

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Benyoucef BENKHEDDA-Alger1



Faculté des Sciences  
Département Informatique

---

MASTER 01 Analyse et Science des Données  
Projet d'Extraction d'Information

## Identification d'individus par l'iris

**Réalisé et Présenté Par :**

RABIAI Mehdi Ayoub  
ZAHAF Boualem Nadjib  
BARÇA Ahmed Fateh  
LIMANI Mohamed Amine

**Sous l'Encadrement de :**

M<sup>me</sup> SOUAMI Feryel

## Table des matières :

Introduction .....	3
Environnement de développement .....	4
Langage et bibliothèques utilisés : .....	4
Base de données : .....	4
I    Méthodes traditionnelles de traitement d'image .....	5
I.1    Phase de l'Enrôlement : .....	5
I.1.1    Module d'acquisition : .....	5
I.1.2    Module de Pré-traitement : .....	5
Etapas suivies pour la détection des cercles : .....	5
Égalisation d'histogramme dans la région de l'iris .....	6
I.1.3    Module d'extraction des caractéristiques : .....	6
Transformation de caractéristiques invariante à l'échelle (SIFT).....	6
Keypoint Filtering (Filtrage des points clés) .....	6
I.2    Phase de Reconnaissance : .....	7
I.2.1    Module de mise en correspondance : .....	7
Brute-Force Matcher avec KNN .....	7
I.2.2    Module de décision .....	7
Étape 1 : Extraction des caractéristiques de l'image de test .....	7
Étape 2 : Désérialisation des données stockées .....	8
Étape 3 : Appariement des caractéristiques .....	8
Étape 4 : Détermination du meilleur correspondant .....	8
II    Approche par apprentissage profond et filtres de Gabor .....	8
II.1    Enrôlement : .....	8
II.1.1    Module d'acquisition : .....	8
II.1.2    Pré-traitement : .....	8
Modèle de feuille de caoutchouc de Daugman .....	8
Amélioration de l'image utilisant l'égalisation d'histogramme .....	9
ROI Masking (Masquage ROI) .....	9
(Etapas supplémentaires pour la méthode basée CNN) .....	9
II.1.3    Extraction de caractéristiques : .....	9
Approche CNN : .....	9
Approche par filtre de Gabor : .....	9
II.2    Reconnaissance : .....	10
III    Résultats .....	10
Méthode « traditionnelle » : .....	10
Méthode utilisant CNN : .....	10
Méthode utilisant Gabon : .....	10
IV    Conclusion .....	11
Bibliographie : .....	12

## **Introduction**

Ce projet de reconnaissance de l'iris est le fruit d'une collaboration entre Rabiai Mehdi Ayoub , Zahaf Boualem Nadjib, Barca Fateh Ahmed et Limani Mohamed Amine. Dans ce projet innovant, nous avons exploré trois méthodologies différentes pour développer un logiciel de reconnaissance de l'iris avec une interface époustouflante. Les méthodes utilisées sont les suivantes :

### **1.Méthode Traditionnelle (brute force knn)**

### **2.Réseau Neuronal Convolutionnel (CNN)**

### **3.Filtre de Gabor + Réseau Neuronal**

Ce projet met en lumière l'efficacité de la combinaison de méthodes modernes et traditionnelles pour la reconnaissance de l'iris et démontre l'approche collaborative et innovante adoptée par notre équipe. Notre objectif était de créer un système fiable et précis, capable d'identifier les individus par leurs iris, en utilisant les avancées les plus récentes en matière de traitement d'images et d'intelligence artificielle.

# Environnement de développement

## Langage et bibliothèques utilisés :

Le script pour l'implémentation d'une solution pour l'identification d'individus par l'iris a été réalisé en python.

L'utilisation des bibliothèques suivantes fu nécessaire :

- Open CV
- NumPy
- Miscellaneous operating system interfaces (os)
- Sys
- Math
- Random
- Copy
- Inspect
- Matplotlib
- Pickle

## Base de données :

Pour ce projet de reconnaissance de l'iris, nous avons utilisé la base de données **CASIA-Iris-Syn**, disponible sur Kaggle. Cette base de données est particulièrement adaptée aux projets de reconnaissance de l'iris en raison de sa diversité et de la qualité des images qu'elle contient.

La base de données **CASIA-Iris-Syn** comprend des images d'iris synthétiques générées pour imiter les conditions réelles de capture des images d'iris. Chaque image est soigneusement annotée et les variations incluent différents angles de capture, différentes conditions de lumière et des caractéristiques variées des iris, ce qui en fait une ressource précieuse pour l'entraînement et le test des modèles de reconnaissance de l'iris.

[CASIA-Iris-Syn sur Kaggle.](#)

En résumé, la base de données **CASIA-Iris-Syn** a joué un rôle crucial dans le développement de notre projet, offrant un ensemble de données riche et diversifié pour entraîner et évaluer nos modèles de reconnaissance de l'iris.

# I Méthodes traditionnelles de traitement d'image

## I.1 Phase de l'Enrôlement :

### I.1.1 Module d'acquisition :

Pour lire les images de la base de données la fonction « read\_image » a été créée. Elle utilise une fonction de la bibliothèque « OpenCv » afin de retourner une matrice Numpy qui représente l'image stockée en niveaux de gris.

### I.1.2 Module de Pré-traitement :

Le pré-traitement et l'extraction des caractéristiques se font à l'intérieur de la fonction « extract\_regions » en utilisant plusieurs fonctions dédiées.

La fonction « extract\_regions » commence par lire l'image grâce à « read\_image », par la suite vient l'étape de la localisation de l'iris dans l'image.

#### Étapes suivies pour la détection des cercles :

Pour cela, la fonction « locate\_iris\_boundaries » a été créée. Elle utilise principalement deux fonctions pour localiser les limites de l'iris dans l'image :

- « detect\_inner\_circle » pour détecter le cercle qui se trouve à l'intérieur de l'iris (les bords de la pupille).
- « detect\_outer\_circle » qui s'occupe de détecter les limites extérieures de l'iris.

**Détection de bords de Canny :** Utilisée pour sa compétence à détecter un large spectre de bords dans les images, la fonction « cv2.Canny() » est mise en œuvre avec deux valeurs de seuil : un seuil bas de 20 et un seuil élevé de 100. Cette méthode garantit que les bords liés présentant des gradients forts sont également pris en compte.

**Dilatation et flou gaussien :** Après la détection des bords, les bords subissent une dilatation à l'aide d'un noyau pour augmenter leur épaisseur, ce qui aide à la détection de cercles ultérieure. De plus, un flou gaussien est appliqué pour lisser l'image, réduisant le bruit et les détails. La taille du noyau pour le flou est déterminée dynamiquement :

$$(\text{blur\_size} = 2 * \text{random.randrange}(5, 11) + 1)$$

Ce qui aide à généraliser la détection des bords dans diverses images et conditions.

**Hough Circle Transform :** Transformation de Hough pour les cercles : Cette technique détecte les cercles dans des images imparfaites grâce à un mécanisme de vote dans un espace paramétrique, dérivé de chaque point de bord détecté. Les paramètres, en particulier le seuil de l'accumulateur (circle\_param2), sont ajustés dynamiquement pour s'adapter à différentes qualités d'image et tailles d'iris, améliorant la probabilité d'identifier correctement les limites de l'iris.

Les résultats retournés par cette fonction sont donc les coordonnées des deux cercles (interne et externe) qui sont aux limites de l'iris. Un message signal un problème dans le cas contraire.

### **Égalisation d'histogramme dans la région de l'iris**

Par la suite, « extract\_regions » fait appel à une autre fonction « equalize\_iris\_region » qui prend les résultats retournés par « locate\_iris\_boundaries » en paramètres, afin d'égaliser l'histogramme de la région de l'iris (et non pas l'œil en entier). Cela aide à améliorer le contraste et à normaliser l'illumination de l'iris.

**ROI Masking (Masquage ROI) :** Un masque est créé en délimitant deux cercles, l'un pour la limite extérieure de l'iris et l'autre à l'intérieur pour la pupille, et en soustrayant le cercle intérieur du cercle extérieur pour isoler la région de l'iris.

**Égalisation :** La fonction « cv2.equalizeHist() » est utilisée pour appliquer l'égalisation d'histogramme à la région de l'iris isolée, améliorant considérablement le contraste des caractéristiques, ce qui est vital pour détecter des points clés distincts dans les zones de faible contraste de l'iris.

Une fois l'histogramme égalisé, la fonction « segment\_regions » est appelée pour extraire la région de l'iris à partir de l'image égalisée. Cette étape découpe la région d'intérêt de l'iris à partir de l'image globale en utilisant les coordonnées des cercles interne et externe.

## **I.1.3 Module d'extraction des caractéristiques :**

### **Transformation de caractéristiques invariante à l'échelle (SIFT)**

Suite à ce processus, toujours à l'intérieur de la fonction « extract\_regions », une nouvelle fonction « detect\_keypoints » est appelée pour détecter les keypoints de la région de l'iris en utilisant le moteur SIFT (Scale-Invariant Feature Transform) de la bibliothèque « OpenCv ».

### **Keypoint Filtering (Filtrage des points clés)**

Un filtrage des keypoints est effectuée par la même fonction pour ne garder que ceux à l'intérieur de l'iris.

Pour terminer, la fonction « compute\_descriptors » est appelée. Elle permet, à l'aide de la méthode « compute » du moteur « SIFT », de calculer les descripteurs correspondant aux keypoints détectés.

Les descripteurs sont des vecteurs numériques qui décrivent les caractéristiques locales des keypoints détectés. Ils capturent des informations telles que la texture, la forme et l'intensité des pixels environnants autour de chaque keypoint.

Une fois les descripteurs calculés, les résultats sont assignés aux clés « keypoints » et « descriptors » dans le dictionnaire « regions['entire'] ».

La fonction « extract\_regions » retourne enfin le dictionnaire « regions » qui contient les keypoints des régions détectées de l'iris, ainsi que leurs descripteurs.

## **I.2 Phase de Reconnaissance :**

### **I.2.1 Module de mise en correspondance :**

La fonction « match\_all\_regions » est responsable de la comparaison des iris extraites de deux images et d'identifier les points de correspondance entre ces régions. Elle utilise la fonction « find\_matches » pour trouver les correspondances entre les descripteurs.

#### **Brute-Force Matcher avec KNN**

Dans cette dernière fonction un objet « BFMatcher » (Brute Force Matcher) est créé. Ce « matcher » est utilisé pour trouver les correspondances entre les descripteurs des keypoints des deux régions. La méthode « knnMatch » du « matcher » est utilisée pour trouver les deux meilleures correspondances ( $k=2$ ) entre les descripteurs des régions.

**Validation géométrique :** Chaque correspondance subit une validation basée sur la cohérence des propriétés géométriques entre les caractéristiques appariées :

**Consistance angulaire :** Calculée à l'aide de « math.atan2() », cette métrique garantit que les points clés appariés maintiennent des positions angulaires similaires par rapport à leurs cercles intérieurs respectifs.

**Distance radiale normalisée :** La distance radiale du centre du cercle intérieur à chaque point clé est normalisée par la différence de rayon entre les cercles intérieur et extérieur, s'accommodant des variations de taille d'iris et de distance de la caméra.

**Sérialisation des points clés :** Les attributs fondamentaux de chaque point clé (position, taille, angle, réponse) sont sérialisés en utilisant le module « pickle » de Python. Ce processus est essentiel pour transformer des structures de données complexes en un format gérable pour un stockage et une récupération faciles, facilitant ainsi la gestion efficace des données et la comparaison à travers de grands ensembles de données.

### **I.2.2 Module de décision**

Cette section décrit le processus de détection et d'identification des individus à partir des données sérialisées de l'iris. C'est une phase critique qui exploite tout le travail antérieur en matière de prétraitement, d'extraction de caractéristiques et de sérialisation des données.

#### **Étape 1 : Extraction des caractéristiques de l'image de test**

Le flux de travail commence par l'extraction des régions et le calcul des descripteurs pour les points clés dans une nouvelle image de test. Ce processus est le miroir de la méthode utilisée pour les images dans la base de données, garantissant la cohérence dans la représentation des caractéristiques.

### **Étape 2 : Désérialisation des données stockées**

Ensuite, le script charge les données sérialisées de chaque individu stocké dans la base de données. Ces données incluent les points clés et leurs descripteurs associés, préparés et sauvegardés lors du traitement initial des images de la base de données.

### **Étape 3 : Appariement des caractéristiques**

Les caractéristiques extraites de l'image de test sont appariées à chaque candidat dans la base de données sérialisée en utilisant l'algorithme « BFMatcher ». Les correspondances sont ensuite filtrées selon plusieurs critères pour garantir la précision :

### **Étape 4 : Détermination du meilleur correspondant**

Le script identifie l'individu avec le plus grand nombre de correspondances valides. Si le nombre de correspondances valides dépasse un seuil prédéterminé, l'identité de l'image de test est confirmée ; sinon, aucune correspondance n'est déclarée, indiquant l'absence de l'individu dans la base de données.

## **II Approche par apprentissage profond et filtres de Gabor**

### **II.1 Enrôlement :**

#### **II.1.1 Module d'acquisition :**

Le processus d'acquisition et de pré-traitement (jusqu'à la normalisation) sont les mêmes que pour la première approche.

#### **II.1.2 Pré-traitement :**

Le type de normalisation choisit pour cette méthode est celui de Daugman :

#### **Modèle de feuille de caoutchouc de Daugman**

Ce modèle facilite le déroulement de l'iris de coordonnées cartésiennes à polaires, assurant que chaque point à l'intérieur de l'iris est mappé à une paire spécifique  $(r, \theta)$ . Ici,  $r$  varie de la limite de la pupille au limbe, et  $\theta$  représente l'angle autour de la pupille. Cette normalisation garantit que les caractéristiques de l'iris sont cohérentes malgré les variations des conditions d'imagerie. Les transformations sont définies comme :

$$I(x(r,\theta),y(r,\theta)) \rightarrow I(r,\theta)$$

$$x(r,\theta) = r \cos(\theta)$$

$$y(r,\theta) = r \sin(\theta)$$



### **Amélioration de l'image utilisant l'égalisation d'histogramme :**

Cette technique ajuste le contraste des images en redistribuant l'histogramme des valeurs d'intensité. Elle améliore le contraste dans les régions aux valeurs d'intensité proches, améliorant l'efficacité des processus d'extraction de caractéristiques subséquents.

### **ROI Masking (Masquage ROI)**

Tout comme pour la première approche, un masquage ROI est appliqué avant de pouvoir sauvegarder les données autant que « pre-processed data »

### **(Etapas supplémentaires pour la méthode basée CNN)**

Une normalisation des images qui consiste à mettre à l'échelle les valeurs des pixels dans une plage uniforme entre 0 et 1, réalisée en divisant chaque valeur de pixel par la valeur maximale possible de pixel (255 pour une image en niveaux de gris).

La formule pour cette opération est :

$$\text{Valeur de pixel normalisée} = \frac{\text{Valeur de pixel originale}}{255}$$

Un encodage type « One Hot Encoder » est par la suite appliqué afin de préparer les données pour les étapes suivantes.

## **II.1.3 Extraction de caractéristiques :**

### **Approche CNN :**

**Couches convolutives :** Ces couches utilisent des filtres apprenables qui sont appliqués à l'image d'entrée. Chaque filtre couvre un petit champ récepteur mais s'étend sur toute la profondeur du volume d'entrée. L'opération de convolution implique le calcul du produit scalaire entre le filtre et les régions de l'image, produisant une carte d'activation 2D. Ce processus est exprimé mathématiquement comme suit :

$$\text{Sortie } (i,j) = \sum_m \sum_n \text{Entrée } (i+m,j+n) \times \text{Noyau}(m,n)$$

**Couches convolutives :** Ces couches réduisent la taille spatiale de l'entrée en la divisant en régions rectangulaires et en sélectionnant la valeur maximale de chaque région. Ce processus diminue la charge computationnelle pour les couches subséquentes et introduit une forme d'invariance de translation aux cartes de caractéristiques.

### **Approche par filtre de Gabor :**

**Les filtres de Gabor :** sont des filtres passe-bande spécialisés conçus pour capturer les détails texturaux fins de l'iris, cruciaux pour reconnaître les motifs uniques de l'iris. La formulation mathématique d'un filtre de Gabor est une onde sinusoïdale modulée par une fonction gaussienne :

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma)=\exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda}+\psi\right)$$

$$x' = x \cos\theta + y \sin\theta$$

$$y' = -x \sin\theta + y \cos\theta$$

Où :

- $\lambda$  est la longueur d'onde du facteur sinusoïdal (sinusoïdal factor).
- $\theta$  représente l'orientation de la normale aux bandes parallèles d'une fonction de Gabor.
- $\psi$  est le décalage de phase.
- $\sigma$  est l'écart type/enveloppe gaussienne (standard deviation of the Gaussian envelope).
- $\gamma$  est le rapport d'aspect spatial.

Le résultat de l'application de ces filtres est un vecteur de « features » qui ne sera pas normalisé pour éviter la perte d'informations. Cette étape prend en moyenne 35 secondes.

Après une phase d'entraînement, chacun des deux modèles « CNN » et « Gabon » sont enregistrés dans un fichier H5.

## II.2 Reconnaissance :

Les deux modèles, formés sur des ensembles de données d'images d'iris, sont exploités pour extraire des caractéristiques et effectuer des prédictions précises. Lorsqu'une image d'iris est fournie en entrée, chaque modèle analyse l'image en identifiant des schémas et en extrayant des caractéristiques significatives. Ces caractéristiques sont ensuite utilisées pour l'identification de l'iris et la reconnaissance de l'individu auquel il appartient.

## III Résultats

### Méthode « traditionnelle » :

**Taux de précision :** 98%

**Temps de prédiction (sans compter le processus de normalisation):** 200 millisecondes

### Méthode utilisant CNN :

**Taux de précision :** 98% ~ 100%

**Temps de prédiction (sans compter le processus de normalisation):** 120 millisecondes

### Méthode utilisant Gabor :

**Taux de précision :** 99%

**Temps de prédiction (sans compter le processus de normalisation):** 120 millisecondes

## **IV Conclusion**

La présente étude a exploré trois approches distinctes pour l'identification d'iris, en utilisant à la fois des méthodes traditionnelles (brute force knn) et des techniques basées sur des réseaux de neurones convolutionnels (CNN) ainsi que des filtres de Gabor. Notre analyse a révélé que chaque méthode présente ses propres avantages et performances spécifiques. La méthode traditionnelle, en exploitant des techniques bien établies telles que la transformation de caractéristiques invariante à l'échelle (SIFT) et la validation géométrique, a démontré une précision remarquable de 98%. Cependant, les approches basées sur CNN et les filtres de Gabor ont surpassé cette précision avec un taux atteignant jusqu'à 100%, tout en maintenant un temps de prédiction quasi-constant de 120 millisecondes. Cette amélioration de la précision témoigne de l'efficacité des techniques d'apprentissage profond dans la reconnaissance d'iris.

## Bibliographie :

### DATABASE :

[https://www.kaggle.com/datasets/monareyhanii/casia-iris-syn?resource=download&fbclid=IwZXh0bgNhZW0CMTEAAAR17wBHSyfCr77fxa8soTJv5PSa8uV43dxKbR2ISf02z8yDAzeOZZqCnzJc\\_aem\\_AZP3JzGuTqAvYS4FQArw5QIk\\_qlJywLCoHCQCLYLfRqmKij1jJyEi4jFKoDd4uB1zx9XCRBY5Fq1LTuUo1rMgfi](https://www.kaggle.com/datasets/monareyhanii/casia-iris-syn?resource=download&fbclid=IwZXh0bgNhZW0CMTEAAAR17wBHSyfCr77fxa8soTJv5PSa8uV43dxKbR2ISf02z8yDAzeOZZqCnzJc_aem_AZP3JzGuTqAvYS4FQArw5QIk_qlJywLCoHCQCLYLfRqmKij1jJyEi4jFKoDd4uB1zx9XCRBY5Fq1LTuUo1rMgfi)

### GITHUB :

- <https://github.com/OmarMedhat22/iris-Recognition-Casia-Interval-v4>
- <https://github.com/kranthi419/Iris-recognition-based-on-Gabor-and-Deep-Convolutional-Networks/>
- [https://github.com/yj7082126/OpenCV\\_IrisRecognition](https://github.com/yj7082126/OpenCV_IrisRecognition)
- [https://www.researchgate.net/publication/229007521\\_A\\_new\\_accurate\\_technique\\_for\\_Iris\\_boundary\\_detection](https://www.researchgate.net/publication/229007521_A_new_accurate_technique_for_Iris_boundary_detection)
- <https://github.com/sarthak-arora1997/Iris-Recognition>