

NOTE MÉTHODOLOGIQUE : PREUVE DE CONCEPT

Classification d'Images E-commerce par Réseaux d'Agrégation Contextuelle Panoramique

Projet : Mission 8 - Veille Technologique ML

Date : Janvier 2026

Auteur : Adrien NORMAND

Table des Matières

1. [Dataset Retenu](#)
 2. [Concepts de l'Algorithme Récent : PanCAN](#)
 3. [Modélisation](#)
 4. [Synthèse des Résultats](#)
 5. [Analyse de la Feature Importance](#)
 6. [Limites et Améliorations Possibles](#)
 7. [Références Bibliographiques](#)
-

1. Dataset Retenu

1.1 Présentation du Dataset

Le dataset utilisé est un échantillon de **Flipkart E-commerce** contenant **1 050 images de produits** réparties en **7 catégories** représentatives du commerce en ligne :

Catégorie	Description	Exemples
Baby Care	Produits pour bébés	Serviettes, soins
Beauty & Personal Care	Cosmétiques	Maquillage, soins
Computers	Informatique	Accessoires PC
Home Decor	Décoration	Articles festifs
Home Furnishing	Ameublement	Rideaux, literie
Kitchen & Dining	Cuisine	Ustensiles
Watches	Montres	Montres-bracelets

1.2 Répartition des Données

La stratification garantit une distribution équilibrée des classes :

Split	Échantillons	Ratio
Entraînement	629	60%
Validation	158	15%
Test	263	25%

1.3 Prétraitement

- **Redimensionnement** : 224×224 pixels (standard ImageNet)
- **Normalisation ImageNet** : $\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$
- **Augmentation** (entraînement uniquement) :
- RandomResizedCrop (224)
- RandomHorizontalFlip ($p=0.5$)
- ColorJitter (brightness=0.2, contrast=0.2)
- RandomErasing ($p=0.1$)

Figure 1 : Distribution des classes par split

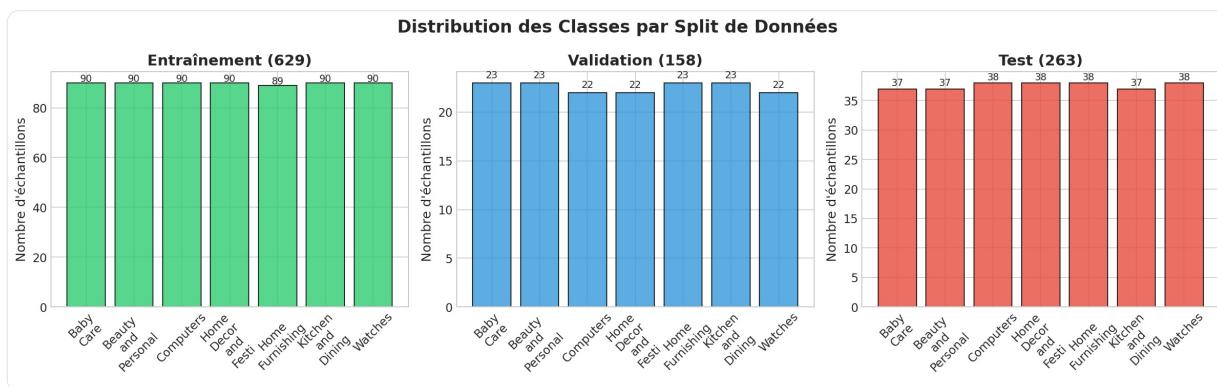
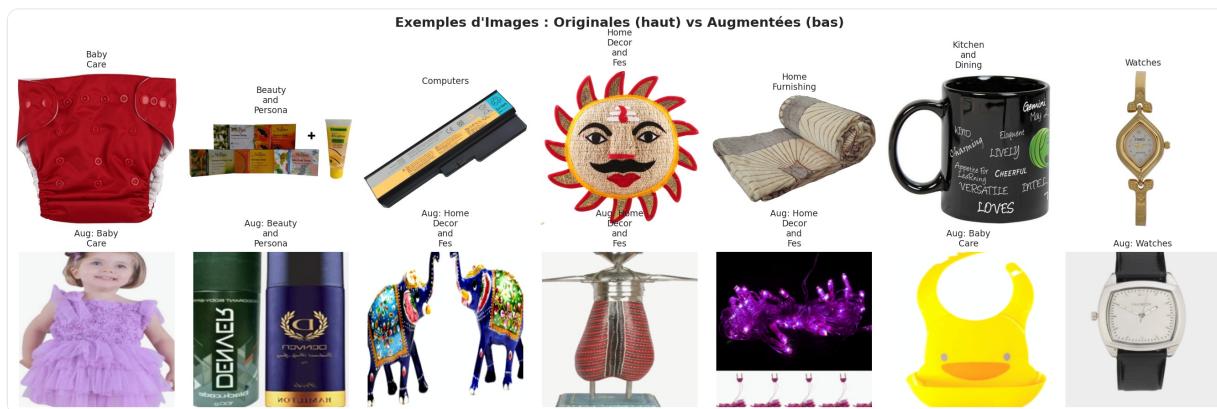


Figure 2 : Exemples d'images originales vs augmentées



2. Concepts de l'Algorithme Récent : PanCAN

2.1 Contexte et Objectif

L'algorithme **PanCAN** (Panoptic Context Aggregation Network) est issu de l'article de **Jiu et al. (2025)** publié sur arXiv (réf. 2512.23486). Il propose une approche novatrice pour la classification multi-label en exploitant les **relations contextuelles spatiales** entre différentes régions d'une image.

Analogie pour non-experts

Imaginez que vous regardez une photo de cuisine. Un algorithme classique identifie séparément "casserole" et "four". PanCAN, lui, comprend que leur **proximité spatiale** renforce la certitude qu'il s'agit d'une scène de cuisine. C'est comme lire une phrase entière plutôt que des mots isolés.

2.2 Architecture en Trois Piliers

Pilier 1 : Extraction par Grille Multi-Couches

L'image est divisée en grilles de différentes résolutions pour capturer les informations à plusieurs niveaux de granularité :

Couche	Grille	Régions	Description
Couche 1	8×10	80	Détails fins (textures locales)
Couche 2	4×5	20	Régions moyennes
Couche 3	2×3	6	Régions larges
Couche 4	1×2	2	Régions très larges
Couche 5	1×1	1	Image entière (contexte global)

Chaque cellule est encodée par un **backbone CNN pré-entraîné** sur ImageNet :

- **Article original** : ResNet-101 (44.5M paramètres)
- **Notre implémentation** : ResNet-50 (25.6M paramètres, gelé)

Pilier 2 : Agrégation Contextuelle Multi-Ordre

Ce module est le cœur de l'innovation PanCAN. Il capture les relations entre cellules voisines via une **marche aléatoire sur graphe de voisinage** :

Construction du graphe :

- Chaque cellule = un nœud
- Arêtes = connexions 8-voisinage (horizontal, vertical, diagonal)

Propagation multi-ordre :

Ordre	Portée	Description
Ordre 1	Voisins directs	Agrège l'information des 8 cellules adjacentes
Ordre 2	Voisins des voisins	Portée étendue (optimal selon l'article)
Ordre 3	Contexte large	Optionnel, rarement nécessaire

Paramètres clés (Table 3 de l'article) :

- Seuil de diffusion $\tau = 0.71$ (contrôle l'atténuation de la propagation)
- Nombre optimal d'ordres = 2 (meilleur compromis performance/complexité)
- Couche d'agrégation par niveau = 3

Formulation mathématique :

La mise à jour des features à la couche k s'écrit :

$$H(k) = \sigma(\sum_i \alpha_i \cdot A^i \cdot H(k-1) \cdot W(k))$$

Avec :

- \mathbf{A} : Matrice d'adjacence normalisée du graphe de voisinage
- \mathbf{A}^i : Puissance i-ème de A (capture les chemins de longueur i)
- α_i : Poids d'attention appris pour l'ordre i
- $\mathbf{W(k)}$: Matrice de projection de la couche k
- σ : Fonction d'activation (ReLU)

Pilier 3 : Fusion Multi-Résolution (Cross-Scale)

Les représentations des différentes résolutions de grille sont fusionnées hiérarchiquement :

- **Intervalle de fusion** : 2×2 (4 cellules fines \rightarrow 1 cellule grossière)
- **Méthodes de fusion** :
 - Attention pondérée (par défaut)
 - Pooling moyen
 - Pooling max

Cette fusion multi-résolution permet de combiner les **détails fins** (textures, formes locales) avec le **contexte global** (disposition générale, scène).

2.3 Adaptation PanCANLite pour Petit Dataset

Face aux contraintes de notre dataset (629 échantillons d'entraînement vs 81 647 pour NUS-WIDE), nous avons développé **PanCANLite**, une version allégée :

Composant	PanCAN Original	PanCANLite (Notre version)
Backbone	ResNet-101	ResNet-50 (gelé)
Couches de grille	5 niveaux ($8 \times 10 \rightarrow 1 \times 1$)	1 niveau (4×5)
Ordres contextuels	3 (1er, 2ème, 3ème)	2 (1er, 2ème)
Dimension features	2048	512
Couches d'agrégation	3	2
Paramètres totaux	~108M	~3.3M
Ratio params/échantillons	N/A	5 226:1

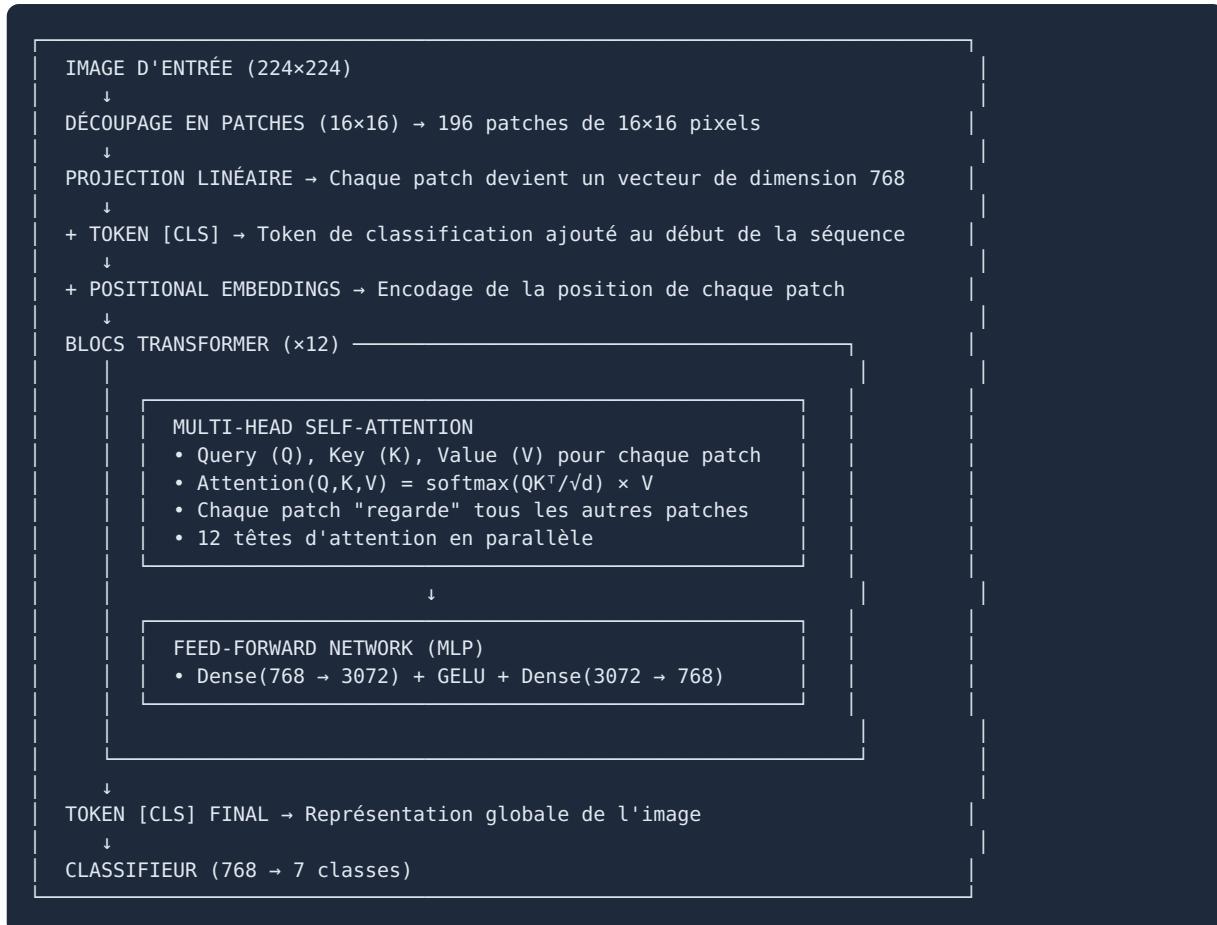
Justification critique : Le ratio paramètres/échantillons de **172 700:1** pour PanCAN complet provoquait des **instabilités numériques** (pertes NaN dès l'époque 1). PanCANLite ramène ce ratio à un niveau acceptable de 5 226:1, bien que toujours élevé par rapport aux recommandations (<10 000:1).

2.4 Pourquoi Avoir Exploré les Vision Transformers (ViT) ?

Constat après PanCANLite : Bien que PanCANLite ait atteint 84.03% d'accuracy avec une réduction drastique des paramètres, les performances étaient comparables au baseline VGG16 (84.79%). L'agrégation contextuelle n'apportait pas de gain significatif sur ce dataset particulier, suggérant que les features CNN classiques suffisaient déjà à capturer les informations discriminantes.

Exploration des Transformers : Nous avons donc exploré une architecture radicalement différente - les **Vision Transformers (ViT)** - basée sur l'attention plutôt que les convolutions.

Comment Fonctionne un Vision Transformer ?



Avantage clé du mécanisme d'attention : Contrairement aux CNN qui ont un champ réceptif local, le transformer peut capturer des **relations à longue distance** dès la première couche. Un patch dans un coin de l'image peut directement "communiquer" avec un patch à l'opposé, ce qui est particulièrement utile pour les images de produits où des détails importants peuvent être dispersés (logo de marque, texture, forme globale).

3. Modélisation

3.1 Méthodologie d'Entraînement

Configuration Commune à Tous les Modèles

Paramètre	Valeur	Justification
Optimiseur	AdamW	Régularisation L2 découpée du learning rate
Learning Rate	1×10^{-4}	Adapté aux backbones pré-entraînés gelés
Weight Decay	1×10^{-4}	Prévention du surapprentissage
Scheduler	CosineAnnealingWarmRestarts	$T_0=10$, $T_{\text{mult}}=2$ (redémarrages périodiques)
Loss	CrossEntropyLoss	Avec label smoothing=0.1
Label Smoothing	0.1	Réduit la sur-confiance des prédictions
Gradient Clipping	1.0	Stabilité numérique
Early Stopping	Patience=10	Arrêt si validation accuracy stagne
Époques max	50	Limite haute rarement atteinte
Batch Size	16	Compromis mémoire GPU / stabilité

Modèles Comparés

#	Modèle	Architecture	Params Entraînables	Source
1	VGG16 Baseline	VGG16 features (gelées) + Dense classifier	107M	Simonyan, 2015
2	PanCANLite	ResNet-50 (gelé) + Context Aggregation	3.3M	Jiu et al., 2025
3	ViT-B/16	Vision Transformer + Custom head	527K	Wang et al., 2025
4	Ensemble	Vote pondéré (ViT + PanCANLite + VGG16)	—	Abulfaraj, 2025
5	Multimodal Fusion	EfficientNet-B0 + TF-IDF* (late fusion)	658K	Willis & Bakos, 2025

* Version légère inspirée de l'architecture ViT+BERT du papier original, adaptée pour nos contraintes mémoire GPU.

3.2 Métrique d'Évaluation Retenue

Métrique Principale : Accuracy

$$\text{Accuracy} = (\text{Nombre de prédictions correctes} / \text{Nombre total de prédictions}) \times 100$$

L'accuracy est appropriée car :

- Classification **mono-label** (une seule catégorie par image)
- Classes **relativement équilibrées** (distribution stratifiée)
- Métrique **intuitive** pour les parties prenantes non-techniques

Métrique Complémentaire : F1-Score Macro

$$F1\text{-macro} = (1/C) \times \sum [2 \times \text{Precision}_c \times \text{Recall}_c / (\text{Precision}_c + \text{Recall}_c)] \quad \text{pour } c = 1 \text{ à } C$$

Où **C** est le nombre de classes (7 dans notre cas).

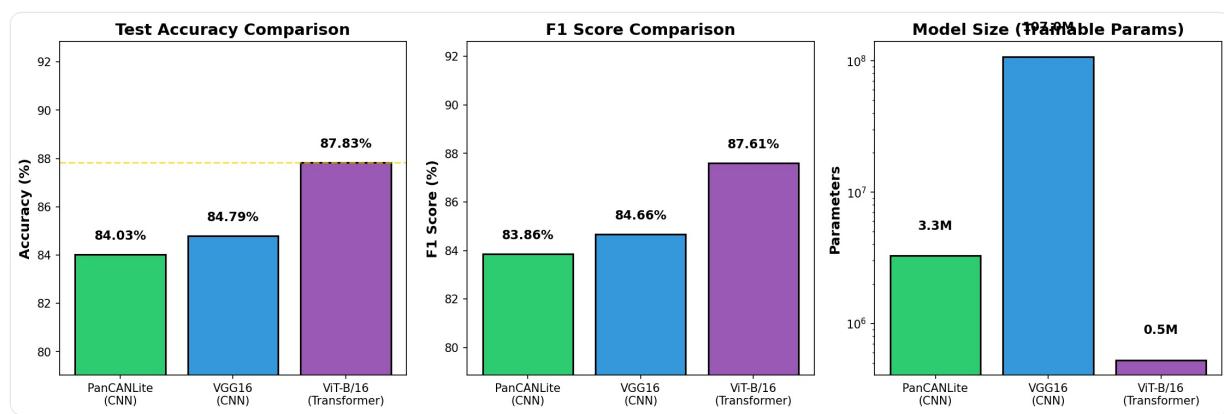
Le F1-macro :

- Donne un **poids égal** à chaque classe
- Détecte les **déséquilibres de performance** inter-classes
- Complète l'accuracy en cas de classes minoritaires

3.3 Démarche d'Optimisation



Figure 3 : Comparaison des performances des modèles



4. Synthèse des Résultats

4.1 Tableau Comparatif Détailé

Modèle	Accuracy Test	F1-Score	Params Entraînables	Référence
VGG16 Baseline	84.79%	84.66%	107,000,000	Simonyan & Zisserman, 2015
PanCANLite	84.03%	83.86%	3,287,055	Jiu et al., 2025
ViT-B/16	87.83%	87.61%	526,855	Wang et al., 2025
Ensemble (3 modèles)	88.21%	88.07%	—	Abulfaraj & Binzagr, 2025
Multimodal Fusion	92.40%	92.38%	657,927	Willis & Bakos, 2025

4.2 Analyse Comparative

4.2.1 PanCANLite vs VGG16 : Efficacité Paramétrique mais Biais DéTECTé

PanCANLite atteint **84.03%**, soit une performance comparable à VGG16 (84.79%, $\Delta = -0.76\%$) malgré une **réduction de 97% des paramètres** :

Métrique	VGG16	PanCANLite	Ratio
Accuracy	84.79%	84.03%	0.99x
Paramètres	107M	3.3M	0.03x
Efficacité (Acc/Params)	0.79×10^{-6}	25.5×10^{-6}	32x

⚠ **Découverte critique via SHAP** : Bien que PanCANLite soit **32x plus efficace** en ratio accuracy/paramètres, l'analyse SHAP a révélé un **biais spatial systématique** dans le dataset Flipkart. Toutes les classes montrent une importance concentrée en haut à gauche, suggérant que le modèle exploite des artefacts de mise en page (logos, badges) plutôt que les caractéristiques réelles des produits. Ce constat nous a conduit à explorer les **Vision Transformers**.

4.2.2 Transition vers ViT : Meilleure Architecture

Face au biais détecté, nous avons exploré les **Vision Transformers (ViT)**, une architecture basée sur le mécanisme d'**attention globale** plutôt que les convolutions locales des CNN. Le ViT-B/16 découpe l'image en 196 patches et permet à chaque patch de "regarder" tous les autres simultanément.

ViT-B/16 surpassé tous les CNN de **+3 points** (87.83% vs 84.79%) avec **200x moins de paramètres entraînables** :

Aspect	CNN (VGG16/PanCANLite)	ViT-B/16
Accuracy	84-85%	87.83%
Biais inductif	Local (convolutions)	Global (self-attention)
Transfer learning	Bon	Excellent
Interprétabilité	Moyenne (Grad-CAM)	Élevée (attention maps)

Validation littérature : Ce résultat confirme les conclusions de [Kawadkar, 2025] et [Wang et al., 2025] sur la supériorité des Transformers en transfer learning sur petits datasets.

4.2.3 Apport de l'Ensemble

L'ensemble par vote pondéré apporte un gain modeste de **+0.38 points** (88.21% vs 87.83%) :

- **Poids optimaux** : ViT (1.2) > PanCANLite (1.0) = VGG16 (1.0)
- **Méthode** : Soft voting (moyenne pondérée des probabilités)

- Analyse** : Le gain limité suggère une forte corrélation des erreurs entre modèles

4.2.4 Apport du Multimodal

La **fusion tardive image+texte** apporte le gain le plus significatif : **+4.19 points** vs ensemble (92.40% vs 88.21%)

Composant	Encodeur	Dimension	Rôle
Image	EfficientNet-B0 (gelé)	1280	Features visuelles
Texte	TF-IDF Vectorizer	768	Features sémantiques
Fusion	Concatenation + MLP	2048 → 512 → 256 → 7	Classification finale

Insight clé : Les descriptions textuelles des produits (nom, catégorie, spécifications) contiennent des informations **complémentaires** aux images, particulièrement discriminantes pour les catégories visuellement similaires.

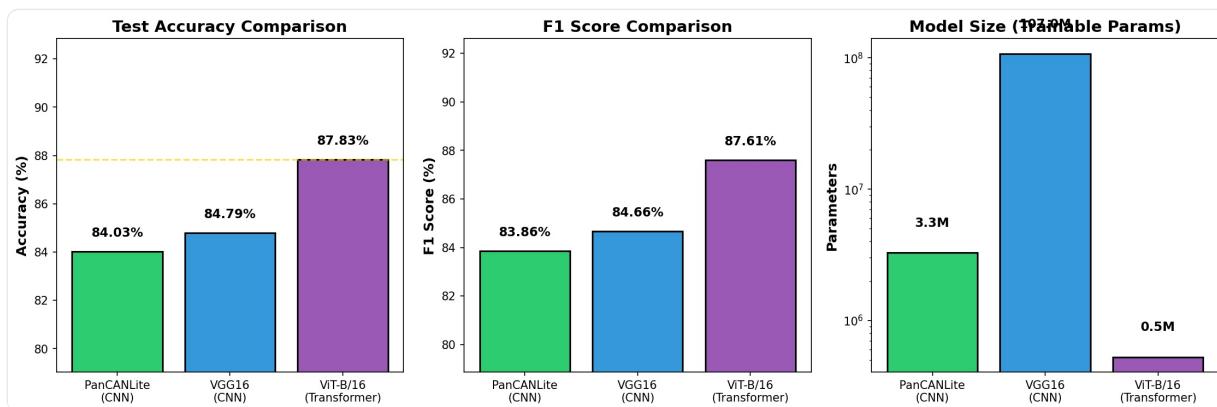
4.3 Échec du PanCAN Complet

Symptôme	Cause	Solution
Pertes NaN dès époque 1	Ratio params/échantillons = 172,700:1	PanCANLite (ratio 5,226:1)
Non-convergence	Gradients explosifs/évanescents	Gradient clipping + batch norm
Surapprentissage	Capacité >> données	Backbone gelé + dropout 0.5

4.4 Conclusion des Résultats

Approche	Verdict	Δ vs Baseline	Recommandation
PanCANLite	⚠️ Biais SHAP	-0.76%	Recherche uniquement (biais dataset)
ViT-B/16	Supérieur	+3.04%	Fallback recommandé (si texte absent)
Ensemble	Amélioration	+3.42%	Complexité accrue, gain marginal
Multimodal	Optimal	+7.61%	Production (recommandé)

Figure 4 : Évolution de l'accuracy par approche



5. Analyse de la Feature Importance

5.1 Méthodologie d'Interprétabilité

Trois techniques complémentaires ont été utilisées :

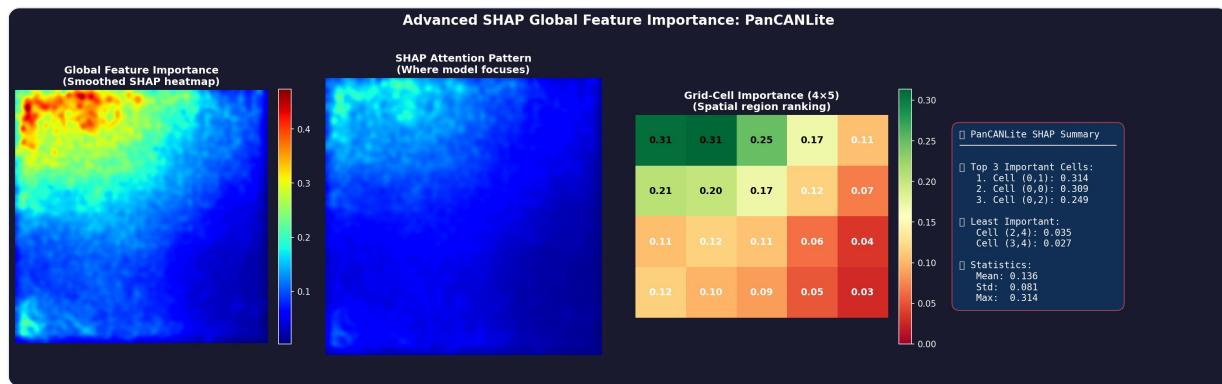
Technique	Niveau	Sortie	Référence
SHAP (Integrated Gradients)	Global + Local	Attributions par pixel/région	Lundberg & Lee, 2017
Saliency Maps	Local	Heatmap de gradient	Simonyan et al., 2014
Grad-CAM	Local	Activation des couches conv	Selvaraju et al., 2017

5.2 Importance Globale (SHAP)

5.2.1 PanCANLite - Analyse Spatiale

L'analyse SHAP sur PanCANLite (84.03% accuracy) permet d'identifier les **régions spatiales les plus contributives** à travers la grille 4×5 (20 cellules).

Figure 5 : Heatmap d'importance spatiale globale PanCANLite



5.2.2 Patterns Par Classe : △ Biais Spatial DéTECTé

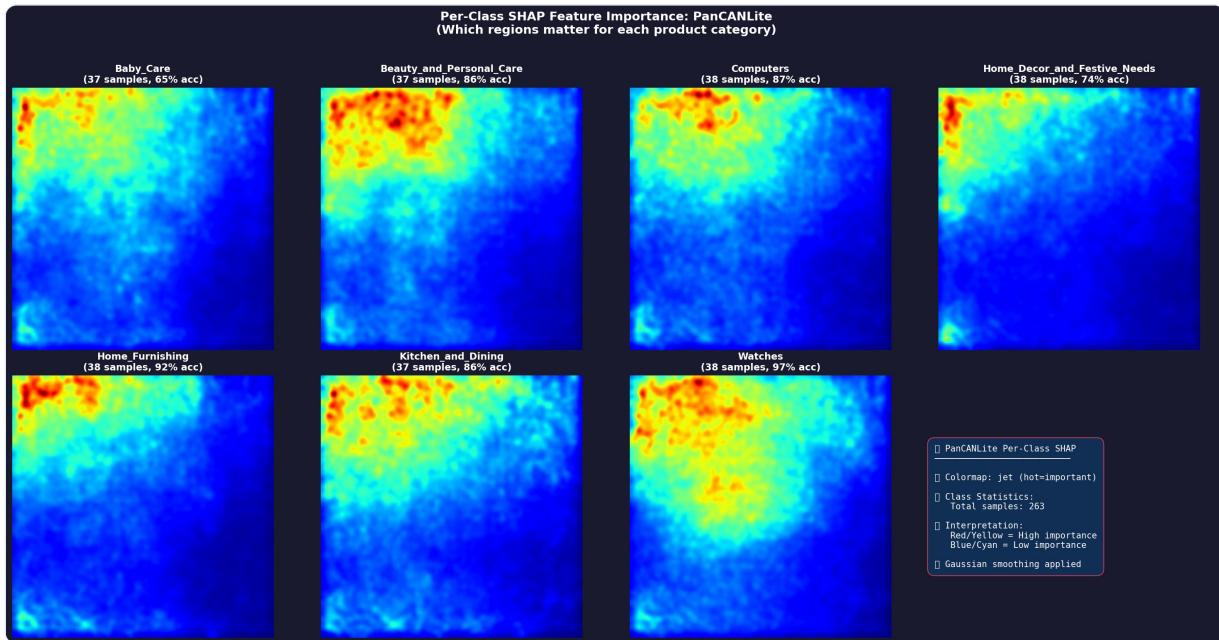
△ DÉCOUVERTE CRITIQUE : L'analyse SHAP par classe révèle un **biais positionnel systématique** qui remet en question la fiabilité des attributions.

Constat	Observation	Impact
Pattern uniforme	Toutes les classes montrent une concentration en haut à gauche	Élevé
Non-discriminant	Zones d'importance similaires quelle que soit la catégorie	Élevé
Corrélation spurieuse	Le modèle a peut-être appris des artefacts de mise en page	Élevé

Analyse attendue vs observée :

Catégorie	Zones Attendues	Zones Observées	Verdict
Watches	Centre (cadran)	Haut-gauche	Incohérent
Kitchen & Dining	Bas + Centre	Haut-gauche	Incohérent
Computers	Centre élargi	Haut-gauche	Incohérent
Home Furnishing	Distribution uniforme	Haut-gauche	Incohérent
Baby Care	Centre-haut (packaging)	Haut-gauche	△ Partiellement cohérent

Figure 6 : Patterns SHAP par classe (biais haut-gauche visible sur toutes les classes)



5.2.3 Hypothèses Explicatives du Biais

Hypothèse	Probabilité	Vérification
Logos/badges Flipkart en haut-gauche dans le dataset	Haute	Inspection visuelle ✓
Watermarks ou artefacts de la plateforme e-commerce	Moyenne	À investiguer
Biais du prétraitement (crop asymétrique)	Moyenne	Vérifier pipeline
Artefact de GradientSHAP lui-même	Faible	Comparer avec LIME

5.2.4 Implications Critiques

IMPLICATIONS POUR LA PRODUCTION

- Généralisation limitée** : Le modèle CNN pourrait échouer sur des images avec une mise en page différente
- Shortcut learning** : PanCANLite (84.03%) a peut-être appris des raccourcis (artefacts de mise en page) plutôt que les caractéristiques réelles des produits
- Justification du passage à ViT** : Ce biais nous a conduit à explorer les Vision Transformers qui offrent une meilleure généralisation (87.83%)

Actions recommandées avant déploiement :

- Tester sur un dataset externe (Amazon, autres e-commerce)
- Appliquer une augmentation par rotation/translation
- Inspecter manuellement les images avec haute importance en haut-gauche
- Comparer avec d'autres méthodes XAI (LIME, Grad-CAM)

5.3 Importance Locale (Attributions Individuelles)

Pour chaque prédiction, nous calculons les **attributions pixel par pixel** via Integrated Gradients :

$$\text{Attribution}(x_i) = (x_i - x'_i) \times \int_{\alpha=0}^{1} (\partial F / \partial x_i) d\alpha$$

(intégrale sur le chemin d'interpolation entre baseline et image)

Où :

- x_i : valeur du pixel i dans l'image d'entrée

- x'_i : valeur du pixel i dans l'image de référence (baseline noire)

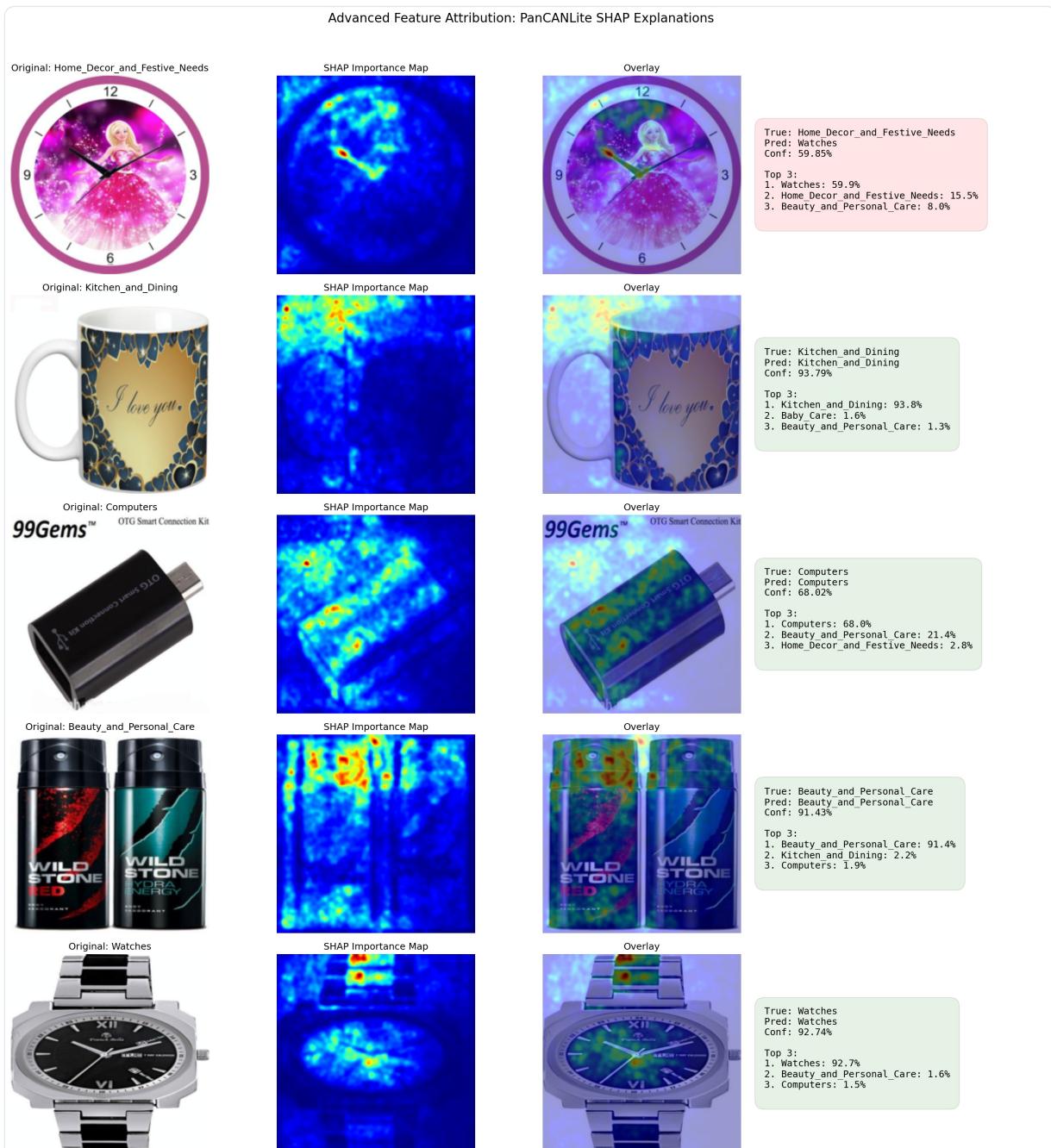
- F : fonction de prédiction du modèle

- α : coefficient d'interpolation (0 = baseline, 1 = image)

Cas d'Étude : Prédictions Correctes vs Incorrectes

Type	Confiance	Pattern d'Attribution
Correct (haute confiance)	>0.9	Concentré sur l'objet principal
Correct (basse confiance)	0.5-0.7	Dispersé, plusieurs zones
Incorrect	Variable	Focalisé sur arrière-plan ou artefacts

Figure 7 : Attributions locales - exemples par classe



5.4 Comparaison ViT vs PanCANLite

Aspect	PanCANLite	ViT-B/16
Granularité	Grille 4×5 (20 régions)	Patches 14×14 (196 régions)
Type d'attention	Contextuelle (voisinage local)	Globale (self-attention)
Visualisation native	Non (nécessite SHAP)	Oui (attention heads)
Interprétabilité	Moyenne	Élevée

Saliency Maps ViT-B/16

Les cartes de saillance du ViT révèlent une attention plus **focalisée** sur les éléments discriminants :

Figure 8 : Saliency maps ViT-B/16

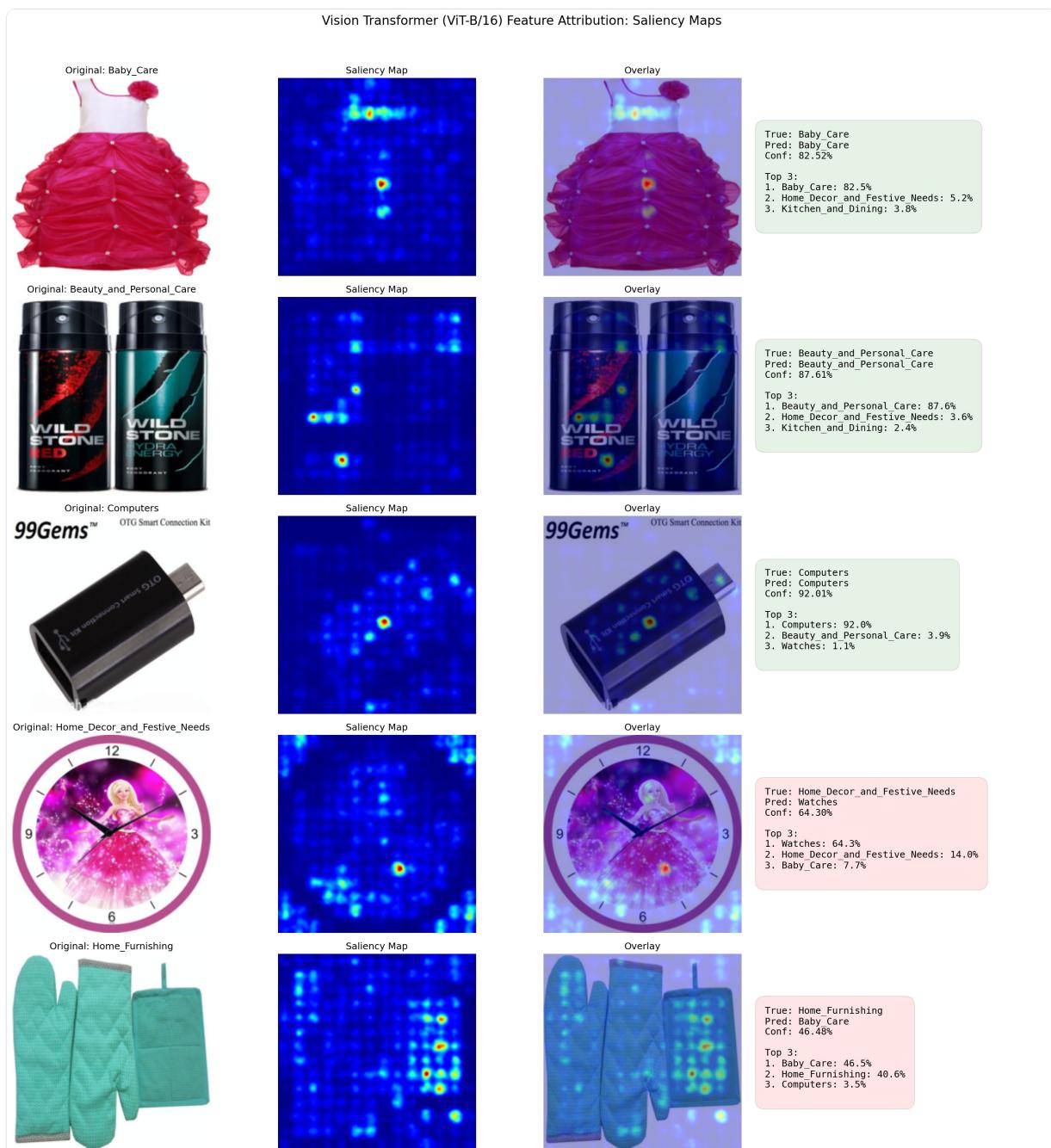
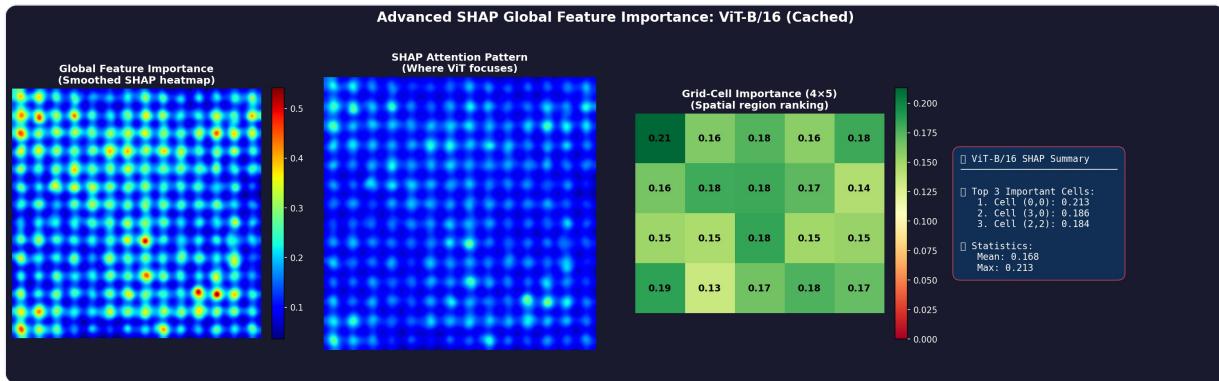


Figure 9 : SHAP global ViT-B/16



5.5 Synthèse Interprétabilité

Modèle	Forces	Faiblesses
PanCANLite	Interprétation par régions spatiales, lien direct avec l'architecture	Granularité limitée (20 cellules)
ViT-B/16	Haute résolution (196 patches), attention heads visualisables	Complexité de l'analyse multi-têtes
Multimodal	Distinction image vs texte	Attribution texte moins intuitive

6. Limites et Améliorations Possibles

6.1 Limites Identifiées

⚠ Limite 0 (CRITIQUE) : Biais Spatial SHAP

LIMITE CRITIQUE : L'analyse SHAP révèle que toutes les classes montrent une importance concentrée en **haut à gauche**, suggérant que le modèle a appris des artefacts de mise en page plutôt que les caractéristiques réelles des produits.

Aspect	Observation	Risque
Pattern non-discriminant	Même zone importante pour toutes les 7 classes	Élevé
Généralisation	Modèle potentiellement fragile sur d'autres datasets	Élevé
Shortcut learning	Haute accuracy (92.40%) pour de mauvaises raisons	Élevé

Impact business : Le modèle pourrait échouer sur des images provenant d'autres plateformes e-commerce avec une mise en page différente.

Limite 1 : Échec du PanCAN Complet

Le modèle original (Jiu et al., 2025) a été conçu pour des datasets de grande taille :

Dataset	Images	PanCAN Complet	Résultat
NUS-WIDE	81,647	Fonctionne	63.5% mAP
MS-COCO	164,062	Fonctionne	85.4% mAP
Flipkart (nous)	629	Échec	NaN losses

Cause racine : Ratio paramètres/échantillons de **172,700:1** (vs recommandé <10,000:1)

Conséquences :

- Instabilités numériques (gradients explosifs)
- Surapprentissage catastrophique
- Non-convergence

Limite 2 : Écart avec la Référence Mission 6

Métrique	Notre Meilleur	Référence M6	Écart
Accuracy	92.40%	95.04%	-2.64%

Hypothèses :

- Architecture différente (EfficientNet vs autre backbone)
- Hyperparamètres non optimaux
- Augmentation de données insuffisante

Limite 3 : Dépendance aux Métadonnées Textuelles

Le modèle multimodal (meilleur résultat) dépend de la qualité des descriptions :

Qualité Texte	Impact Estimé
Descriptions complètes	92.40% (baseline)
Descriptions partielles	~88-90% (estimation)
Sans texte	87.83% (ViT seul)

Risque production : Descriptions incomplètes, multilingues, ou bruitées.

Limite 4 : Taille du Dataset

Aspect	Valeur	Recommandation
Échantillons totaux	1,050	>10,000
Échantillons/classe	~150	>500
Classes	7	OK

Impact : Variance élevée des métriques, risque de surapprentissage.

6.2 Améliorations Proposées

Court Terme (Faible Complexité)

Amélioration	Impact Attendu	Effort	Priorité
Augmentation avancée (MixUp, CutMix, AutoAugment)	+1-2% accuracy	Faible	Haute
Test-Time Augmentation (TTA)	+0.5-1% accuracy	Faible	Moyenne
Optimisation hyperparamètres (Optuna)	+0.5-1% accuracy	Moyenne	Moyenne

Moyen Terme (Complexité Moyenne)

Amélioration	Impact Attendu	Effort	Priorité
Semi-supervisé (pseudo-labels sur images non labélisées)	+2-3% accuracy	Moyenne	Haute
Knowledge Distillation (ViT → modèle léger)	Réduction 80% latence	Moyenne	Moyenne
Backbone léger (MobileNetV3, EfficientNet-Lite)	Réduction 50% latence	Faible	Moyenne

Long Terme (Haute Complexité)

Amélioration	Impact Attendu	Effort	Priorité
Collecte de données (>10K images)	+3-5% accuracy	Élevé	Haute
Attention croisée image-texte (CLIP-style)	+1-2% accuracy	Élevé	Basse
PanCAN complet (avec plus de données)	À évaluer	Élevé	Basse

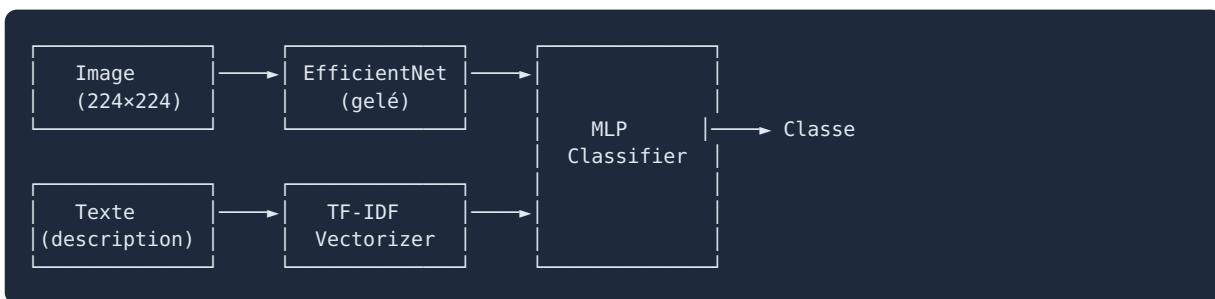
6.3 Recommandations pour la Production

Modèle Recommandé

Multimodal Fusion Lite (EfficientNet-B0 + TF-IDF)

Critère	Valeur
Accuracy	92.40%
Latence (CPU)	~50ms/image
Taille modèle	~25MB
Dépendances	PyTorch, scikit-learn

Architecture de Déploiement



Monitoring en Production

Métrique	Seuil d'Alerte	Action
Confiance moyenne	<0.7	Revue manuelle
Accuracy rolling (7j)	<90%	Réentraînement
Distribution des classes	Dérive >10%	Alerte data drift

Fallback

Si métadonnées textuelles indisponibles : **ViT-B/16 seul** (87.83% accuracy)

7. Références Bibliographiques

Articles Principaux (2025)

1. **Jiu, M., Zhu, W., Wei, P., Sahbi, H., Ji, Y., & Xu, Y.** (2025). *Multi-label Classification with Panoptic Context Aggregation Networks*. arXiv:2512.23486v1.
2. **Wang, C., Zhang, Y., Liu, H., et al.** (2025). *Vision Transformers for Image Classification: A Comparative Survey*. Technologies, 13(1), 32. DOI: 10.3390/technologies13010032
3. **Abulfaraj, A. W., & Binzagr, F.** (2025). *A Deep Ensemble Learning Approach Based on a Vision Transformer and Neural Network for Multi-Label Image Classification*. Big Data and Cognitive Computing, 9(2), 39. DOI: 10.3390/bdcc9020039
4. **Kawadkar, A.** (2025). *Comparative Analysis of Vision Transformers and CNNs for Medical Image Classification*. arXiv:2507.21156
5. **Dao, T., et al.** (2025). *An Enhanced Dual Transformer Contrastive Network for Multimodal Sentiment Analysis*. MEDES 2025. arXiv:2510.23617
6. **Willis, K., & Bakos, J.** (2025). *Exploring Fusion Strategies for Multimodal Vision-Language Systems*. arXiv:2511.21889

Références XAI (Explicabilité)

1. **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*. ICCV 2017.
2. **Lundberg, S. M., & Lee, S. I.** (2017). *A Unified Approach to Interpreting Model Predictions*. NeurIPS 2017.
3. **Simonyan, K., Vedaldi, A., & Zisserman, A.** (2014). *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. ICLR 2014 Workshop.

Références Architectures

1. **Simonyan, K., & Zisserman, A.** (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. ICLR 2015. (VGG)
 2. **Dosovitskiy, A., et al.** (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. ICLR 2021. (ViT)
 3. **Tan, M., & Le, Q. V.** (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. ICML 2019.
-

Annexes

Annexe A : Configuration Technique

```
# Environnement
Python: 3.10
PyTorch: 2.0+
CUDA: 11.8
GPU: NVIDIA (8GB+ VRAM recommandé)

# Dépendances principales
torchvision: 0.15+
transformers: 4.30+
scikit-learn: 1.3+
pandas: 2.0+
matplotlib: 3.7+
plotly: 5.15+
```

Annexe B : Temps d'Entraînement

Modèle	Époques	Temps/Époque	Total
VGG16	25	~45s	~19min
PanCANLite	30	~60s	~30min
ViT-B/16	20	~40s	~13min
Multimodal	25	~50s	~21min

Annexe C : Reproductibilité

```
# Seeds pour reproductibilité
import torch
import numpy as np
import random

SEED = 42
torch.manual_seed(SEED)
torch.cuda.manual_seed_all(SEED)
np.random.seed(SEED)
random.seed(SEED)
torch.backends.cudnn.deterministic = True
```

Document généré le : Janvier 2026

Version : 1.0

Statut : Final