les tailles des images. Cela nous permet d'avoir une représentation plus fine des variations en terme de hauteur et de largeur des images. Le rapport d'aspect (Largeur/Hauteur) nous indique que les images sont en moyenne carrées mais que tout de même, la moitié des images sont rectangulaires.

Classes	Dog	Cat	Flower	Bird	Car
Mean Largeur/Hauteur	1.01	1.04	1.13	1.28	1.37
1^{er} quartile Largeur/Hauteur	0.75	0.75	0.75	1.00	1.33
4^{eme} quartile Largeur/Hauteur	1.33	1.33	1.49	1.50	1.50

TABLE 1 – Rapport d'aspect Largeur/Hauteur en fonction des classes

Après cette analyse, nous avons pris la décision de redimensionner toutes les images pour qu'elles aient la même taille. Nous avons choisi d'opter pour une taille de 224×224 pixels, considérée comme la norme pour des images nettes et qui nous permettra d'utiliser des modèles pré-entraînés sur ImageNet. En plus de réduire considérablement la taille de stockage du dataset (passant d'environ 35 Go à environ 6 Go), cette approche nous a permis de conserver les détails essentiels des images.

Deux options se sont présentées pour redimensionner les images à notre taille cible de 224×224 pixels : le redimensionnement pur et simple et le découpage des images. En découpant les images, il existait un risque de perte d'informations cruciales si le sujet de l'image n'était pas centré. Cependant, avec le redimensionnement pur et simple, il y avait un risque de distorsion, où certaines parties de l'image pouvaient être étirées ou compressées, entraînant une perte d'informations visuelles.

Nous avons opté pour le compromis suivant :

— Si les images sont carrées ou presque carrées, correspondant à aspect ratio largeur/hauteur entre 0.9 et 1.1, la déformation est inférieure à 10%. Nous redimensionnons directement l'image en 224×224 pixels sans découpage.

— En revanche, pour les images où l'aspect ratio n'est pas dans cette plage, nous avons choisi de découper de manière centrées la hauteur ou la largeur (selon la taille la plus petite), puis de redimensionner le résultat en 224×224 pixels.

Cette approche garantit une adaptation adéquate des images limitant la perte d'information en préservant leur contenu visuel central et limitant la déformation. Au total, 89 % des images ont bénéficié d'un découpage et d'un redimensionnement (Table 2).

Classe	Pourcentage
Cat	82
Dog	86
Bird	86
Flower	83
Car	97
All Classes	89

TABLE 2 – Tableau présentant le pourcentage d'images ayant été découpées et redimensionnées

Nous avons vérifié manuellement pour chaque classe que le redimensionnement avait été effectué correctement. Comme on peut le voir (Table 3), la hauteur et la largeur minimales et maximales sont bien de 224 pixels, ce qui équivaut à des images de 50 176 pixels.

	Min Largeur	Max Largeur	Min Hauteur	Max Hauteur
All Classes	224	224	224	224

TABLE 3 – Largeur et Hauteur en pixel après redimmensionnement

Finalement, nous avons étudié les variations d'intensité des pixels dans un soucis de normalisation. Les images présentaient une dynamique moyenne de 251, un écart type de 10 ainsi qu'une valeur maximale de dynamique de 255. La Table 4 illustre ce propos.

	Red	Green	Blue
Min Intensité	0	0	0
Max Intensité	255	255	255

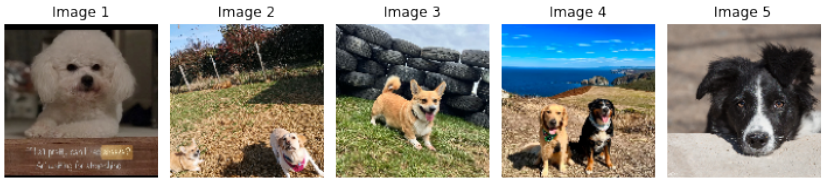
TABLE 4 – Minimum et maximum d'intensité des pixels en fonctions des différents canaux

Afin de normaliser l'intensité des pixels de toutes les images pour qu'elle se situe entre 0 et 1, nous avons effectué une opération de division par 255. On peut voir (Table 5) qu'après normalisation, l'intensité minimale est 0 et l'intensité maximale est 1.

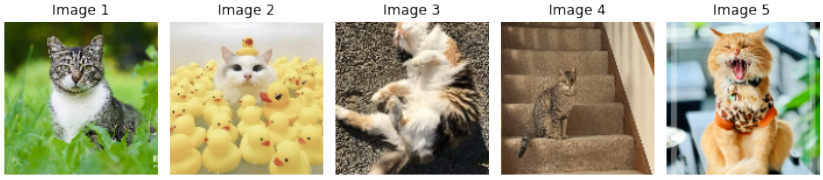
	Red	Green	Blue
Min Intensité	0	0	0
Max Intensité	1	1	1

TABLE 5 – Minimum et maximum d'intensité des pixels en fonctions des différents canaux après normalisation

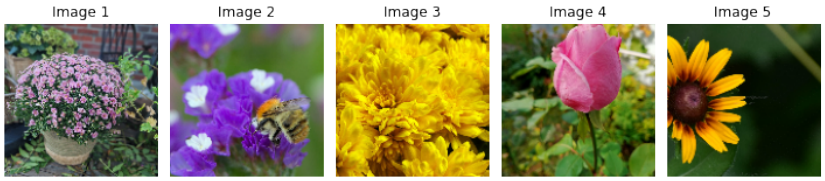
Après l'application de ces divers traitements, nous avons constitué une base de données comprenant 5000 images, avec 1000 images pour chacune des 5 classes. Toutes ces images ont été normalisées et redimensionnées à une taille de 224 × 224, comme illustré sur la Figure 6.



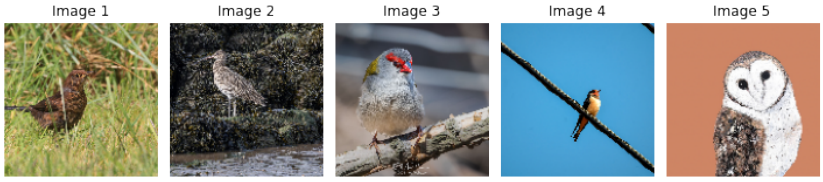
(a) Classe dog



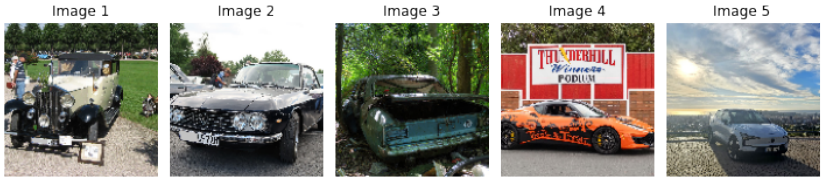
(b) Classe cat



(c) Classe flower



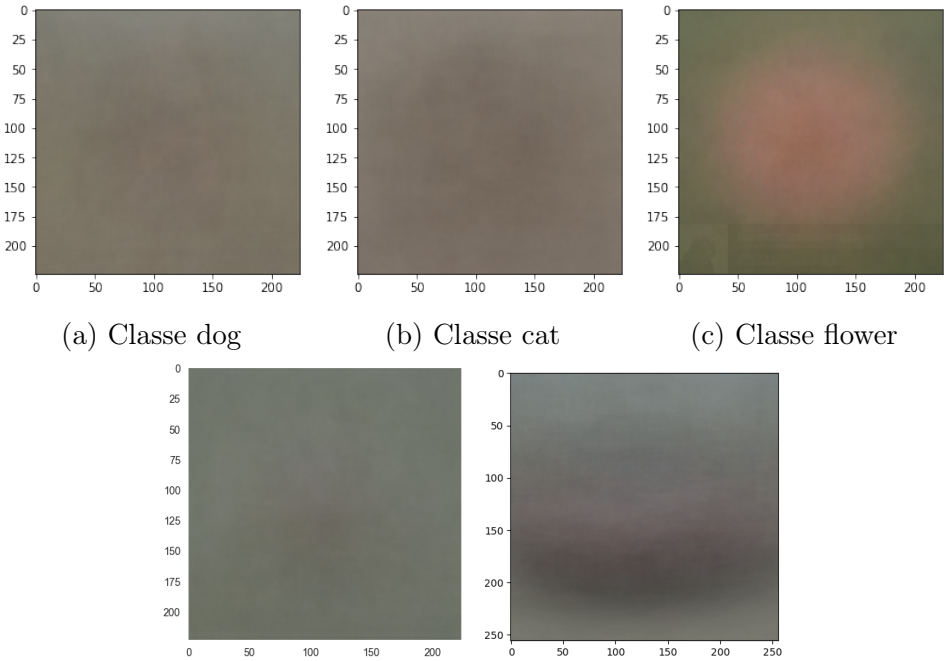
(d) Classe bird



(e) Classe car

FIGURE 6 – Exemple d'images appartenant à chaque classe

Nous avons examiné l'image moyenne par classe (Figure 7). Nous observons que pour les classes "dog", "cat", et "bird" l'image moyenne n'apporte que peu d'informations distinctives. En revanche, pour les classes "flower" et "car" un pattern de classe commence à émerger, significatif pour l'oeil humain. Cela peut être expliqué par le fait que les images de la classe "car" et "flower" sont fréquemment centrées et ont une uniformité dans leur composition. En revanche, les images des classes "cat", "dog" et "bird" varient davantage en termes d'environnement et de position, ce qui conduit à une image moyenne moins distinctive pour ces classes.



(a) Classe dog (b) Classe cat (c) Classe flower

(d) Classe bird (e) Classe car

FIGURE 7 – Images moyennes représentatives de chacune des classes

II. Expérimentations

1. Choix de modèles

Nous avons choisi VGG16 et ViT-B16. En effet, ces modèles sont réputés pour leur efficacité dans des tâches telles que le tagging d'images et ont été initialement pré-entrainés sur ImageNet. Ces deux modèles offrent une modularité qui leur permet d'être utilisés comme base pour des techniques de transfert learning.

Pour ajuster ces réseaux neuronaux à notre ensemble de classes spécifiques, une modification structurelle a été apportée : la couche de sortie originale de chaque modèle, communément désignée sous le terme de "head", a été retirée. En remplacement, un classifieur Support Vector Machine (SVM) linéaire a été intégré. Cette substitution vise à adapter la tâche de classification des modèles en les rendant conformes aux spécificités de notre jeu de données cible, à savoir 5 classes dans notre dataset contre 1000 pour ImageNet.

Modèle	VGG16	ViT-B16
Architecture	Convolutionnelle	Vision Transformer
Lancement	2014	2020
Nombre de Paramètres	138 M	86 M
Nombre de couches	16	16
Prétraitement	-	Patch + position embeddings
Taille des filtres	3x3	-
Facteur de réduction de pooling	2	-
Vitesse d'apprentissage	Lent	Rapide
Atout	Compréhension locale	Compréhension globale

TABLE 6 – Comparaison des caractéristiques entre VGG16 et Vit-B16

2. Validation croisée

Nous avons procédé à une validation croisée en 5 folds afin d'évaluer la performance du modèle de manière robuste et afin d'éviter la dépendance de la division des données.

a. Méthodes d'évaluation

Afin de réaliser nos expérimentations sur ce dataset, nous avons tout d'abord mélangé aléatoirement les images, puis nous avons partitionné notre jeu de données avec 90% en Train et 10% en Test. Pour l'entrainement, nous avons effectué une cross validation avec 5 folds dans le but d'avoir une idée de la robustesse de notre modèle grâce à son accuracy moyenne et à sa variance. Puis, afin d'évaluer la généralisation du modèle nous l'avons ré-entrainé sur l'ensemble de l'échantillon d'entraînement avant de l'évaluer sur l'échantillon de test.

Afin d'évaluer les modèles, nous avons utilisé différentes métriques, à savoir l'accuracy par fold, l'accuracy moyenne sur les folds, la variance de l'accuracy entre les folds, et l'accuracy en Test (Tables 7 et 8). Nous avons également effectué les matrices de confusions (Figure 8) des classes et le temps d'entraînement des modèles.

3. Analyse des résultats

a. Performances des modèles

Le temps d'entraînement de VGG16 était en moyenne 3 fois plus long que celui de ViT-B16 avec respectivement un temps d'exécution de 35.06 secondes et de 11.436 secondes.

	Accuracy VGG16
Fold 1	0.900
Fold 2	0.912
Fold 3	0.920
Fold 4	0.921
Fold 5	0.907
Average	0.912
Variance	6.22×10^{-5}

TABLE 7 – Accuracy VGG16

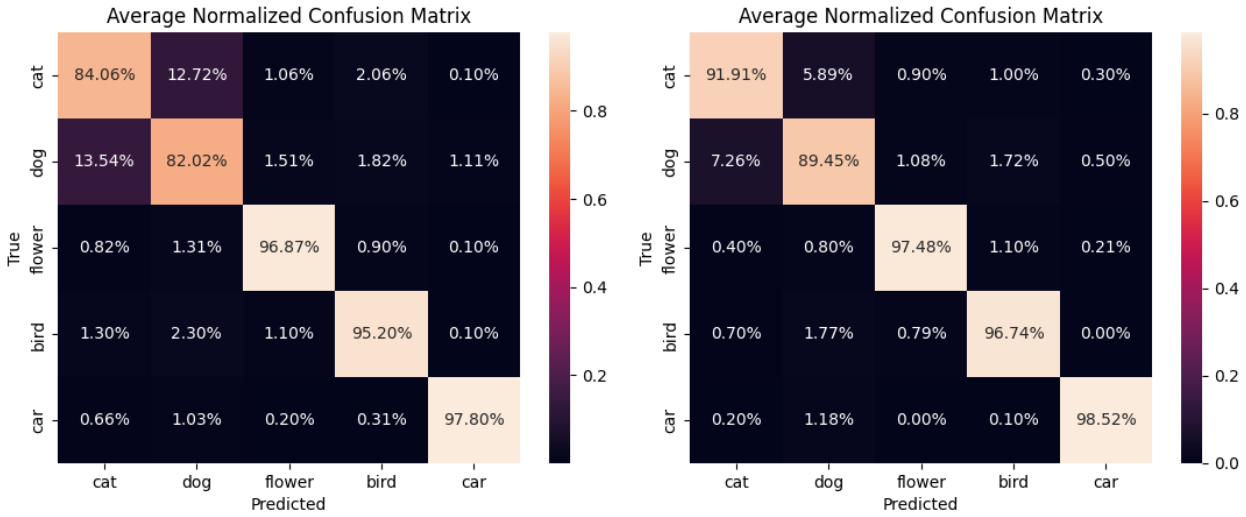
	Accuracy ViT-B16
Fold 1	0.946
Fold 2	0.956
Fold 3	0.945
Fold 4	0.940
Fold 5	0.953
Average	0.98
Variance	3.32×10^{-5}
Test	0.970

TABLE 8 – Accuracy ViT-B16

On observe que l'accuracy moyenne obtenue pour le modèle ViT-B16 lors de la validation croisée est supérieure à celle obtenue par le modèle VGG16. En effectuant un test de Wilcoxon-Mann-Whitney on trouve une p-value de 0.0079, on peut donc conclure que la performance moyenne de ViT-B16 est significativement plus élevée sur l'ensemble des folds que celle de VGG16. Les variances sont relativement faibles dans les deux cas, ce qui montre une stabilité des performances des modèles entre les différents plis. Enfin, les temps d'entraînement et d'exécution de ViT-B16 étaient deux fois plus rapides que ceux de VGG16. Le modèle ViT-B16 semble donc présenter de meilleures performances que VGG16 sur notre cas d'étude.

b. Matrices de confusion

Nous avons tracé les matrices de confusion pour les modèles VGG16 et ViT-B16 (Figure 8) sur les résultats moyens obtenus sur les 5 folds. Elles nous permettent d'évaluer la performance des modèles en examinant la répartition des prédictions par rapport aux classes réelles.



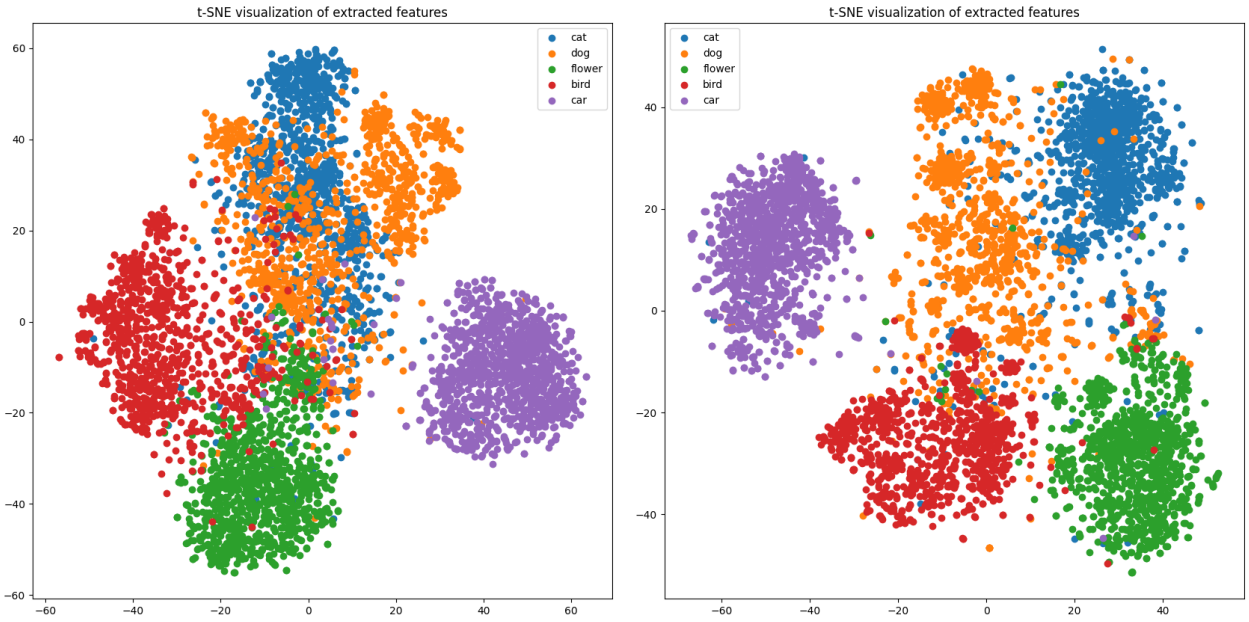
(a) Matrice de confusion VGG16 (b) Matrice de confusion ViT-B16

FIGURE 8 – Comparaison des matrices de confusion entre VGG16 et ViT-B16

Les valeurs sur les diagonales principales représentent les taux de classifications correctes pour chaque classe. Ces valeurs sont élevées, ce qui indique que les modèles ont bien performé. On note toute fois de meilleurs performances pour ViT-B16. Une confusion notable est observée entre les images des classes "cat" et "dog". Ces confusions sont réduites pour ViT-B16.

c. Analyse des clusters via t-SNE

Nous avons réalisé une analyse des clusters pour chaque modèle à l'aide de t-SNE (Figure 9). Cette technique explore la distribution des points de données dans un espace de dimension réduite, nous permettant ainsi de visualiser les regroupements et les relations entre différentes classes de nos modèles.



(a) t-SNE VGG16 (b) t-SNE ViT-B16

FIGURE 9 – Comparaison des t-SNEs entre VGG16 et ViT-B16

Pour VGG16, les points de données semblent se regrouper par catégorie, mais avec une séparation moins nette entre certains clusters. Il existe des chevauchements prononcés, en particulier entre les classes "cat" et "dog", ce qui est cohérent avec les confusions observées dans la matrice de confusion.

Par rapport à VGG16, ViT-B16 montre une meilleure séparation des clusters, indiquant que le modèle extrait des caractéristiques plus discriminantes qui facilitent la distinction entre les catégories.

Les deux modèles présentent une meilleure séparation pour certaines classes, notamment celle de "car", ce qui suggère que les caractéristiques spécifiques à ces classes sont bien saisies par les deux modèles.

III. Mise en production

Lors de cette étude, une comparaison a été réalisée entre deux modèles de deep learning, VGG16 et ViT-B16, pour une tâche de tagging d'images. Cette étude a permis de mettre en évidence la supériorité du modèle ViT-B16. En termes d'accuracy globale, ViT-B16 a démontré des performances significativement supérieures (p-value < 0.05). De plus, il a présenté une plus grande stabilité entre les différents folds avec une variance deux fois moins élevée que pour le modèle VGG16. On note également une meilleure discrimination entre les classes ayant une forte proximité sémantique comme les classes "cat" et "dog". Cette idée est d'ailleurs renforcée par l'analyse t-SNE qui a révélé que ViT-B16 parvenait à séparer les données plus efficacement que VGG16. L'ensemble de ces observations nous ont conduites à décider de ne conserver que le modèle ViT-B16 pour la mise en production.

Concernant les avantages, le modèle ViT-B16 se distingue par ses performances et sa robustesse, le classant ainsi parmi les modèles *state-of-the-art* dans le domaine. Il s'agit d'un modèle particulièrement adapté à notre tâche puisqu'il a été pré-entraîné sur ImageNet qui est composée de classes sémantiquement proches et il y a peu de *domain shift*. Le *transfer learning* a ainsi permis d'effectuer un entraînement rapide, environ 11 secondes pour 4 000 images, sur notre dataset ce qui est un avantage considérable lors de ré-entraînements. Par ailleurs, la tâche de scrapping est une opération relativement simple à exécuter au vu de l'implémentation d'une fonction qui requiert uniquement le nom du tag à scrapper.

Cependant, on peut également noter plusieurs pistes d'amélioration. Tout d'abord, une confusion notable a été observée dans la classification de tags proches, comme "cat" et "dog", nécessitant une analyse plus approfondie pour d'autres classes sémantiquement proches telle que "fox" avant la mise en production car cela pourrait entraîner une baisse des performances du modèle. Le fait de se limiter à un nombre restreint de classes, comparativement à des datasets plus vastes comme CIFAR-100, peut également restreindre la portée du modèle qui avait initialement pour objectif de faire de la recommandation de hashtags. L'absence d'automatisation dans la sélection de ces hashtags pour définir des classes concrètes est également un frein à l'expansion de la taille du vocabulaire de notre modèle. En effet, nous nous sommes placés dans un cas où les tags retenus étaient concrets, indépendants du contexte, et sans intersection ou inclusion (il n'y a pas de classe 'cat' et 'animal'). Dans l'optique d'un cas d'utilisation réel, une solution envisageable

serait le relâchement de ces contraintes. On pourrait alors par exemple choisir de faire de l'étiquetage multi-labels : le modèle retourne tous les tags pour lesquels le modèle a une confiance supérieure à un certain seuil pré-défini.

Enfin, concernant le traitement des images, leur taille uniforme (224x224 pixels) pose un défis pour l'inférence sur des images de tailles différentes, nécessitant soit un redimensionnement, soit un vote majoritaire sur une fenêtre glissante.

Pour la mise en production, il sera essentiel de réitérer régulièrement le processus d'entraînement afin d'adapter le modèle aux tendances actuelles. Une étude concernant l'évolution des tendances et la fréquence de publication serait alors indiquée pour optimiser les intervalles entre les entraînements. Par ailleurs, la recommandation de tags aux utilisateurs, plutôt qu'un tagging automatique sans supervision, est une approche préconisée. En effet, cela permettrait de limiter l'introduction de biais lors de réentraînements ultérieurs : sans cela on risquerait de finir par utiliser majoritairement des données étiquetées par notre modèle ce qui n'est pas souhaitable.

En conclusion, le modèle ViT-B16 customisé pour notre tâche se révèle être une solution efficace, présentant des performances élevées pour le tagging d'images et qui est adapté au besoin de ré-entraînement régulier. Toutefois, des améliorations sont nécessaires pour gérer les limitations actuelles, notamment en termes d'inférence sur des classes proches, ou encore concernant la diversité des hashtags considérés. Ces ajustements permettront d'optimiser davantage le système pour son application dans des contextes réels et diversifiés.