

# *Image Progegssing*

## *Brain tumor segmentation*

<https://github.com/PeppeJerry/BRATS-Image-processing/>

Giuseppe Murgolo  
g.murgolo2@studenti.poliba.it  
Polytechnic of Bari  
Bari, Edoardo Orabona, 4

## *Introduzione*

La segmentazione del cervello per la rilevazione del tumore rappresenta un compito cruciale nell'ambito della medicina neurologica. Trovare la massa tumorale, in modo preciso e tempestivo, permette di iniziare un percorso terapeutico fin dalle fasi iniziali, aumentando notevolmente l'efficacia delle cure che ne seguiranno.

La *Medical Segmentation Decathlon (MSD)* è una competizione internazionale che punta alla ricerca di un buon algoritmo *general-purpose* per la segmentazione di anomalie, tra cui anche il tumore al cervello; i dataset per l'analisi del tumore saranno gli stessi della challenge poiché pubblici per ulteriori studi autonomi.

L'obiettivo di questo progetto è quello di esplorare possibili pipeline per migliorare l'efficacia delle tecniche di segmentazione, le quali inizialmente saranno K-means e Otsu thresholding, a seguito di appropriate trasformazioni sui dati.

Per migliorare la qualità delle scansioni MRI, l'immagine verrà preprocessata con opportune tecniche quali *Histogram Equalization* per migliorare il contrasto del tumore, *PCA* per effettuare denoising e rimozione dei dettagli superflui, e infine applicazione di un *filtro gaussiano* per rendere l'immagine più liscia.

Per un ulteriore confronto con le tecniche di segmentazioni disponibili, verrà analizzata la possibilità di utilizzare un *Gaussian Mixture Model (GMM)*, un modello probabilistico come alternativa alle tecniche precedentemente enunciate.

A seguito delle segmentazioni, le partizioni del tumore fornite verranno analizzate con le stesse metriche della challenge *MSD*, ovvero *Dice Similarity Coefficient (Dice)* e *Normalized Surface Dice (NSD)*.

Le conclusioni includono delle note di riferimento a una possibile integrazione di algoritmi di image processing nei moderni modelli di deep learning.

# Sommario

<b>Introduzione.....</b>	<b>1</b>
<b>Sommario .....</b>	<b>2</b>
<b>Capitolo 1   Introduzione al problema.....</b>	<b>4</b>
1.1 Challenge – Medical Segmentation Decathlon.....	4
1.1.1 Commenti sull’algoritmo vincitore .....	4
1.2 Risonanza magnetica (MRI) del Cervello .....	4
1.2.1 FLAIR (Fluid-Attenuated Inversion Recovery).....	5
1.2.2 T1w (T1-weighted) .....	5
1.2.3 T1w Gd (T1-weighted with Gadopentetato) .....	5
1.2.4 T2w (T2-weighted) .....	5
<b>Capitolo 2   Analisi dei dati.....</b>	<b>6</b>
2.1 Introduzione ai dati .....	6
<b>Preprocessing .....</b>	<b>6</b>
2.2 Min-max normalization .....	6
2.3 Histogram Equalization (HE) .....	6
2.3.1 Implementazione nel progetto.....	7
2.4 Principal Component Analysis – PCA.....	7
2.4.1 Implementazione nel progetto.....	8
2.5 Filtro gaussiano.....	8
2.5.1 Implementazione nel progetto.....	8
Resoconto preprocessing .....	8
<b>Capitolo 3   Trasformazioni.....</b>	<b>9</b>
3.1 Homomorphic Filter .....	9
3.1.1 Implementazione nel progetto.....	9
3.2 Log Transform .....	10
3.2.1 Implementazione nel progetto.....	10
Resoconto trasformazioni .....	10
<b>Capitolo 4   Segmentazione.....</b>	<b>11</b>
4.1 K-means .....	11
4.1.1 Implementazione nel progetto.....	11
4.2 Otsu thresholding.....	12
4.2.1 Implementazione nel progetto.....	12
4.3 Gaussian Mixture Model .....	12
4.3.1 Implementazione nel progetto.....	12

<b>Capitolo 5   Metriche e Risultati .....</b>	<b>13</b>
5.1 Dice Similarity Coefficient (DSC o Dice) .....	13
5.2 Normalized Surface Dice (NSD) .....	13
5.3 Note finali sulle metriche.....	13
<b>Risultati 14</b>	
5.4 Criterio di valutazione .....	14
5.5 Trasformazioni.....	14
5.5.1 Commenti sui risultati ottenuti.....	14
5.6 Segmentazioni.....	15
5.6.1 Commenti sui risultati ottenuti.....	15
5.7 Pipeline (Trasformazione-Segmentazione).....	16
5.7.1 Commenti sui risultati ottenuti.....	16
5.8 Risultati migliori ottenuti sui campioni analizzati .....	17
<b>Capitolo 6   Conclusioni .....</b>	<b>18</b>
6.1 Note conclusive con sviluppi futuri .....	18
6.1.1 Preprocessing .....	18
6.1.2 Tecniche di trasformazione .....	18
6.1.3 Tecniche di segmentazione .....	18
6.1.4 Analisi della Gaussian Mixture Model .....	19
6.2 Appunti di riferimento al deep learning.....	19
6.2.1 Introduzioni di variabili latenti.....	19
6.2.2 Ensemble learning .....	19
6.2.3 Problema di piccoli dataset .....	19
<b>Bibliografia.....</b>	<b>19</b>

# Capitolo 1 | Introduzione al problema

---

## 1.1 Challenge – Medical Segmentation Decathlon

La MSD è una competizione internazionale al fine di valutare algoritmi di segmentazione delle immagini mediche su una vasta gamma di compiti clinici.

### Di cosa tratta

La MSD è una sfida che punta all'analisi delle immagini biomediche sui quali gli algoritmi di segmentazione competono. Il problema era quello di identificare un buon algoritmo “*general-purpose*” in grado di ottenere buone prestazioni anche in mancanza di specifiche, quali parte del corpo da analizzare e natura dei dati da segmentare.

Grazie alla competizione è stato possibile sponsorizzare delle nuove soluzioni versatili in grado di lavorare su una vasta gamma di problemi di segmentazione di varia natura pur mantenendo delle buone prestazioni.

### Come si è svolto

La competizione prevedeva una fase di sviluppo concentrandosi su un allenamento su **sette compiti di segmentazione noti** (Cervello, Cuore, Fegato, Polmoni, Pancreas, Prostata, Ippocampo e Cuore) per poi valutare le prestazioni dei modelli nel gestire una **serie di compiti nascosti** (Milza, Vena epatica e Colon).

### 1.1.1 Commenti sull'algoritmo vincitore

La sfida si è conclusa con la vittoria della nnU-Net la quale è stata in grado di mantenere un elevato grado di generalizzazione sui diversi task, sia noti che nascosti.

Dopo la challenge, l'algoritmo è stato monitorato per ulteriori anni a seguire portando a nuovi risultati in task clinici.

### no new U-Net (nnU-Net)

È un framework automatizzato che permette di segmentare immagini (prevalentemente mediche) e basato sulla U-net, nota per la sua struttura ad U e utilizzata per compiti di segmentazione.

Ciò che distingue i due modelli è la capacità di **adattarsi automaticamente** al compito assegnato senza bisogno di configurazioni manuali rendendo la sua implementazione su vari compiti abbastanza semplice, anche per una persona non esperta nel settore.

## 1.2 Risonanza magnetica (MRI) del Cervello

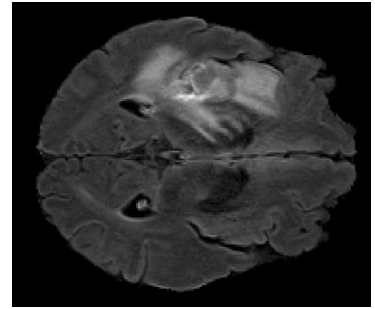
È una struttura dati del tipo  $(x, y, z, t)$  dove le prime coordinate  $(x, y, z)$  indicano il **livello registrato** d'intensità magnetica e di onde radio, paragonabile al valore d'intensità di grigio di una immagine ma registrato in modo completamente diverso, utilizzato poi per ricostruire la struttura interna cerebrale. Con  $t$  viene indicato il **tipo di protocollo** utilizzato per la MRI, ognuno di essi è in grado di fornire **specifiche caratteristiche di contrasto e sensibilità** al fine di evidenziare vari aspetti e patologie.

### 1.2.1 FLAIR (Fluid-Attenuated Inversion Recovery)

Questo protocollo punta a **sopprimere i segnali dei liquidi** (come il liquido cerebrospinale) mediante l'utilizzo di una particolare tecnica d'inversione mettendo in contrasto le anomalie.

È molto utile per individuare **lesioni** (come tumori) nelle zone più chiare della scansione.

Nelle immagini FLAIR, le zone con più liquidi (quali CSF) appaiono **più scure**, mentre le lesioni comportando alterazioni appaiono **più chiare** e luminose (forte contrasto).

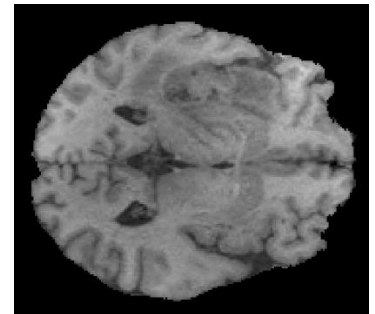


### 1.2.2 T1w (T1-weighted)

Con T1 pesate viene indicata una sequenza di base della MRI dove vengono messe in evidenza **rilassazioni longitudinali dei protoni**, che cambia a seconda delle caratteristiche dei tessuti.

Il grasso e strutture ad alta densità appaiono **più luminosi** e diventa più chiaro visualizzare strutture anatomiche ottenendo una visione della morfologia del cervello.

Questo aumento di dettagli viene utilizzato per **identificare anomalie strutturali** (e.s. Tumori).

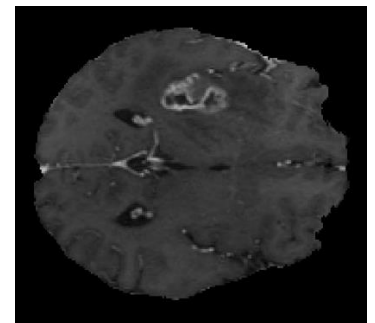


### 1.2.3 T1w Gd (T1-weighted with Gadopentetato)

Variante della T1w con l'aggiunta di un **mezzo di contrasto** (il Gadopentetato) in grado di aumentare il contrasto paramagnetico, alterando il rilassamento dei protoni, e rendendo le aree di accumulo più luminose rispetto alle immagini T1.

Il gadolinio migliora il contrasto e di conseguenza rende visibili strutture come vasi sanguigni, infezioni e tumori.

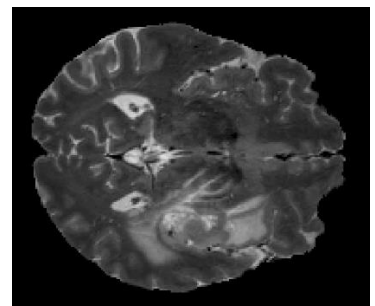
Le aree patologiche che assorbono contrasto appaiono **più luminose** differenziandolo dal tessuto circostante.



### 1.2.4 T2w (T2-weighted)

Protocollo che mette in risalto le differenze di **rilassamento trasversale**, ovvero che permette di evidenziare tessuti ricchi d'acqua, rendendo l'immagine più sensibile alle lesioni (Tumori) ed elevata quantità di liquido.

I liquidi (cerebrospinali inclusi) appaiono più luminosi, la sostanza grigia più luminosa rispetto a quella bianca risultando un **contrasto inverso alla T1w**.



# Capitolo 2 | Analisi dei dati

---

## 2.1 Introduzione ai dati

### NIFTI (Neuroimaging Informatics Technology Initiative)

File identificati dall'estensione “.nii”, ampliamento utilizzato per la memorizzazione e lo scambio di dati medici, in particolare neurologici.

In questo caso, il formato fornisce una matrice 4D avente la MRI dello stesso cervello nei diversi protocolli precedentemente analizzati (*FLAIR*, *T1w*, *T1w gd* e *T2w*).

### FLAIR

Ai fini dello studio, verranno utilizzate le immagini **FLAIR** poiché risultano avere un miglior contrasto del tumore che potrebbe migliorare la precisione degli algoritmo di segmentazione risparmiando un calcolo computazionale eccessivo.

### imagesTr, imagesTs e labelsTr

Per lo studio, sono state prese le immagini appartenenti alla cartella “**imagesTr**” contenente l'immagine del cervello completo con tumore ( $x$ ) con relativo label “**labelsTr**” avente l'immagine del tumore ( $Y$ ).

Le immagini presenti nella cartella “**imagesTs**” sono immagini di MRI senza label che saranno ignorate per lo studio ma possono essere utilizzate come immagini di test per sviluppi futuri.

## Preprocessing

## 2.2 Min-max normalization

La min-max normalization è una tecnica largamente utilizzata con l'obiettivo di ridimensionare la scala dei dati in un range  $[a, b]$ , tipicamente viene scelto il range  $[0, 1]$ .

Grazie a questa normalizzazione è possibile attenuare le **differenze di scale** tra i pixel, riducendone la variabilità e riuscendo a **mantenere la distribuzione originale dei dati** (Migliore rappresentazione visiva).

$$z = \frac{x - \min}{\max - \min}$$

Con  $x \in [\min, \max]$  campione originale e  $z \in [0, 1]$  il campione normalizzato.

La visualizzazione dei dati **non viene alterata** da questa normalizzazione dato che vengono espresse le stesse informazioni in una scala differente.

## 2.3 Histogram Equalization (HE)

La HE è una procedura automatica per permettere di migliorare il contrasto di un'immagine senza richiedere l'inserimento di parametri da tarare da parte dell'utente.

L'idea alla base è quella di **ridistribuire le intensità dell'immagine** partendo da un istogramma con una distribuzione e ottenendo un nuovo istogramma che segue una particolare distribuzione.

### Trasformazione T

Nella HE viene definita una funzione  $T(x) = z$ , monotona crescente nell'intervallo d'interesse  $[a, b]$  con  $x$  pixel dell'immagine originale e  $z$  il valore dopo l'equalizzazione.

È importante che anche  $z \in [a, b]$  affinché la trasformata sia corretta.

## Probability density function PDF

È la probabilità che un pixel casuale abbia il valore d'intensità  $I(k)$ :

$$p_k = \frac{N^{\circ} \text{ pixel con } I(k)}{N^{\circ} \text{ Tot pixel}}$$

## Cumulative Distribution Function (CDF)

I valori d'intensità di un'immagine sono visti come variabili casuali che operano nel range  $[a, b]$  al quale è associata una CDF che ne **descrive le probabilità** di assegnare un dato valore in un dato momento.

Data la definizione della CDF:

$$CDF(x \leq t) = \sum_{k=0}^t P_k$$

È possibile notare come tale definizione sia coerente con le caratteristiche fondamentali della trasformata  $T$ .

### 2.3.1 Implementazione nel progetto

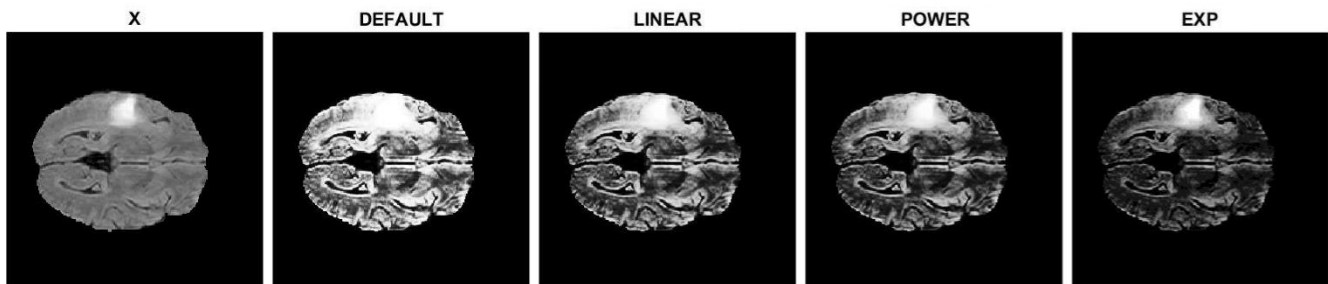
È possibile modificare la definizione della  $PDF$  di riferimento dell'istogramma in modo da poter ulteriormente migliorare il contrasto del tumore.

Per tale scopo, si è deciso di utilizzare una **funzione esponenziale**:

$$pdf(k) = \exp(\lambda * k) \rightarrow k = 0, 1, \dots, (N - 1)$$

Con  $\lambda$  parametro che determina la **rapidità del cambiamento** e  $N$  è il numero **totale di luminosità**.

**Nota:** ai fini dello progetto, si è deciso di utilizzare solo questa  $PDF$  ma nel codice sono state implementate diverse alternative.



## 2.4 Principal Component Analysis – PCA

La PCA è una tecnica utilizzata principalmente per la **riduzione della dimensionalità dei dati** permettendo di analizzare le componenti principali dell'informazione  $x$ , ovvero parti costituenti di  $x$  non correlate tra loro.

### Procedimento (Note)

- 1- Viene effettuata una normalizzazione dei dati (In genere Z-score)
- 2- Viene determinata la matrice di covarianza e i suoi autovalori e autovettori
$$C = x^T x \quad \det(C - \lambda I) = 0$$
- 3- Vengono selezionate un numero  $k < p$  componenti con  $p$  numero massimo di componenti.
$$W = [w_1, w_2, \dots, w_p]$$
- 4- Ricostruzione dell'input  $x$  con un numero di componenti inferiori (Si assume  $W$  ordinato in ordine decrescente rispetto al valore degli autovalori  $\lambda$ )

$$z = xW_k$$

Con  $z$  la ricostruzione di  $x$  avendo salvato le  $k$  componenti più importanti.

### 2.4.1 Implementazione nel progetto

Nel caso specifico del progetto, si è deciso di utilizzare la PCA come tecnica di denoising e riduzione dei dettagli superflui, salvando un numero inferiore di componenti principali che massimizzano la quantità di informazioni salvate relative al tumore.

Per rendere il processo più efficiente, non viene scelto direttamente un numero di componenti  $k$  da mantenere ma una percentuale di **cumulative explained variance** da mantenere.

#### Procedimento

- 1- Normalizzazione dei valori di  $W$  dividendo i valori per la somma di tutti gli elementi:

$$V = \frac{1}{\sum_{k=1}^p w_i} W$$

- 2- Definizione della somma cumulativa delle componenti principali normalizzate:

$$CEV(k) = \sum_{k=1}^k v_i$$

- 3- Verifica della condizione  $CEV(k) \leq \tau$  con  $\tau$  una soglia espressa in percentuale.

**Nota:** vengono mantenute  $k$  componenti che producono il valore più grande di  $CEV(k)$  e rimangono al di sotto del valore di soglia  $\tau$ .

## 2.5 Filtro gaussiano

Lo scopo del filtro gaussiano è quello di levigare ulteriormente le immagini mediante l'utilizzo di un filtro a bassa frequenza, dove è possibile definirne la **dimensione e deviazione standard** riducendo inoltre il rumore ed i dettagli nell'immagine.

### 2.5.1 Implementazione nel progetto

Nel contesto delle immagini 3D, il filtro gaussiano è tridimensionale:

$$G(x, y, z) = \frac{1}{\sqrt{(2\pi)^3} \sigma^3} e^{-\frac{x^2+y^2+z^2}{2\sigma^2}}$$

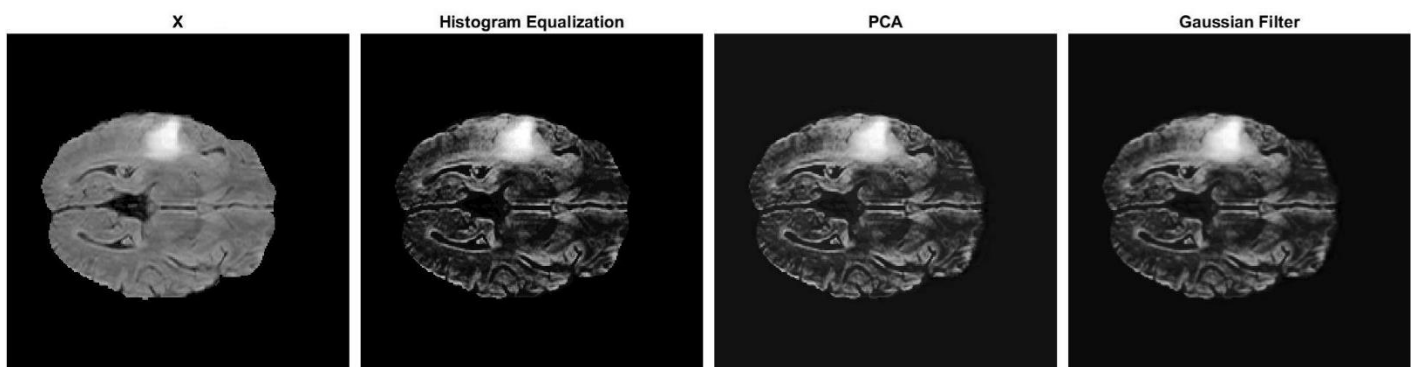
Con  $\sigma$  viene determinata la **forza della sfocatura** e la **dimensione della finestra**, questo perché nel caso di matlab, mediante l'utilizzo del comando `imgaussfilt3(Img,  $\sigma$ )`, viene determinata la dimensione della finestra come multiplo di  $\sigma$  (in genere 3 o 6 volte):

$$W_{size} = \lceil 6\sigma \rceil + 1$$

**Perché è possibile determinarlo automaticamente:** il tutto segue dal decadimento della funzione gaussiana all'aumentare della distanza dal pixel centrale rendendo il suo contributo quasi nullo dopo essere sufficientemente distanti.

### Resoconto preprocessing

L'immagine viene normalizzata con la **minmax normalization** ( $X$ ), contrasto migliorato con la **Histogram Equalization**, riduzione del rumore e dettagli superflui con la **PCA** ed infine ulteriore rimozione del rumore e levigatura con **Gaussian Filter**.





## Capitolo 3 | Trasformazioni

### 3.1 Homomorphic Filter

È una tecnica che punta a **migliorare l'illuminazione** di una immagine digitale.

Una immagine è caratterizzata da una componente di illuminazione e riflettanza, considerati rispettivamente a bassa e alta frequenza; le componenti si combinano in modo moltiplicativo che rende la loro **divisione molto difficile**:

$$M(x, y) = I(x, y) * R(x, y)$$

L'immagine  $M$  è rappresentata dalla componente di illuminazione  $I$  e riflettanza  $R$ .

Mediante una trasformazione nel **dominio delle frequenze**, è possibile **separare linearmente** tale componenti.

Questa trasformazione include un livellamento dell'illuminazione su tutta l'immagine, migliorando il contrasto e mitigando il **rumore moltiplicativo**.

#### Procedimento

1- Viene calcolato il logaritmo dell'immagine:

$$\ln(M(x, y)) = \ln(I(x, y) * R(x, y)) = \ln(I(x, y)) + \ln(R(x, y))$$

Grazie al logaritmo, le componenti di illuminazione e riflettanza sono stati **linearmente separati**.

Per semplicità,  $\ln(I(x, y)) = I'(x, y)$ ,  $\ln(R(x, y)) = R'(x, y)$  e  $\ln(M(x, y)) = M'(x, y)$

2- Applicazione della trasformata di Fourier  $F$  sull'immagine logaritmica  $M'$ :

$$F\{M'(x, y)\} = F\{I'(x, y)\} + F\{R'(x, y)\}$$

3- Applicazione del filtro in frequenza  $H$  sulla trasformata in frequenza

4- Applicare la trasformata inversa di Fourier e ritornare nel dominio originale.

#### 3.1.1 Implementazione nel progetto

Essendo i dati di struttura 3D, è necessario utilizzare il comando  $fftn(x)$  per applicare una **trasformata di Fourier N-dimensionale**, con relativa funzione inversa  $ifftn(x)$ .

Dato che si vuole mitigare l'effetto dell'illuminazione (globale a bassa frequenza) e amplificare le componenti di riflettanza (dettagli ad alta frequenza quale il tumore), è stato implementato un filtro omomorfo gaussiano per ridurre la disomogeneità dell'illuminazione:

$$H(u, v) = (\gamma_H - \gamma_L) \left( 1 - \exp \left( -\frac{D(u, v)^2}{2D_0^2} \right) \right) + \gamma_L$$

- $\gamma_H$  è un fattore di **amplificazione** delle **alte frequenze**.
- $\gamma_L$  è un fattore di **attenuazione** delle **basse frequenze**.
- $D(u, v)$  è la **distanza nel dominio** dal centro  $D(u, v) = \sqrt{u^2 + v^2}$
- $D_0$  è la **frequenza di taglio** che determina la larghezza di transizione tra alta e bassa frequenza.

Il filtro viene applicato all'immagine logaritmica  $M'$  e si effettua la trasformata inversa di Fourier:

$$S(x, y) = H(u, v) * F\{M'(x, y)\} \quad | \quad s(x, y) = F^{-1}\{S(x, y)\}$$

**Nota:** in realtà, viene effettuata la trasformazione  $\ln(1 + x)$  poiché si vuole evitare il caso in cui  $x = 0$ , in matlab è possibile utilizzare  $\log1p(x)$  per tale scopo.

## 3.2 Log Transform

Questa trasformazione è ampiamente utilizzata per migliorare la visibilità di dettagli in un'immagine, particolarmente in situazioni con grandi variazioni di intensità luminosa.

### Utilizzo della tecnica

Quando l'immagine è caratterizzata da **regioni più scure** con aree più chiare estese, questa trasformazione punta al **sopprimere le aree luminose** e migliorare il contrasto complessivo dell'immagine.

$$f(x) = c * \log(1 + N * x)$$

Con  $x$  l'immagine,  $Y$  il risultato della trasformazione con  $c$  e  $N$  dei parametri da tarare.

### 3.2.1 Implementazione nel progetto

Dato che l'immagine è prevalentemente scura con, in genere, il tumore in risalto più chiaro, l'applicazione di questa formula potrebbe **rendere più omogeneo** il tumore rispetto al contesto globale. Questo approccio viene evitato mediante una **inversione delle intensità** dell'immagine originale:

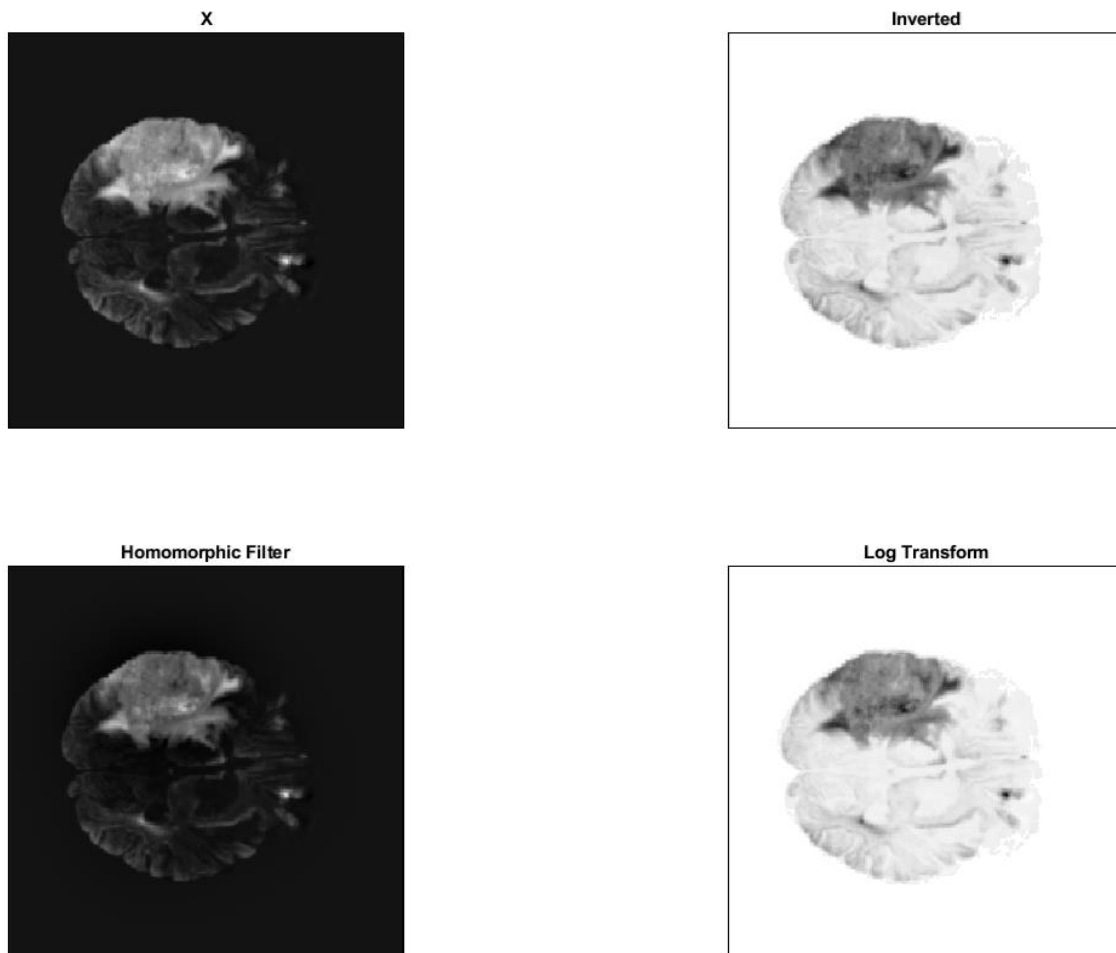
$$Y = -x + \max(x)$$

$S$  rappresenta l'immagine originale  $M$  al quale sono stati invertiti i valori delle intensità; grazie a ciò, il tumore viene rappresentato come una **area scura** in un contesto di **illuminazione chiara**.

Applicando la tecnica in questo caso permetterà di **migliorare il contrasto** del tumore.

### Resoconto trasformazioni

L'immagine  $x$  viene processata ottenendo le trasformazioni risultato dell'applicazione di un **Homomorphic filter** e una **Log transform**. Si precisa che la **Log transform** necessita di invertire le intensità in modo tale rappresentare il tumore come scuro ed il resto come chiaro.



# Capitolo 4 | Segmentazione

## 4.1 K-means

K-means è un algoritmo di **clustering** molto utilizzato nel machine learning non supervisionato. Con questo algoritmo, vengono generati  $k$  gruppi di item omogenei (clusters) partendo da una **scelga casuale dei centri**, detti centroidi.

L'algoritmo è iterativo e ad ogni step vengono determinati dei **nuovi centroidi** fino a che non si raggiunge una convergenza (o non vengono superate un numero massimo di iterazioni).

### Formalmente

Sia  $x^{(i)}$  un punto generico, esso appartiene al cluster  $c^{(i)}$  se la sua distanza  $\mu^{(i)}$  dal centroide risulta essere la più piccola rispetto a quella degli altri cluster con distanze  $\mu^{(1)}, \dots, \mu^{(n)}$ .

- 1- Viene scelto il numero di cluster  $k \ll m$  ( $m$  numero totale di campioni  $x^{(i)}$ )
- 2- Vengono scelti casualmente dei centroidi  $c^{(i)}$
- 3- Vengono determinati i nuovi centroidi tali da ridurre il costo:

$$J(\mu, c) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

In altre parole, i centroidi vengono scelti in modo tale da minimizzare la somma delle distanze quadrate tra i punti  $x^{(i)}$  e il centroide  $\mu_{c^{(i)}}$  del cluster a cui appartengono.

### K-means++

Questa variante del K-means è **direttamente implementata in matlab** dal comando `kmeans(x, k)` e permette di **ottimizzare la scelta dei centroidi** mediante l'utilizzo di una euristica.

L'idea è di scegliere dei centroidi che siano più probabili ad appartenere a cluster differenti:

- 1- Viene scelto il primo centroide casualmente
- 2- Ogni futuro centroide viene scelto con un **approccio probabilistico** in cui la probabilità di essere scelto è proporzionale al quadrato della distanza di ogni punto dal centroide più vicino:

$$p(x) = \frac{D(x)^2}{\sum_{i=1}^m D(i)^2}$$

Con  $D(x)$  è indicata la distanza di  $x$  dal centroide più vicino tra quelli già presenti ma è importante sottolineare che **non vengono scelti con certezza**, introducendo una lieve componente di casualità.

### 4.1.1 Implementazione nel progetto

Nel caso specifico di una immagine, il centroide  $\mu_{c^{(i)}}$  rappresenta un valore d'intensità tale da essere rappresentato centrale a **valori simili d'intensità**.

**Nota:** per una scansione MRI, tali valori rappresentano risposte dei protoni nei tessuti cerebrali in funzione del campo magnetico generato (differente concettualmente da una immagine in scala di grigi).

L'efficacia di questo algoritmo viene migliorata da contrasti netti tra le aree d'interesse (Tumore) rispetto al resto dell'immagine (cervello e background).

## 4.2 Otsu thresholding

Il metodo di Otsu viene utilizzato per migliorare la ricerca di una threshold ottimale al fine di separare diverse classi (tipicamente sfondo e oggetto) minimizzando la varianza intraclasse.

### Funzionamento iterativo

Questo metodo funziona meglio per istogrammi con **distribuzioni multimodali**, vengono separate le classi mediante il calcolo delle thresholds  $T$  e minimizzando la varianza intraclasse.

1- Vengono calcolate media e varianza per le classi

$$\mu_x = \frac{1}{\text{len}(x)} \sum_{p_i \in x} p_i \quad | \quad \sigma_x^2 = \frac{1}{\text{len}(x)} \sum_{p_i \in x} (\mu_x - p_i)^2$$

**Nota:**  $p_i$  rappresenta l'intensità del pixel  $i$  nella classe  $x$ .

2- Vengono calcolati i pesi per classe

$$W_x = \frac{\text{len}(\text{Pixel}_{\text{classe}_x})}{\text{len}(\text{Pixel}_{\text{totali}})}$$

3- Viene calcolata la **within class-variance**

$$\sigma_W^2 = \sum_{i=1}^{\text{len}(T)} W_i \sigma_i^2$$

Questo processo viene ripetuto generando ogni possibile combinazione di threshold  $T = \{t_1, \dots, t_n\}$  salvando il valore che minimizza  $\sigma_W^2$ .

### 4.2.1 Implementazione nel progetto

A seconda del livello di contrasto fornito dal preprocessing, in genere sono presenti circa 2 o 3 classi (salvo eccezioni) le quali rappresentano **background**, **cervello**, **tumore** o solo semplicemente **background** e **tumore** se il contrasto è sufficientemente netto.

Mediante l'utilizzo del comando  $T = \text{multithresh}(\text{image}, N)$  è possibile definire il numero di threshold da generare ( $N$  soglie implica la presenza di  $[N + 1]$  classi).

Mediante l'utilizzo del comando  $\text{map} = \text{imquantize}(x, T)$  è possibile ottenere la mappa segmentata.

## 4.3 Gaussian Mixture Model

Un Gaussian Mixture Model (GMM) è un **modello probabilistico** particolare in cui la distribuzione di probabilità di un item è dato dalla **combinazione di più distribuzioni normali** (dette anche *gaussiane*). Questa tecnica è utilizzata per il clustering in cui è possibile accettare approssimare la distribuzione dei dati ad un prodotto di più gaussiane, caso contrario il suo utilizzo è **sconsigliato**.

$$p(x) = \sum_{k=1}^K \pi_k * N(x|\mu_k, \Sigma_k)$$

La probabilità  $p(x)$  che  $x$  appartenga a una delle  $K$  componenti gaussiane è data dalla sommatoria di  $\pi_k$  (**peso** della  $k$ -esima gaussiana) moltiplicato  $N(x|\mu_k, \sigma_k)$  (**distribuzione  $k$ -esima** con media  $\mu_k$  e matrice di covarianza  $\Sigma_k$ ).

**Expectation-Maximization (Nota):** Algoritmo che stima i valori di  $\pi_k$ ,  $\mu_k$  e  $\Sigma_k$ .

### 4.3.1 Implementazione nel progetto

Come già affermato, è importante che i dati rispettino nei limiti la condizione di possedere una **distribuzione gaussiana**.

L'utilizzo di questo modello per la segmentazione potrebbe **non risultare adatta per tutte le trasformazioni** (sia quelle proposte dal progetto che non) poiché è possibile che la distribuzione venga mutata al punto da uscire dai limiti della premessa principale.

# Capitolo 5 | Metriche e Risultati

## 5.1 Dice Similarity Coefficient (DSC o Dice)

L'indice Dice è una **metrica di similarità** che misura il livello di sovrapposizione di due set:

$$Dice = 2 \frac{|A \cap B|}{|A| + |B|}$$

La metrica è normalizzata fornendo un risultato  $Dice \in [0,1]$  il quale indica il **grado di sovrapposizione** dei due insiemi.

### Nota progetto

Il coefficiente di similarità di Dice misurerà il grado di sovrapposizione tra l'immagine binaria del tumore reale ( $Y$ , *ground-truth*) e l'immagine predetta  $f(x)$ , considerando i pixel classificati come tumore (1) e non tumore (0).

$$Dice = 2 \frac{|Y \cap f(x)|}{|Y| + |f(x)|}$$

## 5.2 Normalized Surface Dice (NSD)

Questa metrica risulta essere più specifica rispetto alla *Dice* poiché al posto di valutare due insiemi nella loro interezza, viene valutato il **grado di sovrapposizione dei bordi** di due immagini.

$$Dice = 2 \frac{|S_A \cap S_B^{(\tau)}| + |S_B \cap S_A^{(\tau)}|}{|S_A| + |S_B|}$$

Con  $S_A$  e  $S_B$  sono indicati tutti i **pixel di contorno** dei due insiemi  $A$  e  $B$ , mentre con  $S_A^{(\tau)}$  e  $S_B^{(\tau)}$  vengono identificati tutti i pixel presenti in un intorno  $\tau$  dai contorni  $S_A$  e  $S_B$ .

### Nota progetto

In questo caso, è necessario generare le maschere  $S_A$  e  $S_B$  dai bordi dalle immagini  $Y$  e  $f(x)$  della classe 1, rispettivamente inerente ai pixel del tumore.

Per rilevare i bordi è necessario utilizzare tecniche, come filtri o altre tecniche adatte allo scopo.

Una volta aver estratto i bordi, è necessario creare una seconda maschera per  $S_A^{(\tau)}$  e  $S_B^{(\tau)}$  in cui vengono presi i pixel presenti in un intorno  $\tau$  dalle maschere presenti in  $S_A$  e  $S_B$ .

## 5.3 Note finali sulle metriche

Le metriche utilizzate sono le stesse presenti nella challenge *MSD* ma in vista di sviluppi futuri sarebbe possibile utilizzarne altre al fine di estrarre diverse conclusioni e potendo fare paragoni con più modelli.

### Metriche alternative possibili

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

- **Precision:** qualità della predizione dei pixel correttamente predetti ( $TP$ ) rispetto al numero dei pixel totali predetti (tiene in considerazione dei pixel erroneamente predetti  $FP$ )
- **Recall:** quantità dei pixel correttamente predetti ( $TP$ ) rispetto al totale (Si tiene conto dei pixel che non sono stati predetti correttamente  $FN$ )
- **Specificity:** equivalente della *Recall* ma valutata sui pixel della parte negativa (background).

# Risultati

## 5.4 Criterio di valutazione

Le metriche sono state analizzate in funzione delle trasformazioni, segmentazioni e le corrispettive coppie trasformazione-segmentazione per analizzare le tecniche e la pipeline più efficiente.

Per lo studio, sono stati analizzati i risultati di 10 campioni scelti casualmente. Per migliorare l'accuratezza delle statistiche sarebbe possibile aumentare il numero di sample analizzati.

**Campioni analizzati:** 371, 328, 308, 307, 277, 262, 205, 139, 92, 44.

## 5.5 Trasformazioni

A seguire vengono mostrati i risultati riscontrati dalle diverse tecniche di **trasformazione**.

### Legenda trasformazioni

X intensità inverse → **Inv**

| Logaritmo → **Log**

| Homomorphic Filter → **HF**

	Dice				NSD			
Tipo	Mean	Max	Min	STD	Mean	Max	Min	STD
<b>Inv</b>	0.4685	0.8557	0	0.2651	0.3279	0.7019	0	0.2201
<b>Log</b>	0.5611	0.8484	0.2045	0.2043	0.4418	0.8219	0.0959	0.2218
<b>HF</b>	0.5440	0.8022	0.0398	0.2014	0.4496	0.7828	0.0256	0.2088

### 5.5.1 Commenti sui risultati ottenuti

Le trasformate proposte mostrano come diversi aspetti dei campioni analizzati vengano messi in risalto. È possibile notare come la trasformata **Log** e **HF** risultino essere **mediamente migliori** della trasformata più semplice **Inv**, sia analizzando la metrica *Dice* che *NSD*.

La trasformata **Log** risulta essere più efficace da un punto di vista **quantitativo**, date le migliori prestazioni nella metrica *Dice*, con un buon valore di massimo registrato ed una bassa deviazione dei risultati; questa trasformazione è stata in grado di fornire ottimi risultati anche nel caso peggiore, ottenendo un risultato di minimo nettamente più alto rispetto alle rimanenti trasformate.

La trasformata **HF** risulta essere leggermente più efficace da un punto di vista **qualitativo**, rispetto alla **Log**, in termini della metrica *NSD*, delineando in maniera più efficace i bordi dell'immagine *ground-truth* e una deviazione molto più bassa. Sebbene il valore di massimo e minimo registrato nel caso *NSD* risultino peggiori rispetto alla trasformazione **Log**, la trasformata **HF** risulta essere più consistente nei suoi risultati concentrandosi in un range più ristretto.

La trasformata **Inv** ha ottenuto mediamente risultati peggiori in ambe le due metriche, ma è possibile notare come il risultato che più l'ha distinta è nel caso del valore di massimo registrato nella metrica *Dice*, ciò sottolinea come le rimanenti trasformazioni, sebbene in grado di migliorare i risultati in generale, possano alterare l'input evitando il raggiungimento di migliori performance.

## 5.6 Segmentazioni

A seguire vengono mostrati i risultati riscontrati dalle diverse tecniche di **segmentazione**.

### Legenda trasformazioni

Otsu thresholding → *Otsu*

Gaussian Mixture Model → *GMM*

K-means → *KM*

	Dice				NSD			
Tipo	Mean	Max	Min	STD	Mean	Max	Min	STD
<i>KM</i>	0.5590	0.8516	0.1017	0.2149	0.4581	0.8154	0.049	0.2243
<i>Otsu</i>	0.5969	0.8557	0.2208	0.1735	0.4999	0.8219	0.1024	0.1901
<i>GMM</i>	0.4177	0.8189	0	0.2508	0.2508	0.6097	0	0.1754

### 5.6.1 Commenti sui risultati ottenuti

La segmentazione *Otsu* risulta essere la migliore sotto tutti i punti di vista, risulta essere mediamente migliore, ha ottenuto i migliori risultati di massimo e di minimo con un'ottima consistenza nella deviazione dei risultati ottenuti.

La segmentazione *KM* mostra risultati promettenti che si avvicinano molto a quelli della tecnica migliore *Otsu* potendo vantare di ottime prestazioni con risultati di massimo simili, ma con performance peggiore più accentuate nel caso del risultato minimo e nella deviazione dei valori.

La *GMM* risulta essere la segmentazione **meno efficiente** da ogni punto di vista rispetto alle altre tecniche proposte; è possibile notare come risulti essere più efficiente da un punto di vista quantitativo (Metrica *Dice*) rispetto all'aspetto qualitativo (Metrica *NSD*).

### Analisi dei risultati *KM* e *Otsu*

La somiglianza nei risultati di *Otsu* e *KM* può essere ricondotta alla natura dei modelli:

#### Varianza intraclasse Otsu

$$\sigma_W^2 = \sum_{i=1}^{len(T)} W_i \sigma_i^2$$

#### Funzione di costo K-means

$$J(\mu, c) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu_{c(i)}\|^2$$

Nell'ipotesi che il *KM* ottenga sempre il risultato ottimale, dove questa cosa è sempre vera nel caso *Otsu* perché completamente deterministica, la *Otsu* pesa il valore della varianza  $\sigma_i^2$  della classe  $i$  in base al numero di pixel appartenenti alla stessa classe con un apposito peso  $W_i$ . Nel caso del *KM*, ogni valore di pixel ha la stessa importanza nel minimizzare la funzione di costo.

$$J(\mu, c) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu_{c(i)}\|^2 = \frac{1}{m} \sum_{j=1}^{N^o \text{ Classi}} \sigma_j^2$$

In altre parole, il *KM* può essere visto come un caso specifico della *Otsu*.

$$J(\mu, c) = \frac{1}{m} \left( \sum_{j=1}^{N^o \text{ Classi}} \sigma_j^2 \right) = \sum_{j=1}^{N^o \text{ Classi}} W_j \sigma_j^2$$

### Analisi dei risultati *GMM*

Come già accennato in precedenza, il metodo necessita di una distribuzione dei dati tendente ad una gaussiana, cosa che in generale non viene mai rispettata entro determinati limiti. Nonostante tutto, si osserva come in alcune situazioni il metodo risulti essere efficiente, ma in generale non è consistente.



## 5.7 Pipeline (Trasformazione-Segmentazione)

A seguire sono mostrati i risultati riscontrati dalle specifiche pipeline utilizzate:

	Dice				NSD			
Tipo	Mean	Max	Min	STD	Mean	Max	Min	STD
<i>Inv_KM</i>	0.4745	0.8516	0.1017	0.2553	0.3443	0.6827	0.0491	0.2331
<i>Inv_Otsu</i>	0.5682	<b>0.8557</b>	0.2201	0.1946	0.4149	0.7019	0.1024	0.1944
<i>Inv_GMM</i>	0.3629	0.7825	0	0.3161	0.2245	0.5645	0	0.2082
<i>HF_KM</i>	0.6108	0.8022	0.3927	<b>0.1555</b>	0.5383	0.7828	<b>0.2323</b>	0.1698
<i>HF_Otsu</i>	0.6061	0.8004	0.3935	0.1564	<b>0.5458</b>	0.7713	0.2261	0.1601
<i>HF_GMM</i>	0.4152	0.7855	0.0398	0.2333	0.2649	0.6073	0.0256	0.1704
<i>Log_KM</i>	0.5917	0.8484	0.2800	0.2173	0.4916	0.8154	0.1397	0.2370
<i>Log_Otsu</i>	<b>0.6164</b>	0.8405	<b>0.3401</b>	0.1821	0.5393	<b>0.8219</b>	0.1533	0.2024
<i>Log_GMM</i>	0.4750	0.8189	0.2045	0.2034	0.2947	0.6097	0.0959	<b>0.1550</b>

### 5.7.1 Commenti sui risultati ottenuti

La pipeline con valori più precisi e consistenti risulta essere la *Log\_Otsu*, buone prestazioni sia in via quantitativa che qualitativa. Questa combinazione risulta essere anche la più semplice da configurare poiché sono presenti pochi parametri da tarare, considerando anche che la *Otsu* non richiede alcun tipo di configurazione.

Le pipeline *HF\_KM* e *HF\_Otsu* risultano essere ottime alternative; in particolare, la combinazione *HF\_Otsu* risulta essere più precisa in media in termini di *NSD*, permettendo quindi delineare meglio i bordi della zona interessata. Va sottolineato che nonostante il miglior risultato della *Log\_Otsu* nel valore massimo registrano nella *NSD*, le combinazioni che coinvolgono la trasformazione *HF* (Ad eccezione della segmentazione *GMM*) sono anche molto più stabili, in termini di deviazione standard, potendo quindi ottenere risultati più robusti per delineare la zona.

La peggiore combinazione risulta essere la *Inv\_GMM* poiché non in grado di fornire alcun risultato consistente, con valori di minimo e massimo peggiori. Sebbene anche la pipeline *Log\_GMM* risulti essere una delle meno efficienti mediamente, i valori di massimo e minimo suggeriscono a possibili migliorie poiché in alcune situazioni è in grado di fornire buoni risultati, in particolare nel caso *Dice*.

### Efficienza di implementazione

In via dei risultati ottenuti, una combinazione efficiente potrebbe prevedere un utilizzo incrociato delle tecniche *Log\_Otsu* e *HF\_Otsu*.

In caso di dati multidimensionali con numero di elementi eccessivi per singola immagine, le tecniche *Log\_KM* e *HF\_KM* sono valide alternative con un costo computazionale ridotto.



## 5.8 Risultati migliori ottenuti sui campioni analizzati

Campione	Dice			NSD		
	Valore	Trasformazione	Segmentazione	Valore	Trasformazione	Segmentazione
139	0.4151	<i>Log</i>	<i>Otsu</i>	0.3653	<i>HF</i>	<i>KM</i>
205	0.7101	<i>Inv</i>	<i>KM</i>	0.5964	<i>Log</i>	<i>KM</i>
262	0.4797	<i>HF</i>	<i>KM</i>	0.2323	<i>HF</i>	<i>KM</i>
277	0.8189	<i>Log</i>	<i>GMM</i>	0.6847	<i>Inv</i>	<i>Otsu</i>
307	0.7339	<i>Log</i>	<i>Otsu</i>	0.6252	<i>Log</i>	<i>Otsu</i>
308	0.8022	<i>Log</i>	<i>KM</i>	0.7367	<i>Log</i>	<i>Otsu</i>
328	0.8557	<i>Inv</i>	<i>Otsu</i>	0.8219	<i>Log</i>	<i>Otsu</i>
371	0.8230	<i>Log</i>	<i>Otsu</i>	0.7136	<i>Log</i>	<i>Otsu</i>
44	0.6648	<i>HF</i>	<i>KM</i>	0.5720	<i>HF</i>	<i>KM</i>
92	0.4359	<i>HF</i>	<i>KM</i>	0.5419	<i>HF</i>	<i>KM</i>

## Capitolo 6 | Conclusioni

---

### 6.1 Note conclusive con sviluppi futuri

Dall'analisi fornita dal progetto, è possibile affermare l'importanza di diversificare le tecniche utilizzate al fine di cogliere diversi aspetti dell'immagine analizzata e per migliorare la qualità della predizione.

#### 6.1.1 Preprocessing

Lo step di preprocessing è stato fondamentale al fine di migliorare la qualità delle predizioni che ne sono susseguite. Tecniche come *filtro gaussiano* e *PCA* sono state in grado di raffinare la qualità dell'immagine; al fine di migliorare ancor di più le performance, sarebbe necessario utilizzare diversificare le tecniche da utilizzare, con particolare attenzione ai parametri da tarare (e.g. intensità del filtro gaussiano e percentuale di varianza cumulativa da mantenere della *PCA*).

La tecnica che potrebbe portare a risultati più promettenti è la *Histogram Equalization*; sarebbe possibile effettuare studi più approfonditi per cercare una *CDF migliore* in grado di aumentare in maniera netta il contrasto tra le classi background, cervello e tumore presenti nell'immagine.

La scelta di una buona *CDF* potrebbe migliorare la distribuzione dei dati e differenze d'intensità tra classi migliorando a monte dalle tecniche di trasformazione e di segmentazione a seguire.

#### 6.1.2 Tecniche di trasformazione

La tecnica di trasformazione *logaritmica* e l'applicazione di un *homomorphic filter* sono state in grado di differenziarsi fornendo diverse informazioni, in termini di metriche *Dice* e *NSD*.

Per una migliore qualità delle segmentazioni, sarebbe possibile analizzare come combinazioni in parallelo o in serie di queste tecniche modifichino il risultato ottenuto.

I risultati ottenuti suggeriscono a un possibile approccio combinato della trasformata *logaritmica* con una ridifinitura dei bordi in funzione del *homomorphic filter*, migliorando la qualità della predizioni.

Possibili trasformate alternative da implementare, in via di sviluppi futuri, sono la *Discrete Wavelet Transform* e la *Fractional Fourier Transform*; la loro capacità di analisi di diverse strutture dell'immagine potrebbe essere determinante per migliorare la predizione con uno studio basato sulla frequenza.

#### 6.1.3 Tecniche di segmentazione

Sebbene la *Otsu Thresholding* si è dimostrata come la tecnica di segmentazione migliore, sarebbe possibile utilizzare un approccio più efficiente per l'assegnazione dei pesi  $W_i$  della varianza intraclass; un ipotetico miglioramento potrebbe prevedere un cambio del valore dei pesi in funzione di una diversa importanza tra le classi presenti (background, cervello e tumore).

La tecnica di segmentazione *K-means* è risultata una ottima alternativa alla *Otsu*, ulteriori analisi delle due tecniche potrebbe mettere in risalto le loro somiglianze e contrasti; sarebbero interessanti ulteriori analisi per la ricerca di una tecnica ibrida che riesca a cogliere e combinare i punti di forza delle due tecniche.

### 6.1.4 Analisi della Gaussian Mixture Model

La segmentazione *GMM*, sebbene sia stata la meno performante, ha dimostrato che in alcune situazioni, una segmentazione basata sulla distribuzione dei dati sia in grado di fornire buoni risultati (Campione 277, metrica *Dice* a monte di una trasformata *logaritmica*). L'utilizzo di una segmentazione in funzione delle distribuzioni dei dati potrebbe portare alla creazione di *Mixture Model* generici non limitandosi esclusivamente alla gaussiana ma su distribuzioni ad hoc per campione analizzato.

Il problema della *GMM*, riscontrato nel progetto, è derivato nell'approssimare la distribuzione della classe cervello e tumore a una gaussiana, dove nel caso del background è sufficientemente approssimabile (i valori nel background sono molto bassi e quindi basta una gaussiana con una varianza tendente a 0 potrebbe risultare sufficiente)

## 6.2 Appunti di riferimento al deep learning

L'utilizzo incrociato di algoritmi di image processing con modelli di deep learning rappresenta una interessante alternativa che permetterebbe non solo di migliorare le rispettive tecniche ma di espanderle.

### 6.2.1 Introduzioni di variabili latenti

Il modello fornito dal progetto è soggetto a limitazioni per quanto riguarda la scelta dei parametri; per limitare questo problema, è possibile allenare modelli di deep learning con lo scopo di **supporto**, il tutto al fine di stimare al meglio tali parametri, limitare il costo computazionale derivante da grandi modelli di deep learning e di migliorare la qualità delle predizioni ad hoc per campione analizzato.

### 6.2.2 Ensemble learning

L'utilizzo di algoritmi di image processing in un ensemble di modelli rappresenta una alternativa promettente; nell'ipotesi di avere algoritmi con buone prestazioni, il loro utilizzo permetterebbe di **ridurre il costo di allenamento** di ulteriori modelli di deep learning e di riuscire a ottenere bias differenti per una migliore predizione finale.

### 6.2.3 Problema di piccoli dataset

Una ulteriore alternativa dell'utilizzo di algoritmi di image processing è quella di **ingrandire la dimensione dei piccoli dataset**; in questo caso, il ruolo dell'immagine processing è di **supporto** a un modello di deep learning, utilizzando diverse trasformate per fornire delle **proiezioni** dell'input  $x'$  e dell'output  $Y'$ . Una possibile implementazione potrebbe prevedere l'ingrandimento del dataset mediante la generazione di coppie casuali  $(x, Y')$  e  $(x', Y)$ , le quali sarebbero viste come una sorta di **rumore** introdotto nel modello che potrebbe migliorarne le performance.

## Bibliografia

---

[1] The Medical Segmentation Decathlon ([link](#))

[2] Brain tumor segmentation of MRI images: A comprehensive review on the application of artificial intelligence tools, Computers in Biology and Medicine ([link](#))

[3] Efficient brain tumor segmentation using OTSU and K-means clustering in homomorphic transform, Biomedical Signal Processing and Control ([link](#))

[4] MRI-based brain tumor segmentation using Gaussian mixture model with reversible jump Markov chain Monte Carlo algorithm ([link](#))