Projet préparé pour le module

Gestion des incertitudes

Ecole Centrale de Lille

Master 2 en Management de l'Intelligence Artificielle en Santé

M2 MIAS – Promotion 2023-2024

Amélioration de la qualité des images médicales d'arthrose du genou par la technique de logique floue

Préparé par :

Yao Moya
Batrouni Sara

Supervisé par

Mr. Yassine Ouzar

Le 21 Janvier 2024

Table des matières

INTRODUCTION A LA LOGIQUE FLOUE 2

ENSEMBLE DE DONNÉES CHOISI 3

1. Contexte 3

2. Contenu 3

LOGIQUE FLOUE APPLIQUÉE A NOTRE DATASET 4

A- Objectifs 4

B- Etapes et résultats 5

1. Installation des bibliothèques nécessaires au traitement d'images et à la vision par ordinateur 5

2. Traitement des images 5

2.1. Techniques d'amélioration du contraste 5

2.2. Méthode proposée en adoptant la logique floue 6

2.2.1. Description 6

2.2.2. Démarche technique 6

2.2.3. Démonstration 12

2.2.4. Résultats 19

BIBLIOGRAPHIE 20

INTRODUCTION A LA LOGIQUE FLOUE

La logique floue, ou fuzzy logic en anglais, est une approche en informatique et en intelligence artificielle qui permet de traiter des valeurs de vérité partielles, situées entre les valeurs booléennes vrai et faux. Contrairement à la logique booléenne classique, qui repose sur des valeurs vraies ou fausses, la logique floue permet de modéliser des systèmes complexes en utilisant des variables qualitatives. Elle est basée sur le concept que les humains prennent des décisions en se basant sur des informations imprécises et non numériques. Ainsi, la logique floue permet de mieux appréhender les phénomènes humains par nature difficilement modélisables, en combinant des vérités partielles pour parvenir à une décision acceptable. Elle est utilisée dans divers domaines tels que le contrôle des systèmes, la reconnaissance de formes, et les applications d'intelligence artificielle. La logique floue a été formalisée dans les années 1960 par le Dr. Lotfi Zadeh, de l'Université de Californie à Berkeley, et elle est essentielle au développement de fonctionnalités quasi humaines en intelligence artificielle [1] [2].

La logique floue, ou fuzzy logic en anglais, diffère de la logique traditionnelle de plusieurs manières:

- Traitement des valeurs de vérité : Contrairement à la logique traditionnelle qui se base sur des valeurs booléennes (vrai ou faux), la logique floue permet de traiter des valeurs de vérité partielles, situées entre 0 et 1, ce qui permet de modéliser des concepts imprécis ou vagues [3].s
- Modélisation des systèmes complexes : La logique floue est utilisée pour modéliser des systèmes complexes en utilisant des variables qualitatives, ce qui la rend plus adaptée pour représenter des phénomènes humains difficiles à modéliser de manière binaire [3].
- Degré de vérité : En logique floue, un concept peut posséder un degré de vérité n'importe où entre 0.0 et 1.0, alors que la logique traditionnelle se limite à des concepts complètement vrais (degré de vérité 1.0) ou complètement faux (degré de vérité 0.0) [4].

En résumé, la logique floue étend la logique traditionnelle en permettant le traitement de valeurs de vérité partielles, ce qui la rend plus adaptée pour modéliser des concepts imprécis ou vagues, tels que ceux rencontrés dans les phénomènes humains ou les systèmes complexes.

ENSEMBLE DE DONNÉES CHOISI

1. Contexte

L'arthrose du genou se définit par une dégénérescence du cartilage articulaire du genou, le matériau souple et glissant qui protège normalement les os des frottements et des chocs de l'articulation. Cette affection entraîne également des modifications de l'os situé sous le cartilage et peut affecter les tissus mous avoisinants. L'arthrose du genou est de loin la forme d'arthrite la plus courante qui provoque des douleurs au genou et est souvent appelée simplement arthrite du genou. De nombreux autres types d'arthrite moins courants peuvent également causer des douleurs au genou, notamment la polyarthrite rhumatoïde, la pseudo-goutte et l'arthrite réactive.

2. Contenu

Cet ensemble de données contient des données de radiographie du genou pour la détection de l'articulation du genou et la classification du KL du genou. Les descriptions des grades sont les suivantes :

Grade 0 : image de genou sain.

Grade 1 (douteux) : Rétrécissement douteux de l'articulation avec un possible lippage ostéophytique.

Grade 2 (Minimal) : Présence certaine d'ostéophytes et rétrécissement possible de l'espace articulaire

Grade 3 (modéré) : Ostéophytes multiples, rétrécissement certain de l'interligne articulaire, avec légère sclérose.

Grade 4 (sévère) : Ostéophytes importants, rétrécissement significatif de l'articulation et sclérose sévère.

LOGIQUE FLOUE APPLIQUÉE A NOTRE DATASET

A- Objectifs

L'utilisation de la logique floue sur le dataset d'arthrose du genou s'explique par la nature complexe et nuancée de cette affection. La logique floue offre une approche mathématique adaptée pour traiter l'incertitude et la variabilité dans les données médicales. Voici quelques raisons pour lesquelles la logique floue est pertinente dans ce contexte:

- Incertitude Diagnostique : Les différentes classifications de l'arthrose du genou, allant de Grade 0 à Grade 4, impliquent des degrés variables de sévérité. La logique floue permet de modéliser l'incertitude associée à ces catégories et offre une représentation plus souple des informations diagnostiques.
- Variabilité des Données : Les données radiographiques peuvent présenter une variabilité importante d'une image à l'autre. La logique floue peut être utilisée pour traiter cette variabilité et permettre une analyse plus robuste et adaptable aux nuances présentes dans les images médicales.
- Interprétation Clinique : La logique floue permet de prendre en compte les nuances subtiles dans l'interprétation des radiographies du genou, en alignant mieux les résultats avec la réalité clinique. Elle offre une flexibilité dans la représentation des degrés de sévérité, reflétant ainsi la complexité de la maladie.
- Adaptabilité aux Cas Frontières : Certains cas peuvent être difficiles à classer de manière rigide dans une catégorie spécifique. La logique floue permet de traiter ces cas frontières en attribuant des degrés de membership à plusieurs catégories, ce qui peut être plus proche de la réalité clinique.
- Prise en Compte de la Subjectivité : Les évaluations des radiographies peuvent parfois être subjectives. La logique floue permet de modéliser la subjectivité inhérente aux évaluations médicales en attribuant des degrés de certitude aux différentes catégories.

De plus, la logique floue est préférable à d'autres approches, telles que les réseaux bayésiens, l'approche ensembliste, la théorie des fonctions de croyance, la fonction de masse et les fonctions de plausibilité, pour l'amélioration du contraste et de la qualité d'images médicales d'arthrose en raison de sa capacité à modéliser et gérer l'incertitude de manière plus flexible. Contrairement aux réseaux bayésiens qui reposent sur des probabilités et aux approches ensemblistes et aux théories des fonctions de croyance qui ont leurs propres limitations, la logique floue offre une méthode plus adaptable et robuste pour gérer les incertitudes, ce qui la rend pertinente pour l'amélioration du contraste et de la qualité d'images médicales d'arthrose [5].

a) Objectifs spécifiques

L'amélioration de l'image est une méthode permettant d'améliorer la qualité d'une image et le contraste en est un aspect majeur. Les méthodes traditionnelles d'amélioration du contraste, comme l'égalisation d'histogramme, ont pour effet de sur ou sous améliorer l'image, particulièrement une image de faible résolution.

Ce projet vise à développer un nouveau système d'inférence floue pour améliorer le contraste des images de notre dataset, en surmontant les lacunes des méthodes traditionnelles.

Enfin, les résultats obtenus en utilisant deux approches différentes sont comparés et évalués.

B- Etapes et résultats

Dans le cadre de notre projet, nous avons suivi les étapes suivantes :

1. Installation des bibliothèques nécessaires au traitement d'images et à la vision par ordinateur

- Installation de "cv2" : Nous avons initié l'installation en utilisant la commande pip install cv2, mais malheureusement, cela a généré une erreur indiquant qu'aucune version compatible n'était disponible.
- Installation d'OpenCV avec "pip" : Pour remédier à cela, nous avons utilisé la commande correcte pip install opencv-python pour installer avec succès la bibliothèque OpenCV, version 4.9.0.80.
- Installation de la bibliothèque "glob2" : Une autre dépendance importante, la bibliothèque "glob2", a été installée en utilisant la commande pip install glob2. Cette bibliothèque est couramment utilisée pour la recherche de fichiers dans des répertoires.
- Importation de Modules et Configuration du Chemin : Nous avons importé les modules nécessaires tels que "math", "cv2" (OpenCV), "matplotlib.pyplot" et "numpy" pour faciliter le traitement des images et la visualisation
- De plus, nous avons configuré le chemin du répertoire (PATH) pour pointer vers '4'.

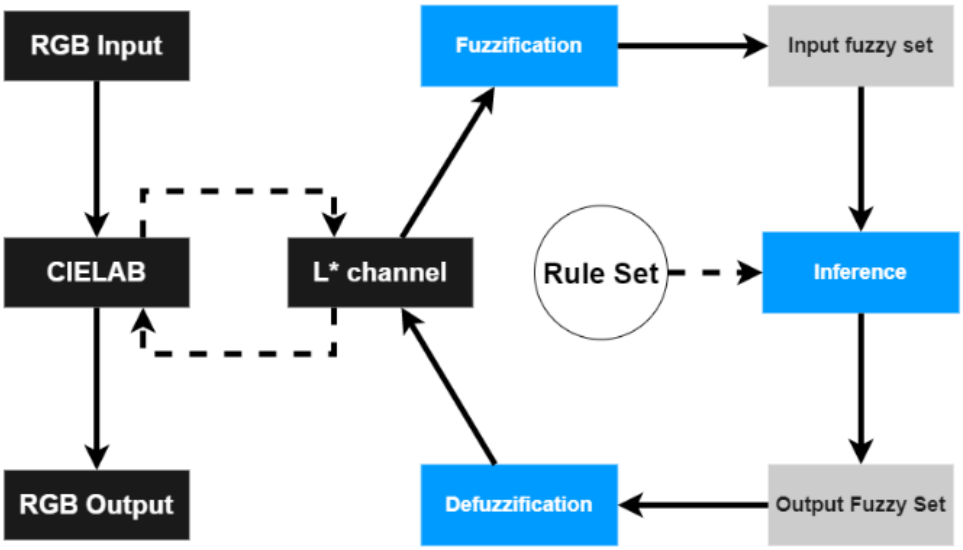
2. Traitement des images

Deux types de techniques de traitement d'images existent :

2.1. Techniques d'amélioration du contraste

- Techniques basées sur la transformation des niveaux de gris : transformation logarithmique, transformation en loi de puissance, transformation linéaire par morceaux, etc.
- Techniques de traitement basées sur l'histogramme : égalisation d'histogramme (HE), spécification d'histogramme, etc.
- La méthode la plus répandue est l'égalisation d'histogramme, qui repose sur l'hypothèse qu'un histogramme en niveaux de gris uniformément distribué présente le meilleur contraste visuel
- D'autres méthodes avancées d'amélioration basées sur l'histogramme comprennent l'égalisation bi-histogramme (BHE), l'égalisation d'histogramme superposé, l'égalisation d'histogramme adaptative multi-échelle, la modification de l'histogramme local avec préservation de la forme, etc.

2.2. Méthode proposée en adoptant la logique floue



2.2.1. Description

- Convertir l'image d'entrée de RGB à CIELAB, en progressant sur le canal L.
- Calculer l'intensité moyenne des pixels - valeur M
- **Fuzzification :** Pour chaque pixel, calculer le degré d'appartenance à chaque classe en fonction de l'intensité du pixel et de la valeur M. Intensité [0, 255].
- **Inférence:** Calculer l'ensemble flou de sortie à partir de l'intensité du pixel d'entrée sur la base de l'ensemble de règles proposé.
- **Défuzzification:** Pour chaque pixel, calculer la valeur du centroïde de son ensemble flou de sortie.
  Centroïde dans [-50, 305]
- Normaliser l'intensité du pixel de sortie de [-50, 305] à [0, 255].
- Fusionner le canal L modifié avec les canaux AB d'origine, convertir l'image de sortie de CIELAB en RVB.

2.2.2. Démarche technique

a. ***Fuzzification***

Dans un premier lieu, nous allons détailler la démarche adoptée pour l'étape de fuzzification (b).

Nous avons élaboré un ensemble de fonctions dans le cadre de la fuzzification de l'intensité des pixels :

Voici les élements mis en place :

- Fonction Gaussienne (G):
  Nous avons défini une fonction gaussienne appelée G qui prend en compte un point x, une moyenne (mean), et un écart-type (std). Cette fonction gaussienne est utilisée pour évaluer la contribution d'un pixel à une catégorie spécifique en fonction de son intensité.

- Fonctions de Membres pour la Fuzzification:
  Plusieurs fonctions de membres ont été créées pour représenter différentes catégories d'intensité des pixels, allant de "Extrêmement sombre" à "Extrêmement lumineux". Chaque fonction de membre utilise la fonction gaussienne G pour calculer le degré d'appartenance d'un pixel à une catégorie spécifique en fonction de sa luminosité relative par rapport à la moyenne (M).

- Catégories d'intensité : Nous avons défini les catégories suivantes :
  - Extrêmement sombre
  - Très sombre
  - Sombre
  - Légèrement sombre
  - Légèrement lumineux
  - Lumineux
  - Très lumineux
  - Extrêmement lumineux

```python
# Fonction Gaussienne:
def G(x, mean, std):
    return np.exp(-0.5*np.square((x-mean)/std))

# Autres Fonctions:
def ExtremelyDark(x, M):
    return G(x, -50, M/6)

def VeryDark(x, M):
    return G(x, 0, M/6)

def Dark(x, M):
    return G(x, M/2, M/6)

def SlightlyDark(x, M):
    return G(x, 5*M/6, M/6)

def SlightlyBright(x, M):
    return G(x, M+(255-M)/6, (255-M)/6)

def Bright(x, M):
    return G(x, M+(255-M)/2, (255-M)/6)

def VeryBright(x, M):
    return G(x, 255, (255-M)/6)

def ExtremelyBright(x, M):
    return G(x, 305, (255-M)/6)
```

L'objectif de cette fuzzification est de quantifier de manière graduelle et continue l'intensité des pixels dans une image, en attribuant à chaque pixel un degré d'appartenance à chacune des catégories définies. Cette approche permet de représenter de manière plus flexible et nuancée la luminosité des pixels dans le cadre d'une logique floue.

Dans la suite du code, nous avons effectué les étapes suivantes :

- Boucle sur les Moyennes (M) :

Une boucle a été mise en place pour itérer sur différentes valeurs de moyennes (M), avec les valeurs spécifiques de 128, 64, et 192.

- Création d'un Espace de Valeurs (x) :

Nous avons défini un espace de valeurs x allant de -50 à 305, qui représente la plage d'intensité des pixels.

- Calcul des Degrés d'Appartenance pour Chaque Catégorie :

Pour chaque valeur de M, nous avons calculé les degrés d'appartenance pour chaque catégorie d'intensité en utilisant les fonctions de membres définies précédemment (ExtremelyDark, VeryDark, Dark, SlightlyDark, SlightlyBright, Bright, VeryBright, ExtremelyBright).

- Visualisation des Fonctions d'Appartenance :

Pour chaque valeur de M, une figure a été créée pour visualiser les fonctions d'appartenance de chaque catégorie. Les lignes représentent les différentes catégories, et les points d'inflexion des courbes indiquent le degré d'appartenance.

- Affichage de Marques Importantes :

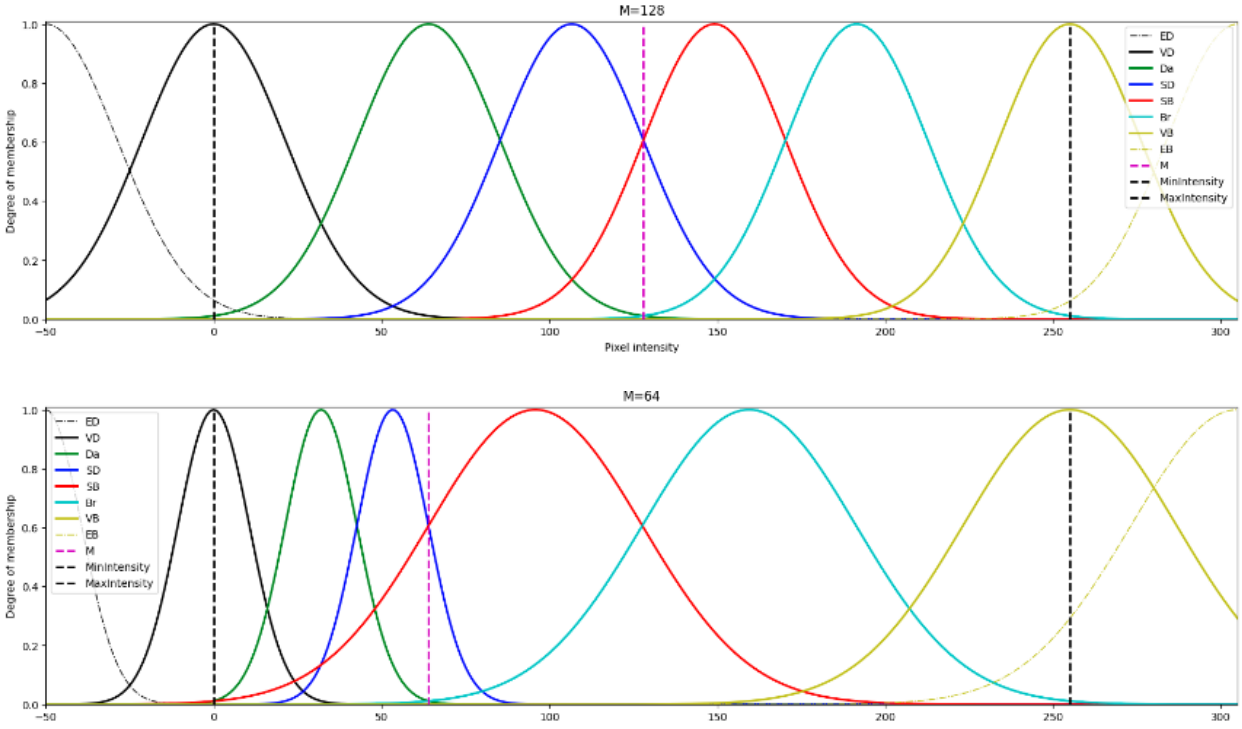
Des marques importantes ont été ajoutées, telles que la ligne verticale représentant la moyenne (M), et des lignes pointillées indiquant les valeurs minimales et maximales d'intensité des pixels (MinIntensity et MaxIntensity).

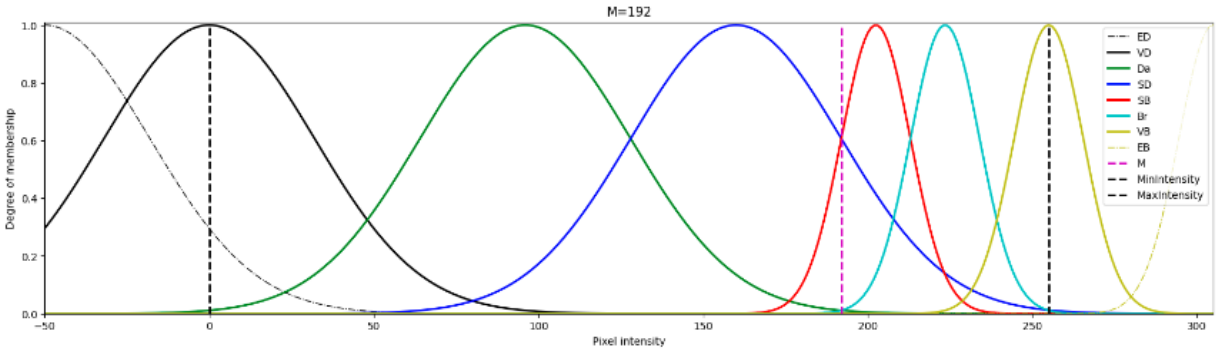
- Paramètres Esthétiques de la Figure :

Les paramètres esthétiques, tels que la couleur, l'étiquetage des axes, et les légendes, ont été ajustés pour rendre la visualisation claire et informative.

- Affichage des figures:

Pour chaque Valeur de M, une figure a été affichée :





- Ensemble de Règles : Les règles suivantes décrivent les relations logiques entre l'intensité d'entrée des pixels et la catégorie de luminosité correspondante en sortie. Chaque règle spécifie une correspondance entre une catégorie d'entrée et une catégorie de sortie dans le système de logique floue.

    - SI l'entrée est Très sombre ALORS la sortie est Extrêmement sombre
    - SI l'entrée est Sombre ALORS la sortie est Très sombre
    - SI l'entrée est Légèrement sombre ALORS la sortie est Sombre
    - SI l'entrée est Légèrement lumineuse ALORS la sortie est Lumineuse
    - SI l'entrée est Lumineuse ALORS la sortie est Très lumineuse
    - SI l'entrée est Très lumineuse ALORS la sortie est Extrêmement lumineuse

b. ***Inférence et défuzzification (Méthode Mamdani)***

Les étapes suivantes montrent comment les fonctions implémentées ont été utilisées pour effectuer l'inférence floue et la défuzzification, daboutissant ainsi à une sortie finale du système de logique floue basé sur la méthode Mamdani.
Voici une description de ce que nous avons fait en utilisant les fonctions fournies :

- Génération des Ensembles Flous de Sortie pour Chaque Classe (OutputFuzzySet):

Nous avons utilisé la fonction OutputFuzzySet pour générer des ensembles flous de sortie pour chaque classe de sortie (Extrêmement sombre à Extrêmement lumineux). Ces ensembles flous représentent la contribution de l'entrée à chaque catégorie en fonction des fonctions de membres et de la moyenne spécifiée (M).

- Agrégation des Ensembles Flous de Sortie (AggregateFuzzySets):

Nous avons agrégé les ensembles flous de sortie générés pour chaque classe en utilisant la fonction AggregateFuzzySets. Cette agrégation a été réalisée en utilisant l'opérateur maximum (Union), ce qui signifie que chaque pixel peut appartenir à plusieurs catégories de sortie avec différents degrés.

- Calcul des Degrés d'Appartenance pour Chaque Classe (Infer):

La fonction Infer a été utilisée pour calculer les degrés d'appartenance pour chaque classe de sortie en fonction des fonctions de membres et de la moyenne spécifiée (M). Ces degrés d'appartenance représentent à quel point chaque pixel appartient à chaque catégorie de sortie.

- Génération des Ensembles Flous de Sortie pour l'Inférence (OutputFuzzySet dans Infer):

Dans le cadre de l'inférence floue, la fonction OutputFuzzySet a été appelée pour chaque classe de sortie, générant ainsi des ensembles flous de sortie basés sur les degrés d'appartenance calculés précédemment.

- Agrégation des Ensembles Flous de Sortie pour l'Inférence (Infer):

Les ensembles flous de sortie pour l'inférence ont été agrégés en utilisant la fonction AggregateFuzzySets. Cela a permis de combiner les contributions de chaque classe de sortie en un seul ensemble flou agrégé.

- Calcul du Centre de Gravité (Défuzzification) (Infer):

Enfin, nous avons utilisé la fonction Infer pour calculer le centre de gravité de l'ensemble flou agrégé. Cela représente la valeur défuzzifiée, c'est-à-dire la réponse finale du système de logique floue en termes de l'intensité des pixels.

Ensuite, nous avons effectué une analyse pour différentes intensités de pixels (64, 96, 160, 192) :

- Boucle For pour Chaque Intensité de Pixel :

Une boucle for a été utilisée pour itérer sur différentes intensités de pixels (64, 96, 160, 192).

- Définition de la Moyenne (M) et de la Plage d'Entrée (x) :

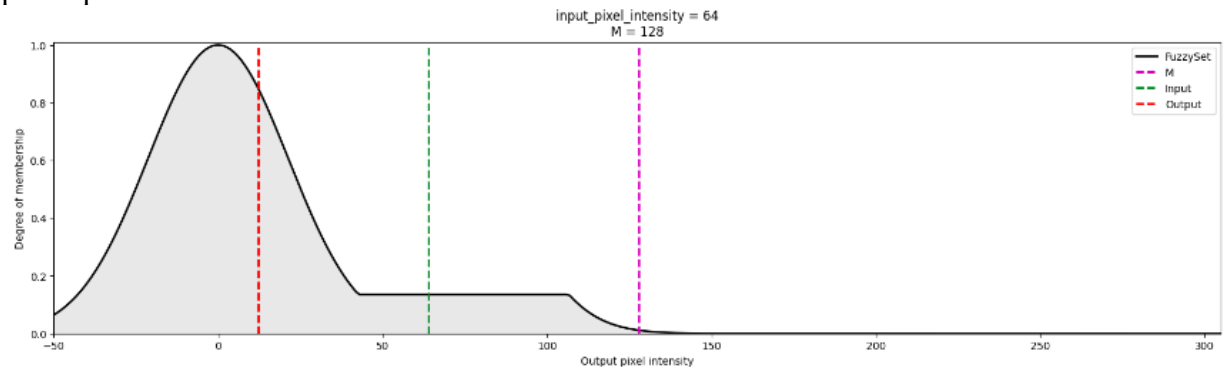
La moyenne (M) a été fixée à 128, et la plage d'entrée (x) a été définie de -50 à 305.

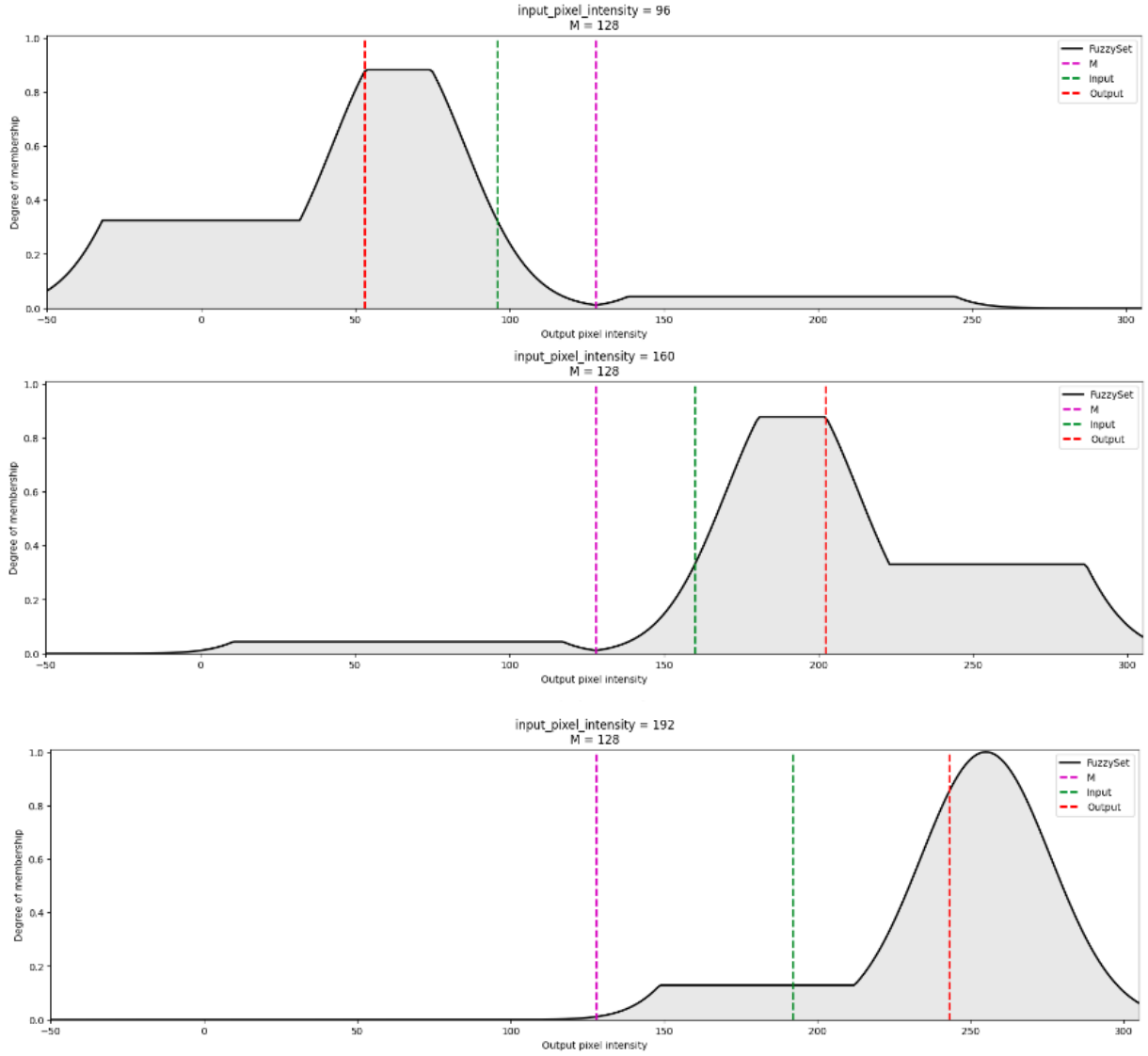
- Appel de la Fonction Infer pour l'Inférence Floue et la Défuzzification :

La fonction Infer a été appelée pour chaque intensité de pixel. En option, l'ensemble flou de sortie a également été récupéré.

- Tracé des Résultats :

Les résultats ont été tracés sur un graphique pour chaque intensité de pixel. Le graphique inclut l'ensemble flou de sortie, la moyenne (M), l'intensité de pixel d'entrée, et la sortie défuzzifiée (centroïde). Le remplissage de la zone sous la courbe de l'ensemble flou de sortie est également représenté. Chaque graphique représente une analyse détaillée pour une intensité de pixel spécifique.





Ensuite, nous avons effectué une analyse des cartographies d'entrée-sortie (IO) pour différentes moyennes (M) :

- Définition des Moyennes (means) :

Une liste de moyennes a été définie, comprenant les valeurs (64, 96, 128, 160, 192).

- Boucle For pour Chaque Moyenne :

Une boucle for a été utilisée pour itérer sur chaque valeur de la liste des moyennes.

- Définition de la Plage d'Entrée (x) :

La plage d'entrée (x) a été définie de 0 à 255, représentant les valeurs d'intensité de pixels en niveaux de gris.

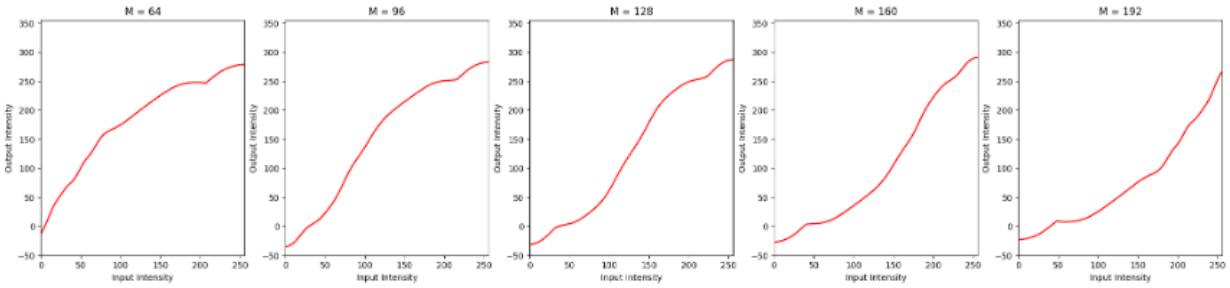
- Calcul des Cartographies d'Entrée-Sortie (Infer) :

Pour chaque valeur de la plage d'entrée, la fonction Infer a été appelée pour calculer la cartographie d'entrée-sortie. Le temps d'exécution (%time) a également été mesuré pour chaque itération.

- Tracé des Résultats :

Les résultats ont été tracés sur un graphique pour chaque spécifique, permettant de visualiser comment différentes moyennes influent sur la cartographie d'entrée-sortie du système de logique floue, offrant ainsi une compréhension de la transformation de l'intensité d'entrée en intensité de sortie pour différentes conditions.

```
CPU times: total: 62.5 ms

Wall time: 136 ms

CPU times: total: 46.9 ms

Wall time: 147 ms

CPU times: total: 78.1 ms

Wall time: 251 ms

CPU times: total: 93.8 ms

Wall time: 195 ms

CPU times: total: 156 ms

Wall time: 283 ms
```



2.2.3. Démonstration

Nous avons commencé par définir trois méthodes de traitement d'image pour améliorer le contraste qui vont etre implementées dans la suite : FuzzyContrastEnhance, HE (histogramme égalisé) et CLAHE (Contrast Limited Adaptive Histogram Equalization). Ces méthodes fournissent différentes approches pour améliorer le contraste des images en utilisant des techniques floues, d'égalisation d'histogramme et d'égalisation adaptative limitée par le contraste.

a. Méthode FuzzyContrastEnhance

Cette méthode utilise une approche floue pour améliorer le contraste de l'image. Voici une explication des étapes principales :

- Conversion de l'image RGB en espace de couleur LAB.
- Extraction du canal L (luminance) de l'espace LAB.
- Calcul de la valeur moyenne M du canal L et ajustement de M en fonction de sa relation avec 128.
- Précalcul de la transformation floue en utilisant la fonction Infer.
- Application de la transformation floue au canal L.
- Mise à l'échelle du canal L résultant entre 0 et 255.
- Conversion de l'image LAB modifiée en RGB.

b. Méthode HE (Histogram Equalization)

C'est la méthode traditionnelle d'égalisation d'histogramme, qui améliore le contraste en redistribuant les niveaux de gris. Les étapes principales comprennent :

- Conversion de l'image RGB en espace de couleur LAB.
- Application de l'égalisation d'histogramme uniquement sur le canal L.
- Conversion de l'image LAB modifiée en RGB.

c. Méthode CLAHE (Contrast Limited Adaptive Histogram Equalization)

Il s'agit d'une méthode d'égalisation d'histogramme adaptative limitée par le contraste. Les principales étapes sont les suivantes :

- Création d'un objet CLAHE avec des paramètres spécifiques.
- Conversion de l'image RGB en espace de couleur LAB.
- Application de l'égalisation d'histogramme adaptative au canal L en utilisant CLAHE.
- Conversion de l'image LAB modifiée en RGB.

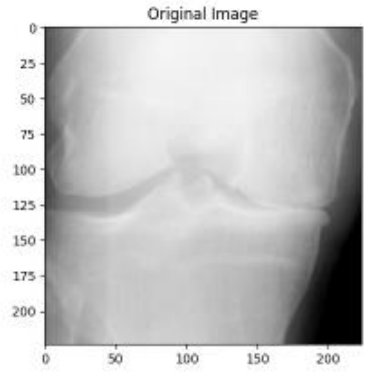
Pour télecharger nos images, on les a converti en format RGB, et stocké dans un tableau NumPy pour une manipulation ultérieure dans l'analyse d'image ou d'autres tâches. Le tableau Numpy a indiqué qu'il y a 44 images au total, chacune ayant une taille de 224x224 pixels et 3 canaux de couleur.

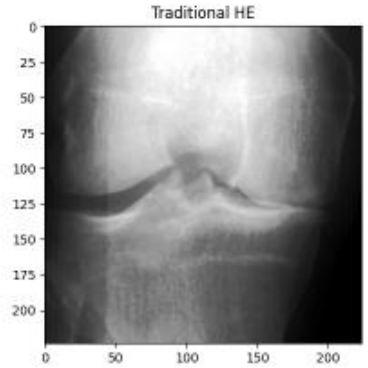
```
(44, 224, 224, 3)
```

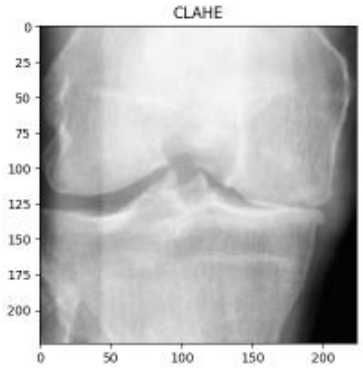
Ensuite, à l'aide d'une boucle, nous avons itéré sur chaque image du tableau.
Pour chaque image, la méthode Fuzzy Contrast Enhance (FCE) a été appliquée, utilisant une approche basée sur la logique floue pour améliorer le contraste.

Nous avons également appliqué deux méthodes classiques d'amélioration de contraste, Histogram Equalization (HE) et Contrast Limited Adaptive Histogram Equalization (CLAHE), à chaque image.
Les résultats de chaque méthode (Original, FCE, HE, CLAHE) ont été affichés côte à côte dans une figure pour une comparaison visuelle. Les images originales et améliorées ont été présentées dans une figure, chaque image accompagnée de titres correspondants.
L'affichage a permis une observation visuelle des effets de chaque méthode sur le contraste des images.

Sample Photo 1

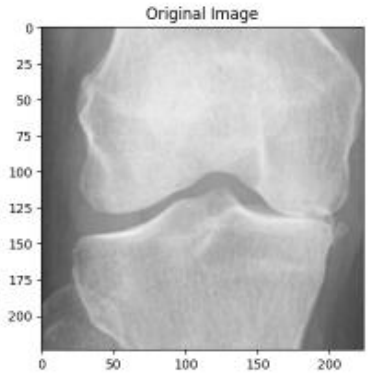




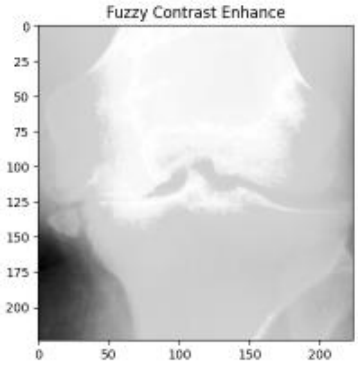


Sample Photo 2

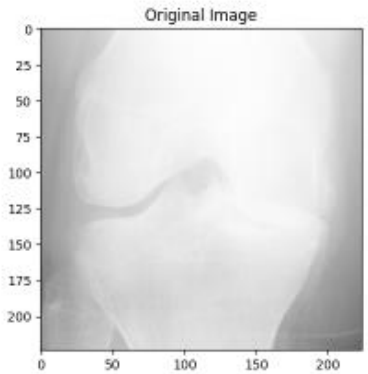
¶



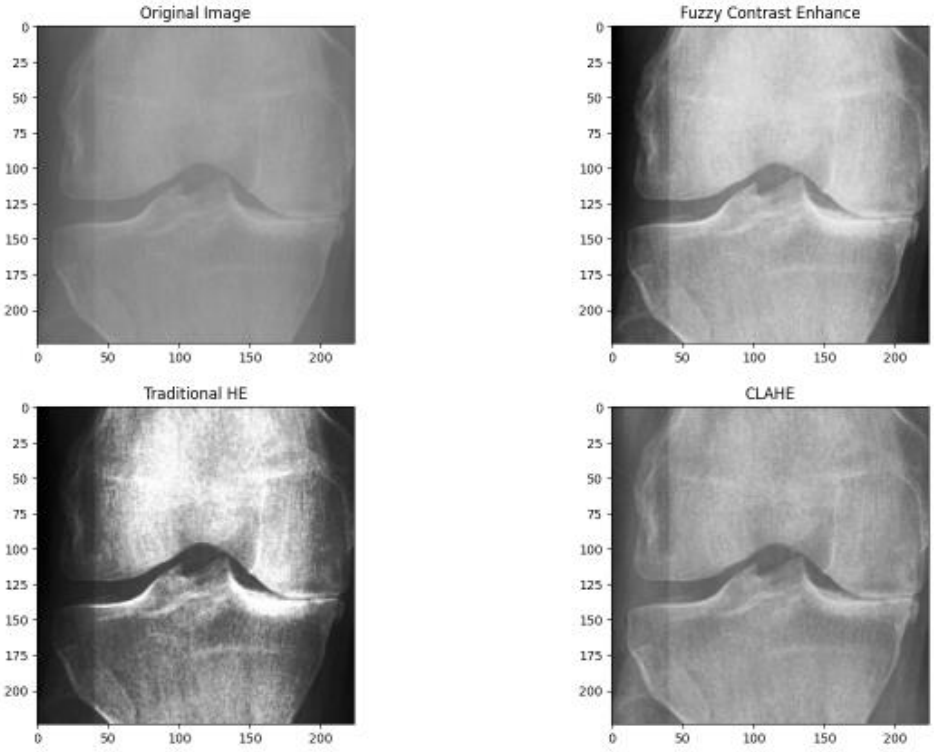
Sample Photo 3



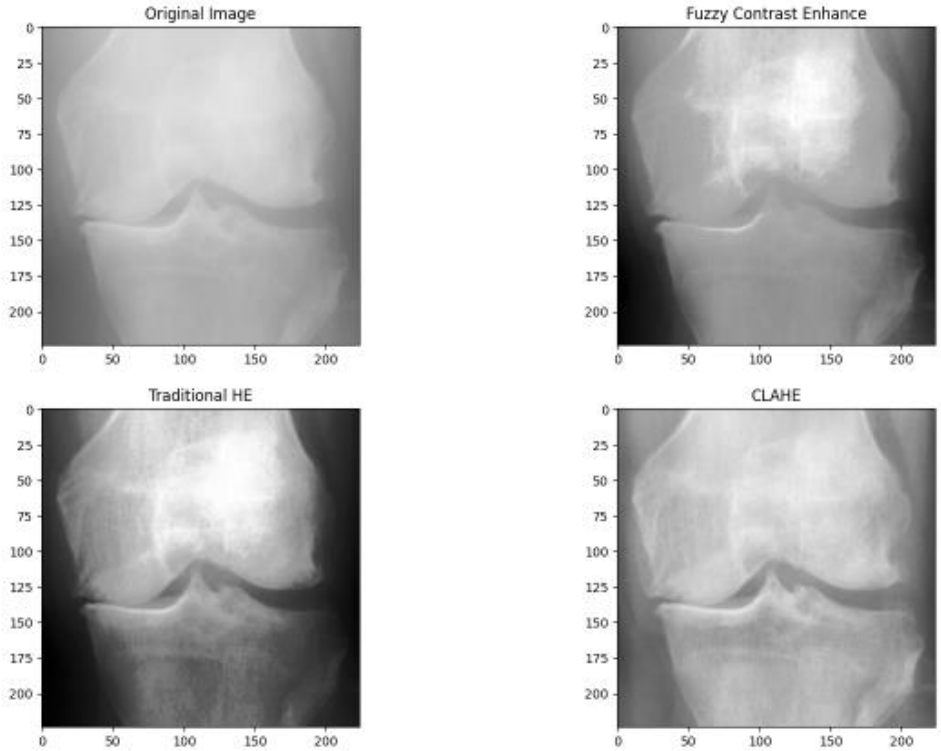
Sample Photo 4



Sample Photo 5

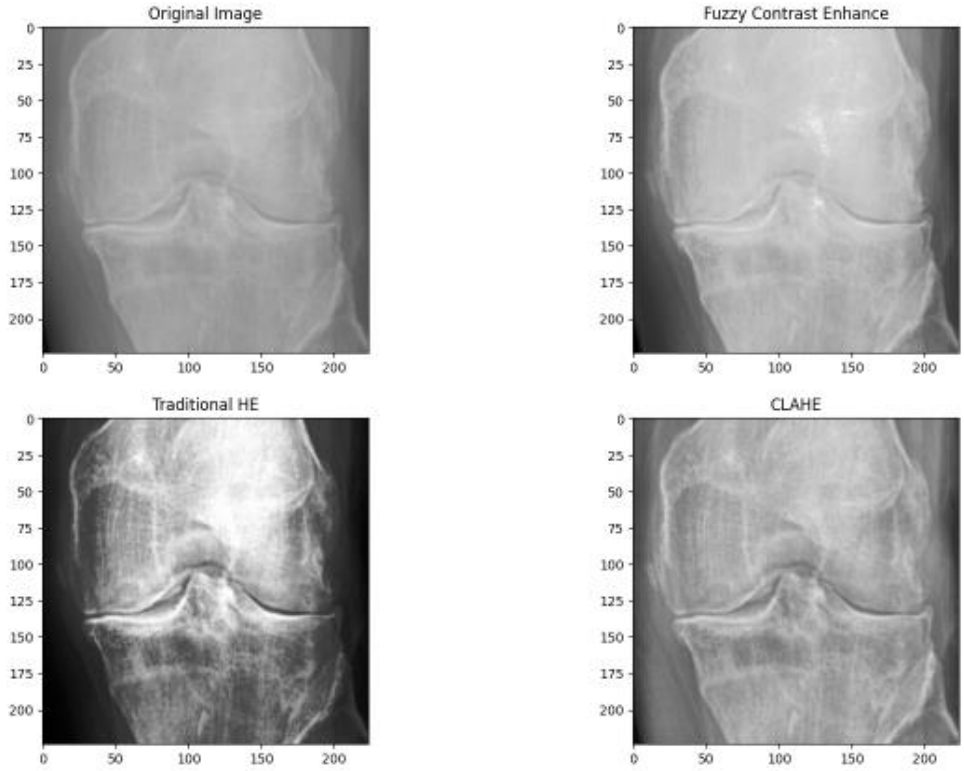


Sample Photo 6

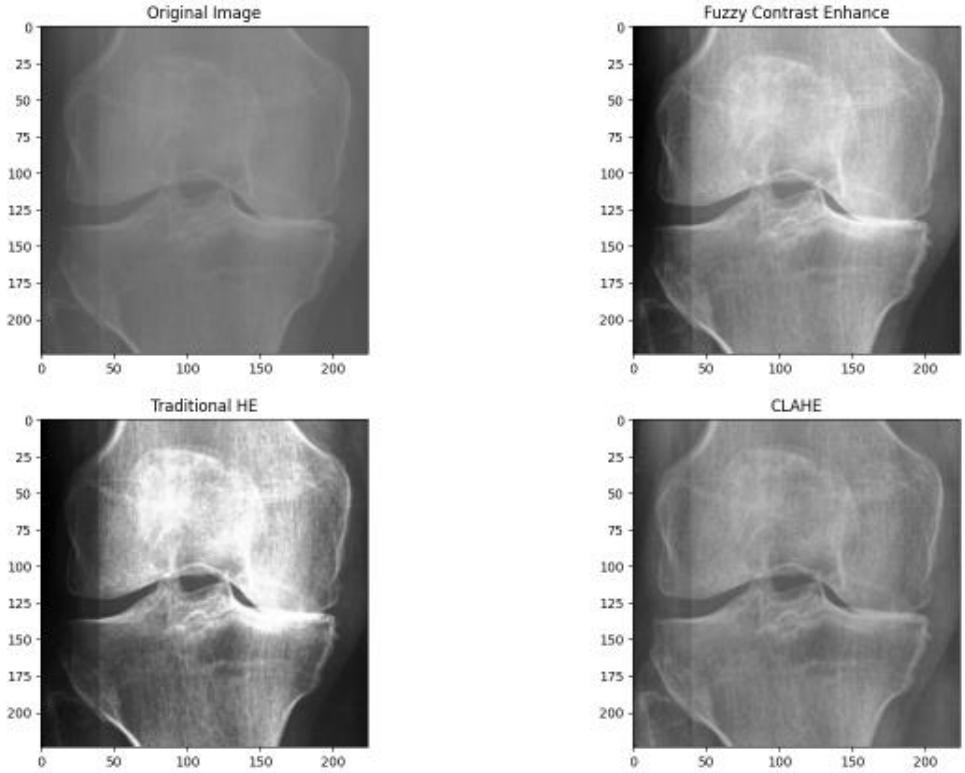


Sample Photo 7

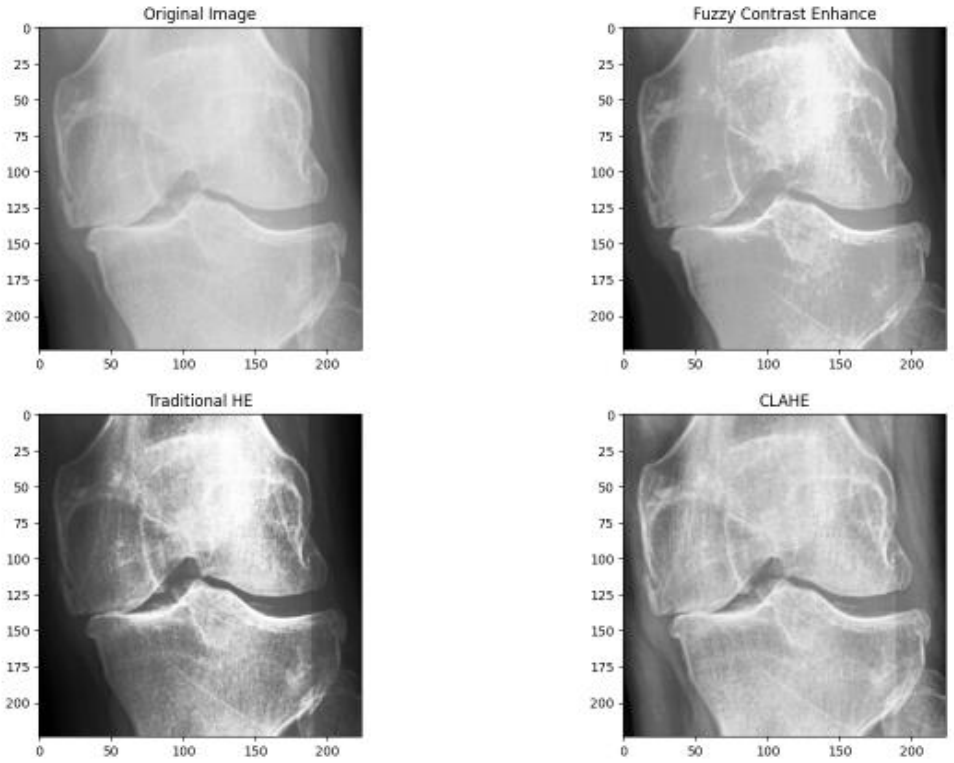
¶



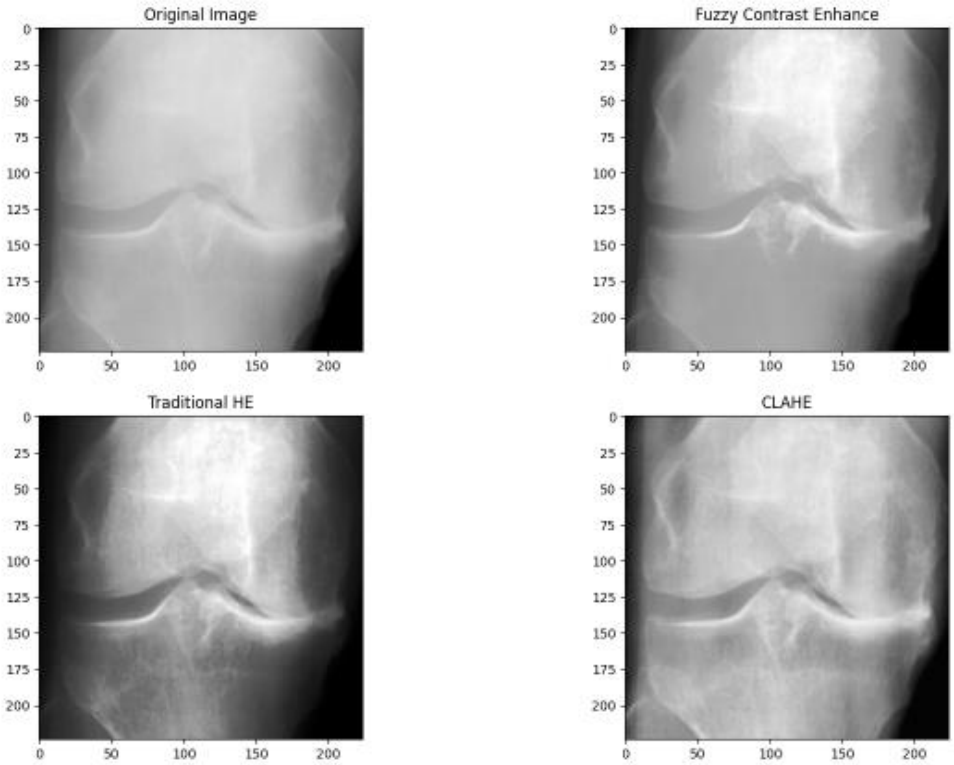
Sample Photo 8



Sample Photo 9



Sample Photo 10



S

La suite des images est affichée dans le lien du code ci-joint.

2.2.4. Résultats

Nous avons mesuré le temps d'exécution pour chaque méthode sur un sous-ensemble de 10 images.

FCE : Temps total - 578 ms, Temps mural - 1.68 s
HE : Temps total - 62.5 ms, Temps mural - 37.2 ms
CLAHE : Temps total - 141 ms, Temps mural - 46.7 ms

Nous avons calculé le Peak Signal-to-Noise Ratio (PSNR) pour évaluer la qualité de l'image restaurée.

PSNR pour FCE : 76.20
PSNR pour HE : 75.98
PSNR pour CLAHE : 77.65

Observations :

Les résultats visuels ont montré que FCE a une capacité notable à améliorer le contraste des images, bien que la méthode CLAHE ait obtenu le PSNR le plus élevé.
Les temps d'exécution indiquent que FCE prend plus de temps que HE mais moins de temps que CLAHE.
Le choix entre les méthodes dépendra des exigences spécifiques en termes de qualité de restauration et de performances temporelles.

BIBLIOGRAPHIE

[1] “Que signifie logique floue ? - Definition IT de LeMagIT,” LeMagIT. Accessed: Jan. 20, 2024. [Online]. Available: https://www.lemagit.fr/definition/logique-floue

[2] “Logique floue : définition et cas d’usage.” Accessed: Jan. 20, 2024. [Online]. Available: https://www.journaldunet.fr/intelligence-artificielle/guide-de-l-intelligence-artificielle/1501877-logique-floue-definition/

[3] “Fuzzy Logic (Stanford Encyclopedia of Philosophy).” Accessed: Jan. 20, 2024. [Online]. Available: https://plato.stanford.edu/entries/logic-fuzzy/

[4] “What is ‘fuzzy logic’? Are there computers that are inherently fuzzy and do not apply the usual binary logic? | Scientific American.” Accessed: Jan. 20, 2024. [Online]. Available: https://www.scientificamerican.com/article/what-is-fuzzy-logic-are-t/

[5] I. Bloch and H. Maitre, “Les m´ethodes de raisonnement dans les images”.

Image contrast enhancement using fuzzy logic
https://arxiv.org/ftp/arxiv/papers/1809/1809.04529.pdf

A fast and efficient color image enhancement method based on fuzzy-logic and histogram
http://dx.doi.org/10.1016/j.aeue.2013.08.015