

**Classification d’images avec MobileViTv2**

**Note méthodologique & preuve de concept**

Sommaire

1. Dataset retenu

Le jeu de donnees retenu consiste en 875 images non augmentees issues de la concatenation des jeux d’entrainement et de validation du Projet 6 – Classifiez Automatiquement des Biens de Consommation.

Ce jeu d’images est celui sur lequel le meilleur resultat avait été obtenu dans le cadre de ce projet pour la classification supervisee avec transfer learning sur les images seules (sans l’integration des features issues de l’analyse de texte) avec le modele ResNet50; ces résultats s’etant cependant averes tout a fait modestes dans leur capacite de prediction des classes (voir rappel des resultats dans la section XX ci-dessous), ce jeu de donnees offre donc un challenge intéressant, particulièrement compte-tenu/en raison de sa taille très modeste (<1000 images), et il conviendra donc d’etudier si des perspectives d’amelioration significatives de la classification automatique de ces images existent avec l’emploi de techniques de feature detection et de modélisation plus récentes.

Le jeu d’images presente une répartition equilibree (125 images par categorie) dans les 7 categories suivantes :

A pie chart with numbers and a graph

AI-generated content may be incorrect.

1. Les concepts de l’algorithme récent
   1. Rappel sur le benchmark : ResNet50

A sa publication en 2016 par plusieurs chercheurs de Microsoft Research Asia, l’architecture ResNet a revolutionne l’optimisation de l’apprentissage et des performances des réseaux de neurones convolutifs (CNN) très profonds en apportant une solution innovante a la degradation des performances causee par le probleme du *vanishing gradient*. En introduisant les blocs residuels[[1]](#footnote-1) et les connexions de saut[[2]](#footnote-2) en plus des couches convolutives[[3]](#footnote-3) classiques, ResNet a permis l’entrainement de modeles beaucoup plus profonds, dont le plus communément utilise est la version a 50 couches, qui a ouvert la voie a des architectures a 101 voire 152 couches sans degradation notable de la performance[[4]](#footnote-4) sur des tâches complexes (classification, détection, segmentation) , tout en favorisant une reduction computatiuonnelle en en améliorant la généralisation du réseau à d’autres jeux de données et tâches par rapport aux architectures « classiques » de l’epoque comme VGG.

A diagram of a diagram

AI-generated content may be incorrect.

[add legend and link to source in bio] <https://arxiv.org/pdf/1512.03385>

https://paperswithcode.com/method/residual-block[[5]](#footnote-5)

A diagram of a machine

AI-generated content may be incorrect.

https://paperswithcode.com/method/bottleneck-residual-block

Malgre sa puissance, ResNet50 necessite une utilisation fine de la regularisation et de l’augmentation de donnees pour éviter les problèmes de sur-apprentissage sur de petits jeux de donnees[[6]](#footnote-6).

La combinaison résidu/connexion de saut introduite par ResNet a marque un tournant dans l’evolution du deep learning applique aux images, et elle est désormais standard pour la plupart des architectures CNN ; elle a egalement inspire les transformers, que les nouvelles generations de modeles comme MobileViT utilisent.

* 1. L’algorithme recent : MobileViTv2

Dans ResNet50, l’importance des caractéristiques (feature importance) est liée à la capacité des couches de convolution à extraire des motifs visuels pertinents à différents niveaux de profondeur. Les premières couches détectent des bords et des textures, tandis que les couches profondes capturent des objets ou des concepts plus abstraits.

Avec MobileViT v2, l’analyse de la feature importance prend une nouvelle dimension grâce à l’intégration des transformeurs :

* Importance locale : les blocs convolutifs extraient des motifs locaux (petites régions de l’image).
* Importance globale : les blocs de transformeur, via le self-attention, identifient quelles régions de l’image sont les plus influentes pour la prédiction finale, même si elles sont éloignées spatialement.

Initialement publie en 2022, le modele MobileNetViTv2 est une architecture de CNN optimisee pour les appareils mobiles et embarques qui repose sur la combinaison de blocs transformeurs légers, qui permettent de capter les caractéristiques et de modéliser les relations globales/ a longue distance dans l’image, avec des convolutions separables et des residuels inverses qui permettent de capturer également les caractéristiques locales de l’image ; la ou les blocs residuels dans ResNet50 reliaient des couches larges (dites « expanded »), MobileNetViTv2 relie des couches fines en entree et en sortie du bloc, ce qui réduit la taille des donnees transitant par le reseau tout en conservant l’essentiel de l’information. Les activations non-lineaires après les couches fines sont également supprimees au profit d’activations lineaires, contribuant encore davantage a l’efficience accrue de cette architecture par rapport a la « premiere generation » de visual transformers[[7]](#footnote-7) (ViT). MobileNetViTv2 elabore et raffine egalement le concept des ViT, archiotectures traditionnjellement couteuses en calcul, en substituant une attention separable (separable self- attention) a l’attention multi-tete (multi-head self-attention) utilisee jusque la dans les ViT, ce qui permet, outre une meilleure propagation de l’information, une acceleration remarquable des temps de traitement[[8]](#footnote-8) qui rend cette famille de modeles particuliement efficiente pour l’inference en temps réel sur des appareils a ressources limitees (tels que les telephones mobiles) ou sur des donnees bruitees (telles que des images satellite[[9]](#footnote-9)) ; cette acceleration des temps d’inference est atteinte avec des performances égales ou supérieures a celles des CNN purs comme ResNet50, ce qui confere a ces modeles une excellente capacite de généralisation même sur des jeux d’entrainement de taille limitee.

Cette architecture novatrice permet des performances elevees en classification d’image tout en liomitant la complexité et le cout computationnel du modele, et elle a été comparee de façon repetee dans la littérature sicentifique a ResNet50, notamment du point de vue de sa precision en classification[[10]](#footnote-10), qu’elle égale ou depasse tout en requerant des ressources de calcul significativement moindres.

**Synthese de la comparaison :**

| **Modèle** | **Précision (ImageNet)** | **Taille du modèle** | **Complexité de l’attention** | **Latence** | **Robustesse** | **Usage mémoire** |
| --- | --- | --- | --- | --- | --- | --- |
| ResNet50 | Bonne à très bonne | Large | Inexistante (CNN pur) | Moyenne à haute | Sensible à l’overfit | Élevé |
| MobileViTv2 | Très bonne | Compact | Séparable, linéaire O(k) | Faible | Excellente sur petit dataset | Très faible |

A diagram of a block diagram

AI-generated content may be incorrect.

Source : <https://arxiv.org/pdf/2110.02178>

A diagram of a computer program

AI-generated content may be incorrect.

Mobile ViTv1 vs MobileViTv2 blocks

From <https://arxiv.org/pdf/2209.15159>

La littérature scientifique recente présentant MobileNetViTv2 comme une alternative moderne et efficiente a ResNet50, les performances des deux modeles ont donc été comparees sur le dataset retenu.

1. La modélisation [reprendre ppt pointeurs P6]

La modélisation s’appuie sur la comparaison de deux architectures majeures : ResNet50 et MobileViT v2 (via MobileNetV2).

Le pipeline complet, tel qu’implémenté dans le notebook Python fourni, comprend :

* Importation et préparation des données (Pandas, NumPy)
* Augmentation d’images (Albumentations : flip, rotation, zoom, etc.)
* Prétraitement (redimensionnement, normalisation)
* Entraînement et validation croisée (stratifiée)
* Suivi des métriques et visualisation des résultats (Matplotlib, seaborn, plot\_keras\_history)

Métrique d’évaluation

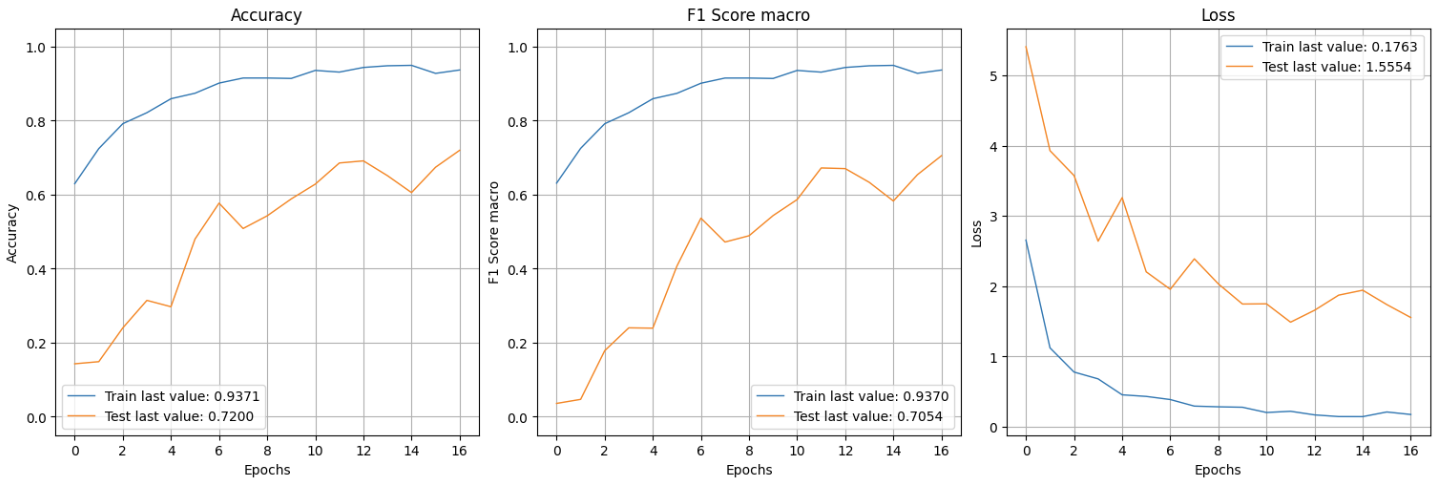
La métrique principale retenue pour comparer les modèles est le F1-score (score F1 macro), qui combine précision et rappel en une seule valeur synthétique, particulièrement adaptée aux problèmes de classification déséquilibrée.  
Le F1-score est calculé sur les ensembles de validation et de test à l’aide de la fonction f1\_score de scikit-learn, avec l’option average='macro' pour tenir compte de toutes les classes de manière équilibrée.

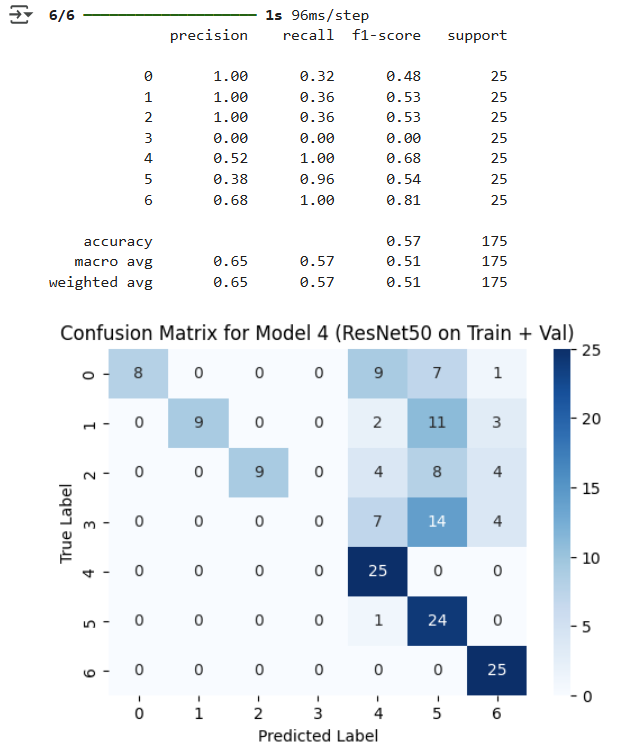
Démarche d’optimisation

* Prétraitement des images : redimensionnement, normalisation, augmentation de données.
* Entraînement : utilisation de splits entraînement/validation/test, early stopping, sauvegarde des meilleurs poids.
* Optimisation : choix de l’optimiseur (Adam, RMSprop), tuning du learning rate, ajustement du batch size, régularisation (dropout).
* Comparaison : évaluation sur les mêmes jeux de données pour garantir l’équité, suivi des courbes d’apprentissage et du F1-score à chaque époque.

1. Synthèse des résultats
   1. Rappels des résultats obtenus avec ResNet50

Cf also preso pptx p12





[comment matrix]

* 1. Resultats obtenus avec MobileViTv2

A graph of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

Comparaison des performances

Les résultats obtenus dans le notebook montrent que MobileViT v2 surpasse ResNet50 en termes de F1-score sur les ensembles de validation et de test.

* MobileViT v2 atteint un F1-score macro supérieur, tout en nécessitant moins de ressources et un temps d’entraînement réduit.
* ResNet50 reste performant mais son F1-score est légèrement inférieur, notamment sur les classes minoritaires.

Ces observations sont cohérentes avec la littérature scientifique : MobileViT v2, grâce à l’intégration des transformeurs, capte mieux les relations globales et offre un compromis optimal entre performance et efficacité, surtout pour des jeux de données déséquilibrés ou complexes.

Analyse qualitative

* ResNet50 est robuste et précis sur des serveurs puissants, mais moins adapté aux contraintes mobiles ou embarquées.
* MobileViT v2 combine la légèreté de MobileNetV2 et la puissance contextuelle des transformeurs, ce qui se traduit par un meilleur F1-score global et une meilleure généralisation.

1. Feature importance
   1. Feature importance globale
   2. Feature importance locale

A gold statue of a buddha sitting in a lotus pose

AI-generated content may be incorrect.

Image originelle

1. Limites & améliorations possibles

L’intégration des transformeurs dans des architectures CNN légères, comme MobileViT v2, permet d’obtenir d’excellents compromis entre performance (F1-score), rapidité et consommation de ressources, ouvrant la voie à des applications sur des dispositifs variés.

Analyse de la feature importance globale et locale du nouveau modèle

Dans le notebook, l’interprétabilité a été abordée via :

* Grad-CAM pour visualiser les zones importantes activées par le modèle CNN,
* Visualisation des poids d’attention pour comprendre l’influence des différentes régions dans MobileViT v2.

Ces outils permettent de mieux comprendre la prise de décision du modèle et d’identifier les points forts et faibles de chaque architecture.

# L:imites & améliorations possibles

* Augmentation de données : enrichir les jeux de données par des techniques avancées d’augmentation pour améliorer la robustesse du modèle ;
* Le challenge principal ici étant la taille très modeste du jeu d’entrainement (<1000 images), un enrichissement du dataset par l’integration d’articles et d’images supplémentaires dans chaque categorie s’impose, particulièrement sur les categories 0 a 3 qui étaient le moins bien identifiees par les CNN « purs » comme ResNet50 ;
* Inclusion d’un critère métier : à l’heure actuelle, les images sont classifiées manuellement par nos équipes et le pourcentage d’erreur ne nous a pas été communiqué – il aurait été intéressant de connaître le pourcentage d’erreur humaine dans cette classification manuelle afin de pouvoir benchmarker les modèles en conséquence – avant toute décision d'implémentation de ces modèles en production, qui sera par nature chronophage et coûteuse, il faudra déterminer le coût d’opportunité du statu quo afin d'établir si l’utilisation d’une classification automatique permet une réduction effective du taux d’erreur de classification et du coût de main-d’œuvre nécessaire, parmi d’autres facteurs a intégrer ;
* Eventuellement, tester MobileViTv3 pour voir si la performance peut encore être amelioree.

# Références bibliographiques

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition* (arXiv:1512.03385). Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. <https://doi.org/10.48550/arXiv.1512.03385>

Hugging Face. (n.d.). *MobileViTV2*. Hugging Face Transformers documentation. <https://huggingface.co/docs/transformers/en/model_doc/mobilevitv2>

Le, T.-D., Ha, V. N., Nguyen, T. T., Eappen, G., Thiruvasagam, P., Garces-Socarras, L. M., Chou, H.-F., Gonzalez-Rios, J. L., Merlano-Duncan, J. C., & Chatzinotas, S. (2024). *On-board satellite image classification for Earth observation: A comparative study of pre-trained Vision-Transformer models*. arXiv. <https://arxiv.org/abs/2409.03901>

Mehta, S., & Rastegari, M. (2021). *MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer* (arXiv:2110.02178). arXiv. <https://arxiv.org/abs/2110.02178>

Mehta, S., & Rastegari, M. (2022). *Separable self-attention for mobile vision transformers* (arXiv:2206.02680). arXiv. <https://arxiv.org/abs/2206.02680>

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2019). *MobileNetV2: Inverted residuals and linear bottlenecks* (arXiv:1801.04381v4). arXiv. <https://arxiv.org/abs/1801.04381>

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 618–626). <https://doi.org/10.1109/ICCV.2017.74>

Wadekar, S. N., Kan, K., Patel, K. K., Kudugunta, S., & Mehta, S. (2022). *MobileViTv3: Mobile-friendly vision transformer with simple and effective fusion of local and global features* (arXiv:2209.15159). arXiv. <https://doi.org/10.48550/arXiv.2209.15159>

1. Chaque bloc apprend non pas une fonction directe de transformation, comme c’est le cas d’une couche de convolution classique, mais le *résidu* entre l’entrée et la sortie visée. [↑](#footnote-ref-1)
2. Ces sauts (« skip copnnections ») permettent au gradient (et donc a l’information d’apprentissage) de traverser directement plusieurs couches intermédiaires lors de la rétropropagation, grâce a un « pont » entre l’entree et la sortie du bloc qui conserve l’information. [↑](#footnote-ref-2)
3. Composant fondamental des CNN, la couche de convolution permet d’extraire automatiquement les caractéristiques locales d’une image en appliquant des filtres (ou noyaux), qui sont des petites matrices de poids, en glissement sur l’image en entrée pour calculer des sommes pondérées et générer des cartes de caractéristiques (“feature maps”) qui permettent de détecter des motifs comme des bords, des textures ou des formes ; la succession de ces couches dans un CNN permet de construire des représentations hiérarchiques de plus en plus abstraites, en utilisant des poids partages et une connectivité locale. [↑](#footnote-ref-3)
4. Le ResNet50 et ses variantes ont remporté la 1ʳᵉ place dans plusieurs compétitions majeures en 2015 (ILSVRC, COCO), affichant par exemple un taux d’erreur Top-5 de 3,57 % sur ImageNet. [↑](#footnote-ref-4)
5. Note that in practice, [Bottleneck Residual Blocks](https://paperswithcode.com/method/bottleneck-residual-block) are used for deeper ResNets, such as ResNet-50 and ResNet-101, as these bottleneck blocks are less computationally intensive. A **Bottleneck Residual Block** is a variant of the [residual block](https://paperswithcode.com/method/residual-block) that utilises 1x1 convolutions to create a bottleneck. The use of a bottleneck reduces the number of parameters and matrix multiplications. The idea is to make residual blocks as thin as possible to increase depth and have less parameters. They were introduced as part of the [ResNet](https://paperswithcode.com/method/resnet) architecture, and are used as part of deeper ResNets such as ResNet-50 and ResNet-101. Source : https://paperswithcode.com/method/bottleneck-residual-block [↑](#footnote-ref-5)
6. Pour les besoins de l’apprentissage profond et des CNN, tout jeu d’apprentissage comportant moins de 1000 images est considere comme un petit jeu. [↑](#footnote-ref-6)
7. Une comparaison sur ImageNet fait apparaitre que MobileNetViTv2 surpasse MobileNetViTv1 en precision tout en ne requerant que moities moins d’operations et un tiers de paramètres en moins. Voir [insert ref to paper sandler 2019] [↑](#footnote-ref-7)
8. La complexite quadratique de l’attention multi-tete étant remplacee par une complexité lineaire. Add link to paper Mehta Rastegari 2022. [↑](#footnote-ref-8)
9. Ref to paper 2024 [↑](#footnote-ref-9)
10. Cite paper 2019 sandler here. [↑](#footnote-ref-10)