

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. MEZZOUDJ Saliha

Theme

Analysis and optimization of machine learning
techniques for regression and classification

Authors:

- Ibrahim Zoj
- Abdelkabder secteur
- Ayoub Casquita
- BELKACEM salim
- Abderahmane casquita

TABLE DES MATIÈRES

1	Méthodologie	1
1.1	Configuration & Dépendances	1
1.2	Ingestion des Données	1
1.3	Prétraitement des données	1
1.4	Extraction de Features	2
1.5	Bag of Visual Words	2
1.6	Réduction de Dimension	2
1.7	Classification	3
1.8	Génération de Légende	3
1.9	Segmentation	3
1.10	Évaluation	3
1.11	Main Script	4
1.12	Interface Utilisateur	4
2	Résultats & Analyse	5
2.1	Légende et Définitions	5
2.2	Tableau Comparatif des Performances	6
2.3	Analyse Détaillée par Modèle	6
2.3.1	SVM (Support Vector Machine)	7
2.3.2	MLP Classifier (Réseau de Neurones)	7
2.3.3	Naive Bayes	7
2.3.4	LightGBM	7
2.3.5	Decision Tree	8
2.3.6	Random Forest	8
3	Conclusion Générale	9

CHAPITRE 1

MÉTHODOLOGIE

1.1 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.

1.2 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.

1.3 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).

1.4 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.

1.5 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.

1.6 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 décorrélérer les variables et d'accélérer la convergence des modèles de classification.

1.7 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.

1.8 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]**”.

1.9 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).

1.10 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.

1.11 Main Script

Fichier : main.py

Orchestrator 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.

1.12 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.

CHAPITRE 2

RÉSULTATS & ANALYSE

2.1 Légende et Définitions

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

Métrique	MCatégorie	MExplanation Simple	MInterprétation
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.

Métrique	MCatégorie	MExplication Simple	MInterprétation
CIDEr	NLP (Texte)	Métrique spécifique au Image Captioning . Elle donne plus de poids aux mots rares et importants.	Le score le plus robuste pour valider la description.

2.2 Tableau Comparatif des Performances

Résultats obtenus après entraînement sur 5000 images du dataset Pascal VOC 2007 (BoVW K=500).

Modèle	Accuracy	F1-Score (Micro)	BLEU-4	CIDEr	Observation
SVM (RBF)	0.081	0.275	0.329	1.69	Meilleur Modèle global
MLP Classifier	0.061	0.233	0.303	1.42	Très confiant, excellent pour la démo
Naive Bayes	0.059	0.231	0.320	1.46	Rapide et surprenant
LightGBM	0.056	0.226	0.269	1.20	Correct mais moins performant ici
Decision Tree	0.008	0.191	0.301	1.11	Moyen, manque de généralisation
Random Forest	0.011	0.048	0.142	0.23	Échec (Biais sur la classe “Person”)

2.3 Analyse Détailée par Modèle

Critère de classement : Le F1-Score (Micro) a été privilégié car c'est la métrique la plus robuste pour la classification multi-label déséquilibrée.

2.3.1 SVM (Support Vector Machine)

Score F1 : 0.2748 | CIDEr : 1.6911

- **Analyse:** Le grand gagnant. Avec son noyau RBF (non-linéaire) et la gestion des poids (class_weight='balanced'), il a su tracer les meilleures frontières entre les classes dans l'espace complexe des histogrammes visuels.
- **Verdict:** C'est le modèle à mettre en avant dans le rapport comme la référence technique.

2.3.2 MLP Classifier (Réseau de Neurones)

Score F1 : 0.2335 | CIDEr : 1.4242

- **Analyse :** Il arrive en deuxième position, très proche du sommet. Les réseaux de neurones (même simples comme ce Perceptron Multi-Couches) sont excellents pour capter les corrélations non-linéaires entre les mots visuels.
- **Point fort :** Lors des tests en temps réel, c'est lui qui affiche les **probabilités les plus tranchées** (confiance élevée), ce qui le rend idéal pour la démonstration.

2.3.3 Naive Bayes

Score F1 : 0.2307 | CIDEr : 1.4648

- **Analyse :** La “surprise” du projet. Il talonne le réseau de neurones de très près.
- **Explication :** Notre technique (Bag-of-Visual-Words) transforme les images en “documents de mots”. Or, Naive Bayes est mathématiquement conçu pour l’analyse de texte (fréquence des mots). Il est donc naturellement très à l’aise sur ce type de données, malgré sa simplicité apparente.

2.3.4 LightGBM

Score F1 : 0.2266 | CIDEr : 1.2058

- **Analyse :** Performance honorable, mais légèrement en retrait par rapport au top 3.
- **Pourquoi ?** Bien que très puissant sur des données tabulaires (Excel), le Gradient Boosting a eu un peu plus de mal à généraliser sur les histogrammes

BoVW réduits par PCA comparé à la géométrie du SVM. Il reste cependant un modèle valide pour ce projet.

2.3.5 Decision Tree

Score F1 : 0.1912 | CIDEr : 1.1125

- **Analyse :** Un décrochage net dans les performances.
- **Problème :** Un arbre de décision unique est trop “rigide”. Il a tendance à apprendre par cœur le bruit des données d’entraînement (overfitting) et échoue à reconnaître des objets qu’il n’a pas vus exactement sous le même angle.

2.3.6 Random Forest

Score F1 : 0.0485 | CIDEr : 0.2303

- **Analyse :** L’échec du comparatif.
- **Cause :** Le modèle n’a pas réussi à gérer le fort déséquilibre des classes (beaucoup de “Person”, peu d’“Avion”). Il est tombé dans un piège statistique : pour minimiser son erreur, il a décidé de prédire “Personne” ou “Rien” la plupart du temps. C’est un biais d’apprentissage majeur.

CHAPITRE 3

CONCLUSION GÉNÉRALE

Ce projet démontre la viabilité de l'approche classique (Non-Deep Learning) pour la description d'images.

1. **L'impact des Données :** Passer de 500 à 5000 images a été décisif pour obtenir des résultats cohérents.
2. **Limites du BoVW :** L'approche par “sac de mots visuels” perd l'information spatiale (forme globale). Cela explique les confusions fréquentes entre objets ayant des textures similaires (ex: poils de chat vs texture de canapé).
3. **Choix Final :** Pour une application réelle basée sur cette architecture, nous retiendrions le couple **SVM** (pour la précision) ou **MLP** (pour la confiance).