

Année Universitaire 2023-2024

“Aim for simplicity in Computer Science. Real creativity won't make things more complex.  
Instead, it will simplify them.”

***Damian Duffy Mingle***

## **RESUME**

Le projet de redressement d'images avec Flask propose une solution innovante pour simplifier la gestion d'images de documents inclinées. En réponse à la problématique de la manipulation d'images tournées, l'application offre une plateforme conviviale où les utilisateurs peuvent télécharger des images inclinées, les soumettre au système, et recevoir une version redressée contenant un texte lisible.

Cette initiative repose sur la nécessité de résoudre les défis liés à la manipulation d'images de documents prises sous des angles non conventionnels. Plutôt que de recourir à des logiciels complexes, le projet s'appuie sur des technologies web modernes, notamment le framework Flask, ainsi que sur des bibliothèques telles qu'OpenCV et Pillow, pour automatiser le processus de redressement d'images.

Le but principal de cette application est de fournir une solution intuitive et efficace permettant de convertir des images inclinées en versions plates, prêtes à être lues aisément. Les techniques de traitement d'images, telles que la détection de contours et la transformation de perspective, sont exploitées pour redresser les images et mettre en évidence le contenu textuel.

Ce rapport explorera en détail les aspects techniques de l'implémentation, les méthodes de traitement d'images employées, ainsi que les résultats obtenus. L'objectif final est de présenter une solution accessible et pratique pour le redressement automatique d'images inclinées, améliorant ainsi l'expérience de gestion de documents pour les utilisateurs.

# OUTILS UTILISÉS

## I. Introduction

Explorons les outils essentiels qui ont été employés dans le développement du projet. Chaque choix technologique a été méticuleusement considéré pour assurer une implémentation efficace et cohérente. L'objectif est de fournir une vision concise des outils logiciels et des bibliothèques qui ont contribué au succès de l'application, offrant ainsi un aperçu du paysage technologique qui sous-tend ce projet.

## II. Outils utilisés

### II. 1 Flask :



Un framework open-source de développement web en Python. Il est classé comme microframework car il est très léger. Flask a pour objectif de garder un noyau simple mais extensible.

### II. 2 Numpy :



Bibliothèque destinée à manipuler des tableaux.

### **II. 3 Pillow :**



Python Imaging Library est une bibliothèque de traitement d'images pour le langage de programmation Python. Elle permet d'ouvrir, de manipuler, et de sauvegarder différents formats de fichiers graphiques. La bibliothèque est disponible librement selon les termes de la Python Imaging Library license.

### **II. 4 OpenCV :**



OpenCV est une bibliothèque libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société de robotique Willow Garage, puis la société ItSeez se sont succédé au support de cette bibliothèque.

### **II. 5 GitHub :**



Github, est une entreprise de développement et services logiciels aux États-Unis. Github développe notamment la plateforme Github, la structure Electron.

## II. 6 Matplotlib :



Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous forme de graphiques. Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy.

## IV. Conclusion

En bref, le choix stratégique d'outils tels que Flask, OpenCV, Pillow, NumPy, et la gestion via GitHub a été essentiel pour le succès de notre projet de redressement d'images. Ces technologies ont collaboré de manière harmonieuse pour créer une interface utilisateur réactive, des fonctionnalités avancées de traitement d'images, et une gestion collaborative efficace du code source. En résumé, ces choix ont été déterminants pour l'efficacité, la robustesse et l'évolutivité de notre application. Le dépôt GitHub a également joué un rôle crucial en facilitant la collaboration continue au sein de l'équipe de développement.

# RÉALISATION

## I. Introduction

Ce chapitre offre une plongée visuelle dans la concrétisation de notre projet. À travers des captures d'écran détaillées, nous explorons l'interface utilisateur, mettant en lumière les fonctionnalités clés de l'application. Ces captures d'écran fournissent un aperçu immédiat de l'expérience utilisateur, illustrant les étapes du processus, depuis le téléchargement d'une image jusqu'à l'obtention de sa version redressée.

## II. Présentation visuelle

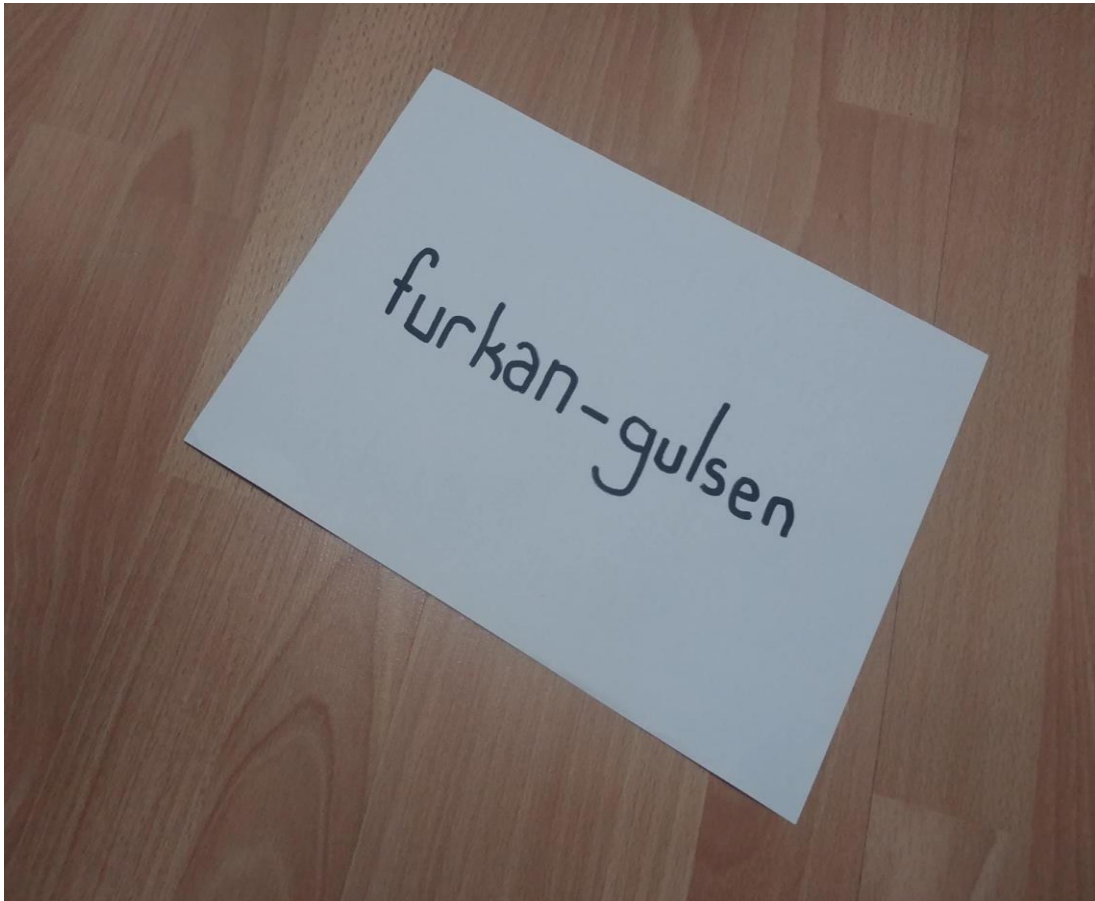
En examinant ces captures d'écran, les lecteurs auront une compréhension concrète de la manière dont l'application prend vie, offrant ainsi une perspective complète sur la réalisation réussie de notre projet.

### III.1 Page de Téléchargement d'Images - Détection de Documents :



Lorsque l'utilisateur accède à la page de téléchargement d'images de notre application web, il a la possibilité de télécharger des images contenant des documents.

Prenons l'exemple d'une image illustrant un document posé sur un bureau. Cette image est téléchargée via l'interface web de notre application :



Une fois l'image téléchargée, notre application entreprend un processus de traitement d'image sophistiqué pour détecter et améliorer le document présent dans l'image. Les étapes clés de ce processus sont les suivantes :

#### **1. Prétraitement :**

L'image originale est redimensionnée pour assurer une hauteur de 500 pixels tout en préservant les proportions. Conversion de l'image redimensionnée en niveaux de gris. Application d'un flou gaussien pour réduire le bruit. Détection des contours grâce au détecteur de bords Canny.



## **2. Identification du Document :**

Les contours détectés sont analysés pour identifier les formes quadrilatérales. Les contours les plus grands, supposés être le document, sont retenus. Une approximation est réalisée pour déterminer si le contour a quatre sommets, confirmant qu'il s'agit d'un document.

## **3. Correction de la Perspective :**

La perspective de l'image originale est ajustée en fonction des coordonnées du document identifié, donnant une vue de dessus.

## **4. Amélioration de l'Image :**

L'image résultante est convertie en format RGB. Elle est redimensionnée pour une largeur de 500 pixels, garantissant une présentation claire tout en conservant les proportions.

## **5. Affichage du Document Amélioré :**

L'image améliorée est convertie en format base64 et présentée à l'utilisateur via l'interface web. Seul le document est retourné, mettant en évidence le texte présent sur celui-ci. Ce processus crée une expérience semblable à celle d'un scanner, où l'application détecte et améliore les documents dans une image, fournissant ainsi une version optimisée du document pour une lisibilité maximale.

### III.2 Page de Résultats - Affichage du Document :

Après que l'utilisateur a téléchargé une image et que le processus de traitement d'image a été effectué avec succès, la page de résultats affiche l'image améliorée du document. Explorons cette section de l'application en mettant en lumière les caractéristiques clés.



- Lorsque l'utilisateur accède à la page de résultats, l'image du document est présentée dans un format optimisé. Cette image a subi un traitement qui inclut la détection des contours, la correction de la perspective, et l'amélioration de la qualité visuelle.
- Le document affiché comprend non seulement le contenu original mais également une amélioration visuelle. Le texte présent sur le document est clairement visible et optimisé pour une lecture facile. Les étapes de traitement ont été conçues pour garantir que le texte est net et bien défini, améliorant ainsi la lisibilité globale du document.

## **IV. Conclusion**

L'objectif de cette section est de présenter visuellement les résultats tangibles du développement, permettant aux lecteurs de se familiariser avec l'apparence et le flux de l'application. Les captures d'écran sélectionnées mettent en évidence les points saillants de l'interface, soulignant la convivialité de la conception et la manière dont les utilisateurs interagissent avec les fonctionnalités de redressement d'images.

## CONCLUSION GENERALE ET PERSPECTIVES

En conclusion de ce projet de redressement d'images avec Flask, nous avons réussi à créer une solution pratique et efficace pour le traitement automatique d'images inclinées. L'utilisation judicieuse d'outils tels que Flask, OpenCV, Pillow, NumPy, et la gestion via GitHub a jeté les bases d'une application robuste répondant aux besoins spécifiques du redressement d'images.

L'exploration des captures d'écran dans le chapitre de réalisation a offert un aperçu visuel des fonctionnalités de l'application, mettant en évidence la convivialité de l'interface et la facilité d'utilisation pour les utilisateurs finaux.

Cependant, ce projet ne s'arrête pas là. Des perspectives d'amélioration et d'extension se dessinent à l'horizon. Les développements futurs pourraient inclure l'intégration de fonctionnalités avancées de traitement d'images, une amélioration de la gestion des erreurs, et peut-être l'exploration d'algorithmes plus sophistiqués pour une détection de contours plus précise.

La collaboration au sein de l'équipe de développement via GitHub a été cruciale pour le succès du projet, et cette approche collaborative peut être étendue pour accueillir des contributions externes et des améliorations continues.

En définitive, ce projet a démontré la fusion réussie entre les technologies web modernes et le traitement d'images avancé. Les bases solides établies ouvrent la voie à des applications similaires dans des domaines tels que la numérisation de documents, la reconnaissance optique de caractères (OCR), et bien d'autres.

Nous sommes convaincus que cette application de redressement d'images contribuera positivement à la simplification des flux de travail liés à la gestion de documents inclinés, tout en servant de base solide pour des projets futurs dans le domaine passionnant de la vision par ordinateur et du traitement d'images.