Université Cadi Ayyad

Faculté des Sciences Semlalia Marrakech

Département d'Informatique

Projet du 14 Mai au 12 Juillet 2021

Présenté à la Faculté des sciences Semlalia de Marrakech

(FSSM) pour obtenir Le diplôme Licence Fondamentale

Sciences Mathématiques et Informatique
(SMI)

Mémoire de projet de fin d'étude

Sujet :

Automatisation du processus du choix des meilleurs opérateurs pour accomplir une tâche en traitement d'image

---

Réaliser et soutenu par :

- **EL BOUGA Latifa**
- **IKEN Samya**

Sous la direction de :

- **Pr. QAFFOU Issam**

Soutenu le 15 juillet 2021 devant la commission d'examen :

- **Pr. AGOUTI Tariq**
- **Pr.QAFFOU Issam**

Année universitaire : 2020/2021

Dédicace

Nous dédions ce travail:

À nos très chers parents

notre raison d'être, notre raison de vivre, la lanterne qui éclaire notre chemin et nous illumine de douceur et d'amour. C'est aussi notre source de force, de tendresse, de noblesse et D'affection.

À nos professeurs adorables

C'est grâce à vous qu'on a pu atteindre ce stade, grâce à votre professionnalise et à vos conseils précieux, nous sommes arrivés à cette étape marquante dans notre cursus scolaire.

À nos amis

Aussi nombreux que vous soyez nous ne saurons vous citer
Aucun mot ne pourrait exprimer notre reconnaissance et notre gratitude.
Nous vous souhaitons une bonne continuité, un bon avenir et une vie pleine de bonheur.

MERCI.

IKEN Samya & EL BOUGA Latifa

Remerciements

Au terme de ce travail, nous tenons à exprimer nos profonds remerciements à toutes les personnes qui ont contribué au bon déroulement de ce projet de fin d'études.

Ainsi, nous exprimons notre profonde gratitude et nos vifs remerciements à notre encadrant **Pr. QAFFOU Issam** non seulement pour son assistance lors du projet, mais pour sa contribution lors de notre cursus scolaire.

Nous remercions particulièrement **Pr. QAFFOU Issam** le responsable de la Licence fondamentales « Sciences Mathématiques et Informatique » pour sa disposition et ses orientations durant cette année de formation, et aussi **Pr. OIRRAK Ahmed** le chef du département informatique qui nous a offert la salle DI2 pour travailler.

Nous remercions également toutes les personnes qui nous ont apporté des retours, ainsi que de précieux conseils sur notre application tout au long de son développement.

À l'issue de trois agréables années passées au sein de la Faculté des Sciences SEMLALIA (FSSM), nous adressons des remerciements particuliers à l'ensemble du corps professoral et administratif pour l'inestimable qualité de l'enseignement qui nous a été dispensé et particulièrement le cadre professoral du département informatique.

Liste de figures

Table de matière

Résumé

Le traitement d'image joue aujourd'hui un rôle dans de nombreux domaine (médicale, astronomie, photographie, …), cependant les contraintes d'exploitation par exemple la complexité algorithmique, justifiant la multiplicité des techniques développées dans le domaine du traitement d'image.

Depuis l'apparition des applications de traitement d'image à notre temps actuel, on se pose devant l'exploitation de trois phases, chaque phase contient plusieurs opérateurs, qu'on change un opérateur forcément le résultat change, donc le problème est qu'il est le bon opérateur à choisir puisque j'ai plusieurs opérateurs, chaqu'un donne un résultat différent.

Nombreux stratégies ont été apportées dans ce domaine-là, pour nous, on opte pour l'utilisation de l'apprentissage par renforcement et en particulier l'algorithme Q-learning qui se pose sur le principe de l'exploration et l'exploitation, pour trouver la meilleure combinaison d'opérateurs à appliquer pour une tâche particulière.

Le résultat en expérimentation prouve vraiment que le Q-learning arrive à trouver parmi autres la meilleure combinaison qu'on peut appliquer.

Introduction générale

On désigne par traitement d'images l'ensemble des méthodes et des techniques opérant sur celles-ci, dans le but d'améliorer leurs qualités, de la corriger et de la modifier afin d'extraire de l'information.

Dans ce mémoire, nous aborderons aux trois phases traitement d'image qui sont : le filtrage par convolution (prétraitement), la segmentation d'image(traitement) et le filtrage morphologique (post-traitement).

Chaque phase de traitement d'image contient un nombre important d'opérateurs, citons par exemple pour la première phase : l'opérateur médian et moyenneur, pour la deuxième phase : l'opérateur Prewitt, Sobel et pour la troisième phase : l'érosion et la dilatation, et les appliquant sur une image donnée (voir les figures ci-dessous), en remarquant que si l'opérateur change le résultat est forcément changer, parfois on obtient des résultats encombrés.

- Pour la première phase :

Image originale

Application du filtre médian

Application du filtre moyenneur

- Pour la deuxième phase :

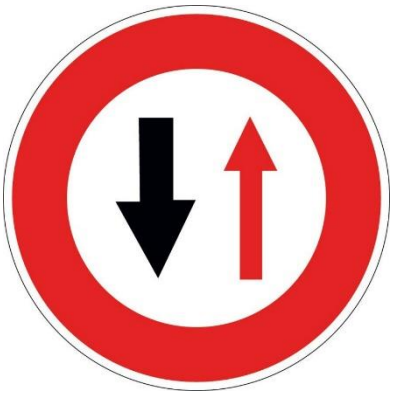
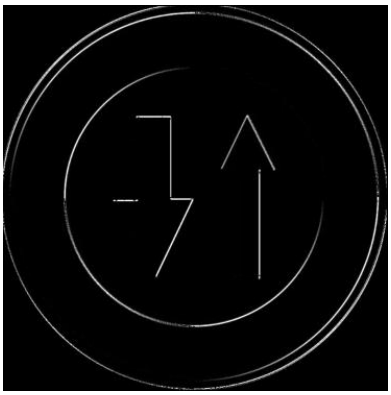


Image originale



Application du filtre Prewitt

Application du filtre Sobel

- Pour la troisième phase :

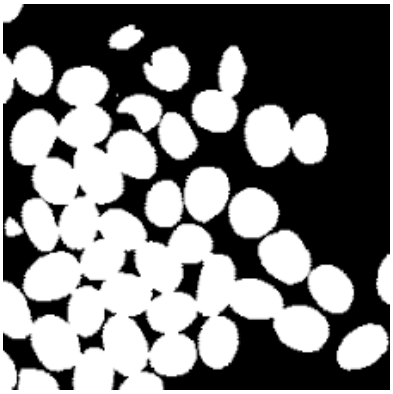
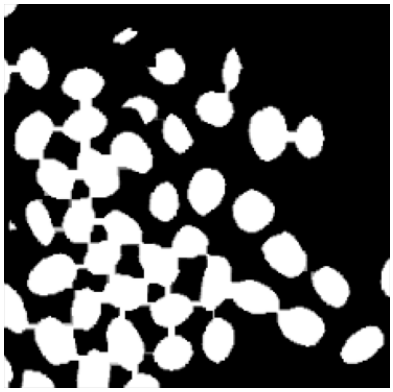
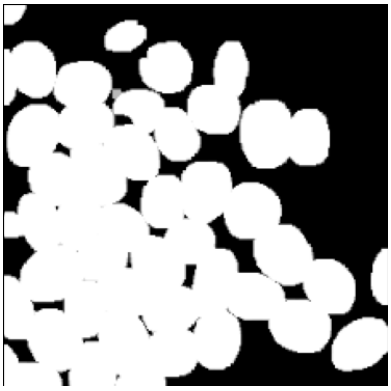


Image originale



Application du filtre érosion



Application du filtre dilatation

D'après les figures ci-dessus, on déduire que si le résultat change par un opérateur, il est normal qu'il va changer par une combinaison d'opérateurs.

La résolution de ce problème se controversé surtout dans le domaine de traitement d'image. Alors beaucoup de solutions sont apportées à cet égard.

Afin de résoudre le problème posé et d'améliorer son résultat, nous proposons une nouvelle approche qui consiste à lier le système de base de traitement d'image par l'apprentissage par renforcement, plus précisément l'algorithme Q-learning. Cette dernière est surement l'un des

algorithmes plus utilisés dans la communauté de l'apprentissage par renforcement en raison de sa simplicité.

Par voie de conséquence, ce mémoire est organisé en trois chapitres :

Le premier chapitre est consacré à la présentation du contexte générale qui contient une explication détailler de différents opérateurs du système de base de traitement d'image.

Le deuxième chapitre est dédié au vif du sujet à savoir l'explication du principe du Q-learning, ainsi la problématique du sujet et la solution proposée.

Le troisième chapitre s'intéresse à faire des expérimentations et des discutions.

Chapitre 1 : contexte général

I. Introduction

Dans ce chapitre, nous abordons les notions de base nécessaires à la compréhension des opérateurs de traitement d'image puis un aperçu sur les différents techniques qu'on a traité.

II. Système de base de traitement d'image

Un système de traitement d'image numérique est composé de :

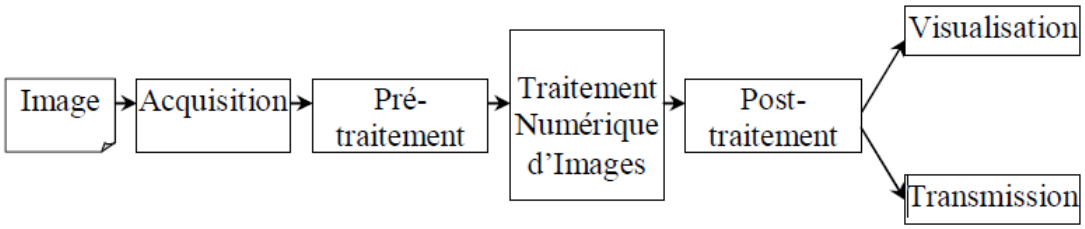


Figure 1 : Schéma de système de traitement d'image.

1) Prétraitement d'image

Cette phase a lieu juste après l'acquisition des images et a pour but objectif d'améliorer les caractéristiques d'une image. Il s'agit de supprimer le bruit ou les petits variations présentes dans une image. L'intensité d'un pixel est transformée en fonction des intensités sur un petit voisinage du pixel.

Plusieurs filtres sont utilisés pour la réduction de bruit, ils sont divisés en deux catégories :

- Filtres linéaires : comprend tous les opérateurs pouvant exprimer leur résultat comme une combinaison linéaire de niveaux de gris d'un voisinage de l'image. Ces filtres possèdent des caractéristiques spectrales, on parle ainsi de filtre passe-bas (l'image devient floue) ou de filtre passe-haut (les contours ressortent).
- Filtres non linéaires : comme le filtre médian, ils permettent de réduire le bruit impulsionnel (multiplicatif), ce que les filtres linéaires n'arrivent pas à faire.

Dans ce qui suit quelque filtres (passe-bas) réducteurs de bruit que nous avons traité.

1.1) Filtre Moyenneur

Les filtres moyenneurs, comme leurs noms l'indique calculent la moyenne des pixels situées dans le voisinage de chaque pixel. Ces filtres permettent de réduire le bruit dans l'image, réduit les détails non-important, ce qui rend les zones homogènes plus lisses.

Les filtres qui calculent une moyenne utilisent en général des voisinages carrés {(3*3), (7*7), …}. Plus le filtre grossit plus le lissage devient important et plus le flou s'accentue. Pour un filtre de taille 3*3 la matrice de voisinage correspondante est la suivante :

$$H = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Pour des filtres de taille n*n, la matrice est de la forme :

$$\frac{1}{n^2} \times \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$$

Voici ci-dessous le résultat de filtre moyenneur avec trois tailles :

Image source moy 3*3 moy 5*5 moy 7*7

Figure 2 : Filtre moyenneur avec différent taille du noyau.

1.2) Filtre Médian

Le filtre médian sélectionne la valeur médiane des pixels du voisinage. Pour un voisinage carré 5*5, par exemple, les valeurs des 25 voisins sont ordonnées par ordre croissant, et la 13-ème valeur est gardée comme résultat.

Image source med 3*3 med 5*5 med 7*7

Figure 3 : Filtre médian avec différent taille du noyau.

1.3) Filtre Gaussien

Dans le traitement d'image on introduit une fonction gaussienne a deux dimensions G (x, y) :

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{(x^2+y^2)}{2\sigma^2}}$$

Le paramètre sigma s'appelle la déviation standard, et détermine la largeur de la couche Gaussienne. Si sigma < 1 le filtre est utilisé pour réduire le bruit, et si sigma > 1 c'est dans le but de fabriquée une image qu'on va utiliser pour faire un « un masque flou » personnalisé. Alors notons que plus le sigma est grand, plus la cloche Gaussienne est large et plus le flou appliqué à l'image sera marqué.

Image source Gauss 3*3 Gauss 5*5 Gauss 5*5

Figure 4 : Filtre Gaussien avec différent taille du noyau.

2) Traitement d'image

Les filtres existent à ce niveau-là met en évidence les variations de luminance qui caractérisent les contours des objets d'une image.

Cette mise en évidence peut servir à détecter les contours des objets présents dans une image, à différencier des zones de l'image, à faire la segmentation d'image et à extraire des informations souvent pertinentes pour caractériser l'image.

Dans ce mémoire nous avons traité deux catégories de méthodes pour segmenter une image:

- La première concerne la segmentation par contours : on se basant par exemple sur le calcul du gradient de l'image par l'opérateur Prewitt, Sobel ou Roberts, calcul de Laplacien, ou l'application d'opérateur Canny, etc.
- La deuxième concerne la segmentation par région : c.-à-d la décomposition d'une image par la recherche de zones possédant des attributs communs, soit de la luminosité, soit de texture, … nous allons traiter ici le K-means.

2.1) Filtre Prewitt

Les filtres de Prewitt sont de la forme suivante :

$$H_0 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \; H_{90} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Le résultat de leur application sur une image en niveau de gris est donné sur la figure suivante, pour les directions horizontales et verticales. Les intensités ont été normalisées entre 0 et 255, le gris correspond à un gradient nul.

Figure 5 : De gauche à droite : Image source, image après le filtre Prewitt.

2.2) Filtre Sobel

C'est un opérateur plus simple qui donne toutefois des résultats corrects, est de la forme suivante :

$$\mathbf{G_x} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \mathbf{A} \quad \text{et} \quad \mathbf{G_y} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * \mathbf{A}$$

A est l'image source, Gx et Gy deux images qui en chaque point contiennent des approximations respectivement de la dérivée horizontale et verticale de chaque point.

En chaque point, les approximations des gradients horizontaux et verticaux peuvent être combinées comme suit pour obtenir une approximation de la norme du gradient :

$$\mathbf{G} = \sqrt{\mathbf{G_x}^2 + \mathbf{G_y}^2}$$

Le résultat de l'application du ce filtre par une image en niveau de gris est donné sur la figure suivante :

Figure 6 : De gauche à droite : Image source, image après le filtre sobel

2.3) Filtre Roberts

L'opérateur Roberts est l'un des opérateurs qui utilise des opérateurs de différence entre deux pixels adjacents dans la direction diagonale pour approximer l'amplitude du gradient afin de détecter les contours.

La forme du Roberts est :

$$G_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}; \quad G_y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

L'obtention de l'approximation de la norme du gradient Roberts est la même que celle du Sobel et Prewitt.

Le résultat de l'application du ce filtre par une image en niveau de gris est donné sur la figure suivante

Figure 7 : De gauche à droite: Image source, Image après l'opérateur Roberts

2.4) Filtre Laplacien

L'opérateur Laplacien est un opérateur qui nécessite un seul masque, contrairement aux opérateurs du gradient qui nécessitent deux masques, mais les informations d'orientation des bords sont perdues. Son noyau est de la forme suivante :

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Le résultat d'application de ce filtre sur une image est le suivant :

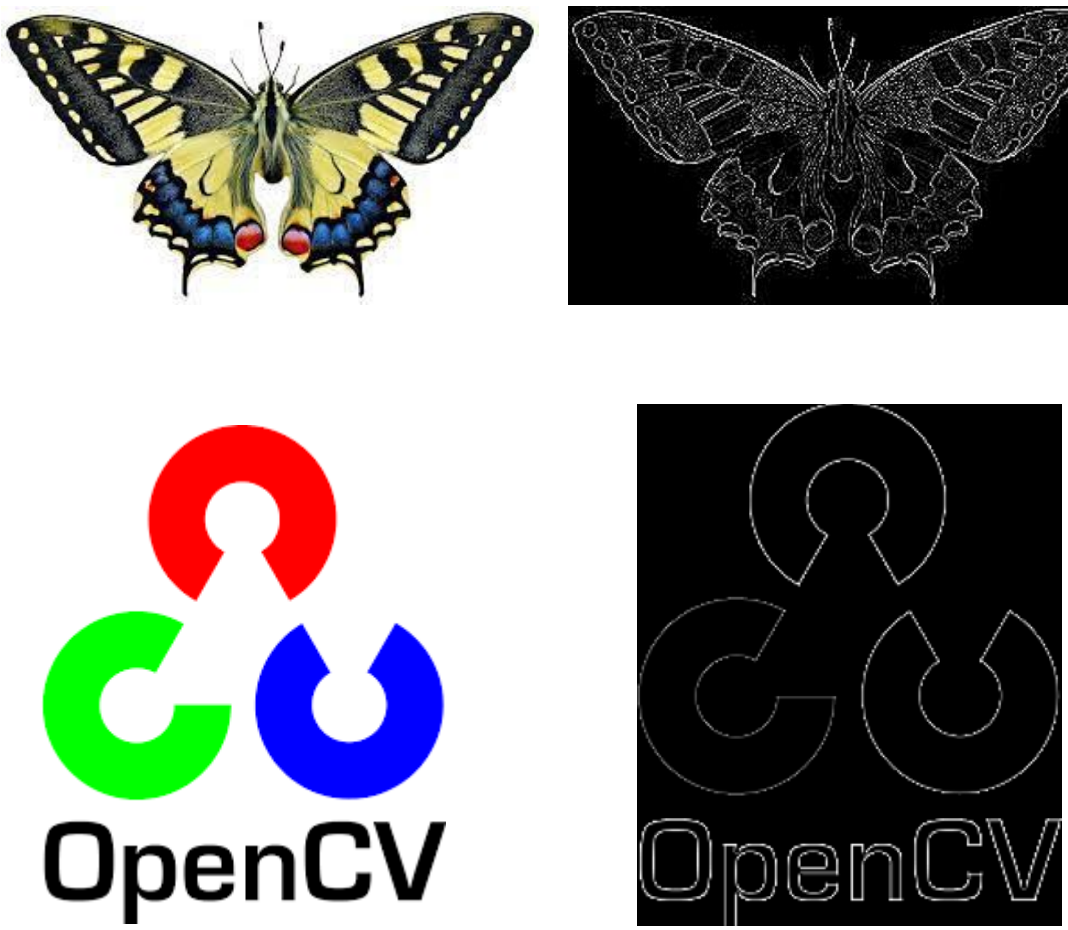


Figure 8 : De gauche à droite : Image source, image après le filtre Laplacien

2.5) Filtre Canny

Est un algorithme de détection de contour populaire astucieux () pour trouver les contours de l'image, il renvoie l'image détectée par les bords en niveaux de gris.

Voici une implémentation de ce filtre par l'utilisation de la fonction prédéfinis ''Canny ()'' existe dans la bibliothèque cv2 :

Figure 9 : De gauche à droite : Image source, image après le filtre Canny

2.6) K-means

Le k-means est un algorithme de machine Learning non-supervisé de clustering, facile à comprendre et à utiliser. Est une technique permet de regrouper en K clusters distincts des ensembles de données volumineux, où K doit être pré-spécifié par l'utilisateur. Ainsi les données similaires se retrouveront dans un même cluster. Par ailleurs, une donnée dit aussi une observation ne peut se retrouver que dans un cluster à la fois.

Il peut exploiter le K-means lors de la segmentation d'image pour détecter des régions présentant des caractéristiques d'homogénéité, en prédéterminant le nombre K : nombre de zones similaires qui peuvent être distinguées par l'œil nu.

Parfois, le mal choix du k peut conduire à un partitionnement trop fragmenté dans l'image résultat, ou peut conduire à savoir potentiellement des clusters trop généralistes. Dans ce cas, des moyens plus précis doivent être utilisés pour déterminer le K optimal.

L'application de cet algorithme par les images suivantes passées du filtre médian donne :

Figure 10 : De gauche à droite : Image source, image après segmentation par le k-means.

3) Post-Traitement d'image

Le post traitement accompli tout le travail et toutes les retouches qui sont fait sur une image après le prétraitement et le traitement d'image, citons par exemple la modification du cadrage des objets inclus dans l'image, la suppression des aspérités, la saturation des couleurs, la balance des blancs, la comblassions des trous, enlevassions des défauts sur l'image par exemple une taches d'une poussière. etc.

Toutes ces fonctionnalités sont obtenues en utilisant ce qu'on appelle la morphologie mathématique. Cette dernière contient un ensemble d'opérateurs qui travaillent sur les voisinages locaux de chaque pixel. La forme de ce voisinage est appelée élément structurant. Comme pour les filtres de prétraitement, on peut utiliser des éléments structurant de taille et de forme variées. Les éléments structurants couramment utilisés sont le disque et le carré qui permet d'accélérer les calculs.

Dans ce qui suit nous verrons quelques opérateurs morphologiques,

3.1) Erosion

L'érosion est un opérateur consiste à chercher tous les pixels pour lesquels l'élément structurant centré sur ce pixel touche l'extérieur de la structure. Le résultat est une structure rognée (figure 11). On remarque la suppression des pixels de la limite de l'objet « j »

Figure 11 : De gauche à droite : Image source, image après l'Erosion

Cet opérateur permet aussi de supprimer quelque bruit blanc, voilà un exemple :

Figure 12: De gauche à droite : Image source, image après la suppression des points blancs par l'Erosion.

3.2) Dilatation

La dilatation est un opérateur consiste à déplacer l'élément structurant sur chaque pixel de l'image, et regarder si les éléments structurant intersecté la structure d'intérêt. Le résultat est une structure qui plus grosse que la structure d'origine (figure 13). On remarque l'augmentation de la zone blanche.

Figure 13: De gauche à droite : Image source, image après la Dilatation

Cet opérateur permet de supprimer des points noirs, voilà un exemple :

Figure 14: De gauche à droite : Image source, image après la suppression des points noirs par Dilatation.

3.3) Ouverture

C'est une érosion suivie de la dilatation, utile pour éliminer le bruit (éliminer les points blancs).

Figure 15:De gauche à droite, Image source, image après l'ouverture

3.4) Clôture

C'est la dilatation suivie de l'érosion, utile pour fermer les petits trous a l'intérieur des objets ou de petits points noirs sur l'objet.

Figure 16: De gauche à droite: Image source, image après le Clôture.

3.5) Gradient morphologique

C’est la différence entre la dilatation et l’érosion d’une image.

Figure 17: De gauche à droite: Image source, image après le Gradient morphologique

3.6) Tophat

C’est la différence entre l’image d’entrée et l’ouverture.

Figure 18: De gauche à droite: Image source, image après Top hat

3.7) Black-hat

C’est la différence entre l’image d’entrée et sa clôture.

Figure 19: De gauche à droite: Image source, image après Black-hat

III. Chaîne de traitement d’image

La chaîne de traitement d’image est de la forme suivante :

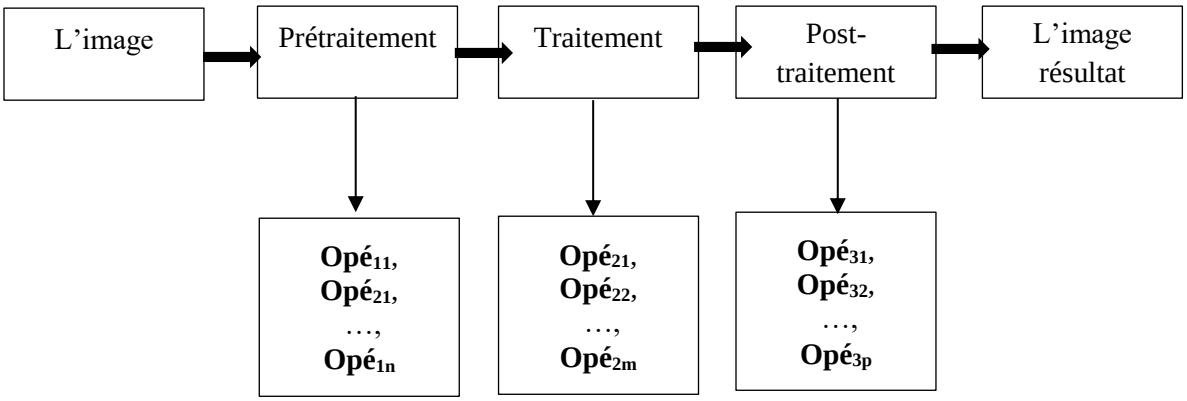


Figure 20:chaîne de traitement d'image

Elle consiste à choisir un opérateur dans chaque phase, alors la combinaison des opérateurs de manière générale est la suivante :

Combinaison = (Opé$_{1i}$, Opé$_{2j}$, Opé$_{3k}$) où i∈[1,..,n] , j∈ $[1, \dots, m]$ ***et*** $k \in [1, \dots, p]$

Figure 21:Formule de combinaison d'opérateurs .

Notons qu'avant d'appliquer un filtre appartient au phase prétraitement, il est nécessaire de convertir l'image en RGB, puis en GRAY , le résultat de ce filtre est considéré comme l'entré des filtres appartient au phase traitement, et ainsi de suite, afin d'aboutir l'image résultat.

Prenons par exemple les deux combinasons :

- (median, canny, dilatation)
- (gaussien, canny, dilatation)

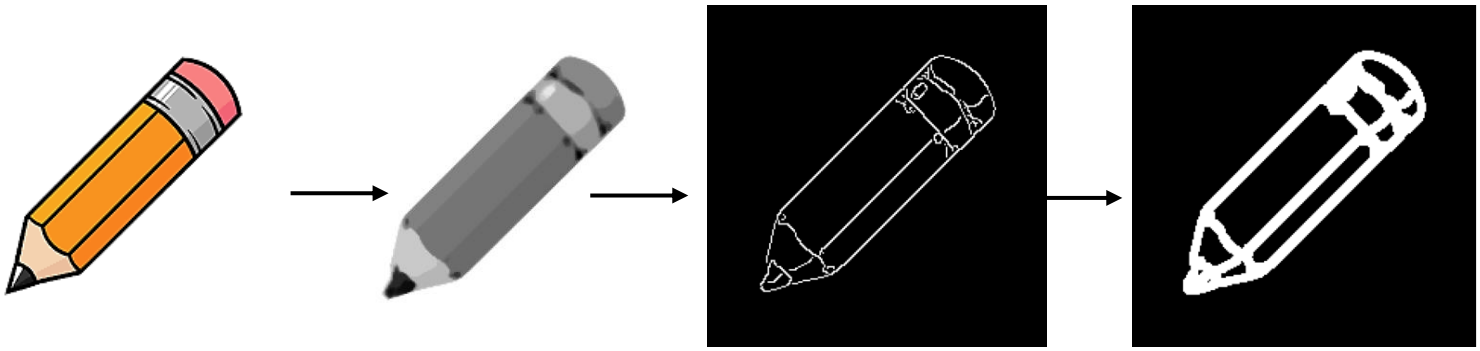


Image initial image en MedianBlur après canny après dilatation

Figure 23:Exemple1: combinaison (median, canny, dilatation).

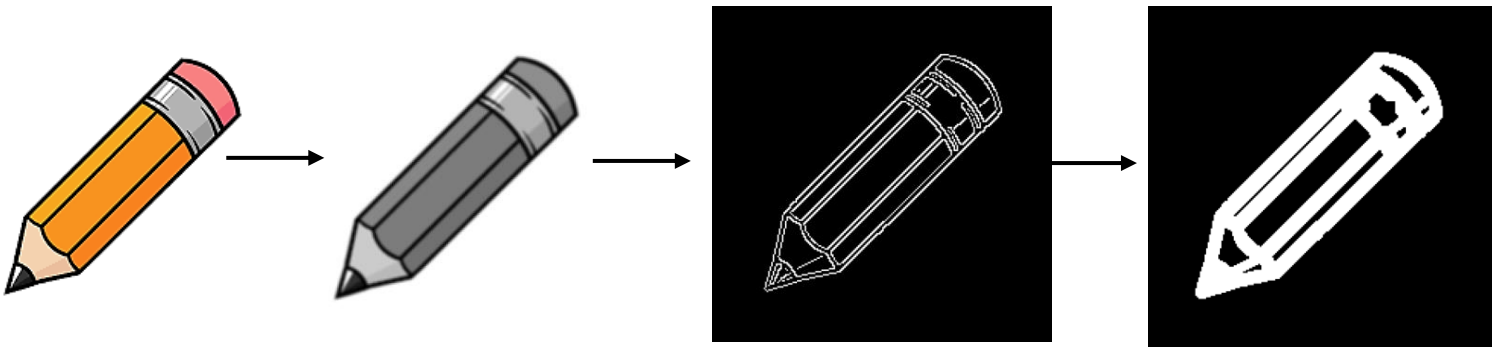


Image initial image en GausssBlur après canny après dilatation

Figure 22:Exemple2: combinaison (gaussien, canny, dilatation).

On remarque bien qu'un changement de filtre affecte sûrement sur le résultat final, de sorte qu'il est nécessaire de trouver des solutions efficaces qui permettent de préciser la bonne combinaison pour une tâche donnée .

Chapitre 2 : solution proposée

I. Introduction

L'apprentissage automatique (Machine Learning) est une technologie de l'intelligence artificielle qui permet aux ordinateurs d'apprendre, à partir de données, sans avoir été programmés explicitement à cet effet.

Le machine Learning est composé de plusieurs types d'apprentissage qui sont :

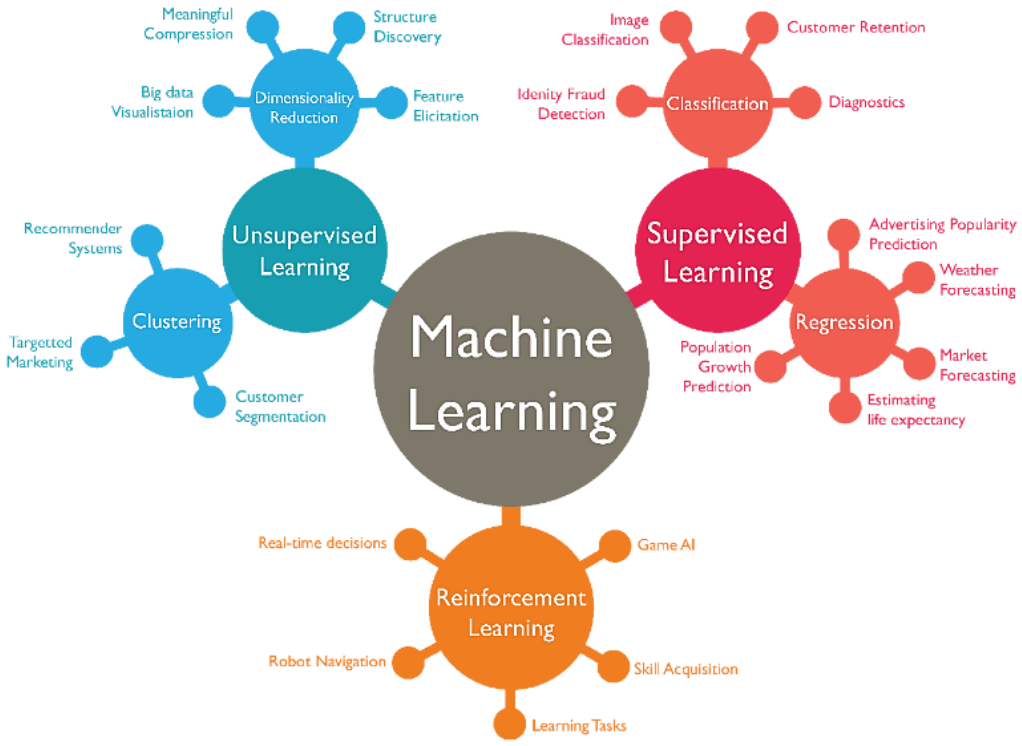


Figure 24: Différents types d'apprentissage

- **L'apprentissage supervisé :** se fait sur la base d'une vérité, i-e, nous avons une connaissance préalable de ce que devrait être les valeurs de sortie de nos échantillons. Donc l'objectif est d'apprendre une fonction qui, à partir d'un échantillon de données et des résultats souhaités, se rapproche le mieux de la relation entre l'entré et la sortie observable dans les données.

  Les algorithmes qu'on peut faire par l'apprentissage supervisé on trouve la régression linéaire, régression non linéaire, forêt aléatoire, arbre de décision, …

L'apprentissage non supervisé : c'est quand le système ne dispose pas d'exemple, et que le nombre de classes et leur nature n'ont pas été prédéterminés. L'algorithme doit découvrir par lui-même la structure en fonction des données.

Exemples des algorithmes non supervisé l'algorithme K-means, réduction de la dimensionnalité, classification hiérarchique, modèles de distribution, …

- **L'apprentissage par renforcement :** correspond au cas où l'algorithme apprend un comportement étant donnée une observation. L'action de l'algorithme sur l'environnement produit une valeur de retour qui guide l'algorithme d'apprentissage.

Dans ce qui suit on s'intéresse par ce dernier, puis faire une présentation détailler du problématique du sujet, en outre nous montrerons la solution proposée.

II. Apprentissage par renforcement :

L'apprentissage par renforcement est utilisé pour apprendre à un agent comment se comporter dans un environnement afin de maximiser sa performance.

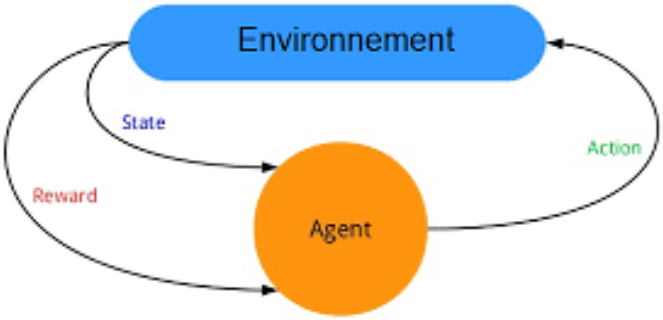


Figure 25: Schéma d'apprentissage par renforcement

Exemple : On considère les bébés comme des agents posés dans un environnement (le monde), dans cet environnement ils prendront des actions et le monde va répondre avec une certaine manière, soit avec une punition si l'action est aboutie à un échec, soit avec une récompense si les actions qu'on prenez nous donne un bon résultat et finalement ils vont essayer de prendre plus de ce genre d'action.

Donc, dans le RL c'est la même chose, un bébé qui prend à marcher, il va essayer plusieurs reprises, des fois il va tomber mais finalement après des essais et des erreurs il va réussir de trouver un comportement qui lui permettre d'optimiser ces récompenses dans le monde.

1) Exploration & Exploitation

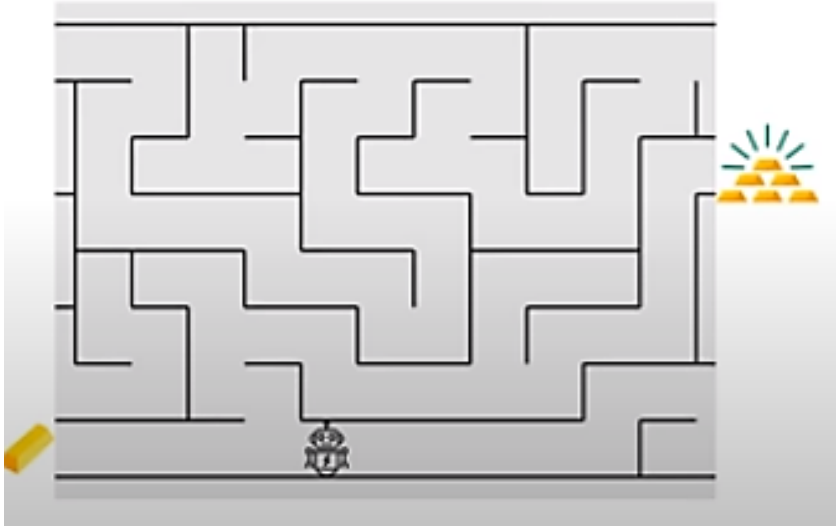


Figure 26 : Labyrinthe d'un agent

La figure ci-dessus représente un agent plongé dans un labyrinthe ayant deux chemins, l'un fait gagne un euro et l'autre fait gagne 10 euros.

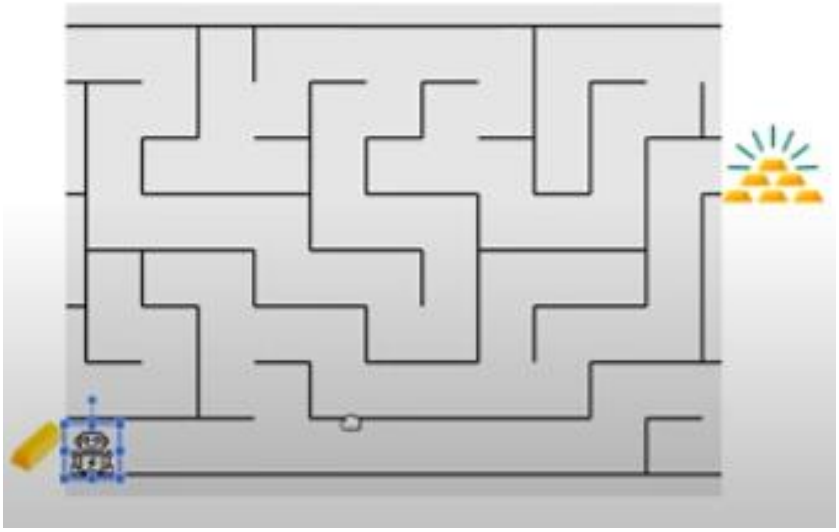


Figure 27: Premier exploration du labyrinthe

En faisant l'exploration, l'agent dit « j'ai trouvé une solution qui m'a fait gagner un euro, donc à chaque fois je parcourrai le même chemin » et c'est l'exploitation. Mais, il faut maximiser les récompenses, pour cela, il faut augmenter la partie d'exploration et diminuer la partie d'exploitation, et au fur et à mesure inverser la part. la figure ci-dessous montre le résultat d'augmentation de la partie d'exploration.

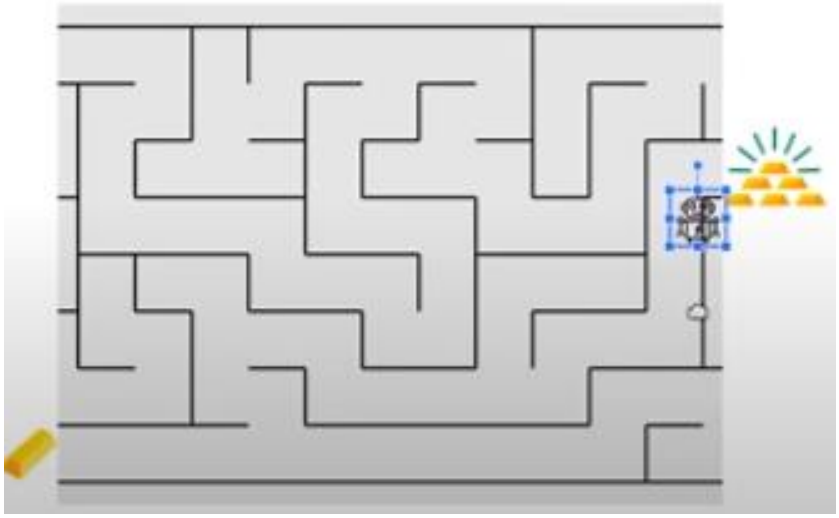


Figure 28: Deuxième exploration du labyrinthe

2) Q-learning

Le Q-learning fait partie des méthodes d'apprentissage par renforcement sans modèle. Il s'agit d'apprendre par l'expérience les actions à effectuer en fonction de l'état actuel.

On associe à chaque paire (**s**, **a**) (où l'action **a** est exécutée depuis l'état **s**), une grandeur Q par laquelle on veut apprendre l'intérêt de l'action **a** quand on est dans l'état **s**. Q (**s, a**) est l'espérance de gain, c'est-à-dire la somme :

- De l'espérance de gain immédiat dû à l'action choisie,
- Et des espérances de gain de tous les états, pondérées par la probabilité d'atteindre chacun d'eux depuis l'état **s** en menant l'action **a**.

À chaque arrivée sur un état avec une telle action, on a besoin d'une formule qui permet de mettre à jour Q (**s, a**) en fonction de sa valeur précédente et de l'expérience qui vient d'être vécue.

D’où l’algorithme :

Initialiser Q (**s**, **a**) aléatoirement

Répéter (Pour chaque épisode) :

Initialiser **s**

Répéter (pour chaque étape d’épisode) :

Choisir une action **a** depuis **s** en utilisant la politique spécifiée par Q

Exécuter l’action **a**

Observer la récompense **r** et nouvel état **s’**

Q[s, a] = Q[s, a] + α *(r + γ * max (Q [s’, a’]) – Q[s, a])

s = s’

Jusqu’à (**s** est terminer)

Avec :

- **s :** l’état actuel
- **a :** l’action actuelle
- **α :** taux d’apprentissage, peut être défini comme le degré d’acceptation de la nouvelle valeur par rapport à l’ancienne. Ci-dessus, nous prenons la différence entre la nouvelle et l’ancienne valeur, puis nous multiplions cette valeur par le taux d’apprentissage. Cette valeur est ensuite ajoutée à notre valeur q précédente, ce qui la fait évoluer dans la direction de notre dernière mise à jour.

- **γ :** facteur d’actualisation, il est utilisé pour équilibrer la récompense immédiate et future. Dans notre règle de mise à jour ci-dessus, vous pouvez voir que nous appliquons la décote à la récompense future. En général, cette valeur peut varier entre 0.8 et 0.99.
- **r :** (reward) la récompense après avoir effectué une certaine action à un état donné. Une récompense peut survenir à n’importe quel pas de temps donné ou seulement au pas de temps terminal.

III. Solution

Pour accomplir une tache de vision, l'utilisateur doit avoir connaissance approfondie des opérateurs à appliquer. En effet, le résultat d'un traitement d'image dépend de l'opérateur appliqué, les utilisateurs inexpérimentés devraient essayer tous les combinaisons d'opérateurs possibles pour trouver celle qui donne le meilleur résultat.

Dans ce cas, le processus manuel n'est pas facile. D'où la résolution de ce problème se préoccupe à utiliser l'algorithme Q-learning qui génère à partir d'une liste d'opérateurs toute chaine opérationnelle à utiliser afin d'atteindre l'objectif.

Afin de faciliter l'utilisation de l'algorithme, nous optons de réaliser une interface utilisateur dont sa conception est comme suivie :

1) Conception de l'interface utilisateur

Dans cette partie de conception de l'interface utilisateurs, nous nous sommes focalisées sur le processus de Modélisation UML avec une définition d'UML et reproduire les besoins de notre projet sous forme de différent diagramme d'UML qui sont suivent :

- Diagramme de cas d'utilisation.
- Diagramme de séquences.

L'UML signifie le langage de modélisation Unifié ; c'est un langage visuel constitué d'un ensemble des diagrammes qui permet de modéliser un problème de façon standard, représenter le logiciel ç développer, … etc. ce n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML.

- **Diagramme de cas d'utilisation**

Les cas d'utilisation permettent de représenter le fonctionnement du système vis-à-vis de l'utilisateur : c'est donc une vue du système dans son environnement extérieur.

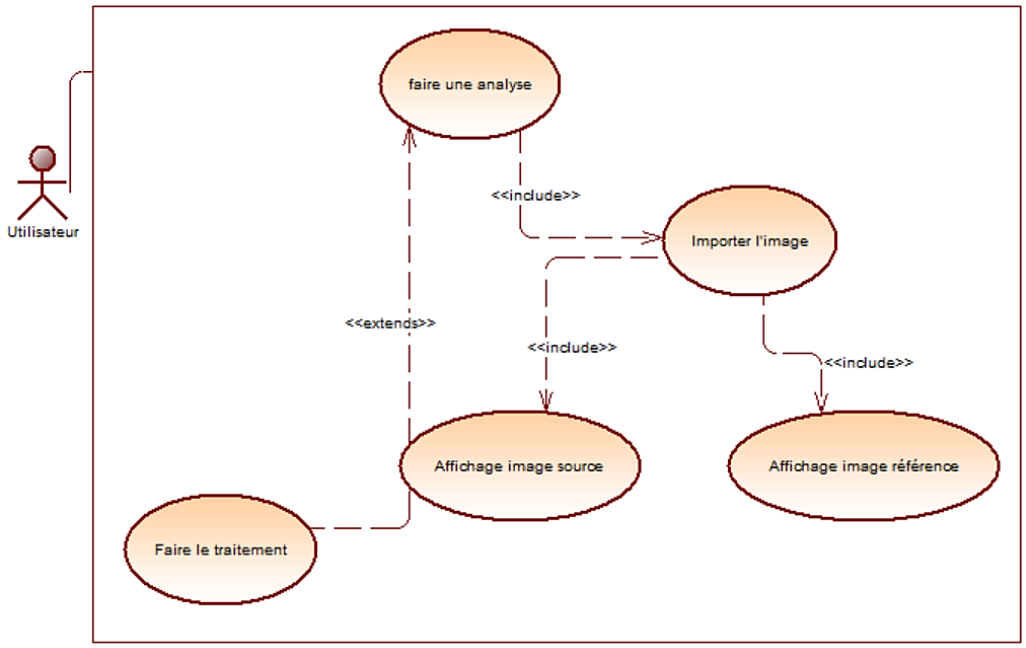


Figure 29: Diagramme de cas d'utilisation d'un utilisateur avec l'interface

- **Diagramme de séquence**

Le diagramme de séquence permet de décrire les différents scénarios d'utilisation du système.

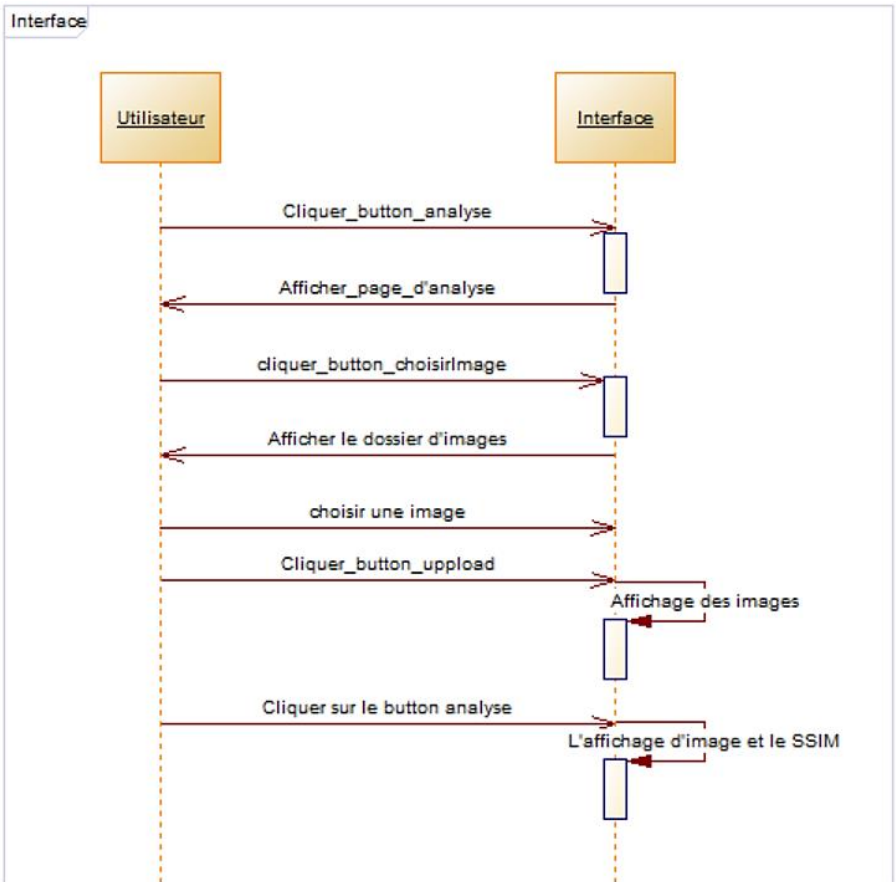


Figure 30: Diagramme de séquence d'un utilisateur avec l'interface

Chapitre 3 : expérimentation et discutions

I. Introduction

Après avoir élaboré la conception de notre application, nous aborderons dans ce chapitre d'expliquer de façon détailler les phases de sa réalisation. Ces phases sont considérées comme étant la concrétisation finale de toutes les méthodes de la conception.

Nous décrivons d'abord les ressources logicielles, ainsi les package python utilisées lors du développement du projet, ainsi nous présentons une interprétation de l'application développée.

II. Outils utilisés

- Ressources logicielles :

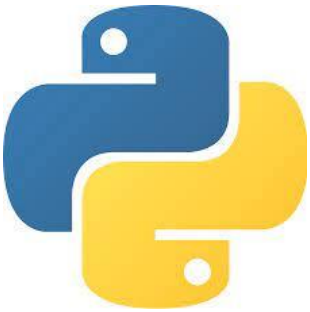


Figure 31: logo Python.

Python est un langage de programmation structuré et orienté objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions. Il conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Figure 32: Logo Flask

Flask est un micro framework open-source de développement web en Python. Il est classé comme microframework car il est très léger.

Figure 33: Visual Studio Code.

Visual Studio Code est éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOs. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la compétition intelligence du code, etc.

Figure 34: Logo Google Colaboratory.

Google colaboratory est un service cloud, offert par Google (gratuit), permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais de navigateur. C'est un environnement particulièrement adapté au machine Learning, à l'analyse de données et à l'éducation.

Figure 35:PowerAMC

PowerAMC est un logiciel de conception créé par la société SAP, qui permet de modéliser les traitements informatiques et leurs bases de données associées.

- Packages python utilisées :

Figure 36:Logo OpenCv.

OpenCv (Open Computer Vision) **:** est une bibliothèque graphique libre développée par « intel », spécialisée dans le traitement d'image. Elle propose la plupart des opérations classique en traitement d'image tel que : la lecture, l'affichage, la segmentation et la morphologie mathématique d'image.

Figure 37:Logo Numpy

Numpy : est une bibliothèque destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

Figure 38:Logo Scikits-image.

Scikit-image : est un ensemble d'algorithme pour le traitement d'image et la vision par l'ordinateur. Par exemple :

- **Skimage.metrics** : metrics correspond aux images par exemple métrique de distance, similarité, etc.
- **Skimage.metrics.structural_similarity :** calculer l'indice de similarité structurelle moyen entre deux images.

Random : choisit un élément au hasard dans une liste.

III. Mise en pratique l'application du Q-learning et traitement d'image

Dans cette partie nous implémenterons l'algorithme de Q-learning dont le but sera d'apprendre à un agent à choisir la meilleure combinaison d'opérateurs qui atteint l'objectif.

L'objectif de notre application est d'approcher une image source, passant par les trois phases de traitement d'images, à une image référence éditée par un expert.

La première chose dont nous avons besoin dans le cadre de notre projet est de créer un environnement avec lequel notre agent va interagir. On considère que cet environnement soit une image.

En fonction de l'environnement dans lequel un agent évolue, il existe trois tâches distincts : Le choix aléatoire d'un filtre par convolution, puis celle de la segmentation et enfin un pour le filtre par morphologie mathématique. La combinaison de ces trois tâches nous définit l'action, il y a donc deux états possibles, l'une fait référence à l'image originale, l'autre est l'image résultat.

Dans cet environnement, notre agent doit choisir à chaque épisode une combinaison parmi 147 combinaisons possibles, en effet, nous le définissons dans :

- La première phase : trois opérateurs à choisir,

```
list_filres  = [median, gaussien, moyenneur ]
```

- La deuxième phase : sept opérateurs à choisir

```
list_segmentation = [canny, sobel, perwit, laplacien, findContours, kmeans,roberts]
```

- Et la troisième phase : sept opérateurs à choisir

```
list_morphologie   = [dilatation, erosion, ouverture, TopHat, cloture, gradient, BlackHat]
```

Le programme présenté, implémente un algorithme de Q-learning. Ce dernier a pour objectif de déterminer une stratégie optimale, qui permettra à l'agent de choisir les actions qui le mèneront à choisir la meilleure action quel que soit l'état initial (l'image originale) où il se trouve.

La deuxième chose consiste à introduire à notre algorithme de Q-learning la Q-table. Il s'agit d'une matrice qui stocke, pour chaque état, une valeur correspondant à la somme attendue des récompenses que chaque action nous reporterait. Dans un premier temps, elle est initialisée à zéro puisque nous n'avons aucune connaissance de notre environnement, avec une ligne par état et une colonne par action.

En outre, nous allons entrainer notre agent à évoluer 100 fois. Il est également nécessaire de choisir un taux d'apprentissage (Learning rate) $\boldsymbol{\alpha}$ (0.1) et un facteur d'actualisation $\boldsymbol{\gamma}$ (0.9), qui nous servira pour la mise à jour itérative de la table Q.

Débute alors la boucle d'apprentissage par renforcement qui va, pour chaque épisode, réinitialiser l'environnement et le total des récompenses obtenues, puis faire évoluer notre exploration, alors, notre agent commence par choisir une action, puis l'applique à l'environnement avant de mettre à jour la Q-table grâce aux informations obtenues.

Par ailleurs, puisque l'agent est à l'état finale, c'est-à-dire, lorsque on obtient l'image résultat, on la compare avec une image référence éditée par expert par le calcul de l'indice de similarité entre les deux, s'il est supérieur à 80%, alors l'agent doit est récompensé par 1, sinon par 0.

L'un des concepts centraux du Q-Learning est l'équilibre à trouver entre l'exploration et l'exploitation tout au long de l'entrainement du modèle. En effet, il est nécessaire que notre agent continue à explorer son environnement tout en utilisant l'intelligence accumulée dans la Q-table à mesure des épisodes, autrement dit, pour optimiser son résultat l'agent doit augmenter l'exploration, diminuer l'exploitation et au fur et à mesure inverser la part.

La figure ci-dessous résume la démarche d'application du Q-learning et traitement d'image :

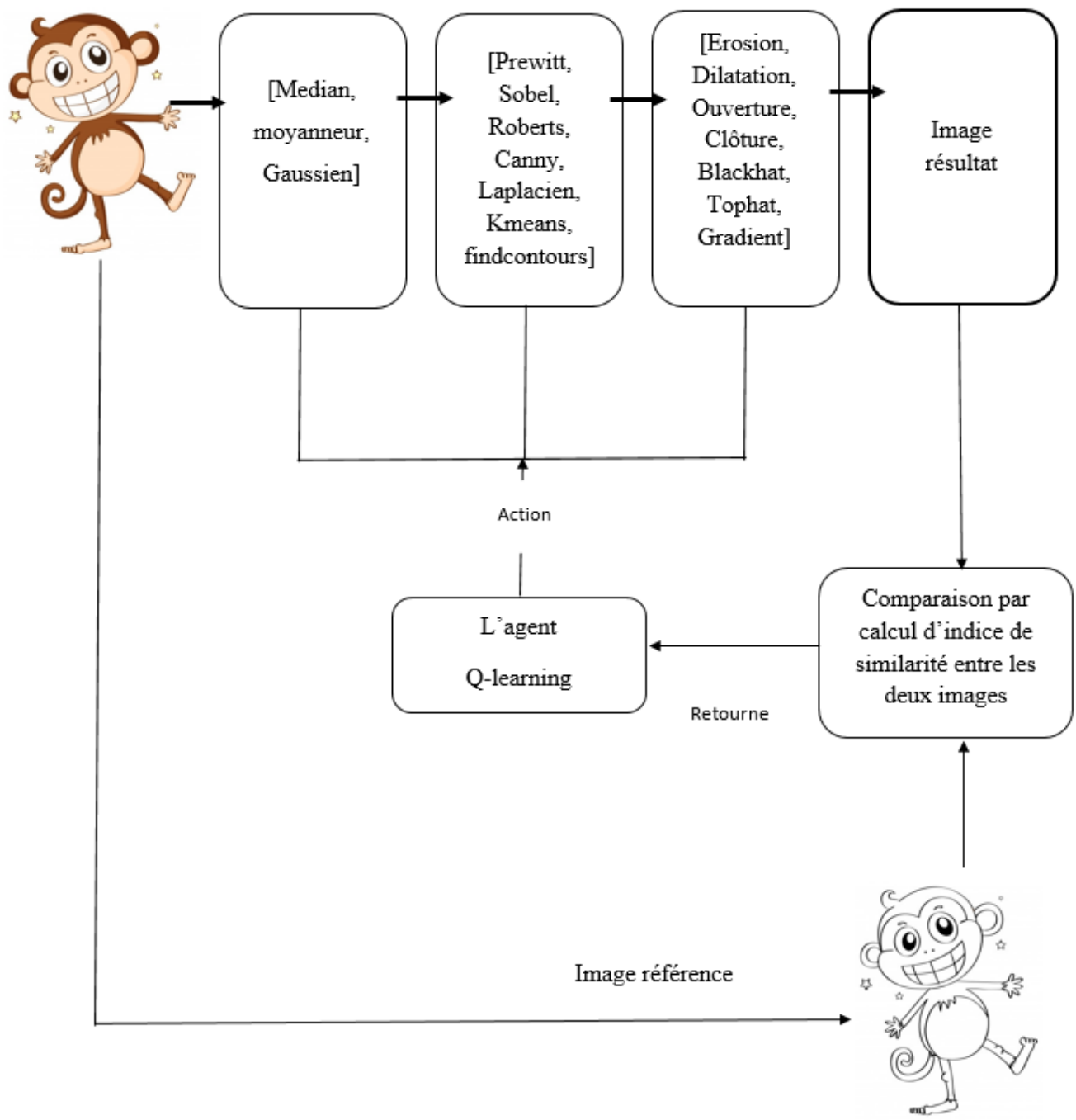


Figure 39: récapitulatif sur la démarche d'application du Q-learning et le traitement d'image

- Appliquant tout d'abord l'algorithme sur l'image « du singe » ci-dessus :

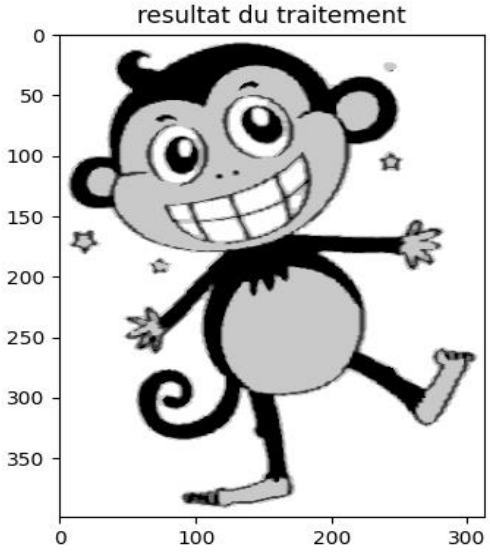


Figure 40: résultat de traitement

Le résultat du traitement donne comme meilleure combinaison : (Gaussien, Kmeans, Ouverture), dont l'indice de similarité entre l'image résultat et l'image référence est supérieure à 77%.

En l'appliquant sur d'autres images « texturées » :

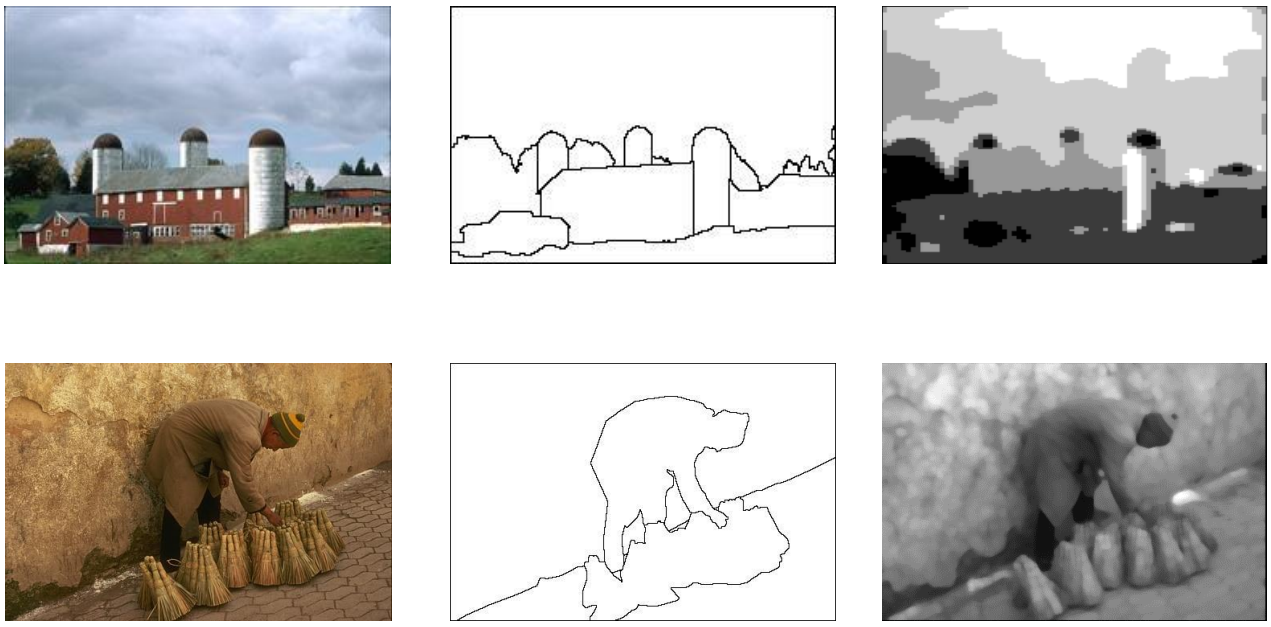


Figure 41: De gauche à droite : image source, image segmentée par un expert, image résultat

D’après la vision du résultat après le traitement, on remarque qu’il y a une différence importante entre l’image résultat et l’image référence, cette différence dû au mal choix du paramètre des opérateurs. Pour améliorer la qualité du résultat, il est nécessaire de faire une étude approfondie sur les paramètres des opérateurs.

IV. Interface utilisateur

Dans ce qui suit, nous présenterons notre interface graphique, qui contient deux espaces, l’une c’est juste une page d’accueil où en afficher le nom de notre sujet (figure 42), l’autre pour l’utilisation de l’algorithme développé (figure 43).

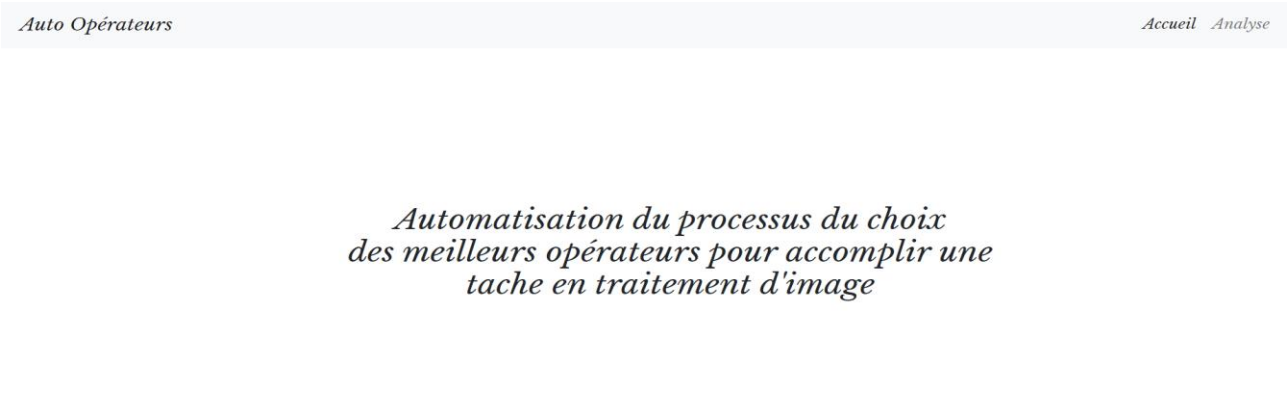


Figure 42: Interface : Page d'accueil

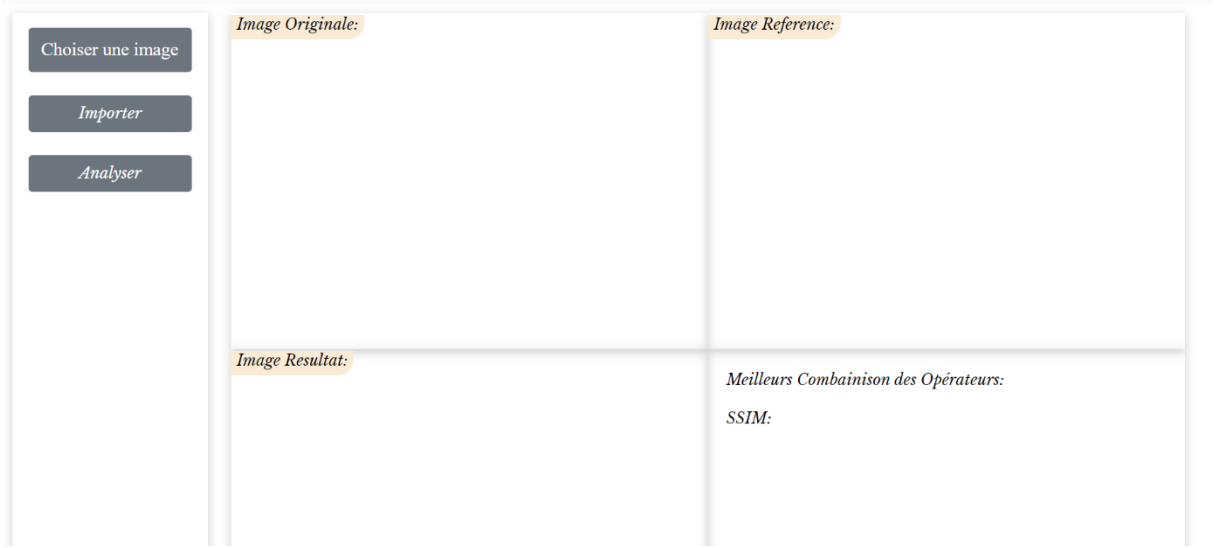


Figure 43: Page d'analyse

La figure ci-dessous montre la démarche d'utilisation de l'interface, à savoir si l'utilisateur est importé l'image, après l'avoir choisi, elle est montrée et accompagnée par sa référence.

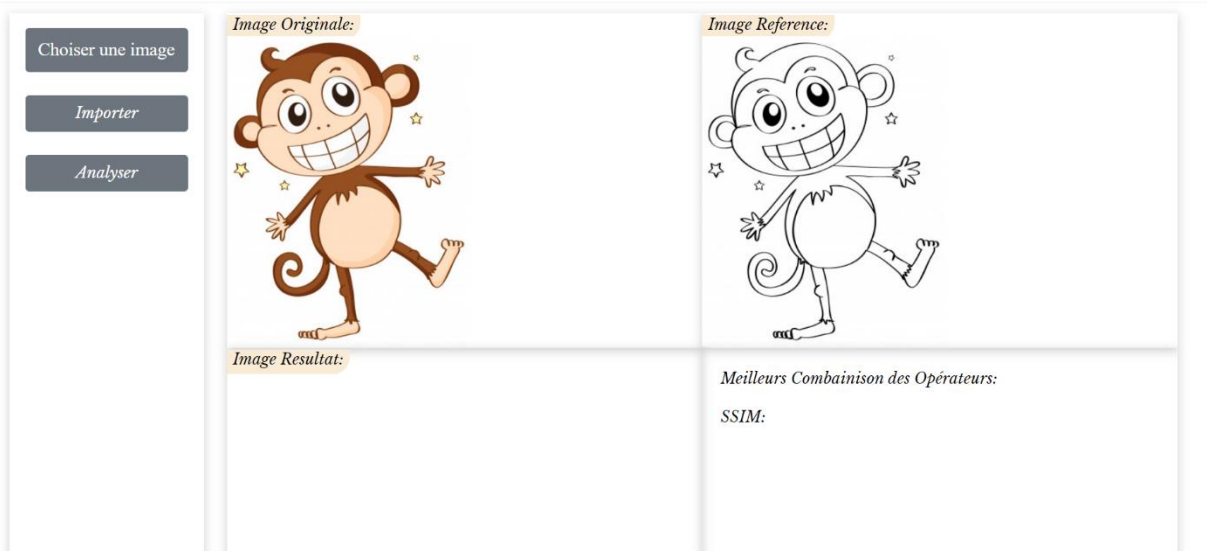


Figure 44: Page d'accueil après l'importation des images

L'utilisateur peut utiliser notre algorithme en appuyant sur le bouton analyser. Comme l'image le montre en dessous.

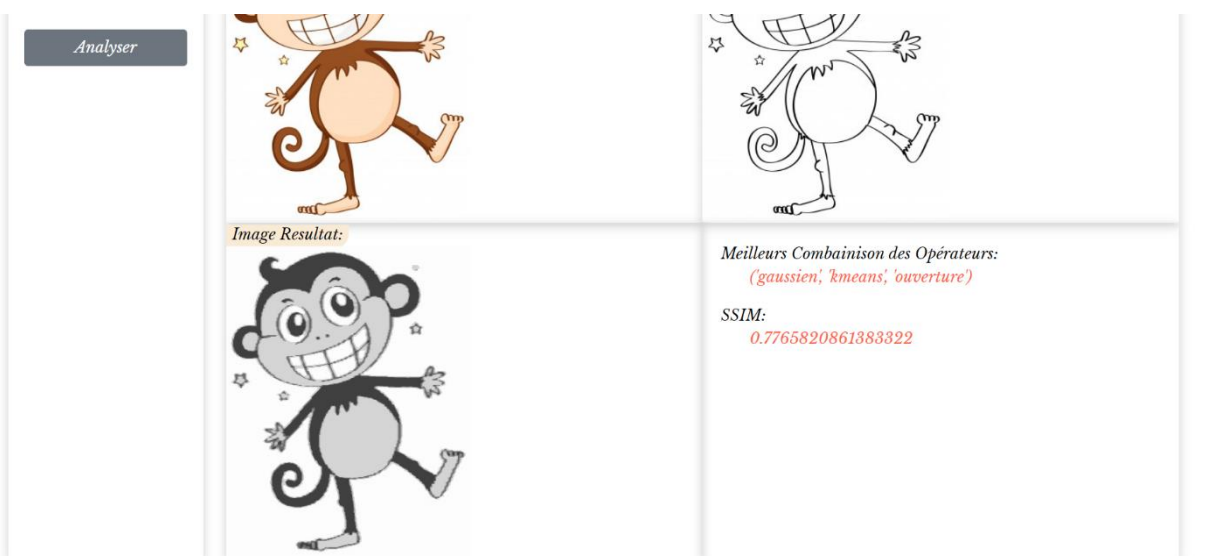


Figure 45: Page d'analyse après le traitement

Alors le traitement est effectué, et Au bout de quelque temps son résultat va indiquer dans sa zone assignée, en bas à droite (figure 45).

Conclusion générale

Au cours de la période de réalisation de ce projet, on a eu l'opportunité de mettre en œuvre différentes connaissances acquises durant nos études à la faculté des sciences semlalia de Marrakech et acquérir de nouveaux outils de développement.

Notre travail s'est fixé comme objectifs de satisfaire le maximum des besoins des informaticiens intéressant au domaine du traitement d'image. Nous avons réalisé un algorithme Q-learning pour le meilleure choix d'opérateurs pour une application de traitement d'image.

Cet algorithme facilitera la tâche à savoir la détermination de combinaison d'opérateurs, à travers le choix efficace d'un opérateur dans les trois phases de système de base de traitement d'image, en se basant sur l'apprentissage par renforcement en particulier l'algorithme Q-learning, qui se pose sur l'exploration et l'exploitation. Cependant le résultat du traitement peut être améliorer si nous enrichissons l'algorithme par une approche d'étude sur les paramètres des opérateurs.

La réalisation de ce projet a été très bénéfique, côté programmation, ainsi en ce qui concerne la communication, le travail en équipe, etc. C'est pour cela, nous aimerons nous orientés vers la programmation car c'est un domaine que nous avons trouvé très intéressant et que nous voudrions approfondir nos consciences

Références

[1] https://youtu.be/PKNxUF9CGn8 consulté le 16-05-2021

[2] https://en.wikipedia.org/wiki/Q-learning consulté le 20-05-2021

[3] http://webia.lip6.fr/~thomen/Teaching/BIMA/cours/segmentation.pdf consulté le 05-06-2021

[4] https://www.math.univ-paris13.fr/~chaussar/Teaching/2016-2017/Master2/TP/TP1/TP1%20-%20Prise%20en%20main.pdf consulté le 07-06-2021

[5] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/ consulté le 15-06-2021

[6] https://www.lama.univ-savoie.fr/mediawiki/index.php/Segmentation_d%27image_par_d%C3%A9tection_de_contours_et_algorithme_%22ligne_de_partage_des_eaux%22 consulté le 23-06-2021

[7] https://youtu.be/IhbAwAgv1dU consulté le 26-06-2021

[8] https://www.youtube.com/watch?v=4iDryb7S8CM consulté le 10-07-2021