**Note méthodologique : preuve de concept**

## Dataset retenu

Le jeu de données sélectionné est identique à celui utilisé dans le Projet 6 de la formation Data Scientist d’OpenClassrooms, intitulé "Classifiez automatiquement des biens de consommation".

Ce jeu de données comprend 1050 articles, chacun étant accompagné d'une image et d'une description. Dans le cadre de ce projet, nous allons uniquement utiliser les images.

Une image contenant texte, capture d’écran, Approvisionnement général

Description générée automatiquement

Chaque article est associé à une des 7 catégories :

* Watches
* Baby Care
* Home Furnishing
* Computers
* Beauty and Personal Care
* Kitchen & Dining
* Home Decor & Festive Needs

A noter que le jeu de données contient exactement le même nombre d’articles par catégories.

Dans ce projet, nous allons effectuer une tâche de classification afin de prédire la catégorie associée à un article en fonction de son image.

## Les concepts de l’algorithme récent

Suite à un état de l’art de la littérature scientifique réalisé sur Google Sholar, Pubmed et Arxiv, la technique de modélisation récente mise en place dans ce projet dans se nomme Vision Transformer.

Cette méthode a été proposée dans un article publié en 2020 sur Arxiv, intitulé "[An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale](https://arxiv.org/abs/2010.11929)" par Dosovitskiy et al.

### Introduction

Les Vision Transformers (ViTs) représentent une avancée majeure dans le domaine de la classification d'images. Contrairement aux modèles traditionnels basés sur les réseaux de neurones convolutifs (CNN), les ViTs utilisent une architecture de transformateurs, initialement développée pour le traitement du langage naturel. Cette approche permet au modèle de traiter une image en la découpant en petits patches (comme des "mots") et en appliquant des mécanismes d'attention pour extraire les caractéristiques pertinentes. L'article de Dosovitskiy et al. propose que les Transformers, avec leur capacité à capturer des dépendances globales, peuvent surpasser les réseaux de neurones convolutifs (CNN) dans certaines tâches de Computer Vision.

### Fonctionnement de l’algorithme

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

#### Découpage en patches

Au lieu de traiter une image comme une grille de pixels, les ViTs la transforment en une séquence de "patches" (morceaux). L’image d'entrée est divisée en 𝑁 patchs de taille fixe 𝑃 × 𝑃. Par exemple, une image de 224×224 pixels découpée en patchs de 16×16 pixels donne 196 patchs (224/16=14 et 14×14=196).

Chaque patch est ensuite aplati en un vecteur de dimension 𝐷 via une projection linéaire.

*Ajout des Embeddings de Position*

Pour conserver l'information de position, des embeddings de position sont ajoutés aux vecteurs de patchs. Ces embeddings permettent au modèle de conserver la notion de l'ordre spatial des patchs. Ces encodages positionnels sont ajoutés de manière fixe ou par des fonctions trigonométriques, contrairement aux Transformers en traitement du langage naturel, où les encodages positionnels sont généralement appris.

*Structure du Transformer*

La partie centrale des ViTs est composé de plusieurs blocs identiques, chacun contenant deux sous-couches principales : un mécanisme de self-attention multi-têtes et un réseau feed-forward.

Chaque bloc de l'Encoder commence par un mécanisme d'attention multi-têtes pour capturer les dépendances globales entre les patches :

* Calcul des Scores d'Attention : Pour chaque tête d'attention, les vecteurs d'entrée sont projetés en trois ensembles distincts de vecteurs appelés clés (K), requêtes (Q), et valeurs (V).
* Application de l'Attention : Les scores d'attention sont calculés en utilisant le produit scalaire entre les requêtes et les clés, suivi d'une normalisation softmax. Ces scores sont ensuite utilisés pour pondérer les valeurs.
* Combinaison des Têtes : Les résultats des différentes têtes d'attention sont concaténés et projetés dans l'espace d'origine par une couche linéaire.

Après le mécanisme d'attention multi-têtes, chaque vecteur passe par un réseau feed-forward qui se compose généralement de deux couches linéaires séparées par une activation non linéaire (souvent une fonction ReLU).

L'Encoder est constitué de plusieurs blocs identiques (typiquement 12, 24 ou 32 blocs) empilés. Chaque bloc suit le schéma décrit ci-dessus, permettant ainsi de modéliser les interactions complexes entre les différents patchs d'image à différents niveaux de profondeur.

#### Sortie du Transformer

Après le passage à travers tous les blocs de l'Encoder, le vecteur de sortie est ensuite passé à travers une couche linéaire (parfois une activation softmax) pour effectuer la classification finale.

## La modélisation

### Prétraitement des Données

Le projet a commencé par le prétraitement des données. Les étapes clés incluent :

* Chargement des données : Les données ont été chargées à partir d'un fichier CSV contenant des informations sur les produits, y compris les catégories et les images associées.
* Extraction des catégories : Une fonction a été créée pour extraire la catégorie générale des articles à partir d'une chaîne de texte. Cette catégorie a été utilisée comme cible pour la classification.
* Encodage des labels : Les catégories extraites ont été encodées en utilisant LabelEncoder pour transformer les étiquettes de texte en valeurs numériques.
* Organisation des images : Les images ont été redimensionnées à 224x224 pixels et organisées dans des dossiers selon leurs catégories respectives pour faciliter le traitement ultérieur.

### Création du Jeu de Données

Création du jeu de données à partir des images organisées :

* Utilisation de datasets : La bibliothèque datasets a été utilisée pour créer un jeu de données structuré à partir des dossiers d'images. Chaque image a été associée à son étiquette correspondante.
* Traitement des images : L’extracteur de caractéristiques pré-entraîné (ViTFeatureExtractor) a été utilisé pour prétraiter les images en les redimensionnant et en les convertissant au format requis par le modèle ViT.
* Division des données : Le jeu de données a été divisé en ensembles d'entraînement (60%), de validation (20%) et de test (20%) en utilisant train\_test\_split de datasets.

### Modélisation

Le modèle mis en place dans ce projet est le modèle TFViTModel pré-entraîné sur ImageNet-21k (14 million d’images, 21 843 classes) de la bibliothèque transformers. Ce modèle sert d'extracteur de caractéristiques pour les images.

Le modèle a été encapsulé dans une couche Keras appelée ViTLayer afin de l’intégrer dans un pipeline de modélisation Keras.

Les entrées du modèle ont ensuite été définies. Dans notre cas sont des images de forme (3, 224, 224) représentant les canaux de couleur et les dimensions de l'image.

Enfin, une couche de couche de classification dense avec une activation softmax a été ajoutée. Cette couche prend les caractéristiques extraites par le modèle ViT et les utilise pour prédire les catégories des images.

Le modèle a été compilé avec l'optimiseur adam, la fonction de perte sparse\_categorical\_crossentropy, et la métrique accuracy.

### Métrique d'Évaluation

La métrique d'évaluation principale utilisée est l’accuracy qui permet de mesurer la proportion de prédictions correctes parmi toutes les prédictions effectuées par le modèle. Cette métrique est appropriée pour ce projet pour plusieurs raisons :

* Nature des Catégories : Les catégories des produits dans le jeu de données sont mutuellement exclusives. Chaque image appartient à une seule catégorie, ce qui signifie qu'une classification correcte est binaire : soit l'image est classée correctement, soit elle ne l'est pas.
* Classes équilibrées : Le jeu de données contient exactement le même nombre d'observations pour chaque classe. Cet équilibrage des classes renforce la pertinence de l’accuracy comme métrique d'évaluation car elle offre une évaluation juste et équitable de la performance du modèle sur toutes les classes.

Une image contenant texte, diagramme, capture d’écran, cercle

Description générée automatiquement

* Simplicité et Interprétabilité : L’accuracy est une métrique intuitive et facile à interpréter. Elle permet à différentes parties prenantes, y compris celles qui ne sont pas spécialistes en machine learning, de comprendre rapidement les performances du modèle.

### Entraînement et Évaluation

Le modèle a été entraîné sur les données d'entraînement et évalué sur les ensembles de validation et de test :

* Entraînement : Le modèle a été entraîné pendant 5 époques en utilisant model.fit, avec les données d'entraînement et de validation.
* Évaluation : Les performances du modèle ont été évaluées sur l'ensemble de test pour obtenir la précision finale.

## Une synthèse des résultats

*Présentez une synthèse des résultats comparés entre la technique récente et les techniques utilisées précédemment et une conclusion, en 2 pages maximum.*

## L’analyse de la feature importance du nouveau modèle

Pour les modèles de Computer Vision, l’importance des caractéristiques est plus complexe à calculer que pour des modèles tabulaires, mais il est possible d'obtenir l'interprétabilité du modèle grâce à une carte de chaleur de l'attention.

Le mécanisme d'attention permet au modèle de se concentrer sur certaines parties des données d’entrée tout en ignorant les autres. Cela aide à comprendre quelles parties de l'image influencent le plus la décision du modèle.

Une image contenant texte, ordinateur

Description générée automatiquement

Ici, les zones où la couleur est plus claire indiquent que le modèle prête plus d'attention à ces parties de l'image. Cela signifie que les plus influentes pour la décision du modèle se trouvent vers le milieu de cette image.

## Les limites et les améliorations possibles

### Limites de l'Approche de Modélisation

#### Complexité et Temps de Calcul :

L'utilisation d'un modèle Vision Transformer demande beaucoup de puissance de calcul et de mémoire, ce qui implique un besoin élevé en ressources matérielles pour le traitement des images, l'entraînement du modèle, et la phase de prédiction. Par conséquent, cela peut restreindre l'utilisation pratique de ce modèle, en particulier sur des systèmes ayant des capacités matérielles limitées.

#### Taille du Dataset :

Avec un échantillon de seulement 1050 images, le dataset est relativement petit pour entraîner efficacement un modèle aussi complexe que ViT. Un petit dataset peut entraîner un surapprentissage (overfitting) où le modèle performe bien sur les données d'entraînement mais échoue à généraliser sur de nouvelles données.

C’est pourquoi, plutôt que de former un modèle ViT complet à partir de zéro, la démarche mise en place a consisté à utiliser un modèle pré-entraîné et effectuer un fine-tuning sur le dataset spécifique sélectionné. Cela permet de bénéficier des connaissances déjà acquises par le modèle sur un large corpus de données.

#### Interprétabilité :

Les modèles de deep learning, en particulier les transformers, sont souvent considérés comme des "boîtes noires". Il est difficile d'expliquer pourquoi le modèle prend certaines décisions, ce qui peut poser problème dans des applications nécessitant une interprétabilité élevée.

### Améliorations Envisageables

#### Augmentation des Données :

Utiliser des techniques d'augmentation de données (rotation, zoom, recadrage, etc.) pour augmenter la diversité des images dans le dataset. Cela peut aider à réduire le surapprentissage et améliorer la généralisation du modèle.