**Note méthodologique : preuve de concept**

## **Dataset retenu**

C’est un jeu de données de biens à la consommation venant de l’entreprise fictive « Place de marché ». Il possède des données textuelles associées à des images pour chaque produit (un peu plus de 1000 images). Il y a plusieurs catégories de produits (informatique, montres, entretien de la maison…).

## **Les concepts de l’algorithme récent (CLIP)**

**1. Contexte et travaux connexes**

CLIP (Contrastive Language-Image Pre-training) repose sur des avancées en **apprentissage zéro-shot** et **modélisation multimodale**. Inspiré de travaux antérieurs, il exploite le **langage naturel** pour améliorer la généralisation en vision par ordinateur.

CLIP s’inscrit dans une approche plus récente qui combine **langage naturel** et vision à l’aide des Transformers. Il s’appuie sur des modèles comme **VirTex**, **ICMLM** et **ConVIRT**, chacun explorant différentes méthodes d’apprentissage supervisé par le texte.

Grâce à ces avancées, CLIP renforce la capacité des modèles à généraliser sans données spécifiques d’entraînement, améliorant ainsi les performances du **transfert zéro-shot.**

**2. Approche**

CLIP repose sur une **tâche de pré-entraînement simple mais à grande échelle** pour atteindre des formances compétitives en **classification zéro-shot**.

Plutôt que d’utiliser un entraînement supervisé classique, CLIP s’appuie sur **les textes associés aux images sur internet**. Le modèle apprend en identifiant le bon texte parmi **plus de 30 000 choix aléatoires**, ce qui l’oblige à reconnaître un **large éventail de concepts visuels**.

Grâce à cet apprentissage, CLIP peut être appliqué à **n’importe quelle tâche de classification visuelle** sans entraînement supplémentaire. Il suffit de lui fournir des descriptions textuelles correspondant aux classes cibles pour qu’il attribue l’image à la catégorie la plus probable.

**2.1 Coût des jeux de données**

Les modèles de deep learning nécessitent **d’énormes quantités de données**. Les modèles de vision sont généralement entraînés sur des **jeux de données annotés manuellement**, qui sont **coûteux à produire** et limités à un ensemble restreint de concepts visuels.

**Par exemple, ImageNet, l’un des plus grands projets de ce type, a mobilisé plus de 25 000 travailleurs pour annoter 14 millions d’images couvrant 22 000 catégories d’objets.**

A l’inverse, **CLIP s’entraîne directement sur des paires texte-image disponibles publiquement sur internet**, supprimant ainsi la nécessité de constituer des **jeux de données annotés coûteux**. Cette réduction de la dépendance aux données labellisées a déjà été étudiée dans des travaux antérieurs, notamment via :

* *L’apprentissage auto-supervisé*
* *Les méthodes contrastives*
* *L’auto-entraînement*
* *Les modèles génératifs*

**2.2 Généralisation limitée des modèles classiques**

Un modèle entraîné sur **ImageNet** excelle à classifier **ses 1000 catégories spécifiques**, mais ne peut pas généraliser sans nouveau jeu de données, nouvelle architecture et réentraînement.

A l’inverse, CLIP peut être appliqué immédiatement à une large variété de tâches de classification sans nécessiter d’exemples d’entraînement supplémentaires.

Pour adapter CLIP à une nouvelle tâche, il suffit de **décrire textuellement les catégories visuelles souhaitées**. CLIP **génère alors automatiquement un classificateur linéaire** basé sur ses représentations visuelles, souvent avec une précision proche des modèles entièrement supervisés.

**3. Limitations**

Bien que CLIP soit performant pour reconnaître des objets courants, **il rencontre des difficultés sur des tâches plus abstraites ou systématiques**, comme par exemple :

* *Compter le nombre d’objets dans une image*
* *Evaluer la distance du véhicule le plus proche dans une photo*

Sur ces deux types de tâches, CLIP en zéro-shot est à peine meilleur qu’un choix aléatoire. CLIP est également **moins efficace** que des modèles spécialisés sur des tâches de classification fine, comme par exemple :

* *Différencier des modèles de voitures*
* *Distinguer des variantes d’avions*
* *Reconnaître des espèces de fleurs*

En outre, **CLIP généralise mal aux images absentes de son jeu de pré-entraînement**. Enfin, les classificateurs zéro-shot de CLIP sont sensibles à la formulation des requêtes. Une **optimisation manuelle des prompts est parfois nécessaire** pour améliorer ses performances.

**4. Conclusion**

CLIP explore l’application du pré-entraînement sur du langage naturel à d’autres domaines que le NLP, notamment **la vision par ordinateur.**

Les résultats montrent que **CLIP apprend naturellement plusieurs tâches** lors de son pré-entraînement, à l’instar des modèles GPT, et que le transfert zéro-shot est une approche efficace.

**La modélisation**

**1. Introduction**

L’objectif de cette étude est de développer un modèle de classification d’images à l’aide de **réseaux de neurones convolutionnels (CNN)** pré-entraînés, en utilisant **VGG16** et **MobileNetV2**. Afin d’améliorer la robustesse du modèle, nous avons mis en place une stratégie de **data augmentation** et évalué les performances via plusieurs métriques.

**2. Méthodologie de modélisation**

**2.1 Prétraitement des données**

Les images ont été prétraitées pour être compatibles avec les architectures de **VGG16** et **MobileNetV2** :

* *Redimensionnement des images*
* *Conversion en tenseurs*
* *Encodage des classes*

Les données ont été divisées en trois ensembles stratifiés :

* *70 % pour l’entraînement*
* *15 % pour la validation*
* *15 % pour le test*

**3. Évaluation des modèles**

**3.1 Métrique d’évaluation**

Nous avons retenu la **précision (accuracy)** comme métrique principale, accompagnée de l’analyse de la loss. L’évaluation a été réalisée sur le jeu de test après entraînement, avec stockage des **meilleurs poids** selon une variable.

**3.2 Comparaison des performances**

Nous avons comparé les modèles avec et sans data augmentation, en observant :

* *L’accuracy en entraînement et en test*
* *L’évolution des courbes de loss pour détecter un éventuel sur-apprentissage*
* *L’analyse des erreurs en affichant des images mal classées*

**3.3 Démarche d’optimisation**

L’optimisation du modèle a suivi une approche itérative basée sur l’analyse des courbes d’apprentissage et des performances en test. Voici la démarche appliquée :

**3.3.1 Analyse initiale des performances**

* Observation des courbes de loss et d’accuracy pour détecter un éventuel surajustement
* Comparaison des performances entraînement vs validation pour évaluer la généralisation
* Identification des erreurs fréquentes en affichant les images mal classées

**3.3.2 Détection du surajustement**

* L’analyse des courbes a montré une divergence entre la loss d’entraînement et celle du test
* Le modèle atteignait une plus haute précision en entraînement mais une baisse en test
* Ces résultats montraient une mémorisation excessive des données

**3.3.3 Stratégies d’optimisation appliquées**

* Renforcement de la data augmentation avec des ajouts de transformations plus variés
* Augmentation du Dropout pour réduire le risque de surajustement
* Ajout de Régularisation L2 sur la couche dense pour limiter l’impact des poids trop élevés
* Diminution du Learning Rate pour stabiliser l’apprentissage
* Application d’un Early Stopping plus strict pour arrêter l’entraînement dès que la validation loss stagnait.

**Une synthèse des résultats**

**1. Résultats des modèles CNN classiques (VGG16 et MobileNetV2)**

Les modèles **VGG16** et **MobileNetV2** ont été évalués en utilisant une approche supervisée classique, avec un **pré-entraîne**ment sur **ImageNet** suivi d’un fine-tuning sur notre jeu de données. Nous avons également appliqué une **data augmentation** afin d’améliorer la robustesse et éviter le surajustement.

**Une image contenant texte, diagramme, ligne, Tracé

Le contenu généré par l’IA peut être incorrect.**

**Une image contenant texte, Police, logiciel, capture d’écran

Le contenu généré par l’IA peut être incorrect.**

On observe que l’ajout de data augmentation a globalement **réduit le surapprentissage** en rendant la validation et le test plus cohérents avec l’entraînement. Cependant, il ne permet pas toujours une nette amélioration de la précision en test.

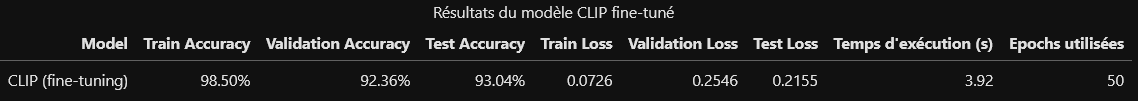
**2. Résultats de CLIP après fine-tuning**

Contrairement aux CNN classiques, **CLIP** est pré-entraîné sur une large base de données image-texte et peut être utilisé en **zero-shot** ou **fine-tuné** sur un jeu de données spécifique, ce qui va être le cas pour comparer nos résultats.

Après fine-tuning, les résultats obtenus sont les suivants :

Une image contenant texte, ligne, diagramme, Tracé

Le contenu généré par l’IA peut être incorrect.



Nous constatons que CLIP obtient une **précision bien plus élevée en test** (93,04%) que les modèles CNN traditionnels (~80% en moyenne). Cette performance s’explique par son **apprentissage multimodal**, qui lui permet de mieux comprendre le lien entre les images et leur description textuelle.

**3. Comparaison des performances et observations**

**Observations principales :**

* CLIP est **bien plus performant** que les CNN traditionnels, notamment grâce à son pré-entrainement sur de larges bases de données.
* Les CNN nécessitent un **grand volume de données annotées**, tandis que CLIP se base sur la correspondance image-texte.
* Le temps d’entraînement est **drastiquement réduit** avec CLIP, ce qui en fait une alternative très efficace pour une implémentation rapide.

**4. Conclusion et perspectives**

L’utilisation de **CLIP** dans ce projet démontre une approche plus rapide et performance pour la classification d’images par rapport aux modèles CNN classiques. Cependant, CLIP **reste dépendant de la qualité des descriptions textuelles fournies**, tandis que les modèles CNN sont plus robustes sur des tâches spécifiques après entraînement.

En perspectives, il peut être intéressant de tester d’autres variantes de CLIP (ViT-B/16 entre autres) ou encore améliorer l’optimisation des prompts textuels pour maximiser la performance.

**L’analyse de la feature importance globale et locale du nouveau modèle**

Contrairement à d’autres modèles, les CNN ne manipulent pas directement des features explicites comme « âge », « revenu » ou d’autres variables. Ces modèles travaillent sur des features extraites automatiquement sous forme de cartes de convolution. Ces features sont difficiles à interpréter individuellement.

Si on veut analyser l’importance locale, on peut utiliser entre autres une technique appelée « Grad-CAM). Elle permet de générer une zone de chaleur montrant quelles zones de l’image ont influencé la prédiction.

**Les limites et les améliorations possibles**

**1. Limites des modèles VGG16 et MobileNetV2**

* **Données annotées** : Ces modèles nécessitent un volume important d’images labellisées, ce qui peut être limitant avec un jeu de données modeste.
* **Fine-tuning rigide** : L’ajout de nouvelles catégories demande souvent un réentraînement complet.
* **Interprétabilité limitée** : Ils ne fournissent pas d’explications intrinsèques sur leurs décisions, d’où l’usage de techniques comme Grad-CAM

**2. Limites du modèle CLIP**

* **Sensibilité aux prompts** : La performance varie en fonction de la formulation des descriptions textuelles.
* **Distinctions fines** : Il peut avoir du mal à différencier des classes visuellement proches si le texte n’exprime pas suffisamment les nuances

**3. Pistes d’amélioration**

* **VGG16/MobileNetV2 :**
  + Tester différents optimisateurs et ajuster les hyperparamètres (learning rate, régularisation…).
  + Renformer la data augmentation pour limiter le surapprentissage.
* **CLIP :**
  + Optimiser la formulation des prompts.
  + Envisager des méthodes et modèles de CLIP pour comparer.