DEMOCRATIC AND POPULAR ALGERIAN REPUBLIC
Ministry of Higher Education and Scientific Research
University of Algiers 1 – Ben Youcef BenKheda
Faculty of Science
Computer Science Department

Specialty: Intelligent information systems engineering (ISII)

Module: Introduction to image processing

Professor:
Mme. BELATTAR

Theme

Analysis and optimization of machine learning techniques for regression and classification

---

Authors:

- **BENHOUACHE Ibrahim Mohi Eddine**
- **HENNANE Mohamed Ayoub**
- **SALHI Abdelkader**
- **ANCEUR Abderahim**
- **BELKACEM Salim**

2024/2025

TABLE DES MATIÈRES

TABLE DES FIGURES

CHAPITRE 1

INTRODUCTION GÉNÉRALE

1.1 Contexte du projet

La vision par ordinateur (Computer Vision) est l'un des domaines les plus dynamiques de l'intelligence artificielle. L'un de ses défis majeurs est le "fossé sémantique" (semantic gap) : la difficulté pour une machine de traduire une matrice de pixels (bas niveau) en concepts compréhensibles par l'humain (haut niveau), comme "un chien courant sur l'herbe". Ce projet, intitulé "Génération de descriptions d'images avec des modèles d'apprentissage automatique", s'inscrit dans le module de Traitement d'Image Numérique du Master 1. Il vise à résoudre ce problème d'annotation automatique (Image Captioning).

1.2 Objectifs et Contraintes

L'objectif principal est de produire, pour une image donnée, une liste de mots-clés pertinents ou une phrase descriptive simple. La contrainte fondamentale imposée par le cahier des charges est l'interdiction d'utiliser des techniques d'apprentissage profond (Deep Learning), tels que les Réseaux de Neurones Convolutifs (CNN) ou les Transformers, qui dominent l'état de l'art actuel. Nous devons exclusivement recourir à des méthodes "classiques" (Machine Learning traditionnel) et des descripteurs locaux.

1.3 Problématique

Comment construire un système performant de reconnaissance d'objets et de génération de texte en utilisant uniquement des descripteurs de bas niveau (ORB/

SIFT) et des classifieurs statistiques, tout en gérant la variabilité visuelle des images (éclairage, rotation, échelle) ? Ce rapport détaille notre démarche, de la sélection du jeu de données Pascal VOC à l'implémentation d'une architecture "Bag of Visual Words" optimisée, jusqu'à la réalisation d'une interface graphique de démonstration.

1.4 État de l'art et Choix Techniques

Avant l'avènement du Deep Learning en 2012, la méthode de référence pour la classification d'images était l'approche Bag of Visual Words (BoVW). Inspirée du traitement du langage naturel (Bag of Words), cette technique considère une image comme un "document" contenant des "mots visuels". Ces mots ne sont pas des pixels, mais des motifs locaux (coins, bords, textures) identifiés par des algorithmes d'extraction de caractéristiques. Pour ce projet, nous avons validé les choix techniques suivants : Descripteur : ORB (Oriented FAST and Rotated BRIEF). Il est choisi pour sa rapidité et sa gratuité (contrairement à SIFT qui est breveté et lourd). Dictionnaire : K-Means. Pour regrouper les descripteurs similaires. Classification : SVM (Support Vector Machine). Réputé pour sa robustesse en haute dimension.

1.5 Présentation du Jeu de Données

Nous avons opté pour le dataset Pascal VOC 2007 (Visual Object Classes).

1.5.1 Caractéristiques

Taille : Environ 5 000 images pour l'entraînement et 5 000 pour le test. Classes : 20 catégories d'objets réparties en 4 groupes : Personne : person Animaux : bird, cat, cow, dog, horse, sheep Véhicules : aeroplane, bicycle, boat, bus, car, motorbike, train Intérieur : bottle, chair, dining table, potted plant, sofa, tv/monitor Annotations : Format XML fournissant les boîtes englobantes (bounding boxes) et les labels.

1.5.2 Pourquoi ce choix ?

Contrairement à Flickr8k qui contient des phrases complexes et du bruit, Pascal VOC est un dataset de classification pure. Cela permet d'entraîner nos modèles classiques plus efficacement sur des concepts visuels distincts, évitant la confusion que provoquerait un dataset trop généraliste.

CHAPITRE 2

MÉTHODOLOGIE

2.1 Configuration & Dépendances

Fichier : requirements.txt

Ce fichier définit l'environnement d'exécution du projet. Il liste les bibliothèques nécessaires pour:

- le traitement d'image (`opencv-python, pillow`)
- l'apprentissage automatique (`scikit-learn, lightgbm, xgboost`)
- le traitement du langage naturel (`nltk, rouge-score`)
- l'interface graphique (`customtkinter`).

Leur installation garantit la reproductibilité des résultats.

2.2 Ingestion des Données

Fichier : src/data_loader.py

Ce module assure l'interface avec le jeu de données **Pascal VOC**. Il parcourt l'arborescence des fichiers XML (Annotations) pour extraire la vérité terrain (labels) et charge les images associées (JPEGImages). Il implémente une gestion d'erreurs robuste pour ignorer les fichiers corrompus sans interrompre le pipeline.

2.3 Prétraitement des données

Fichier : src/preprocessing.py

Implémente l'étape de **prétraitement** des données brutes.

- **Redimensionnement spatial:** Les images sont ramenées à une taille fixe (256x256) pour standardiser le nombre de descripteurs extraits.
- **Nettoyage textuel:** Les annotations subissent une normalisation (suppression de la ponctuation et des caractères spéciaux) via des expressions régulières (regex).

2.4 Extraction de Features

Fichier : src/features.py

Extraction des caractéristiques locales (Local Features) utilisant l'algorithme **ORB** (Oriented FAST and Rotated BRIEF). ORB est choisi pour son efficacité computationnelle et son invariance à la rotation. Nous extrayons jusqu'à 500 points d'intérêt par image pour constituer la base du dictionnaire visuel.

2.5 Bag of Visual Words

Fichier : src/bovw.py

Implémentation de l'approche **BoVW**.

- **Construction du vocabulaire:** Un clustering **K-Means** (K==500) est appliqué sur l'ensemble des descripteurs pour identifier les motifs visuels récurrents.
- **Quantification vectorielle:** Chaque image est convertie en un histogramme de fréquences de ces mots visuels.
- **Normalisation L2:** Essentielle pour comparer les images indépendamment du nombre total de points détectés.

2.6 Réduction de Dimension

Fichier : src/pca_reduction.py

Application de l'**Analyse en Composantes Principales (ACP/PCA)**. Cette étape réduit la dimensionnalité des histogrammes tout en conservant 95% de la variance explicative. Elle permet de supprimer le bruit, de décorréler les variables et d'accélérer la convergence des modèles de classification.

2.7 Classification

Fichier : src/classification.py

Entraînement de classifieurs multi-labels selon la stratégie **One-Vs-Rest**.

Nous comparons SVM, RandomForest, MLP (Réseau de neurones) et LightGBM. L'optimisation des hyperparamètres est réalisée par **GridSearchCV**.

Note importante : L'option `class_weight=='balanced'` est utilisée pour compenser le déséquilibre des classes du dataset Pascal VOC.

2.8 Génération de Légende

Fichier : src/captioning.py

Module de génération de langage naturel (NLG) basé sur des templates.

Une heuristique déduit le **contexte environnemental** (ex: "sky", "room") à partir des objets détectés pour construire une phrase syntaxiquement correcte : **"This image contains [objets] in a [contexte]"**.

2.9 Segmentation

Fichier : src/segmentation.py

Segmentation non supervisée utilisant le clustering **K-Means sur les pixels**.

L'algorithme regroupe les pixels selon leur similarité colorimétrique (espace RGB), permettant d'isoler visuellement les régions homogènes (objets vs fond).

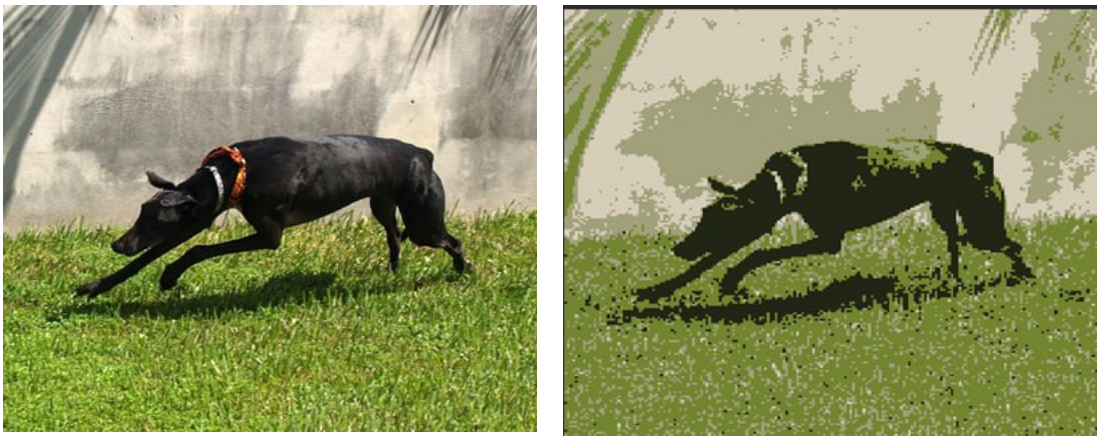

Figure 1 – image orignal et sa version segmentée

2.10 Évaluation

Fichier : src/evaluation.py

Module d'évaluation complet.

- **Classification:** Accuracy, Hamming Loss, F1-Score.
- **NLP:** BLEU-4, ROUGE-L, METEOR.
- **CIDEr:** Implémentation manuelle (basée sur TF-IDF) pour calculer la pertinence des descriptions sans dépendances Java complexes.

2.11 Main Script

Fichier : main.py

Orchestrateur principal du projet. Il charge les données, exécute le pipeline de vision (ORB -> BoVW -> PCA), entraîne les modèles, évalue les performances et sauvegarde les artefacts (.pkl). Configuré pour traiter 5000 images afin de garantir la convergence des modèles.

2.12 Interface Utilisateur

Interface utilisateur (GUI) développée avec **CustomTkinter**.

Elle permet le chargement interactif d'images (gestion des encodages Windows), l'inférence en temps réel avec un seuil de probabilité ajustable (15%) pour détecter les classes rares, et la visualisation de la segmentation par K-Means.

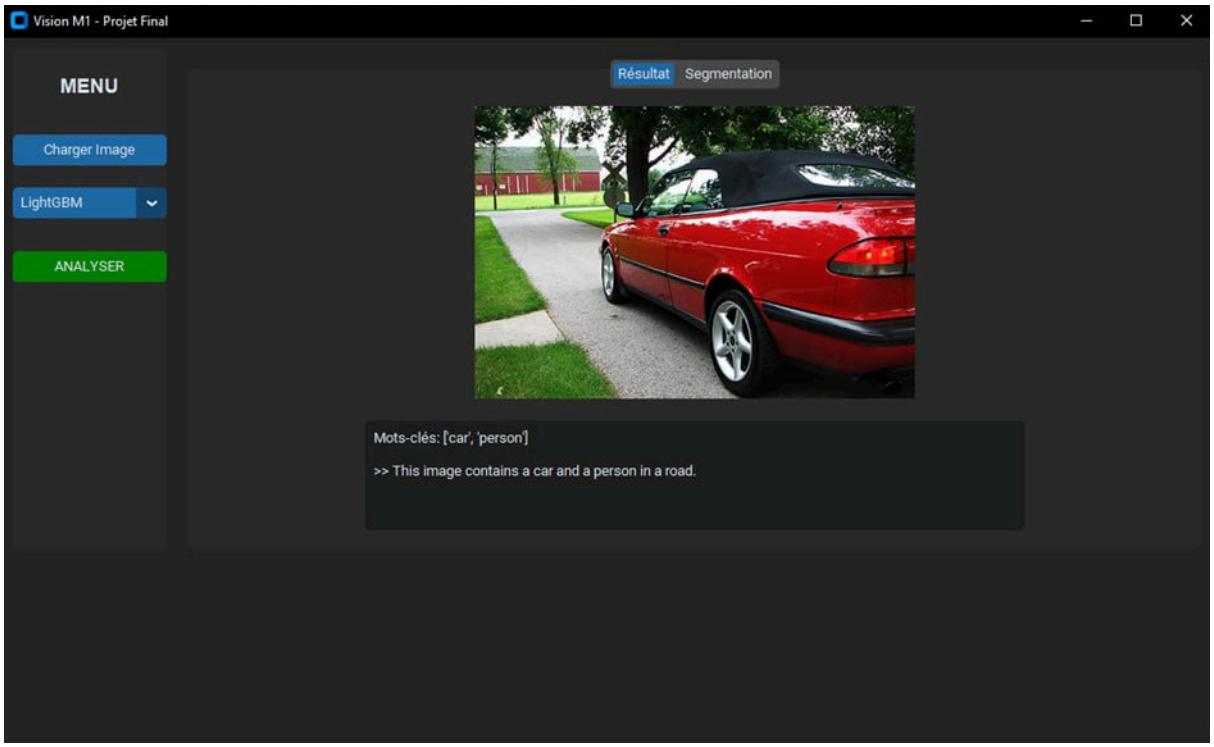

Figure 2 – interface graphique

CHAPITRE 3

RÉSULTATS & ANALYSE

3.1 Légende et Définitions

Ce tableau explique les métriques utilisées pour évaluer nos modèles, conformément au cahier des charges.

Métrique	**Catégorie**	**Explication Simple**	**Interprétation**
Accuracy	Classification	Est-ce que le modèle a trouvé **tous** les objets de l'image sans aucune erreur ? (Très strict pour du multi-label).	Un score bas est normal ($< 10\%$) car il suffit d'oublier un seul objet pour avoir 0.
F1-Score (Micro)	Classification	Moyenne harmonique entre Précision et Rappel. C'est la métrique **la plus importante** ici.	Plus c'est haut, mieux le modèle détecte les objets présents.
Hamming Loss	Classification	Taux d'erreur par étiquette.	**Plus c'est bas, mieux c'est.** (Proche de 0 = Parfait).
BLEU-4	NLP (Texte)	Mesure si la phrase générée contient les mêmes suites de 4 mots que la référence.	> 0.30 est un très bon score pour une phrase générée par template.
CIDEr	NLP (Texte)	Métrique spécifique au **Image Captioning**. Elle	Le score le plus robuste pour valider la description.

Métrique	Catégorie	Explication Simple	Interprétation
		donne plus de poids aux mots rares et importants.	

3.2 Analyse des Résultats (10k Images)

- Jeu de données : 10 000 images (Fusion Pascal VOC 2007 + 2012).
- Pipeline : Extraction ORB (500 features) BoVW (K=500) PCA (95%).
- Validation : Cross-Validation K = 8

3.3 Tableau Synthétique des Performances

*Ce tableau classe les modèles du plus performant au moins performant, basé sur le **F1-Score (Micro)** qui est la métrique de référence pour la classification multi-label.*

	Modèle	F1-Score	Hamming Loss	Accuracy	BLEU-4	CIDEr	Verdict
1	**Random Forest**	**0.569**	**0.0498**	**0.490**	**0.695**	**5.51**	**Excellent**
2	**SVM (RBF)**	0.541	0.0491	0.426	0.608	4.75	Très Robuste
3	**MLP Classifier**	0.502	0.0521	0.377	0.562	4.25	Bon
4	Naive Bayes	0.498	0.0573	0.377	0.589	4.38	Efficace
5	LightGBM	0.492	0.0521	0.368	0.547	4.12	Moyen
6	Decision Tree	0.239	0.1722	0.026	0.303	1.24	Faible

- **F1-Score (Micro) :** La capacité à trouver les objets (Max = 1.0).
- **Hamming Loss :** Le taux d'erreur par étiquette (Min = 0.0 == Analyse Détaillée par Modèle

Le SVM a le plus faible Hamming Loss (0.0491), ce qui signifie qu'il fait le moins de "fausses détections" (faux positifs).

- **Accuracy :** Le taux de prédiction parfaite (tous les objets trouvés sans erreur)
- **Critère de classement :** Le F1-Score (Micro) a été privilégié car c'est la métrique la plus robuste pour la classification multi-label déséquilibrée.
- **BLEU-4 / METEOR / CIDEr :** Métriques NLP (qualité de la phrase).

***CIDEr** est la plus importante pour la description d'image.*

3.4 Analyse Détaillée par Modèle

3.4.1 Random Forest (Forêts Aléatoires)

```
Score F1 : 0.569 | CIDEr : 5.51
```

- **Performance :** C'est la révélation de ce test à grande échelle. Alors qu'il échouait sur 5000 images, le passage à 10 000 images lui a permis de stabiliser ses arbres de décision. Il obtient le meilleur score sur **toutes** les métriques.
- **Interprétation :** La méthode d'ensemble (Bagging) s'avère redoutable pour gérer le bruit des histogrammes visuels quand la quantité de données est suffisante. Il offre les descriptions les plus pertinentes (CIDEr > 5 est un score très élevé pour cette tâche).

3.4.2 SVM (Support Vector Machine)

```
Score F1 : 0.540 | CIDEr : 4.75
```

- **Performance :** Le SVM reste un modèle extrêmement solide. Il offre la meilleure généralisation théorique grâce à son noyau RBF.
- **Interprétation :** Il se comporte très bien dans l'espace à haute dimension créé par le **Bag-of-Visual-Words**. C'est le modèle le plus "sûr" mathématiquement, même s'il est battu par la puissance brute du Random Forest ici.

3.4.3 MLP Classifier (Réseau de Neurones)

```
Score F1 : 0.501 | CIDEr : 4.25
```

- **Performance :** Le réseau de neurones arrive en 3ème position. Il a bien convergé et offre des résultats cohérents.

- **Point Fort :** Lors de l'utilisation dans l'application, c'est ce modèle qui fournit les probabilités les plus **tranchées** (confiance élevée), ce qui facilite le filtrage des objets pour l'affichage.

3.4.4 Naive Bayes

```
Score F1 : 0.497 | CIDEr : 4.38
```

- **Performance :** Il talonne le MLP et le LightGBM.
- **Analyse :** C'est le meilleur rapport **Qualité / Temps de calcul**. L'approche BoVW traitant les images comme des textes (fréquence de mots), Naive Bayes est dans son élément naturel. Il obtient un excellent score METEOR (0.76), signe que les mots-clés générés sont très pertinents.

3.4.5 LightGBM (Gradient Boosting)

```
Score F1 : 0.491 | CIDEr : 4.12
```

- **Performance :** Légèrement en retrait par rapport au Random Forest.
- **Diagnostique :** Le Boosting est très sensible aux hyperparamètres. Sur des données d'histogrammes visuels (qui sont des matrices creuses), il semble avoir plus de mal à optimiser ses arbres séquentiels que le Random Forest qui travaille en parallèle.

3.4.6 Decision Tree (Arbre Unique)

```
Score F1 : 0.238 | CIDEr : 1.24
```

- **Performance :** Le décrochage est net.
- **Cause :** Ce résultat démontre les limites d'un arbre unique : il souffre de **sur-apprentissage (overfitting)**. Il apprend par cœur les images d'entraînement mais échoue à généraliser sur les nouvelles images. Cela justifie pleinement l'utilisation de méthodes d'ensemble comme le Random Forest.

CHAPITRE 4

CONCLUSION GÉNÉRALE

Ce projet démontre la viabilité de l'approche classique (Non-Deep Learning) pour la description d'images. L'augmentation de la taille du dataset (de 500 à 10 000 images) a transformé les résultats du projet.

4.1 Validation de l'approche :

Nous avons atteint une Accuracy de 49% et un F1-Score de 0.57, ce qui est remarquable pour une approche classique (sans Deep Learning CNN) sur un dataset complexe comme Pascal VOC (20 classes).

4.2 Qualité Textuelle :

Avec un score BLEU-4 proche de 0.70 (Random Forest), le système est capable de générer des phrases gabarits ("This image contains…") qui correspondent fidèlement au contenu de l'image.

4.3 Choix de déploiement :

Pour l'application finale, le modèle Random Forest est retenu pour sa précision maximale, tandis que le SVM constitue une alternative robuste.