Rapport de projet

Laurent Antoinette, Romain Campillo, Tony Nguyen

9 mai 2023

Table des matières

1 Présentation du Sujet **4**
- 1.1 Le problème . . . 4
- 1.2 . . . 4
- 1.3 Les différentes approches . . . 4
  - 1.3.1 k-Nearest Neighbor . . . 4
  - 1.3.2 Template matching . . . 4
- 1.4 Objectifs . . . 4
- 1.5 Cahier des charges . . . 4
  - 1.5.1 Besoins et contraintes . . . 4
  - 1.5.2 Résulats attendus . . . 5

2 Technologies utilisées **6**
- 2.1 Langages et outils . . . 6
  - 2.1.1 Python3 . . . 6
  - 2.1.2 OpenCV . . . 6
  - 2.1.3 You Only Look Once . . . 6
- 2.2 . . . 6

3 Développements Logiciel : Conception, Modélisation, Implémentation **7**
- 3.1 Interface Homme-Machine . . . 7

4 Algorithmes et Analyse **9**
- 4.1 YOLO . . . 9
- 4.2 Histogramme . . . 9

5 Analyse des Résulats **10**

6 Gestion du Projet **11**

7 Bilan et Conclusions **12**

8 Bibliographie **13**

9 Annexes **13**

Table des figures

1 Diagramme de cas d'utilisation 7
2 Fenêtre de l'IHM quand on modifie une image 7
3 Fenêtre de l'IHM après présentation des charactères reconnu 8
4 Exemple d'une image après filtrage 9
5 Images de nos 3 lignes crée à partir de l'image de départ 9
6 Imagettes des charactères crée à partir de la 2e ligne 9
7 Imagettes des charactères crée à partir de la 3e ligne 10

1 Présentation du Sujet

1.1 Le problème

L'essence du problème est q'un texte dans un document est très diffrent d'une photo du même document. Dans le 1er cas, le format rend le contenu comprehensible par un ordinateur, il peut donc chercher un mot dans le texte ou le modifier pour corriger des fautes d'orthographes. Dans le 2e cas, l'image représente la même chose. MAIS, la machine ne comprend le sens de l'information représenté. Il voit les pixels mais pas les charactères. De plus, des imperfections dans la capture du document change la photo (ex : luminosité, angle ...), il y a une modification des informations alors que le texte ne change pas.
Réussir à lire du texte sur des images permettrait de faciliter la digitalisation de document papier. Ainsi, libérer les Humains de quelques tâches redondantes. On peut imaginer ce genre de système en train de traiter des formulaires ou encore d'améliorer les moteurs de recherches avec des images qui contient du texte.

1.2

Ce problème est particulièrement intéressant. En effet, tout ce qui traite de l'information appartient à notre domaine d'étude. En plus de l'automatisation. Nous avons pu constaté une explosion de l'inteligence artificielle ces dernières années. Notament les réseaux neuronal. La reconnaissance d'image fait parties des objectifs qu'ils ont déjà atteint. Ce projet est tout à fait intéressant pour les futures étudiants en master IASD.

1.3 Les différentes approches

1.3.1 k-Nearest Neighbor

Méthode statistique, il y a un plan avec plein d'objet qui forme des troupeau, notre lettre est quelque part dans le plan, on trace un cercle autour, notre lettre est identifié comme l'objet présent au plus grand nombre à l'intérieur du cercle.

1.3.2 Template matching

En gros, comparaison de l'image avec la base de donnée entière à l'aide d'une fonction de distance.

1.4 Objectifs

Notre but est de créer un logiciel dont une personne ce servirait pour tranformer une image qui contient des symboles en un texte manipulable par un ordinateur.
Notre but initiale sera de reconnaitres des images simples et de bonnes qualités avec des symboles de l'alphabet latin (26 lettres) seulement en majuscules sur 1 lignes.

1.5 Cahier des charges

1.5.1 Besoins et contraintes

Les besoins

Capturer une image L'utilisateur pourra prendre des photos par notre logicielle à l'aide d'une webcam. Mais il pourra également utiliser des images depuis son système de fichiers. Tout cela à travers une interface Homme-Machine.

Pré-traitement Le logiciel réduira le bruit et rendra l'image plus net à l'aide de plusiseurs filtres prédéfini.

Segmentation de l'image Les différents charactères présents sur l'image seront localiser à l'aide d'histogramme de projection.

Extraction des charactères Les charactères seront ensuite découpé pour formé leurs propres imagettes .ie une image qui contient TEST devriendra 4 petites imagettes contenant respectivement T E S T.

Reconnaissance Les imagettes feront l'objet d'une reconnaissance de façon individuelle. Dans notre cas nous choisissons d'utiliser un réseau neuronal convolutif.

Présenter Une fois que les imagettes sont reconnu en charactères. Il est nécessaire de les assembler et d'afficher à l'utilisateur les mots reconnus.

Les contraintes Nous nous fixons comme contraintes de ne pas utiliser de services comme Google Collab car Google possède un modèle économique type "Freemium" (gratuit mais payant). Nous souhaitons créer un logiciel suffisament performant pour pouvoir être exécuter sur l'une de nos machines personelles.
De plus, une webcam est nécessaire, ou alors un appareil photo numérique.

1.5.2 Résulats attendus

Notre but est de réaliser un projet qui soit capable de reconnaître des lettres manuscrites sur un fond blanc. Les lettres seront en charactères majuscule non-lié et sans accents ni charactère spécial.

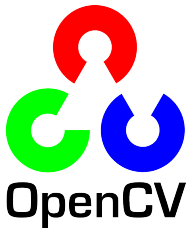
2 Technologies utilisées

2.1 Langages et outils

2.1.1 Python3

2.1.2 OpenCV

Dans ce projet, nous choisissons d'utiliser la librairie open source de vision par ordinateur (Open Computer Vision). Elle contient les fonctionnalités de pré-traitement d'image nécessaire dans ce projet. Nous aurons notament besoins des fonctions de filtrage d'image.



2.1.3 You Only Look Once

You Only Look Once (YOLO) est une archicture de Réseau Neuronal Convolutif (CNN) de type "Region based" (R-CNN) qui est capable de à la fois de localiser des objets dans une image et en même temps, les classifier.

Expliquons (un peu tout petit peu) plus en détail les réseaux neuronals
Les réseaux de neurone simule plus ou moin le fonctionnement de notre cerveau. Parlons d'abord de l'éléménent le plus petit et on va progresser jusqu'aux CNN.

Le neurone va prendre plusieurs entrées, des nombres, faire une opération avec une fonction d'activation et en sortie retourner le résultat.

Réseau de neurone L'architecture des réseau de neurone simple sont organisé en couche. La 1er couche qui est l'entrée (sensé reprensenter les yeux). La dernière couche représente la sortie, par exemple si le réseau est entrainé pour reconnaitre des chiens, la sortie doit retourner une valeur proche de 1 si c'est un chien et 0 sinon. Entre le début et la fin du réseau, les couche intermédiaires sont connectés vers l'avant et de façon complète. Un neurone est connecté à tout les neurones de la couche précédente et suivante mais pas à ceux de la même couche.
Le désavantages est que cette approche est qu'il est nécessaire de lui donner des "charactéristique". Pour reconnaitre si cette animal est un chien on ne va pas luis donner l'image directement, on va lui dire entrée si ça a des poils, la couleur de son pelage, le nombre de pattes, la forme des oreilles.

Les CNN blablabla

R-CNN

2.2

3 Développements Logiciel : Conception, Modélisation, Implémentation

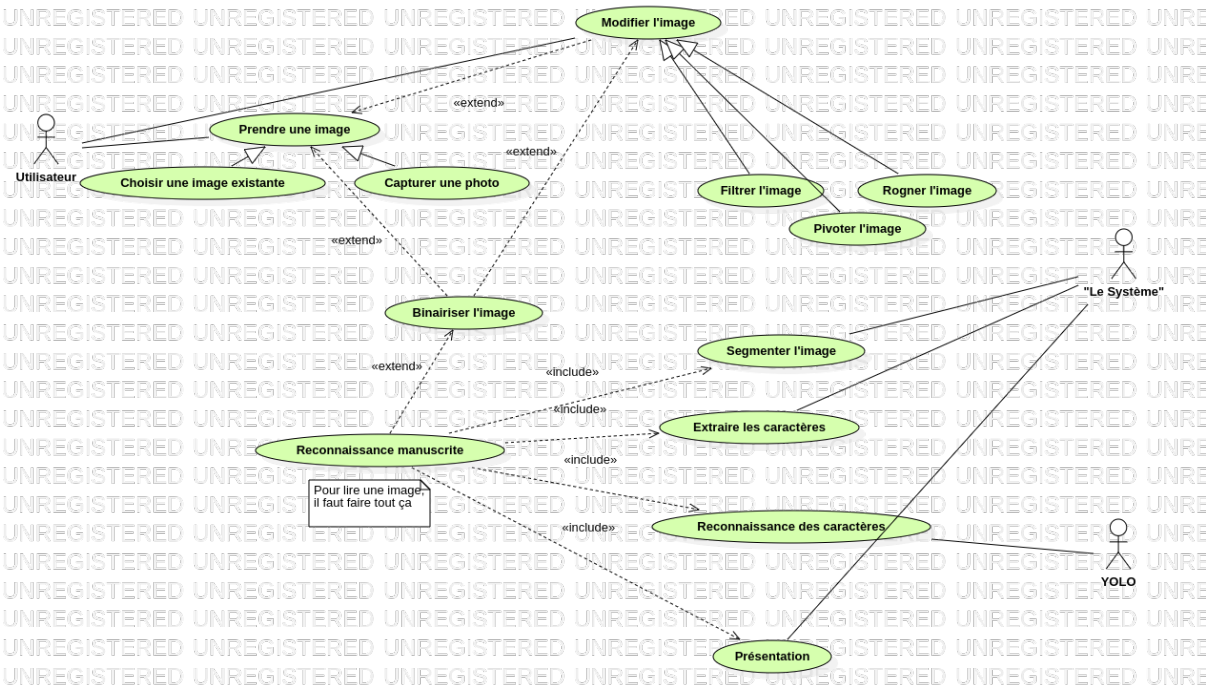


FIGURE 1 – Diagramme de cas d'utilisation

3.1 Interface Homme-Machine

L'IHM nous permet de prendre des photos avec la webcam connecté à l'ordinateur ou choisir à partir d'une fenêtre un chemin vers une image.

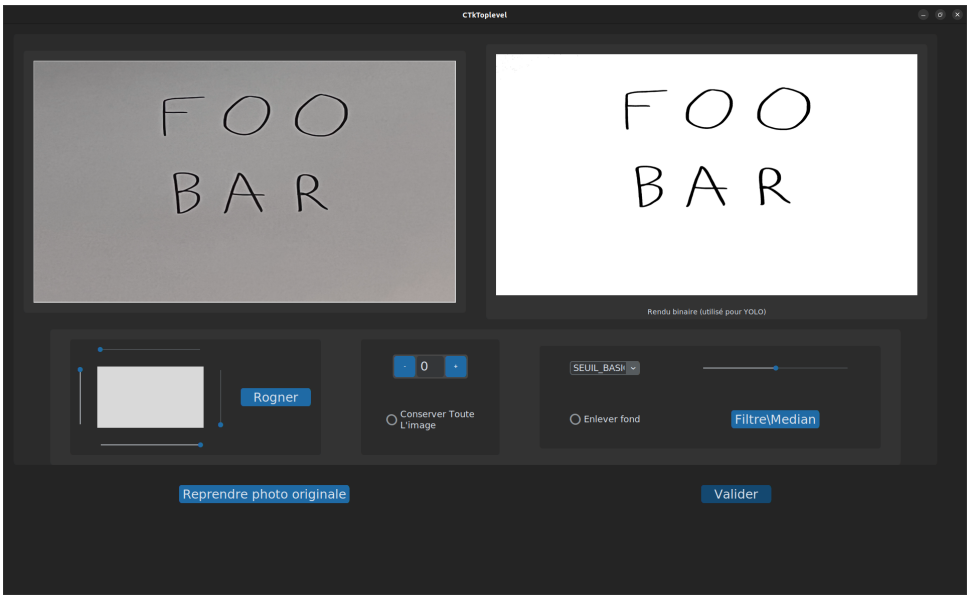


FIGURE 2 – Fenêtre de l'IHM quand on modifie une image

Après avoir entrée une image, il es possible qu'il y ait beaucoup de bruit, dans ce cas là, nous pouvons rogné l'image pour enlever des ombres sur les bords, faire des rotation et filtrer pour enlever le bruit comme illustrer sur la figure 2

FIGURE 3 – Fenêtre de l'IHM après présentation des charactères reconnu

Sur la figure 3, on peut voir que après avoir obtenu une image de bonne qualité, notre programme va segmenter les charactères, en faire des imagettes et donner ces imagettes individuellement à YOLO. Ensuite après cette reconnaissance, l'IHM présente le résultat.

Il est possible de voir les imagettes généré dans le dossier "./images/imageSegmentee"

FIGURE 4 – Exemple d'une image après filtrage

FIGURE 5 – Images de nos 3 lignes crée à partir de l'image de départ

4 Algorithmes et Analyse

4.1 YOLO

4.2 Histogramme

On voit sur la figure 5 à la page 9 que due à un mauvais filtrage, nous obtenons une ligne qui ne devrait pas être présente.

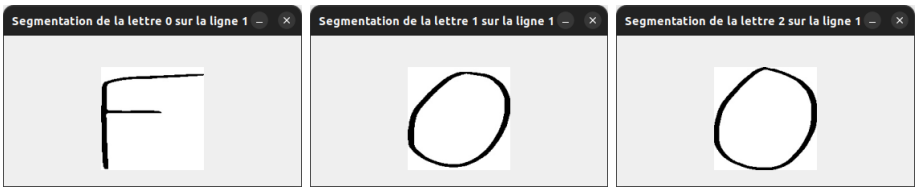


FIGURE 6 – Imagettes des charactères crée à partir de la 2e ligne

FIGURE 7 – Imagettes des charactères crée à partir de la 3e ligne

5 Analyse des Résulats

TBD

6 Gestion du Projet

Laurent : filtrage, segmentation Romain : IHM, YOLO Tony : rapport

7 Bilan et Conclusions

tests

Cependant, certains points laissent à désirer dans notre projets. Malheureusement, une contribution de l'utilisateur est nécessaire. Pour une bonne reconnaissance des charactères, l'utilisateur doit rogner l'image. Il serait plus pratique que le *rognagne soit fait automatiquement.*

Ensuite, au niveau de la segmentation, notre algorithme n'est pas capable de séparer des lettre cursives et ignore totalement les espaces.

Et enfin, dans le futur, le but sera que YOLO reconnaise aussi l'alphabet minuscule et mais aussi l'alphabet français (avec les accents).

8 Bibliographie

TBD

9 Annexes

TBD