Brain Cancer Segmentation

1 Introduzione

L'applicazione creata ha come tema principale l'analisi di immagini mediche. Nello specifico, vengono analizzate immagini di *risonanze magnetiche* (MRI) al cervello di pazienti. L'applicazione si suddivide in 2 parti principali, **Segmentazione del tumore** e **Interfaccia grafica** (GUI).

2 Dataset utilizzati

Per svolgere l'intero programma sono stati utilizzati 2 dataset differenti

- Dataset **formato .mat** scaricato su kaggle. Il dataset contiene 3064 RMI T1 con mezzo di contrasto svolte a 233 pazienti con 3 tipi di tumori differenti: meningioma (708), glioma (1426), e pituitary tumor (930). Il dataset è suddiviso in 4 directory contenenti 766 file ciascuna. Ogni file è in formato matlab(.mat) e contiene le seguenti informazioni
  - *cjdata.label* : 1 for meningioma, 2 for glioma, 3 for pituitary tumor
  - *cjdata.PID* : patient ID
  - *cjdata.image* : image data
  - *cjdata.tumorBorder*
  - *cjdata.tumorMask* : a binary image with 1s indicating tumor region
- Dataset **formato .jpg** scaricato su kaggle. Questo contiene 3264 files così divisi, ci sono due directory: "Testing" e "Training", ogni directory contiene altre 4 directory: glioma, pituitary, meningioma, no-tumor. Il formato delle immagini è ".jpg".

3 Segmentazione del tumore

La segmentazione del tumore è un task molto complesso in quanto le immagini mediche possono avere intensità di luce differenti e riconoscere il contorno di un area anomala all'interno del tessuto cerebrale non sempre risulta essere banale. Inoltre acquisire un dataset non è semplice per ragioni di privacy sui pazienti. Per raggiungere l'obiettivo, abbiamo suddiviso il problema in tre principali parti

- Skull Stripping
- Estrazione del colore medio del tumore
- Segmentazione

3.1 Skull Stripping

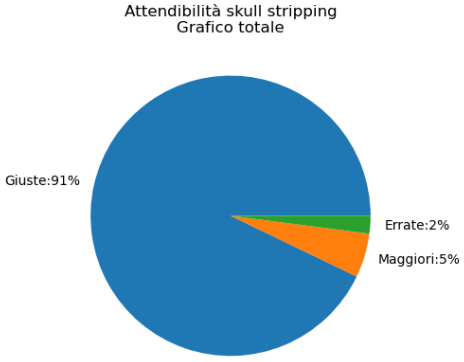
Per Skull Stripping si intende l'estrazione del **tessuto cerebrale** da una MRI andando a eliminare i bordi che possono influenzare la "*tumor detection*".

Algortimo

Il processo di skull stripping viene fatto andando a calcolare **l'intensità del pixel** presente con maggior frequenza considerando un range, in un immagine in scala di grigi, tra 20 e 190(non vengono considerati i pixel vicini al nero e vicini al bianco). Viene applicato un *filtro di Canny* per delimitare i contorni nel range che va da -10 a +10 rispetto all'intensità del pixel trovato precedentemente. Si cerca di trovare il contorno che più di tutti ha la forma di un cervello andando a prendere solo il tessuto cerebrale (di colore grigio) estrapolando il **contorno più esterno**. L'immagine ritornata non contiene altre parti del corpo che, avendo intensità di luce diverse rispetto al tessuto cerebrale, possono ostacolare il riconoscimento di un tumore, in quanto queste parti potrebbero essere proprio scambiate per tumori. L'obiettivo è quello di avere un immagine con solo il cervello, in modo che, qualsiasi differenza netta di intensità possa essere, con molta probabilità, il **segmento di un tumore**.

3.1.1 Testing

Per valutare l'accuratezza del programma sono stati effettuati dei test sull'intero dataset formato .mat. Per capire se l'estrazione del contenuto del cervello è avvenuta correttamente, il programma inizialmente calcola **l'area del contorno più esterno** (tessuto cerebrale + bordo), effettua lo *skull stripping* e, sul contorno che viene ritornato, calcola *l'area* e *la circolarità*. Se l'area è troppo piccola rispetto all'area più esterna, o se il contorno è poco circolare, con buona probabilità il processo di skull stripping non ha funzionato correttamente, in caso contrario assumiamo che il processo abbia funzionato correttamente. E' presente un caso medio in cui con buona probabilità il processo ha estratto il tessuto cerebrale ma non è riuscito a levare interamente il bordo.



3.2 Estrazione del colore medio del tumore

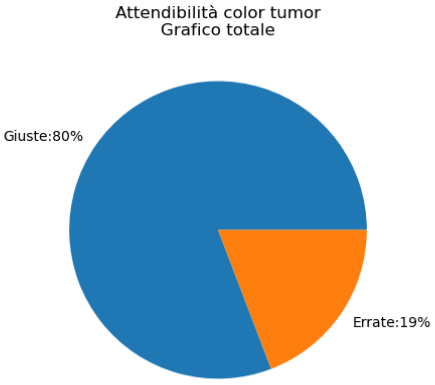
In questa fase l'obiettivo è quello di analizzare l'immagine che contiene solo il **tessuto cerebrale senza bordi** e trovare il **colore medio del tumore** (qualora sia presente).

Algoritmo

Viene calcolato il colore medio del cervello andando a prendere l'intensità del pixel presente con maggiore frequenza all'interno dell'immagine con solo il tessuto cerebrale. A questo punto vengono analizzati tutti i contorni che, oltre a rispettare alcune caratteristiche come la circolarità e l'area, hanno un colore medio distante rispetto al colore medio del cervello. Differenza tra **colore medio del cervello** e **colore medio del contorno**,uniti ad *area* e *circolarità* sono le tre proprietà che caratterizzano e distinguono ogni contorno trovato, il contorno che rispetta maggiormente queste proprietà viene scelto come possibile tumore, e viene restituito il suo colore medio. Il colore di questo contorno può non essere del tutto preciso e, come vedremo nel prossimo step, è soltanto un ulteriore parametro che viene utilizzato per confrontarlo con il colore medio dei contorni che vengono analizzati successivamente con maggiore precisione.

3.2.1 Testing

In questa fase, per valutare la correttezza sono stati effettuati dei test su 200 immagini prese da una delle 4 directory del dataset formato .mat. Per sapere se il colore trovato fosse corretto oppure no, è stato confrontato analizzando il **colore medio della maschera del tumore** presente su ogni immagine nel dataset (*cjdata.tumorMask*).



3.3 Segmentazione

Acquisito il **colore medio del tumore** e **l'immagine del tessuto cerebrale** diventa più facile analizzare l'immagine per effettuare la "*tumor detection*" e infine la "*tumor segmentation*".

Algortimo

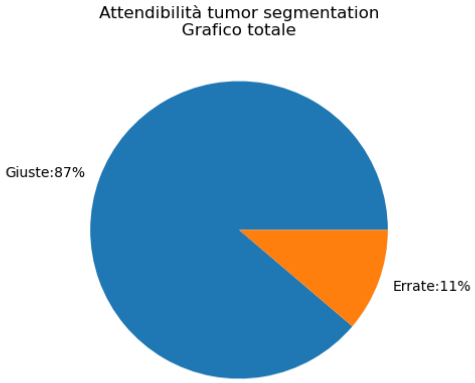
Viene utilizzato l'algoritmo **K-Means Clustering** con un k=6 che partiziona l'immagine in modo che ogni pixel appartiene ad uno tra i **6 cluster**, ogni cluster ha un **intensità di colore differente**. Per ogni intensità (cluster), creo una **maschera**, ovvero un immagine nera in cui solo i pixel che fanno riferimento al cluster corrente che sto analizzando sono interamente bianchi. Trovo i **contorni della maschera corrente** e, per ogni contorno controllo che rispetti alcune proprietà, quali: circolarità di un certo valore, area rispetto al contorno più esterno (precedentemente calcolato), non troppo piccola e non troppo grande, differenza non troppo elevata tra colore medio del contorno, calcolato rispetto all'immagine iniziale e colore medio del tumore calcolato nello step precedente. Tutti i contorni che rispettano queste proprietà vengono associati ad un **valore numerico univoco** che viene calcolato a partire da queste caratteristiche attraverso una **media pesata**, il contorno che ha la media pesata più alta e sopra un certo valore viene considerato un **tumore**. Se la media pesata più alta non supera questo valore allora vuol dire che il programma non ha trovato nessun tumore.

3.3.1 Testing

Anche qui sono stati effettuati dei test per valutare la precisione della segmentazione su 200 immagini prese da una delle 4 directory del dataset formato .mat. Per valutare se, data un immagine in input, il programma trova il tumore correttamente, sono state confrontate le seguenti immagini

- Immagine nera con solo **il tumore trovato dal programma**
- Immagine nera con **il tumore presente nel file .mat** (*cjdata.tumorMask*)

La funzione per il confronto immagini ritorna un **p** da 0 a 1 che indica quanto due immagini sono uguali (più p è vicino ad 1 e più le immagini sono simili), per ogni p maggiore o uguale a 0.97 assumiamo che il tumore sia stato preso con esattezza, in tutti gli altri casi no. Quindi vengono considerati sbagliati i casi in cui la segmentazione non è precisa o i casi in cui il tumore è presente ma il programma non lo trova.



Conlusione Segmentazione

E' possibile concludere che, con una certa approssimazione, la segmentazione del tumore ha una **percentuale di accuratezza del 90%**, per cercare di migliorare ancora di più andando ad alzare la percentuale è possibile provare a migliorare *l'accuratezza dello skull stripping* e/o *l'estrazione del colore medio*, questo perchè influenzano la segmentazione in quanto il programma di segmentazione prende in input questi dati.

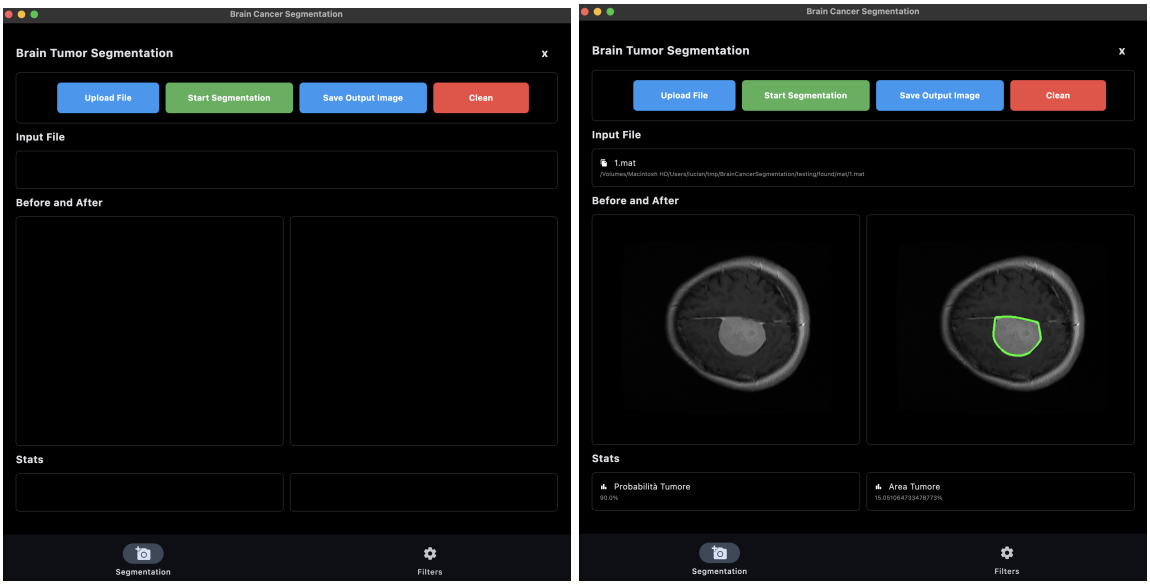
4 Interfaccia grafica

L'interfaccia grafica (*GUI*) a sua volta è suddivisa in 2 sezioni **Brain Tumor Segmentation** e **Segmentation Filters**.

4.1 Brain Tumor Segmentation

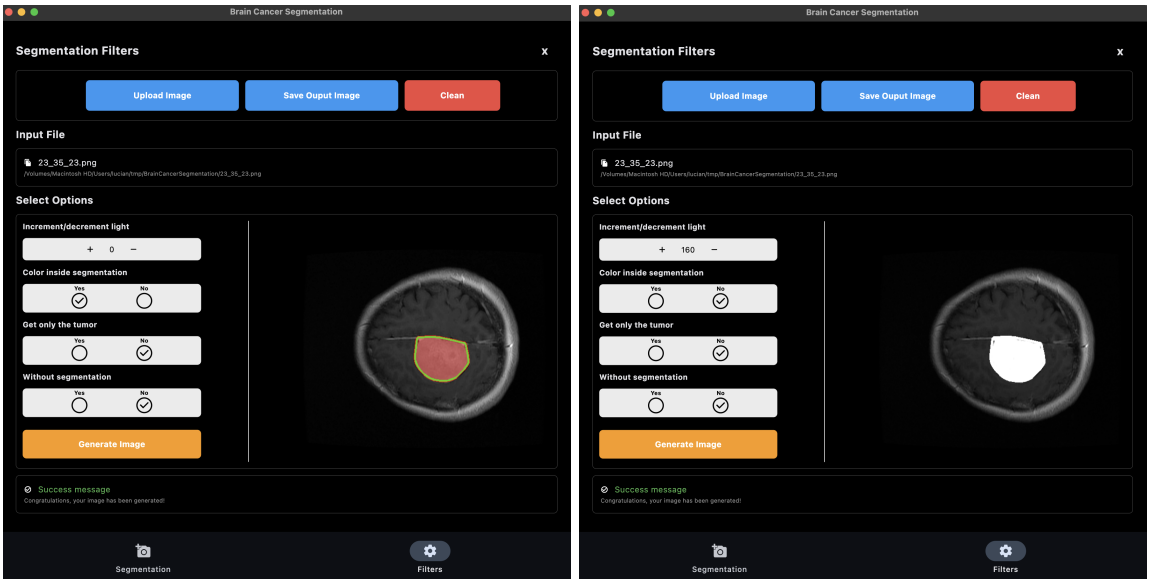
Eseguendo il file **run.sh** situato nella directory principale, si apre un interfaccia in cui l'utente ha la possibilità di caricare un immagine dal file system (formati accettati .mat/.jpg/.png). L'immagine scelta,apparirà nello

spazio riservato alle due immagini "*Before and After*", sotto la sezione "*Before*". Successivamente,è possibile avviare la **segmentazione selezionando** l'apposito pulsante "*Start Segmentation*". *L'immagine con il tumore* (*qualora ci fosse*) segmentato apparirà sulla destra,sotto la sezione "*After*". Inoltre, viene mostrata **l'area del tumore trovato** e **la probabilità** che il programma abbia trovato effettivamente un tumore. Se il programma non trova nessun tumore,a destra comparià la scritta "*TUMOR NOT FOUND*". Infine, l'utente può salvare l'immagine del tumore segmentato, selezionando il pulsante "*Save Output*", scegliendo, all'interno del proprio computer, dove andarla a memorizzare.



4.2 Segmentation Filters

L'utente può caricare un immagine che il programma ha generato in un primo momento, ovvero un immagine con il **tumore segmentato** e applicare dei filtri. Ad esempio è possibile **aumentare/diminuire il colore del tumore** in modo da aumentare il contrasto con il resto della foto. E' possibile vedere e salvare l'immagine **contenente solo il tumore**, su sfondo nero e **colorare l'interno della segmentazione**. Infine, è possibile **eliminare il segmento del tumore** che il programma aveva creato, portando l'immagine come era prima della segmentazione.



4.3 Esempio di esecuzione

Illustriamo al seguente link un esempio di esecuzione dell'applicazione.