

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

*Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj*

*Faculté des Sciences et de la Technologie*

*Département d'Electronique*

# *Mémoire*

*Présenté pour obtenir*

LE DIPLOME DE MASTER

FILIERE : ELECTRONIQUE

**Spécialité : Systèmes Embarqués**

Par

- **DERRADJ AFFEF SAHAR**
- **BEN CHIKH MASSOUDA**

*Intitulé*

## *Algorithmes d'Alignement d'Image à Base de Réseaux de Neurones Convolutionnels (CNN)*

*Évalué le : 15 / 09 / 2021*

*Par la commission d'évaluation composée de\* :*

<i>Nom &amp; Prénom</i>	<i>Grade</i>	<i>Qualité</i>	<i>Etablissement</i>
<i>Dr. Mohamed Lamine TALBI</i>	<i>MCA</i>	<i>Président</i>	<i>Univ-BBA</i>
<i>Pr. MESALI ZOUBEIDA</i>	<i>MCB</i>	<i>Encadreur</i>	<i>Univ-BBA</i>
<i>Dr. DIFFELLA NACIRA</i>	<i>MCB</i>	<i>Examineur</i>	<i>Univ-BBA</i>

\* Conformément à :

- L'arrêté n°055 du 21 janvier 2021 Fixant dispositions exceptionnelles autorisées en matière d'organisation et gestion pédagogique, de l'évaluation et de la progression des étudiants, durant la période COVID-19 au titre de l'année universitaire 2020-2021 ;
- Procès-verbal de la réunion de l'équipe du domaine des Sciences et Technologies du mois de Mai 2021.

---

\* Conformément à :

- L'arrêté n°055 du 21 janvier 2021 Fixant dispositions exceptionnelles autorisées en matière d'organisation et gestion pédagogique, de l'évaluation et de la progression des étudiants, durant la période COVID-19 au titre de l'année universitaire 2020-2021 ;
- Procès-verbal de la réunion de l'équipe du domaine des Sciences et Technologies du mois de Mai 2021.

## REMERCIEMENTS

*Nous remercions d'abord Dieu Tout-Puissant de nous avoir donné  
la volonté et continue de travailler.*

*Profitez de cette occasion pour exprimer nos sincères  
remerciements et Mes profonds remerciements à Mme **Messali  
Zubaydah**, notre chef de mémoire et Mme **Sayed Ahmed Somaya**,  
pour : Leurs précieux conseils et orientations nous ont guidés tout  
au long de notre recherche.*

*Nous remercions également sincèrement les membres du jury pour  
leur sollicitude. Mener nos recherches en acceptant d'inspecter et  
d'enrichir notre travail à travers eux Suggérer.*

*Nous remercions sincèrement ceux Elle a aidé et favorisé le  
développement et le succès de cette information. De cette belle  
année scolaire. Nous remercions tous les enseignants pour leur  
dévouement et leur patience, Contribuez à notre formation.*

*Enfin, je tiens à exprimer ma plus sincère gratitude à tous les  
parents et amis qui se soucient de nous Vous nous avez toujours  
encouragés à compléter ce document. Merci à tous.*

## *Dédicace*

*Je dédie ce mémoire :*

*À ma très chère mère*

*Quoi que je fasse ou que je dise, je ne pourrai jamais te remercier  
comme il se doit. Ta présence à côté de moi a toujours été ma  
source de force*

*À mon très cher père*

*Tu as toujours été à mon côté pour me soutenir et m'encourager.*

*Que ce travail traduit ma gratitude et mon affection.*

*À mon cher frère et mes très chères soeurs pour leur*

*Patience, leur amour, leur soutien et leur*

*Encouragement.*

*À tous ceux que j'aime.*

*À tous ceux qui ont toujours cru en moi et poussé à réussir, je vous  
dédie ce travail.*

## Résumé

Aujourd'hui, l'apprentissage en profondeur est un nouveau domaine de recherche qui a récemment émergé au sein de la recherche en apprentissage automatique en simulant des neurones, et il a donné des résultats impressionnants et efficaces qui n'étaient pas possibles auparavant en raison de sa grande précision qui dépasse parfois l'esprit humain. Les réseaux de neurones convolutifs sont des réseaux de neurones multicouches spécialisés dans les tâches de reconnaissance de formes. Notre travail consiste à aligner l'image en utilisant les algorithmes Le Net, le ResNet, la transformation géométrique, la corrélation croisée normalisée, pour préserver les détails et la taille de la base de l'image ainsi que la profondeur où elle a un grand effet pour obtenir les meilleurs résultats.

**Mots-clés :** Alignement d'images, réseaux de neurones convolutifs, apprentissage profond, apprentissage automatique, neurones, Le Net le ResNet, la transformation géométrique, corrélation croisée normalisée.

---

## Abstract

Today, deep learning is a new area of research that has recently emerged within machine learning research by simulating neurons, and it has yielded impressive and effective results that were not previously possible due to of its great precision which sometimes exceeds the human mind. Convolutional neural networks (CNN) are multilayer neural networks specializing in pattern recognition tasks. Our job is to align the image using the Net algorithmes, ResNet, Geometric Transformation, and Normalized Cross Correlation (NCC), to preserve the details and size of the image base as well as the depth where it has a large effect to achieve the best results.

**Key Word :** Image alignment, convolutional neural networks (CNN), deep learning, machine learning, neurons, Le Net, le ResNet, transformation geometrique, normalized cross correlation.

---

## ملخص

اليوم التعلم العميق هو مجال بحث جديد ظهر في الأونة الأخيرة ضمن بحوث تعلم الآلة عن طريق محاكاة الخلايا العصبية فأعطت نتائج مبهرة وفعالة لم تكن ممكنة من قبل نظرا لدقتها العالية التي تتجاوز في بعض الأحيان العقل البشري. تعتبر الشبكات العصبية التلافيفية هي شبكات عصبية متعددة الطبقات متخصصة في مهام التعرف على الأنماط. يتمثل عملنا في محاذاة الصورة ، للحفاظ على تفاصيل وحجم قاعدة الصورة وكذلك العمق حيث له تأثير كبير للحصول على أفضل النتائج. الكلمات المفتاحية: محاذاة الصور، الشبكات العصبية التلافيفية، التعلم العميق، تعلم الآلة، الخلايا العصبية.

Le Net, le ResNet, la transformation géométrique, corrélation croisée normalisée.

## Sommaire

Remerciements .....	
Dédicace .....	
Résumé.....	I
Abstract .....	II
Moulakhas.....	III
Sommaire.....	IV
Listes des Figure.....	V
Liste des Tableaux.....	VI
Notation et Abréviatio.....	VII
Introduction Générale.....	1

### **CHAPITRE 1 : LES RESEAUX DE NEURONES ARTIFICIELS**

1.1 Introduction.....	3
1.2 Réseaux de Neurones Artificiels.....	3
1.2.1 Principe de Base.....	3
1.2.2 Fonction d'Activation.....	4
• Algorithmes.....	5
1.2.3 Apprentissage d'un Réseau de Neurone.....	6
1.3 Réseaux de Neurones Convolutifs.....	6
1.3.1 Architecture de Réseaux de Neurones Convolutifs .....	6
1.3.2 Différents Modules d'un Réseau de Neurones convolutifs.....	8

• Couches de convolutive (CONV).....	8
• Couches de Pooling (POOL).....	8
• Couches de Correction (RELU).....	8
• Couches Entièrement Connectée (FC).....	9
• Couche de Perte (LOSS).....	9
1.4 Réseaux de Neurones Utilisés.....	9
1.4.1 LeNet.....	9
1.4.2 ResNet.....	10
1.5 Différence entre CNN et Réseaux de Neurones Artificiels.....	10
1.6 Conclusion.....	11

## **CHAPITRE 2 : ALGORITHMES d'ALIGNEMENT D'IMAGES**

2.1 Introduction .....	13
2.2 Principe d'Alignement .....	13
2.3 Algorithmes classiques d'Alignement d'Image.....	14
2.3.1 Algorithme d'alignement par Corrélation Croisée.....	15
2.3.2 Algorithme d'alignement par l'Information Mutuelle .....	17
2.3.3. Algorithme d'alignement par transformation géométrique .....	19
2.3.3.1 Translation en 3D.....	19
2.3.4 Rotation en 3D.....	19
2.4 Algorithme d'alignement d'images à base de CNN .....	20
2.5 Critères d'évaluation .....	22
2.5.1 Moyenne Erreur MSE (l'Erreur Quadratique).....	22.
2.5.2 Mesure de similarité.....	22

2.6 Conclusions.....	22
<b>CHAPITRE3:IMPLEMENTATION DES ALGORITHMES D'ALIGNEMENT D'IMAGES</b>	
3.1 Introduction.....	24
3.2 Résultats et Discussions.....	24
3.2.1 Nature de données et image teste.....	24
3.2.2 Matériel et software utilise .....	25
3.2.3 Algorithme conventionnels d'alignement d'images .....	25
• Algorithme d'alignement par Corrélation Croisée.....	29
• Algorithme d'alignement par transformation géométrique .....	29
• Algorithme d'alignement par l'Information Mutuelle .....	32
Algorithme d'alignement d'images à base de CNN .....	34
.3.3 Conclusion.....	44
Conclusion Générale.....	45
Référence.....	46

### Liste des Figures

<b>Chapitre 1:</b> .....	
<b>Figure 1.1</b> Modélisation d'un neurone artificiel.....	4
<b>Figure 1.2</b> Fonctions d'activation couramment utilisées.....	5
<b>Figure 1.3</b> Le principe général de l'algorithme de la retro-propagation de gradient.....	6
<b>Figure 1.4</b> Une couche du CNN en 3 dimensions.....	7
<b>Figure 1.5</b> max Pooling avec un filtre 2x2 et un pas de 2.....	8

<b>Figure 1.6</b> Architecture du réseau de neurones LeNet pour un exemple de classification en 4 classes météorologiques.....	9
<b>Figure 1.7</b> l'architecture de ResNet.....	10
<b>Chapitre 2:</b> .....	
<b>Figure 2.1</b> Processus d'alignement .....	14
<b>Figure 2.2</b> Algorithme d'alignement des images par corrélation croisée.....	16
<b>Figure 2.3</b> Exemple d'alignement de deux images par corrélation croisée.....	17
<b>Chapitre 3 :</b> .....	
<b>Figure 3.1</b> Organigramme de méthode CC sur Matlab.....	26
<b>Figure 3.2</b> Image lena256 format tif de taille 255 × 255pixels.....	26
<b>Figure 3.3:</b> Image lena256 désalignée (à gauche) et image lena256 alignée par CC (adroit).....	27
<b>Figure 3.4</b> Image Parrot format tif de taille 255 × 255 pixels.....	27
<b>Figure 3.5</b> Image Parrot désalignée (à gauche) et image Parrot alignée par CC (à droit).....	27
<b>Figure 3.6</b> Image Foreman format tif de taille 255 × 2565puxe.....	28
<b>Figure3.7</b> Image Foreman désalignée (à gauche) et image Foreman alignée par CC (a droit).....	28
<b>Figure 3.8</b> Images covid_19 test originales tirées de google.....	28
<b>Figure 3.9</b> Image covid_19 désalignée (à gauche) et image covid_19 alignée par CC (à droit).....	29
<b>Figure 3.10</b> Organigramme de méthode transformation géométrique sur Matlab.....	30
<b>Figure 3.11</b> L'image origine avant la transformation géométrique (avant l'alignement).....	30
<b>Figure 3.12</b> Les image 3D dans l'espace de travail et visualisez-el.....	31
<b>Figure 3.13</b> Trois plans de coupe passant par le centre des volumes transformés.....	31
<b>Figure 3.14</b> L'image après le processus de rotation.....	32

<b>Figure 3.15</b> Organigramme de méthode l'information Mutuelle sur Matlab.....	33
<b>Figure 3.16</b> image originale MRI .....	33
<b>Figure 3.17</b> image originale CI .....	34
<b>Figure 3.18</b> image Registered(aligne).....	34
<b>Figure 3.19</b> Illustration d'image de dimensions 28×28, vecteur d'entrée de dimension 784.....	35
<b>Figure 3.20</b> Échantillon de la base de données de reconnaissance de chiffres manuscrits MNIST.....	35
<b>Figure 3.21</b> L'architecture du modèle d'auto-encodeur proposé .....	36
<b>Figure 3.22</b> les images d'apprentissage (en haut), les images tournées à 330 degrés (en bas).....	37
<b>Figure 3.23</b> auto-encodeur convolutif .....	37
<b>Figure 3.24</b> résultat de jeu nous avons utilisé 100 Epoch .....	38
<b>Figure 3.25</b> courbe graphique représentant les changements de (accuracy ,loss , mse) lors de l'utilisation epoch 100 .....	39
<b>Figure 3.26</b> résultat de jeu nous avons utilisé 150 Epoch .....	40
<b>Figure 3.27</b> courbe graphique représentant les changements de (accuracy ,loss , mse) lors de l'utilisation epoch 150 .....	40
<b>Figure 3.28</b> résultat de jeu nous avons utilisé 50 Epoch .....	41
<b>Figure 3.29</b> courbe graphique représentant les changements de (accuracy ,loss , mse) lors de l'utilisation epoch 50 .....	41
<b>Figure 3.30</b> résultat de jeu nous avons utilisé 200 Epoch .....	42
<b>Figure 3.31</b> courbe graphique représentant les changements de (accuracy ,loss , mse) lors de l'utilisation epoch 200.....	43
<b>Figure 3.32</b> la différence après avoir appliqué 100 époques et 150 époques.....	43
<b>Figure 3.33</b> la différence après avoir appliqué 50 époques et 200 époques.....	44

## Liste de Tableaux

**Tableaux 1.1** : les différences entre ANN et CNN.....10

**Tableaux 3.1** : Combaretion entre ANN et CNN.....44

## Notation ET Abréviation

**ANN**: Artificiel Neural Network.

**Tanh**: tangente hyperbolique.

**ReLU**: Unité de rectification linéaire.

**Logistique**: Les fonctions sigmoïdes.

**MLP**: le perceptron multicouche.

**CONV**: convolution.

**POOL**: Pooling.

**FC**: entièrement connectée.

**LOSS**: la perte.

**FFT**: Fast Fourier Transform

**MI**: Mutual Information

**GI**: transformation geometric

**MSE**: Mean Squared Error

**SSIM** : Mesure de similarité

**TF**: Transform Fourier

**2D**:Two-Dimension

**3D**: Three-Dimension

**CNN**: convolution nouerons net work

**IRM** : image par résonance magnétique

**CI** : computer zed tomographie

# **INTRODUCTION GENERALE**

## **1 Introduction**

Les travaux menés au cours de ce mémoire s'inscrivent dans le cadre de l'alignement d'images comme méthode de mise en correspondance de deux ou plusieurs images. Regarder la même scène à différents moments, de différents points de vue provoque un déplacement de cette même scène. Une étape primordiale avant n'importe quel traitement qui consiste à déplacer ou modifier une image pour qu'elle ressemble à une autre, doit être effectué. Cette étape est l'étape d'alignement d'images. Les algorithmes d'alignement sont nombreux et variés. En effet, la multiplicité des approches, des inspirations, des formalités, des applications et des évaluations, due en partie au manque relatif de comparaison entre les algorithmes, peut être un frein.

## **2 Problématique**

Dans ce projet, nous nous intéressons aux algorithmes d'alignement basés sur les réseaux de neurones convolutifs (Convolutional Neural Network CNN). L'une des formes les plus impressionnantes d'architecture ANN est le réseau neuronal convolutif (CNN). Les CNN sont principalement utilisés pour résoudre des tâches difficiles de traitement des images. Pour montrer l'intérêt et l'avantage de CNN, nous implémentons également les algorithmes d'alignement conventionnels basés sur les mesures statistiques, à savoir l'algorithme d'alignement par corrélation croisée et l'algorithme d'alignement par mesure d'information mutuelle. Nous implémentons également l'algorithme d'alignement par transformation géométrique.

## **3 Organisation du manuscrit**

Nous avons organisé notre travail en trois chapitres en plus de cette introduction:

- Dans le premier chapitre: nous présentons les réseaux de neurones artificiels et convolutifs. Nous mettrons l'accent sur la différence entre CNN et ANN.
- Le deuxième chapitre est consacré à la description des algorithmes d'alignement d'images. Plus précisément, nous détaillerons l'algorithme d'alignement à base de corrélation croisée, l'algorithme d'alignement à base d'information mutuelle et l'algorithme d'alignement à base

de transformation géométrique. Nous introduisons l'algorithme d'alignement à base de CNN à la fin de ce chapitre.

- Dans le troisième chapitre: nous présenterons la partie expérimentale de notre travail. Nous détaillerons les principales étapes d'implémentation des algorithmes d'alignement considérés. Les différents algorithmes d'alignement sont testés sur plusieurs jeux de données. L'algorithme d'alignement à base de CNN est testé sur la base de données MNIST (des chiffres écrit à la main). C'est une base très utilisée en apprentissage automatique. Une étude comparative quantitative et qualitative sera établie entre les différents algorithmes d'alignement. L'évaluation quantitative sera basée sur deux critères : l'erreur quadratique moyenne (MSE) et la mesure de similarité.

Enfin nous terminerons ce manuscrit par une conclusion générale.

## Chapitre 1

### Concepts de Base des Réseaux de Neurones Convolutifs

---

#### **Résumé**

*Dans ce Chapitre, nous allons d'abord présenter les réseaux de neurones artificiels conventionnels puis les réseaux convolutifs ainsi que leur fonctionnement. Enfin, nous discutons la différence entre ANN et CNN.*

---

#### **Sommaire:**

---

<a href="#">1.1</a> Introduction .....	3
<a href="#">1.2</a> Réseaux de neurones artificiels.....	3
<a href="#">1.3</a> Réseaux de neurones convolutifs.....	6
<a href="#">1.4</a> Réseaux de neurones utilisés.....	9

---

### 1.1 Introduction

Avec l'essor des réseaux de neurones artificiels (ANN), le domaine de l'apprentissage automatique a récemment subi d'énormes changements. Ces modèles informatiques inspirés de la biologie, peuvent dépasser de loin les performances des formes précédentes d'intelligence artificielle dans les tâches courantes d'apprentissage automatique. L'une des formes les plus impressionnantes d'architecture ANN est le réseau neurones convolutif (CNN). Les CNN sont principalement utilisés pour résoudre des tâches difficiles de reconnaissance de formes basées sur des images [1]. En raison de leur architecture simple et précise, ils offrent un moyen simplifié.

Dans ce Chapitre, nous allons d'abord présenter les réseaux de neurones artificiels puis les réseaux convolutifs ainsi que leur fonctionnement. Enfin, nous discutons la différence entre ANN et CNN.

### 1.2 Réseaux de Neurones Artificiels

#### 1.2.1 Principe de Base

L'unité de calcul de base du réseau de neurones artificiels est définie dans [2] en 1959. Les neurones artificiels reçoivent d'autres neurones ou provenant de sources externes avec des valeurs  $x_1, x_2, \dots, x_i$ . La synapse se connecte et calcule-la sortie  $y$ . Chaque entrée  $x_i$  a un poids associé  $w_i$ . Il est attribué en fonction de l'importance relative par rapport aux autres éléments. La valeur d'entrée  $x$  du neurone correspond à la somme pondérée de son entrée, L'autre élément de poids  $b$  est appelé biais. Ensuite, le neurone applique une fonction  $f$  (fonction d'activation) sur cette somme, comme le montre la Figure 1.1.

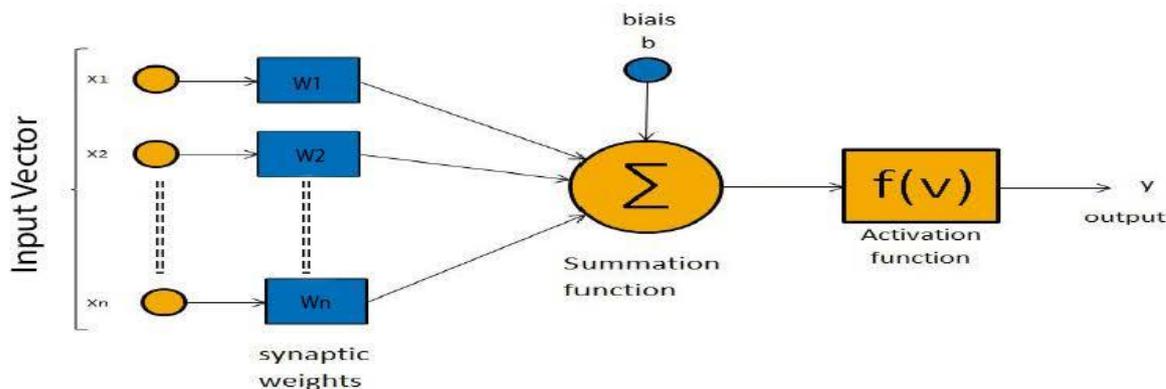


Fig 1.1: Modélisation d'un neurone artificiel [3].

La sortie du neurone  $y$ , appelée activation de sortie,  $f$  la fonction d'activation

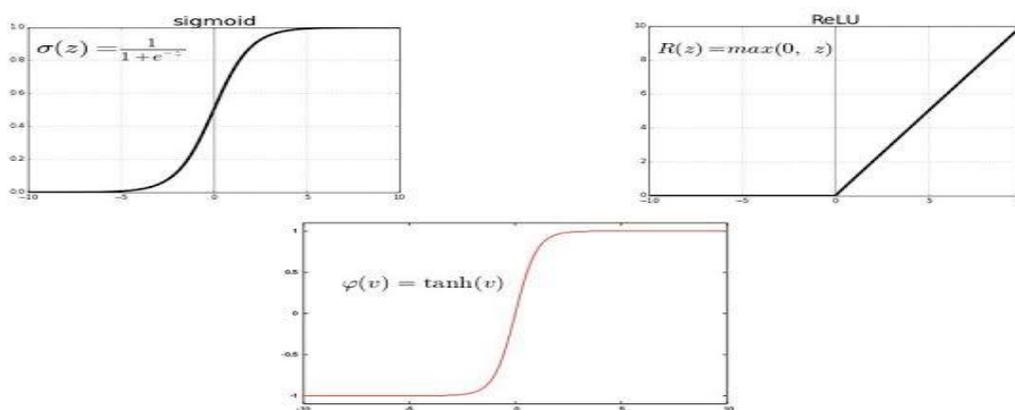
### 1.2.2 Fonction d'Activation

La fonction d'activation est une fonction mathématique utilisée pour un signal émis par un neurone artificiel. L'expression « fonction d'activation » vient du terme biologique « potentiel activation », un seuil de stimulation qui, s'il est atteint, provoque la réponse d'un neurone.

La fonction d'activation est fréquemment une fonction non linéaire. L'utilisation de fonctions d'activation non linéaires est tout simplement indispensable pour la simple et bonne raison que les fonctions linéaires ne fonctionnent qu'avec une seule couche neuronale. En ajoutant un biais, le neurone calcule le produit scalaire entre les signaux d'entrée et le vecteur de poids. Le résultat est utilisé pour activer la fonction. Suite à l'opération de produit scalaire, la fonction d'activation est utilisée pour introduire une opération de non-linéarité. Cette non-linéarité permet plusieurs variations d'état internes sur la même classe d'objet [4].

Or, le choix d'une fonction d'activation s'avère être un élément clé des réseaux neuronaux. Cette fonction a un impact significatif sur les performances du réseau. Dans un réseau de neurones, il est donc essentiel de choisir le bon type de fonction d'activation. Diverses fonctions ont été utilisées au fil des ans. Seuls les plus importants ont été mentionnés dans cette section. Les fonctions sigmoïdes (logistique), tangente hyperbolique (Tanh) et activation unité de rectification linéaire (ReLU) sont les plus couramment utilisées comme c'est illustré dans la Figure 1.2 [4].

## Fonction d'activation



10

Fig.1.2 : Fonctions d'activation couramment utilisées [5].

- **L'algorithme de la rétro-propagation du gradient (back-propagation)**

Les réseaux de neurones sont des outils fortement non-linéaires qui utilisent généralement des techniques itératives pour l'apprentissage [4]. IL existe différents algorithmes, mais les techniques les plus conseillées pour l'apprentissage des réseaux de neurones sont les algorithmes de gradient descendant avec taux d'apprentissage variable, Rétro-propagation, L'algorithme de Fletcher-Reeves, algorithme de Quasi-Newton et l'algorithme de Levenberg-Marquardt [6]. Ces méthodes fournissent généralement de meilleurs résultats que les algorithmes classiques comme la descente de gradient mais ils sont également plus gourmands en termes de mémoire et nécessitent davantage de calculs.

L'apprentissage se conduit avec une méthodologie précise. La première, est la plus simple, connue sous le nom de règle de Hebb, indique que lorsque deux neurones sont excités en même temps, il faut modifier les coefficients synaptiques pour renforcer cette excitation simultanée. La règle de Widrow-Hoff [6] établie en 1960, indique que la modification de ses coefficients est proportionnelle à l'erreur entre le résultat souhaité, le résultat réel et aux valeurs d'entrée. L'algorithme de rétro-propagation du gradient suit la démarche de

Widrow-Hoff pour définir une notion d'erreur sur un exemple puis calcule la contribution de cette erreur de chacun des poids synaptiques.. A partir de cette erreur, est calculé le gradient qui à son tour est propagé de la couche de sortie vers la couche d'entrée, d'où le terme de rétro-propagation. Cela permet la modification des poids du réseau et donc

l'apprentissage. L'opération est répétée pour chaque vecteur d'entrée et cela jusqu'à ce que le critère d'arrêt soit vérifié [6].

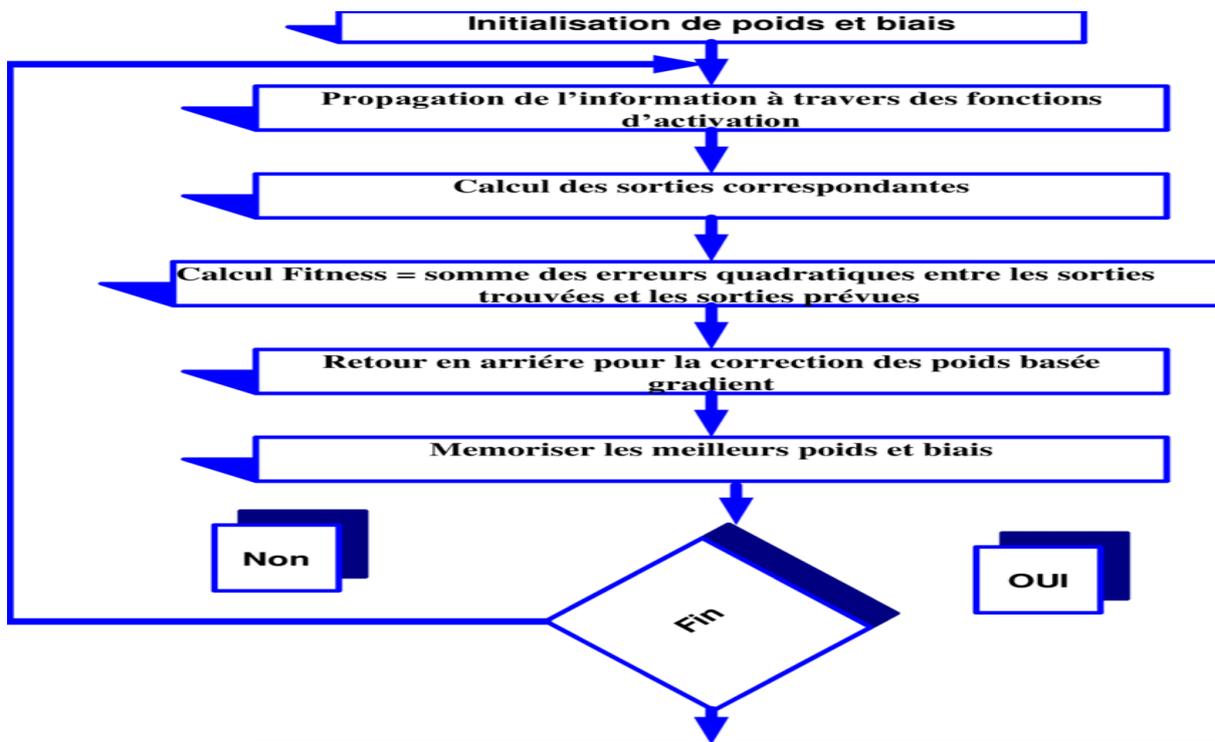


Fig1.3 : Le principe général de l'algorithme de la retro-propagation de gradient [7].

### 1.2.3 Apprentissage d'Un réseau de Neurones

La phase d'apprentissage consiste à ajuster les poids du réseau de neurones afin d'obtenir les meilleurs résultats de régression ou de classification. Cibler Le but de cette optimisation est de minimiser la fonction de coût, qui calcule l'erreur entre les deux données d'apprentissage et données prédites par le réseau de neurones.

## 1.3 Réseaux de Neurones Convolutifs (CNN)

### 1.3.1 Architecture de Réseaux de Neurone convolutifs

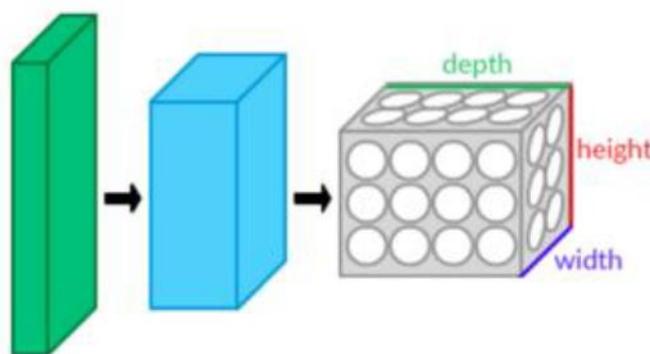
Les réseaux de neurones convolutifs sont basés sur le perceptron multicouche (MLP) et s'inspirent du comportement du cortex visuel des vertébrés. Malgré son efficacité dans le traitement des images, les MLP peinent à gérer des images volumineuses, ce qui est dû à la croissance exponentielle du nombre de connexions proportionnellement à la taille de l'image.

Par exemple, si nous prenons une image  $32 \times 32 \times 3$  (32 de large, 32 de haut, 3 canaux de couleur), un seul neurone entièrement connecté dans la première couche cachée du MLP aurait 3072 entrées ( $32 \times 32 \times 3$ ). Du coup, une image  $200 \times 200$  pourrait traiter 120 000 entrées par neurone, ce qui est énorme quand on le multiplie par le nombre de neurones. CNN vise à minimiser le nombre d'entrées dans son ensemble. Le CNN vise à réduire le nombre d'entrées tout en maintenant un niveau élevé de "corrélacion spatiale locale" entre les images naturelles. Contrairement au MLP, le CNN présente les caractéristiques distinctives suivantes :

**1. Volumes 3D de Neurones** est le premier. La couche neuronale n'est plus seulement une surface mais a évolué en un volume avec une profondeur. Si l'on considère un seul récepteur CNN, les  $n$  neurones associés forment l'équivalent d'une première couche de MLP.

**2. Connectivité locale** en utilisant un récepteur de champ qui limite le nombre d'entrées qu'un neurone peut avoir tout en maintenant l'architecture MLP, CNN s'assure que les « filtres » fournissent la réponse la plus forte à un motif d'entrée localisé dans l'espace, résultant en une représentation de l'entrée. Une représentation comme celle-ci prend moins d'espace mémoire. De plus, comme le nombre de paramètres à estimer est réduit, leur estimation est plus pour un volume de données fixe.

**3. Poids partagés** Dans les CNN, les paramètres de filtrage d'un neurone sont les mêmes pour tous les autres neurones du même noyau. Ce paramètre est précisé dans une "carte de fonction". La Figure 1.3 illustre une couche CNN en trois dimensions. [8]



**Fig. 1.4 :** Une couche du CNN en 3 dimensions. (Vert = volume d'entrée, bleu = volume du champ récepteur, gris = couche de CNN, cercles = neurones artificiels indépendants)

### 1.3.2 Différents Modules d'un Réseau de Neurones Convolutif :

À la lumière de ce cadre général, nous allons maintenant nous concentrer sur les différents modules couramment utilisés dans les réseaux CNN.

- **Couche de convolutive (CONV) :**

La couche convolutive est le composant le plus important de toute architecture CNN. Il contient un ensemble de noyaux convolutifs (également appelés filtres), qui sont convolués avec l'image d'entrée (métriques à N dimensions) pour générer une carte des caractéristiques.

- **Couche de pooling (POOL)**

Après chaque couche convolutive, il peut y avoir une couche de mise en commun. Cette couche est sous-échantillonnée le long de la dimension spatiale d'une entrée donnée, ce qui réduira encore le nombre de paramètres dans l'activation. Il existe plusieurs façons d'effectuer cette mise en commun, comme la prise de valeur moyenne ou maximale, ou la combinaison linéaire de neurones dans le bloc., La Figure 1.5 montre la mise en commun maximale sur une fenêtre  $2 \times 2$  [9].

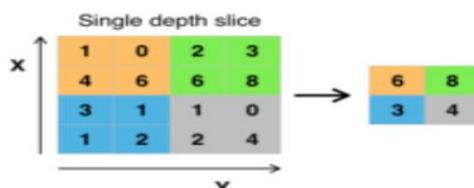


Fig 1.5 : max Pooling avec un filtre 2x2 et un pas de 2.[10]

- **Couches de correction (RELU)**

Il est possible d'améliorer l'efficacité du traitement en intercalant une couche entre les couches de traitement qui effectuera une fonction mathématique (fonction d'activation) sur les signaux de sortie.

La fonction ReLU (abréviation de Unités Rectifié linéaires) :  $F(x) = \max(0, x)$  Cette fonction force les neurones à renvoyer des valeurs positives.



- Couches entièrement connectée (FC)

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones d'un canapé (module) entièrement connecté sont connectés à toutes les sorties du canapé précédent. Leurs fonctions d'activation peuvent ainsi être calculées à l'aide d'une multiplication matricielle suivie d'un décalage de polarisation.

- Couches de perte (LOSS)

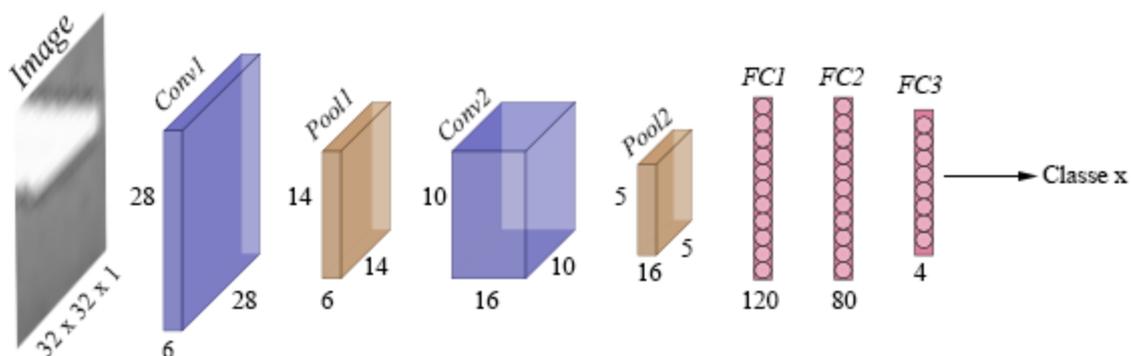
La couche de perte spécifie comment l'entraînement du réseau pénalise la différence entre le signal attendu et le signal réel. Normalement, elle est le dernier canapé du réseau. Il est possible d'utiliser une variété de fonctions de perte adaptées à des tâches spécifiques. Softmax est une fonction qui calcule la distribution de probabilité sur les classes de sortie [11].

## 1.4 Réseaux de Neurones Utilisés

Dans cette section, les réseaux de neurones convolutifs utilisés dans sont présentés.

### 1.4.1 LeNet

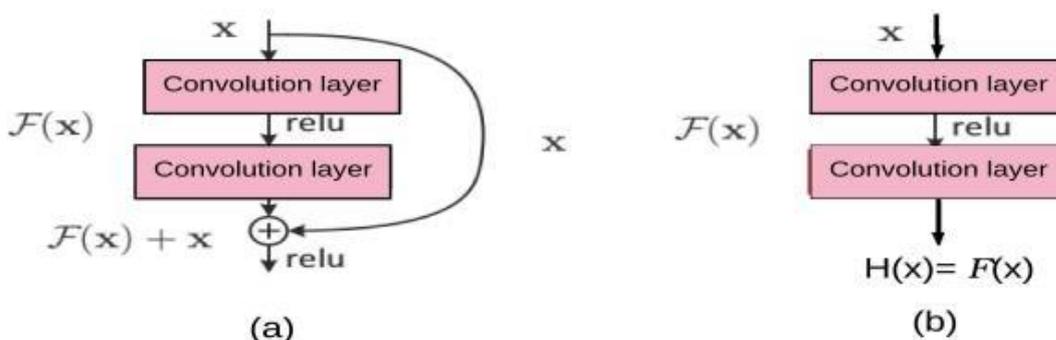
LeNet est l'une des plus anciennes architectures CNN et constitue la page d'accueil pour la classification des caractères manuscrits. Il a été introduit par LeCun et al. en 1998 [12]. Le Net est illustré dans la Figure 1.6. Il contient deux blocs constitués d'une couche d'Enveloppement suivi d'une couche pooling, deux canapés entièrement Similaire aux visualisations en couches classiques.



**Fig. 1.6:** Architecture du réseau de neurones LeNet pour un exemple de classification en 4 classes météorologiques (adapté de [13]).

### 1.4.2 ResNet

En raison du problème de décroissance du gradient dans le modèle CNN profond, He et al de Microsoft ont proposé le modèle ResNet et introduit l'idée de "connexion de saut d'identité" pour résoudre le problème de décroissance du gradient. L'architecture de ResNet utilise le mappage résiduel ( $H(x) = F(x) + x$ ) au lieu d'apprendre l'application directe ( $H(x) = F(x)$ ), ces blocs sont appelés blocs résiduels. L'architecture ResNet complète se compose de nombreux blocs résiduels. Couche convolutive  $3 \times 3$ . La Figure 1.7 illustre la différence entre la cartographie directe et la cartographie résiduelle.



**Fig. 1.7 :** Architecture du réseau ResNet [14].

### 1.5 Différence entre les réseaux ANN et CNN [15].

Nous résumons les principaux différents entre l'ANN et CNN dans tableau 1.1

**Tableau 1.1** différence entre ANN et CNN

	ANN	CNN
<b>Type de données</b>	Données tabulaires, données textuelles	Données d'images
<b>Partage de paramètres</b>	Oui	no
<b>Entrée de longueur fixe</b>	Oui	oui

<b>Connexions récurrentes</b>	No	no
<b>Dégradé de disparition et d'explosion</b>	oui	oui
<b>Relation spatiale</b>	No	oui
<b>Performance</b>	ANN est considéré comme moins puissante que CNN	CNN est considéré comme moins puissant qu'ANN
<b>Application</b>	Reconnaissance faciale et vision par ordinateur	Reconnaissance facial, numérisation de texte et traitement du langage naturel
<b>Principaux avantage</b>	Avoir une tolérance aux pannes, Capacité à travailler avec des connaissances incomplètes	Haute précision dans les problèmes de reconnaissance d'images, partage du poids
<b>Désavantages</b>	Dépendance matérielle, comportement inexplicable du réseau	Grandes données d'entraînement nécessaires, ne pas encoder la position et l'orientation de l'objet

## 1.6 Conclusion

En conclusion, il existe des cas spécifiques dans lesquels le ANN pourrait être préféré au CNN, et vice versa. Ils sont tous deux uniques dans leur façon de travailler mathématiquement, ce qui leur permet d'être meilleurs pour résoudre des problèmes spécifiques.

En général, CNN a tendance à être un moyen plus puissant et plus précis de résoudre les problèmes de classification et d'alignement. Le CNN reste dominant pour les problèmes où les ensembles de données sont limités et où les entrées d'images ne sont pas nécessaires.

Les CNN sont la solution plus répandue pour les problèmes de vision par ordinateur et d'apprentissage automatique qui dépendant de l'image

## Chapitre 2

### *Algorithmes Conventionnels d'Alignement d'Images*

---

#### **Résumé**

Dans ce chapitre nous aborderons les principes d'alignement. Nous présenterons les différents algorithmes d'alignement d'image : les algorithmes classiques (Corrélation Croisé, l'information mutuelle, transformation géométrique). Nous allons introduire les réseaux de neurones convolutionnels (CNN) les critères d'évaluation.

---

#### **Sommaire:**

---

2.1 Introduction .....	13
2.2 Principe d'Alignement d'Images.....	13
2.3 Algorithmes classiques d'Alignement d'Image .....	14
2.4 Algorithme de L'image a base de CNN .....	20
2.6 Conclusion.....	22

---

## 2.1 Introduction

L'alignement d'images est un problème récurrent en vision par ordinateur qui consiste à aligner une image dite cible pour qu'elle soit en correspondance avec une image source. Le but de cette méthode est de minimiser l'écart entre les structures similaires d'une scène afin de compenser les différences spatiales entre eux qui sont dues aux systèmes d'acquisition ainsi qu'au sujet étudié. Il s'agit alors de trouver la transformation géométrique qui permette de mieux les superposer et ainsi d'en faciliter l'interprétation.

Dans ce chapitre nous allons participer à l'état de l'art relatif à l'alignement des images en étudiant le problème d'alignement, en présentant les méthodes d'alignement qui sont basées sur des mesures statistiques

## 2.2 Principe d'Alignement d'Images [16] [17] [24] [22]

On fait recours à un algorithme d'alignement pour des nombreuses applications de traitement automatique d'image et à chaque fois qu'on veut combiner les informations fournies par deux images obtenues à différents instants, par différents points de vue. Le résultat de l'alignement peut être donné soit sous la forme d'un vecteur de paramètres caractérisant la transformation qui met en commun les informations fournies par une ou plusieurs sources à différents instants soit sous celle d'une image résultante de l'application.

Pour bien expliquer le principe d'alignement d'image, soient images  $I_1$  et  $I_2$ . Mais on trouve qu'il y a un problème d'alignement qui consiste à déterminer la transformation qui aligne parfaitement ces deux images. Les étapes importantes d'un algorithme d'alignement s'annoncent tels : quelles informations doivent être utilisées pour guider l'alignement ? Comment pouvez-vous déterminer à quel point deux images sont similaires ? Comment déformer une image ? Quelle est la meilleure façon de trouver la meilleure déformation ?

Ces questions susmentionnées conduisent au formalisme suivant :

L'alignement consiste à déterminer une transformation  $T$  comme :

$$T = \arg \max S(f_1(I_1), f_2(T(I_2))) \quad (2.1)$$

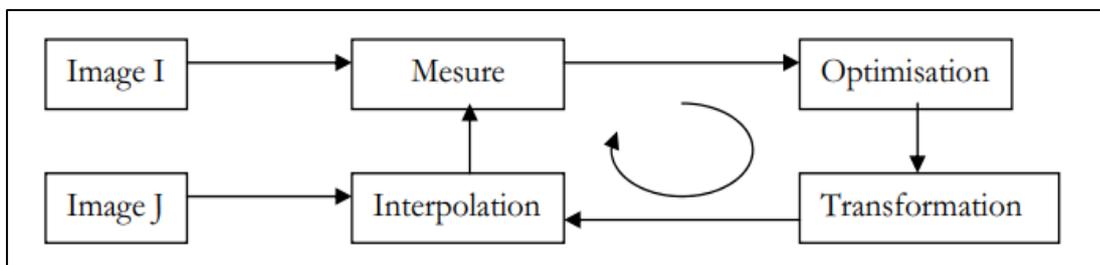
- $I_2 \approx (I_2)$ : La transformation ou déformation doit être la différence entre les fonctions  $I_1$  et  $T(I_2)$ . soit la plus faible possible en fonction d'une mesure donnée de dissemblance entre deux signaux. D'autre part, cette transition devrait également être degré maximal de similitude entre les fonctions  $I_1$  et  $T(I_2)$ .
- $T$  doit être suffisamment régulier pour que les caractéristiques  $I_2$  soient respectées.
- Comme nous le verrons dans l'équation,  $f_1$  et  $f_2$  sont des fonctions qui extraient les caractéristiques de l'image
- $S$  est une fonction de similarité définie sur l'espace des caractéristiques extrapolées.
- « **Arg max** » la procédure d'optimisation de la mesure de similarité, recherchant le maximum dans l'espace de recherche défini par les paramètres de  $T$ , si au contraire  $S$ .

Notons que comme mesure de dissemblance, cette procédure sera « arg min », recherchant la plus petite valeur dans l'espace de recherche défini par les paramètres de  $T$ .

Lors de l'alignement d'un ensemble de  $m$  signaux, dans notre cas  $m$  images d'une série d'inclinaison ( $I_1, I_2 \dots, I_m$ ), les symboles  $I_1$  et  $I_2$  représentent une paire d'images successives de la série d'inclinaison,  $I_1$  est l'image de référence.

Le but est de trouver  $m$  transformations (**translation ou rotation**) suffisamment régulières pour que les fonctions déformées (**images alignées :  $T(I_1)$  et  $T(I_2)$** ) soient alignées de manière optimale par rapport à une quantité particulière de données. Dans les prochains paragraphes, nous passerons en revue les différentes méthodes

La figure suivante représente le traitement d'alignement dans lequel les images  $I$  et  $J$  sont respectivement cible et l'image source.



**Figure 2.1.** Processus d'alignement [22]

### 2.3 Algorithmes classiques d'Alignement d'Image

Les algorithmes classiques d'alignement peuvent être classés selon leurs complexités comme suit :

- Algorithme de Corrélation Croisée
- Algorithme basé sur l'Information Mutuelle.
- Algorithme basé sur Transformations géométriques.

### 2.3.1 Algorithm de Corrélation Croisée [16] [24]

Formellement, on note  $I_j, y$  la valeur du pixel en coordonnées spatiales  $(x, y)$

Pour les valeurs complètes de  $x=0 \dots (k-1)$  et  $y = 0 \dots, (k-1)$  dans les images d'une série d'inclinaison de  $\mathbf{J}$  images  $(L-1) (L-1)$  et  $0 \leq x < k$  et  $0 \leq y < L$

La transformée de Fourier  $F(I_j, y)$  de  $I_j, y$  est alors :

$$f(I_j, X_y) = \sum_{x=0}^{k-1} \sum_{y=0}^{l-1} I_j(x, y) \exp(-i2\pi(xk/k + yl/l)) \quad (2.2)$$

Et la transformée de Fourier inverse est :

$$F^{(-1)}(I_j, X_y) = 1 / Kl = \sum_{x=-k/2}^{k/2-1} \sum_{l=-l/2}^{l/2-i} I_j^{(l/2-i)}_{(l-l/2)}(x, y) \exp(-i2\pi(xk/k + yl/l)) \quad (2.3)$$

$i$  Dénote  $\sqrt{-1}$

Et :

$$(\omega_1, k, \omega_2, l)T = 2\pi(K k, L l)T \quad (2.4)$$

Les traductions dans le domaine spatial sont équivalentes à des différences de phase dans le domaine de Fourier même si les images sont désalignées en raison des traductions.

La phase spectrale sera des fonctions linéaires. Si l'image  $I_j$  est désalignée par rapport à l'image  $I_i$  avec une translation de  $\mu \mathbf{j} \equiv (\mu_1 \mathbf{j}, \mu_2 \mathbf{j}) T$  pixels (i.e.  $\mu_1 \mathbf{j}$ ) dans la direction  $\mathcal{X}$  et  $\mu_2 \mathbf{j}$  dans la direction  $y$ ), alors le désalignement entre les deux images se produit.

La direction des  $\mathcal{X}$  et  $\mu_2 \mathbf{j}$  dans la direction des  $y$ , donc le désalignement entre les deux images  $I_i$  et  $I_j$  sera :

$$\mu i j \equiv (\mu i - \mu j)T \quad (2.5)$$

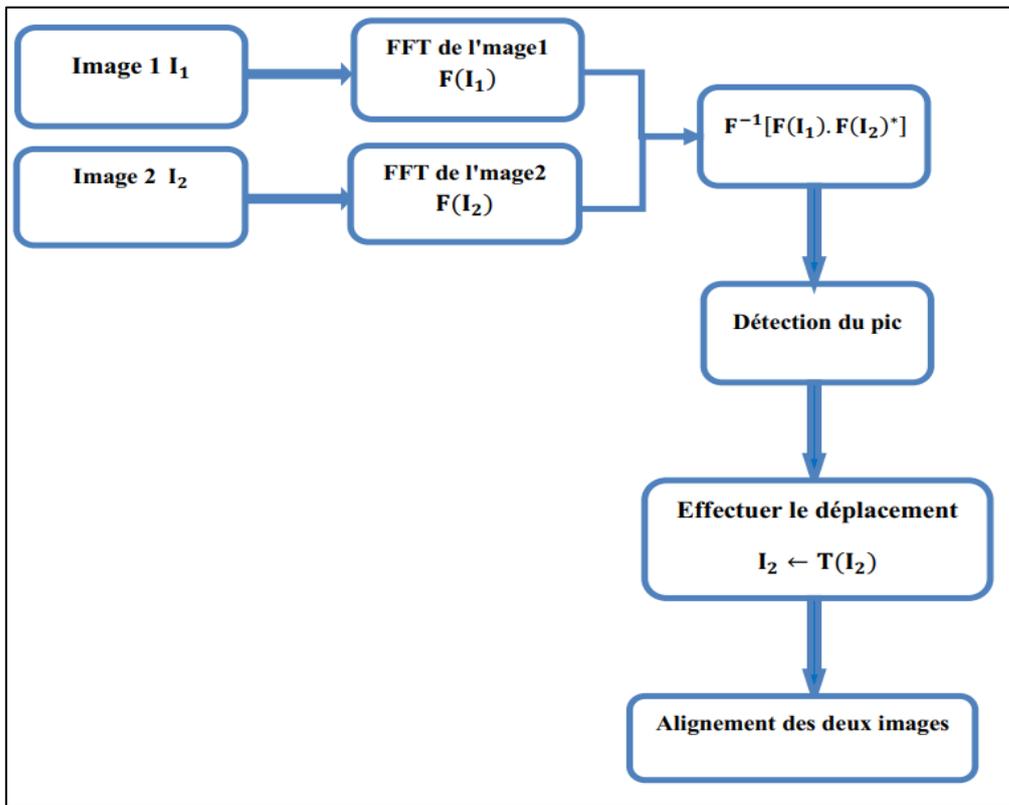
Et :

$$\phi_{i,j,k,l} = \omega_{kl} T(\mu_i - \mu_j) T \text{ mod } 2\pi \quad (2.6)$$

La fonction de corrélation croisée entre deux images  $I_i$  et  $I_j$  est donnée par la formule suivante :

$$c(x,y) = \sum_s \sum_t I_i(s,t) I_j(x+s,y+t) \quad (2.7)$$

La Figure 2.2 illustre l'algorithme d'alignement d'images par corrélation croisée.

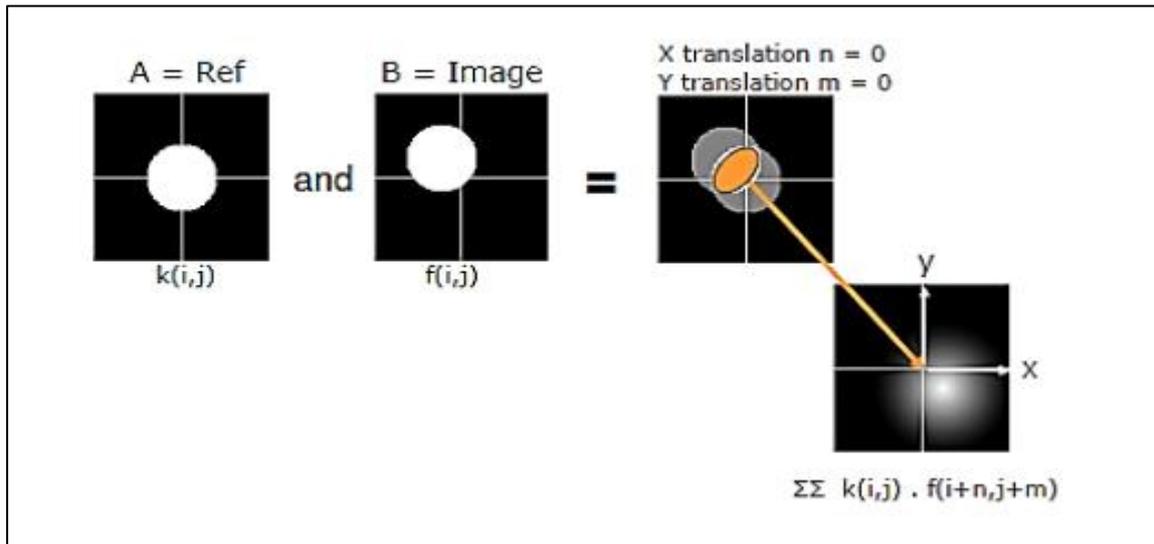


**Figure 2.2.**Algorithme d'alignement des images par corrélation croisée.[16]

L'algorithme d'alignement d'inter corrélation entre deux images  $I_i$  et  $I_j$  réside dans la recherche du vecteur  $ij$  pour lequel la fonction d'inter corrélation renvoie une image. Il y a deux images,  $I_1$  et  $I_2$ , que nous voulons aligner de la même manière.

La première étape consiste à calculer les transformées de Fourier  $F(I1)$  et  $F(I2)$  des deux variables. En prenant l'inverse de Fourier du produit de  $F(I1)$  par le cumulé de  $F(I2)$ , nous avons une nouvelle image. Sur cette image, il suffit de détecter le point du plus grand pic d'intensité :

Pour bien expliquer cette méthode on illustre un exemple d'alignement de deux images par corrélation croisée dans la figure suivante :



**Figure 2.4.** Exemple d'alignement de deux images par corrélation croisée [16]

### 2.3.2 Algorithme basé sur l'Information Mutuelle [27] [23] [19] [20]

L'information mutuelle qui est un outil de la théorie de l'information mesure la quantité d'information qu'une image donne de l'autre. En fait, on appelle information mutuelle entre deux événements I et J la quantité mentionnée par la formule suivante :

$$IM(I;J) = h(I) - h\left(\frac{I}{J}\right) \quad (2.8)$$

Où h est la mesure de l'incertitude d'un événement que s'exprime par :

$$h(E) = -\log P\{E\} \quad (2.9)$$

Avec  $P\{E\}$  et la probabilité de l'évènement

Selon l'équation (2.8) l'information apportée par J sûr I est la diminution de l'incertitude sur lorsque J est réalisés peut déduire que :

$$IM(I;J) = h(I) + h(J) - h(I \cap J) \quad (2.10)$$

Cette expression nous permet de dire que :

$$IM(I;J) = IM(I;J) \quad (2.11)$$

Notons que si  $I$  et  $J$  Independent donc :

$$IM(I;J) = 0 \quad (2.12)$$

Plus concrètement, pour une image de référence  $I$  et une image flottante  $J$ , l'information mutuelles' exprime par :

$$IM(I;J) = H(I) + H(J) - H(I,J) \quad (2.13)$$

Avec  $H(X)$  l'entropie de  $X$  elle est donne par :

$$H(X) = - \sum P(x) \log P(x) \quad (2.14)$$

Et on peut exprimer l'information mutuelle par :

$$IM(I;J) = \sum P_{ij} \cdot \log \frac{P_{ij}}{P_i P_j} \quad (2.15)$$

Où  $p_{ij}$  (respectivement  $p_i, p_j$ ) est la probabilité de trouver un point de l'image  $I$  (respectivement l'image  $J$ ) ayant l'intensité  $i$  (respectivement l'intensité  $j$ ) , tandis que  $p_i, p_j$  est la probabilité trouver un point de l'image  $I$  ayant l'intensité  $i$  et que son homologue dans l'image  $J$  à l'intensité  $j$ . On appelle aussi  $p_{ij}$  l'histogramme conjoint.

Pour avoir la loi de probabilités, deux grandes méthodes existent :

- ✓ la méthode basée sur une loi de probabilité discrète.
- ✓ la méthode avec estimation des fonctions de densité de probabilité.

La première méthode s'appuie sur l'histogramme conjoint normalisé. Ainsi pour chercher une solution, il faut une optimisation stochastique. En effet, toute la formulation du problème de recalage se trouve principalement dans le domaine discret, ainsi on est en présence d'une fonction objective non dérivable. Cette méthode est très lente mais beaucoup plus précise.

Par exemple, si deux photos représentent la même structure anatomique, une image contiendra plus d'informations que l'autre. Il y aura une information mutuelle entre les deux images, ainsi que l'utilisation d'intensités différentes. Les deux images seront transformées en une

transformation de distribution d'entropie. Lorsqu'elles sont combinées, la valeur d'entropie reste la même et différentes intensités sont utilisées. Maximisation de l'information n'augmentera, ni diminuera la quantité d'information. De ce fait, l'information mutuelle est bien adaptée au recalage multimodal.

### **2.3.3 Algorithme basés sur Transformations géométriques [22] [25]**

Durant la phase d'acquisition, plusieurs problèmes peuvent survenir, particulièrement, le mouvement du spécimen. Deux sortes de mouvements peuvent être considérées lors du déplacement du spécimen : la translation et la rotation en 3D.

#### **2.3.3.1 Translation en 3D [19]**

Rappelons que le repère d'acquisition du microscope est constitué d'un champ électronique perpendiculaire (OX ; OY) et d'un axe focal (OZ). L'image peut être déplacée dans (OX ; OY) ce qui conduit au même résultat.

Se déplaçant sur le plan du détecteur (i.e. image de projection). Le déplacement de l'image en fonction de (OZ) affectera l'échelle de l'objet projeté dans l'image de projection. En d'autres termes, la translation 3D de l'objet affecte en même temps la translation et la mise en échelle 2D de l'image de projection. Nous commençons notre approche par une estimation de l'objet 3D. De ce fait, nous pouvons corriger la translation et la mise en échelle des images de projections directement avec la translation sur l'objet 3D. Dans l'espace continu, la translation 3D est définie comme suit :

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}_{(O'X'Y'Z')} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}_{(OXYZ)} \quad (2.19)$$

Où  $(x ; y ; z ; 1) T$  et  $(x_0 ; y_0 ; z_0 ; 1) T$  sont respectivement le système de coordonnées  $(O X Y Z)$  et  $(OX_0Y_0Z_0)$ .

#### **2.3.4 Rotation en 3D [23]**

Il est possible que l'objet tourne autour du plan du détecteur  $(OX; OY)$  lorsque l'angle d'inclinaison est augmenté. Ces deux types de rotations peuvent être présentés par le système des trois angles d'Euler  $(\alpha; \beta; \gamma)$ .

En utilisant le même principe mathématique évoqué dans la section précédente, nous pouvons également modéliser la rotation 3D. En effet, au lieu d'appliquer la rotation sur l'objet lui-même, nous allons l'appliquer directement sur les lignes de projections. À cet effet, nous utilisons une matrice  $\alpha\gamma W$  de taille  $M \times N$ , laquelle projette le vecteur de l'objet  $\mathbf{f}$  vers le vecteur de projection  $\mathbf{p}_i$  suivant l'orientation décrite par le triplet  $(\alpha_i; \beta_i; \gamma_i)$  comme suit :

$$P = \omega(\alpha_i + \beta_i + \gamma_i)f \quad (2.20)$$

Comme la matrice de projection utilisée dans le cas 2D, les coefficients de chaque ligne de cette matrice  $W$  sont les poids de projection des voxels qui interagissent avec le faisceau de projection qui lui correspond. Il faut mentionner que, même si les deux matrices  $T$  et  $W$  sont énormes, elles sont creuses.

#### **2.4 Algorithme de L'image à base de CNN [28] [29]**

Les réseaux neuronaux convolutifs ou ConvNets sont conçus pour traiter des données qui se présentent sous la forme de tableaux de valeurs en  $N$  dimensions pour  $N \in \mathbb{N}^{+*}$ . Par exemple, une image couleur se compose de trois tableaux 2D contenant des intensités de pixels dans les trois canaux de couleur RVB (rouge, vert, bleu). Mais de nombreux autres types de données se présentent sous la forme de tableaux à multiples dimensions :

- 1D pour les signaux et les séquences, y compris la langue.
- 2D pour images ou spectrogrammes audios.
- 3D pour les images vidéo ou volumétriques.

Le principe des ConvNets repose sur quatre idées clés qui exploitent les propriétés des signaux naturels :

- ✓ Les connexions locales.
- ✓ Les poids partagés.
- ✓ La couche de regroupement (pooling) facultative.

L'architecture d'un ConvNet typique est structurée en une série d'étapes. Les premières étapes sont composées de deux types de couches: les couches convolutives et les couches de regroupement (pooling).

La couche de convolution est l'élément central des réseaux neuronaux convolutifs. Elle compose au minimum leur première couche. Son objectif est de détecter la présence de caractéristiques (features) dans les images d'entrée. Cela est réalisé grâce à un filtrage par convolution qui consiste à faire glisser une fenêtre représentative de la caractéristique sur l'image d'entrée et à calculer le produit de convolution entre la caractéristique et chaque portion de l'image balayée. Dans ce contexte, le concept de caractéristique est assimilé au filtre. Dans chaque couche convolutive, chaque filtre est répliqué sur tout le champ visuel. Ces unités répliquées partagent la même paramétrisation (vecteur de poids et biais), c'est à dire que les poids sont partagés, et forment une carte de caractéristiques (featuremap). Cela signifie que tous les neurones d'une même couche convolutive répondent aux mêmes caractéristiques. Cette réplication permet ainsi de détecter les caractéristiques quelle que soit leur position dans le champ visuel. C'est l'invariance par translation et c'est une caractéristique fondamentale des réseaux neuronaux à convolution. La couche de regroupement (*pooling*) se place entre les couches convolutives. Elle permet d'appliquer à chacune des cartes de caractéristiques une réduction de leur taille tout en préservant les caractéristiques les plus importantes (en ne gardant que les valeurs maximales par exemple). Elle permet ainsi de réduire le nombre de paramètres du réseau et donc les calculs nécessaires. Elle permet aussi de rendre le réseau moins sensible à la position des caractéristiques. La couche de convolution est caractérisée par trois hyperparamètre :

- ✓ la profondeur de la couche c'est à dire le nombre de noyaux de convolution(ou nombre de neurones associés à un même champ récepteur) ;
- ✓ le pas: il contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
- ✓ le remplissage à zéro (*zero padding*) : parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. Cela contrôle la dimension spatiale du volume de sortie.

Un réseau neuronal convolutif prend en entrée un signal  $x(u)$ , qui est dans notre cas une image. Une couche neuronale interne  $x_j(u, k_j)$  de profondeur  $j$  est indexée de la même variable

de translation  $u$ , habituellement sous-échantillonnée, et un indice de canal  $k_j$ . Une couche  $x_j$  est calculée à partir de  $x_{j-1}$  en appliquant un opérateur linéaire  $w_j$  suivi par une non-linéarité point par point  $\rho$ :

$$x_j = \rho W_j x_{j-1} \quad (2.21)$$

La non-linéarité  $\rho$  transforme chaque coefficient  $\alpha$  du tableau  $W_j x_{j-1}$ , et satisfait la condition de contraction.

## 2.5 Critères d'Evaluation

Dans ce travail nous avons utilisé deux critères d'évaluation pour savoir quelle est la meilleure méthode de l'alignement d'image.

### 2.5.1 Moyenne Erreur MSE (l'Erreur Quadratique)

Le MSE peut être écrite comme la somme de la variance de l'estimateur et le biais quadratique de l'estimateur, fournissant un moyen utile pour calculer le MSE [16]

$$MSE = \frac{1}{M} \sum (f_1 - f_2)^2 \quad (2.23)$$

Où  $f_1$  est l'image original  $f_2$  et l'image alignée et  $M$  la taille de l'image.

### 2.5.2 Mesure de similarité [26] [21]

Cette mesure est basée sur une combinaison du niveau de gris et de la distance euclidienne.

La mesure de similarité est l'élément le plus critique dans le processus du recalage. Ce critère correspond à la mesure de la ressemblance entre deux images.

## 2.6 Conclusion

Nous avons défini et détaillé l'alignement et ses problématiques, tel que dans un premier temps, consacré toute une partie à définir ce qu'est l'alignement, quel en est le principe mathématique de base ainsi qu'à la description détaillée des grandes classes de méthodes de l'alignement à partir des données : les méthodes basées sur des mesures statistiques et les méthodes basées sur CNN. nous avons constaté que les mesures statistiques sont largement exploitées pour résoudre le problème d'alignement, nous avons expliqué les critères d'évaluation (MSE et SSIM). Dans le chapitre qui suit, nous allons implémenter les

algorithmes d'alignement considérés. Une étude comparative sera établie entre les différents algorithmes.

## Chapitre 3

### *Implémentation des algorithmes d'alignement d'image*

#### **Résumé**

*Dans ce chapitre nous présenterons les résultats de*

*Simulation obtenus au cours de la réalisation de ce travail. Une étude comparative quantitative et qualitative sera établie entre les algorithmes d'alignement conventionnels d'alignement et l'algorithme d'alignement à base de CNN*

---

#### **Sommaire :**

---

3.1 Introduction .....	24
3.2 Résultats et Discussions.....	24
3.2.1 Nature de données et image teste.....	24
3.2.2 Matériel et software utilise .....	25
3.2.3 Algorithme conventionnels d'alignement d'images .....	25
3.3 Conclusion .....	44

---

### **3.1 Introduction**

Dans ce chapitre nous présentons les résultats expérimentaux obtenus dans le présent travail et les commenter tout en menant une étude comparative avec les résultats des méthodes d'alignement bien connus selon les critères d'évaluation adoptés, à savoir MSE et la mesure de similarité, nous expliquons l'intérêt des techniques étudiées pour l'alignement des images. L'accent est mis sur les méthodes d'alignement basées sur des mesures statistiques tel que la fonction de corrélation croisée normalisée (NCC) et l'information mutuelle (MI) et la méthode basée sur les CNN.

Notre objectif est d'illustrer les performances des méthodes présentées dans ce manuscrit, à cet effet nous traitons plusieurs jeux de données synthétiques par différentes méthodes.

### **3.2 Résultats et Discussions**

#### **3.2.1 Nature Des Données et images test**

Dans l'algorithme d'alignement par Corrélation Croisée, nous avons choisi des images niveaux de gris tirées de la bibliothèque Toolbox du Matlab, version R2018b, lena256 (format tif) de taille 255×255 pixels, Parrot (format tif) de taille 255x255pixels, Foreman (format tif) de taille 255x255 pixels.

La Figure III.1 illustre ces images originales, pour ce type d'images, et pour illustrer le problème d'alignement, nous générons d'abord les images désalignées, pour appliquer enfin les différentes procédures d'alignement.

Et nous avons également choisi une autre image en couleur covid\_19 (format jpg) de taille 150 x600 pixels. Figure 3.2 illustre ces images test.

Alors que dans l'algorithme d'alignement par transformation géométrique nous avons choisi une image niveaux de gris IRM.

La Figure 3.3 : Illustre ces images originales, IRM (format png) de taille 40 x 140.

Pour l'algorithme d'alignement par l'information mutuelle nous avons choisie deux images niveaux de gris IRM (format. Tif) de taille 230 X 230 et CT (format. Tif) de taille 512 X 512.

Et dans Algorithme d'alignement d'images à basse CCN nous avons choisie des images niveaux gris centrées de 28 pixels 70000 exemples le premier 60000 forment l'ensemble d'entraînement et les autres 10000 sont utilisés comme ensemble de test.

### **3.2.2 Matériel et software utilise**

Dans ce travail nous avons utilisé un PC HP son nom est DESKTOP-4TE571J et le processor Intel(R) Coré (TM) i5-4200M CPU @ 2.50GHz 2.50 GHz et avec une RAM 4.00 GB (3.88 GB usable) , et pour la programmation nous avons utilisé MATLAB versions R2018b et ...., python version 3.7 sous Anaconda Navigator Snyder.

### **3.2.3 Algorithme conventionnels d'alignement d'images :**

Dans cette section, nous présentons les résultats d'alignements par la méthode statistique : méthodes basées sur la Corrélacion Croisée notée (CC), et celle d'alignement basées transformation géométrique et l'information mutuelle, des codes Matlab ont été écrit aux fins d'implémenter les algorithmes étudiés. et nous présentons les résultats d'alignements par la méthode avec CNN.

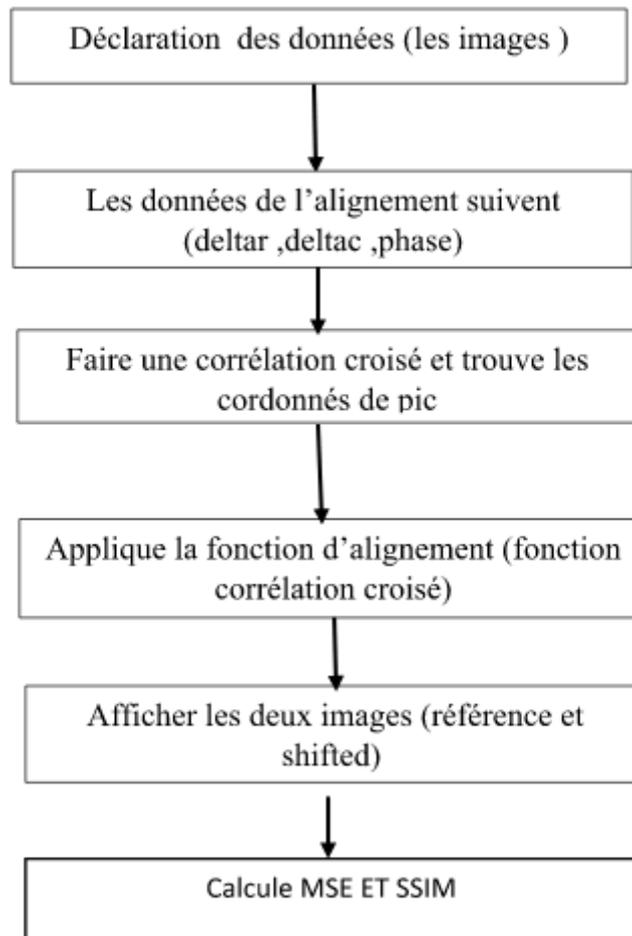
- **Algorithme d'alignement par Corrélacion Croisée**

La Figure un organigramme de programme sur Matlab de méthode cross corrélation (CC). Les réponses visuelles de l'alignement sont illustrées sur les Figures ces figures représentent l'image désalignée et l'image alignée respectivement de l'image originale **3.2** et **3.4** et **3.6** et **3.8**.

Nous avons appliqué la méthode CC, développée sous Matlab, pour aligner l'image qui est décalée de l'image originale, sur l'axe x et l'axe y par les valeurs 15.00 et 20.00 respectivement, nous

avons pris d'autres.

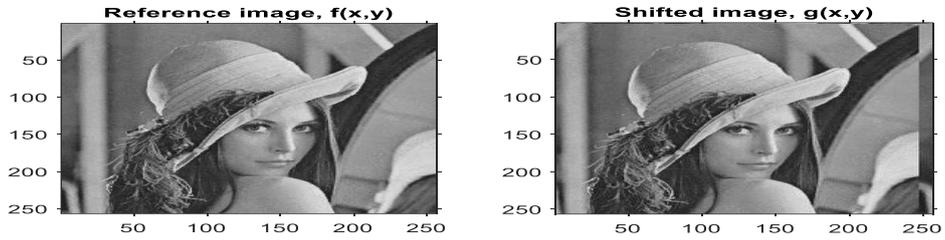
Exemples de Toolbox Matlab illustrés dans les Figures.



**Figure 3.1 :** Organigramme de méthode CC sur Matlab



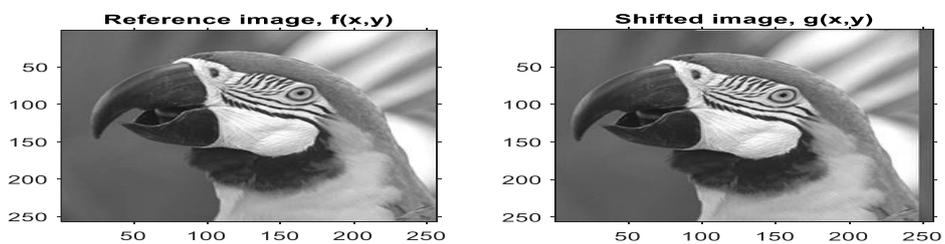
**Figure 3.2 :** Image lena256 format tif de taille 255 × 255 pixels



**Figure 3.3 :** Image lena256 désalignée (à gauche) et image lena256 alignée par CC (a droit)



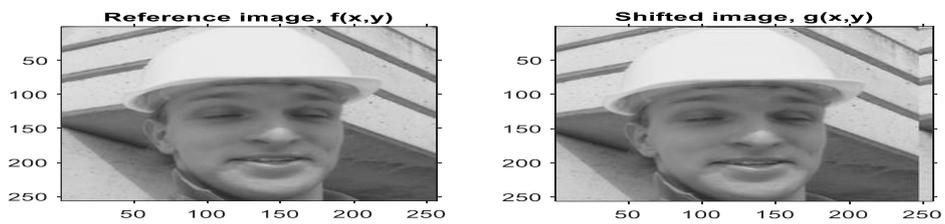
**Figure 3.4 :** Image Parrot format tif de taille  $255 \times 255$  pixels



**Figure 3.5 :** Image Parrot désalignée (à gauche) et image Parrot alignée par CC (à droit)



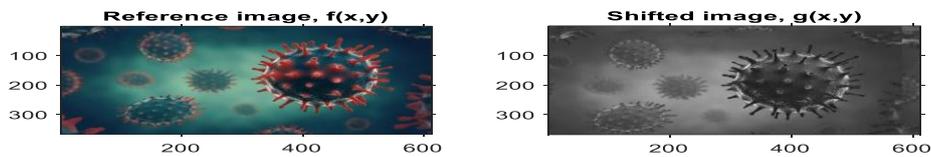
**Figure 3.6** : Image Foreman format tif de taille  $255 \times 256$  pixels



**Figure 3.7** : Image Foreman désalignée (à gauche) et image Foreman alignée par CC (à droite)



**Figure 3.8** : Images covid\_19 test originales tirées de google



**Figure 3.9 :** Image covid\_19 désalignée (à gauche) et image covid\_19 alignée par CC (a droit)

➤ **Avantages et Inconvénients (Corrélation Croisée)**

**a) Les avantages**

Automatique. Relativement rapide.

Pas toujours performant dû au changement d'angle.

Possibilité de jouer avec des filtres pour améliorer l'alignement.

**b) Les inconvénients**

Autosimilarité de l'image.

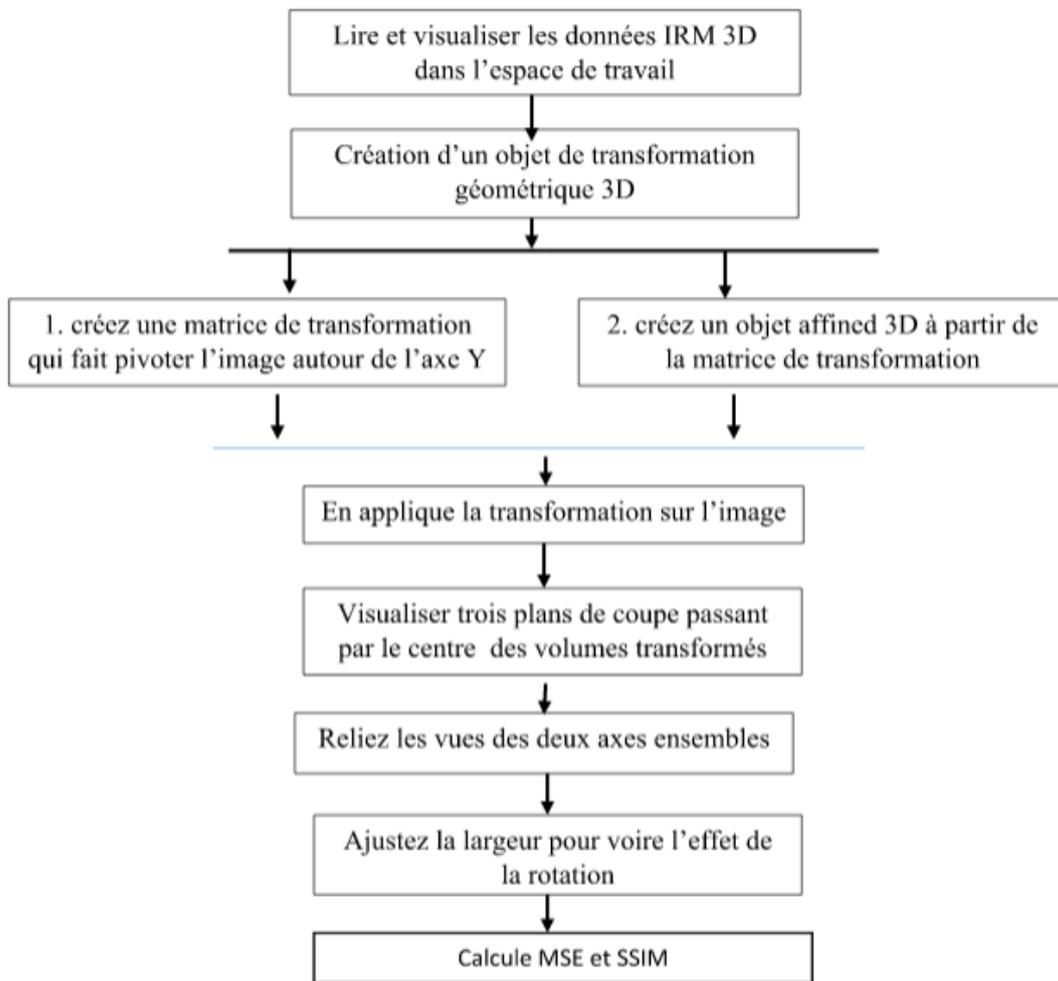
Temps de calcul assez long.

Problème de propagation des erreurs.

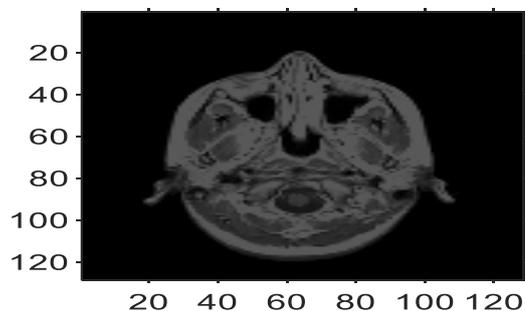
• **Algorithme d'alignement par transformation géométrique**

Illustre l'organigramme de programme sur Matlab de méthode transformation géométrique réponses visuelles de l'alignement sont illustrées sur les Figures ces figures représentent l'image désalignée et l'image alignée respectivement de l'image originale.

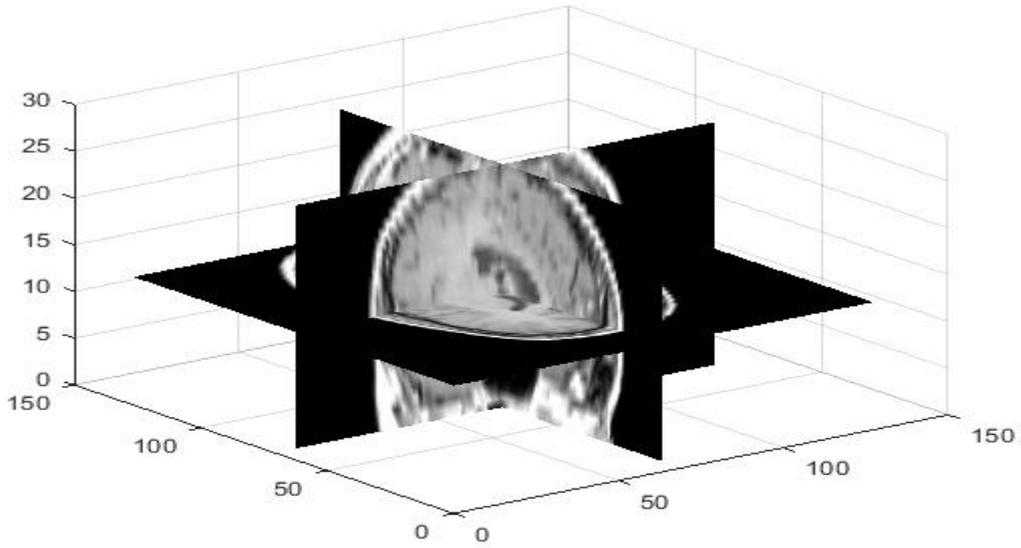
Nous avons appliqué la méthode transformation géométrique, développée sous Matlab, pour créer un objet de transformation géométrique en 3D, faire pivoter une image et visualiser l'alignement des pistes.



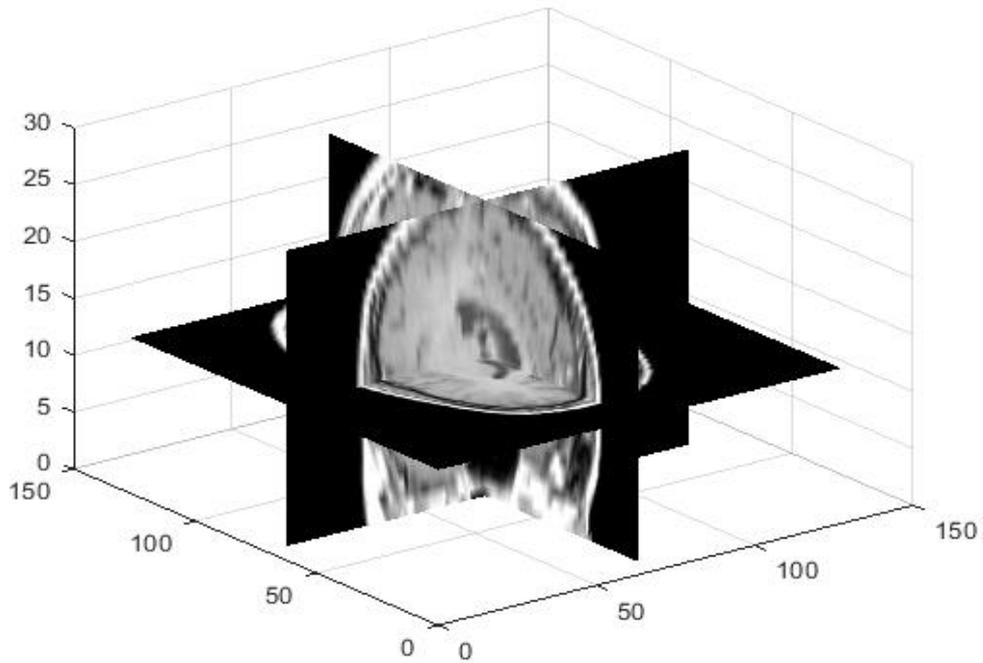
**Figure 3.10 :** Organigramme de méthode transformation géométrique.



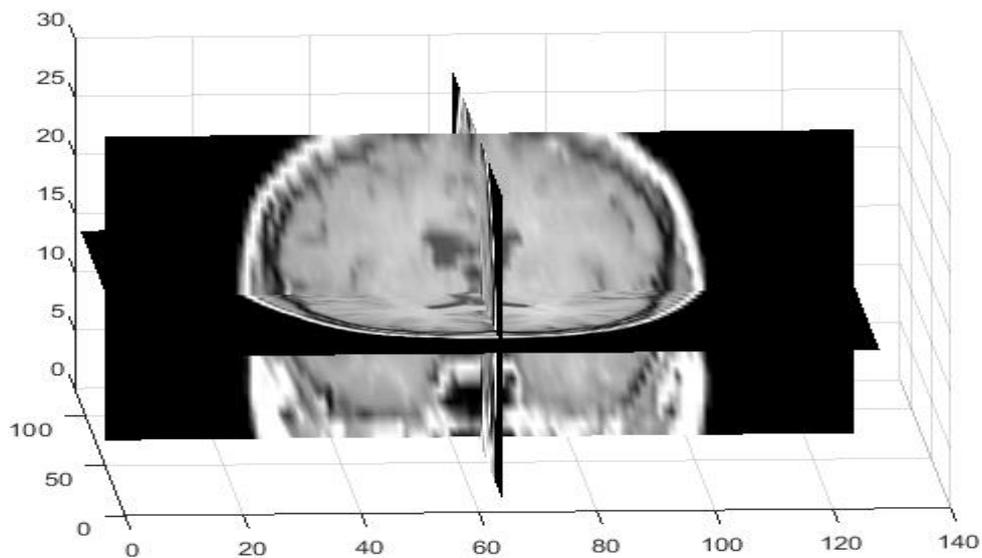
**Figure 3.11 :** L'image origine avant la transformation géométrique (avant l'alignement)



**Figure 3.12 :** Les image 3D dans l'espace de travail.



**Figure 3.13 :** Trois plans de coupe passant par le centre des volumes transformés.



**Figure 3.14** L'image après le processus de rotation

➤ **Avantages et Inconvénients (transformation géométrique) :**

**a) Avantage**

Optimisation des calculs en se concentrant sur des zones de l'image plutôt que de considérer la totalité et le détail de l'image. Utilisation de données déjà issues de l'image et donc plus pertinentes.

Pallier aux problèmes d'artefacts dans les images initiales, qui risquent de poser le problème de différences d'intensité entre les 2 images à recalculer.

**b) Inconvénients**

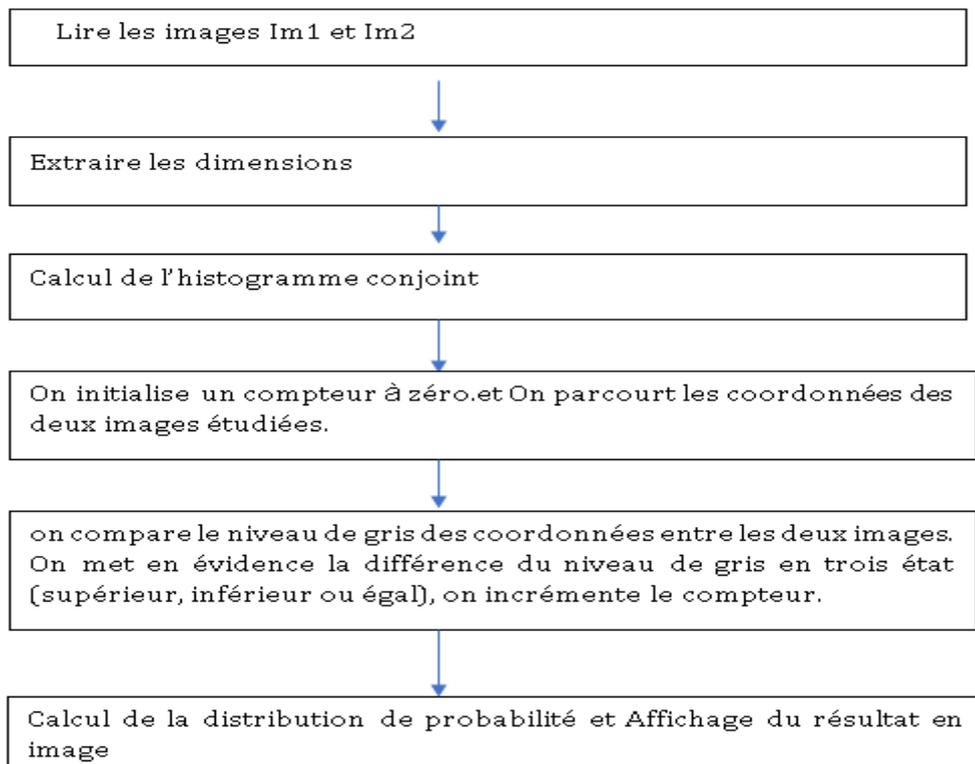
Le choix des primitives est très arbitraire.

Difficulté de calculer les primitives optimales.

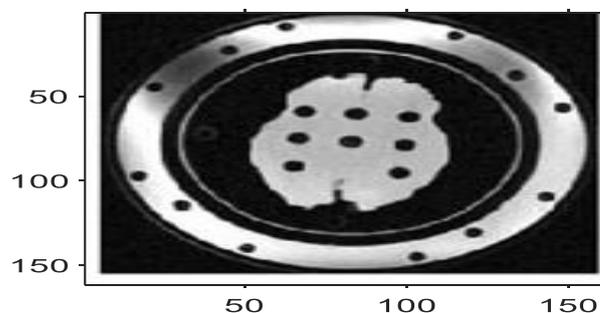
• **Algorithme d'alignement par l'Information Mutuelle :**

La Figure c'est un organigramme de programme sur Matlab de méthode l'Information Mutuelle, et la figure les deux images originales que nous utilisons pour mesure de l'information mutuelle pour une rotation de l'image MRI sur l'image CI auteur de son

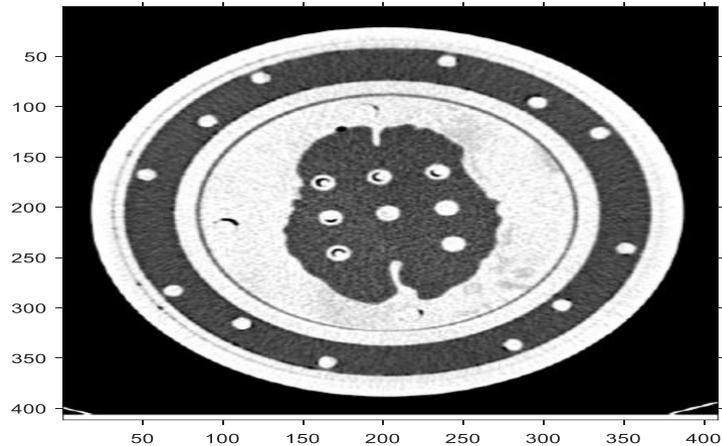
center nous avons choisi d'utiliser une méthode itérative qui pour chaque itération recherche alternativement pour chaque degré de liberté le maximum sur sa propre dimension. Ainsi, à la première itération, on trouve la meilleure translation en x, puis à partir de cette translation, on recherche la meilleure translation en y, puis à l'aide de ces deux translations mises à jour, on recherche la meilleure rotation. Puis on recommence.



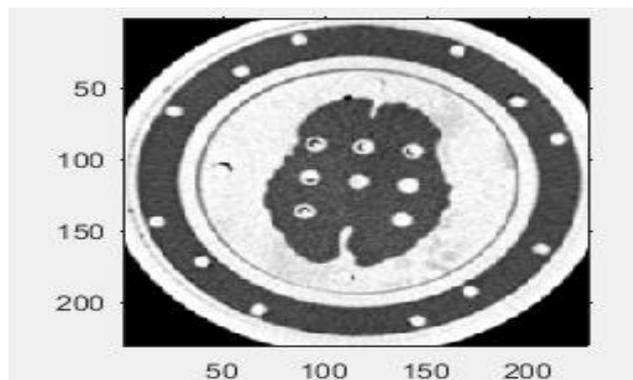
**Figure 3.15 :** Organigramme de méthode l'Information Mutuelle sur Matlab



**Figure 3.16 :** image originale MRI



**Figure 3.17 :** image originale CI

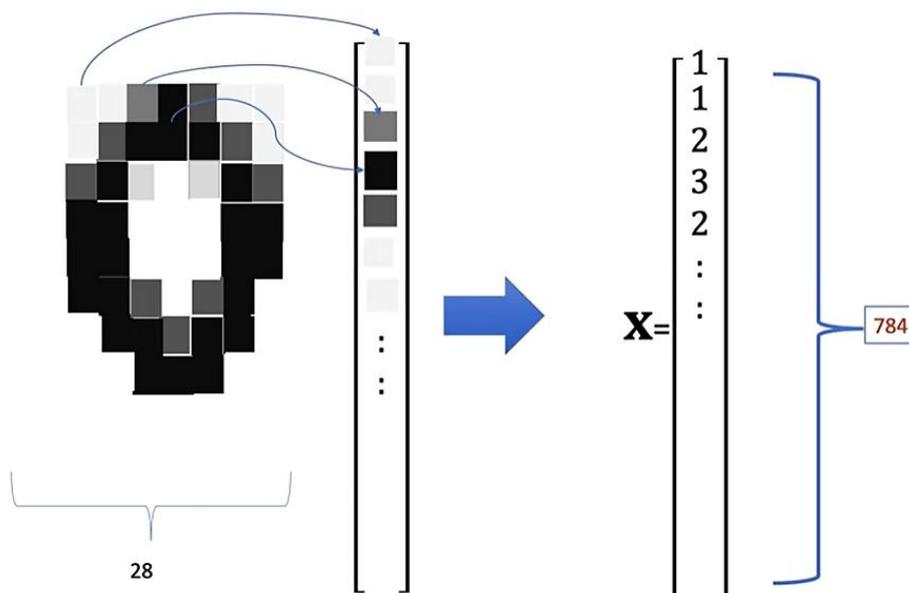


**Figure 3.18 :** image Registered(aligne)

- **Algorithme d'alignement d'images à base de CNN :**

Dans cette partie, nous allons implémenter un réseau CNN :

Les principales applications des auto-encodeurs sont la détection d'anomalies ou le débruitage d'images, mais ils sont également capables de résoudre d'autres problèmes, dans notre projet, nous avons entraîné un auto-encodeur convolutif pour aligner des chiffres manuscrits ayant subi une rotation et les ramener à leur position initiale. Nous avons utilisé l'ensemble de données MNIST. Les images d'entrée sont de dimensions  $28 \times 28$  pixels soit un vecteur d'entrée de dimension 784, Figure 3.19. Les images ont été ajustées de telle façon que la valeur des pixels soit comprise entre 0 et 1. Pour chaque image, il y a 10 étiquettes possibles, correspondant aux chiffres de 0 à 9. Comme nous n'avons besoin que des images de l'ensemble de données pour coder et décoder, nous n'utiliserons pas ces étiquettes. Un extrait est représenté en Figure 3.20

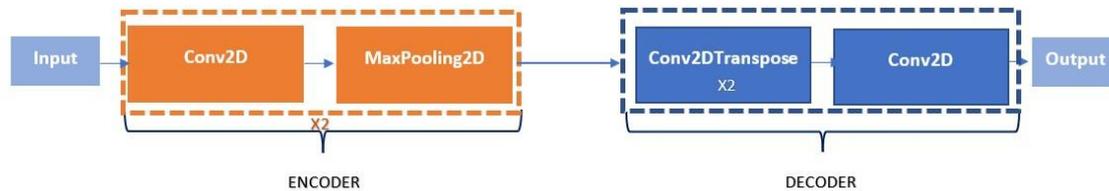


**Figure 3.19 :** Illustration d'image de dimensions 28×28, vecteur d'entrée de dimension 784.

3	8	6	9	6	4	5	3	8	4	5	2	3	8	4	8
1	5	0	5	9	7	4	1	0	3	0	6	2	9	9	4
1	3	6	8	0	7	7	6	8	9	0	3	8	3	7	7
8	4	4	1	2	9	8	1	1	0	6	6	5	0	1	1
7	2	7	3	1	4	0	5	0	6	8	7	6	8	9	9
4	0	6	1	9	2	2	3	9	4	4	5	6	6	1	7
2	8	6	9	7	0	9	1	6	2	8	3	6	4	9	5
8	6	8	7	8	8	6	9	1	7	6	0	9	6	7	0

**Figure 3.20 :** Échantillon de la base de données de reconnaissance de chiffres manuscrits MNIST.

En général, une architecture neuronale comprend une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Dans la stratégie d'apprentissage des architectures profondes proposée ces dernières années, l'initialisation des poids se fait via un auto-encodeur (ou auto-assocateur). Un auto-encodeur comprend deux modules : un encodeur et un décodeur. La figure 3.21, montre l'architecture du modèle d'auto-encodeur que nous avons proposé pour aligner les images. Nous allons détailler cette architecture dans les paragraphes suivants.



**Figure 3.21** : L'architecture du modèle d'auto-encodeur proposé

Afin de construire notre auto-codeur et faire l'entraînement de nos données, nous avons utilisé la bibliothèque Keras, le but de cette bibliothèque est de permettre la constitution rapide de réseaux neuronaux. Dans ce cadre, Keras ne fonctionne pas comme un framework propre mais comme une interface de programmation applicative (API) pour l'accès et la programmation de différents frameworks d'apprentissage automatique. Theano, Microsoft Cognitive Toolkit (anciennement CNTK) et TensorFlow font notamment partie des frameworks pris en charge par Keras. Nous avons fait l'exécution de notre programme sous la plateforme Colaboratory, souvent raccourci en "Colab", qui nous a permis d'écrire et d'exécuter notre code Python. Il offre les avantages suivants :

1. Aucune configuration requise
2. Accès gratuit aux GPU
3. Partage facile

Dans le paragraphe suivant, nous expliquerons les étapes que nous avons suivies afin d'aligner nos données :

#### **A. Préparation des données :**

Comme nous l'avons mentionné précédemment, nous avons utilisé l'ensemble de données MNIST et fait pivoter chaque image selon un angle choisi par l'utilisateur. La tâche des modèles était de les réaligner dans leur état d'origine. Les réseaux neuronaux ne traitent pas les données brutes, comme les fichiers texte, les fichiers image JPEG codés ou les fichiers CSV. Ils traitent des représentations vectorisées et standardisées, par conséquent, notre première étape consiste à lire les données, puis à les normaliser (les valeurs des pixels entre 0 et 1). Dans les expérimentations, l'ensemble d'apprentissage prédéfini de MNIST (70 000 exemples) est séparé en un ensemble d'apprentissage à proprement dit (60 000 exemples, training dataset) et un ensemble de test (10 000 exemples, test dataset). La figure3.22 montre

une extraction des images d'apprentissage et une version de celles-ci avec distorsion (rotation).

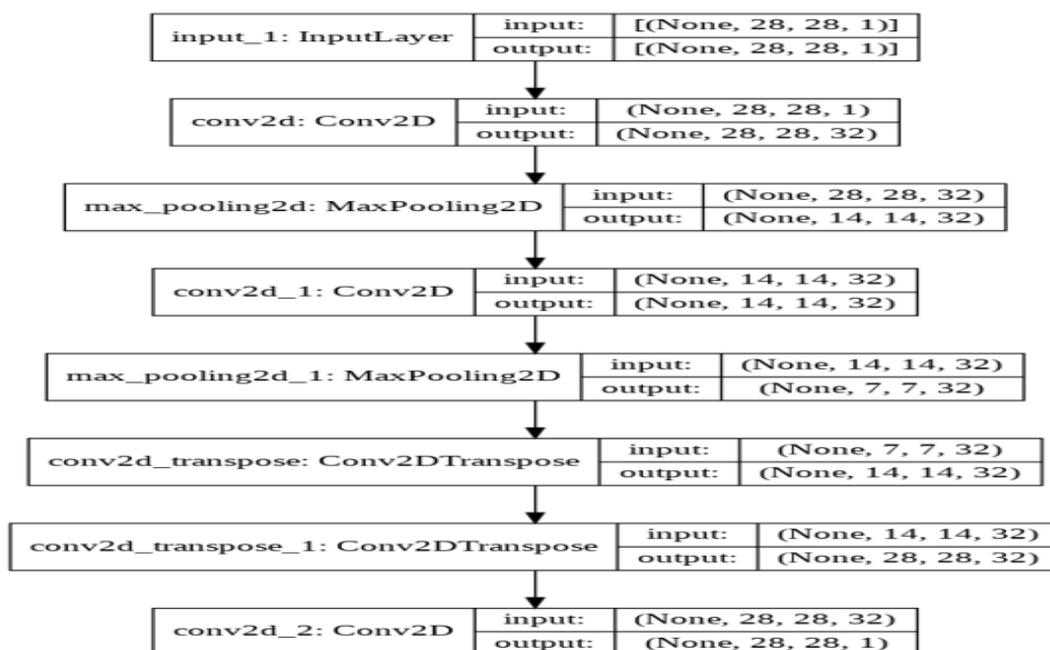


**Figure 3.22 :** les images d'apprentissage (en haut), les images tournées à 330 degrés (en bas).

**B. Création du modèle 'Auto-Encodeur' :**

Pour aligner les images, nous avons proposé un auto-encodeur convolutif qui se compose uniquement de couches convolutives dans l'encodeur et de couches convolutives transposées dans le décodeur.

Dans l'encodeur, les données d'entrée passent par deux couches convolutives avec des noyaux 3x3 et de taille du filtre 32. Deux couches de maxpooling sont ajoutées après la première et la deuxième couche convolutive. La couche de pooling (en anglais pooling layer) (POOL) est une opération de sous-échantillonnage typiquement appliquée après une couche convolutive. Chaque opération de pooling sélectionne la valeur maximale de la surface. Le décodeur reflète cette architecture avec des couches convolutives transposées sans les couches de maxpooling. Nous avons utilisé l'API fonctionnelle pour construire notre auto-encodeur convolutif. La Figure x montre l'architecture complète de l'auto-encodeur convolutif.

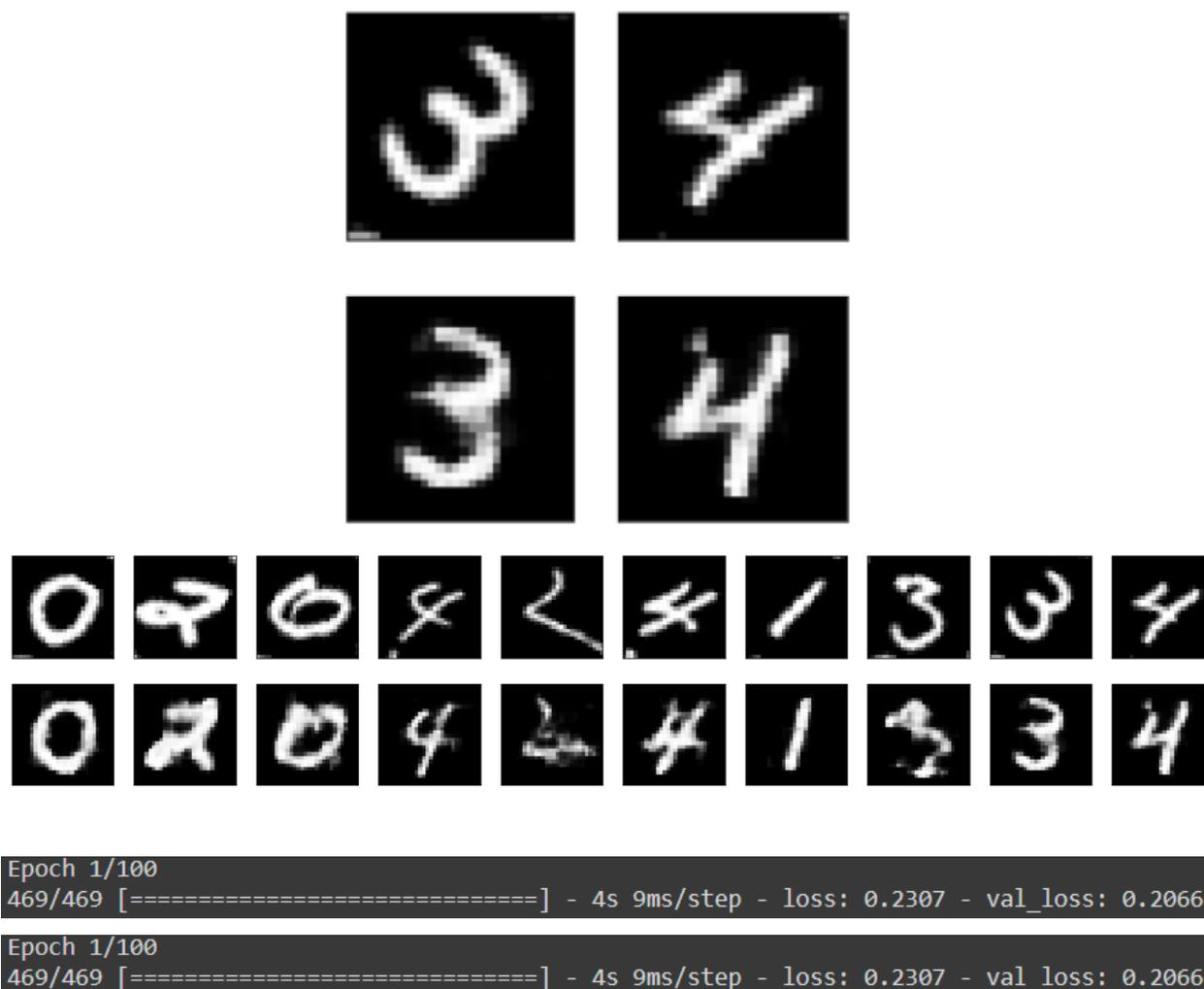


**Figure 3.23 :** auto-encodeur convolutif.

**C. Expériences :**

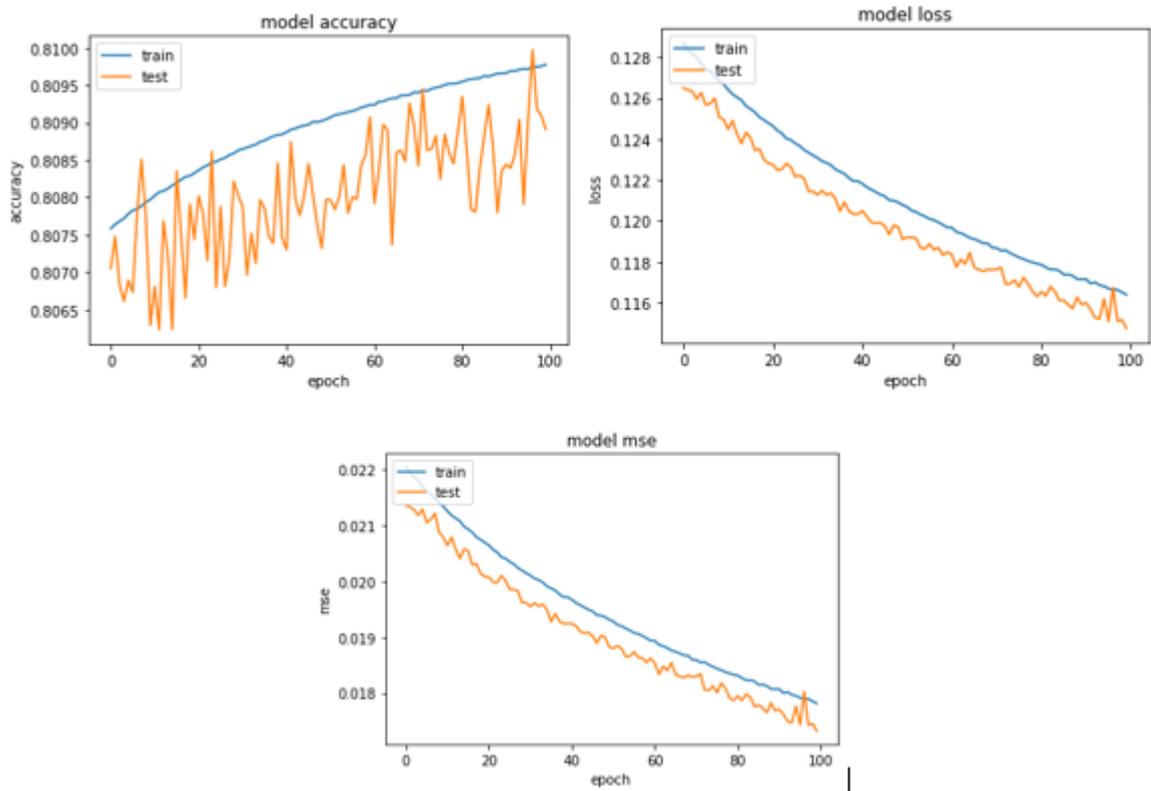
Dans cette partie, nous montrons les résultats que nous avons obtenus en utilisant différentes époques [50, 100, 150, 200, 250] et une taille de batch de 128, optimiseur 'ADAM'. Les expériences sont appliquées sur des images tournées pour démontrer l'efficacité de l'architecture proposée pour l'alignement.

Pour évaluer les résultats, nous avons calculé le MSE, nous avons tracé les courbes de la fonction de précision (Accuracy) et de la fonction de perte (Loss, Binary crossentropy), et nous avons, bien sûr, vérifié la qualité visuelle des images obtenues. Tous les résultats que nous avons obtenus, en utilisant le GPU de Colab.

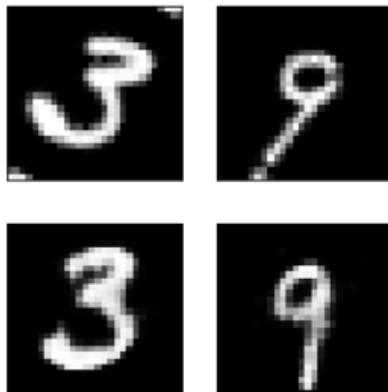
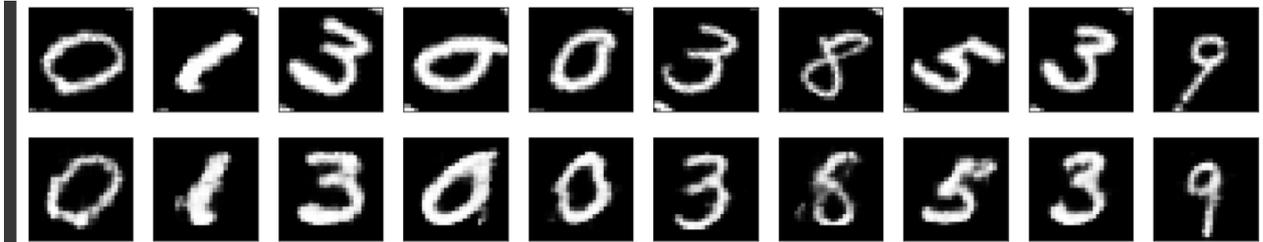


**Figure 3.24 :** résultat de jeu nous avons utilisé 100 Epoch

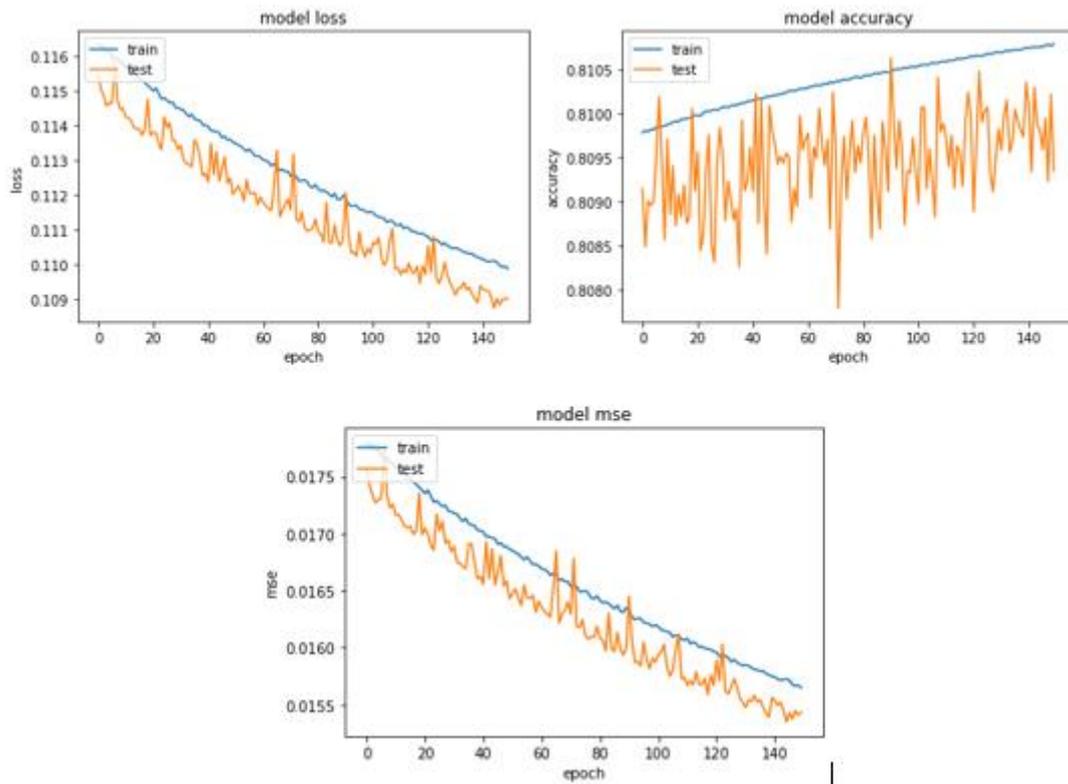
### CHAPITRE3 : IMPLEMENTATION DES ALGORITHMES D'ALIGNEMENT D'IMAGES



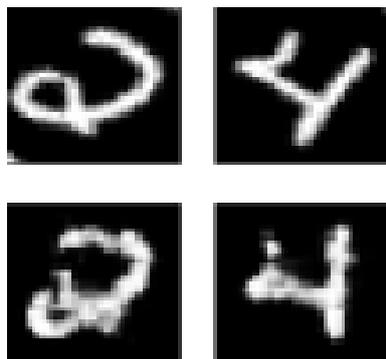
**Figure 3.25 :** courbe graphique représentant les changements de (accuracy ,loss , mse) lors de l'utilisation epoch 100

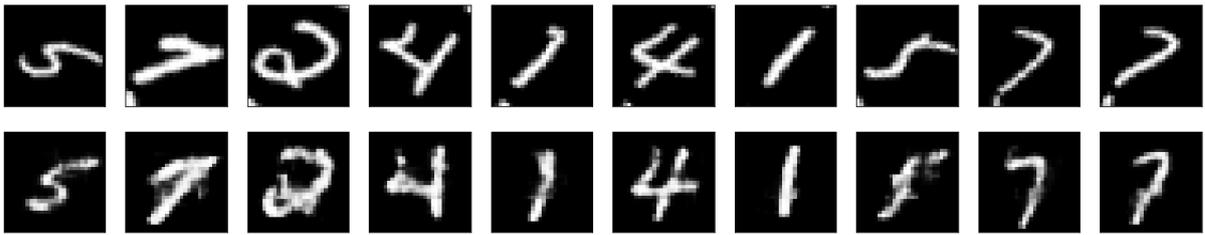


**Figure 3.26** : résultat de jeu nous avons utilisé 150 Epoch

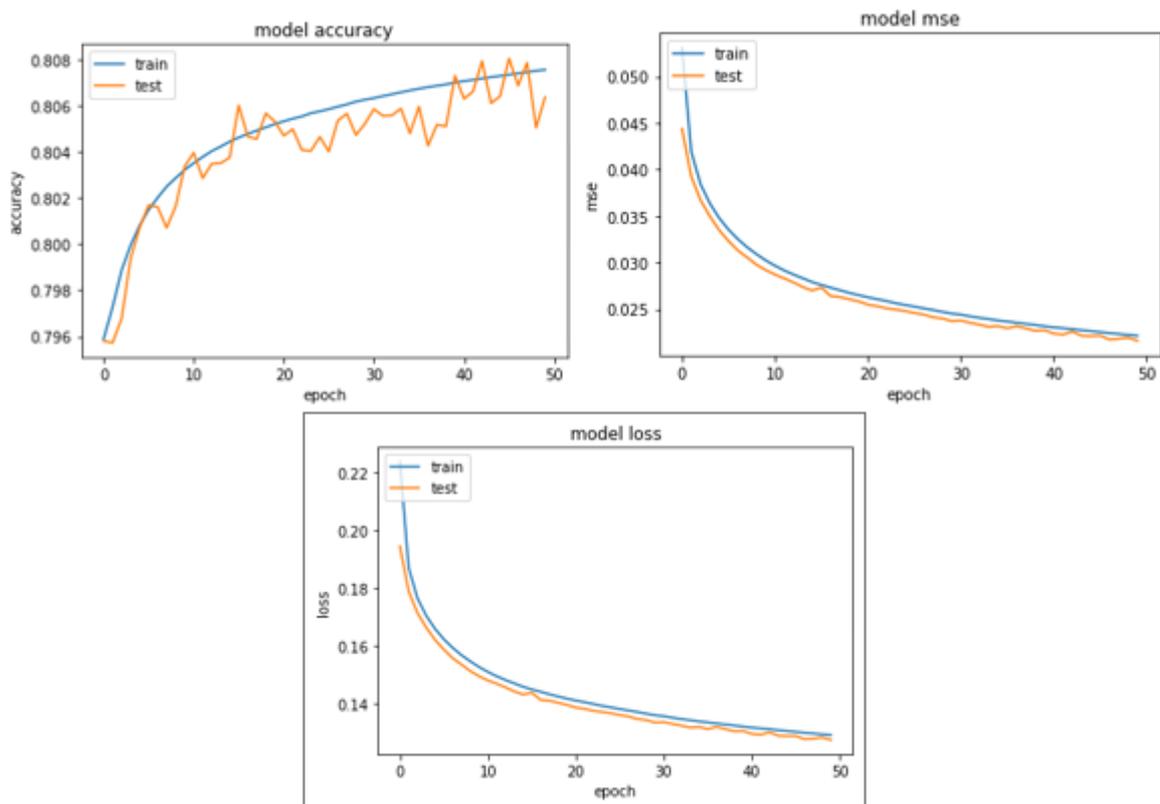


**Figure 3.27** : courbe graphique représentant les changements de (accuracy ,loss , mse) lors de l'utilisation epoch 150





**Figure 3.28** : résultat de jeu nous avons utilisé 50 Epoch



**Figure 3.29** : courbe graphique représentant les changements de (accuracy ,loss , mse) lors de l'utilisation epoch 50

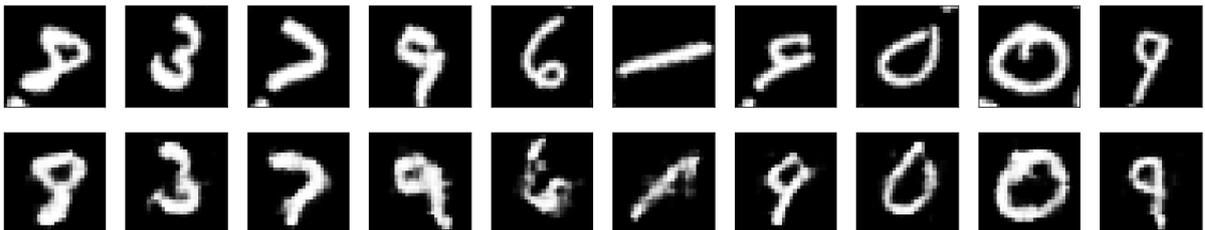
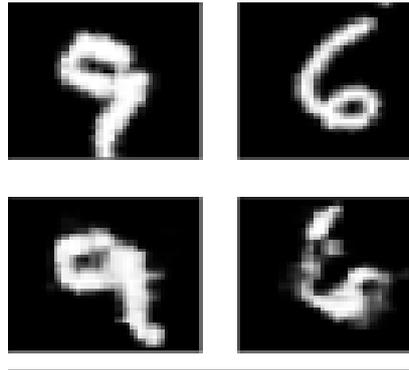
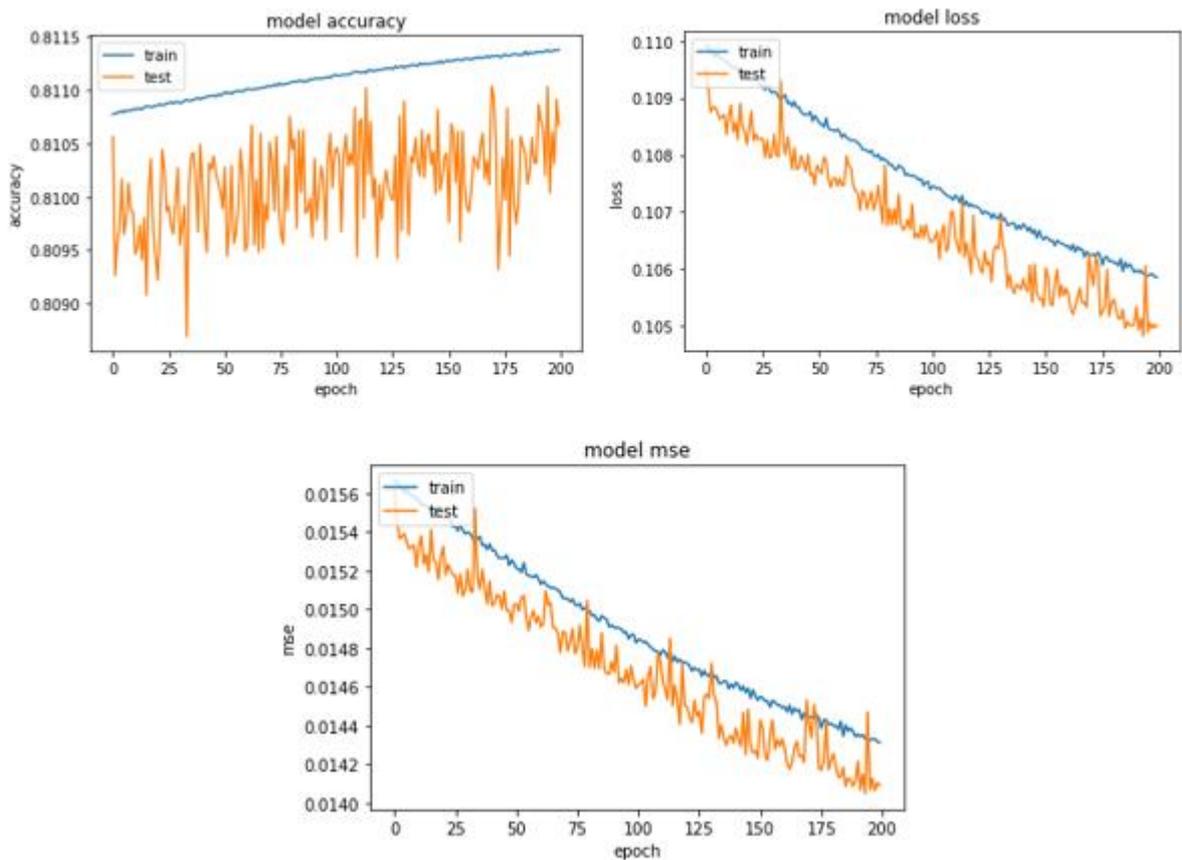


Figure 3.30 : résultat de jeu nous avons utilisé 200 Epoch



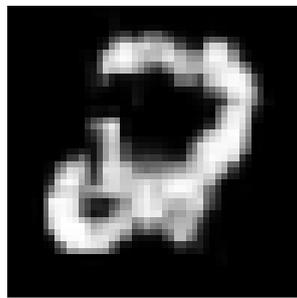
**Figure 3.31 :** courbe graphique représentant les changements de (accuracy ,loss , mse) lors de l'utilisation epoch 200

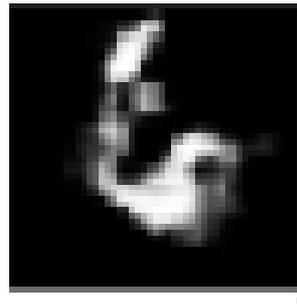
Comme vous pouvez le constater, lorsque nous augmentons le nombre d'époques, nous obtenons des images de meilleure qualité, mais nous constatons une grande fluctuation dans la courbe de précision, vous pouvez voir la différence après avoir appliqué 100 époques et 150 époques et 50 époques et 200 époques dans la Figure 3.32 ,3.33.

Nous réussissons à minimiser le MSE et la fonction de perte. Comme perspective, nous pouvons ajouter plus de couches et essayer d'autres fonctions d'optimisation pour optimiser les résultats.



**Figure 3.32 :** la différence après avoir appliqué 100 époques et 150 époques.





**Figure 3.33** : la différence après avoir appliqué 50 époques et 200 époques.

**la comparaison entre les réseaux ANN et CNN**

Le Tableau 3.1 résume la différence entre l'alignement par approches conventionnelles et alignement à base de CNN.

**Tableau 3.1** la comparaison entre ANN et CNN

Methodes cassique				Methode avec CNN			
		MSE	SSIM			MSE	SSIM
CC	image1	0.2029	72 %	CCN	Epoch50	0.0221	
	image2	0.1901	87.15%		Epoch100	0.0178	
	image3	0.1718	89.86%		Epoch150	0.0156	
	image4	0.0615	96.37%		Epoch200	0.0143	
MI	image1	0.2002	92.15%				
TG	image1	0.1545	82.21%				

**3.3 conclusion**

Dans ce chapitre, nous avons présenté les résultats de notre implémentation. Tous d'abord, nous avons implémenté les algorithmes d'alignement à base d'intercorrélacion, l'information mutuelle et la transformation géométrique. Les algorithmes sont appliqués sur des images

### **CHAPITRE3 : IMPLEMENTATION DES ALGORITHMES D'ALIGNEMENT D'IMAGES**

niveaux de gris où nous avons généré le désalignement. Puis nous avons implémenté l'algorithme d'alignement par CNN. Dans cette partie, nous avons utilisé la base MNIST.

A travers l'étude expérimentale, nous avons constaté que le recalage avec la méthode CNN est plus performant en matière de précision et de temps, ceci se voit à travers les valeurs de simulation obtenues. La méthode classique est moins performante en matière de temps, mais les résultats de recalage sont satisfaisants.

### Conclusion général

Dans ce projet, nous avons discuté des bases des réseaux de neurones en particulier et des réseaux de neurones convolutifs en particulier.

Nous essaierons de créer une synthèse complète sur Le travail réalisé dans cette thèse, l'objectif principal était d'établir, dès le départ Pour finir, une étude comparative des différentes méthodes d'alignement d'images réelles Le but est de mettre en évidence les avantages et les inconvénients de chaque manière et pouvoir suggérer des possibilités de progression.

Dans ce modeste ouvrage, on rappelle d'abord le formalisme général de problème d'alignement, puis nous avons examiné les méthodes d'alignement a base CNN, classique Nous nous sommes principalement appuyés sur la mise en œuvre de méthodes dans le cadre du programme développé, Matlab et Python, qui vise à traiter les séries d'images. , jusqu'à la vraie série d'images.

L'un des principaux problèmes du traitement d'images est la capacité d'analyser et de Traiter plusieurs images dans un référentiel partagé pour le regroupement ou la comparaison Les informations impliquées, ce problème connu sous le nom d'alignement ou (enregistrement), Lorsqu'il s'agit de comparer plusieurs sources d'informations différentes obtenues au fil du temps, avec différents capteurs ou de différents points de vue.

L'alignement d'images, en anglais "registration", désigne la procédure visant à trouver une transformation géométrique approuvée de deux ou plusieurs images.

Le but est de situer les images dans un même repère géométrique, c'est un champ Recherche avec de multiples applications qui comprend un grand nombre de technologies.

Pour résoudre ce problème, il faut utiliser des réseaux de neurones convolutifs plus profonds qui sont répartis sur le GPU plutôt que sur le CPU sur des bases plus grandes.

**Bibliographie**

[1] : <https://arxiv.org/abs/1511.08458>.

[2] : Dahmane, K. (2020). *Analyse d'images par méthode de Deep Learning appliquée au contexte routier en conditions météorologiques dégradées* (Doctoral dissertation, Université Clermont Auvergne)

[3] : <https://images.app.goo.gl/6tumfwoc7NhM8eQL6>

[4] : BENKADDOUR, M. K. (2020). *Reconnaissance des formes et classification automatique : Application à l'identification biométrique* (Doctoral dissertation).

[DS ELEC BENKADDOUR MohammedKamel.pdf](#)

[5]: <https://images.app.goo.gl/dzpaqrhv8nqt2uzG6>

[6]: Widrow, B., & Lehr, M. A. (1990). 30 years of adaptive neural networks : perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9), 1415-1442.

[7]: [https://www.researchgate.net/figure/Lalgorithme-de-la-retro-propagation-de-gradient\\_fig7\\_333200979](https://www.researchgate.net/figure/Lalgorithme-de-la-retro-propagation-de-gradient_fig7_333200979)

[8] : Daha, B. (2018). Un miroir intelligent interactif basé sur Raspberry Pi.

[9] : <http://dspace.univ-bouira.dz:8080/jspui/bitstream/123456789/11300/1/pfe%20mehdi.pdf>

[10] : <https://images.app.goo.gl/dzpaqrhv8nqt2uzG6>

[11]: <http://dspace.univ-tlemcen.dz/bitstream/112/12235/1/Classification-des-images-avec-les-reseaux-de-neurones.pdf>

[12]: LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

[13] : Audebert, N. (2018). *Classification de données massives de télédétection* (Doctoral dissertation, Université Bretagne Sud).

## **BIBLIOGRAPHIE**

[14]: He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[15]: <https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/>

[16] : Abdelmounaim, M. (2016). *Processus D'alignement des Séries D'image Microscopique à Niveau De Gris* (Doctoral dissertation, UNIVERSITE DE MOHAMED BOUDIAF M'SILA FACULTE DE TECHNOLOGIE).

[17] : Chaïma, B. C. (2019). *Mise en oeuvre et évaluation des méthodes d'alignement et tomographie électronique* (Doctoral dissertation, UNIVERSITE MOHAMED BOUDIAF-M'SILA).

[18] : Andriamanampisoa, F. T. (2008). *Recalage multimodal 3D utilisant le modèle élastique, la méthode des éléments finis et l'information mutuelle dans un environnement parallèle* (Doctoral dissertation, Université de Toulouse, Université Toulouse III-Paul Sabatier).

[19] : Sarifuddin, M., Missaoui, R., Vaillancourt, J., Hamouda, Y., & Zaremba, M. (2003). Analyse statistique de similarité dans une collection d'images. *Revue des Nouvelles Technologies de l'Information*, 1(1), 239-250.

[20] : Lamraoui, M., & Boutaouche, F. (2019). Recalage des paires bilatérales et temporelles dans le cadre de l'analyse des mammographies.

[21] : [https://profs.etsmtl.ca/hlombaert/rapport-information\\_mutuelle.pdf](https://profs.etsmtl.ca/hlombaert/rapport-information_mutuelle.pdf)

[22] : Rojbani, H. (2016). *Alignement-reconstruction simultanée de tomogramme électronique et extraction de volume de ribosome* (Doctoral dissertation, Université de Strasbourg).

[23] : Authesserre, J. B. (2010). *Alignement paramétrique d'images: proposition d'un formalisme unifié et prise en compte du bruit pour le suivi d'objets* (Doctoral dissertation, Université Sciences et Technologies-Bordeaux I).

[24] : [https://profs.etsmtl.ca/hlombaert/rapport-information\\_mutuelle.pdf](https://profs.etsmtl.ca/hlombaert/rapport-information_mutuelle.pdf)

[25] : Boucher, A. (2013). *Recalage et analyse d'un couple d'images: application aux mammographies* (Doctoral dissertation, Université René Descartes-Paris V).

[26]: Chen, H. M., & Varshney, P. K. (2003). Mutual information-based CT-MR brain image registration using generalized partial volume joint histogram estimation. *IEEE Transactions on medical imaging*, 22(9), 1111-1119.

## **BIBLIOGRAPHIE**

[27] : Etienne, C. (2019). *Apprentissage profond appliqué à la reconnaissance des émotions dans la voix* (Doctoral dissertation, Université Paris-Saclay).

[28]: Andrade, N., Faria, F. A., & Cappabianco, F. A. M. (2018, October). A practical review on medical image registration: From rigid to deep learning based approaches. In *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (pp. 463-470). IEEE.

[29]: Phil, K. (2017). *Matlab deep learning with machine learning, neural networks and artificial intelligence*. Apress, New York.