

Plateforme intelligente de suivi des poubelles par image pour la prévention des dépôts sauvages

- Solution Delivery pour master Camp Data -

Contexte :

Face au manque de données précises sur les déchets dits abandonnés et à l'urgence d'agir pour limiter leur prolifération, le projet Wild Dump Prevention (WDP) propose une approche innovante visant à dresser un état des lieux aussi exhaustif que possible de la problématique des déchets sauvages. S'appuyant sur la démarche AI for Good, WDP vise non seulement à cartographier les dépôts existants, mais surtout à anticiper l'apparition de nouveaux sites de dépôt, en se concentrant notamment sur les zones où les poubelles débordent fréquemment, car elles deviennent souvent des points de départ de dépôts sauvages.

Ainsi, dans un contexte où le volet préventif se limite encore trop souvent à des actions de sensibilisation ou de formation, l'IA peut offrir une capacité de prédiction accrue, permettant une meilleure anticipation des risques de débordement et contribuant à la réduction des déchets abandonnés dans l'espace public.

Problème

Le manque de suivi en temps réel de l'état des infrastructures de collecte (poubelles, conteneurs) entraîne souvent une réaction tardive de la part des services municipaux ou des prestataires privés responsables. Cela favorise l'apparition de comportements inciviques.

Lorsque les débordements ne sont pas détectés rapidement, les déchets s'accumulent dans l'espace public, dégradent le cadre de vie et peuvent rapidement se transformer en dépôts sauvages difficiles à maîtriser. De plus, la diversité des lieux et des contextes rend la planification des tournées de ramassage plus complexe.

Objectif

Développer une plateforme intelligente de détection de l'état des poubelles publiques (pleines ou débordantes, vides) à partir d'images collectées sur le terrain pour améliorer la gestion des déchets urbains et la prévention des dépôts sauvages

Compétences

Une équipe mixte, esprit innovant et novateur, une capacité d'analyse critique à dominante SI, et d'une bonne maîtrise de l'analyse de données.

Changements

Aujourd'hui, très peu d'initiatives permettent une surveillance proactive des dispositifs de collecte. La maintenance des bacs est souvent déclenchée trop tard, une fois le problème devenu visible dans l'espace public. Ce projet propose une alternative numérique simple, peu coûteuse et plus efficace, capable d'améliorer la performance du service public de gestion des déchets en s'appuyant sur des données de terrain. Il permettra de prédire avec précision les zones à risque et d'anticiper les périodes les plus propices à l'apparition de dépôts.

Sorties

Une plateforme web capable de :

- Collecter des images de poubelles via upload citoyen, agent, caméra embarquée
- Détecter automatiquement leur état (pleine, vide) à partir de caractéristiques visuelles simples
- Enrichir les données collectées avec des métadonnées (localisation, date, caractéristiques d'image)
- Cartographier dynamiquement les zones à risque de débordement

Impact

- Réduction de l'empreinte écologique de l'homme et le risque des dépôts sauvages par une action préventive efficace et une meilleure gestion des points de collecte officiels

Structure du projet :

D'un point de vue technique, ce projet intègre l'IA afin de mettre en place une plateforme web capable de :

- Collecter des images de poubelles (via upload et stockage)
- Proposer une interface d'annotation manuelle (pleine / vide)
- Extraire automatiquement des caractéristiques simples (dimensions, taille du fichier, couleur moyenne, histogrammes, contraste, contours, etc.)
- Mettre en place des règles conditionnelles pour simuler une classification (sans utiliser de modèles de machine learning)
- Développer un tableau de bord avec des statistiques et des graphiques (localisation, date, caractéristiques d'image) et cartographier dynamiquement les zones à risque de débordement
- Tester la plateforme dans un contexte réel et rédiger une documentation technique

1. Plateforme Web (Flask ou Django)

La plateforme web constitue le cœur du projet, accessible via un navigateur, et permet aux utilisateurs de gérer les images et les annotations facilement. Les principales fonctionnalités attendues sont :

- **Formulaire d'upload d'image :**
 - L'utilisateur peut sélectionner et envoyer une image depuis son ordinateur.
 - Vérifications basiques (formats acceptés : JPG, PNG...).
 - Stockage physique de l'image sur le serveur (dans un répertoire dédié).
 - Les données sont accessibles via ce lien [[Lien](#)]
- **Affichage de l'image à l'écran :**
 - Après upload, l'image est affichée directement sur l'interface.
 - Des boutons d'annotation sont proposés :
 - Pleine
 - Vide
 - Chaque clic enregistre l'annotation dans la base de données.
- **Base de données :**
 - Chaque image est liée à :
 - Son chemin de stockage,
 - La date d'ajout,
 - Son annotation (pleine/vide),
 - Les métadonnées extraites (taille, dimensions, couleur moyenne, ...).

2. Extraction de caractéristiques (feature extraction)

Pour chaque image ajoutée, un script Python récupère automatiquement des caractéristiques simples permettant d'alimenter la base et de préparer l'analyse future :

- **Taille du fichier :**
 - Exprimée en Ko/Mo, via la fonction `os.path.getsize()`.
- **Dimensions :**
 - Hauteur et largeur (en pixels), extraites avec Pillow (PIL).
- **Couleur moyenne :**
 - Calcul de la moyenne des valeurs RVB (Rouge, Vert, Bleu) sur toute l'image,
 - Exemple : moyenne rouge = 102, vert = 98, bleu = 105,
 - Permet de mesurer la "luminosité" ou "saturation" globale d'une image.
- **Histogramme des couleurs** (distribution des niveaux de gris ou RVB),
- **Niveau de contraste** (différence max-min des pixels),
- **Détection de contours** (avec OpenCV ou Pillow : Sobel/Canny simple),
- **Histogramme de luminance**
- ...

Ces informations sont :

- Stockées automatiquement dans la base,
- Affichées dans l'interface utilisateur après upload.

3. Mécanisme de classification simple

Ici, sans machine learning, on simule un processus de décision automatique à partir des caractéristiques extraites.

- **Objectif :**
 - Proposer un algorithme basé sur des règles conditionnelles, faciles à modifier.
- **Exemple de règle :**
 - *Si la couleur moyenne < seuil sombre (ex. 100) ET la taille du fichier > X Mo → classer l'image comme pleine automatiquement.*
- **Fonctionnalité :**
 - Les règles peuvent être :
 - Codées from scratch pour démarrer,
 - Ou configurables via l'interface.
- Le système doit :
 - Exécuter la règle automatiquement après upload,
 - Marquer l'image avec un label automatique (visible dans la base + interface).


4. Tableau de bord (visualisation & suivi)

Pour offrir une vue globale du projet et des données collectées, un dashboard interactif est intégré.


- **Indicateurs clés :**
 - Nombre total d'images uploadées,
 - Répartition des annotations : % pleines vs % vides,
 - Distribution des tailles des fichiers,
 - Localisation, date d'acquisition des images afin de cartographier dynamiquement les zones à risque de débordement
- **Visualisation :**
 - Côté back-end (Python) :
 - Utilisation de matplotlib pour générer des graphes statiques (en PNG) à afficher.
 - Côté front-end (JavaScript) :
 - Utilisation de chart.js pour produire des graphes dynamiques directement dans la page web.

Évaluation des risques et démarche Green IT

Afin de garantir la robustesse et l'innovation du projet, l'intégration d'une évaluation des risques ainsi que l'adoption d'une démarche Green IT / d'éco-conception s'imposent aujourd'hui non seulement comme une nécessité, mais également comme un véritable atout différenciant pour vos futurs profils d'ingénieurs du numérique. L'évaluation des risques permettra d'identifier et de documenter les principaux dangers liés à la réalisation technique et à l'usage de la plateforme, tandis que l'approche Green IT visera à limiter l'impact environnemental et la consommation d'énergie, tout en favorisant l'inclusion sociale et la sensibilisation à la problématique des déchets sauvages. Nous vous mettons à dispositions deux documents :

1.  [Questionnaire d'écoconception de services numériques](#) (Doc Word - – approche macro entreprise sous forme de questions thématiques (Stratégie, Spécifications,

Architecture, UX/UI, Contenus, Frontend, Backend, hébergement, Algorithmie) dont certaines questions sont récurrentes pour tous les projets.

2. le  [tableau de synthèse de l'évaluation qualitative et quantitative de l'impact de votre projet](#) permet d'aborder en deux thèmes généraux n°1 (**Software** : design - coding - hébergement, stockage, transport de données) n°2 (**User device** (PC, tablet, smart phone,...) et/ou éléments numériques d'un produit).

Ces deux approches se complètent et se retrouveront dans le cahier des charges.

Technos à utiliser :

- **Back-end** : Python (Flask, Django, etc.),
- **Gestion des images** : Pillow, os, shutil,...
- **Base de données** : SQLite / PostgreSQL,...
- **Front-end** : HTML/CSS + Bootstrap (ou autre), Chart.js (pour les graphes dynamiques),
- **Visualisation** : matplotlib (Python) ou Chart.js (web).

Livrables :

- Code complet de la web app (back + front),
- Base de données structurée avec les images et annotations
- Rapport technique expliquant :
 - la structure du projet,
 - le fonctionnement des extractions de caractéristiques,
 - la logique des règles de classification,
 - des captures d'écran montrant les principales fonctionnalités,
 - une évaluation des risques
 - une justification de démarche Green IT

Niveaux de complexité

Niveau 1-basique (Must)

Compétences attendues :

- Mise en place d'une plateforme web simple (upload, affichage image, annotation), utilisation d'outils existants pour annotation comme par exemple Scalabel
- Détermination des caractéristiques de base (taille, dimensions, couleur) et leur stockage dans une base de données
- Définir des règles conditionnelles de classification codées en dur (directement intégrées dans le code).
- Visualisation des statistiques basiques via matplotlib ou graphes statiques.

Niveau 2 –Intermédiaire (Should)

Compétences attendues :

- Développement complet de l'interface d'annotation UX : navigation entre les images, raccourcis clavier, affichage des métadonnées au survol des annotations, etc
- Extension des caractéristiques extraites avancées comme (histogrammes, contraste, contours, etc.).
- Définir des règles de classification configurables via l'interface (utilisateur peut définir ses propres règles) avec une sauvegarde dynamique dans la base.
- Tableau de bord interactif avec graphes dynamiques (Chart.js) et filtres.

Niveau 3 –Avancé (could have)

Compétences attendues :

- Intégration de module de vérification de la conformité des données stockées dans une base
- Développement d'un tableau de bord avancé avec des indicateurs en temps réel, via des technologies telles que WebSocket ou AJAX. Ces indicateurs incluent notamment la localisation, la population, les jours de marché, les déclarations de travaux BTP, la météo, le jour de la semaine et la date d'acquisition des images, afin de cartographier dynamiquement les zones à risque de débordement.
- Optimisation de performance (compression image, gestion mémoire interne, gestion asynchrone de l'upload et de l'extraction de features pour ne pas bloquer l'interface, pagination pour les listes d'images, optimisation des requêtes BDD, etc.).
- Optimisation des performances : compression des images, gestion de la mémoire interne, gestion asynchrone de l'upload et de l'extraction des caractéristiques pour ne pas bloquer l'interface, pagination des listes d'images, optimisation des requêtes vers la BDD, etc
- Version multilingue de la plateforme.

Trame d'évaluation technique

- Rappel de l'appel d'offre (2 points)
 - Contexte : Décrire le contexte général du projet.
 - Problématique : Définir la problématique spécifique à résoudre.
- Méthodologie (4 points)
 - Conception de la chaîne : Acquisition → annotation → stockage → traitement des données.
 - Pertinences des caractéristiques et des règles de classification choisies
- Implémentation et expérimentation (5 points)
 - Bibliothèques et Framework utilisés (justification des choix techniques)
 - Détails des phases d'entraînement et de validation : Explication et interprétation de ces phases.
 - Fonctionnalités implémentées
 - Utilisation des règles configurables
 - UX du dashboard implémenté
 - Modularité / maintenabilité du code
- Résultats (5 points)
 - Résultats de la classification (Performance des modèles sur le jeu de données) : Taux de classification correcte avec des métriques adaptées à la classification (Accuracy, Precision, Recall,...)
 - Résultats de l'explicabilité (Evaluation des techniques d'explicabilité fournies par les modèles avec des exemples concrets)
 - Cartographie dynamiquement les zones à risque de débordement
 - Performance de la plateforme (temps de réponse)
- Démo (2 points)
 - Démonstration des différentes fonctionnalités du projet.
- Appréciation des Experts (2 points) : Innovation et Créativité, Impact Potentiel, Qualité de la Documentation.

Timeline 10 min de présentation, 5 min Demo Et 5 min de Q&R