

# Détection et comptage d'objets par IA

Dans de nombreux domaines industriels et scientifiques, le comptage automatique d'objets dans des images représente un enjeu majeur. Que ce soit pour l'inventaire en temps réel dans un entrepôt, la détection des réapprovisionnements dans les rayons de supermarchés, l'analyse de données biologiques, le comptage de personnes dans un lieu donné, ou encore le contrôle qualité en production, la capacité à détecter et dénombrer automatiquement différents types d'objets est devenue un enjeu important.

L'usage de l'IA s'est là aussi imposé : associée au *computer vision*, elle peut regrouper les objets semblables (apprentissage non supervisé), identifier des objets précis (classification), localiser des objets connus (détection), suivre le mouvement d'objets dans un lot d'images (tracking). Et bien sûr on peut combiner : détection et classification sont souvent réalisées ensembles, ou détection puis tracking.

## 1 Contexte

On s'intéresse ici à une tâche générique : des objets connus (par exemple la photo d'une lettre d'un texte, ou la photo d'un symbole sur un plan) doivent être retrouvés au sein d'une image. Dans la version la plus simple, l'image recherchée et les exemplaires à retrouver sont strictement identiques. Cela s'appelle du "spotting". Les images à retrouver peuvent être orientées différemment (rotation de 90, 180 ou 270 degrés). On considèrera cependant que les dimensions sont inchangées (l'image à rechercher a les mêmes dimensions dans l'images où rechercher). Une fois localisés, il faut identifier les objets et être capable d'afficher, pour chaque type d'objet, le nombre trouvé, et les endroits. On considèrera qu'un objet à retrouver est dans un fichier image dont le nom identifie la famille de l'objet (par exemple, une lettre "a" sera dans un fichier "a.png").

## 2 Objectif du Projet

Le projet vise à développer un système de vision par ordinateur capable de :

1. Détecter des objets dans une image à partir d'un catalogue de référence (le catalogue sera un lot d'images dans un dossier)
2. Classifier ces objets selon leur famille d'appartenance (il faut associer à chaque objet trouvé une des images du catalogue)
3. Compter le nombre d'occurrences de chaque type d'objet
4. Fournir une visualisation claire des résultats (objets localisés matérialisés sur l'image à traiter, nombre d'objets / classe en console).

## 3 Données Disponibles

### Pour le développement

- Des images à compter, contenant des objets, réparties par famille dans 3 sous-dossiers : texte, plan,
- Des dossiers "catalogue" contenant, pour chaque famille, des objets types à localiser et compter

## Données de test

- Quelques images de test par famille, sont utilisées pour l'évaluation

## 4 Travail à Réaliser

### 4.1 Phase 1 : recherche bibliographique

- État de l'art des techniques de détection d'objets
- Comparaison des différentes approches : classiques vs deep learning
- Analyse des avantages et limitations de chaque approche

### 4.2 Phase 2 : conception

On impose un système simple : localisation d'objets par détection des composantes connexes, classification au moyen d'un réseau Bayésien d'une part et d'un k-ppv d'autre part.

- Expliquez le principe de fonctionnement
- Design de l'architecture du système : quels sont les éléments, et les données échangées
- Définition des métriques d'évaluation : comment évaluer la performance ?
- Planification des expérimentations : définir un protocole de tests

### 4.3 Phase 3 : implémentation

Vous veillerez à la modularité du code, notamment pour faciliter l'évaluation : programmes sous la forme d'une API d'un côté, scripts utilisant cette API de l'autre pour tester/évaluer, et développer un outil simple de comptage de symboles dans un plan ou de caractères dans une image de texte.

- Développement du pipeline de traitement
- Implémentation de la solution choisie
- Mise en place des tests
- Documentation du code

### 4.4 Phase 4 : évaluation et optimisation

- Tests sur différents scénarios
- Analyse des performances actuelles (précision obtenue, temps d'exécution en rapport avec la taille de l'image traitée, le volume du catalogue, la taille moyenne des objets du catalogue...)
- Optimisation des paramètres (au sens large : programme, environnement de compilation, algorithmes, paramétrage des algorithmes...)
- Proposition d'améliorations (si d'autres pistes vous semblent intéressantes mais que vous ne les avez pas exploitées)

## 5 Livrables Attendus

### 1. Rapport technique

- La synthèse bibliographique
- La justification des choix techniques
- La description de l'implémentation
- L'analyse des résultats

## 2. Code source documenté

- Pipeline complet de traitement
- Scripts d'évaluation
- Instructions d'installation et d'utilisation

## 6 Critères d'évaluation

Le projet comporte une partie théorique (IA) et une partie pratique (implémentation, évaluation, en Python). Il y aura donc deux évaluations séparées, une pour chaque matière.

- IA : Qualité de la recherche bibliographique
- IA : Conception du système et de l'API
- Python : Qualité du code et de la documentation
- Python : Evaluation et réglage de la performance du système

## 7 Conseils méthodologiques

- Commencez par une approche simple puis complexifiez progressivement
- Testez régulièrement votre système sur différents cas d'usage
- Documentez vos choix et vos expérimentations
- Prévoyez du temps pour l'optimisation des performances

## 8 Contraintes techniques

- Langage de programmation : Python
- Bibliothèques autorisées : OpenCV, TensorFlow/PyTorch, NumPy, Pandas
- Format des images : PNG, JPG

**Organisation** : le projet devra être réalisé en groupe de 2-3 étudiants sur une durée de 8 semaines. **Rendu attendu pour le 8 janvier 2025 à 12h00**