


# Note méthodologique : preuve de concept de l'algorithme CLIP d'Open AI

## Dataset retenu

*Les données utilisées pour entraîner et évaluer le modèle proviennent d'un précédent travail de classification de biens de consommations élaboré pour l'entreprise Place de marché dans le cadre du lancement de leur plateforme d'e-commerce.*

*Les données à ma disposition se composent d'un dossier contenant 1050 images de biens de consommations et d'un fichier tabulaire au format CSV renseignant la description et la catégorie de produit de chaque image.*

|\_  Images

|\_  flipkart\_com-ecommerce\_sample\_1050

*Après un premier pré-traitement nous pouvons identifier 7 (sept) catégories de produit avec 150 (cent cinquante) images pour chacune.*

Catégories	Nombre d'images
Home Furnishing	150
Baby Care	150
Watches	150
Home Decor & Festive Needs	150
Kitchen & Dining	150
Beauty and Personal Care	150
Computers	150

*En voici quelques exemples :*

Home Decor & Festive Needs



Watches



Baby Care



Home Furnishing



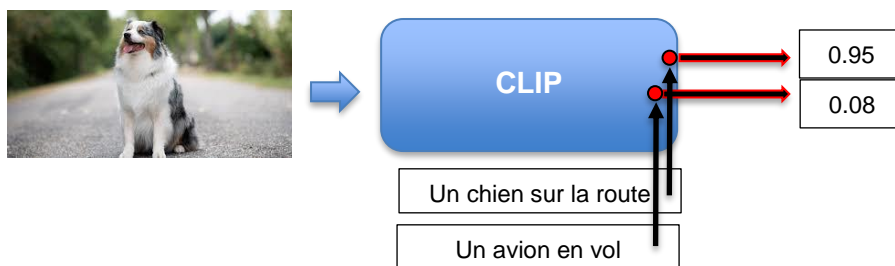
## Les concepts de l'algorithme récent

C'est l'algorithme CLIP d'Open AI, datant de 2021, qui a été choisi pour faire une comparaison « state-of-the-art » avec le travail de modélisation précédemment réalisé sur les produits Prêt à dépenser.

L'algorithme a fait l'objet d'une publication scientifique disponible [ici](#).

Le code complet de CLIP est open source sur [GitHub](#).

CLIP a été entraîné pour calculer la similarité entre une image et un texte. En confrontant une image et un texte de description, la similarité renvoyée prendra une valeur entre -1 et 1. -1 signifie que le texte et l'image sont totalement opposés et 1 qu'ils sont très similaires. En revanche pour une valeur de 0, le texte et l'image n'ont aucun rapport.



Pour obtenir ces résultats, CLIP transforme l'image et le texte en deux vecteurs de dimensions fixes puis calcule le cosinus de l'angle (similarité cosinus). Ainsi deux vecteurs proches dans l'espace (~même direction et ~même sens) auront une similarité cosinus proche de 1.



Figure 1. Exemple de similarité cosinus

Les neurones d'encodage d'image et de texte de CLIP ont été entraînés sur une énorme quantité de données images et textes disponibles sur internet. Les poids du modèle ont été ajustés de façon à augmenter la similarité des images et des textes correspondants, et inversement de réduire la similarité des images et des textes n'ayant aucun rapport. En raison du très grand volume de données fournis à CLIP pour son entraînement, cela

lui donne de très bonnes performances en « zero-shot », c'est-à-dire en lui donnant un seul lot d'images et de textes sans le réentraîner spécifiquement sur ce dernier.

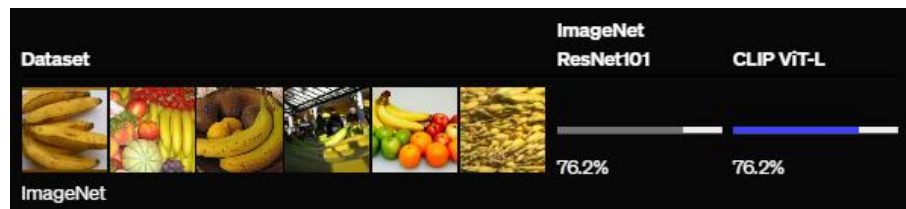


Figure 2. Précision de CLIP (en "zero-sho") et de ResNet101 sur des images ImageNet

# La modélisation

## Découpage des données

Les données images sont séparées en trois parties, la première pour entraîner l'algorithme, la seconde pour optimiser ses performances par une évaluation à chaque boucle d'entraînement, et la dernière pour son évaluation finale (non lues dans l'entraînement). Pour chaque lot d'images, on prend soin de conserver la même proportion d'images de chaque catégorie de produit.

60% - train	630 images
20% - validation	210 images
20% - test	210 images

Figure 3. Découpage des données images

## Architecture du modèle

Pour l'entraînement, le modèle sera composé de l'encodeur d'image de CLIP suivi de deux couches de réseau de neurones linéaires avec un fonction d'activation reLU. Cette architecture classique sera simple à mettre en œuvre et à entraîner. De cette manière, le vecteur calculé par CLIP correspondant à l'image sera l'entrée du réseau de neurone linéaire  $x A^T + b$  avec  $x$  et  $b$  les vecteurs des coefficients et des biais et  $A$  le vecteur de l'image encodée.

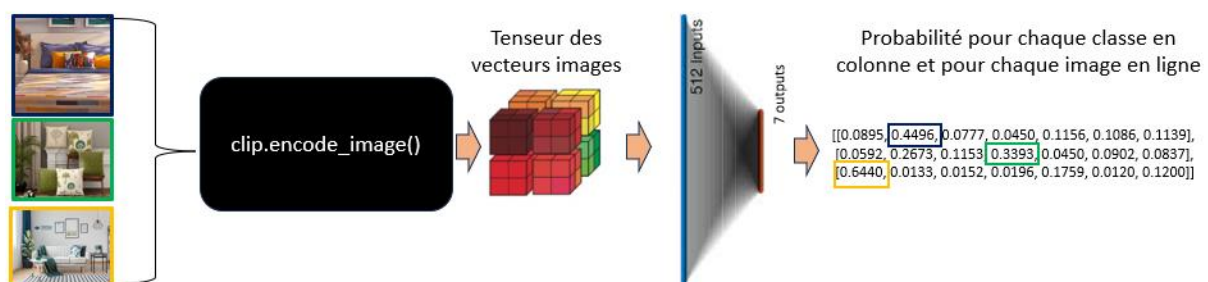


Figure 4. Architecture du modèle

Ainsi, seules les deux couches du réseau de neurone après encodage est entraîné.

## Fonction de perte

L'erreur à minimiser à chaque boucle d'entraînement sera le Log-Loss (aussi appelé cross-entropy Loss), ce qui revient à minimiser pour chaque classe la somme des écarts entre la probabilité d'appartenance prédite par le modèle et la probabilité réelle des images.

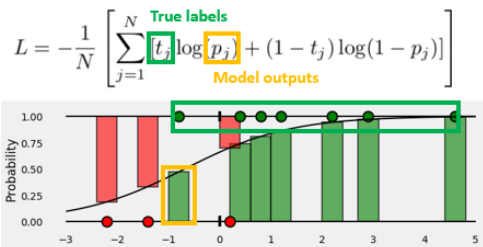


Figure 5. Illustration graphique du log loss

## Optimiseur

Afin de mettre à jour les poids des neurones selon les variations de l'erreur à chaque itération, l'optimiseur Adam a été choisi. Adam est une méthode améliorée de la descente de gradient stochastique (SGD) qui permet une meilleure convergence du fait que le pas de mise à jour des poids varie en fonction des gradients calculés. Plus les gradients (taux de variation des erreurs en fonction des poids) sont grands, plus le pas (vitesse) de mise à jour sera réduit, et inversement.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta}{\epsilon + \sqrt{\hat{\mathbf{v}}_{t+1}}} \hat{\mathbf{m}}_{t+1}$$

$\mathbf{w}_{t+1}$ : poids à l'itération  $t + 1$

$\mathbf{v}_{t+1}$  moment d'ordre deux du gradient à  $t + 1$

$\hat{\mathbf{m}}_{t+1}$  moment d'ordre un du gradient à  $t + 1$

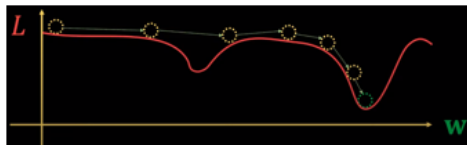


Figure 6. Illustration d'une optimisation Adam

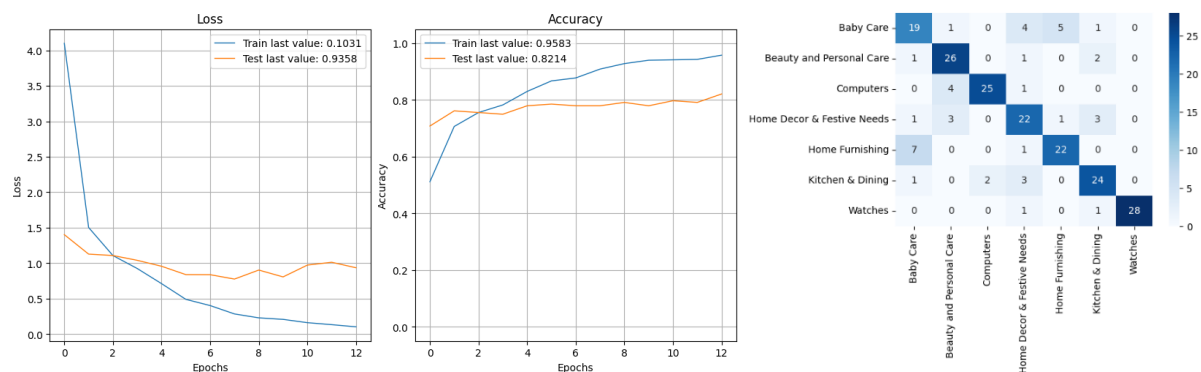
# Une synthèse des résultats

## Modèle VGG16

Dans le cadre de la classification des produits pour Prêt à dépenser, nous avons entraîné les dernières couches de neurones d'un algorithme VGG16 en adaptant la sortie pour nos sept classes de produits.

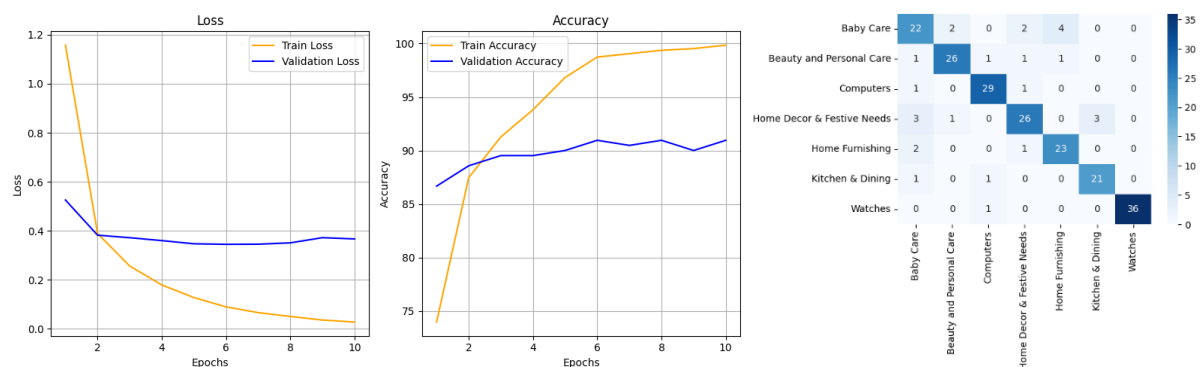
Nous avons la même fonction de perte (cross entropy) et l'optimiseur RMSProp (Root Mean Square Propagation) qui module les pas de mise à jour en fonction des gradients d'erreurs.

Nous avons entraîné le modèle sur 630 images. Il n'y avait plus d'amélioration significative des taux d'erreur avec plus de 8 itérations. Nous obtenons une précision générale de 80% sur l'ensemble des 210 images de test.



## Modèle CLIP

En comparaison, le modèle mis en place avec CLIP a atteint sa meilleure performance au bout de la troisième itération. Nous avons près de 90% de précision sur l'ensemble des 210 images de test.



## Conclusion sur les résultats

Le modèle précédent basé sur l'architecture de l'algorithme VGG16 obtient une précision de 80% au bout de 5-8 itérations tandis que le nouveau modèle basé sur l'algorithme

*CLIP permet en seulement 3 itérations de dépasser cette performance avec près de 90% de précision.*

*Autrement dit, le nouveau modèle est plus précis en nécessitant moins d'itération d'entraînement que l'ancien modèle.*

# L'analyse de la feature importance globale et locale du nouveau modèle

## Importance locale

*La méthode LIME (Local Interpretable Model-agnostic Explanations) a été utilisée pour l'analyse de la feature importance des images dans les prédictions du modèle.*

*LIME est une méthode simple de mise en œuvre pour mettre en évidence les zones (segments ou super pixels) les plus significatives dans une image pour la prédiction de sa classification par un modèle.*

*Pour une image donnée, LIME crée plusieurs copies. Pour chacune d'elles, certaines zones (segments) similaires sont bruitées (les pixels sont mis en gris ou en noir). Les probabilités d'appartenance des catégories sont calculées par notre modèle. Ensuite, un modèle linéaire simple est entraîné sur les valeurs d'écarts de probabilité obtenus pour les copies de l'images perturbées et pour l'image originale. Les coefficients appris du modèle linéaire fournissent une mesure de l'importance relative des différents segments de l'image. Ces importances relatives peuvent être visualisées graphiquement.*

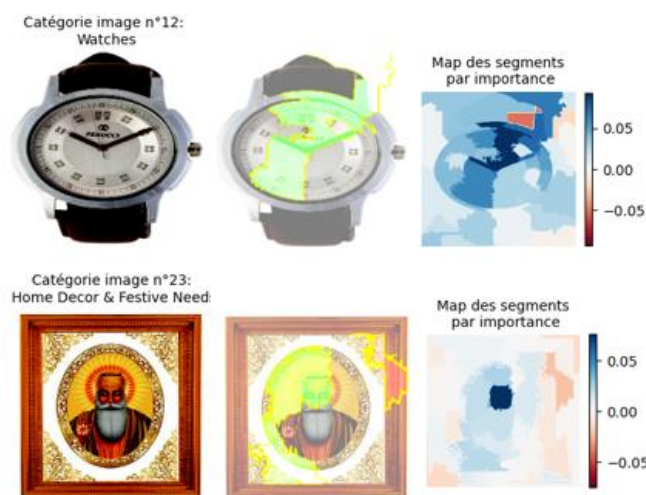


Figure 7. Exemple des segments importants pour deux images

*Pour l'illustration ci-dessus, nous avons généré 100 images avec des perturbations pour calculer le niveau d'importance de 100 segments sur les images avec la méthode LIME. Nous pouvons voir par exemple que pour la photo de **montre**, les segments les plus importants incluent notamment une partie du cadran et de l'aiguille. Pour l'image du **cadre religieux**, c'est la zone centrale incluant le visage du personnage.*

## Importance globale

*Pour avoir une estimation de l'importance globale des caractéristiques des images dans les prédictions, nous nous sommes basés sur la même démarche. LIME permet de*



calculer l'importance des différents segments d'une image à la fois. Ainsi, pour déterminer les segments les plus importants au niveau global pour une catégorie de produit donnée, nous avons moyenné les valeurs d'importances calculées sur un échantillon d'image de cette catégorie.

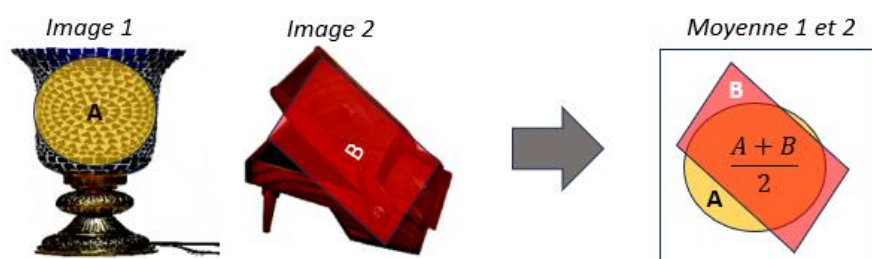


Figure 8. Illustration de la moyenne d'importance des segments A et B sur 2 images nommées 1 et 2

Ainsi, avec la génération de 100 segments par image nous avons moyenné l'importance des segments sur 10 images par catégorie de produit, soit un total de 70 images en entrée pour le traitement et de 700 images générées pour les calculs.

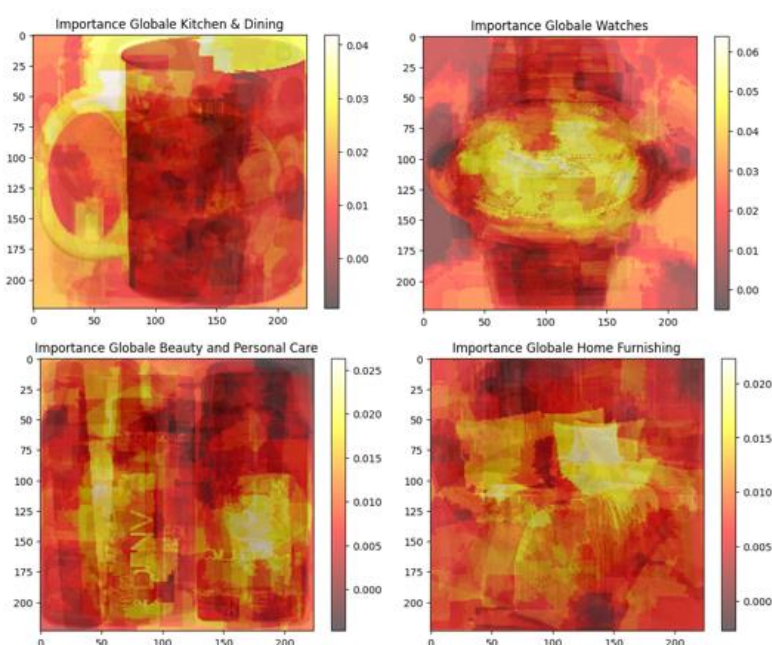


Figure 9. Importance globale pour 4 catégories de produits

L'illustration ci-dessus montre l'importance moyenne des segments pour quatre catégories par degré d'intensité relative. L'échantillon utilisé pour la catégorie **cuisine** regroupe beaucoup d'images de tasse, ainsi on peut voir que les poignées des tasses sont souvent des éléments importants pour la classification de ces produits par le modèle. Pour les **montres**, c'est la zone centrale avec le cadran et les aiguilles qui prédomine. Pour la catégorie **soins de beauté**, on peut apercevoir des écritures en surbrillance claire, ce qui montre que le modèle reconnaît ces produits notamment grâce aux écritures sur les flacons. Pour la catégorie **maison**, on peut reconnaître des formes d'oreillers et de draps de lit qui sont des éléments cruciaux pour leur.

# Les limites et les améliorations possibles

## Modélisation

*Tel que présenté, le modèle montre un plafond de performance avec une précision maximum de 90% dans la classification des produits. D'un côté nous ne disposons que de 1050 images pour l'entraînement et l'évaluation. D'un autre côté, bien que CLIP ait été entraîné sur un très grand nombre d'images, il est possible que l'extraction des caractéristiques sur notre lot de données soit limité. En effet, CLIP a par exemple pu voir des images très variées incluant des paysages, des animaux, des fruits etc. Or nos images représentent principalement des objets (montres, tasses, tableaux, rideaux, décorations, etc.), ce qui est un périmètre très restreint.*

*Plusieurs approches sont envisageables pour apporter des améliorations.*

*L'une d'entre elles pourrait être d'entraîner les dernières couches du transformer de la partie textuelle et les dernière couches du CNN de la partie image de CLIP en forçant la convergence de similarité cosinus des vecteurs de nos images et des vecteurs de nos catégories textuelles. De cette façon, nous exploiterions le plein potentiel de CLIP (avec à la fois son transformer et son CNN) en le spécialisant sur nos données.*

*Une autre approche pourrait être d'améliorer la modélisation présentée plus haut en ajoutant une ou plusieurs couches entièrement connectées à la sortie de l'encodeur image de CLIP pour gagner en généralisation et en performance. A cela nous pourrions en plus faire de l'augmentation de données en créant artificiellement une vingtaine d'images modifiées par catégorie (rotation, zoom, bruit aléatoirement générés sur chaque duplicata).*

## Interprétabilité

*La méthode LIME telle qu'utilisée présente quelques inconvénients. LIME est adapté pour visualiser localement les segments les plus important dans une image. Cependant pour généraliser l'importance dans toute une catégorie nous pourrions rencontrer une limite. En effet, dans notre cas nous avons superposé les niveaux d'importances d'un échantillon de dix images pour chaque catégorie. Or, en augmentant la taille de l'échantillon pour gagner en généralité dans l'analyse, il pourrait devenir difficile d'interpréter visuellement les résultats tant la dispositions/localisation des formes/segments importants peuvent varier d'une image à l'autre. Ainsi, pour gagner en interprétabilité dans l'importance globale, nous pourrions par exemple appliquer la méthode image\_plot de SHAP en parallèle de LIME pour apporter des informations sous un autre point de vue sur les régions des images ayant les plus hauts niveaux d'importance dans la classification.*

*Nous pourrions dans un premier temps faire un premier travail de sélection d'une image représentative de chaque catégorie. Dans un second temps, nous pourrions confronter les prédictions de ces images dans une visualisation générées avec image\_plot.*