Raconter

**Pas de technique, pas de détails.**

*Présenter le projet La problématique (Ce qu’on va réaliser) ce que l’organisme demande, des contraintes (la technologies l’architecture…)*

Introduction :

Dorénavant, le covid-19 et le changement climatique, sont deux événements qui changeront notre façon de vivre et orienteront les innovations dans plusieurs secteurs, et nécessairement le secteur bancaire. Le déplacement aux agences pour créer un compte bancaire devient dépasser. C’est pour cela les filiales de la Société Générale dans la zone africaine ont pensé à lancer le projet d’Entrer En Relation EER qui permet aux leurs clients de créer un compte bancaire 100% à distance et sans déplacer aux agences. Le parcours d’EER, qui ne doit pas dépasser 10 minutes, demande au client de justifier ses informations saisies en téléchargeant des documents justificatifs sinon le compte ne sera pas créé.

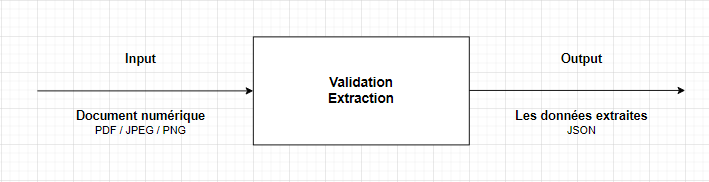
*Lemme la clé (la problématique ⬄ la solution) : la porte : la liaison entre la problématique et la solution*

Après le téléchargement (Upload) des documents justificatifs, un agent de la filiale sera chargé pour valider la compatibilité entre les informations saisies et les justificatifs. La complexité de cette tâche varie proportionnellement avec le nombre de demande d’ouverture d’un compte. D’où la nécessité d’avoir une solution qui permet la validation de documents et l’extraction des données sera très utile et même aussi réutilisable.

*Présenter la solution*

Mon Projet de Fin d’Etudes intitulé : « une solution intelligente d’extraction des données structurées et validation des documents justificatifs pour le parcours Entrée En Relation client à distance » il s’agit de réaliser une solution intelligente, intégrable dans le projet d’EER, qui permet de valider les documents téléchargés et l’extraction des données structurées à partir de ces documents.

Alors, on résume la solution à réaliser dans ce diagramme simple :



Input ou l’entré c’est un document numérique de type PDF, JPEG ou PNG. (Avant le traitement, on convertit tous en JPEG. Pour le PDF on convertit ces pages en des images JPEG. A mon avis le types de l’image n’impacte pas le résultat d’extraction, malgré j’ai trouvé quelques postes at article recommande le format FITT :

<https://www.dynamsoft.com/blog/insights/scan-settings-for-best-ocr-accuracy/#:~:text=Lossless%20compression%20is%20the%20option,with%20each%20edit%20and%20save>.

<https://stackoverflow.com/questions/10193001/whats-the-best-image-input-type-for-tesseract>

).

Le cœur d’application ou traitement : il s’agit de deux opérations globales. La première c’est la validation de document c’est-à-dire la détection de type de document : Carte d’identité, Passeport, permet de conduit, … Techniquement, ce n’est qu’une classification d’images. La deuxième, c’est l’extraction des données structurées. (Plus de détails dans la suite)

Output ou la sortie, c’est un objet JSON qui contient les données extraites, où cas d’un document invalide cet objet sera vide.

***Noter bien : que dans ce PFE on s’intéresse par le traitement des document numérique (Classification & Extraction), ni de IHM, ni de gestion de base de données. ;***

***Scénario d’exécution****: (Dans ce scénario je vous montre les grands partie)*

*Pour la mise en production, cette solution doit traiter aux utilisateurs de n’importe quelle filiale après le téléchargement de ces documents justificatifs, la*

*Minimum : Client upload document => recevoir le doc => Image processing => validation de doc => l’extraction des données sous format XML / JSON => l’envoi des donnée (Au client ou bien autre serveur).*

* *Noter bien : Si l’utilisateur envoie un pdf on le converte en image (une démarche pour OCR)*

*Avancé : Client upload doc ou prend une image et detection en Real-time le doc => recevoir => Image processing => Classification (validation) => extraction => Structure les données\* => l’envoie des données.*

*Noter Bien : Que l’application EER servie 15 filiales.*

*Dans une filiale parmi les quinze un utilisateur veut créer un compte*

*Détailler la Problématique les difficultés (1. Document justificatif comment savoir s’il s’agit d’un document justificatifs, le type numérique de document (pdf, jpeg, png), la qualité de l’image (la visibilité, bien encadré), le temps de réponse (Les solutions intelligentes Machine Learning), ).*

1. *Les documents (les documents justificatifs ne sont pas bien définis)*
2. *Dataset*

* *L’utilisateur télécharger un document de type : pdf, jpeg ou png (traitement d’image) (Recevoir un fichier)*
* *Classification de ce document (1)*
* *L’extraction des données structurées (Envoie des données (sous format XML, ou JSON))*

Avant de

*Détailler la solution : la conception et l’architecture Mise en production (Théoriquement Comment ?).*

*=> API (Serveur - client),*

*=> Rendre le fichier en format lisible (Pre-processing)*

*=> Construire une solution qui permet la classification de document (Model ML de classification, CNN, RNN) (Suivant quels critères)*

*=> L’extraction des caractères OCR Optique Character Recognition*

Cette application (la solution) doit servir 15 filiales, autrement-dit elle doit traiter 15 exemplaires pour chaque document justificatif. Avant d’aller plus loin il faut répondre à la question suivante : c’est quoi un document justificatif et quelle est la liste des document justificatif

Préexistant :

**Techniquement parlant**

Projet => {…sous projets == sous problème…}

Morceau (Tu développer Comment j’ai fait ? (Ce qu’on va faire pour réaliser ce morceau d’application), les technologies adoptées l’architecture, les difficultés rencontrées ...)

**API** (Backend - Frontend) (Server (Python)- Client (Application web/Mobile)) **90% (Tensorflow server)**

**Image Processing: (Resize, Wrap (Segmentation), Orientation, Denoising…) 100%**

**Modèle ML**

**Modèle Détection : (Object Detection SSD\_MobileNet | Faster-RCNN (Coco | id-card -detection)) Sur Frontend la technologie adopté (Blockage) les contraintes Temps de calcule (la précision). Sur Backend (Pour Wrap)**

**Modèle De classification: CNN**

**Modèle D’extraction:**

**OCR (Tesseract résulta) (tr résultat (github))**