Développer une interface OCR

Présenté par Steve Dos Santos
le 31 Mars 2025

Contexte et Objectif

- Automatiser le traitement de ~6000 factures via OCR.
- Extraire les informations clés et les stocker en base de données.
- Interface web simple et efficace pour la visualisation.

Fonctionnalités Principales

- Extraction des données (numéro de facture, date, nom du client, etc.).
- Stockage des informations en base de données.
- Interface web pour visualiser et rechercher des factures ou en ajouter.
- Tableau de bord de monitoring (statuts, logs, temps de traitement).

Technologies Utilisées

- OCR : Tesseract + QReader pour extraire les informations.
- Base de données : PostgreSQL pour le stockage des informations.
- Back-end : FastAPI pour l'API et la gestion des requêtes.
- Front-end : Jinja2, HTML/CSS pour l'affichage.
- Déploiement : Docker pour conteneurisation.

Présentation de l’OCR

- Définition : Reconnaissance optique de caractères.
- Fonctionnement : Extraction, analyse, post-traitement.
- Défis : Images de qualité pour éviter les erreurs de reconnaissance.

Challenges et Solutions

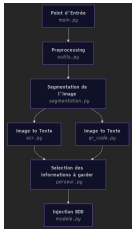
- Optimisation du preprocessing pour améliorer la reconnaissance.
- Gestion des erreurs et suivi dans le tableau de monitoring.
- Sécurisation : authentification, restriction des accès.

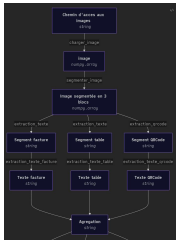
Modèle et Schémas

- Modèle physique des données

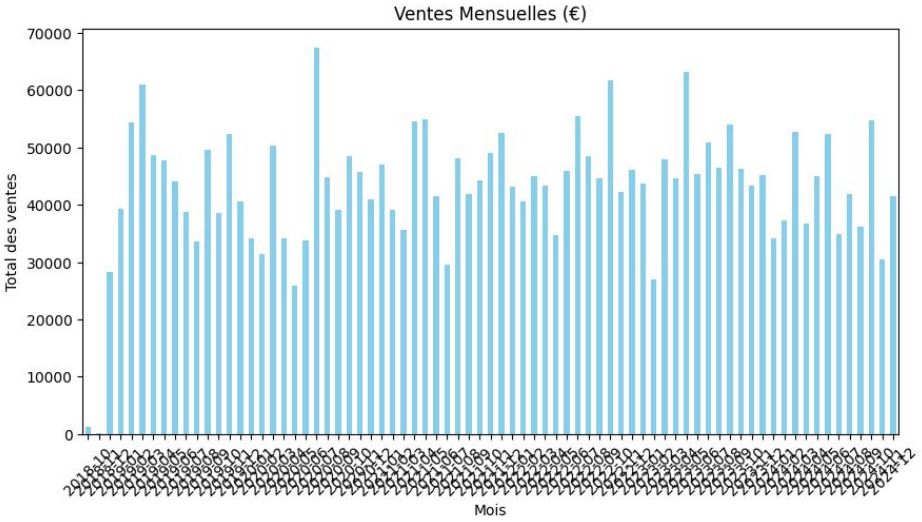
- Schéma Conceptuel

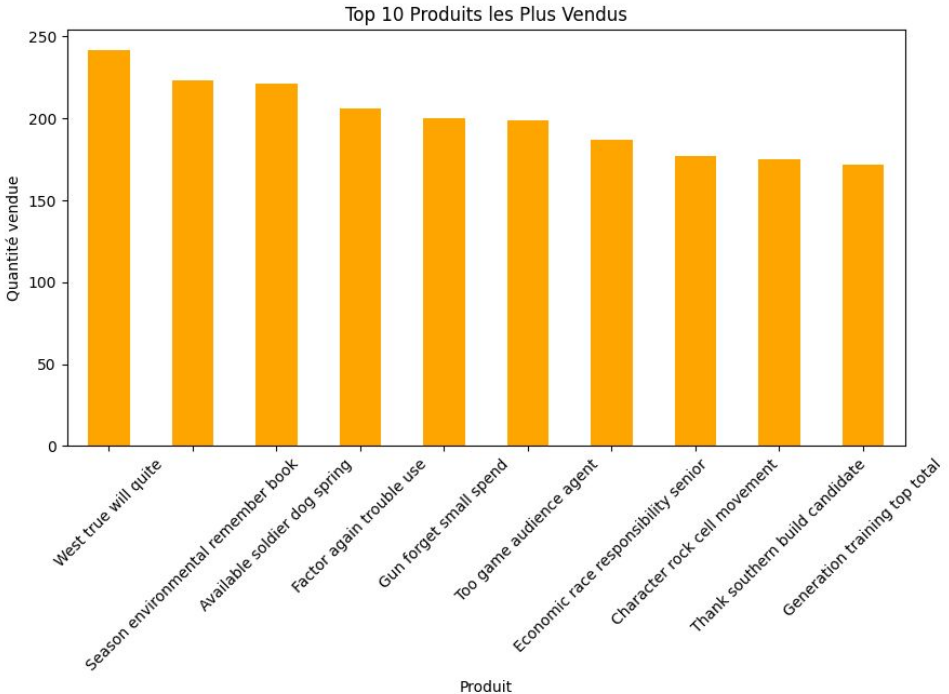
- Schéma Fonctionnel





Analyse de données





Démonstration

- Figma

- Application Web

Perspectives et Améliorations

- Traiter toutes les factures
- Implémenter des tests
- Implémenter des vérifications (OCR/QR Code)
- Implémenter une vraie authentification
- Ajouter toutes les données de monitoring en bdd
- Pipeline Ci/Cd
- Utiliser la bdd pour la segmentation RFM

Conclusion

https://github.com/VestiC1/ProjetOCR

Merci pour votre écoute !

Avez vous des questions ?