Note méthodologique : preuve de concept

I- Dataset retenu

Nous travaillons ici sur le jeu des données du projet 6 ***« Classifiez automatiquement des biens de consommation ».***

1- Rappel du contexte et de la problématique

L'entreprise Place de marché souhaite lancer une marketplace qui permettra à des vendeurs de proposer des articles à des acheteurs en postant une photo et une description.
Afin de fluidifier l'expérience des utilisateurs, l'attribution de la catégorie d'un article qui est actuellement manuelle doit être automatisée.
Pour ce faire, nous devons dans un premier temps étudier la faisabilité d'une classification automatique des articles en fonction des descriptifs et des images, ensuite réaliser une classification automatique en se basant sur les images des différents produits. Ce dataset contient un ensemble de 1050 images et descriptifs classées en 7 catégories de 150 images chacune.

2- Point sur les travaux précédemment réalisés à mettre à jour par une méthode nouvelle

Dans le projet 6, nous avons étudié la faisabilité d'une classification des produits via les images qui étaient fournies. Nous avons donc utilisé la base convolutionnelle pré entraînée VGG16 comme extracteur de features pour l'ensemble des images, ensuite nous avons appliqué deux réductions successives de dimension (une via PCA en conservant 99% de la variance et une deuxième via t-SNE), nous avons au final calculé le score ARI et projeté les clusters en 2D pour une analyse visuelle et conclusion sur la faisabilité.
En ce qui concerne nos analyses actuelles sur la veille technique, nous utiliserons le jeu des données des images, nous nous servirons d'une méthode récente pour l'extraction des features et l'analyse de la faisabilité de la classification. Nous comparerons ensuite les deux méthodes et tirerons des conclusions.

II- Les concepts de l'algorithme récent

Le concept nouveau que nous souhaitons présenter est celui de Vision Transformer ViT. Modèle Vision Transformer (ViT) pré-entraîné sur ImageNet-21k (14 millions d'images, 21 843 classes) à la résolution 224x224.

1- Description du modèle

Le Vision Transformer (ViT) est un modèle de codeur transformateur (de type BERT) pré-entraîné sur une grande collection d'images de manière supervisée, à savoir ImageNet-21k, à une résolution de 224x224 pixels.
Les images sont présentées au modèle sous la forme d'une séquence de parcelles de taille fixe (résolution 16x16), qui sont intégrées de manière linéaire. On ajoute également un jeton [CLS] au début d'une séquence pour l'utiliser dans les tâches de classification. On ajoute également des encastrements de position absolue avant d'envoyer la séquence aux couches de l'encodeur Transformer.
Il convient de noter que ce modèle ne fournit pas de têtes affinées, car celles-ci ont été réduites à zéro par les chercheurs de Google. Cependant, le modèle inclut le pooler pré-entraîné, qui peut être utilisé pour des tâches en aval (telles que la classification d'images).
En pré-entraînant le modèle, il apprend une représentation interne des images qui peut ensuite être utilisée pour extraire des caractéristiques utiles pour des tâches en aval

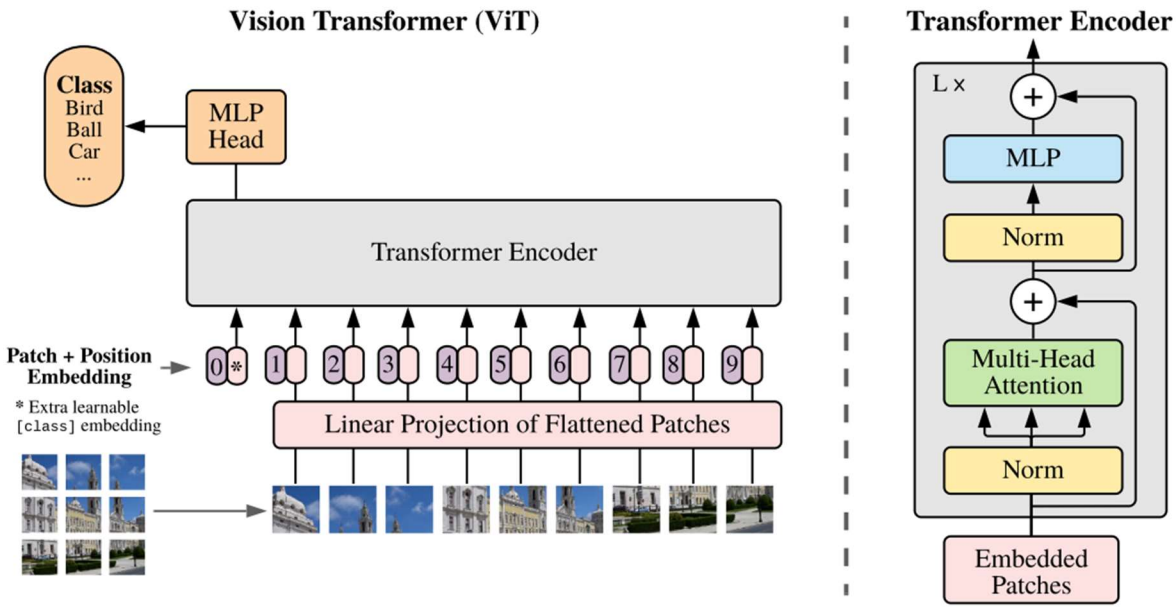


Schéma synthétique de traitement

III- Méthodologie de Modélisation de ViT (google/vit-base-patch16-224-in21k)

1- Prétraitement de l'Image

- *Taille de l'image : Les images en entrée sont redimensionnées à une taille de 224x224 pixels.*
- *Découpage en patches : L'image est divisée en patches de taille fixe (par exemple, 16x16 pixels chacun). Pour une image de 224x224, cela génère 14x14 = 196 patches.*
- *Linear Embedding : Chaque patch est aplatit en un vecteur (256 dimensions) et ensuite projeté en une représentation de 768 dimensions via une couche linéaire. Ce processus est similaire à ce qui se fait pour transformer des mots en vecteurs denses en NLP.*

2- Ajout d'un Token de Classe et des Embeddings de Position

- *Un Token de Classe est ajouté aux embeddings des patches. Ce token ([CLS]) permet de capturer l'information globale de l'image pour la classification.*
- *Embeddings de position : ViT n'ayant pas de structure convolutionnelle, il n'a pas de biais spatial intégré. On ajoute donc des embeddings de position pour indiquer la localisation de chaque patch dans l'image, afin de préserver la structure spatiale de l'image.*

3- Passage par le Modèle Transformer

Les embeddings des patches sont ensuite passés dans une série de blocs Transformer (12 pour le modèle google/vit-base), similaires à ceux utilisés en NLP. Chaque bloc est composé de :

- *Multi-Head Self-Attention (MHSA) : Capture les relations entre les patches indépendamment de leur distance dans l'image.*
- *Feed Forward Networks (FFN) : Applique une transformation non linéaire pour renforcer les relations capturées.*
- *Add & Norm : Mécanisme de normalisation et de connexion résiduelle pour stabiliser l'apprentissage.*

4- Extraction du Token de Classe ([CLS])

Après plusieurs blocs, seul le vecteur du Token de Classe ([CLS]) est récupéré pour la tâche de classification. Ce vecteur encapsule l'information globale de l'image après que les relations entre les patches aient été capturées par les différents blocs.

5- Classification Finale

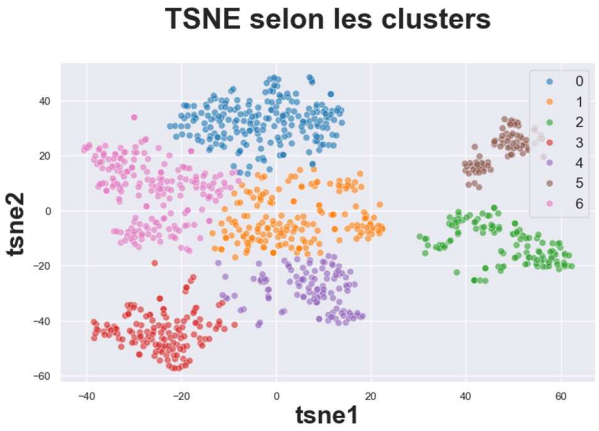
- *Le vecteur [CLS] passe à travers une couche linéaire de classification.*
- *Dans le cas du modèle google/vit-base-patch16-224-in21k :*
    - *Le modèle est pré-entraîné sur ImageNet-21k avec 21 000 classes.*
    - *La couche de classification est ajustée pour la tâche spécifique en fine-tuning.*

6- Utilisation des Features Extraites

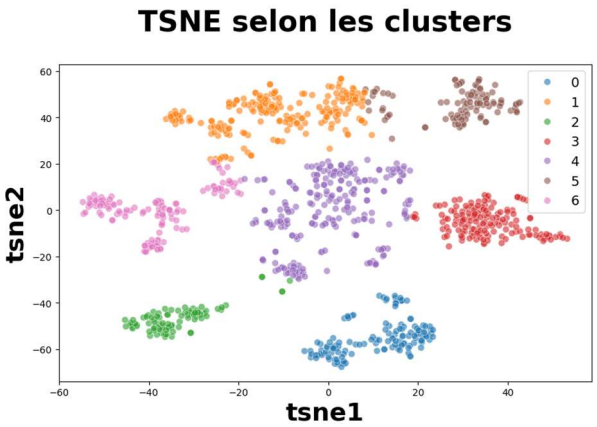
- *Pour la classification : Le vecteur [CLS] est utilisé pour prédire la classe de l'image.*
- *Pour des tâches de type clustering ou transfert : On peut utiliser les features de la dernière couche cachée pour capturer des informations utiles sans fine-tuning.*

IV- Une synthèse des résultats

	VGG16	ViT
Temps de traitement	*8 minutes*	*13 minutes*
Longueur embedding	*(1050, 4096)*	*(1050, 768)*
Après réduction de dimension via PCA	*(1050, 793)*	*(1050, 531)*
Après réduction de dimension via t-SNE	*(1050, 2)*	*(1050, 2)*
Score ARI après clustering	***0.46***	***0.57***



Clusters après extraction VGG16



Clusters après extraction ViT

<u>Conclusion :</u>

Avec l'extracteur de ViT, les clusters sont mieux déterminés et mieux séparés comme nous pouvons l'observer sur les visualisations graphiques. Le score ARI de 0.57 nettement supérieur vient bien confirmer cette analyse. Nous pouvons ainsi dire que pour ce jeu de données le modèle ViT est plus adapté que celui de VGG16.

V- Les limites et les améliorations possibles

	VGG16	ViT
Architecture	*Fonctionne bien sur des images où les caractéristiques locales sont essentielles (par exemple, la reconnaissance d'objets simples).*	*excelle sur des images plus complexes ou des tâches où les relations globales au sein de l'image sont importantes (comme la compréhension de scènes plus complexes).*
Extraction des features	*Se concentre davantage sur des aspects locaux (motifs, textures).*	*Apprend les relations globales, ce qui peut être un avantage pour des images complexes ou des tâches nécessitant une vue d'ensemble.*
Fine-tuning	*Le fine-tuning de VGG16 est plus direct*	*ViT offre plus de flexibilité pour s'adapter à des données très différentes.*
Performance en termes de précision et de vitesse	*VGG16 est plus rapide pour des images simples*	*ViT peut surpasser VGG16 sur des tâches plus complexes au prix d'une plus grande consommation de ressources.*
Points forts	*Adapté à des tâches de classification simples avec des objets distincts.* *Plus rapide sur des tâches nécessitant des images de taille moyenne ou petite.* *Requiert moins de données pour obtenir des résultats corrects.*	*Excellente performance pour des tâches où les relations globales entre les objets sont importantes.* *Meilleure capacité d'adaptation et de fine-tuning sur des tâches complexes ou des bases de données très variées.* *Apprend mieux les relations spatiales complexes.*

- **VGG16** est un choix solide pour des tâches où la structure **locale** de l'image (motifs, textures) est importante et où les relations globales sont moins cruciales. Il est **rapide** et bien adapté aux petites bases de données.
- **ViT** est plus performant dans des contextes où les relations entre différentes parties de l'image jouent un rôle essentiel. Il est mieux adapté aux tâches complexes mais peut nécessiter **plus de données** et de puissance de calcul.

En fonction des besoins spécifiques du projet, ViT pourrait apporter des **améliorations significatives** dans la gestion d'images complexes, alors que VGG16 reste un bon choix pour des tâches de classification plus simples et rapides.

Sources et articles consultées pour cette veille technique

Article :

- https://arxiv.org/abs/2010.11929
- https://huggingface.co/google/vit-base-patch16-224-in21k
- https://github.com/google-research/vision_transformer.git