

Projet Deep Learning: Classification des Styles Vestimentaires

Lisa Weisbecker, Thierry XU, Tom Bonetto, Clémentine Grethen, Leo Meissner

3 novembre 2023

1 Introduction

La classification de styles vestimentaires est un problème de vision par ordinateur qui consiste à attribuer une étiquette de style vestimentaire à une image. Ce problème est particulièrement intéressant pour les applications de e-commerce, où les sites Web peuvent recommander des produits similaires en fonction des préférences des utilisateurs. Le but de notre projet est donc de reconnaître le style vestimentaire qu'une personne porte à partir d'une photo, on se concentrera sur 5 styles vestimentaires :

1. Vintage
2. Classique-Minimaliste
3. Gothique
4. Street-wear
5. Chic

2 Construction de la base de données

Chaque membre du groupe a construit la base d'image d'un style, la méthode de construction (les outils, les sources,...) est donc différente pour chaque style. Nous avons décidé de choisir environ 900 images par style en prenant des styles distincts afin d'éviter les confusions. Pour l'étiquetage des données, nous avons placé chaque style dans un dossier, se situant dans le dossier *src/images*. Un autre dossier *donnees* décrit la répartition des différentes images selon les données d'entraînement, de validation et de test.

Voici le lien github vers notre BDD :

https://github.com/TItygrosminet/Projet_Deep_Learning.git

2.1 Style Vintage

Le style vestimentaire vintage est un style de mode qui fait référence aux vêtements et accessoires d'époques passées, généralement des décennies allant des années 1920 aux années 1980. Ce style se caractérise par des vêtements rétro, souvent colorés et avec des motifs originaux, des coupes classiques et des détails qui rappellent l'histoire de la mode. Le style vintage est souvent associé à une attitude nostalgique, à une passion pour l'histoire de la mode et à une volonté de se démarquer du style de la mode contemporaine. Il peut inclure des pièces d'époque authentiques, des vêtements réédités ou encore des créations inspirées de styles passés.

En effet, le style vestimentaire vintage connaît une popularité croissante depuis quelques années, avec de plus en plus des gens qui cherchent à intégrer des pièces vintage dans leur garde-robe. Cette tendance est en partie due à l'intérêt pour la mode durable et éthique, avec de nombreux consommateurs qui préfèrent acheter des vêtements d'occasion plutôt que de soutenir l'industrie de la fast-fashion. Le style vintage est également considéré comme intemporel et élégant, offrant une alternative aux tendances de la mode éphémères.

Pour construire la base de données associée au style vintage, nous avons utilisé l'extension Chrome *Image Downloader* pour récupérer les images affichées sur le site web. Quasiment un tiers de la base provient de Google Image via les recherches "style vestimentaire vintage homme" et "style vestimentaire vintage femme". Les deux tiers restant (soit environ 600 images) ont été trouvés grâce à Pinterest. Ce style s'étendant sur une longue période, nous avons fait des recherches par tranches de dizaines d'années : "90s fashion style", "70s fashion style", etc et en alternant homme/femme. La difficulté ici n'était pas de trouver les images, mais plutôt de les filtrer, en effet le style vintage étant revenu à la mode, il est souvent mélangé avec d'autres styles et l'image devient peu pertinente d'un point de vue classification. Cette étape d'élimination des images peu pertinentes a été faite à la main en ayant au préalable éliminé les doublons avec *Duplicate Photo Finder* sur Windows.



FIGURE 1 – Exemples d’images pour le style vintage

2.2 Style Classique-Minimaliste

Le style minimalist-classique est un type de mode qui se concentre sur des vêtements simples, intemporels, élégants et que l’on peut porter tous les jours. Les couleurs utilisées dans le style minimalist-classique sont généralement neutres, telles que le blanc, le noir, le gris, le beige et le marine, et sont souvent utilisées en combinaison pour créer un look cohérent. On retrouve parfois quelques couleurs un peu plus flash. Les vêtements ne sont pas surchargés d’ornements ou de motifs, mais plutôt conçus pour être élégants et fonctionnels. Le côté classique du style minimalist-classique se manifeste dans les choix de vêtements intemporels tels que les chemises, les pantalons bien ajustés, les blazers, les trench-coats, des chaussures classiques ou des jeans.

Pour la construction de notre base, nous avons pris des images sur des sites de vêtements classiques tels qu’Uniqlo, sur des réseaux sociaux en utilisant des recherches par mots-clés (Pinterest, Instagram,...) ou bien directement sur Google Image. Pour enregistrer les images plus rapidement nous avons utilisés des extensions Chrome telles que "image downloader". Nous avons ensuite revu les images une à une pour éliminer celles qui n’étaient pas satisfaisantes (Style trop proche d’un autre, ...). Pour finir, la vérification des doublons a été faite grâce au Gemini2 sur mac.



FIGURE 2 – Exemples d’images pour le style gothique

2.3 Style Gothique

Le style gothique est un style vestimentaire atypique qui prend inspiration dans le punk ou "l'emo". La couleur principale retrouvée est très majoritairement le noir, mais certaines tenues comportent du rouge ou du bleu foncé. Il s'agit souvent de robes très détaillées ou d'habits déchirés ou révélateurs. Au niveau des textures, les matières reconnaissables sont le cuir, la dentelle où des motifs quadrillés (de type résille). On remarque aussi des accessoires très caractéristiques comme des chaînes et des bijoux métalliques.

La construction de cette base de données s'est avérée assez compliquée car ce n'est pas un style qu'on peut trouver "tout fait" sur des modèles en boutique en ligne populaire. Les photos trouvées en ligne montrent rarement le corps en entier. Nous avons rencontré également de certaines difficultés à intégrer des modèles masculins à la base de données, car c'est un style qui semble majoritairement porté par des femmes.

Pour réaliser la base de données, nous avons utilisé des extensions comme "bulk image downloader" ou "download album from instagram" pour le téléchargement, puis un logiciel permettant de détecter les doublons dans le dossier.

Les images viennent principalement de Pinterest, d'Instagram et de Google Image.



FIGURE 3 – Exemples d’images pour le style classique

2.4 Style Street-wear

Le style streetwear est un style vestimentaire qui a émergé de la culture urbaine et hip-hop dans les années 80 et 90. Les vêtements portés dans ce style sont souvent des vêtements décontractés et confortables, tels que des hoodies, des t-shirts oversize, des jeans baggys ou des pantalons de survêtement. Quant aux couleurs, celles fréquemment associées au streetwear sont le noir, le blanc, le gris, le kaki, le bleu marine, le rouge et le jaune. Les imprimés, tels que les logos, les graphismes ou les motifs de camouflage, sont également courants dans le streetwear. Les bijoux portés dans le style streetwear sont souvent simples et discrets, tels que des chaînes en argent ou en or, des bracelets en cuir ou en

tissu, ou des boucles d'oreilles en or ou en acier inoxydable. Le style streetwear est souvent associé à une attitude décontractée et cool, et est populaire auprès des jeunes et des personnes soucieuses de leur apparence dans les milieux urbains.

Pour construire la base de données associée au style streetwear, nous avons surtout pris des photographies sur *Pinterest*. En effet, ce dernier est un site qui regroupe beaucoup plus de photographies de style vestimentaire et les référence mieux que sur Google Image. L'extension Chrome *Image Downloader* a également été utilisé pour récupérer les images affichées sur le site web. Par la suite, les images ont été revu une à une pour éliminer les images non conformes, c'est à dire, les images où le corps n'était pas visible entièrement par exemple. Enfin, la dernière étape consistait à retirer tous les doublons avec *Duplicate Photo Finder* sur Windows et à renommer toutes les images avec le label *Streetwear*.



FIGURE 4 – Exemples d'images pour le style streetwear

2.5 Style Chic

Le style chic est également synonyme d'élégance et de sophistication. Les tissus utilisés sont souvent de qualité supérieure tels que le satin, la soie ou encore le velours. Les accessoires sont également très importants pour compléter la tenue. Pour les femmes, des chaussures à talons hauts, des bijoux étincelants et des pochettes élégantes sont des incontournables. Pour les hommes, une belle montre, une cravate assortie ou une pochette de costume peuvent faire la différence. Le style chic est donc un style vestimentaire qui nécessite une attention particulière aux détails pour être réussi. C'est un choix de mode idéal pour les événements importants où l'on souhaite se démarquer tout en restant élégant et raffiné.

Afin d'acquérir nos données, nous utilisons une extension chrome de téléchargement d'image tel que image Downloader. Nous avons d'abord commencé par des recherches googlées telles que « costume » ou « tenue de soirée » mais la plupart des images ne correspondaient pas à ce dont nous avions besoin. Des recherches de compte instagram et de n'ont pas fonctionné non plus. Nous nous sommes alors dirigé vers des sites de prêt à porter où chaque image correspondait à notre intention de recherche. Nous avons ainsi pu récupérer facilement toutes les images dont nous avions besoin pour cette classe.

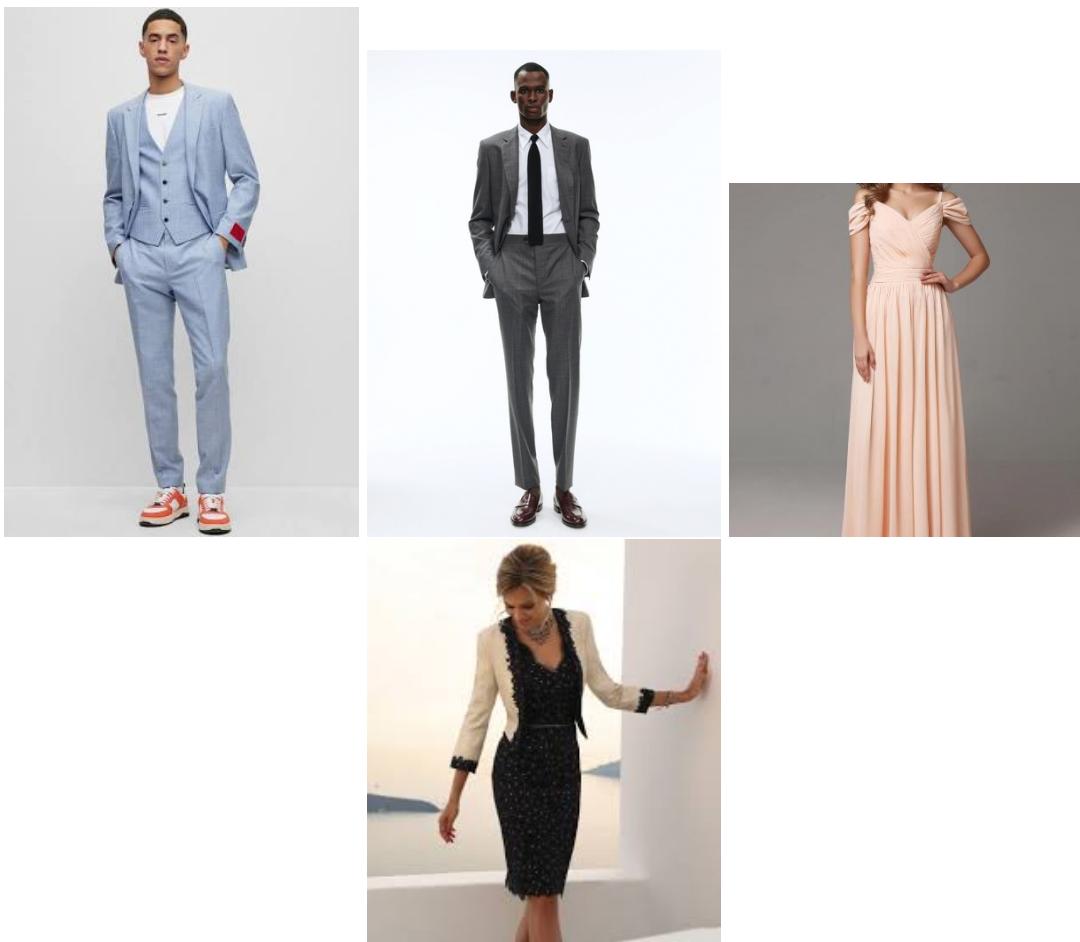
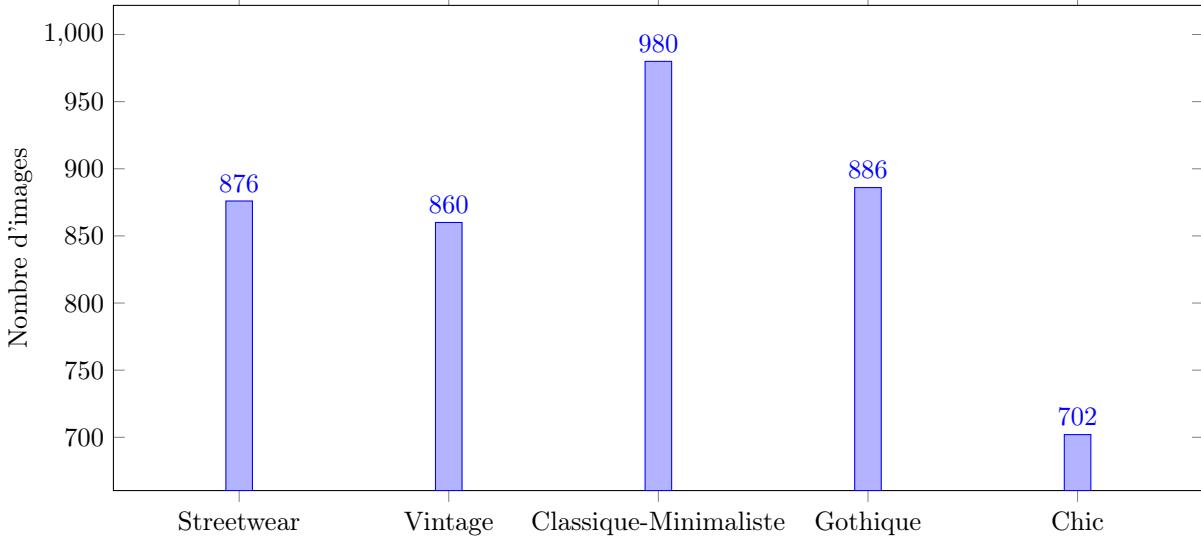


FIGURE 5 – Exemples d'images pour le style chic

2.6 Répartition des données

Le graphique ci-dessous illustre la répartition du nombre d'images en fonction du style dans notre base de données.



La répartition des données en données d'entraînement, de validation et de test est importante pour entraîner et évaluer un modèle de deep learning. En général, on utilise une proportion de 70% pour les données d'entraînement et 15% pour les données de validation et de test. Cette répartition permet d'avoir suffisamment des données pour entraîner le modèle, ajuster ses paramètres et tester sa performance sur des données inconnues.

Pour cela, nous avons divisé chaque classe avec des proportions 70/15/15. Ensuite, les différentes subdivisions d'images sont réparties dans des dossiers *train*, *validation* et *test*, contenant à leur tour un dossier pour chaque classe.

3 Pronostic

Pour résumer, nous allons donc avoir une classification de style vestimentaire à 5 classes. Nous pensons que notre sujet est un problème de classification assez complexe. À titre de comparaison, on peut prendre la classification de vêtements, qui elle se concentre sur la reconnaissance de la catégorie d'un vêtement spécifique (t-shirt, robe, etc), les catégories de vêtements sont clairement définies, car ils ont des caractéristiques distinctes pour les différencier. En revanche, notre classification elle, implique la reconnaissance d'un ensemble de vêtements qui peuvent être regroupés en fonction de leur style général. Cependant, avec les styles vestimentaires, il est bien plus difficile d'identifier des caractéristiques uniques, car ceux-ci peuvent avoir des vêtements présents dans différents styles, on peut retrouver les pantalons cargo dans le style vintage comme dans le style street-wear, deux vêtements ayant le même style peuvent avoir des couleurs, motifs, textures différentes. La frontière entre les différentes classes est assez floue, même pour un humain, il est parfois difficile de savoir qu'elle est le style vestimentaire d'une personne. Les poses des modèles peuvent également varier considérablement, ce qui peut affecter la reconnaissance des vêtements. Par exemple, un sweat facilement identifiable dans une certaine pose ou un certain angle de photo peut être difficile à reconnaître dans une autre pose ou un autre angle de photo.

Il y a de fortes chances que la classe Classique-Minimaliste fasse baisser la précision avec les classes

Vintage et Chic, car ces 3 styles présentent souvent des points communs, en revanche les classes Gothique et Streetwear devrait normalement être les styles classés avec la meilleure précision due à leurs caractéristiques singulières.

Pour ce qui est des résultats : Dans un premier temps, si on utilise le transfer learning en utilisant par exemple la base de convolution VGG16 sans la réentraînée, nous pensons qu'il sera difficile de dépasser les 55% de précisions. On pourra peut-être augmenter de quelques pourcents la précision en ré-entraînant la base avec un faible taux d'apprentissage. Il existe sûrement des bases de convolution plus efficaces avec lesquelles on pourrait espérer atteindre 60% voire un peu plus. Avec notre propre base de convolution que l'on aura adaptée au mieux à notre problème, on pense avoisiner les 65% et si on rajoute à cela la data augmentation afin d'augmenter de manière artificielle la taille de notre base de données, nous pensons qu'il est possible d'atteindre les 70%. En revanche, il nous semble compliqué au vu de notre sujet d'atteindre par exemple 80% comme on pourrait avoir sur une classification binaire du type chien/chat.

4 Implémentation

4.1 Rognage

La plupart des images continues dans notre base de données possèdent un fond, qui diffère d'une image à une autre. Un réseau de neurones s'entraîne et apprend toutes informations sur l'entièreté des images. Étant donné que nous aimions classer des styles vestimentaires, il est nécessaire de rogner toutes les images pour ne conserver que le corps. Ainsi, celui-ci n'apprendra que des informations issues du style vestimentaire de la personne.

Pour cela, une des premières solutions a été d'utiliser la vision par ordinateur avec la bibliothèque d'OpenCV sur Python. Néanmoins, nous remarquons vite les limites de cette méthode avec uniquement une centaine d'images rognées sur l'ensemble de notre base de données. De plus, les images rognées ne conservent pas forcément le corps de l'individu sur l'image.

Dans notre cas, les images contiennent uniquement une personne sur un fond. L'objectif est donc de détecter la boîte englobante de la personne et rogner le reste de l'image. Ainsi, nous limitons le nombre de pixels liés au fond. Pour la détection d'objet, un des réseaux de neurones le plus performant est YOLO. Il est possible de récupérer ce réseau de neurones directement sur le github de son créateur Joseph Redmon. Les différents poids sont également récupérables sur le site de celui-ci. Nous avons donc fait passer chacun de nos images dans le réseau de neurones qui détecte les boîtes englobantes des individus. Le réseau de neurones fournit les coordonnées de la boîte sur l'image. Nous avons donc rogné l'image à l'aide ces dernières.

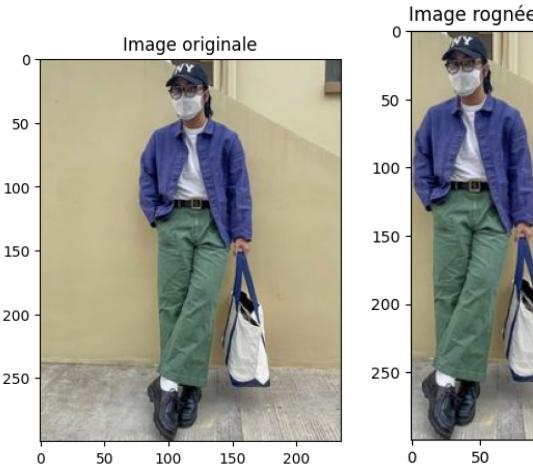


FIGURE 6 – Image non rognée et rognée

4.2 Mélange des genres dans les différents ensembles

Initialement, nous avions défini l'infrastructure de notre base de données comme sur la figure 7. Sur l'arborescence, au lieu de titi, tata et tutu, nous avons les différentes classes : vintage, classique, streetwear, gothique et chic.

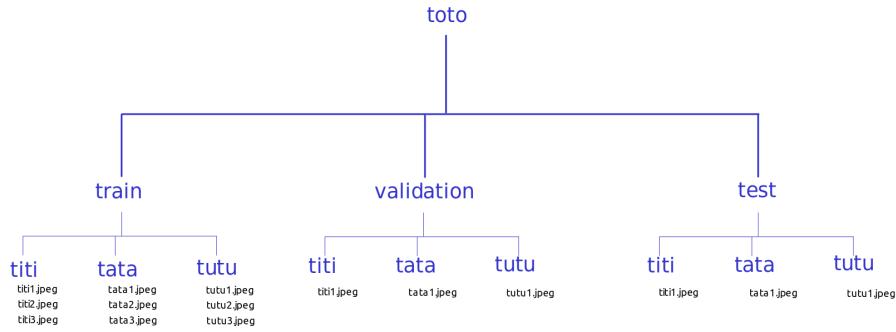


FIGURE 7 – Architecture de la base de données

Définir ainsi notre base de données peut tendre à apprendre les styles selon le genre homme-femme. En effet, pour certains styles comme le style "chic", la tenue diffère énormément selon le genre. Il faut donc essayer d'avoir une proportion égale d'homme et femme dans les données d'entraînement. Or ce critère n'a pas été vérifié lors de l'établissement de notre base de données. Pour cela, nous avons restructuré notre base de données avec un dossier par classe contenant toutes les images de la classe. Par la suite, pour diviser en données d'entraînement, de validation et de test, nous avons utilisé la fonction *train_test_split*. Cette dernière possède un paramètre *random_state* pour contrôler le mélange appliqué aux données avant de les diviser.

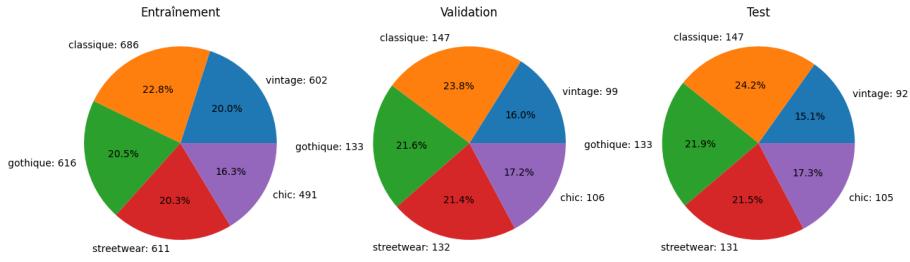


FIGURE 8 – Répartition des classes en fonction des ensembles

Nous obtenons ainsi une répartition du nombre d’images par classe et par ensemble de données. Nous respectons donc bien une répartition d’environ 70/15/15.

4.3 Transfer learning

Dans la résolution d’un problème de vision par ordinateur, on a besoin dans notre modèle d’une base de convolution pour extraire les caractéristiques importantes dans les images qui permettront par la suite de classifier. Le choix de l’architecture de la base de convolution n’est pas une tâche facile c’est pourquoi le transfer learning peut être particulièrement utile. En utilisant le transfer learning, on bénéficie des architectures de base de convolution pré-entraînée qui ont déjà été optimisées sur de vastes ensembles de données, telle que imageNet. Ces bases pré-entraînées on fait l’objet d’études poussées pour obtenir les meilleures performances et les meilleurs résultats possibles, il serait donc dommage de ne pas s’en servir. Il existe de nombreuses bases de convolution avec des tailles, profondeurs et des paramètres très différents.

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	99 MB	0.749	0.921	25,636,712	168
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

FIGURE 9 – Modèles pré-entraînés librairie Keras

4.3.1 Comparaison de différentes bases de convolution

Pour savoir quelle base pré-entraînée est la plus adaptée à notre problème nous avons créé un script *Transfer_Learning.ipynb* qui permet de tester la précision de différents modèles.

Les bases de convolution sont importées à partir de la librairie de Keras. Ces bases pré-entraînées sont déjà optimisées et prêtées à l'emploi. On définit une fonction pour créer et compiler un modèle en utilisant la base de convolution spécifiée. La fonction gèle les couches de la base de convolution pour empêcher leur réentraînement. Ensuite, on rajoute la partie dense pour effectuer la classification spécifique au problème. Cette base possède une couche dense de taille 512 avec une régularisation l1, une couche de dropout à 30% et enfin la couche de sortie avec l'activation softmax. Sachant que l'on va entraîner le modèle plusieurs fois de suite, nous avons opté pour une tête de classification simple et peu profonde pour réduire les temps de calcul au maximum.

Le modèle est compilé avec l'optimiseur *Adam* et un taux d'apprentissage à $3e-4$, une fonction de perte *sparse_categorical_crossentropy* et le calcul de précision associé *sparse_categorical_accuracy*. On itère ensuite sur chaque base de convolution. À chaque itération, on appelle la fonction pour créer un modèle en utilisant la base de convolution correspondante, le modèle est ensuite entraîné sur les données d'entraînement et évalué sur les données de test. Une fois l'entraînement terminé on récupère la précision pour faire la comparaison et le modèle est sauvégarde pour pouvoir appliquer le fine-tunning après.

Il est important de noter que nous avons dû utiliser une taille d'images de 75 pixels car certains modèles n'acceptent pas en dessous. On notera également que le nombre d'époques a été fixé à une valeur haute de 200 car on utilise de l'early stopping avec un delta de $1e-3$ et une patience de 20 époques.

Pour le fine-tunning on réitère sur les bases de convolution en chargeant les modèles entraînés auparavant, on dégèle les couches du modèle sauf celles de la partie dense et on réentraîne cette fois-ci avec un taux d'apprentissage à $1e-5$. Nous le verrons dans la suite du rapport mais c'est le modèle MobileNetV2 qui a obtenu les meilleures performances (vitesse d'entraînement et précision).

4.3.2 Solution élaborée à base de transfer learning

On va maintenant essayer d'obtenir la meilleure précision possible avec MobileNetV2. On écrit un nouveau script *MobileNet.ipynb* dans lequel on passe la taille des images à 128 pixels, malheureusement on ne peut aller au-delà car google colab n'arrivera pas à supporter l'exécution. La tête de classification est maintenant plus dense avec une première couche de taille 1024 puis une de taille 512 puis une de taille 256 puis une de taille 128 et enfin la couche de sortie de taille 5 puisqu'on a 5 classes. Chaque couche dense possède une régularisation de type l2 et on intercale une couche de dropout à 30 % entre chacune d'entre elles. Le taux d'apprentissage ne change pas de même pour les différentes métriques, on change seulement le nombre d'époque à 300 car on n'a remarqué que pendant le fine-tunning il arrivait qu'on atteigne les 200 époques avant que l'early stopping s'active. MobileNetV2 possède moins de paramètres que les autres modèles connus comme ResNet50 ou VGG16, on gagne donc beaucoup de temps à l'entraînement sans pour autant avoir de moins bon résultat.

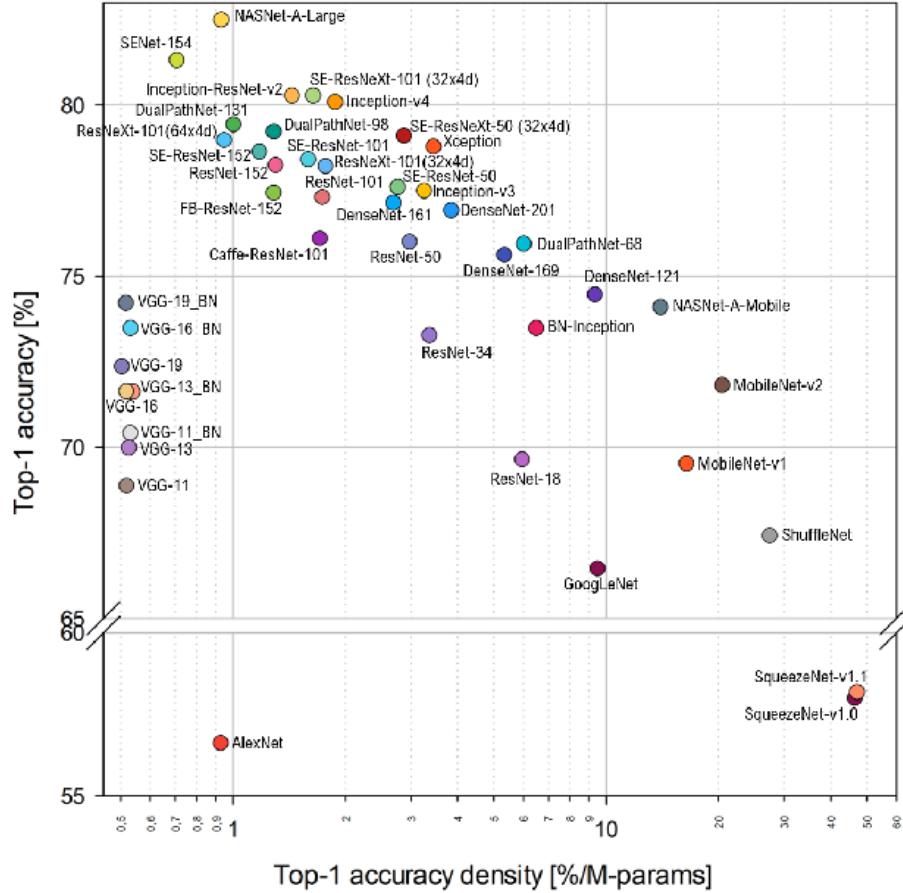


FIGURE 10 – La densité de précision mesure l’efficacité avec laquelle chaque modèle utilise ses paramètres

5 Résultats

Nous allons dans cette partie présenter les différents résultats que nous avons obtenus lors de nos différentes implémentations.

5.1 Comparaison transfer learning

Pour ce qui est de la comparaison des différentes bases de convolution, on a comparé 5 modèles : ResNet50 qui est très long à entraîner, VGG16 long aussi, Xception, NASNetMobile plutôt rapide et enfin MobileNetV2 rapide également. Nous avons un premier résultat de comparaison pré fine-tunning présenté ci-dessous.

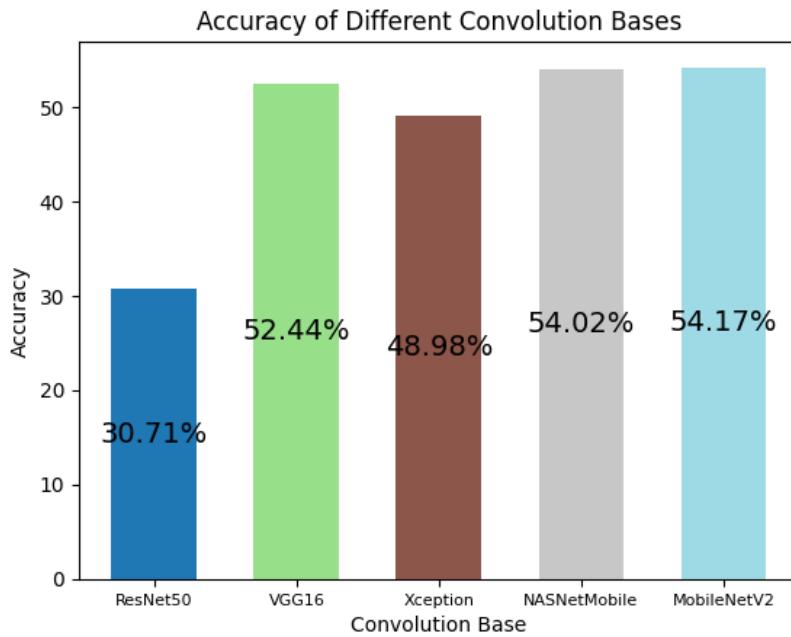


FIGURE 11 – Précision des modèles avant fine-tunning

Comme on peut le voir, ResNet50 a à peine dépassé les 30% sur cet entraînement, non pas qu'il ne puisse faire mieux puisqu'en relançant un nouvel entraînement il a atteint les 50% mais plutôt qu'on est tombé vraisemblablement dans un minimum local. Cela est arrivé plus de fois qu'on le pensait notamment avec le modèle InceptionV3 que nous avons finalement remplacé par NASNetMobile car celui-ci se bloquait souvent autour des 15% précisions sans trop de raison particulière. Hormis ces cas assez rares, dans l'ensemble tous les réseaux sont aux mêmes niveaux avec des résultats un peu meilleurs pour VGG16, NASNetMobile et MobileNetV2.

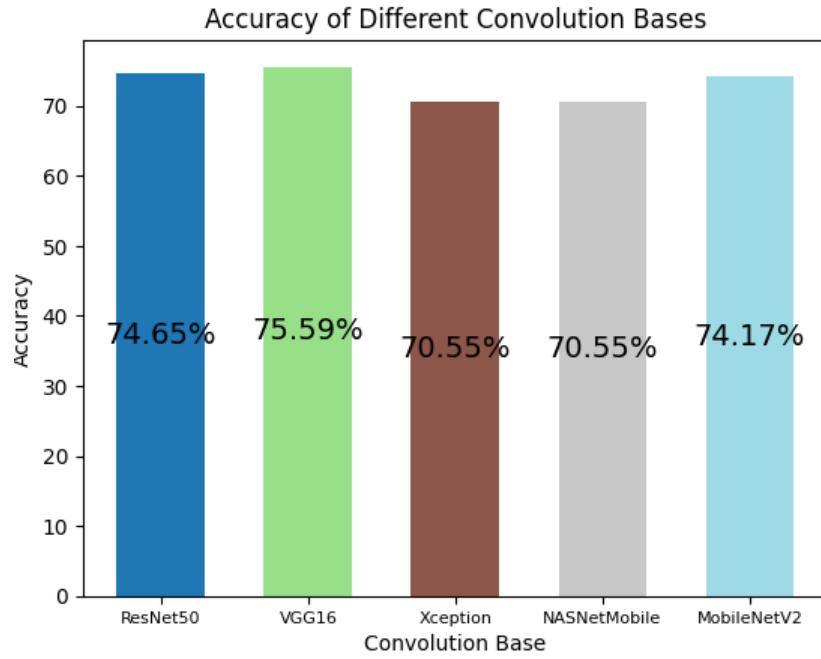


FIGURE 12 – Précision des modèles après fine-tunning

Après le fine-tunning ResNet50 rattrape son retard, tous les modèles atteignent les 70 % de précision, on pourrait se contenter de n’importe lequel d’entre eux mais pour des raisons de temps de calcul on va choisir MobileNetV2 pour la suite car c’est le meilleur en matière d’efficacité.

5.2 Résultats sur l’ancienne base de données ni rognées ni mélangées

Tout d’abord, nous avons tenté d’obtenir des résultats sur notre ancienne base de données. Ces premiers résultats pouvaient nous servir de repère de base. L’idée était de partir de ceux-ci et tenter de les améliorer.

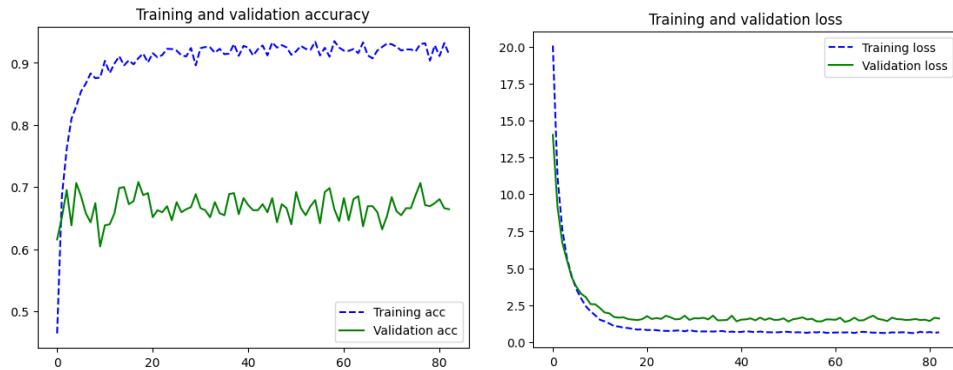


FIGURE 13 – Précision et perte avant fine-tunning

Le premier entraînement avec transfer learning, avant fine-tunning, s’arrête légèrement après 80 epochs. Nous remarquons que la courbe de perte stagne assez rapidement. Au bout de 20 epochs, nous avons déjà atteint la perte minimale et la précision maximale possible sans réentrainé la base de convolution. Ainsi, l’early stopping arrête l’entraînement. On obtient une précision d’environ 90%

sur les données d'entraînement et de 68% sur les données de validation. Avec la courbe de perte, on remarque qu'il n'y a pas trop de surapprentissage.

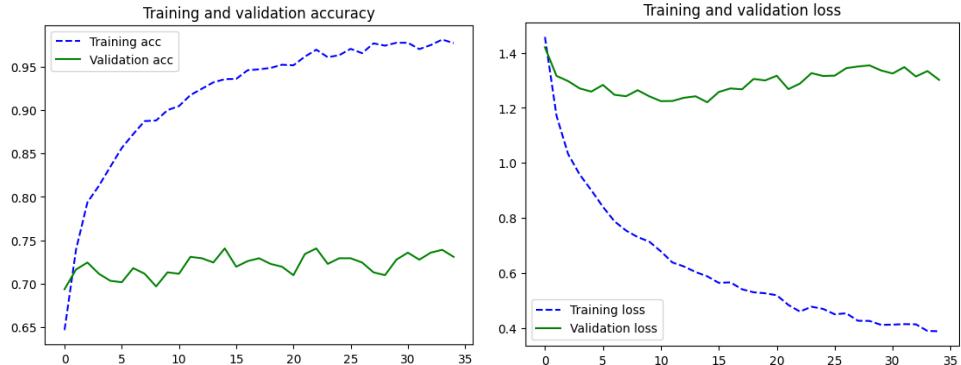


FIGURE 14 – Précision et perte avant fine-tuning

Avec les mêmes paramètres d'early stopping, le fine-tuning prend uniquement 35 epochs. La précision et la perte sont tous les deux meilleurs. On a environ une précision de 98% sur les données d'entraînement et de 73% sur les données de validation.

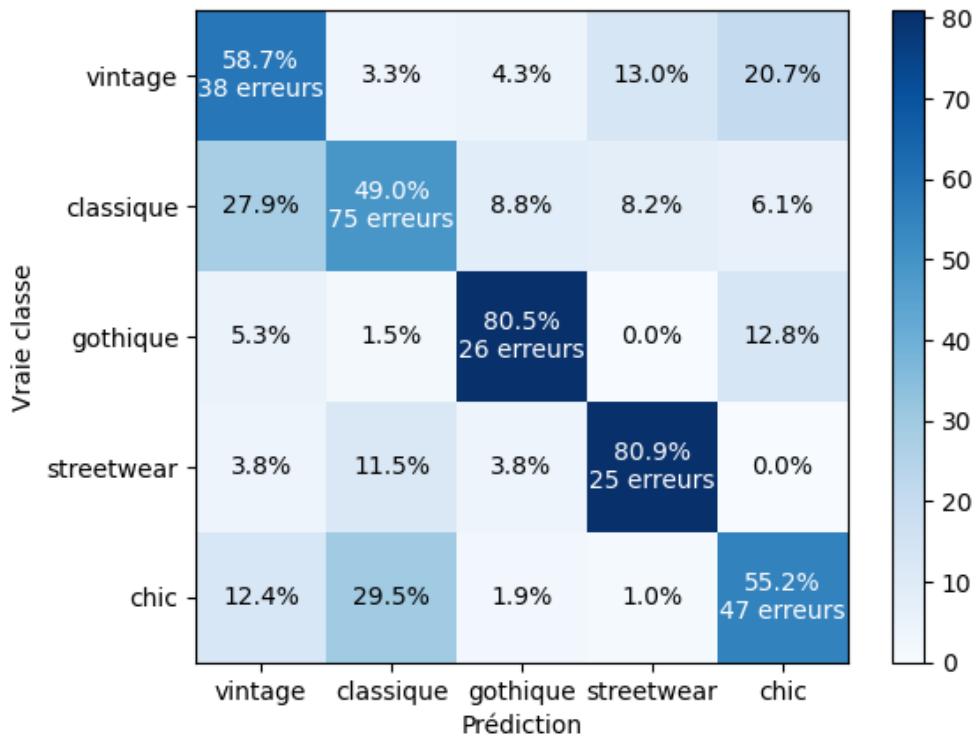


FIGURE 15 – Matrice de confusion sur l'ancienne base de données

Comme dit précédemment lors de nos pronostics, nous avons des erreurs de prédiction sur les classes vintage, classique et chic. Par exemple, il y a 27.9% des images classiques qui finissent par être classé vintage et 20.7% des images vintage qui sont classés chic. De plus, nous ne pouvons pas ignorer le fait que presque 30% des images chic se confondent avec des images de style classique. Ces erreurs étaient

prévisibles et des explications potentielles de ces derniers sont détaillées dans la partie suivante où nous avons restructuré notre base de données.

5.3 Transfer learning : MobileNetV2

Comme on l'a dit dans la partie implémentation, cette partie présente notre résolution la plus élaborée du problème, c'est dans cette partie que l'on a poussé le modèle pour avoir les meilleurs résultats possibles.

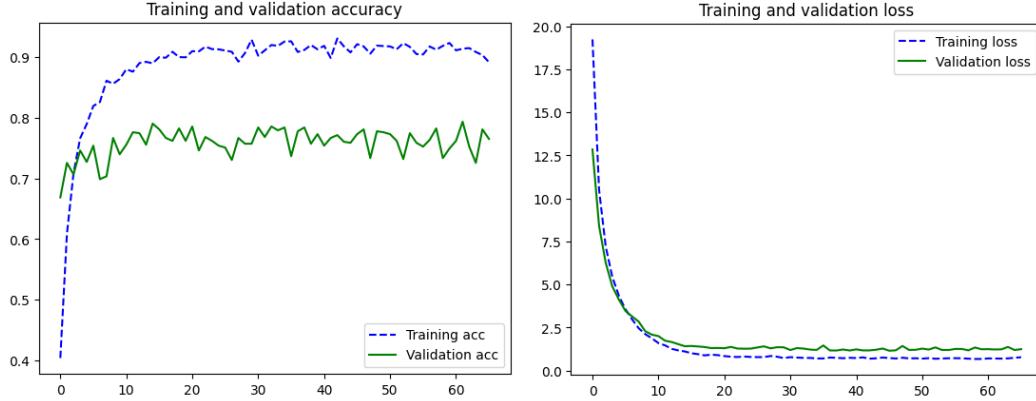


FIGURE 16 – Précision et perte avant fine-tuning.

Le premier entraînement pré fine-tuning s'arrête un peu avant 70 époques. La courbe de perte stagne rapidement car on atteint la précision maximale possible sans réentraîné la base de convolution et l'early stopping arrête donc l'entraînement. On obtient une précision de 90% sur les données d'entraînement et de 76 % sur les données de validation.

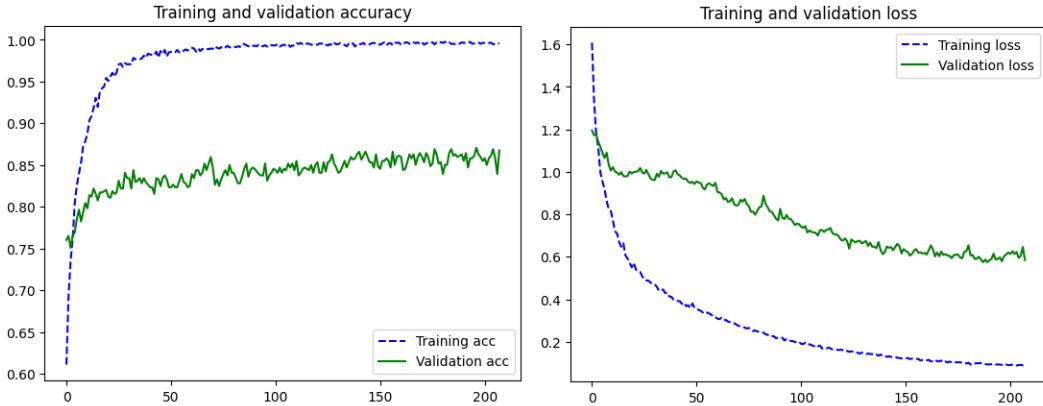


FIGURE 17 – Précision et perte après fine-tuning

Avec les mêmes paramètres d'early stopping, le fine-tuning lui dépasse les 200 époques. On atteint une précision de quasiment 100% sur les données d'entraînement et une précision de 86% sur les données de validation. Ces résultats ont été obtenus sans data augmentation car jusqu'à présent nous avions un problème pour les mettre en place, il s'avère que si on rajoute de la data augmentation, les résultats à ce niveau-là ne change quasiment pas, au contraire on a même obtenu un entraînement donnant de moins bons résultats avec la data augmentation sans pour autant faire de violentes transformations.

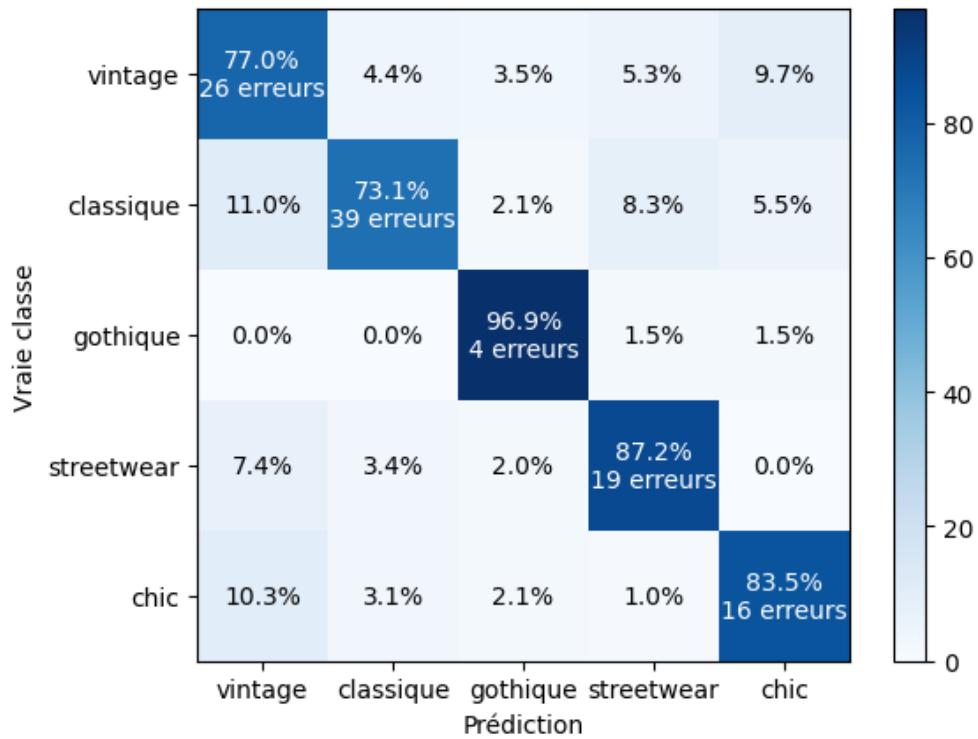


FIGURE 18 – Matrice de confusion

La matrice de confusion confirme ce que nous avions prédit dans la partie des pronostics. Sans surprise c'est la classe gothique qui obtient le moins d'erreurs tellement elle se distingue des autres classes, notamment par la présence dominante de la couleur noire et de motif qu'on ne retrouve nulle part ailleurs. De même il était attendu que ce soit les classes vintage, classique et streetwear qui obtiennent le plus d'erreurs. Si on regarde la colonne vintage, on remarque que c'est avec cette classe qu'il y a la plus de confusion, cela peut s'expliquer de deux manières : soit la bbd de la classe vintage a été moins bien faites que les autres et contient des anomalies qui n'auraient pas été détectées lors de la vérification de celle-ci, soit c'est simplement dû à l'essence même du style vintage, comme on l'a dit dans l'introduction, dans le style vintage on retrouve facilement des styles qui pourraient être qualifiés de chic vintage ou de streetwear vintage ou encore classique vintage. Un vieux smocking peut à la fois être considéré comme vintage et comme chic par exemple. Finalement ces deux explications sont liées, si des anomalies sont présentes dans la bdd c'est justement parce que même nous en tant qu'humains on n'a pas su classer certaines images tellement elle pouvait appartenir à plusieurs, il est d'ailleurs très probable que si l'on cherche bien dans les bdd vintage, classique et streetwear on tombe sur des images présentes dans plusieurs bdd. Une autre remarque intéressante que l'on peut faire grâce à la matrice de confusion est la proportion très faible de confusion entre les classes streetwear et classique : seulement 1%. Quand on y pense, cela est très logique car ces deux styles sont extrêmement différents et ne se confondent jamais, on ne trouvera jamais de baskets dans une tenue chic ou alors de costard dans un tenu streetwear.

On va maintenant étudier des cas précis d'erreurs :

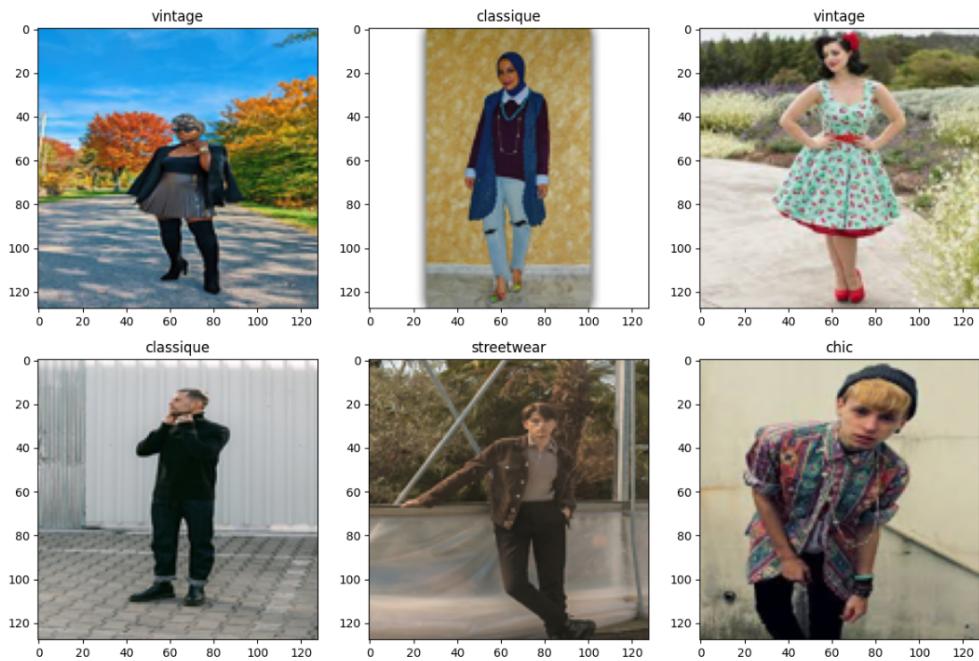


FIGURE 19 – 6 exemples de mauvaises prédictions

1^{re} remarque sur la troisième photo (en haut à droite), la classe prédite est vintage ce qui est censé être une erreur vis-à-vis de notre bdd, or on voit bien que la robe fait très vintage, on a donc ici un exemple d'erreur dans la bdd. Le modèle avait raison dans sa prédiction mais malheureusement cette image se trouve dans la classe chic due à la présence de la robe (google image l'a sûrement associée à une recherche de style vestimentaire chic).

2^{ème} remarque sur la 5^{ème} photo (au milieu en bas) qui a été prédite en streetwear alors qu'elle fait partie de la base classique. Pour un humain il y avait peu de chances qu'on l'on classe cette tenue dans le style streetwear. Pour l'algorithme il est possible qu'il ait été trompé par la pose de la personne ou encore le contexte de l'image.

3^{ème} remarque la dernière photo (en bas à droite) a été classée dans chic alors qu'elle appartient à la classe vintage. Ici il est assez difficile de savoir pourquoi il y a eu cette erreur, une piste possible viendrait de la chemise qui est caractéristique des tenus chics, en revanche les motifs et les couleurs vives auraient dû aiguiller l'algorithme pour faire une meilleure prédiction.



FIGURE 20 – 6 exemples de mauvaises prédictions

Ici les deux 1ères images ont été classés dans classique alors qu'elles appartiennent à chic et streetwear respectivement. Cette prédiction est loin d'être mauvaise et ce serait tout à fait discutable que les images aient été mal rangé dans les bdd. L'élément important dans la première image est les escarpins qui sont plus caractéristiques des tenus chics, pour la deuxième image il est difficile d'en vouloir à notre algorithme car même nous, nous ne sommes pas sur du style vestimentaire.

La troisième image est intéressante, certes on voit tout de suite qu'il peut y avoir confusion entre le style vintage et chic mais ici il est important de noter que l'image est en noir et blanc. Il s'avère que c'est la bdd vintage qui possède le plus d'images en noir et blanc simplement parce que ce sont des images d'époque ou alors parce que cela fait penser au style vintage. En tant qu'humain on a tendance à penser que les images en niveaux de gris sont forcément plus vieilles, il y a donc de forte chance que l'algorithme prenne également cette habitude si la proportion d'image en niveaux de gris n'est pas répartie équitablement dans les différentes bdd.

Pour la quatrième image qui a été prédite en streetwear, l'erreur vient sûrement du fait que la coupe du pull est oversized ce qui est très fréquent voire même omniprésent dans les tenus streetwear. Ici ce que l'algorithme aurait dû prendre en compte serait plutôt la jupe et les escarpins. Sachant que la jupe peut également faire penser à un pantalon oversized.

6 Conclusion

La première chose que l'on tire du travail sur ce projet est la compréhension de l'importance primordiale de la bonne construction d'une base de données : quantité, qualité, structure, répartition. L'amélioration de notre base de données : cropping, suppression des doublons, vérification des anomalies nous a permis de gagner environ 15 % pour la précision. Ainsi, il est important de ne pas se précipiter sur la partie implémentation du réseau de neurones mais plutôt sur un travail approfondi de la base de données. Il faut également choisir judicieusement les outils que l'on souhaite mettre en place par rapport aux nos moyens. Nous sommes limités en temps et en ressources (ram max et temps d'utilisation limite de google colab), il faut donc faire des compromis sur les choix de nos réseaux de neurones : profondeur, largeur, rapport densité précision. D'où le choix du modèle MobileNetV2 qui ne présente pas forcément les meilleures précisions mais qui excellent d'un point de vue efficacité (rapidité et précision).

Par ailleurs, nous rencontré des difficultés avec le rognage des images. Nous avons tenté des approches algorithmiques sans résultat satisfaisant. Nous aurions pu nous arrêter à une solution médiocre, mais nous avons jugé important de passer beaucoup de temps à la recherche d'un résultat qui nous convenait. Par la suite, il s'avère que la meilleure solution était le réseau de neurones YOLO. Ce temps perdu à tester des approches algorithmiques, tout en continuant les recherches pour atteindre nos espérances, était une étape nécessaire pour obtenir nos résultats finaux qui dépassent même nos pronostics.

Pour ce qui est des améliorations, les ressources de calcul à disposition ne nous permettaient pas d'entraîner notre modèle sur des images de taille plus grande que 128 pixels. Or il s'agit d'un des facteurs les plus impactants sur la précision de la classification. Entrainer le modèle sur des machines plus performantes augmenterait nos possibilités. Enfin, il aurait été intéressant d'approfondir la *data augmentation*, notamment sur les transformations visant à changer les couleurs des images pour forcer le réseau à saisir les différents éléments qui composent un style vestimentaire.