Projet OCR Invoices

(Optical Character Recognition System)

RAPPORT DE PROJET

Submitted by :
Saeedullah RUSHAN ZAMIR

Sous la supervision de

M. Emmanuel Goudot
M. Rémi Julien

1. INTRODUCTION

Amélioration du Pre-processing des factures fournisseurs avec OCR.

Notre collaboration avec [Client Name] implique un projet transformateur visant à révolutionner leurs procédures de Pre-processing des factures fournisseurs. En exploitant des technologies d'IA de pointe, notamment la Reconnaissance Optical Character Recognition (OCR), notre objectif est d'étendre les fonctionnalités de leur flux de travail existant pour automatiser le reporting des comptes fournisseurs.

Objectifs du Projet:

Grâce à l'accès aux sources de traitement de [Client Name], y compris l'API pour les factures numérisées, les dépôts Git, la visualisation des données JSON et les bases de données SQL, nous avons pour mission d'intégrer l'API des Services Cognitifs Azure et de mettre en œuvre des fonctionnalités OCR. Cela nous permettra d'extraire les informations pertinentes des factures numérisées, de calculer les métriques essentielles à partir des résultats de l'API et d'établir un seuil de qualité minimum pour l'exactitude de l'OCR.

Automatisation du Processus:

Notre objectif est d'automatiser l'ensemble du processus de Preprocessing, de la numérisation des factures au stockage dans la base de données, tout en assurant une intégration transparente dans une interface web conviviale. Cette interface offrira un accès en temps réel aux données des factures traitées, facilitant ainsi le suivi et l'analyse des comptes fournisseurs.

Livraison et Résultats Attendus:

En documentant notre progression, en versionnant notre code source et en livrant une application web fonctionnelle accompagnée d'une base de données alimentée, nous visons à donner à [Client Name] des informations exploitables, des opérations rationalisées et une efficacité accrue dans leurs processus financiers.

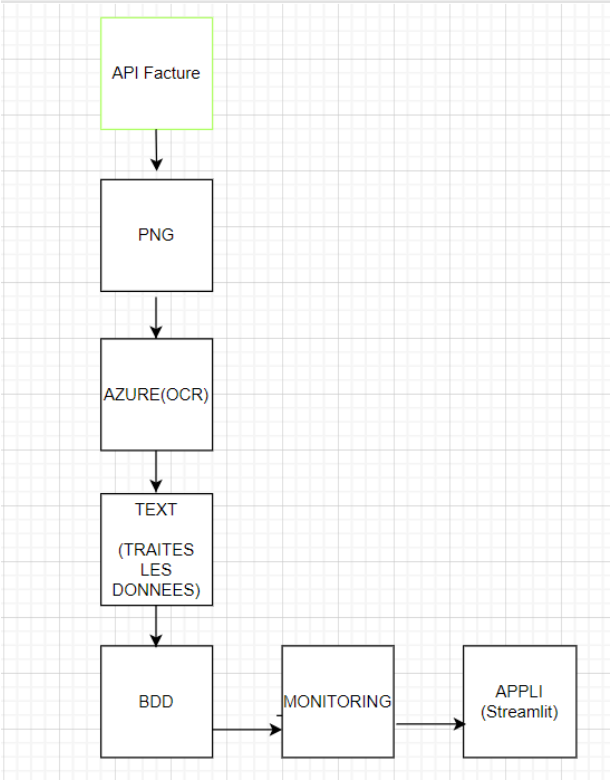
À travers ce projet, nous nous efforçons de démontrer la valeur de l'automatisation pilotée par l'IA dans l'optimisation des flux de travail commerciaux et la réalisation de résultats commerciaux tangibles pour notre client.

Tools and Technologies Used:

- Services Azure pour détecter le texte
- Python
- DBeaver pour la gestion de bases de données
- Connexion avec db en python
- Invoices Access API
- GitHub
- Trello pour la gestion de projet

2. System Design

- **Flow of system**



```
from azure.cognitiveservices.vision.computervision import ComputerVisionClient
from azure.cognitiveservices.vision.computervision.models import OperationStatusCodes
from msrest.authentication import CognitiveServicesCredentials
from dotenv import load_dotenv
from PIL import Image
import requests
import io
import pyzbar.pyzbar as pyzbar
import re
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from textblob import TextBlob
import os
import time
```

INVOICE FAC_2019_0001

Issue date 2019-01-01 08:21:00

Bill to Sarah Smith

Address 0496 Brianna Crossing
New Tabitha, RI 41854

Celui travail élément apporter.	4	x	98.58 Euro
Ad illo adipisci quaerat.	5	x	63.71 Euro
Exercitationem alias dignissimos labore.	2	x	15.50 Euro
Mention back sound center.	2	x	68.79 Euro
Economic everybody north three.	4	x	11.11 Euro
Seulement derrière faute tard.	2	x	89.91 Euro
Whole administration rich final.	4	x	91.62 Euro
TOTAL			1472.19 Euro

3. Text Recognition et Preprocessing Challenges

Au cours de ce projet, la reconnaissance de texte a posé des défis significatifs, notamment en ce qui concerne le formatage et la cohérence du texte extrait. Malgré une extraction réussie à l'aide de l'API des Services Cognitifs Azure, la sortie manquait souvent de la structure nécessaire pour un traitement et une analyse ultérieurs.

Problèmes de formatage avec la sortie de texte brut

Le texte extrait présentait fréquemment des incohérences, entravant l'analyse et le traitement efficaces. Cette incohérence, particulièrement prévalente dans les tâches de traitement des factures, posait des défis en raison du besoin critique de données structurées.

Leveraging Regex for Structuring Extracted Text

Afin d'améliorer la qualité et la lisibilité du texte extrait, j'ai utilisé les expressions régulières (regex) pour le formater. Cette méthode a permis de structurer efficacement le texte en éliminant les caractères spéciaux, en le convertissant en minuscules et en supprimant les espaces blancs superflus. En outre, j'ai incorporé les Librairie NLTK et TextBlob pour effectuer des tâches de pré-processing plus avancées, telles que la tokenisation, la lemmatisation et la correction orthographique. Cette approche simplifiée a contribué à améliorer la qualité et la cohérence du texte tout en maintenant une méthodologie de pré-processing concise.

Avantages des Expressions Régulières dans les Projets OCR

Les expressions régulières (regex) offrent une solution efficace pour structurer le texte extrait dans les projets OCR. Leur utilisation permet une extraction précise des informations, quel que soit le format des documents. De plus, les regex automatisent les tâches de traitement de texte, simplifiant ainsi le flux de travail OCR. En outre, le pré-processing basé sur les regex contribue à améliorer la qualité et la cohérence du texte extrait, offrant ainsi une structure claire et ordonnée.

4. Conception et gestion de bases de données

Dans la mise en œuvre de notre projet OCR Factures, une base de données robuste et bien organisée est cruciale pour stocker et gérer efficacement les données de facture extraites. Vous trouverez ci-dessous un aperçu des aspects de conception et de gestion de la base de données de notre projet :

Base de données Schema:

Le schéma de la base de données est conçu pour prendre en charge différents types de données de facture extraites via OCR. Il comprend des tableaux pour stocker des informations essentielles telles que les numéros de facture, la date d'émission, le nom du client, les articles, la quantité, le prix, les totaux.

Voici un aperçu général du schéma de base de données :

- Invoices Table: Cette table stocke des informations générales sur chaque facture traitée, telles que le numéro de facture, la date, le nom du client et le montant total..
- Product Table: Cette table capture des informations détaillées sur les articles de chaque facture, notamment la description, la quantité, le prix unitaire et le prix total.

Gestion de base de données:

DBeaver, un outil de gestion de base de données polyvalent, est utilisé pour gérer la base de données associée à notre projet de factures OCR. Avec DBeaver, nous pouvons effectuer diverses tâches de gestion de bases de données, notamment :

- **Connectivité de base de données :** établissement de connexions avec le serveur de base de données pour accéder et manipuler les données.
- **Gestion du schéma :** création et modification du schéma de base de données selon les besoins pour s'adapter aux changements ou aux ajouts aux exigences du projet.
- **Manipulation des données :** Insertion, mise à jour et suppression d'enregistrements de données dans les tables de la base de données.

- **Requêtes et rapports :** Exécuter des requêtes SQL pour récupérer des informations spécifiques de la base de données et générer des rapports selon les besoins des parties prenantes du projet.
- **Optimisation des performances :** Surveillance des performances de la base de données et optimisation des requêtes et des index pour garantir une récupération et un traitement efficaces des données.

En gérant efficacement la base de données à l'aide de DBeaver, nous garantissons que les données de facture extraites sont stockées en toute sécurité, organisées systématiquement et facilement accessibles pour une analyse et des rapports plus approfondis au sein de notre système de factures OCR.

5. Application Streamlit

L'application Streamlit est conçue pour faciliter le traitement par reconnaissance optique de caractères (OCR) des images de facture, permettant aux utilisateurs d'extraire le texte des images téléchargées. Il se compose de plusieurs éléments clés :

Page OCR : Cette page offre aux utilisateurs la fonctionnalité permettant de télécharger une image d'une facture et d'effectuer un traitement OCR. Lors du téléchargement d'une image, les utilisateurs peuvent cliquer sur le bouton « Effectuer l'OCR » pour lancer le processus d'extraction de texte. Le texte extrait sera affiché sur la page.
Reporting page : Cette page est un espace réservé pour le reporting automatisé de la comptabilité des fournisseurs. Il peut être développé davantage pour fournir des informations et des analyses basées sur les données de facture extraites.
Monitoring Page: Cette page sert d'espace réservé pour surveiller le service Azure OCR. Il peut afficher des mesures pertinentes et des mises à jour de statut concernant le service de traitement OCR.

L'application exploite Azure Cognitive Services pour le traitement OCR, en utilisant l'API Computer Vision pour extraire le texte des images. Il s'intègre également à une base de données pour interroger les détails des clients en fonction des numéros de facture.

Dans l'ensemble, l'application Streamlit fournit une interface conviviale pour le traitement OCR des factures, permettant une extraction efficace des données textuelles à partir des images de facture.

Présentation de l'Application Streamlit: