

**République Algérienne Démocratique et Populaire**  
**Ministère de l'enseignement Supérieur et de la Recherche Scientifique**  
**Université Benyoucef Benkhedda Alger 1**



**Faculté de Sciences**

Département Informatique

---

**RAPPORT de projet TP**

**Module : Traitement d'images numériques**

Système de comptage des pièces de monnaies.

---

**Réalisé par :**

Moulay Abdellah Asma **GRP3**

Belabbas Rania **GRP3**

Abad Lysa Manal **GRP2**

**Année universitaire : 2024/2025**

# Table des matières

<b>Introduction Générale</b>	<b>1</b>
<b>1 Prétraitement des images</b>	<b>2</b>
1.1 Téléchargement des données : . . . . .	2
1.2 Les méthodes de prétraitement : . . . . .	2
1.2.1 Ajustement de contraste : . . . . .	2
1.2.2 Saturation : . . . . .	2
1.2.3 Rehaussement logarithmique : . . . . .	2
1.2.4 Fonction de puissance : . . . . .	3
1.2.5 Egalisation de l'histogramme : . . . . .	3
1.2.6 Filtre médian : . . . . .	3
1.2.7 Filtre de lissage (moyenne) : . . . . .	3
<b>2 Segmentation</b>	<b>4</b>
2.1 Méthode d'Otsu . . . . .	4
<b>3 Les opérateurs morphologiques</b>	<b>5</b>
<b>4 Comptage et différenciation des pièces</b>	<b>6</b>
4.1 Comptage de nombre de pièces : . . . . .	6
4.2 Différenciation des pièces par taille : . . . . .	6
<b>5 Évaluation</b>	<b>8</b>
<b>6 Interface de comptage des pièces</b>	<b>9</b>
<b>Conclusion Générale</b>	<b>10</b>

# Table des figures

1.1	Les méthodes de prétraitement. . . . .	3
2.1	Implémentation de la méthode d'Otsu. . . . .	4
3.1	Opérateurs morphologiques appliqués. . . . .	5
4.1	Comptage de nombre de pièces. . . . .	6
4.2	Différenciation des pièces par taille . . . . .	7
5.1	Erreur de comptage des pièces. . . . .	8
6.1	Interface Graphique pour le Comptage des Pièces de Monnaie. . . . .	9

# Introduction Générale

Dans un monde de plus en plus automatisé, le traitement d'images numériques occupe une place centrale dans de nombreux domaines tels que la surveillance, la santé, la robotique et les systèmes embarqués. Parmi ses applications, la reconnaissance et l'analyse d'images permettent d'automatiser des tâches complexes nécessitant auparavant une intervention humaine. Ce projet s'inscrit dans cette optique en développant un système de comptage automatique de pièces de monnaie à partir des images.

L'objectif principal de ce projet est de concevoir un programme capable de reconnaître et de compter des pièces de monnaie en utilisant des techniques de traitement d'images. À travers ce projet, plusieurs défis sont abordés, notamment le prétraitement d'images pour améliorer leur qualité, la segmentation pour isoler les pièces du fond, et le filtrage pour éliminer les objets indésirables. Une fois ces étapes réalisées, le système doit être capable de compter les pièces avec précision et de les classer en fonction de leur taille, représentant potentiellement des valeurs différentes.

Le système sera testé sur un jeu de données contenant des images variées de pièces de monnaie, afin d'évaluer sa performance et sa robustesse. Les résultats obtenus permettront de mesurer la précision du comptage et d'identifier les éventuelles limites ou cas d'erreur. En complément, une interface utilisateur simple pourra être développée pour une meilleure visualisation et interaction avec le système.

Ce rapport détaille les différentes étapes de la conception du système, depuis l'analyse des images jusqu'à l'obtention des résultats finaux, en passant par les algorithmes utilisés et les observations relevées. Il reflète l'ensemble des compétences théoriques et pratiques acquises dans le cadre du module de traitement d'images numériques.

# Prétraitement des images

## 1.1 Téléchargement des données :

- Accédez au site Kaggle et téléchargez le fichier compressé contenant les images.
- Décompressez l'archive.
- Vérifiez que toutes les images sont bien accessibles.

## 1.2 Les méthodes de prétraitement :

L'objectif du prétraitement est d'améliorer la qualité de l'image afin de pouvoir l'analyser plus efficacement. Il nous permet d'éliminer les distorsions indésirables et d'améliorer des qualités spécifiques qui sont essentielles pour l'application sur laquelle nous travaillons.

### 1.2.1 Ajustement de contraste :

Cette méthode modifie l'intensité des pixels pour augmenter ou diminuer le contraste de l'image, améliorant ainsi la distinction entre les objets et le fond.

### 1.2.2 Saturation :

Elle limite les valeurs des pixels à une plage définie pour éviter les parties trop sombres ou trop claires, accentuant les détails visibles.

### 1.2.3 Rehaussement logarithmique :

La transformation logarithmique améliore les détails dans les zones sombres, rendant visibles les régions à faible contraste sans surexposer les zones claires.

### 1.2.4 Fonction de puissance :

Cette transformation ajuste l'éclairage global de l'image en fonction d'un exposant, modifiant le contraste général pour éclaircir ou assombrir l'image.

### 1.2.5 Egalisation de l'histogramme :

Réorganise les intensités des pixels pour répartir uniformément les niveaux de gris, améliorant ainsi le contraste global et la visibilité des détails, surtout dans les images avec des intensités sous-utilisées.

### 1.2.6 Filtre médian :

Ce filtre remplace chaque pixel par la médiane de ses voisins, réduisant le bruit tout en préservant les bords et les détails importants.

### 1.2.7 Filtre de lissage (moyenne) :

Ce filtre permet de remplacer chaque pixel par la moyenne des intensités de ses voisins dans une fenêtre donnée. Son objectif est de lisser l'image et d'éliminer le bruit en adoucissant les variations d'intensité

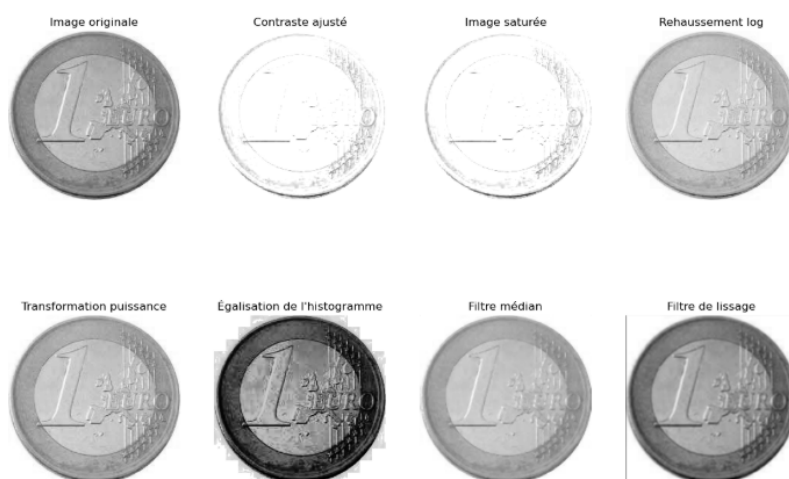


FIG. 1.1 : Les méthodes de prétraitement.

**Remarque :** Il est essentiel de ne pas appliquer toutes ces méthodes sur les images. Certaines techniques peuvent être redondantes ou même contre-productives si elles sont utilisées simultanément, c'est pour ça que dans la suite des étapes, nous allons effectuer plusieurs essais afin de conserver les meilleures méthodes qui conviennent à nos images, et les meilleurs résultats ont été obtenus en utilisant **le filtre de lissage et l'égalisation de l'histogramme**.

## Segmentation

La segmentation d'image est une technique de vision par ordinateur qui divise une image numérique en groupes de pixels distincts afin de faciliter la détection d'objets et les tâches connexes. En décomposant les données visuelles complexes d'une image en segments de forme spécifique, la segmentation d'image permet un traitement de l'image plus rapide et plus avancé.

### 2.1 Méthode d'Otsu

Il existe plusieurs méthodes de segmentation, parmi lesquelles la méthode d'Otsu est particulièrement utilisée.

C'est une technique de segmentation d'images qui permet de réaliser un seuillage automatique en utilisant l'histogramme de l'image. Elle suppose que l'image contient deux classes de pixels (le premier plan et l'arrière-plan). L'objectif est de déterminer un seuil optimal qui maximise la séparation entre ces deux classes en minimisant leur variance intra-classe ou de manière équivalente en maximisant leur variance inter-classe.

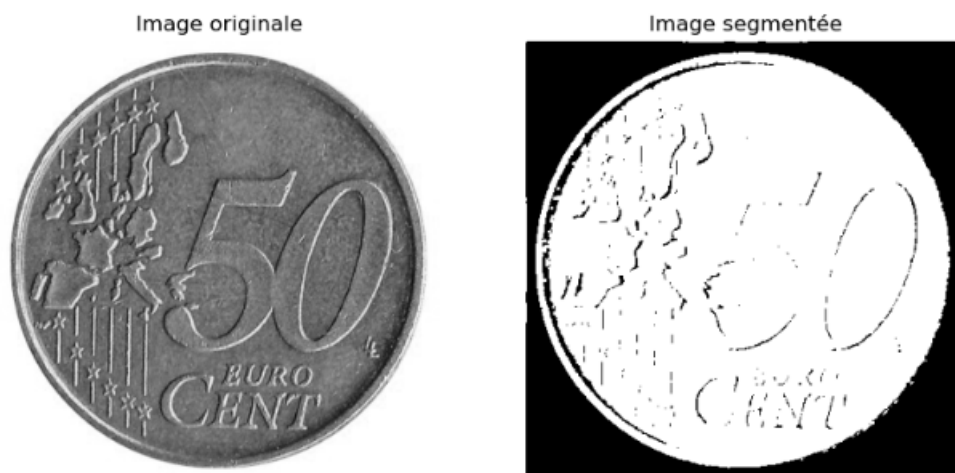


FIG. 2.1 : Implémentation de la méthode d'Otsu.

## Les opérateurs morphologiques

Les opérateurs morphologiques sont des techniques de traitement d'images utilisées pour analyser et modifier la forme des objets dans les images binaires. Ces opérateurs permettent de simplifier les structures, de réduire le bruit en se basant sur un élément structurant qui guide les opérations.

**Opération de Dilatation :** Étend les zones blanches pour combler les trous.

**Opération d'érosion :** Réduit les zones blanches pour supprimer les petits objets parasites.

**Opération d'ouverture :** Supprime le bruit en appliquant une érosion suivie d'une dilatation.

**Opération de fermeture :** Comble les trous dans les objets en appliquant une dilatation suivie d'une érosion.



FIG. 3.1 : Opérateurs morphologiques appliqués.

**Remarque :** Pour obtenir une seule image nettoyée, nous avons choisi l'ouverture suivie d'une fermeture car cette combinaison permet de nettoyer les bruits inutiles tout en préservant l'intégrité des pièces, notamment en comblant les trous et en reconnectant les bords.



# Comptage et différenciation des pièces

## 4.1 Comptage de nombre de pièces :

Nous avons utilisé un algorithme de recherche en profondeur (DFS, pour "Depth-First Search") pour identifier les zones connectées (par un 4-voisinage ou un 8-voisinage) dans une image. À chaque fois qu'un nouvel objet (pièce) est trouvé, il est étiqueté et le compteur d'objets est incrémenté. L'algorithme prend en compte uniquement les pixels de valeur 255 (blanc), représentant les objets. Le résultat final est le nombre total d'objets détectés dans l'image.

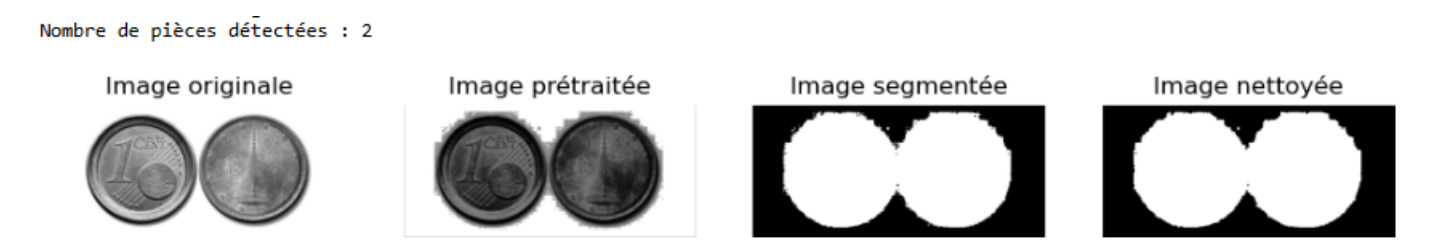


FIG. 4.1 : Comptage de nombre de pièces.

## 4.2 Différenciation des pièces par taille :

Chaque pièce est isolée en identifiant les objets connectés via une recherche par voisinage (DFS). Pour chaque pièce, l'aire (nombre total de pixels blancs composent la pièce) est calculée. Les pièces sont ensuite classées en trois catégories selon leur aire :

- Petite pièce :  $aire < thresholds[0]$  (seuil)
- Moyenne pièce :  $thresholds[0] \leq aire < thresholds[1]$  (entre 200 et 800 pixels)
- Grande pièce :  $aire \geq thresholds[1]$

```
Traitement de l'image 5
Nombre de pièces détectées : 9
Aires détectées : [2779, 3954, 3730, 3517, 2956, 2933, 2329, 1698, 24]
Classification des pièces : {'small': 1, 'medium': 0, 'large': 8}
```

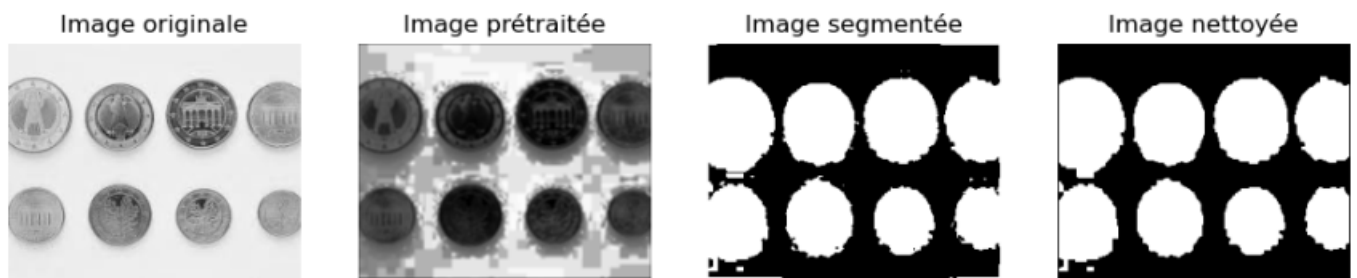


FIG. 4.2 : Différenciation des pièces par taille

**Remarque :** On peut remarquer que malgré les erreurs, lorsque le nombre réel de pièces est de 8 et que le système en détecte 9, la différenciation par taille permet d’analyser ces résultats. En effet, on observe que les 8 grandes pièces sont correctement identifiées, tandis que la petite pièce supplémentaire correspond à du bruit résiduel dans l’image. Cela montre que même en présence de bruit ou d’artefacts, la méthode de différenciation par taille contribue à limiter les impacts des erreurs de détection et à maintenir une bonne interprétation des résultats.

# Évaluation

Pour évaluer la performance du système de comptage de pièces, nous avons comparé le nombre réel de pièces dans chaque image avec les valeurs détectées par le système. L’exactitude a été calculée en comparant les valeurs réelles et détectées, en utilisant la formule suivante :

$$\text{Exactitude} = \frac{\text{Nombre réel de pièces}}{\text{Nombre détecté de pièces}} \times 100$$

L’erreur a été calculée comme étant le complément de l’exactitude, soit :

$$\text{Erreur} = 100\% - \text{Exactitude}$$

Les résultats, comprenant les valeurs réelles, détectées, l’exactitude et l’erreur, sont ensuite enregistrés dans un fichier Excel.

Image	Réel	Détecté	Exactitude (%)	Erreur (%)
0017c3b0e	2	4	50	50
016bad69c	1	12	8.33	91.67
01cf2502d	2	2	100	0
023a9c780	1	183	0.55	99.45
026b6bf6e	8	9	88.89	11.11
03b9ef036	6	30	20	80
0449a7f11	8	29	27.59	72.41
0474cca2a	1	1	100	0
048646a4b	1	46	2.17	97.83
04fb6c4fe	8	13	61.54	38.46

FIG. 5.1 : Erreur de comptage des pièces.

**Remarque :** Le taux d’erreurs sur 10 pièces est tout de même élevé, c’est dû à la présence d’éléments non pertinents dans les images. Malgré l’application de diverses combinaisons de méthodes de prétraitement, de segmentation et d’opérateurs morphologiques, ce résultat représente le meilleur compromis que nous avons atteint.

## Interface de comptage des pièces

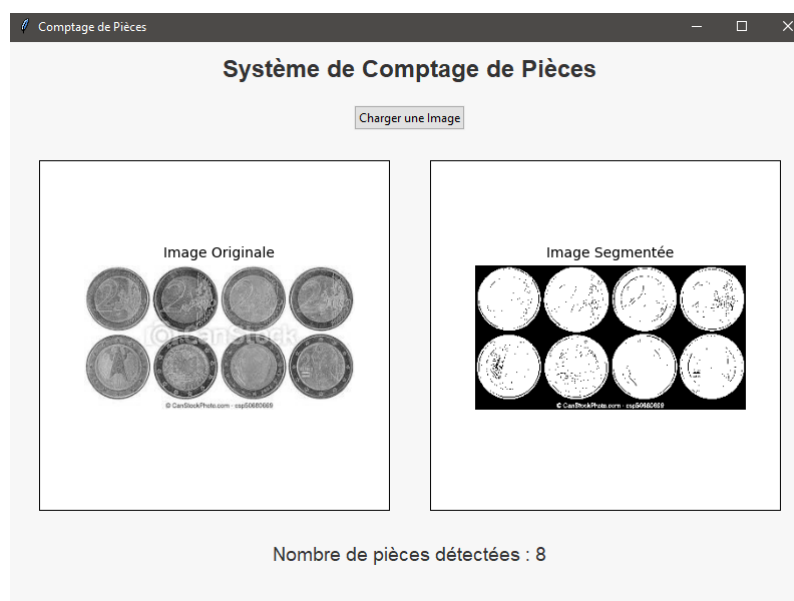


FIG. 6.1 : Interface Graphique pour le Comptage des Pièces de Monnaie.

Cette application permet de charger une image contenant des pièces de monnaie, de réaliser un prétraitement, une segmentation et un comptage des objets détectés à l'aide de la méthode de seuillage d'Otsu et des opérateurs morphologiques. L'interface graphique est construite avec Tkinter, offrant une interface simple pour l'utilisateur. Après le traitement, l'image originale et l'image segmentée sont affichées, et le nombre de pièces détectées est affiché en temps réel.

Cette application facilite ainsi la visualisation des résultats et l'exécution des opérations de comptage sur les objets présents dans l'image.

# Conclusion Générale

Dans ce projet, nous avons développé un système automatisé de comptage de pièces de monnaie basé sur des techniques de traitement d'image. Nous avons appliqué des méthodes de prétraitement pour améliorer la qualité des images, utilisé des techniques de segmentation comme la méthode d'Otsu pour binariser les images et employé des opérateurs morphologiques pour nettoyer les résultats. Le comptage des objets a été effectué à l'aide d'une approche de labellisation connectée, permettant de détecter et de compter avec précision les pièces de monnaie dans les images.

L'application a été intégrée dans une interface graphique simple et intuitive, réalisée avec Tkinter, facilitant ainsi l'interaction avec l'utilisateur. Elle permet non seulement de visualiser les images avant et après traitement, mais aussi de fournir un retour sur le nombre de pièces détectées. Le système a montré de bonnes performances pour des images avec un contraste clair entre les objets et le fond, bien que des ajustements supplémentaires puissent être nécessaires pour des images avec des conditions d'éclairage complexes.

Ce projet ouvre la voie à des applications potentielles dans des domaines tels que l'automatisation de la collecte de fonds, le contrôle des pièces dans les machines automatiques, et la gestion des stocks de pièces dans les commerces.