



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

“Machine learning per l'imaging biomedico: valutazione di DeepLabv3 per la segmentazione dell'ecografia epatica”

Relatore: Prof. Loris Nanni

Laureando: Riccardo Girotto

ANNO ACCADEMICO 2022 - 2023

Data di laurea 19/07/2023

*Alla mia famiglia e ai miei amici,
che mi hanno reso ciò che sono.*

Sommario

L'ecografia epatica è un'esame diagnostico prescritto in caso di sintomi associati ad una malattia del fegato ed è comunemente utilizzata in quanto caratterizzata da rapidità, basso costo, facilità d'uso, portabilità e natura non invasiva. La segmentazione in tempo reale dell'ecografia epatica è essenziale per delineare un quadro clinico chiaro riguardo la morfologia e le condizioni dell'organo (ad esempio la presenza di un carcinoma epatocellulare o di una steatosi epatica) e per assistere i chirurghi e i radiologi nelle procedure terapeutiche. L'obiettivo di questa tesi è quello di studiare l'impatto del *preprocessing* delle immagini sulle prestazioni di un rete neurale addestrata per la segmentazione di ecografie epatiche. La rete è stata realizzata su MATLAB con il modello DeepLabv3Plus con *backbone* ResNet18 ed è stata allenata con un *dataset* di ecografie disponibile pubblicamente contenente 2400 immagini di 8 volontari sani.

Indice

1	Introduzione	7
1.1	Medical imaging e machine learning	7
1.2	Segmentazione e pre-processing	7
2	Dettagli modello	9
2.1	Dataset e architettura	9
2.2	Loss function	10
3	Preprocessing	12
3.1	Contrast enhancement: CLAHE	12
3.2	Edges enhancement: Sobel	13
3.3	Noise reduction: Side windows	14
4	Risultati	17
4.1	Metriche di valutazione	17
4.2	Analisi delle performance	18
4.3	Conclusione	19

Introduzione

1.1 Medical imaging e machine learning

Il campo del *medical imaging* è uno dei pilastri della pratica medica moderna perché consente di visualizzare ed analizzare dettagli anatomici e funzionali all'interno del corpo umano in modo non invasivo. Attraverso l'utilizzo di tecnologie come la risonanza magnetica, la tomografia computerizzata e l'ecografia, è possibile ottenere immagini di alta qualità che forniscono informazioni cruciali per la diagnosi e il monitoraggio dei pazienti.

Nella scienza moderna, l'applicazione del *machine learning* al campo medico sta rivoluzionando la pratica clinica e la ricerca biomedica. Il *machine learning* è una branca dell'intelligenza artificiale focalizzata sulla creazione di algoritmi e modelli che consentono ai computer di apprendere e migliorare le proprie prestazioni con l'esperienza, individuando schemi e correlazioni tra i dati di input senza essere esplicitamente programmati. L'introduzione del *machine learning* nel campo del *medical imaging* ha ulteriormente potenziato le capacità di queste tecnologie diagnostiche. L'apprendimento automatico consente ai computer di analizzare grandi quantità di dati provenienti dalle immagini mediche, identificando pattern e relazioni complesse che potrebbero sfuggire all'occhio umano e facilitando l'automazione di compiti ripetitivi e laboriosi. L'applicazione del *machine learning* rappresenta un'opportunità per migliorare la precisione, l'efficienza e l'efficacia dei processi clinici, aprendo nuove possibilità nella classificazione di patologie, nell'analisi di caratteristiche rilevanti e nella segmentazione di immagini.

1.2 Segmentazione e pre-processing

L'ecografia epatica è una procedura di *imaging medico* che utilizza delle onde sonore ad alta frequenza per produrre immagini del fegato. Nonostante i benefici di questo esame diagnostico, i medici si trovano ad affrontare delle sfide nell'analisi manuale delle ecografie:

1. la forma e la dimensione dell'area di interesse possono variare a seconda dell'orientamento e della posizione della sonda ad ultrasuoni;
2. la qualità delle immagini è generalmente compromessa da rumore di *speckle* e da ombre di altre strutture anatomiche, come quelle di polmoni e costole;
3. i confini anatomici non sono sempre chiari a causa del basso contrasto delle ecografie e di onde sonore rifratte in modo inaccurato da tessuti con differenti echogenicità.

Per superare questi ostacoli e delineare con precisione l'area di interesse in modo più semplice, rapido e indipendente dall'operatore, le intelligenze artificiali utilizzano la segmentazione. Questo processo di divisione di un'immagine digitale in sottogruppi chiamati segmenti, è utilizzato per ridurre la complessità e rappresentarla in un modo più facile da analizzare ed elaborare. Per migliorare le prestazioni e l'accuratezza delle segmentazioni, in particolar modo per fronteggiare gli ultimi due problemi elencati precedentemente, è necessario l'utilizzo del *preprocessing* delle immagini. Esso è una fase di elaborazione delle immagini che avviene prima della segmentazione e che comprende operazioni di modifica, trasformazione e filtraggio delle immagini. Questa tesi si pone l'obiettivo di studiare gli effetti del *preprocessing* sulle ecografie epatiche e si focalizza sulle seguenti tipologie di tecniche:

1. *Contrast enhancement*: regola i livelli di contrasto dell'immagine per aumentare la differenza tra le diverse regioni, migliorando i dettagli e la forma delle strutture;
2. *Edges enhancement*: evidenzia la definizione dei bordi e contorni presenti nell'immagine, rendendoli più nitidi e distinti.
3. *Noise reduction*: rimuove o riduce artefatti indesiderati o interferenze presenti nell'immagine;

Dettagli modello

2.1 Dataset e architettura

Il *dataset* utilizzato per questa tesi è pubblico e proviene da una ricerca sull'applicazione di Dense-PNP-UNet e diverse altre reti neurali sulle ecografie epatiche[1]. I ricercatori hanno estratto le immagini da un *dataset* di video realizzati per la sfida CLUST sul tracciamento degli ultrasuoni epatici presso il Dipartimento di Radiologia dell'Ospedale Universitario di Ginevra, utilizzando una macchina per ultrasuoni Siemens[2]. I video degli ultrasuoni epatici di 8 volontari, codificati in scala di grigi a 8 bit, sono stati elaborati per estrarre 300 fotogrammi per paziente utilizzando i primi 2 minuti di ogni video. Successivamente le immagini sono state annotate manualmente da tre radiologi qualificati utilizzando lo strumento di annotazione Computer Vision Annotation Tool (CVAT), per aggiungere al *dataset* pubblico anche le *ground-truth* di riferimento.

Le immagini più complesse del *dataset*, ovvero quelle del paziente '8', presentano bordi vaghi, ombre causate dai polmoni e interferenze dovute alle costole e per questo sono state utilizzate esclusivamente per testare il modello. Il *dataset* restante è stato suddiviso in *training set* e *validation set* automaticamente da MATLAB (75% - 25%) in ogni sessione di allenamento.

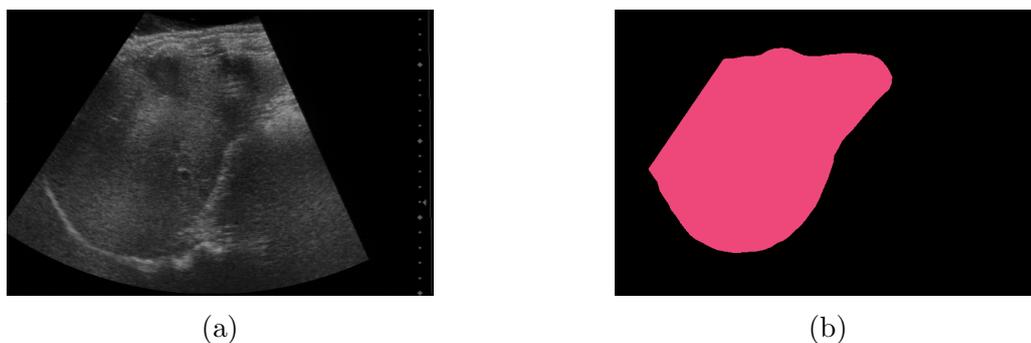


Figura 2.1: La figura rappresenta un'ecografia del paziente '2'(a) e la relativa *ground-truth*(b).

DeepLabv3Plus è un'architettura di rete neurale convoluzionale utilizzata per la segmentazione semantica delle immagini. È una versione migliorata del modello DeepLabv3, il cui obiettivo è migliorare la precisione e la definizione delle predizioni di segmentazione[3]. Il modello adottato per questa tesi utilizza un *backbone* ResNet-18, una rete neurale convoluzionale pre-addestrata su un'ampia quantità di dati per estrarre caratteristiche rilevanti dalle immagini di input a diverse profondità. Questa combinazione di architettura e *backbone* è ampiamente utilizzata per applicazioni che richiedono una

segmentazione accurata, come il riconoscimento di oggetti, l'analisi medica e la guida autonoma[4]. La buona capacità di apprendimento e il consumo relativamente ridotto di risorse computazionali, rendono ResNet18 un buon compromesso tra efficienza e leggerezza computazionale. Per questo motivo il modello utilizzato può essere eseguito anche su dispositivi con risorse limitate, come un portatile con una GPU modesta, senza compromettere significativamente le prestazioni.

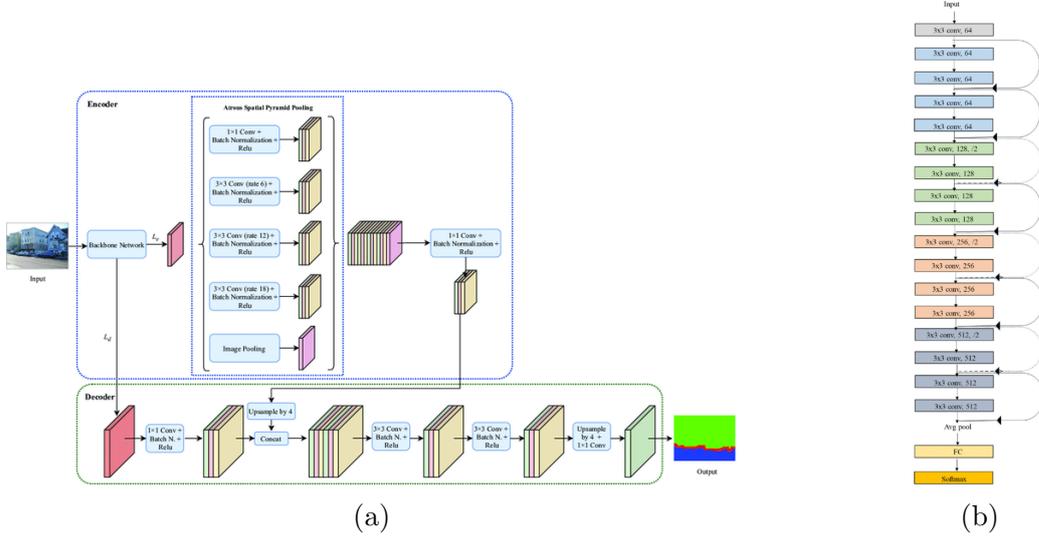


Figura 2.2: La figura rappresenta lo schema del modello DeepLabv3Plus(a) e l'architettura del *backbone* ResNet18(b).

2.2 Loss function

Con il termine *loss function* si intende la funzione che calcola la distanza tra l'output attuale dell'algorithmo e l'output desiderato, quantificando l'errore del modello di *machine learning*. Il suo obiettivo è quello di fornire un segnale di retroazione durante l'allenamento della rete, consentendole di aggiornare i suoi parametri e massimizzare le prestazioni. La scelta della *loss function* è quindi essenziale per permettere alla rete di apprendere in modo corretto e generare maschere di segmentazione accurate. Per segmentare correttamente le ecografie epatiche la *loss function* deve dare importanza alla distribuzione, alle diverse regioni e ai bordi dell'immagine di riferimento. La funzione di perdita utilizzata in questa tesi è composta dalla somma di due *loss* modificate, la *combo loss* e la *surface loss*[1].

La *combo loss* è definita come media pesata della *Dice loss* e della *cross entropy loss* e fornisce informazioni utili sulla distribuzione e sull'area della *ground-truth*. La versione implementata utilizza invece una *focal loss* per penalizzare maggiormente i pixel segmentati in modo errato e si affida alla *Dice loss* per informazioni rilevanti sulla sovrapposizione tra predizione e *ground-truth*. La *combo loss* può essere descritta come:

$$CL(y, \hat{y}) = FL(y, \hat{y}) + DL(y, \hat{y}), \quad (2.1)$$

dove y è la *ground-truth* e \hat{y} è la maschera segmentata dalla rete; FL e DL sono rispettivamente la *focal loss* e la *Dice loss*.

La *surface loss* viene applicata su *datasets* altamente sbilanciati ed affronta variazioni considerevoli dell'immagine in modo più efficace rispetto alle funzioni di perdita *region-based*[1]. Essa è composta da due funzioni di perdita che generano informazioni complementari, ovvero la *Dice loss* generalizzata vista in precedenza e la *boundary loss*, che si concentra sulle aree di confine dell'immagine. Quest'ultima è calcolata come moltiplicazione tra una funzione binaria che rappresenta le maschere generate dalla rete e una mappa pre-calcolata basata sulle distanze di ogni pixel dai bordi ad esso più vicino[5]. Può essere scritta in questo modo:

$$B_L(y, \hat{y}) = \int_{\Omega} \phi_G(p) M_{\theta}(p) dp, \quad (2.2)$$

dove ϕ_G è la mappa delle distanze, M_{θ} la funzione binaria e Ω rappresenta il dominio.

Il valore della *boundary loss* può essere calcolato anche attraverso la convoluzione con un filtro laplaciano, che genera risposte forti attorno alle zone di confine e zero in tutti gli altri punti[6]. Applicando il filtro alla maschera S otteniamo:

$$L(x, y) = \frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} \quad (2.3)$$

L'implementazione realizzata su MATLAB si basa sul calcolo della funzione di perdita come differenza tra la *ground-truth* filtrata Y e la maschera di segmentazione filtrata \hat{Y} , ovvero:

$$L_{BE} = |L(Y) - L(\hat{Y})|^2 = \left\| \frac{\partial^2(Y - \hat{Y})}{\partial x^2} - \frac{\partial^2(Y - \hat{Y})}{\partial y^2} \right\|^2 \quad (2.4)$$

La *surface loss* utilizzata in questa tesi sostituisce alla *Dice loss*, la *combo loss* calcolata al punto 2.1 e utilizza dei pesi per regolare l'influenza delle due funzioni di perdita durante il *training*. I pesi sono inizialmente impostati come 0.99 e 0.01, rispettivamente per la *combo loss* e la *boundary loss*, focalizzando il *learning* nell'area della regione d'interesse(ROI). Durante l'allenamento i pesi variano di 0.1 in ogni epoca per permettere alla rete di concentrarsi sui confini della ROI. La *surface loss* utilizzata può quindi essere scritta come:

$$SL(y, \hat{y}) = \alpha CL(y, \hat{y}) + (1 - \alpha) BL(y, \hat{y}), \quad (2.5)$$

dove y è la *ground-truth*, \hat{y} è la maschera segmentata dalla rete e α ha un valore che decresce da 0.99 a 0.01; CL e BL sono rispettivamente la *combo loss* 2.1 e la *boundary loss* 2.4.

Preprocessing

In questa sezione verranno trattati tre *preprocessing* implementati e testati su MATLAB, che si focalizzano su tecniche diverse. Il *dataset* ha subito una prima fase di elaborazione comune nella quale le immagini sono state ridimensionate per ottenere una dimensione di 512x512 e le *ground-truths* segmentate manualmente sono state trasformate in maschere binarie (colore bianco per la classe "fegato" e nero per il resto dell'immagine).

3.1 Contrast enhancement: CLAHE

L'istogramma di un'immagine è una rappresentazione grafica della distribuzione delle intensità dei pixel presenti in quell'immagine. L'asse orizzontale dell'istogramma rappresenta le diverse intensità dei pixel, mentre l'asse verticale rappresenta il numero di pixel che hanno quella specifica intensità. L'equalizzazione dell'istogramma è una tecnica che viene utilizzata per migliorare il contrasto di un'immagine, modificandola in modo tale che la sua distribuzione delle intensità sia uniforme su tutto lo spettro dinamico. Ciò comporta un allargamento delle differenze di intensità tra i pixel, aumentando il contrasto dell'immagine. Tuttavia nel caso di immagini mediche, a causa del contrasto limitato e di intensità molto diverse, l'equalizzazione classica può portare a un'eccessiva amplificazione del rumore e a un aspetto innaturale dell'immagine. Per superare questa limitazione è stata sviluppata una variante chiamata CLAHE (Contrast Limited Adaptive Histogram Equalization)[7].

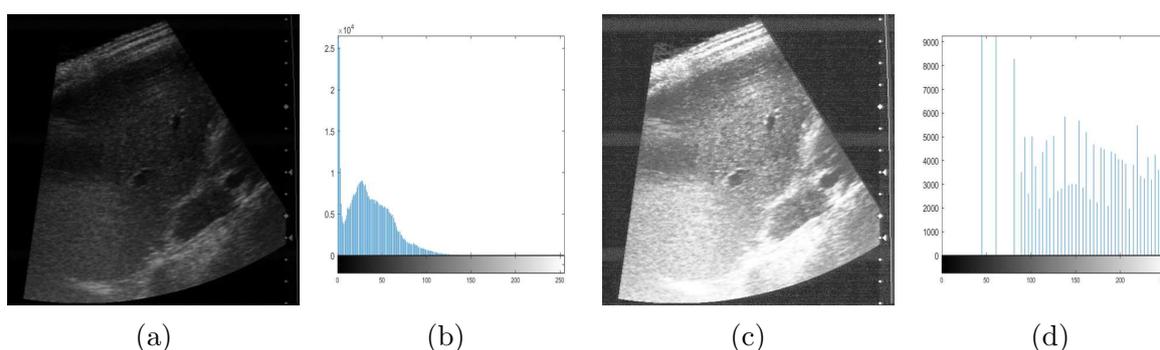


Figura 3.1: La figura rappresenta un'ecografia del paziente '7'(a), il suo istogramma corrispondente(b), l'immagine ottenuta in seguito all'equalizzazione dell'istogramma(c) e l'istogramma dopo l'equalizzazione(d).

CLAHE è un algoritmo di miglioramento del contrasto che si adatta localmente alle caratteristiche dell'immagine. Esso suddivide l'immagine in blocchi più piccoli, chiamati *tiles*, e applica l'equalizzazione dell'istogramma a ciascun *tile* in modo indipendente.

Ciò consente di ottenere un miglioramento del contrasto limitato localmente, evitando l'amplificazione del rumore. L'algoritmo utilizza inoltre un ulteriore meccanismo di limitazione del contrasto per evitare un'eccessiva amplificazione dell'istogramma, soprattutto nelle regioni con intensità molto diverse. Questa tecnica offre un'alternativa efficace all'equalizzazione dell'istogramma tradizionale, consentendo un miglioramento del contrasto localmente adattivo senza compromettere la naturalezza dell'immagine. In questa tesi CLAHE è stato implementato con un $clipLimit = 0.2$, che definisce la limitazione del contrasto, e con una $NumTiles = [16,16]$, che esplicita il numero di colonne e righe in cui viene suddivisa l'immagine (quindi 256 *tiles*).

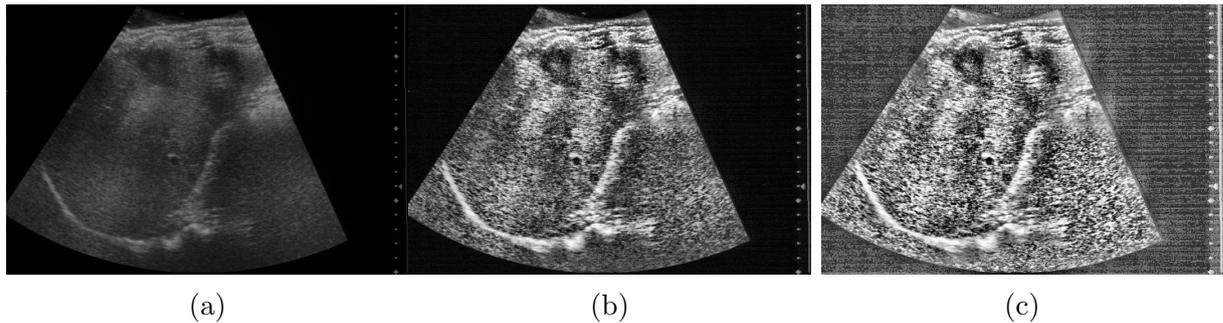


Figura 3.2: Nella figura si possono comparare gli effetti del *preprocessing* CLAHE con $clipLimit = 0.02$ (b) e $clipLimit = 0.2$ (c), rispetto all'immagine originale (a) del paziente '2'.

3.2 Edges enhancement: Sobel

Il secondo *preprocessing* si focalizza, come il precedente, sulle differenze di intensità dei pixel e sul miglioramento del contrasto, ma sfruttando un filtro specifico *edge-preserving* noto come *filtro di Sobel*. Questa tipologia di filtri è ottima per migliorare la qualità dell'immagine prima di una segmentazione riducendo rumore e dati non rilevanti, preservando al contempo i contorni degli oggetti presenti nella stessa.

Il *filtro di Sobel* utilizza due maschere di convoluzione G_x e G_y , rispettivamente per il calcolo del gradiente orizzontale e di quello verticale. Queste maschere sono matrici 3x3 predefinite che contengono i coefficienti di convoluzione necessari per calcolare le derivate parziali rispetto alle direzioni orizzontali e verticali. Il calcolo viene eseguito posizionando la maschera G_x sull'immagine e determinando la somma dei prodotti tra i valori dei pixel dell'immagine e i coefficienti corrispondenti di G_x . Questo processo viene ripetuto per ogni pixel fino ad ottenere un risultato che rappresenta il gradiente; G_y funziona in modo analogo.

I gradienti rappresentano la forza e la direzione delle variazioni dell'immagine e per questo possono essere utilizzati per evidenziare le transizioni di intensità all'interno della stessa. Quindi il *filtro di Sobel* risulta particolarmente vantaggioso nel caso delle ecografie epatiche, caratterizzate da bordi con intensità molto alte o molto basse. Dopo essere

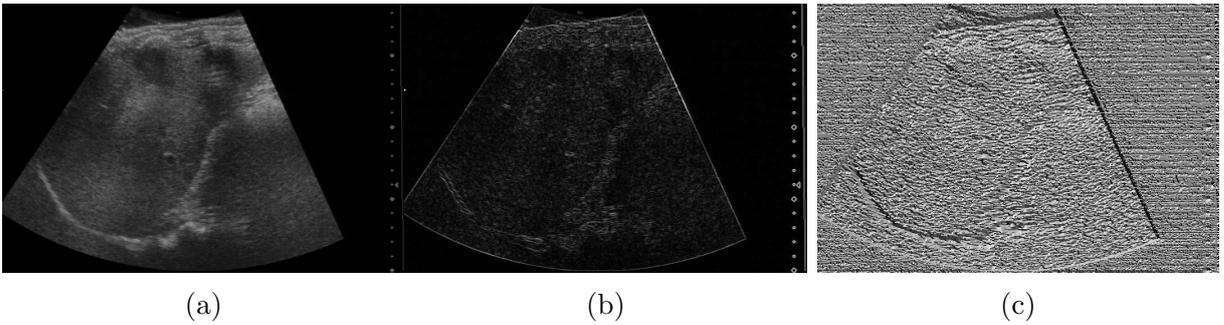


Figura 3.3: La figura rappresenta un'ecografia del paziente '2' (a) e i gradienti G_x e G_y calcolati con il *filtro di Sobel*, rispettivamente (b) e (c).

stato calcolato, il gradiente G_x viene normalizzato ridimensionando tutti i suoi valori in un intervallo $[0,1]$ e viene moltiplicato all'immagine originale enfatizzandone i bordi.

Dopo aver utilizzato il gradiente di Sobel, i risultati sono stati arrotondati e scalati nella corretta visualizzazione dei valori dei pixel ($[0,255]$) e hanno subito un ultimo processo di *contrast enhancement*. *Localcontrast* è una funzione per il miglioramento locale del contrasto implementata da MATLAB. Essa si basa su una tecnica nota come *adattamento del contrasto locale* che si basa sull'analisi delle differenze di intensità tra i pixel all'interno di una determinata regione dell'immagine. La funzione regola il contrasto sulla base della media e della deviazione standard raccolte all'interno di una finestra di dimensioni adattabili dall'utente. L'applicazione di *localcontrast* permette di migliorare ulteriormente la visibilità dei dettagli all'interno dell'immagine evidenziando bordi e strutture. Le immagini sono state processate con una soglia minima di variazione d'intensità $edgeThreshold = 0.2$ e un'entità di adattamento del contrasto $amount = 0.2$.

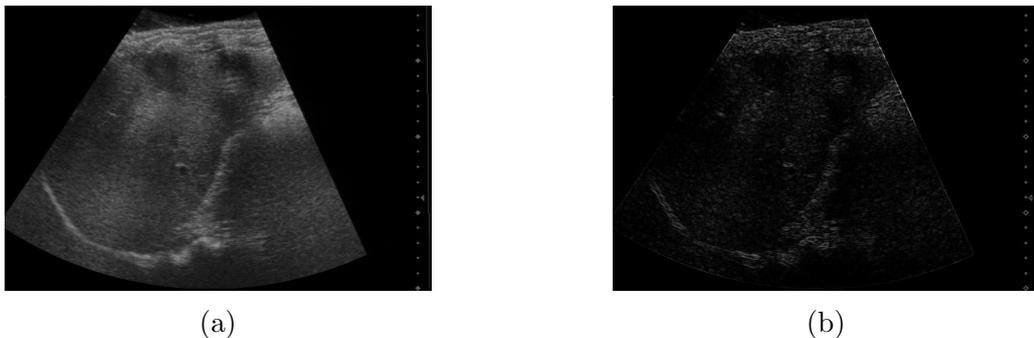


Figura 3.4: Nella figura si possono comparare gli effetti del *preprocessing* con *filtro di Sobel* e *localcontrast*(b), rispetto all'immagine originale (a) del paziente '2'.

3.3 Noise reduction: Side windows

Il terzo *preprocessing* presentato utilizza un algoritmo di *noise reduction* che dà massima priorità alla conservazione dei bordi, noto come *Side Window Filtering (SWF)*[8]. Questo metodo sfrutta l'uso di finestre laterali e un sistema di filtraggio capace di adattarsi ai bordi dell'immagine per ridurre la sfocatura e lo smussamento dei contorni, mantenendone

al contempo la definizione. L'algoritmo utilizza 8 finestre laterali specifiche definite da due parametri: r , che rappresenta il raggio della finestra, e $\theta = k \times \frac{\pi}{2}$, che rappresenta l'angolo tra la finestra e la linea orizzontale; con k che assume valori nell'intervallo $[0,3]$. Le finestre laterali sono disposte in modo tale da allineare i lati della finestra con il lato (*down*, *right*, *up*, *left*) o con l'angolo (*southwest*, *southeast*, *northeast*, *northwest*) del pixel di interesse.

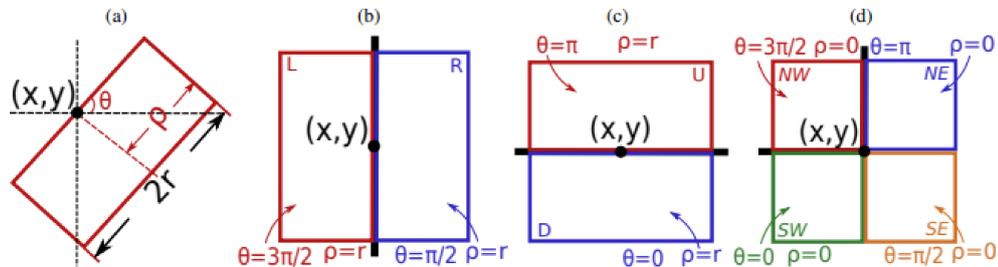


Figura 3.5: La figura rappresenta una generica finestra laterale definita da θ e p compreso tra 0 e r (a), le finestre *left*(rettangolo rosso) e *right*(rettangolo blu)(b), le finestre *up*(rettangolo rosso) e *down*(rettangolo blu)(c) e le finestre *northwest* (rettangolo rosso), *northeast* (rettangolo blu), *southwest* (rettangolo verde) e *southeast* (rettangolo arancione)(d).

Durante il processo vengono applicate 8 convoluzioni tra le finestre filtrate e l'immagine originaria e viene selezionato l'output con distanza minima rispetto all'intensità dei pixel dell'input originale. Il risultato ottenuto rappresenta il valore finale del processo SWF per quel pixel specifico. L'algoritmo SWF è in grado di preservare i bordi perché lavora su finestre locali intorno ai pixel, consentendo di adattare l'elaborazione ai dettagli specifici presenti nelle vicinanze dei bordi. Questo algoritmo è stato implementato nello specifico con un raggio $r = 2$ e con un numero di iterazioni del processo di convoluzione pari a 20, in cui l'immagine originale viene modificata e aggiornata in ogni ciclo.

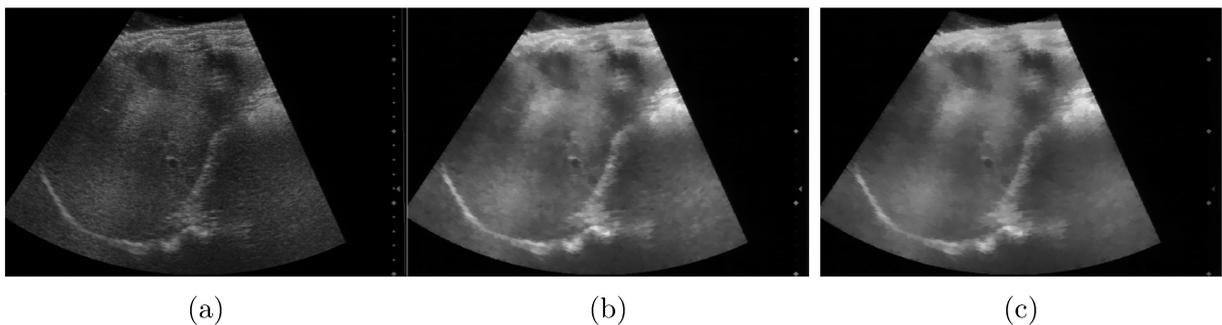


Figura 3.6: Nella figura si possono comparare gli effetti del *preprocessing* SWF con $r=2$, $numIterazioni=20$ (b) e $r=2$, $numIterazioni=50$ (c), rispetto all'immagine originale (a) del paziente '2'.

Infine le immagini ottenute sono state ulteriormente processate con una funzione di *sharpening* implementata da MATLAB, nota come *imsharpen*, che migliora la definizione e la nitidezza dei bordi. L'algoritmo di *sharpening* si basa sulla tecnica di accentuazione

dei bordi, che consiste nell'accentuare le differenze di intensità tra i pixel adiacenti per migliorare la percezione dei dettagli. Per ottenere questo effetto la funzione genera una *mappa di dettaglio*, ottenuta sottraendo all'immagine originale I_0 una sua versione filtrata e sfocata, che viene successivamente scalata con un fattore *amount* e moltiplicata per I_0 . *Imsharpen* enfatizza i bordi dell'immagine mantenendo una buona qualità dell'immagine, ma introduce un rumore minimo nel calcolo della *mappa di dettaglio* e per questo ha una buona sinergia con l'algoritmo SWF. Per ottenere i risultati riportati in questa tesi è stato selezionato un valore di *amount* = 2.

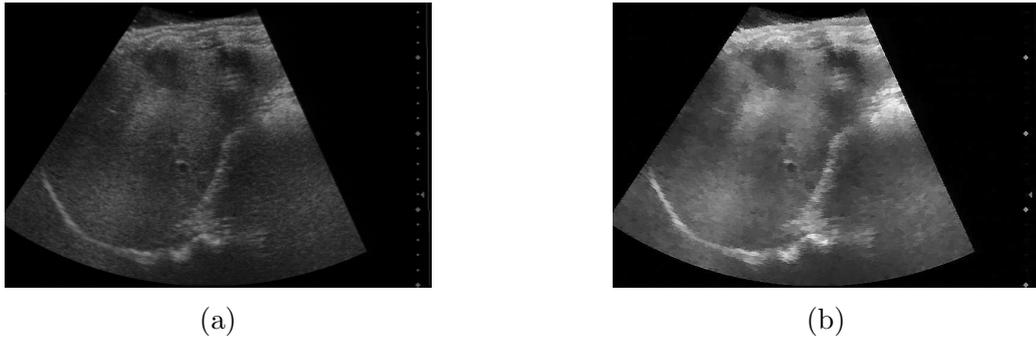


Figura 3.7: Nella figura si possono comparare gli effetti del *preprocessing* con SWF e *imsharpen* con *amount=2*(b), rispetto all'immagine originale (a) del paziente '2'.

Risultati

4.1 Metriche di valutazione

Per valutare l'accuratezza nell'identificazione della ROI e nella valutazione dei bordi durante la segmentazione, sono state utilizzate queste metriche:

- Intersezione sopra unione(IoU): misura l'estensione dell'area di sovrapposizione tra la maschera di segmentazione predetta e la *ground-truth*, rispetto all'intera area delle due maschere. L'IoU di un'immagine viene calcolato come:

$$IoU = \frac{tp}{tp + fp + fn} \quad (4.1)$$

- Coefficiente di Dice(DC): è una misura statistica che confronta l'*overlapping* tra la maschera di segmentazione predetta e la *ground-truth*. Il DC di un'immagine viene calcolato come

$$DC = \frac{2 \cdot tp}{2 \cdot tp + fp + fn} \quad (4.2)$$

- Sensibilità: è una misura quantitativa che indica la percentuale di pixel positivi correttamente classificati rispetto al numero totale di pixel positivi della *ground truth*. La specificità di un'immagine è calcolata come:

$$Sens = \frac{tp}{tp + fn} \quad (4.3)$$

- Specificità: è una misura quantitativa che indica la percentuale di pixel negativi correttamente classificati rispetto al numero totale di pixel negativi della *ground truth*. La specificità di un'immagine è calcolata come:

$$Spec = \frac{tn}{tn + fp} \quad (4.4)$$

- Distanza di Hausdorff(HD): è una metrica di valutazione dei bordi che quantifica la discrepanza tra i bordi della maschera di segmentazione prevista e la *ground truth*. L'HD su due insiemi di punti X e Y può essere calcolata come:

$$HD(X, Y) = \min_{x \in X} \max_{y \in Y} \|x - y\|_2 \quad (4.5)$$

4.2 Analisi delle performance

Al termine di ognuno dei *preprocessing* le immagini elaborate hanno subito una *data argumentation* comune. Essa è una tecnica utilizzata per creare nuovi esempi di dati eseguendo manipolazioni delle immagini, al fine di migliorare la capacità del modello di generalizzare e fare previsioni accurate. La *data argumentation* è fondamentale per prevenire l'*overfitting* del *dataset* utilizzato, composto da ecografie registrate dal medesimo strumento su solamente 8 pazienti. Ogni immagine ottenuta dal *preprocessing* è stata modificata con una trasformazione randomica tra le seguenti:

1. capovolgimento orizzontale, che crea una versione speculare dell'immagine originale;
2. spostamenti orizzontali o verticali, fino al 10%;
3. rotazione oraria o antioraria, fino a 20°;
4. variazione della luminosità, aumentata/ridotta di un intervallo compreso tra il 5 e il 50%.

Al fine di valutare l'impatto dei metodi di elaborazione proposti in precedenza, sono state addestrate quattro reti neurali. Una di queste funge da riferimento e utilizza le ecografie originali, mentre ciascuna delle rimanenti utilizza un *preprocessing* differente. Tutte le reti utilizzano lo stesso *dataset* descritto inizialmente e le medesime opzioni, selezionate per garantire un training efficace anche con una modesta GPU di 2 GB di un portatile. Tra queste è importante segnalare:

- Learning rate = 0.01;
- Numero epoche = 10;
- Dimensione mini batch = 4;
- Ottimizzatore = *stochastic gradient descent with momentum* (sgdm).

Nel confrontare i risultati delle 4 reti DeepLabv3Plus con backbone ResNet18, è interessante notare le differenze nelle prestazioni evidenziate dalla tabella 4.1 e le ragioni dietro tali discrepanze.

Tipo di preprocessing	IoU	DC	Sens	Spec	HD
Nessuno	0.8244	0.9032	0.9430	0.9715	7.2191
CLAHE	0.8226	0.9020	0.9172	0.9772	7.0915
Sobel+localcontrast	0.7610	0.8634	0.9186	0.9587	7.3831
SWF+imsharpen	0.7918	0.8830	0.9116	0.9699	6.7801

Tabella 4.1: Risultati ricavati dal testing dei 4 network addestrati.

La rete che presenta valori di IoU, DC e sensibilità maggiori è quella che utilizza immagini non processate. Questo è dovuto al fatto che le ecografie inalterate conservano le

caratteristiche originali senza alcuna manipolazione, consentendo al modello di estrarre informazioni senza interferenze o distorsioni. E' doveroso precisare che DeepLabv3Plus aumenta notevolmente le prestazioni delle immagini non processate, grazie alla sua architettura complessa e avanzata. Questa rete è caratterizzata da una grande capacità di adattamento e dalla tecnica chiamata *convoluzione dilatata*, che consente di espandere il raggio visivo dei singoli neuroni e cogliere relazioni su ampie porzioni d'immagine.

La seconda rete, che utilizza CLAHE, ha mostrato una specificità superiore rispetto agli altri. Questo valore indica che il modello è in grado di identificare meglio le regioni non interessanti, riducendo il numero di falsi positivi. Questo potrebbe essere dovuto all'aumento del contrasto locale, che migliora la capacità della rete di distinguere il fegato dallo sfondo.

La terza rete, che utilizza il *filtro di Sobel* e successivamente applica *localcontrast*, ha mostrato risultati complessivamente inferiori. La combinazione di questi metodi potrebbe non aver apportato miglioramenti rilevanti nella discriminazione delle regioni di interesse, tuttavia la notevole sensibilità lo rende efficace nel rilevare e identificare correttamente i casi positivi.

Infine la quarta rete, che utilizza SWF e *imsharpen*, presenta l'HD minore. Questo potrebbe essere dovuto all'algoritmo specifico di riduzione del rumore che facilita l'identificazione delle sfumature e dei confini delle regioni di interesse. L'applicazione successiva di *imsharpen*, che aumenta la nitidezza dei contorni, diminuisce l'HD ulteriormente. Il quarto network mostra quindi una migliore aderenza ai contorni di riferimento.

4.3 Conclusione

In questa tesi sono stati analizzati tre diversi metodi di elaborazione delle ecografie epatiche con un modello Deeplabv3Plus con backbone ResNet18. I risultati ottenuti evidenziano la rilevanza del *preprocessing* nell'ambito dell'intelligenza artificiale applicata al *medical imaging*. Ogni metodo applicato alle immagini ha dimostrato di evidenziare informazioni diverse, che possono essere sfruttate in modo vantaggioso a seconda dell'obiettivo dell'applicazione e delle caratteristiche del *dataset*. In conclusione l'uso di *preprocessing* mirati può ampliare le potenzialità delle reti neurali, permettendo loro di acquisire informazioni più specifiche e di aumentare le performance. La ricerca nel campo dei *preprocessing* delle immagini continua a essere un'area di interesse in crescita che offre opportunità di miglioramento e personalizzazione dei modelli di intelligenza artificiale, per affrontare sfide complesse e promuovere l'innovazione nel settore medico.

Bibliografia

- [1] M.Y. Ansari et al. “Dense-PSP-UNet: A neural network for fast inference liver ultrasound segmentation”. In: *Comput Biol Med* (2022). DOI: 10.1016/j.compbimed.2022.106478.
- [2] L. Petrusca et al. “Hybrid Ultrasound/Magnetic Resonance Simultaneous Acquisition and Image Fusion for Motion Monitoring in the Upper Abdomen”. In: *Invest Radiol* 48.5 (mag. 2013), pp. 333–340. DOI: 10.1097/RLI.0b013e31828236c3.
- [3] L. Chen et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *arXiv:1802.02611* (2018). DOI: 10.48550/arXiv.1802.02611. eprint: 1802.02611.
- [4] L. Chen et al. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. In: *arXiv:1606.00915* (2017). DOI: 10.48550/arXiv.1606.00915. eprint: 1606.00915.
- [5] H. Kervadec et al. “Boundary loss for highly unbalanced segmentation”. In: *Proceedings of Machine Learning Research* 102 (ago. 2019). A cura di M. Jorge Cardoso et al., pp. 285–296.
- [6] L. Nanni et al. “An Empirical Study on Ensemble of Segmentation Approaches”. In: *Signals* 3.2 (2022), pp. 341–358. DOI: 10.3390/signals3020022.
- [7] Ali M. Reza. “Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement.” In: *Journal of VLSI Signal Processing* (2004), pp. 35–44.
- [8] H. Yin, Y. Gong e G. Qiu. “Side Window Filtering”. In: *arXiv:1905.07177* (mag. 2019). eprint: 1905.07177.