POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Biomedica

Tesi di Laurea Magistrale

Simulazione di immagini ecografiche transorbitali come Data Augmentation per la segmentazione delle strutture del nervo ottico

Relatore
Prof.ssa KRISTEN MARIKO MEIBURGER
Co-relatore
Ing. FRANCESCO MARZOLA

Candidata
ALESSANDRA NARDELLA
Matricola 267377

A.A 2022/2023
Sessione di Laurea Dicembre 2022

Abstract

Negli ultimi anni l'utilizzo dell'ultrasonografia per lo studio delle strutture del nervo ottico (ON) è stato al centro di diversi studi volti ad indagare un possibile legame tra variazioni delle strutture del nervo ottico e patologie legate al Sistema Nervoso Centrale.
È stato dimostrato che per diametri della guaina del nervo ottico (ONSD) > 5 mm si ha un aumento della pressione intracranica (ICP) al di sopra dei 20 mmHg con specificità maggiore del 93% e sensitività maggiore dell'81%. Gli stessi studi hanno dimostrato anche una correlazione tra il diametro del nervo ottico (OND) e l'atrofizzazione del nervo stesso in soggetti affetti da malattie demielinizzanti come la Sclerosi Multipla.
La misura di ONSD e OND su immagini ultrasonografiche è una tecnica facilmente apprendibile e riproducibile con una bassa variabilità intra-operatore e inter-operatore. La misura di tali strutture potrebbe quindi essere utilizzata per rilevare alterazioni dell'ICP senza la necessità di dover utilizzare dispositivi invasivi.

L'obiettivo di questo lavoro di tesi è quello di migliorare i dati da fornire in ingresso ad una rete per la segmentazione di OND e ONSD attraverso l'utilizzo delle GAN (Generative Adversarial Network) come Data Augmentation.
Il dataset di partenza è formato da 464 immagini ultrasonografiche acquisite su 110 soggetti utilizzando 4 macchinari differenti. L'eterogeneità del dataset risulta un problema in quanto avendo una variabilità dei dati in ingresso troppo ampia l'algoritmo di deep learning non riesce a segmentare in maniera corretta.
Per effettuare Data Augmentation è stato utilizzato SPADE (Spatially Adaptative Normalization), una GAN che genera nuovi dati aventi la stessa distribuzione statistica dei dati reali a partire da una mappa semantica.

Sono state calcolate le metriche per la valutazione della qualità dell'immagine sia sul TrainSet, composto dall'80% delle immagini, che sul TestSet composto dal restante 20% delle immagini. È stato ottenuto un valore di PSNR di 21,4 dB ± 3,03 dB e 19,5 dB ± 2,52 dB rispettivamente per TrainSet e TestSet e un indice di somiglianza strutturale (SSIM) di 0,713 ± 0,096 per il TrainSet e 0,67 ± 0,092 per il TestSet.

Per costruire il dataset di training della rete di segmentazione, sono state selezionate 518 immagini che presentavano valori delle features di tessitura, calcolate nelle label di ON, ONS e bulbo oculare, compresi nella distribuzione delle immagini reali da aggiungere al dataset iniziale.
La rete di segmentazione allenata con le immagini simulate è stata testa sul dataset originale di 464 immagini ottenendo una Dice di 0,633 ± 0,161.

Possiamo quindi affermare che il dataset simulato è utile per allenare una rete di segmentazione e che quindi SPADE è in grado di produrre immagini utili per il Data Augmentation.

Indice

Elenco delle figure

Elenco delle tabelle

1. IL NERVO OTTICO

1.1 Il nervo ottico

Il ***nervo ottico*** è il secondo di 12 paia di nervi cranici ed è una continuazione del Sistema Nervoso Centrale (SNC) che origina dal diencefalo. Come struttura del SNC, il nervo ottico è mielinizzato dagli oligodendrociti ed è avvolto nei tre strati meningei (dura madre, aracnoide, pia madre). Tra la pia madre e l'aracnoide è presente una minima quantità di liquor e l'unione delle tre meningi forma quella che viene chiamata ***guaina del nervo ottico*** [1].
Il nervo ottico rappresenta l'inizio delle vie ottiche, ossia quell'insieme di strutture che, partendo dalla retina, collegano il bulbo oculare al cervello. Questo collegamento è fondamentale per la percezione visiva in quanto il compito del nervo ottico è quello di trasferire gli impulsi elettrici, generati in corrispondenza della retina, al cervello.

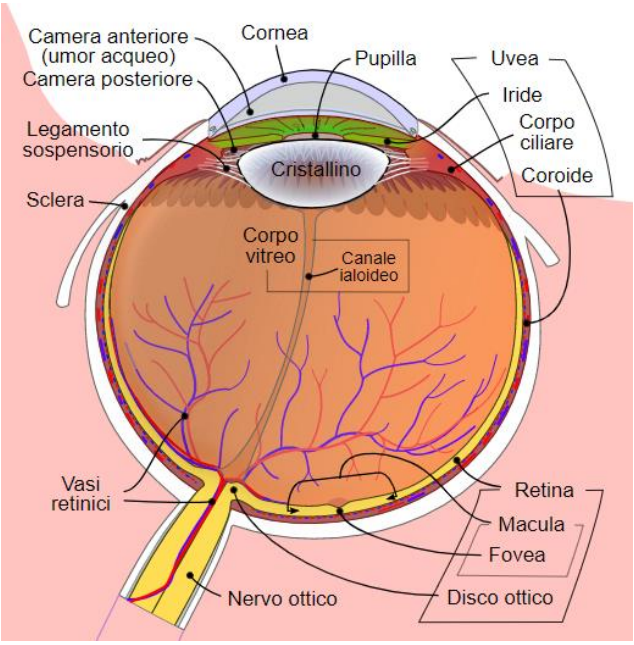

***Figura 1.1**: Anatomia dell'occhio e del nervo ottico* [2].

Il nervo ottico è lungo circa 5 cm, ha un diametro che varia tra i 3 ed i 7 mm ed è costituito da circa 1 milione di fibre nervose. Ogni singola fibra corrisponde a una piccola zona della retina, mentre ogni fascio corrisponde a un'intera area retinica. Le fibre, provenienti dalla retina, convergono tutte in una zona al polo posteriore del bulbo oculare e danno origine al disco ottico, una struttura con un diametro di 1,5 mm, che corrisponde quindi alla porzione iniziale del nervo ottico. Il nervo fuoriesce dall'occhio a livello del disco ottico e subisce un incrocio

delle fibre nervose a livello del chiasma ottico che è la zona inferiore del cervello al di sotto dell'ipotalamo (Figura 1.1).

Il nervo ottico può essere distinto in 4 porzioni:

- Porzione *intrabulbare:* inizia nel bulbo oculare a livello del disco ottico, attraversa la coroide ed il diaframma cribroso del canale sclerale per poi fuoriuscire dall'occhio (1 mm);
- Porzione *intraorbitaria*: continua dal polo posteriore dell'occhio fino al canale ottico ed è la porzione più lunga del nervo (circa 25 mm);
- Porzione *intracanalicolare*: corrisponde al passaggio attraverso il canale ottico (4-10 mm);
- Porzione *intracranica*: è compresa tra il canale ottico e il chiasma ottico (10 mm) [3].

Il nervo ottico, presentando caratteristiche comuni con la sostanza bianca encefalica, risulta particolarmente vulnerabile a malattie demielinizzanti, come la Sclerosi Multipla, ed encefaliti; inoltre, la presenza del liquor, spiega la sua suscettibilità ad essere coinvolto nel corso di meningiti.

1.2 OND e ONDS

Di particolare interesse per lo studio patologico del nervo ottico sono il ***diametro del nervo ottico*** (OND) e il ***diametro della guaina del nervo ottico*** (ONSD). Molti studi hanno dimostrato un'associazione tra la pressione intracranica (ICP) e l'ONSD. Infatti, lo spazio subaracnoideo intra-orbitale che circonda il nervo ottico risponde alle stesse variazioni di pressione del compartimento intracranico e ciò rende la guaina del nervo ottico sufficientemente elastica da consentire una dilatazione rilevabile in risposta all'aumento della pressione intracranica. Gli stessi studi dimostrano anche una correlazione tra l'OND e l'atrofizzazione del nervo ottico in soggetti affetti da malattie demielinizzanti come la Sclerosi Multipla [4].
Nello studio portato avanti da Kimberly et. al [5] su pazienti adulti è stato dimostrato che valori di ONSD > 5 mm sono correlati ad un'elevata pressione intracranica (ICP > 20 cm H_2O) con una sensitività dell'88% e una specificità del 93%; mentre valori di ONSD > 4,5 mm permettono di rilevare un aumento dell'ICP con una sensitività del 100%, ma a discapito di una specificità ridotta (63%). Conclusioni simili sono state tratte da Maissan et al. [6] il cui studio su pazienti affetti da trauma cranico ha dimostrato un aumento dell'ICP al di sopra di 20 mmHg associato ad una dilatazione dell'ONSD > 5 mm con una sensitività del 94% e una specificità del 98%.
Anche per pazienti affetti da problemi cerebrovascolari, come riportato nello studio di Yüzbaşıoğlu et al. [7], è stata trovata una correlazione tra aumento di ONSD e ICP con una specificità del 98.1% e una sensitività dell'81,8% per l'ONSD > 5 mm.

Tutti gli studi hanno dimostrato una correlazione tra variazione dell'ONSD e variazione dell'ICP e hanno identificato 5 mm come misura dell'ONSD al di sopra della quale i pazienti mostrano un aumento dell'ICP. Da questi studi si può quindi intuire che la misura dell'ONSD su immagini ultrasonografiche è un metodo facile, economico e non invasivo che può essere utilizzato per rilevare un possibile aumento della pressione intracranica senza la necessità di dover utilizzare dispositivi invasivi come la sonda intracranica che rappresenta il gold standard per la misurazione dell'ICP.

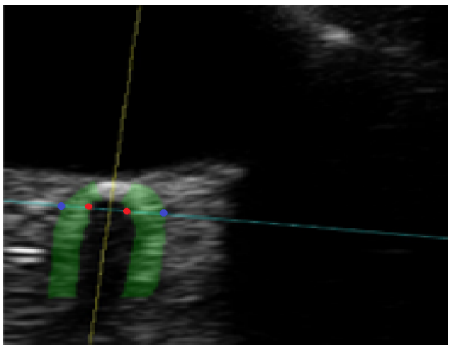

***Figura 1.2**: Ultrasonografia transorbitale del nervo ottico. OND è la distanza tra i due punti rossi, ONSD è la distanza tra i due punti blu.*

La misurazione di ONSD e OND è una tecnica facilmente apprendibile e riproducibile con una bassa variabilità intra-operatore e inter-operatore e la misurazione si effettua, su immagini ottenute tramite ultrasonografia transorbitale, 3 mm posteriormente alla superficie sclerale posteriore del globo. Il diametro del nervo ottico viene misurato in modo che corrisponda alla distanza interna alla pia madre, mentre il diametro della guaina viene misurato tra i bordi esterni iperecogeni dello spazio subaracnoideo, coincidendo quindi con la distanza interna alla dura madre [8].

2. ULTRASONOGRAFIA

2.1 Ultrasonografia

L'***ultrasonografia***, nota anche come ecografia, è una metodica diagnostica indolore e non invasiva che, utilizzando gli ***ultrasuoni*** (US) emessi da particolari sonde appoggiate sulla pelle del paziente, consente di visualizzare gli organi interni e i tessuti molli. Questi suoni vengono generati grazie a dei cristalli che esposti ad una corrente elettrica vibrano con una frequenza tale da creare gli ultrasuoni.
L'esame ecografico non comporta l'impiego di raggi X e, a differenza di altre metodiche come TAC e risonanza magnetica, non presenta alcun effetto collaterale ed è ripetibile più volte senza rischi per la salute.

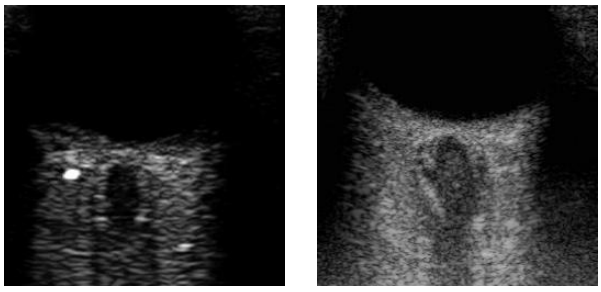

Figura 2.1*:Esempi di immagine US del nervo ottico ottenute da dispositivi diversi.*

Gli ultrasuoni sono onde sonore superiori ai 20KHz non udibili dall'orecchio umano. Queste onde viaggiano all'interno del corpo, fino a quando non colpiscono l'interfaccia tra due tessuti. A seconda del tipo di tessuto presente, le onde sonore possono essere riflesse all'indietro o continuare a viaggiare ulteriormente. Le onde che vengono riflesse all'indietro, chiamate echi, vengono ritrasmesse al dispositivo di imaging a ultrasuoni. In base al tempo di ritorno di ogni eco e alla velocità del suono nel tessuto, il dispositivo calcola la distanza tra la sonda e ciascuna struttura. La distanza e l'intensità di tutti gli echi si trasformano in un'immagine bidimensionale che appare sullo schermo di imaging ad ultrasuoni.

L'immagine finale ha un aspetto che dipende dal tipo di tessuto in analisi che può essere:

- Anecogeno: il tessuto appare nero per l'assenza di echi;
- Iperecogeno: il tessuto appare bianco per presenza di echi ad elevata intensità;
- Ipoecogeno: il tessuto appare grigio chiaro per la presenza di echi di intensità moderata;
- Ecogeno: il tessuto appare grigio chiaro per la presenza di echi di intensità intermedia.

2.1.1 Principi fisici

Gli ultrasuoni sono generati per effetto piezoelettrico. La piezoelettricità è la proprietà di alcuni materiali cristallini di polarizzarsi generando una differenza di potenziale elettrico quando sono soggetti a una deformazione meccanica (***effetto piezoelettrico diretto***) e al tempo stesso di deformarsi in maniera elastica quando sono sottoposti ad una tensione elettrica (***effetto piezoelettrico inverso***). Questo fenomeno si manifesta solo lungo una determinata direzione (anisotropia). Nei dispositivi ad US viene utilizzato l'effetto piezoelettrico diretto per misurare gli echi di ritorno e l'effetto piezoelettrico inverso per generare gli impulsi.
Gli US diffondono nei materiali, compresi i tessuti biologici con una velocità di propagazione che dipende dalla frequenza di emissione tramite la relazione:

$$v = \lambda * f \qquad (2.1)$$

dove v è la velocità di propagazione, λ è la lunghezza d'onda ed f la frequenza.
Dalla 2.1 si può facilmente intuire che maggiore sarà la frequenza utilizzata, minore sarà la lunghezza d'onda e, quindi, migliore sarà la risoluzione spaziale dell'immagine.
La velocità di propagazione dell'onda dipende anche dalla densità del mezzo attraversato tramite una grandezza nota come *impedenza acustica* definita come:

$$Z = p * v \qquad (2.2)$$

dove v è la velocità di propagazione del mezzo e ρ la densità del mezzo stesso.
Propagandosi in un mezzo l'onda US in parte viene riflessa e in parte continua a propagarsi attraverso i tessuti.
La riflessione può essere descritta dalla seguente formula:

$$R = \left(\frac{Z_1 - Z_2}{Z_1 + Z_2}\right)^2 \qquad (2.3)$$

dove R è il coefficiente di riflessione e Z_1 e Z_2 sono i valori di impedenza acustica dei tessuti attraversati dall'onda.
L'onda che non viene riflessa, durante la propagazione può subire attenuazione del fascio; questo comporta una diminuzione dell'ampiezza dell'onda secondo la seguente formula:

$$A = A_0 e^{-\alpha x} \qquad (2.4)$$

$$\alpha = a f^b \qquad (2.5)$$

Dove A è l'ampiezza dopo che l'onda ha percorso un tratto$x$, A_0è l'ampiezza iniziale e α il *coefficiente di assorbimento*. Quest'ultimo dipende dalla frequenza della radiazione e dal materiale considerato attraverso i coefficienti a e b.
Dalla 2.4 e dalla 2.5 si evince quindi che, aumentando la frequenza US diminuisce la sua profondità di penetrazione. È importante quindi scegliere una frequenza che permetta di avere una risoluzione sufficiente per osservare i tessuti in esame e al contempo permetta all'onda di raggiungerli.

2.1.2 Visualizzazione dell'immagine

Esistono diverse strategie di visualizzazione dell'immagine ultrasonografica:

- A-Mode (Amplitude): è la strategia di visualizzazione più semplice. Si utilizza un singolo trasduttore che scansiona una linea attraverso il corpo e gli echi vengono tracciati sullo schermo in funzione della profondità.
- B-Mode (Brightness): è la strategia di visualizzazione più utilizzata. Si utilizza un array lineare di trasduttori che scansiona simultaneamente un piano attraverso il corpo che può essere visualizzato come un'immagine bidimensionale sullo schermo.
- M-Mode (Time Motion): si utilizza una sola linea di scansione e ci valuta come variano le discontinuità nel tempo [9].

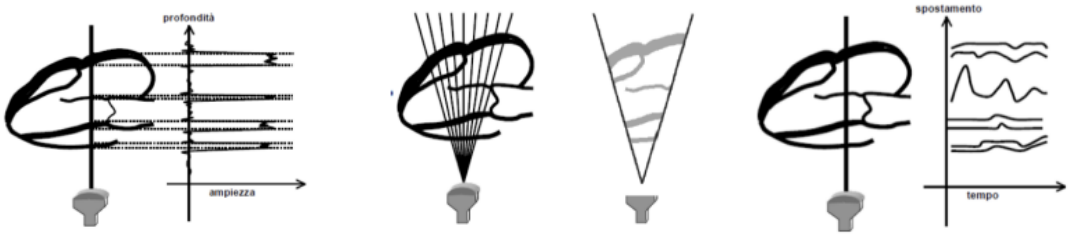

***Figura 2.2**: Strategie di visualizzazione. Da sinistra a destra: A-Mode, B-Mode, M-Mode* [10].

2.1.3 Tipologie di sonde

La sonda (o trasduttore) è lo strumento che emana onde sonore per trasmettere e ricevere il segnale. I trasduttori possono essere di vario tipo, a seconda dell'uso per cui devono essere predisposti.

- Sonda lineare: i cristalli sono disposti in maniera lineare, ha lunghezza fra 2,5 e 10 cm, frequenza elevata (7-15 MHZ) e area di forma rettangolare.

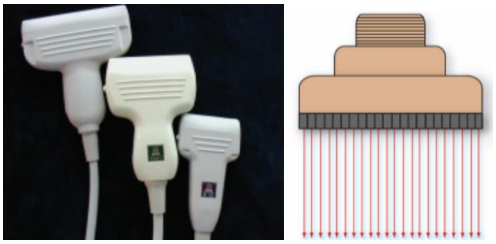

***Figura 2.3**: Sonda lineare* [11].

- Sonda convex: i cristalli sono disposti su un arco di circonferenza generando un'area di scansione a tronco di cono; le dimensioni variano da 20 a 120 mm in funzione delle applicazioni diagnostiche. Lavora con frequenze minori (2-7 MHz).

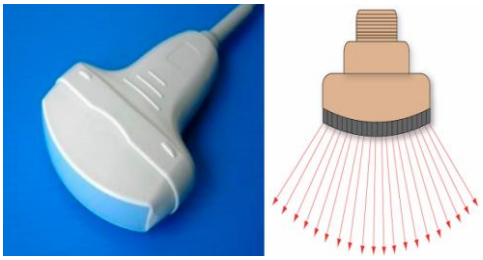

***Figura 2.4**: Sonda convex* [11].

- Sonda phased array: comunemente più piccola delle sonde lineari e convex, è costituita da microcristalli multipli affiancati che vengono attivati con piccolissimi ritardi l'uno dall'altro generando un fascio che può essere inclinato in varie direzioni. L'area di scansione è di tipo conico.

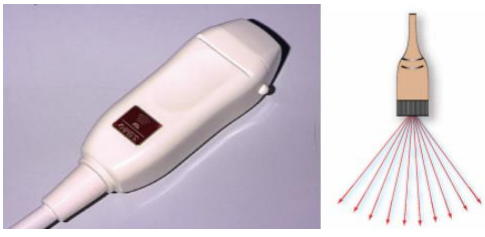

***Figura 2.5**: Sonda phased array* [11].

2.2 Ultrasonografia Transorbitale

L'***ultrasonografia transorbitale*** (TOS) è uno strumento diagnostico non invasivo e di facile utilizzo che consente di quantificare l'OND e l'ONSD. Le principali aree di applicazione sono le patologie con aumento della pressione intracranica e recentemente è stato dimostrato che anche l'atrofia del nervo ottico può essere rilevata in modo affidabile mediante la TOS in pazienti con malattie demielinizzanti croniche del sistema nervoso centrale [8].
L'ultrasonografia transorbitale viene acquisita in modalità B-Mode utilizzando una sonda lineare ad alta frequenza (> 7.5 MHz) che consente una risoluzione spaziale inferiore a 0,4 mm [12].
L'esame viene eseguito su soggetti in posizione supina, con la parte superiore del corpo e la testa sollevati di un angolo di 20 – 30° rispetto all'orizzontale; viene chiesto di mantenere l'occhio in una posizione intermedia e neutra con lo scopo di limitare il più possibile qualunque movimento oculare. La sonda viene, poi, appoggiata sulla zona temporale della palpebra superiore, tenuta chiusa, usando un sottile strato di gel ultrasonico. In questo modo verrà acquisita un'immagine raffigurante la parte anteriore del nervo ottico nel suo decorso longitudinale [13].

Figura 2.6*: Esempio di come viene effettuata un'ultrasonografia transorbitale* [13].

L'ultrasonografia transorbitale è ampiamente utilizzata in medicina. Di conseguenza, è necessario disporre di strumenti di segmentazione efficienti per favorire la diagnosi assistita da computer, gli interventi guidati dalle immagini e la terapia [14].
In questo lavoro di tesi, per la segmentazione del nervo ottico e delle sue guaine, verrà utilizzato un algoritmo completamente automatico basato sull'utilizzo delle reti neurali.

3. RETI NEURALI

3.1 Reti neurali artificiali (ANN)

Le ***reti neurali artificiali*** sono modelli matematici che mimano e modellizzano il comportamento dei neuroni. Sono un sottoinsieme del machine learning e sono l'elemento centrale degli algoritmi di deep learning. Il Deep learning, apprendimento profondo, è l'apprendimento da parte delle macchine attraverso dati appresi grazie all'utilizzo di algoritmi. L'architettura delle reti neurali (Figura 3.1) è caratterizzata da neuroni distribuiti in diversi strati, *layers*, che scambiano informazioni tra di loro attraverso un complesso sistema di connessioni e nodi.
Tutte le reti neurali sono composte da almeno tre strati:

- Un ***input layer***: riceve i dati in ingresso e li comunica al layer successivo;
- Uno o più ***layer nascosti*** (hidden layers): elaborano i dati per ottenere l'output desisderato;
- Un ***output layer***: fornisce in uscita i risultati del modello.

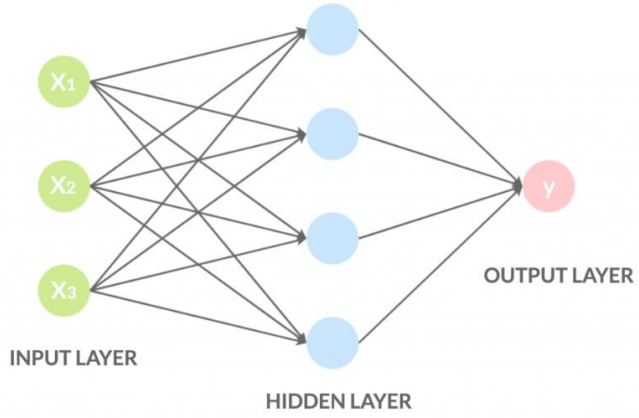

Figura 3.1*: Struttura di una rete neurale. I cerchi rappresentano i neuroni, le frecce le interconnessioni tra i layer* [15].

I neuroni di ogni layer sono collegati con quelli dei layersuccessivi tramite opportune funzioni di attivazione pesate. Il peso è un valore numerico che viene moltiplicato per quello del neurone successivo per stabilire l'importanza di una qualsiasi data variabile. Ogni neurone somma i valori pesati di tutti i neuroni connessi ad esso ed eventualmente aggiunge un valore di bias. Prima di passare i dati allo strato successivo, si trasforma il valore ottenuto dalla somma applicando una funzione di attivazione. Se il valore di questa operazione supera la

soglia prevista, il neurone si attiva e i dati passano allo strato successivo nella rete. L'output di un nodo diventa quindi l'input del nodo successivo. Questo processo di passaggio dei dati da un layer a quello successivo definisce questa rete neurale come una rete *feedforward.*
Dopo che la rete ha passato i suoi input fino ai suoi output, valuta la bontà della sua previsione, rispetto all'output atteso, attraverso una funzione di perdita (loss function). L'obiettivo della rete è quello di minimizzare la funzione di loss aggiustando i pesi e i biases della rete [16]:

$$J(W,b) = \frac{1}{n}\sum_{i=1}^{n} L(\hat{y}_i - y_i) \qquad (3.1)$$

dove L è la loss function, $\hat{y}_i$ è l'output in uscita dalla rete e y_i l'output atteso.

Affinché le connessioni nelle reti neurali siano stabilite correttamente per risolvere il problema, le reti devono essere addestrate. Si possono distinguere due tipi di apprendimento:

- ***Apprendimento supervisionato***: utilizza dati etichettati per addestrare il modello. Vengono forniti gli input ed i relativi output desiderati, con lo scopo di apprendere una regola generale in grado di mappare gli input negli output.
- ***Apprendimento non supervisionato***: vengono forniti solo dei dati in input senza alcun output atteso.

Caratteristica fondamentale delle reti neurali sono le ***funzioni di attivazione***. Si tratta di funzioni non lineari, il che permette di approssimare i dati in modo molto più preciso. Le principali funzioni di attivazione sono:

- Funzione ***Sigmoide***: prende un numero in input e restituisce in output un numero compreso tra 0 e 1. Utilizzata principalmente nei problemi di classificazione binaria.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \qquad (3.2)$$

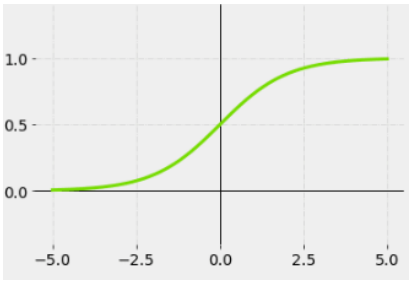

***Figura 3.2**: Andamento della funzione Sigmoide* [17].

- ***TanH*** (Tangente Iperbolica): restituisce un output compreso nell'intervallo [-1,1] e a differenza della Sigmoide l'output è centrato sullo zero.

$$tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \qquad (3.3)$$

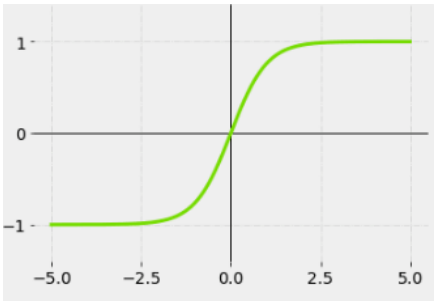

***Figura 3.3**: Andamento funzione Tangente Iperbolica* [17].

- ***ReLu*** (Unità Lineare Rettificata): è una delle funzioni di attivazione più comunemente utilizzata. Restituisce un output compreso tra 0 e infinito. Quando l'input è positivo restituisce in output l'input stesso, se invece l'input è negativo restituisce 0 in output.

$$f(x) = \max(0, x) \qquad (3.4)$$

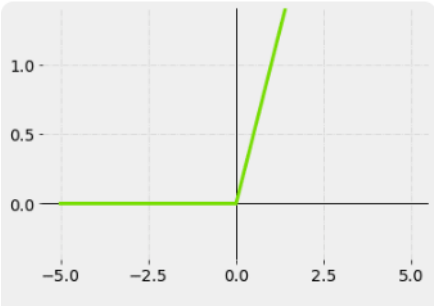

***Figura 3.4**: Andamento funzione ReLu* [17].

- ***Softmax***: è la generalizzazione della funzione Sigmoide. È usata negli algoritmi di classificazione multi-classe per calcolare le probabilità relative. In output restituisce la probabilità di un dato input di appartenere ad una determinata classe.

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \qquad (3.5)$$

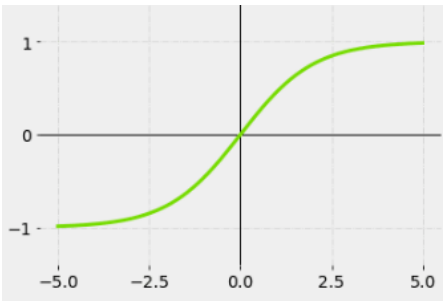

***Figura 3.5**: Andamento funzione Softmax* [17].

3.2 Reti Neurali Convoluzionali (CNN)

Le ***Reti Neurali Convoluzionali*** (CNN) sono una classe di reti neurali artificiali comunemente utilizzate per analizzare le immagini visive. Si distinguono dalle altre reti neurali per le loro prestazioni superiori con input di segnali di immagini, voce o audio.
Le CNN sono caratterizzate da neuroni a 3 dimensioni: altezza, larghezza e profondità e a differenza delle reti neurali artificiali, i neuroni di un layer sono collegati solo a una piccola regione del layer precedente.
Le CNN sono composte da 3 tipi di layers: ***convolutional layers***, ***pooling layers*** e ***fully-connected layers*** [18], [19].

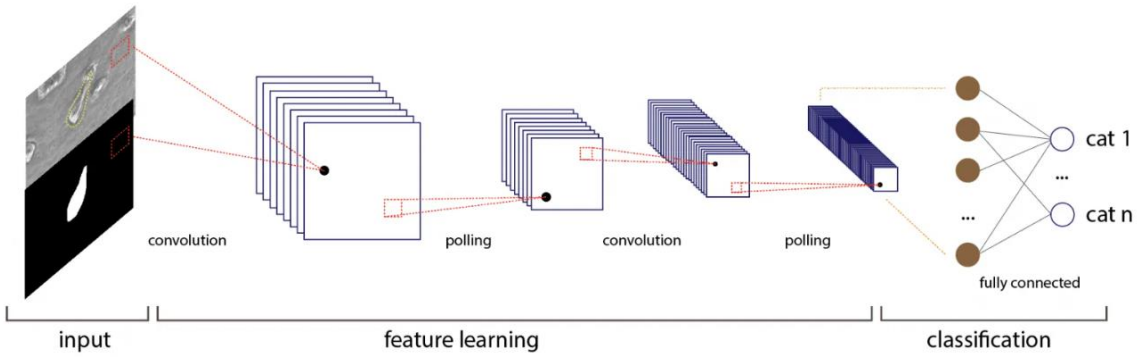

***Figura 3.6**: Architettura tipica di una CNN* [20].

- ***Convolutional Layer***: questo layer gioca un ruolo fondamentale nel modo di operare delle CNN; maggiore è il numero di layer convoluzionali, maggiore è la complessità delle caratteristiche individuabili. I parametri di questo layer consistono in una serie di filtri digitali o *kernel*. Quando i dati raggiungono lo strato di convoluzione, il kernel viene fatto scorrere lungo le dimensioni spaziali dell'input e si esegue la convoluzione ovvero il prodotto scalare tra i valori del filtro e quelli del campo recettivo (porzione

dell'input su cui è applicato il kernel), per produrre in output una mappa di attivazione 2D (Figura 3.6).
La mappa di attivazione avrà dimensione (M-f+1) x (M-f+1), dove M è la dimensione del campo recettivo ed f la dimensione del filtro.
Ogni kernel ha una mappa di attivazione; l'unione di tutte le mappe di attivazione determina la terza dimensione, la profondità.

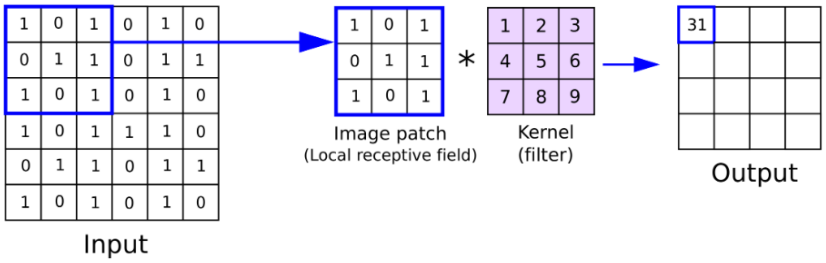

***Figura 3.7**: Esempio di un'operazione di convoluzione* [21].

La dimensione dell'output è influenzata da tre iperparametri che devono essere impostati prima dell'addestramento della rete neurale:

- Numero di filtri: determina la profondità dell'output;
- Stride: indica quanto il kernel si sposta sulla matrice;
- Zero-padding: alla matrice di partenza si aggiunge un bordo di zeri quando i filtri non si adattano all'immagine di input [22].

- ***Pooling Layer***: questo layer riduce la dimensionalità, riducendo il numero di parametri in input. Simile al layer di convoluzione, l'operazione di pooling prevede lo scorrimento di un filtro sull'intero input. Esistono due tipi principali di pooling:
  - Max pooling: il filtro muovendosi lungo l'input seleziona il pixel con valore massimo e lo restituisce in output (è l'approccio più utilizzato);
  - Averege pooling: il filtro muovendosi lungo l'input calcola il valore medio all'interno del campo recettivo e lo restituisce in output.

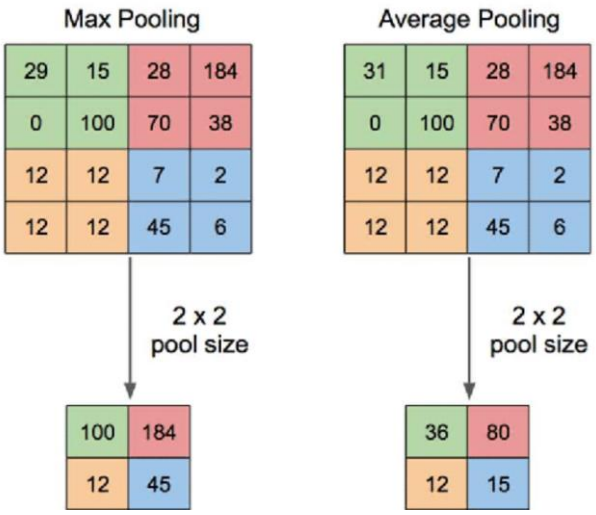

***Figura 3.8**: Esempio di un'operazione di pooling. A sinistra si può osservare un'operazione di Max pooling, a destra un'operazione di Average pooling* [21].

- ***Fully-connected Layer***: questo layer esegue il compito di classificazione in base alle caratteristiche estratte attraverso i livelli precedenti. In questo livello ogni nodo del layer di output è connesso al layer precedente.
  Mentre i layers convoluzionali e di pooling utilizzano una ReLu come funzione di attivazione, i fully-connected layers sfruttano una funzione di attivazione *Softmax* per classificare gli input in modo appropriato, producendo una probabilità compresa tra 0 e 1.

3.2.1 Evoluzione delle CNN

Negli utlimi 10 anni sono state presentate diverse architetture delle CNN ottenute in seguito a modifiche strutturali, regolarizzazione e modifica dei parametri. L'architettura rappresenta un fattore critico per migliorare le prestazioni delle diverse applicazioni [23], [24].
Di seguito verranno discusse le principali architetture sviluppate negli ultimi anni.

AlexNet, sviluppata da Krizhevsky et al. [25] nel 2012, è considerata la prima CNN profonda ad aver mostrato risultati rivoluzionari per la classificazione e il riconoscimento di immagini.
L'AlexNet è costituita da 8 layer: i primi 5 sono layers convoluzionali, alcuni dei quali sono seguiti da layers di Max-pooling; gli utlimi 3 sono fully-connected layers. Prevede l'utilizzo di circa 60 milioni di parametri. Prevede l'utilizzo della ReLu come funzione di attivazione in quanto mostra prestazioni migliori di allenamento rispetto alla Tangente Iperbolica che al tempo rappresentava lo standard e inoltre consente l'allenamento multi-GPU inserendo metà dei neuroni del modello su una GPU e l'altra metà su un'altra GPU; questo significa non solo

che possono essere addestrati modelli più grandi ma che si riducono anche i tempi di addestramento.

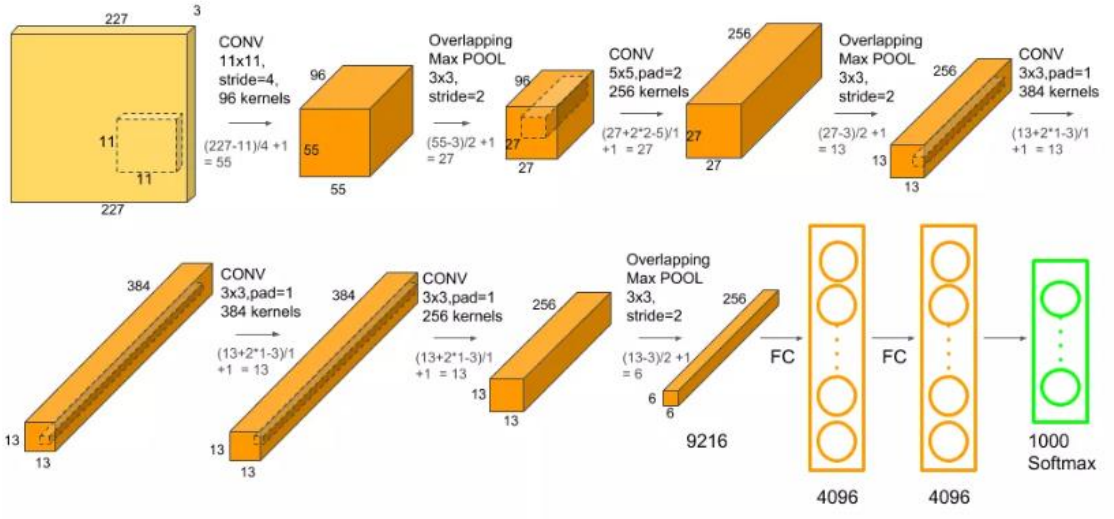

***Figura 3.9**: Architettura dell'AlexNet* [25].

Nel 2014 Simonyan e Zisserman [26] hanno proposto una nuova architettura chiamata ***Visual Geometry Group (VGG)***. Questa rete presenta 19 strati rispetto all'AlexNet per simulare la relazione tra la profondità e la capacità rappresentativa della rete.
In VGG sono stati sostituiti i filtri 11x11 e 5x5 presenti nell'AlexNet con una serie di filtri di dimensioni 3x3. Il limite di questa rete è però rappresentato da un costo computazionale elevato a causa del numero di parametri utilizzati (intorno a 140 milioni).
Lo studio su questa rete ha permesso di dimostrare che è possibile raggiungere prestazioni migliori aumentando il numero di layers. Nonostante questo, non sono state sviluppate VGG più profonde perché superato un certo livello di profondità il modello inizia a presentare il problema della scomparsa del gradiente (vanishing gradient). Questo succede perché quando la rete è troppo profonda, i gradienti da cui viene calcolata la loss function si riducono facilmente a zero dopo diverse applicazioni della regola della catena. Questo comporta il fatto che i pesi non vengono aggiornati e quindi di conseguenza non c'è apprendimento.

He et al. [27] nel 2015 hanno sviluppato la ***ResNet*** (Residual Network) con lo scopo di avere una rete profonda senza il problema del vanishing gradient. La novità della ResNet è l'introduzione di blocchi noti come *skip connection* o *blocchi residuali* che creano un percorso alternativo per il passaggio del gradiente durante la fase di retropropagazione dell'errore. In Figura 3.10 è mostrato un esempio di skip connection.

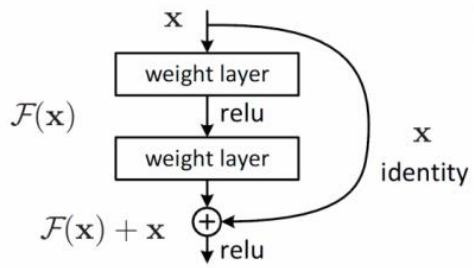

***Figura 3.10**: Skip connection* [27].

Con X viene indicato l'input, con $F(X)$ le trasformazioni non lineari F effettuate sull'input X dai layers intermedi. Nelle skip connection l'input è sommato a $F(X)$producendo l'output:

$$H(X) = F(X) + X \qquad (3.5)$$

Successivamente su $H(X)$ viene applicta una funzione di attivazione ReLu. Un blocco di questo tipo permette di apprendere nuove informazioni sull'immagine senza perdere però quelle dell'input X, ricavate dagli strati precedenti [27].
Sono state sviluppate diverse ResNet in base al numero di layers. La più comune è la ResNet50 costituita da 5 convolutional layers; il primo prevede un solo layer convoluzionale seguito da un layer di Max-pooling, mentre i successivi 4 blocchi convoluzionali sono costituiti da più blocchi residuali impilati raggiungendo rispettivamente un totale di 9, 12, 18, 9 layer. L'ultimo convolutional layer è seguito da un blocco di average-pooling e da un fully-connected layer.

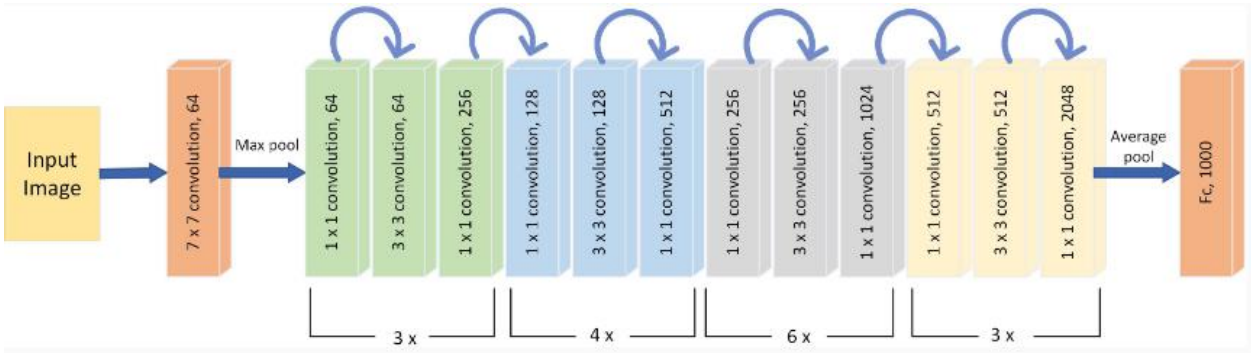

***Figura 3.11**: Architettura ResNet50* [28].

Nel 2015 è stata sviluppata anche la ***U-Net*** da O. Ronneberger et al. [29], una rete convoluzionale pensata per la segmentazione semantica di immagini mediche. Il processo di *segmentazione semantica* prevede che ad ogni pixel dell'immagine venga attribuita un'etichetta o una categoria in modo tale da dividere l'immagine in gruppi di pixel appartenenti alla stessa classe.

La U-Net è una *Fully Convolutional Networks (FCN)* la cui architettura consiste in una parte di ***encoding*** che effettua down-sampling e una parte di ***decoding*** che effettua up-sampling; questa struttura è più o meno simmetrica e crea una forma ad U (da cui prende il nome la rete).

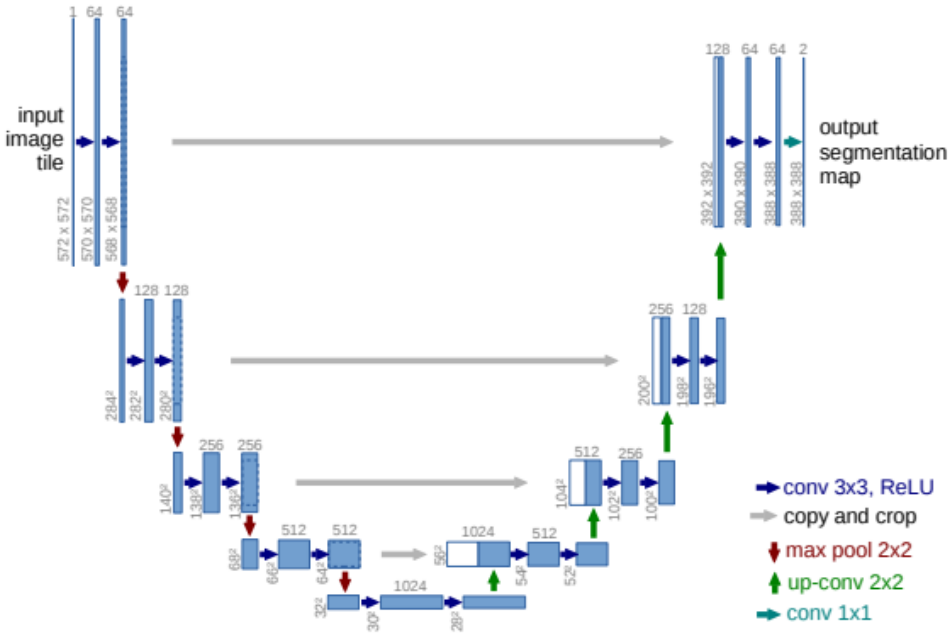

***Figura 3.12**: Architettura U-Net* [29].

La parte di encoding segue la tipica architettura di una CNN; consiste nell'applicazione ripetuta di due convoluzioni 3x3, ciascuna seguita da un layer di attivazione ReLu e da un layer di Max-pooling 2x2 con stride pari a 2. Ad ogni step di down-sampling i canali delle mappe di attivazione vengono raddoppiati. Tramite questo processo vengono ridotte le dimensioni spaziali, ma aumentano le dimensioni delle caratteristiche.
La parte di decoding consiste in un up-sampling della feature map, seguito da una convoluzione 2x2 che dimezza il numero di canali delle mappe di attivazione. A questa operazione seguono la concatenazione con la feature map corrispondente appartenete alla parte di encoding e due convoluzioni 3x3 ciascuna seguita da una ReLu.

3.3 Generative Adversarial Networks (GAN)

Le GAN (Generative Adversarial Networks), o Reti Generative Avversarie, sono un sistema di apprendimento automatico non supervisionato descritto per la prima volta nel 2014 da Goodfellow et al. [30] il cui scopo è quello di generare immagini che siano statisticamente indistinguibili da quelle reali. La generazione di nuove immagini risulta utile per il data augmentation: l'ampliamento di dati a disposizione per l'apprendimento automatico senza raccogliere nuovi elementi ma modificando sinteticamente quelli già presenti. Il data augmentation serve per ridurre l'overfitting, ovvero l'adattamento del modello di apprendimento automatico al campione statistico di dati utilizzato durante il training. La GAN serve quindi a ottimizzare il deep learning ed evitare errori superficiali di generalizzazione dovuti alla scarsità dei dati.
Una GAN è composta da due reti neurali convoluzionali che vengono addestrate in maniera competitiva: un ***generatore*** (G) e un ***discriminatore*** (D). Lo scopo del generatore è quello di apprendere e catturare il più possibile la distribuzione dei campioni reali dati in ingresso e generare nuove immagini. Per ogni iterazione dell'algoritmo, il generatore non vede mai l'input originale, ma vede la variabile casuale latente, ovvero il rumore basato sulle immagini reali in ingresso. Il discriminatore, invece, è un classificatore binario il cui scopo è riconoscere le immagini false e distinguerle da quelle reali (Figura 3.13).

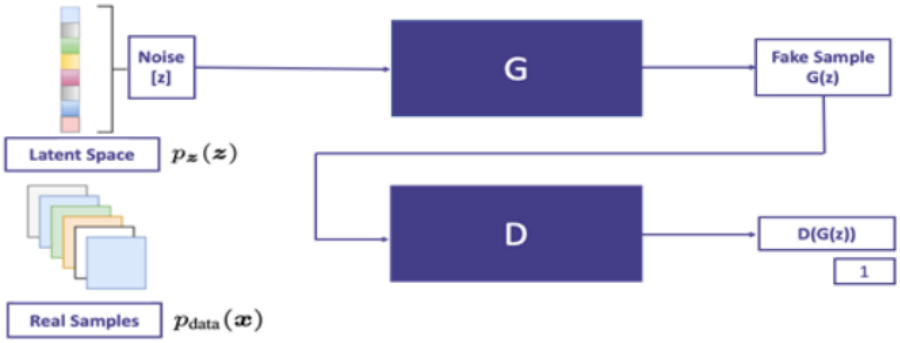

***Figura 3.13**: Architettura di una GAN* [31].

Il discriminatore prende in input l'immagine prodotta dal generatore e fornisce in output un valore che indica quanto l'immagine generata sia vicina all'immagine reale. In particolare, il discriminatore restituisce una probabilità tra 0 e 1 che l'immagine analizzata sia falsa o vera. L'obiettivo è massimizzare la probabilità che il discriminatore riconosca le immagini reali come vere e le immagini generate come false. Prima di addestrare la GAN, il discriminatore viene allenato separatamente a riconoscere i dati reali con un grado di precisione soddisfacente.
Affinché ci sia apprendimento è importante che l'addestramento del generatore e del discriminatore vada di pari passo e che le due reti neurali competano tra di loro a livelli

sempre più elevati in modo tale da migliorare e aumentare la propria efficienza. Le due reti vengono addestrate in maniera alternata tramite retropropagazione dell'errore, mantenendo invariati i parametri del modello generativo durante l'addestramento del discriminatore e, viceversa, mantenendo invariati i parametri della rete discriminativa durante l'addestramento del generatore. L'apprendimento avviene attraverso feedback che vengono forniti sia al generatore che al discriminatore.
I risultati della classificazione del discriminatore rappresentano sia il riferimento per G per ottimizzare le distribuzioni e generare immagini più vicine a quelle reali per ingannare D sia il feedback da fornire a D affinché possa apprendere dall'errore e migliorare le risposte successive.
L'apprendimento consiste nell'ottimizzare un gioco ***minmax*** a due giocatori in cui il generatore vuole minimizzare la funzione obiettivo mentre il discriminatore vuole massimizzare la stessa funzione. La funzione obiettivo che si vuole ottimizzare è la seguente:

$$min_G max_D\, V(D,G) = E_{x\sim pdata(x)}[\log D(x)] \; + \; E_{z\sim pz(z)}[\log\,(1 - \; D(G(z)))] \quad (3.6)$$

P_z è la distribuzione di probabilità dello spazio latente che di solito è una distribuzione gaussiana casuale, P_{data} è la distribuzione di probabilità del set di dati reali, $D(x)$ è la stima del discriminatore relativa alla probabilità che l'istanza di dati reali x sia reale, $G(z)$ è l'output del generatore e $D(G(z))$ è la stima del discriminatore relativa alla probabilità che un'istanza falsa sia reale [32].
La funzione di loss può essere suddivisa in due parti:

- Discriminator loss: durante l'addestramento, il discriminatore classifica entrambi i dati reali e i dati simulati dal generatore; si penalizza per la misclassificazzione di un'istanza reale come falsa o di un'istanza falsa come vera massimizzando la seguente funzione:

$$L^D = \max\,[\log\big(D(x)\big) + \log\Big(1 - D\big(G(z)\big)\Big)] \quad (3.7)$$

  Il Discriminatore vuole portare la probabilità di $D(G(z))$ a 0, quindi vuole massimizzare $1 - D(G(z))$.

- Generator loss: durante l'addestramento, il generatore campiona il rumore casuale e produce un output; l'output passa attraverso il discriminatore e viene classificato come reale o falso in base alla capacità del discriminatore di distinguere l'uno dall'altro. La funzione di perdita del generatore viene quindi calcolata dalla classificazione del discriminatore; viene premiato se inganna con successo il discriminatore, mentre viene penalizzato in caso contrario. Per allenare il generatore deve essere minimizzata la seguente funzione:

$$L^G = \min[\log\big(D(x)\big) + \log\Big(1 - D\big(G(z)\big)\Big)] \quad (3.8)$$

Il generatore vuole portare la probabilità di $D(G(z))$ a 1 in modo tale che il discriminatore sbagli la classificazione; quindi, il generatore vuole minimizzare $1 - D(G(z))$ [33].

Nella teoria del gioco, il modello GAN converge quando il generatore e il discriminatore raggiungono l'***equilibrio di Nash***, ovvero quando i due modelli raggiungono il massimo delle prestazioni e non sono più in grado di migliorare.

Negli utlimi anni sono state sviluppate alcune varianti delle GAN; di seguito verranno analizzate le principali varianti:

- Deep Convolutional GAN (DCGAN);
- CycleGAN;
- Pix2PixGAN.

Deep convolutional GAN (DCGAN): architettura proposta da Radford et al. [34] nel 2016.
È un'estensione diretta delle GAN tranne per il fatto che utilizza strati convoluzionali e convoluzionali trasposti rispettivamente nel discriminatore e nel generatore. Il discriminatore è composto da layers di convoluzione, layers di normalizzazione Batch e utilizza come funzione di attivazione una LeakyReLu. l'input è un'immagine di dimensioni 3x64x64 e l'output è una probabilità che l'input provenga dalla distribuzione di dati reali.
Il generatore invece consiste invece in layers di convoluzione trasposta, layers di normalizzazione Batch e utilizza una ReLu come funzione di attivazione. L'input è un vettore latente ottenuto da una distribuzione normale e l'output è un'immagine RGB di dimensioni 3x64x64. I layers di convoluzione trasposta consentono al vettore latente di essere trasformato in un volume con la stessa forma di un'immagine.

CycleGAN: è un'evoluzione della GAN in cui l'addestramento avviene in maniera non supervisionata. L'obiettivo è apprendere una funzione di mappatura $G : X \rightarrow Y$ tale che le immagini generate da $G(X)$ siano indistinguibili dalle immagini della distribuzione Y attraverso l'utilizzo di una Adversarial loss. Questa architettura permette di apprendere anche una funzione di mappatura inversa $F : Y \rightarrow X$ e introduce una funzione di consistenza per imporre
$F\big(G(X)\big) = X$ e viceversa. Il modello è composto da due generatori (G ed F) per tradurre rispettivamente immagini dal dominio X al dominio Y e viceversa, e due discriminatori (Dx e Dy) per distinguere tra gli esempi reali e quelli generati di ciascun dominio [35].

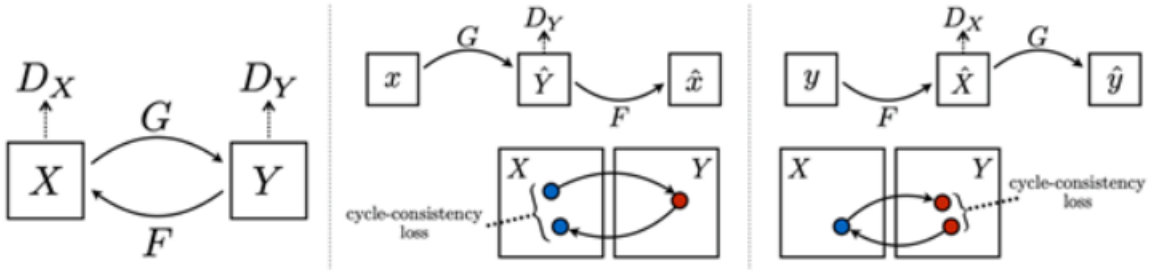

***Figura 3.14**: Procedura di allenamento di una CycleGAN* [35].

Pix2PixGAN: definisce una traduzione automatica da immagine a immagine; una volta forniti sufficienti dati di apprendimento è possibile tradurre una possibile rappresentazione di una scena in un'altra. Il generatore e il discriminatore utilizzano blocchi standard formati da layers di convoluzione, normalizzazione batch e ReLu per creare reti convoluzionali profonde.
Il generatore è realizzato utilizzando il modello dalla U-Net; l'architettura della U-Net è identica a quella presentata nella sezione precedente tranne per le skip-connection tra layers della stessa dimensione nell'encoder e nel decoder che consentono la condivisione delle informazioni tra input e output.
Il discriminatore prende un'immagine di input e un'immagine tradotta e prevede la probabilità dell'immagine tradotta di essere reale o di essere un'immagine generata dall'immagine di input.
Pix2Pix GAN utilizza una PatchGAN che permette di classificare le patch dell'immagine come reali o false invece dell'intera immagine. Il discriminatore viene eseguito in modo convoluzionale sull'immagine in cui si calcola la media di tutte le risposte per fornire l'output finale. La rete emette una singola mappa delle caratteristiche di previsioni reali e false che viene mediata per fornire un punteggio singolo [36].

3.3.1 Applicazioni nella sintesi di immagini ultrasonografiche

L'ultrasonografia è una tecnica di imaging ampiamente utilizzata per l'ispezione delle strutture anatomiche nella diagnosi clinica. Tuttavia, nelle applicazioni mediche, di solito è disponibile solo un numero molto limitato di immagini. I ricercatori hanno cercato di aggirare questo ostacolo tramite Data Augmentation; i metodi più comuni prevedono trasformazioni delle immagini di partenza tramite rotazioni, traslazioni e scaling.
Negli ultimi anni, i metodi di Data Augmentation basati sulla sintesi di immagini tramite l'utilizzo del deep learning hanno guadagnato sempre più interesse. Tra questi, l'utilizzo delle GAN è l'approccio più promettente.
Nel 2017 Tom et al. [37] hanno introdotto un metodo multistadio che include due conditional GAN per trasformare le mappe dei tessuti in immagini ultrasonografiche intravascolari sintetiche. Questo sistema è il primo ad utilizzare le label dei tessuti come input condizionato

della GAN per migliorare la stabilità dell'allenamento. La qualità delle immagini simulate è stata valuta attraverso un test visivo su valutatori esperti, i quali sono stati in grado di distinguere l'immagine reale dalla simulata con una probabilità del 50%.
Sebbene le cGAN siano efficaci e permettano di ottenere immagini controllate dall'utente, spesso le immagini sintetizzate hanno bassa risoluzione e sono affette da artefatti. Per risolvere il problema e per rendere più realistici i dettagli strutturali Shin et al. [38] e Zhang et al [39] hanno introdotto informazioni ausiliarie come lo schizzo e i bordi dello sfondo. L'obiettivo dei due studi è stato quello di utilizzare le immagini simulate per ampliare il dataset per la segmentazione rispettivamente di polipi e vasi sanguigni. È stato dimostrato che l'uso del dataset combinato aumenta la precison e la recall del 10.1 e 19,4% nella segmentazione dei polipi e dell'8,87% la sensitività nella segmentazione dei vasi sanguigni.
A causa dei maggiori dettagli nelle immagini ad alta risoluzione, il discriminatore può facilmente riconoscere le differenze tra le immagini generate e reali, che possono portare al problema della scomparsa del gradiente e rendere difficile l'allenamento. Inoltre, l'addestramento di questi modelli richiede molta memoria, il che limita l'utilizzo di batch di grandi dimensioni per migliorare la stabilità dell'allenamento.
Per risolvere questi problemi Liang et al. [40] nel 2021 hanno proposto una nuova GAN a crescita progressiva basata sugli schizzi per la sintesi US. Lo schizzo del bordo, estratto tramite l'algoritmo di Canny [41], viene sovrapposto alla maschera di segmentazione e questa nuova maschera composta diventa l'input della rete. Inoltre, per generare immagini ad alta risoluzione si utilizza una strategia di allenamento progressiva. Questo studio ha dimostrato che il miglioramento della GAN permette di ottenere immagini più realistiche e che l'aumento del dataset con le immagini simulate permette di avere una rete di segmentazione più robusta.
Per la sintesi di immagini ultrasonografiche, sono stati proposti anche metodi basati sull'utilizzo delle DCGAN. Nel 2019, infatti, Fujioka et al.[42] hanno utilizzato una DCGAN per generare immagini ecografiche mammarie dimostrando che il 22,5% delle immagini simulate e il 14% delle immagini reali risultano indistinguibili e che quindi questo tipo di rete neurale è in grado di generare immagini ultrasonografiche realistiche e di alta qualità.
Nello stesso anno è stata proposta SPADE da Park et al. [43], una GAN in grado di sintetizzare immagini realistiche aventi la stessa distribuzione statistica dei dati reali a partire da una mappa semantica. Lo studio portato avanti su più dataset supera lo stato dell'arte attuale nella sintetizzazione di immagini (pix2pix GAN). Il metodo proposto ha ottenuto un punteggio di mIoU (mean Intersection over-Union) di 35,2, che è di circa 1,5 volte meglio dell'attuale stato dell'arte, e una FID (Fréchet Inception Distance) di circa 2,2 volte migliore dei metodi precedenti.
In questo lavoro di tesi, per effettuare il Data Augmentation, è stata utilizzata SPADE in combinazione con il lavoro di Liang et al. in quanto alle maschere di segmentazione semantica fornite in input alla rete sono state sovrapposte le maschere dei bordi dell'immagine.

4. METODO PROPOSTO

In questo capitolo viene presentato il dataset e l'algoritmo utilizzato per la sintesi di nuove immagini, successivamente utilizzate come Data Augmentation per la segmentazione delle strutture del nervo ottico. Verranno inoltre discusse le metriche utilizzate per la valutazione della qualità delle immagini generate e le features di tessitura.

4.1 Creazione del dataset

Il dataset di partenza per questo lavoro di tesi è composto da 464 immagini di ecografia transorbitale (TOS). Tutte le immagini sono state acquisite in modalità B-mode seguendo il protocollo descritto nel paragrafo 2.2. Le immagini sono state acquisite su 110 soggetti differenti, sia sani che patologici, utilizzando 4 macchinari differenti. I macchinari utilizzati, con il corrispettivo numero di immagini, sono riportati in tabella 4.1.

Macchinario	*Numero immagini*
1: MyLab, Esaote - Homburg (Germania)	207
2: MyLab, Esaote - Torino (Italia)	113
3: Aplio300, Toshiba	93
4: Vivid 7, GE Healthcare	51

***Tabella 4.1**: Numero di immagini acquisite per ogni macchinario.*

Le immagini sono state fornite tutte in formato bitmap (.bmp), ma con dimensioni diverse essendo state acquisite da macchinari differenti.
Prima di procedere con l'allenamento della rete, da ogni immagine è stata ricavata una patch di dimensioni 256x256 contenete il nervo ottico. Le immagini sono anche state rinominate secondo la seguente notazione:

Mac_x_Sub_yy_Im_zz

Dove *x* è il numero corrispondente al macchinario (da 1 a 4), *yy* rappresenta il numero del soggetto (da 1 a 110) e *zz* corrisponde al numero di immagine selezionata per un soggetto specifico.
Il dataset di partenza contiene anche le maschere binarie delle guaine del nervo ottico, di dimensioni 256x256 in formato png, ottenute in un precedente lavoro di tesi [44].

Prima di procedere con l'allenamento della rete è stato necessario creare le maschere di segmentazione delle restanti strutture del nervo ottico.

Il primo step è stato quello di realizzare le maschere del nervo ottico. In MATLAB R2021a è stato sviluppato un algoritmo automatico che, presa in ingrasso la maschera di segmentazione delle guaine del nervo ottico, tramite la funzione ***bwconhull***, restituisce tutta la zona connessa tra le due guaine. Andando a sottrarre all'area ottenuta quella delle guaine si ottiene la maschera del nervo ottico. In figura 4.1 vengono mostrati gli step per la realizzazione della maschera.

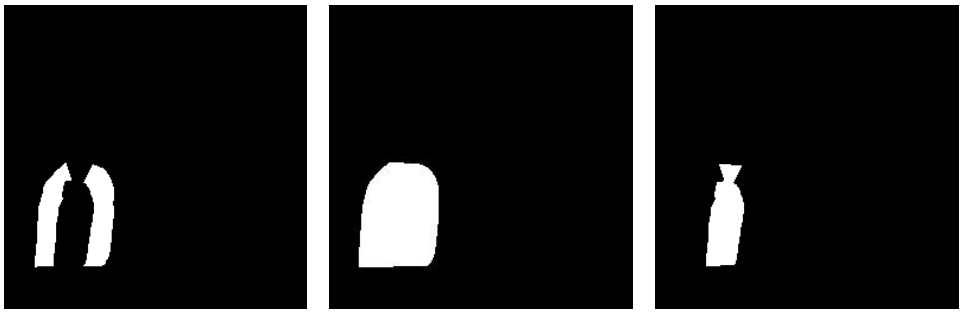

Figura 4.1*: Processo per la creazione delle maschere del nervo ottico. Da sinistra a destra: maschera delle guaine del nervo ottico, convexhull delle guaine, maschera del nervo ottico.*

Una volta ottenute le maschere del nervo ottico si è passato alla creazione delle maschere del bulbo oculare. Le maschere sono state ottenute manualmente tramite l'app ***Image Labeler*** di MATLAB R2021a andando a selezionare i punti corrispondenti al bordo del bulbo oculare cercando di seguire la forma fisiologica dell'occhio. L'unione automatica dei punti permette di ottenere la ROI di interesse a partire dalla quale si crea la maschera binaria (Figura 4.2).

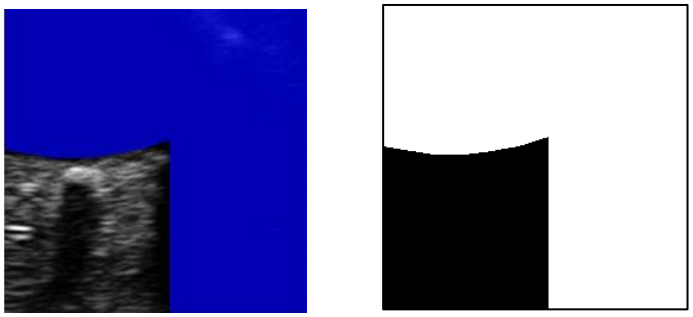

Figura 4.2*: A sinistra è rappresentata l'immagine segmentata manualmente e a destra la corrispondente maschera binaria del bulbo oculare.*

Infine, sono state aggiunte le maschere dei bordi dell'immagine (Figura 4.3). Tale maschera è stata ottenuta automaticamente su MATLAB R2021a tramite la funzione ***edge***. La funzione riconosce i bordi dell'immagine originale tramite l'***algoritmo di Canny*** sviluppato da John Canny nel 1986 [41].

L’algoritmo prevede le seguenti fasi:

- Rimozione del rumore tramite un filtro gaussiano di dimensioni 5x5;
- Calcolo del gradiente di intensità dell’immagine tramite filtro di Sobel; il filtro viene applicato sia in direzione verticale che orizzontale per ottenere la derivata prima in entrambe le direzioni. Da queste due immagini è possibile trovare il gradiente e la direzione del bordo per ogni pixel;
- Rimozione di pixel che non sono massimi locali rispetto all’orientazione del gradiente;
- Selezione finale dei bordi; al fine di selezionare solo i bordi significativi si utilizza un procedimento chiamato isteresi: vengono definite due soglie, una bassa e una alta, che vengono confrontate con il gradiente in ciascun punto. Se il valore del gradiente è inferiore alla soglia bassa il punto è scartato, se è superiore alla soglia alta il punto è accettato come parte del contorno, se invece il punto è compreso tra le due soglie questo viene accettato solamente se contiguo ad un punto già precedentemente accertato. Al termine di questo step si ottiene un'immagine binaria dove ciascun pixel è marcato come appartenente o no ad un contorno.

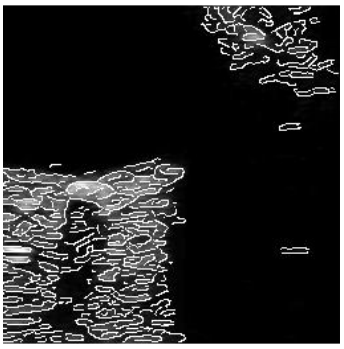

***Figura 4.3**: Esempio di maschera di segmentazione dei bordi sovrapposta all’immagine originale.*

Le maschere di ogni label sono state unite per ottenere le maschere di segmentazione finali che saranno l’input di SPADE per la simulazione delle immagini. Ad ogni pixel è stato assegnato un valore in base alla label di appartenenza:

- 0 → pixel del background;
- 1 → pixel delle guaine;
- 2 → pixel del nervo ottico;
- 3 → pixel del bulbo oculare;
- 4 → pixel dei bordi (Canny).

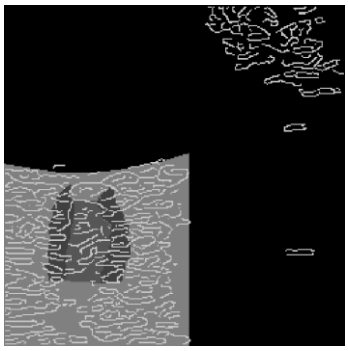

***Figura 4.4**: Esempio di maschera di segmentazione totale.*

4.2 Algoritmo SPADE

Per l'allenamento della rete è stato utilizzato l'***algoritmo SPADE*** (Spatially Adaptative De-Normalization). SPADE è una GAN per la sintesi di immagini aventi la stessa distribuzione statistica delle immagini reali a partire da una mappa di segmentazione semantica proposta per la prima volta nel 2019 da Park et al. [43]. Si tratta di una sintetizzazione di immagini condizionata in quanto SPADE necessita di dati esterni.

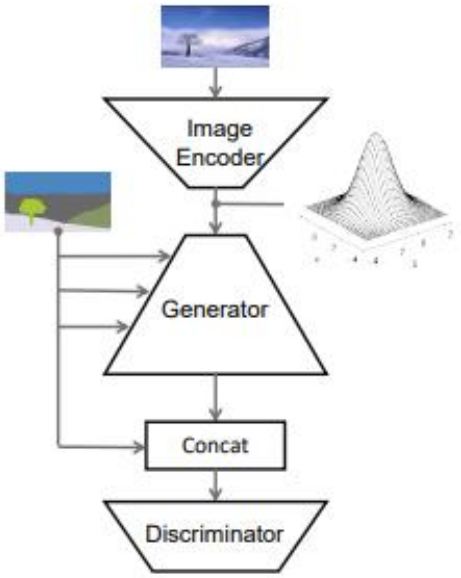

***Figura 4.5**: Architettura generale di SPADE* [43].

L'architettura generale (Figura 4.5) usa il generatore e il discriminatore di una GAN. Il generatore prende in input una distribuzione uniforme di rumore gaussiano a cui viene sommata una distribuzione di media e varianza dell'immagine di input calcolata utilizzando un encoder. L'encoder consiste in 6 layers di convoluzione seguiti da due layers lineari che codifica le informazioni di stile dell'immagine di partenza e restituisce in output media e

varianza. La distribuzione reale, tramite la KL Divergence, viene approssimata nella distribuzione gaussiana che viene data in input al generatore.
La distribuzione del rumore casuale, combinata con la codifica dello stile dell'immagine di input viene data al primo blocco di convoluzione del generatore. L'output del blocco di convoluzione con i suoi filtri appresi viene sommato con un layer chiamato SPADE Residual Block (ResBlk). L'output del blocco viene sovracampionato utilizzando il pooling. Il ResBlk è la combinazione di più layers di normalizzazione SPADE e convoluzioni e prende in input la mappa di segmentazione semantica. Questi blocchi sono di natura residuale. Funzionano su diverse scale/risoluzioni delle maschere di segmentazione, che fungono da dati esterni per modulare i layers di normalizzazione. L'ultimo layer di convoluzione genera un'immagine ad alta risoluzione che viene data in input al discriminatore per l'allenamento.
Il layer di normalizzazione SPADE utilizza la maschera di segmentazione per modulare i layer di attivazione. Le mappe aiutano a preservare le informazioni semantiche dell'immagine di input attraverso il layer di normalizzazione e aiutano anche a mappare correttamente le informazioni di stile codificate dall'input.
Il discriminatore prende in input la concatenazione dell'immagine generata e della maschera di segmentazione semantica. È costituito da più layers di convoluzione che generano un singolo output che può essere reale o falso. D è un discriminatore multiscala che funziona bene per generare immagini ad alta risoluzione. Una volta allenata la rete la parte corrispondente al discriminatore può essere eliminata.

4.3 Allenamento della rete

Prima di procedere con l'allenamento della rete, le immagini sono state convertite in formato jpg e il dataset è stato suddiviso in maniera random in TrainSet, a cui è stato assegnano l'80% delle immagini (371), e in TestSet a cui è stato assegnato il restante 20% (93).
La rete per la generazione delle immagini è stata allenata in PyTorch sul TrainSet e successivamente l'addestramento è stato valutato sul TestSet. In entrambi i casi la rete prende in ingresso una cartella contenente le immagini e una contenente le corrispettive maschere di segmentazione e restituisce in uscita le immagini generate.

Per l'allenamento è stato necessario settare una serie di opzioni da fornire alla rete.
L'algoritmo di ottimizzazione scelto è ADAM (ADAptive Moment estimation) con un learning rate iniziale pari a 0.0002 e con β_1=0 e β_2=0.9.
È stato impostato una Batch size pari a 4 e la rete è stata allenata su 120 epoche settando le opzioni in modo tale da avere un learning rate pari a quello iniziale per 100 epoche, mentre il learning rate decade linearmente a zero per le restanti 20 epoche.

Il generatore è stato allenato con lo stesso discriminatore multiscala utilizzato nel modello pix2pix utilizzando come funzione di perdita la *Hinge Loss*. Tale funzione è utilizzata comunemente per l'allenamento dei classificatori ed è definita come:

$$l(y) = \max(0,1 - t \cdot y) \qquad (4.1)$$

Dove t=±1 è l'output previsto e y l'output del classificatore.

Per ogni epoca, le immagini prodotte dalla rete sono state confrontate con quelle reali tramite quattro metriche di valutazione della qualità delle immagini per valutare l'andamento delle performance del modello nel task di generazione dell'immagine all'aumentare del numero di epoche. Le metriche scelte, che verranno spiegate nel dettaglio nel paragrafo successivo, sono le seguenti:

- Peak Signal-to Noise Ratio;
- Structural Similarity;
- Root Mean Square Error;
- Feature Similarity Index Matrix.

4.4 Metriche di valutazione

Per valutare la qualità delle immagini generate da SPADE sono state calcolate quattro metriche sia sul TrainSet che sul TestSet. Ogni metrica è stata calcolata sull'intera immagine e sulle singole label (ON, ONS e bulbo oculare) andando a confrontare immagine reale e immagine simulata.
Di seguito vengono descritte le metriche adoperate per la valutazione dei risultati.

1. ***Peak Signal-to Noise Ratio (PSNR)***

   Il rapporto segnale-rumore di picco è una misura utilizzata per valutare la qualità di un'immagine compressa rispetto all'originale. È definito come rapporto tra la massima potenza del segnale e la potenza del rumore che può invalidare la fedeltà della sua rappresentazione compressa [45]. Il segnale è considerato come il dato originale e il rumore è l'errore prodotto dalla compressione o distorsione. Il PSNR è solitamente espresso in termini di scala logaritmica di decibel ed è la tecnica di valutazione più comunemente utilizzata per le immagini che hanno delle compressioni lossy quindi in cui vi è perdita di qualità.
   Maggiore è il valore del PSNR maggiore è la somiglianza dell'immagine simulata con quella reale.

Il PSNR (4.3) viene definito attraverso l'errore quadratico medio (MSE) (4.2):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \qquad (4.2)$$

$$PSNR = 10log_{10}\frac{L^2}{MSE} \qquad (4.3)$$

Dove N è il numero di pixel presenti nell'immagine, cioè $N = m * n$ dove n e m rappresentano rispettivamente l'altezza e la larghezza dell'immagine, x_i e y_i si riferiscono all'i-esimo pixel rispettivamente nell'immagine di riferimento e nell'immagine simulata; L è il range dinamico dei valori dei pixel, cioè il massimo valore possibile dei pixel dell'immagine, per un segnale che presenta n bits/pixel $L = 2^n - 1$.

2. Structural Similarity Index (SSIM)

L'indice di somiglianza strutturale (SSIM) è un modello basato sulla percezione. In questo metodo, il degrado dell'immagine è considerato come il cambiamento della percezione in informazioni strutturali. L'informazione strutturale è l'idea che i pixel abbiano forte interdipendenza soprattutto quando sono spazialmente vicini. Questa dipendenza porta informazioni importanti sulla struttura degli oggetti. Il *mascheramento della luminanza* è un termine in cui la parte di distorsione di un'immagine è meno visibile nelle regioni più luminose. Mentre il mascheramento del contrasto è un termine in cui le distorsioni tendono ad essere meno visibili dove c'è un'attività significativa o "trama" nell'immagine.

SSIM stima la qualità percepita di immagini e video, misura la somiglianza tra due immagini: l'originale e la simulata. SSIM è un valore decimale tra -1 e 1, dove 1 indica perfetta similarità, 0 indica che non c'è similarità e -1 indica perfetta anti-correlazione [46].

La differenza rispetto all'MSE e al PSNR è che l'SSIM valuta l'errore assoluto.

L'indice di somiglianza strutturale è descritto dalla seguente formula:

$$SSIM\ (x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{({\mu_x}^2 + {\mu_y}^2 + C_1)({\sigma_x}^2 + {\sigma_y}^2 + C_2)} \qquad (4.4)$$

Dove x e y sono rispettivamente l'immagine reale e l'immagine simulata, μ_x e ${\sigma_x}^2$ la media e la varianza di x, μ_y e ${\sigma_y}^2$ la media e la varianza di y e σ_{xy} la covarianza di x e y.
$C_1 = (k_1 L)^2$ e $C_2 = (k_2 L)^2$ sono due variabili per stabilizzare il denominatore.
La 4.3 si basa su 3 misure di confronto tra i campioni di x e y:

- Luminanza: $$l\,(x,y) = \frac{2\mu_x\mu_y + C_1}{{\mu_x}^2 + {\mu_y}^2 + C_1} \quad (4.5)$$

- Contrasto: $$c\,(x,y) = \frac{2\sigma_x\sigma_y + C_2}{{\sigma_x}^2 + {\sigma_y}^2 + C_2} \quad (4.6)$$

- Struttura: $$s\,(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (4.7)$$

  con $C_3 = C_2/2$.

3. Root Mean Square Error (RMSE)

L'errore quadratico medio (RMSE) è una misura usata frequentemente per valutare la differenza tra i valori previsti da un modello o da uno stimatore e i valori osservati [47]. Rappresenta la deviazione standard dei residui. Il termine residuo si riferisce alla distanza tra il punto previsto e il punto osservato. Essendo la deviazione standard, indica quanto sono distribuiti i residui attorno alla linea di regressione. L'RMSE può essere considerato come una sorta di distanza tra il vettore dei valori previsti e il vettore dei valori osservati.
L'RMSE è sempre non negativo e un valore di 0 indica un perfetto adattamento dei dati. In generale, un valore basso di RMSE è migliore di un valore alto.
L'RMSE (4.8) è la radice quadrata dell'errore quadratico medio (MSE) la cui formula è riportata sopra (4.2):

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2} \quad (4.8)$$

4. Feature Similarity Index Matrix (FSIM)

L'indice di similarità delle caratteristiche (FSIM) mappa le caratteristiche e misura la similarità tra due immagini. Per descrivere l'FSIM è necessario descrivere due criteri [48]:

- Phase Congruency (PC): metodo per rilevare le caratteristiche delle immagini in congruenza di fase. PC è invariante alla variazione di luce nell'immagine. Sottolinea le caratteristiche dell'immagine nel dominio della frequenza. La congruenza di fase è invariante al contrasto.

- Gradient Magnitude (GM): gli operatori di gradiente possono essere espressi tramite maschere di convoluzione. I tre operatori di gradiente comunemente utilizzati sono l'operatore di Sobel, l'operatore di Prewitt e l'operatore di Scharr. GM può essere definito come:

$$GM = \sqrt{{G_x}^2 + {G_y}^2} \tag{4.9}$$

  Dove G_x e G_x sono rispettivamente il gradiente orizzontale e verticale dell'immagine.

FSIM può essere definito e calcolato sulla base di PC_1, PC_2, G_1, G_2 che rappresentano congruenza di fase e gradient magnitude dell'immagine reale e dell'immagine simulata.
La similarità tra le due immagini dipende dalla similarità basata su PC (4.10) e dalla similarità bastata su GM (4.11).

$$S_{PC} = \frac{2PC_1PC_2 + T_1}{{PC_1}^2 + {PC_2}^2 + T_1} \tag{4.10}$$

$$S_G = \frac{2G_1G_2 + T_2}{{G_1}^2 + {G_2}^2 + T_2} \tag{4.11}$$

Dove T_1 e T_2 sono due costanti che dipendono rispettivamente da PC e GM.
La 4.10 e la 4.11 vengono combinate per calcolare la similarità:

$$S_L(x) = [S_{PC}(x)]^{\alpha} \cdot [S_G(x)]^{\beta} \tag{4.12}$$

Dove i parametri α e β vengono utilizzati per regolare l'importanza relativa delle caratteristiche di PC e GM.
FSIM può assumere valori compresi tra 0 e 1.

4.5 Feature di tessitura

Nell'elaborazione di immagini il termine tessitura si riferisce alla disposizione geometrica e ripetitiva dei toni di grigio.
Le proprietà di tessitura non possono essere sempre estratte considerando i valori di un singolo pixel, infatti, per ottenere informazioni valide è spesso necessario calcolare le variazioni in un contorno di pixel. Uno dei metodi più utilizzati per analizzare la distribuzione spaziale dei livelli di grigio è quello statistico o, ad esempio computando la probabilità di co-occorrenza di valori di grigio in differenti distanze e orientamenti. Il metodo statistico può calcolare i valori dei singoli pixel, attraverso le feature di tessitura del primo ordine, oppure su coppie di pixel, attraverso le feature di tessitura di ordini superiori [49].

4.5.1 Feature di tessitura del primo ordine

Le feature di tessitura del primo ordine sono operatori statistici basilari che dipendono solamente dall'istogramma delle luminosità. Permettono di misurare la verosimiglianza nell'osservare un valore di grigio in posizioni random nell'immagine. Le feature di primo ordine si possono calcolare dall'istogramma dei livelli dei grigi dei pixel dell'immagine; questo dipende solo dal singolo grigio del pixel e non dall'interazione con i pixel dell'intorno.
In questo lavoro di tesi sono state calcolate sei feature, ognuna è stata calcola: la media dei toni di grigio dell'immagine, la deviazione standard, la varianza, la skewness, la kurtosis e l'entropia. Ogni feature è stata calcola sia sull'immagine reale che su quella generata e sono poi state confrontate.
In tabella 4.2 sono riportate le sei feature di tessitura con le corrispettive formule matematiche dove M è il numero di righe e N il numero di colonne della roi in esame.

Feature	*Formula*
Media (μ)	$\mu = \sum_{x=1}^{M}\sum_{y=1}^{N}\frac{I(x,y)}{M \cdot N}$
Deviazione standard ($\boldsymbol{\sigma}$)	$\sigma = \sqrt{\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\{I(x,y)-\mu\}^2}{M \cdot N}}$
Varianza ($\boldsymbol{\sigma^2}$)	$\sigma^2 = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\{I(x,y)-\mu\}^2}{M \cdot N}$
Skewness (S_k)	$\mathrm{Sk} = \frac{1}{M \cdot N} \cdot \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\{I(x,y)-\mu\}^3}{\sigma^3}$
Kurtosis (K_t)	$\mathrm{Kt} = \frac{1}{\mathrm{M} \cdot \mathrm{N}} \cdot \frac{\sum_{x=1}^{\mathrm{M}}\sum_{y=1}^{\mathrm{N}}\{\mathrm{I(x,y)}-\mu\}^4}{\sigma^4}$
Entropia (E_1)	$\mathrm{E1} = -\sum_{x=1}^{M}\sum_{y=1}^{N} I(x,y) \cdot log_2\, I(x,y)$

***Tabella 4.2**: Feature di tessitura del primo oridne .*

4.5.2 Feature di tessitura del secondo ordine

Le feature di tessitura del secondo ordine dipendono dalla posizione relativa dei pixel e vanno calcolate secondo direzioni specifiche. Sono basate sulla matrice di co-occorrenza (GLCM: Gray Level Co-occurence Matrix) che rappresenta l'istogramma bidimensionale dei livelli di grigio dell'immagine.

La GLCM è una matrice quadrata di dimensioni pari al numero di livelli di toni di grigio nell'immagine che verrà indicata con la lettera C. Il termine $C(i,j)$ indica il numero di volte in cui un pixel par a i (livello di grigio) si trova adiacente a un pixel di valore j.

Due pixel possono essere adiacenti in orizzontale, in verticale o nelle due direzioni diagonali; sono quindi state calcolate 4 GLCM diverse, ognuna basata su una specifica direzione (0°, 45°, 90°, 135°). In figura 4.5 [50] è mostrato un esempio di costruzione della matrice di co-occorrenza in direzione orizzontale.

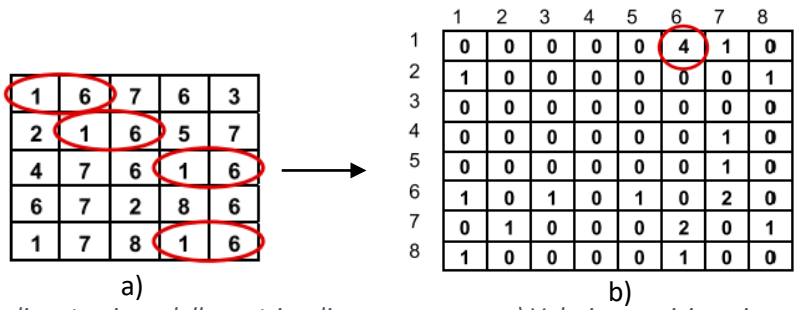

***Figura 4.6**: Esempio di costruzione della matrice di co-occorrenza. a) Valori numerici corrispondenti all'intensità dei pixel dell'immagine. b) GLCM corrispondente* [50].

Dopo aver calcolato la GLCM si passa al calcolo delle feature di Haralick che sono i descrittori matematici della matrice di co-occorrenza.
In tabella 4.3 sono riportate feature di tessitura del secondo ordine con le corrispettive formule matematiche.

Haralick Feature	*Formula*
Contrasto (I_{con})	$I_{con} = \sum_{n=0}^{N-1} n^2 \{ \sum_{i=0}^{N} \sum_{j=0}^{N} P(i,j) \}$
Correlazione (I_{cor})	$I_{cor} = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i,j) P(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$
Energia (I_{enrg})	$I_{enrg} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i,j)^2$
Omogeneità (I_{hmg})	$I_{hmg} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{1}{1+(i-j)^2} P(i,j)$

***Tabella 4.3**: Feature di tessitura del secondo ordine. σ_x, σ_y, μ_x, μ_y sono la deviazione standard e la media di Px e Py, che sono le funzioni di densità di probabilità parziali. px(i)=i^{th} entra nella matrice di probabilità marginale ottenuta sommando le colonne di P(i,j).*

In questo lavoro le feature di tessitura del secondo ordine sono state calcolate per tutte e quattro le direzioni angolari e successivamente è stata fatta una media dei valori ottenuti per ogni immagine.

5. SEGMENTAZIONE

Dopo aver valutato le prestazioni della rete descritta nel capitolo precedente, si passa al task di segmentazione delle strutture del nervo ottico.
In questo capitolo vengono descritti gli step per la creazione di nuove maschere di segmentazione da utilizzare come input della rete allenata precedentemente per fare Data Augmentation. Le nuove immagini ecografiche generate sono state successivamente utilizzate per l'allenamento della rete di segmentazione che verrà descritta in questo capitolo insieme alle metriche utilizzate per valutarla.

5.1 Data Augmentation

Dopo aver allenato SPADE, la GAN viene utilizzata come Data Augmentation e quindi per la sintetizzazione delle nuove immagini da utilizzare successivamente per l'allenamento della rete per la segmentazione delle strutture del nervo ottico.
Il ***Data Augmentation*** è un insieme di tecniche che permettono di ampliare il dataset a disposizione. Ci possono essere due tipologie di augmented data:

- Copie leggermente modificate di dati già esistenti; vengono considerate aumented data le immagini create sulla base di cambiamenti casuali come rotazioni, capovolgimenti, modifiche del colore, aggiunta di rumore, tagli, ecc.
- Dati sintetici realizzati a partire dal dataset iniziale attraverso l'utilizzo delle GAN;

Gli augmented data vengono utilizzati per risolvere il problema dell'*overfitting* ovvero il sovradattamento del modello statistico al campione di dati osservato, che avviene quando il modello ha troppi parametri rispetto al numero di osservazioni eseguito.
L'utilizzo del data augmentation permette di far crescere la capacità di apprendimento delle reti neurali artificiali.

Per riuscire ad ottenere un dataset più ampio sono state create delle nuove maschere di segmentazione (label map) sintetiche a partire dalle quali sono poi state generate le nuove immagini utilizzando l'algoritmo SPADE.

Le nuove label map sono state create a partire dalle maschere associate alle immagini reali andando a definire tre elementi differenti:

- Label della zona periferica (background + bulbo oculare);

- Label Canny;
- Label del complesso ON (nevo ottico + guaine).

Ogni elemento è stato estratto in maniera random da un'immagine diversa e, tramite un algoritmo automatico, descritto dal flow-chart in figura 5.4, sono stati uniti per ottenere la label map finale.
Prima di procedere alla creazione delle nuove label map, sono state analizzate le immagini generate da SPADE per andare ad estrarre caratteristiche comuni alle immagini che presentavano valori delle features di tessitura compresi nella distribuzione delle immagini reali. Tali caratteristiche, ed in particolare l'area delle label, sono state utilizzate per stabilire delle soglie da imporre per la generazione delle label map.

Gli step seguiti per ottenere le label map sono i seguenti:

1. Si estrae, in maniera random, una label della zona periferica. L'area di tale label viene confrontata con una soglia, scelta come descritto precedentemente. Se l'area è maggiore della soglia si passa allo step successivo, altrimenti viene estratta una nuova label della zona periferica.

***Figura 5.1**: Label map della zona periferica.*

2. Si procede estraendo, sempre in maniera random, una label Canny. Dopo aver verificato che la label non sia stata estratta dalla stessa immagine dalla quale è stata estratta la label della zona periferica si procede effettuando il convexhull della label Canny per ottenere l'area della ROI. Come per la label precedente, si verifica che la ROI sia maggiore di una soglia e che allo stesso tempo sia maggiore o uguale all'area ottenuta dall'unione delle labels di Canny e della zona periferica. Se queste due condizioni sono verificate si aggiunge la label Canny, altrimenti se ne estrae un'altra.

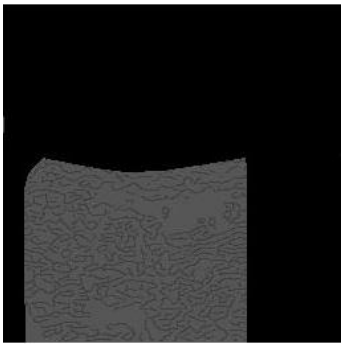

***Figura 5.2**: Label map del complesso ON + Label map della zona periferica.*

3. L'ultimo step prevede l'estrazione random della label del complesso ON. Anche in questo caso si verifica che la label non sia stata estratta dalle stesse immagini dalle quali sono state estratte le label precedenti. Si verifica, successivamente, se l'unione della nuova label e della label della zona periferica è uguale all'area di quest'ultima, questo permette di verificare se il complesso ON è interno al bulbo oculare, e se il complesso ON si trova in posizione fisiologica, ovvero non troppo distante dal bordo del bulbo oculare. Se queste condizioni sono verificate si aggiunge la label e si ottiene la maschera di segmentazione finale, altrimenti si estrae un'altra label.

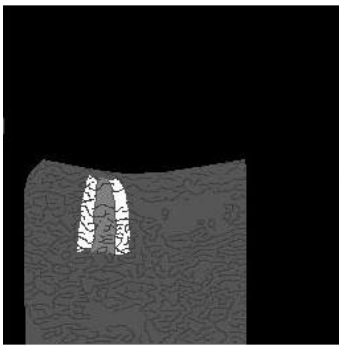

***Figura 5.3**: Label map finale.*

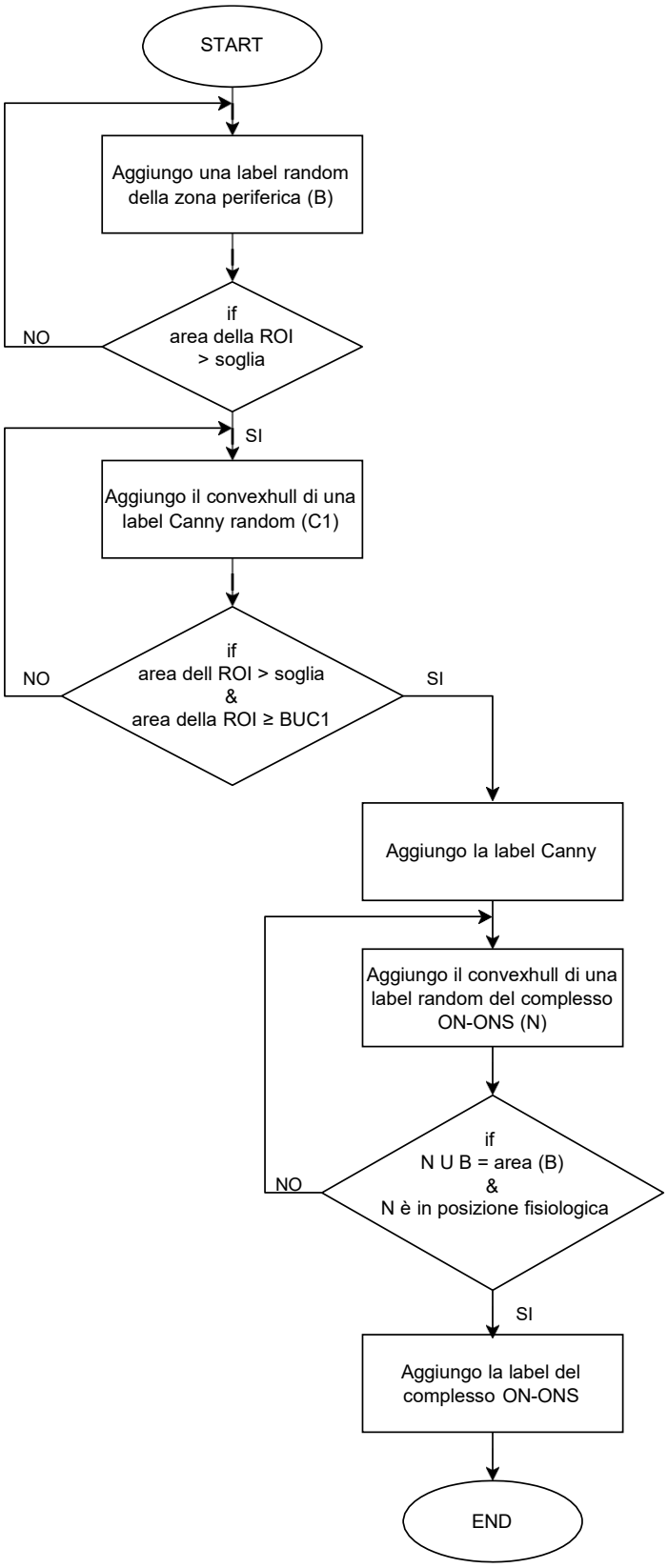

***Figura 5.4**: Flow-chart dell'algoritmo utilizzato per la creazione delle nuove Label Map.*

L'algoritmo appena descritto restituisce in uscita 464 nuove label map ed è stato ripetuto 11 volte, permettendo quindi di costruire 11 dataset diversi da utilizzare come input per la generazione delle nuove immagini.
L'obiettivo di questo lavoro è quello di fare Data Augmentation andando a generare immagini con stesso stile, ma diverso contenuto morfologico e immagini con stili differenti ma stesso contenuto morfologico. Per questo motivo i dataset sono stati costruiti andando a selezionare 5 immagini con stile differente e ad ognuna sono state associate tutte e 464 le maschere di segmentazione ottenendo così 11 dataset composti da 2320 immagini e 2320 maschere.
Ogni dataset è stato utilizzato come input per SPADE ottenendo in output, per ognuno, 2320 immagini simulate. Tutte le immagini ottenute sono state valutate per selezionare quelle da utilizzare per costruire il dataset di training della rete di segmentazione.
La selezione è stata fatta andando a valutare per ogni immagine, sia reale che simulata, le feature di tessitura del primo ordine su tre label:

- ON;
- ONS;
- Bulbo oculare.

Un'immagine viene selezionata se presenta valori delle feature di tessitura compresi nella distribuzione delle immagini reali e quindi se, per ogni feature e per ogni label, viene rispettata la seguente uguaglianza:

$$fs = mean(fr) \pm std(fr)$$

Dove fs e fr sono la feature in esame rispettivamente calcolata sull'immagine simulata e reale.
Utilizzando questo metodo sono state selezionate 518 immagini simulate ad ognuna delle quali è stata associata la corrispettiva maschera di segmentazione a 3 classi:

- Classe 0: pixel del background;
- Classe 1: pixel delle guaine del nervo ottico;
- Classe 2: pixel del nervo ottico.

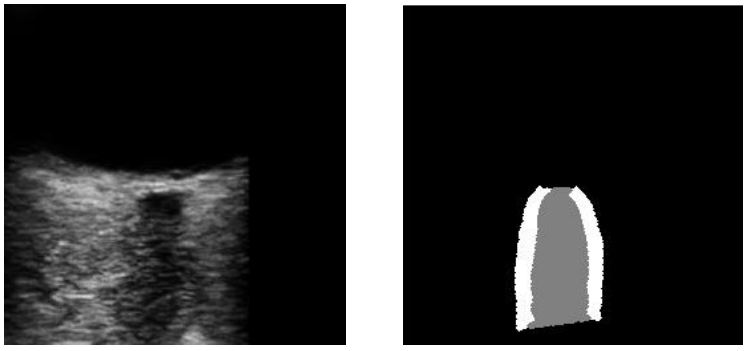

***Figura 5.5**: Esempio di immagine simulata e corrispettiva maschera di segmentazione.*

5.2 Rete per la segmentazione

La rete utilizzata per la segmentazione delle strutture del nervo ottico è una U-Net nella quale è stato utilizzato come encoder un modello di CNN pre-addestrato: ResNet50 e sono stati importati i pesi ottenuti dall'allenamento della CNN sul dataset ImageNet.
La rete è stata allenata 3 volte, prendendo in input dataset diversi:

- Dataset formato solo dalle immagini reali;
- Dataset formato solo dalle immagini simulate;
- Dataset formato da immagini reali e simulate.

L'algoritmo di ottimizzazione scelto per l'allenamento della rete è ADAM con un learning rate iniziale di 0.0002. La funzione di attivazione utilizzata è la *softmax* che restituisce in output la probabilità di un dato input di appartenere ad una determinata classe.
È stato impostato un Batch size pari ad 8 ed è stata scelta come funzione di perdita la *Dice Loss*. Questa funzione di perdita deriva dal Dice Coefficient, una metrica per valutare i risultati della segmentazione ed è definita come segue [51]:

$$DL(y, \hat{p}) = 1 - \frac{2y\hat{p} + 1}{y + \hat{p} + 1} \qquad (5.1)$$

Dove y è il è l'output atteso e $\hat{p}$ l'output predetto dalla rete.

Dopo ogni allenamento la rete è stata testata sul dataset originale e i risultati della segmentazione delle guaine del nervo ottico sono stati salvati come maschere binarie. Gli output delle tre reti sono stati confrontati tramite due metriche di valutazione, che verranno descritte nel paragrafo successivo, che sono:

- Dice similarity coefficient;
- Hausdorff distance.

a) b) c) d)

***Figura 5.6**: Esempio dell'output della rete di segmentazione. a) Immagine reale. b) Segmentazione ottenuta allenando la rete con le immagini reali. c) Segmentazione ottenuta allenando la rete con le immagini simulate. d) Segmentazione ottenuta allenando la rete con immagini reali e simulate.*

5.3 Metriche di valutazione

Per valutare l'accuratezza della segmentazione delle guaine del nervo ottico sono state valutate due metriche che permettono di confrontare i risultati ottenuti con il Ground Truth, rappresentato dalle maschere di ONS.
Le metriche sono state calcolate sugli output di tutte e tre le reti allenate e sono successivamente stati confrontati.
Le metriche utilizzate per la valutazione dei risultati sono le seguenti:

1. ***Dice similarity coefficient (DSC)***

   Il coefficiente di similarità Dice, conosciuto anche come indice di Sørensen–Dice, è un metodo statistico che permette di misurare la somiglianza tra due set di dati. È definito dalla seguente formula:

   $$DSC = \frac{2|A \cap B|}{|A| + |B|} \qquad (5.2)$$

   Dove A rappresenta il ground truth e B il risultato della segmentazione.
   Il coefficiente di Dice può assumere valori compresi ne range [0,1], dove 1 indica massima somiglianza tra output atteso e output predetto.

2. ***Hausdorff distance (HD)***

   La distanza di Hausdorff misura quanto sono vicini tra loro due bordi. Due contorni sono vicini nella distanza di Hausdorff se ogni punto di entrambi i set è vicino ad alcuni punti dell'altro set. HD è la massima distanza di un set dal punto più vicino nell'altro set ed è definita come segue:

   $$HD_{(a \in A, b \in B)} = \max\{d(a,b)\} \qquad (5.3)$$

   Dove $d(a,b)$ è la distanza euclidea tra un punto a (sul bordo di A) e un punto b (sul bordo di B).

6. RISULTATI

In questo capitolo verranno discussi i risultati ottenuti da SPADE nel task di generazione delle immagini, andando a confrontare le metriche di valutazione calcolate sulle immagini simulate e su quelle reali. Verranno inoltre discussi i risultati della segmentazione ottenuta dall'allenamento della U-Net con i tre diversi dataset, andando a confrontare gli output della rete con il ground truth.

6.1 Risultati dell'allenamento di SPADE

Per valutare le prestazioni di SPADE e quindi la qualità delle immagini generate, vengono riportati i risultati delle metriche descritte nel paragrafo 4.4. Ogni metrica è stata calcolata sia sul TrainSet che sul TestSet e per entrambi i set di dati ogni metrica è stata valutata sia sull'intera immagine che sulle singole label (ON, ONS, bulbo oculare).
Di seguito sono riportati i risultati delle metriche calcolate sull'intera immagine (Tabella 6.1), sulla label di ON (Tabella 6.2), sulla label di ONS (Tabella 6.3) e sulla label del bulbo oculare (Tabella 6.4).

IMMAGINE	*PSNR [dB]*	*SSIM*	*FSIM*	*RMSE*
Train Set	21,408 ± 3,036	0,713 ± 0,096	0,869 ± 0,046	0,188 ± 0,107
Test Set	19,531 ± 2,521	0,670 ± 0,092	0,851 ± 0,050	0,280 ± 0,158

***Tabella 6.1**: Metriche di valutazione della qualità delle immagini generate da SPADE calcolate sull'intera immagine. Si riportano i valori medi e le rispettive deviazioni standard.*

ON	*PSNR [dB]*	*SSIM*	*FSIM*	*RMSE*
Train Set	36,840 ± 6,065	0,992 ± 0,006	0,996 ± 0,004	0,159 ± 0,093
Test Set	35,556 ± 7,307	0,990 ± 0,007	0,996 ± 0,005	0,271 ± 0,155

***Tabella 6.2**: Metriche di valutazione della qualità delle immagini generate da SPADE calcolate sulla label del nervo ottico. Si riportano i valori medi e le rispettive deviazioni standard.*

ONS	*PSNR [dB]*	*SSIM*	*FSIM*	*RMSE*
Train Set	35,866 ± 5,058	0,993 ± 0,006	0,996 ± 0,003	0,110 ± 0,060
Test Set	33,932 ± 6,012	0,991 ± 0,006	0,991 ± 0,003	0,244 ± 0,130

***Tabella 6.3**: Metriche di valutazione della qualità delle immagini generate da SPADE calcolate sulla label delle guaine del nervo ottico. Si riportano i valori medi e le rispettive deviazioni standard.*

BULBO	*PSNR [dB]*	*SSIM*	*FSIM*	*RMSE*
Train Set	23,985 ± 3,463	0,913 ± 0,047	0,934 ± 0,039	0,180 ± 0,093
Test Set	22,124 ± 3,104	0,893 ± 0,049	0,893 ± 0,042	0,318 ± 0,129

***Tabella 6.4**: Metriche di valutazione della qualità delle immagini generate da SPADE calcolate sulla label del bulbo oculare. Si riportano i valori medi e le rispettive deviazioni standard.*

Osservando i risultati è subito evidente come i valori delle metriche calcolate sull'intera immagine siano inferiori rispetto ai valori delle metriche calcolate sulle singole label.
È possibile affermare che SPADE genera immagini con una buona qualità in quanto i valori di PSNR sono superiori a 20 dB, che per immagini a 8bit rappresenta il limite inferiore per il quale la qualità dell'immagine può essere considerata buona.
Analizzando i risultati si osserva un indice di similarità superiore a 0,7 se si considera l'intera immagine, mentre per le singole label, ed in particolare per ON e ONS, l'indice di similarità è prossimo a 1. Ciò indica che le immagini simulate sono molto simili a quelle reali e che quindi la GAN riesce a generare immagini realistiche riuscendo a simulare particolarmente bene le strutture del nervo ottico. Quanto appena detto può essere affermato anche andando ad osservare i valori di FSIM, che risultano essere simili a quelli di SSIM sulle singole label, e leggermente superiori se si considera l'intera immagine con una similarità tra immagine reale e simulata che supera l'86%.
Infine, i valori di RMSE prossimi allo 0 indicano che le immagini generate hanno una distribuzione di valori che si adatta molto bene alla distribuzione dei valori dell'immagine reale.

Sono stati inoltre tracciati i grafici per visualizzare l'andamento medio delle metriche all'aumentare del numero di epoche in modo tale da poter valutare le prestazioni dell'algoritmo. Vengono riportati solo i grafici che mostrano l'andamento delle metriche sull'intera immagine in quanto questi sono indicativi anche per l'andamento delle metriche sulle singole label.
Di seguito sono riportati i grafici che mostrano i valori di media ± deviazione standard di: PSNR (Figura 6.1), SSIM (Figura 6.2), RMSE (Figura 6.3), FSIM (Figura 6.4).

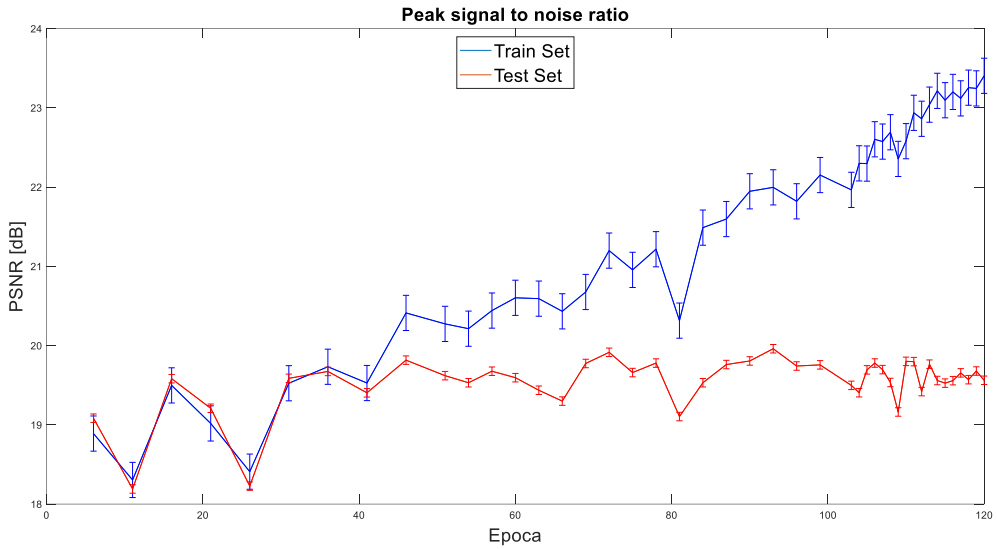

***Figura 6.1**: Andamento di PSNR calcolato sull'intera immagine all'aumentare del numero di epoche.*

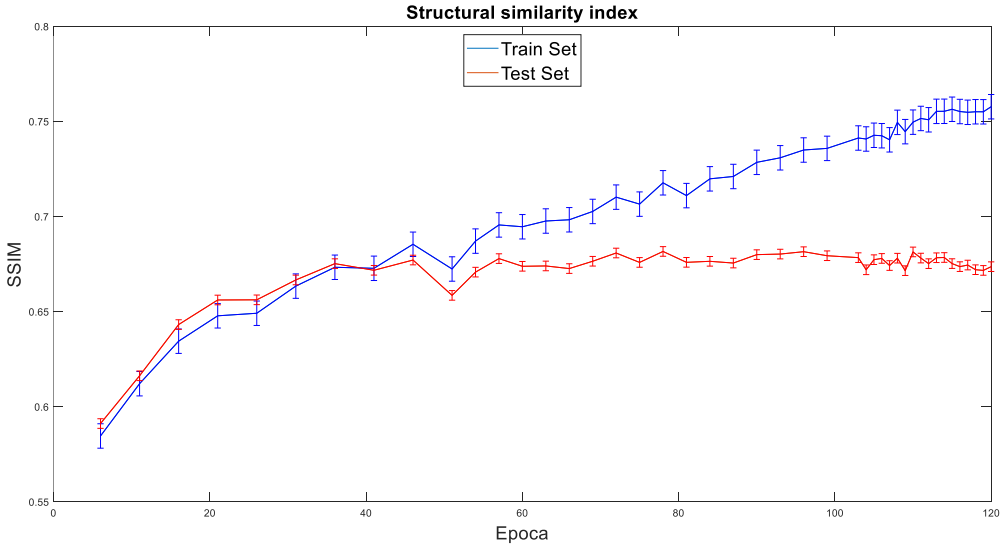

***Figura 6.2**: Andamento di SSIM calcolato sull'intera immagine all'aumentare del numero di epoche.*

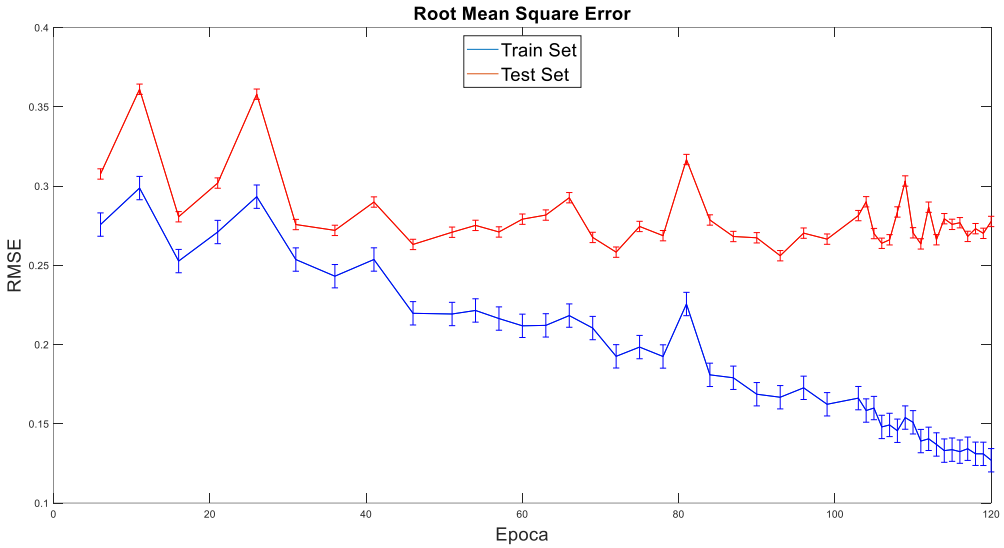

***Figura 6.3**: Andamento di RMSE calcolato sull'intera immagine all'aumentare del numero di epoche.*

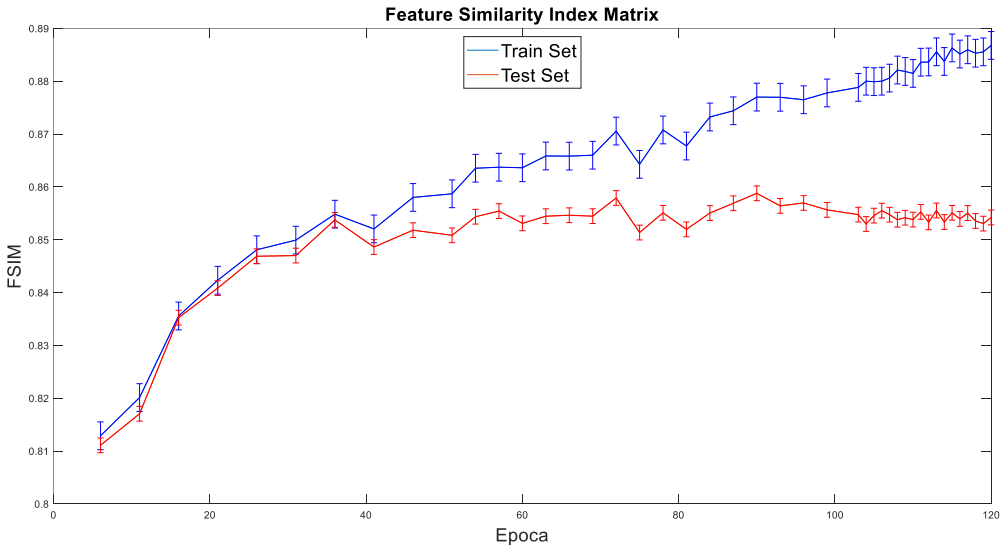

***Figura 6.4**: Andamento di FSIM calcolato sull'intera immagine all'aumentare del numero di epoche.*

Osservando i grafici si nota un trend crescente per PSNR, SSIM e FSIM e un trend decrescente per RMSE. Questi andamenti indicano che all'aumentare del numero di epoche corrisponde un miglioramento della qualità delle immagini generate e un conseguente aumento della similarità con le immagini reali.

I miglioramenti riguardo la qualità delle immagini sono evidenti analizzando la Figura 6.1 in cui si osserva un aumento del PSNR di circa 4,5 dB, mentre l'aumento della similarità è osservabile in Figura 6.2 in cui si passa da un indice di similarità pari a 0,58 in corrispondenza della prima epoca, a un indice di similarità pari a 0,75 nell'ultima epoca.
Si può infine osservare che sul TestSet i valori di deviazione standard sono minori rispetto al TrainSet, e che a partire dalla 100esima epoca l'andamento delle metriche sul TestSet resta più o meno costante.

6.2 Risultati dell'analisi di tessitura

Per valutare l'analisi di tessitura sono state analizzate le features del primo e del secondo ordine descritte nel paragrafo 4.5. Anche in questo caso le feature sono state valutate sull'intera immagine e sulle singole label (ON, ONS e bulbo oculare) e sono state confrontate con le stesse features calcolate sulla corrispondente immagine reale.
Di seguito vengono riportati i grafici che mostrano l'andamento della differenza tra le features calcolate sull'immagine reale e simulata al variare del numero di epoche.

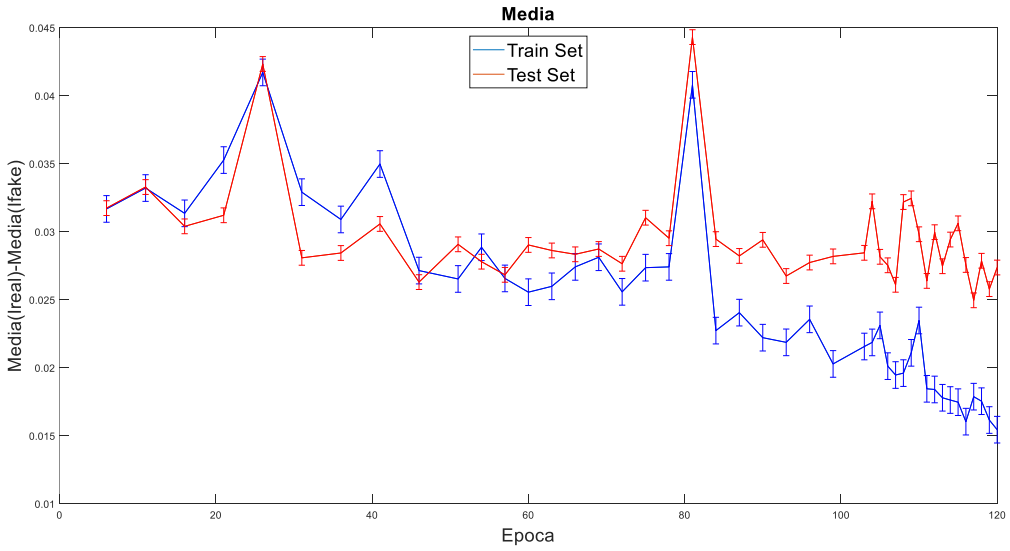

***Figura 6.5**: Differenza tra la media calcolata sull'immagine reale sull'immagine simulata all'aumentare del numero di epoche.*

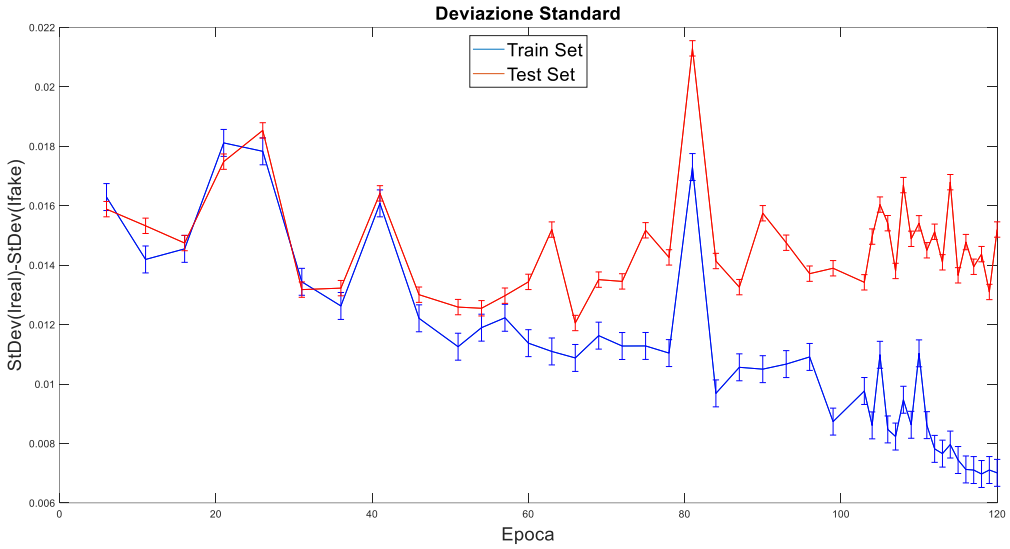

***Figura 6.6**: Differenza tra la deviazione standard calcolata sull'immagine reale sull'immagine simulata all'aumentare del numero di epoche.*

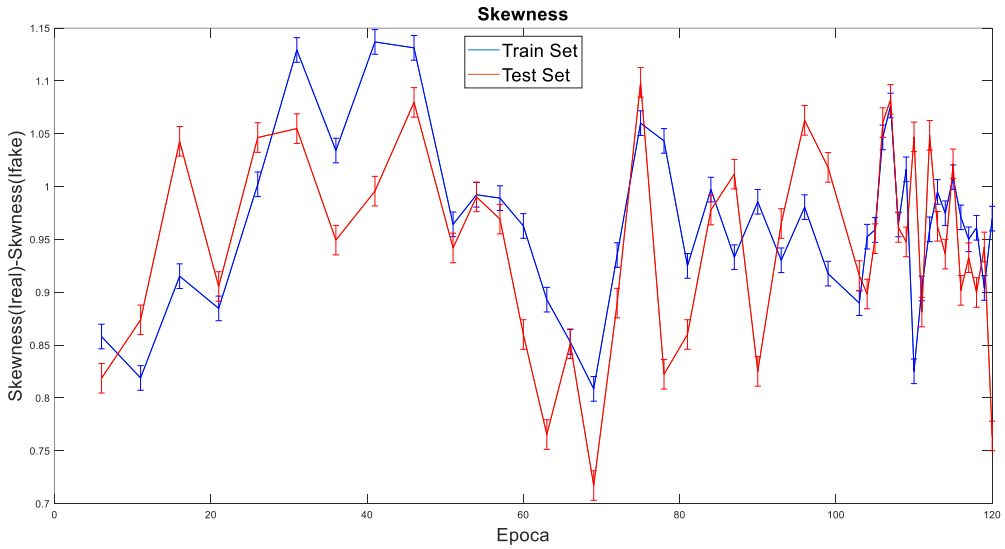

***Figura 6.7**: Differenza tra la skewness calcolata sull'immagine reale sull'immagine simulata all'aumentare del numero di epoche.*

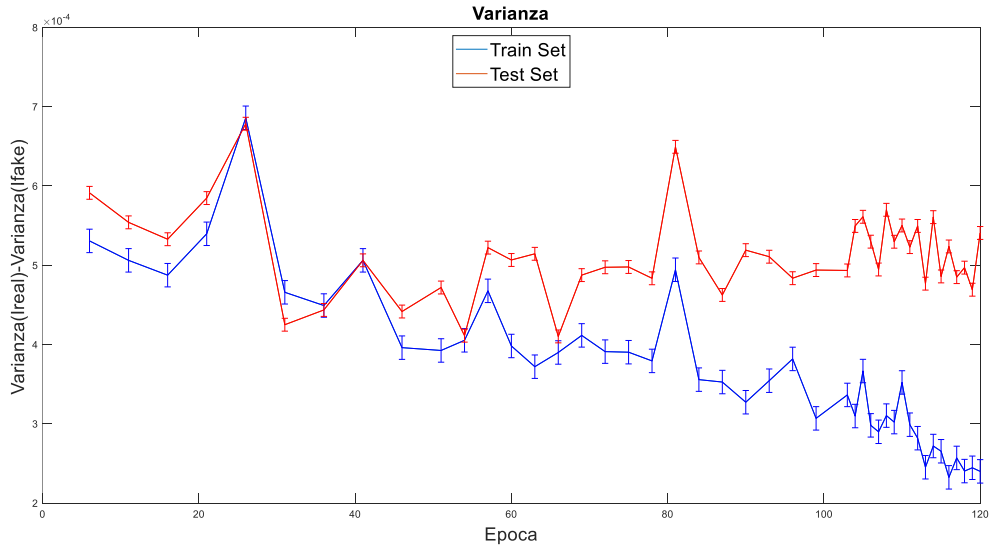

***Figura 6.8**: Differenza tra la varianza calcolata sull'immagine reale sull'immagine simulata all'aumentare del numero di epoche.*

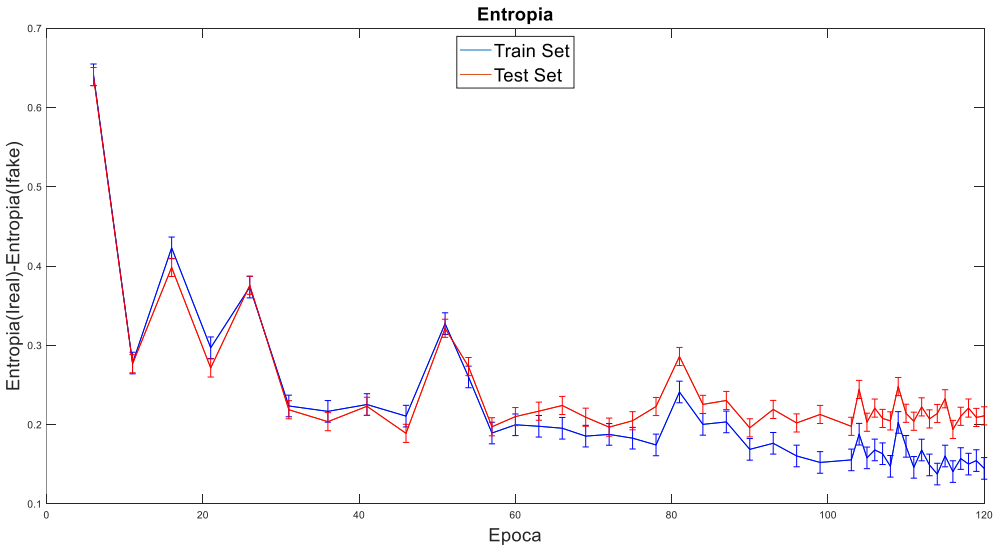

***Figura 6.9**: Differenza tra l'entropia calcolata sull'immagine reale sull'immagine simulata all'aumentare del numero di epoche.*

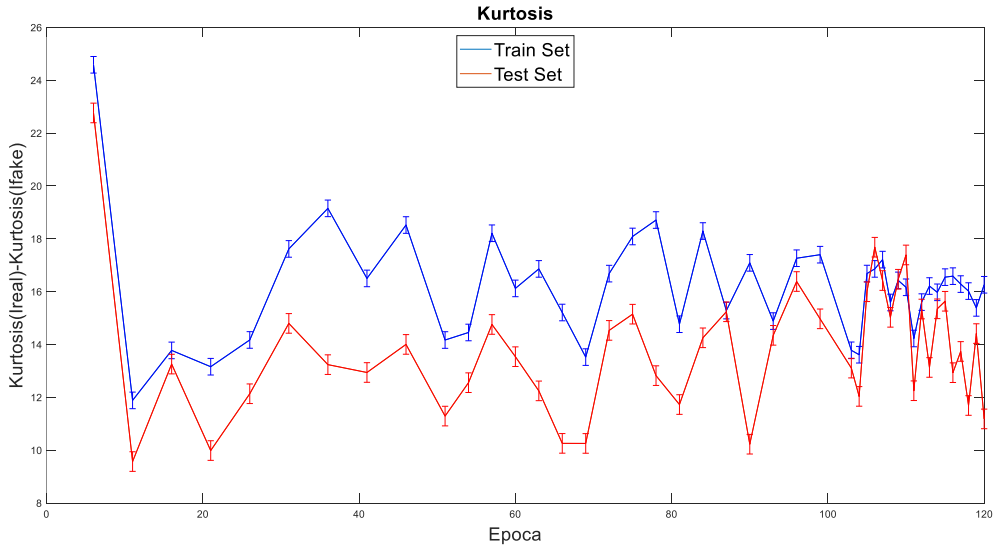

***Figura 6.10**: Differenza tra la kurtosis calcolata sull'immagine reale sull'immagine simulata all'aumentare del numero di epoche.*

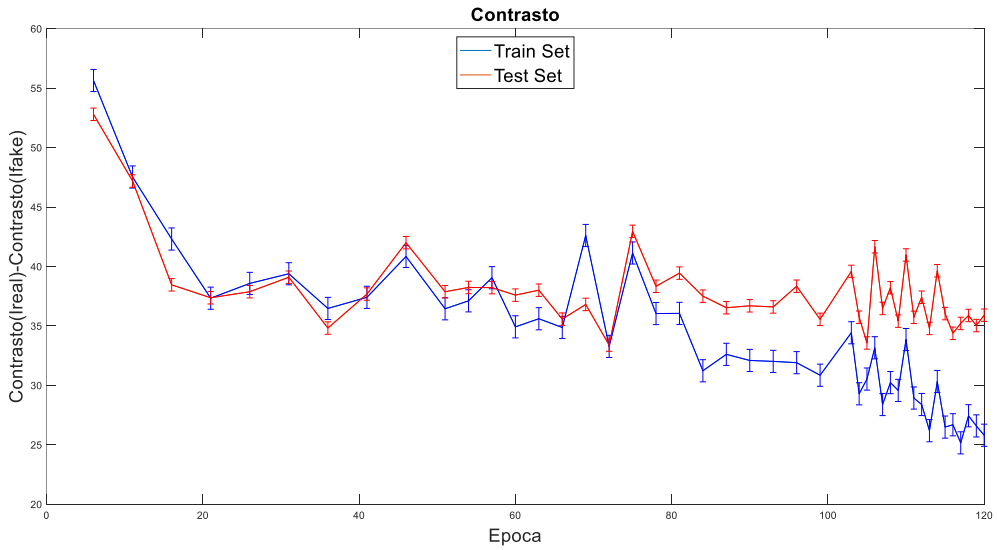

***Figura 6.11**: Differenza tra il contrasto calcolata sull'immagine reale sull'immagine simulata all'aumentare del numero di epoche.*

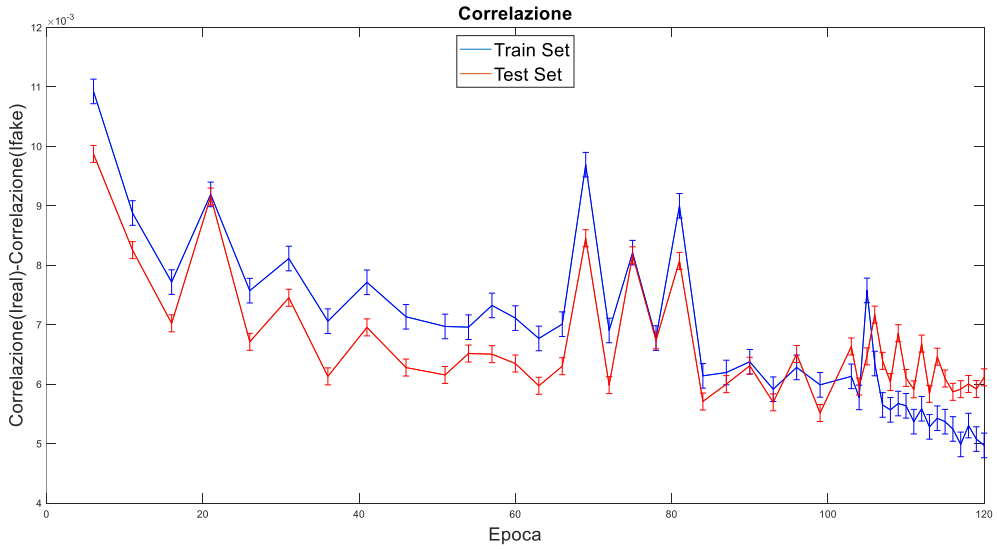

***Figura 6.12**: Differenza tra la correlazione calcolata sull'immagine reale sull'immagine simulata all'aumentare del numero di epoche.*

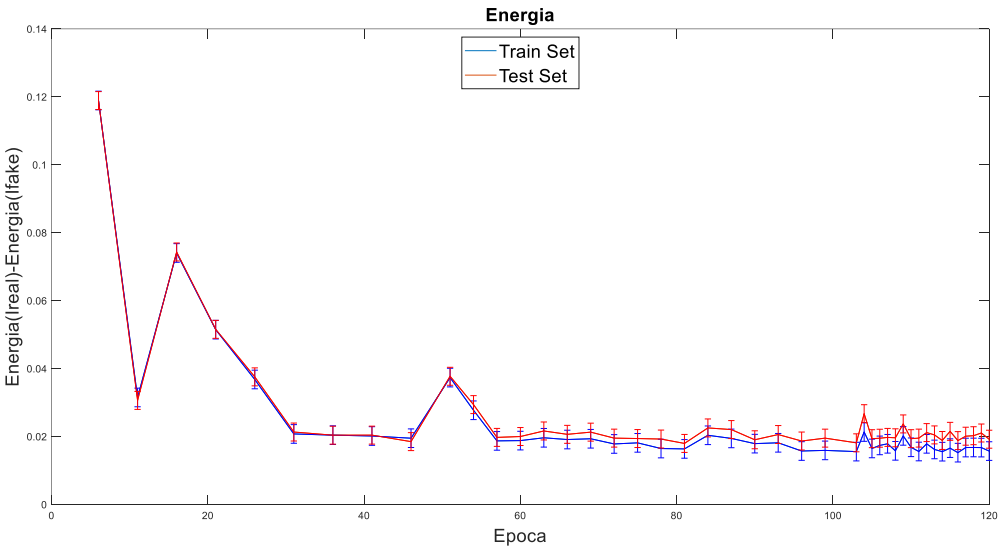

***Figura 6.13**: Differenza tra l'energia calcolata sull'immagine reale sull'immagine simulata all'aumentare del numero di epoche.*

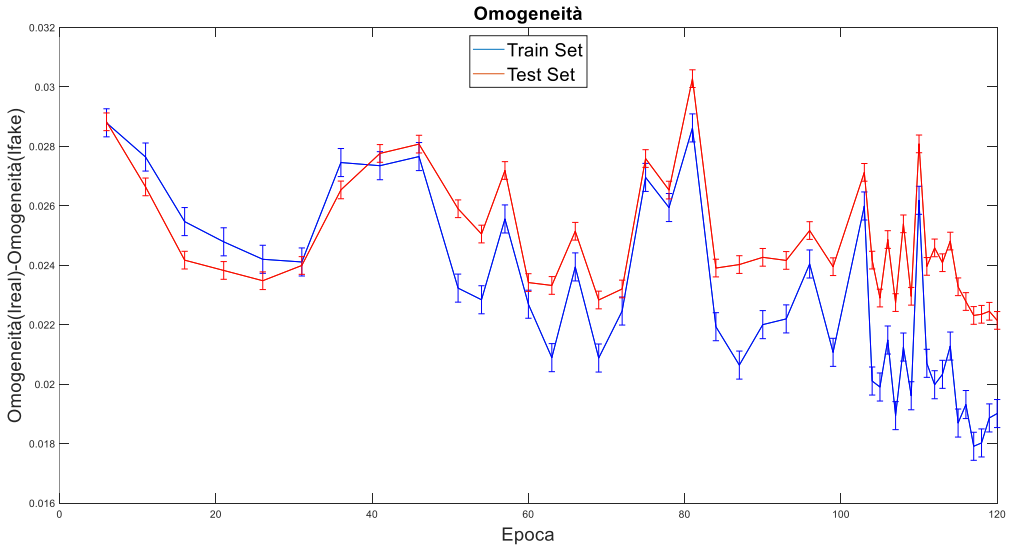

***Figura 6.14**: Differenza tra l'omogeneità calcolata sull'immagine reale sull'immagine simulata all'aumentare del numero di epoche.*

Dai grafici si può osservare un andamento decrescente per tutte le features il che è sinonimo di un miglioramento della texture delle immagini simulate che tende a diventare simile a quella reale man mano che la rete viene allenata.

Tutte le metriche e le feature di tessitura sono state valutate per andare ad individuare l'epoca migliore. È stata scelta l'epoca alla quale corrispondono i risultati migliori di metriche e feature valutate sul TestSet.
Dall'analisi è risultato che l'epoca migliore è la 93; le immagini generate in quest'epoca sono successivamente state utilizzate per l'allenamento della rete di segmentazione.

6.3 Risultati della segmentazione

Per valutare le prestazioni della U-Net utilizzata per la segmentazione delle strutture del nervo ottico, sono state valutate le metriche descritte nel paragrafo 5.3.
Sono state confrontate le segmentazioni prodotte dalla rete allenata con tre dataset differenti:

- Dataset 1: formato solo dalle immagini reali;
- Dataset 2: formato solo dalle immagini simulate;
- Dataset 3: formato da immagini reali e simulate.

	Dataset 1	*Dataset 2*	*Dataset 3*
Dice	0,696 ± 0,146	0,633 ± 0,161	0,702 ± 0,143
Hausdorff distance	22,632 ± 13,113	24,558 ± 15,405	21,815 ± 13,615

***Tabella 6.5**: Metriche di valutazione della segmentazione. Si riportano i valori medi e le rispettive deviazioni standard.*

Osservando i valori riportati in Tabella 6.5 si nota che, indipendentemente dal dataset utilizzato per allenare la rete, i valori di Dice e Haudorff distance sono molto simili.
Questo permette di affermare che la simulazione risulta utile anche se non è evidente un netto miglioramento delle prestazioni della rete andando ad aggiungere le immagini simulate al dataset utilizzato per l'allenamento.
In figura 6.15 vengono mostrati i boxplot relativi al confronto della segmentazione ottenuta dai tre dataset.

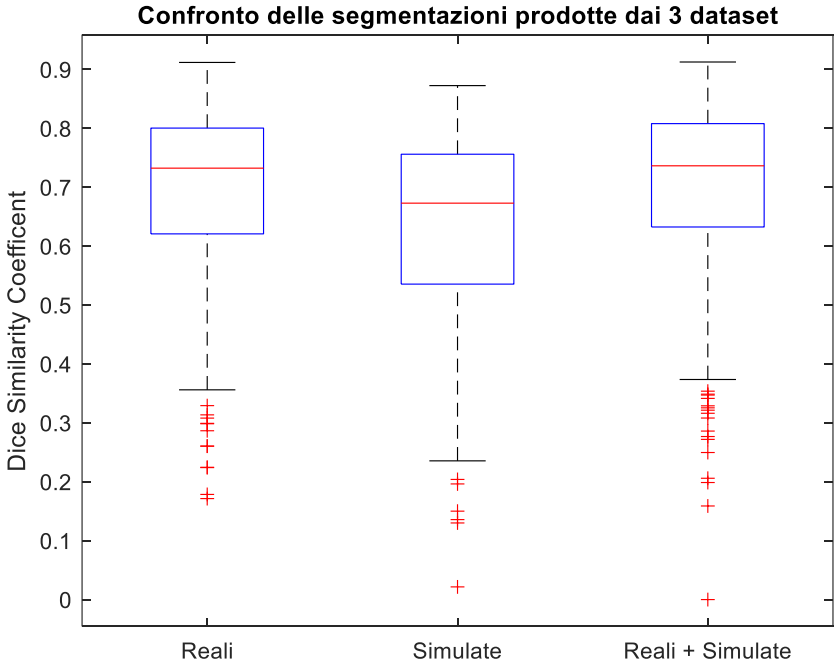

***Figura 6.15**: Confronto dei valori di Dice ottenuti dalla segmentazione prodotta dai tre dataset.*

È stato effettuato anche il Wilcoxon signed-rank test ovvero un test d'ipotesi statistico non parametrico per evidenziare eventuali differenze statistiche significative tra segmentazione ottenuta allenando la rete solo con le immagini reali e segmentazione ottenuta allenando la rete anche con le immagini simulate.
Il test ha restituito un p-valure=0,343 confermando che non ci sono differenze statistiche significative nella segmentazione allenando la rete con i due dataset differenti.

7. CONCLUSIONI

L'obiettivo di questo lavoro è stato quello di migliorare i dati da fornire in ingrasso ad una rete per la segmentazione delle strutture del nervo ottico attraverso l'utilizzo delle GAN, ed in particolare di SPADE, per la simulazione di immagini ecografiche transorbitali.
Le metriche per la valutazione della qualità dell'immagine hanno determinato un valore di PSNR di 21,4 dB ± 3,03 dB e 19,5 dB ± 2,52 dB rispettivamente per TrainSet e TestSet e un indice di somiglianza strutturale di 0,713 ± 0,096 per il TrainSet e 0,67 ± 0,092 per il TestSet. Inoltre, è stato dimostrato che all'aumentare del numero di epoche corrisponde non solo un miglioramento delle metriche ma anche un miglioramento delle feature di tessitura, il che indica che le immagini simulate tendono a diventare sempre più simili a quelle reali.
Confrontando i risultati sul Train Set e sul Test Set si nota un overfitting dei dati che potrebbe essere migliorato andando a costruire un dataset più numeroso per l'allenamento di SPADE.

La rete di segmentazione allenata con le immagini reali e poi allenata aggiungendo le immagini simulate, in fase di test ha fornito un valore di dice di 0,696 ± 0,146 e 0,702 ± 0,143 rispettivamente.
Questi valori hanno dimostrato che non c'è netto miglioramento nell'utilizzare le immagini simulate per allenare la rete di segmentazione, ma poiché i valori sono simili è comunque possibile affermare che il dataset simulato è utile per allenare la rete e che quindi SPADE è in grado di produrre immagini utili per il Data Augmentation.
L'utilizzo del deep learning come metodo di Data Augmentation risulta quindi essere un approccio promettente.

Bibliografia

[1] Austen M. Smith; Craig N. Czyz, “Neuroanatomy, Cranial Nerve 2 (Optic),” *StatPearls*, Nov. 2021, [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK507907/

[2] “Nervo ottico.” https://it.wikipedia.org/wiki/Nervo_ottico

[3] Lorenzo Crumbie MBBS and Elizabeth O. Johnson, “Optic nerve,” *Kenhub*, Jun. 22, 2022. https://www.kenhub.com/en/library/anatomy/the-optic-nerve

[4] I. Anas, “Transorbital Sonographic Measurement of Normal Optic Sheath Nerve Diameter in Nigerian Adult Population.” [Online]. Available: www.mjms.usm.my

[5] H. H. Kimberly, S. Shah, K. Marill, and V. Noble, “Correlation of optic nerve sheath diameter with direct measurement of intracranial pressure,” *Academic Emergency Medicine*, vol. 15, no. 2, pp. 201–204, Feb. 2008, doi: 10.1111/j.1553-2712.2007.00031.x.

[6] I. M. Maissan, P. J. A. C. DIrven, I. K. Haitsma, S. E. Hoeks, Di. Gommers, and R. J. Stolker, “Ultrasonographic measured optic nerve sheath diameter as an accurate and quick monitor for changes in intracranial pressure,” *J Neurosurg*, vol. 123, no. 3, pp. 743–747, Sep. 2015, doi: 10.3171/2014.10.JNS141197.

[7] Y. Yüzbaşioğlu *et al.*, “Bedside measurement of the optic nerve sheath diameter with ultrasound in cerebrovascular disorders,” *Turk J Med Sci*, vol. 48, no. 1, pp. 93–99, 2018, doi: 10.3906/sag-1707-207.

[8] C. Schroeder, A. H. Katsanos, D. Richter, G. Tsivgoulis, R. Gold, and C. Krogias, “Quantification of Optic Nerve and Sheath Diameter by Transorbital Sonography: A Systematic Review and Metanalysis,” *Journal of Neuroimaging*, vol. 30, no. 2, pp. 165–174, Mar. 2020, doi: 10.1111/jon.12691.

[9] A. Carovac, F. Smajlovic, and D. Junuzovic, “Application of Ultrasound in Medicine,” *Acta Informatica Medica*, vol. 19, no. 3, p. 168, 2011, doi: 10.5455/aim.2011.19.168-171.

[10] Giacomo Viola, Filippo Molinari, Kristen Meiburger, and Nicola Michielli, “Segmentazione e caratterizzazione automatica del nervo ottico in immagini ecografiche transorbitali per la valutazione di disturbi neurologici= Automated optic nerve segmentation and characterization in transorbital ultrasound images for neurological disorders assessment,” Politecnico di Torino, Torino, 2019.

[11] Molinari Filippo, “Dispositivi e sonde US.” Politecnico di Torino.

[12] K. M. Meiburger *et al.*, “Automatic Optic Nerve Measurement: A New Tool to Standardize Optic Nerve Assessment in Ultrasound B-Mode Images,” *Ultrasound Med Biol*, vol. 46, no. 6, pp. 1533–1544, Jun. 2020, doi: 10.1016/j.ultrasmedbio.2020.01.034.

[13] P. Lochner *et al.*, “Intra- and interobserver reliability of transorbital sonographic assessment of the optic nerve sheath diameter and optic nerve diameter in healthy adults,” *J Ultrasound*, vol. 19, no. 1, pp. 41–45, Mar. 2016, doi: 10.1007/s40477-014-0144-z.

[14] K. M. Meiburger, U. R. Acharya, and F. Molinari, “Automated localization and segmentation techniques for B-mode ultrasound images: A review,” *Computers in Biology and Medicine*, vol. 92. Elsevier Ltd, pp. 210–235, Jan. 01, 2018. doi: 10.1016/j.compbiomed.2017.11.018.

[15] Gullo Giuseppe, “DEEP LEARNING SVELATO: ECCO COME FUNZIONANO LE RETI NEURALI ARTIFICIALI,” Sep. 15, 2018. https://italiancoders.it/deep-learning-svelato-ecco-come-funzionano-le-reti-neurali-artificiali/

[16] James Liang, “An Introduction to Deep Learning,” 2018. https://towardsdatascience.com/an-introduction-to-deep-learning-af63448c122c

[17] Team I.A. Italia, “Funzioni di Attivazione nel deep learning la Guida Completa.” https://www.intelligenzaartificialeitalia.net/post/funzioni-di-attivazione-nel-deep-learning-la-guida-completa
[18] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.08458
[19] S. Sakib, A. 1#, A. Jawad, K. 2@, and H. Ahmed, “An Overview of Convolutional Neural Network: Its Architecture and Applications,” 2019, doi: 10.20944/preprints201811.0546.v4.
[20] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.08458
[21] S. Sakib, A. 1#, A. Jawad, K. 2@, and H. Ahmed, “An Overview of Convolutional Neural Network: Its Architecture and Applications,” 2019, doi: 10.20944/preprints201811.0546.v4.
[22] IBM Cloud Education, “Convolutional Neural Networks,” Oct. 20, 2020. https://www.ibm.com/cloud/learn/convolutional-neural-networks
[23] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00444-8.
[24] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artif Intell Rev*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, doi: 10.1007/s10462-020-09825-6.
[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks.” [Online]. Available: http://code.google.com/p/cuda-convnet/
[26] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556
[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” [Online]. Available: http://image-net.org/challenges/LSVRC/2015/
[28] M. Talo, “Convolutional Neural Networks for Multi-class Histopathology Image Classification.”
[29] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” May 2015, [Online]. Available: http://arxiv.org/abs/1505.04597
[30] I. J. Goodfellow *et al.*, “Generative Adversarial Networks,” Jun. 2014, [Online]. Available: http://arxiv.org/abs/1406.2661
[31] Shibsankar Das, “6 GAN Architectures You Really Should Know,” Nov. 14, 2021. https://neptune.ai/blog/6-gan-architectures
[32] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative Adversarial Networks: An Overview,” *IEEE Signal Processing Magazine*, vol. 35, no. 1. Institute of Electrical and Electronics Engineers Inc., pp. 53–65, Jan. 01, 2018. doi: 10.1109/MSP.2017.2765202.
[33] Mayank Vadsola, “The math behind GANs (Generative Adversarial Networks),” Dec. 31, 2019. https://towardsdatascience.com/the-math-behind-gans-generative-adversarial-networks-3828f3469d9c
[34] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.06434
[35] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” Mar. 2017, [Online]. Available: http://arxiv.org/abs/1703.10593
[36] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” Nov. 2016, [Online]. Available: http://arxiv.org/abs/1611.07004

[37] F. Tom and D. Sheet, “Simulating Patho-realistic Ultrasound Images using Deep Generative Networks with Adversarial Learning,” Dec. 2017, [Online]. Available: http://arxiv.org/abs/1712.07881

[38] Y. Shin, H. A. Qadir, and I. Balasingham, “Abnormal Colon Polyp Image Synthesis Using Conditional Adversarial Networks for Improved Detection Performance,” 2017, doi: 10.1109/ACCESS.2017.Doi.

[39] T. Zhang *et al.*, “SkrGAN: Sketching-rendering Unconditional Generative Adversarial Networks for Medical Image Synthesis,” Aug. 2019, [Online]. Available: http://arxiv.org/abs/1908.04346

[40] J. Liang *et al.*, “Sketch guided and progressive growing GAN for realistic and editable ultrasound image synthesis,” *Med Image Anal*, vol. 79, Jul. 2022, doi: 10.1016/j.media.2022.102461.

[41] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Trans Pattern Anal Mach Intell*, vol. PAMI-8, no. 6, pp. 679–698, 1986, doi: 10.1109/TPAMI.1986.4767851.

[42] T. Fujioka *et al.*, “Breast ultrasound image synthesis using deep convolutional generative adversarial networks,” *Diagnostics*, vol. 9, no. 4, Dec. 2019, doi: 10.3390/diagnostics9040176.

[43] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, “Semantic Image Synthesis with Spatially-Adaptive Normalization,” Mar. 2019, [Online]. Available: http://arxiv.org/abs/1903.07291

[44] Gentile Denise, Meiburger Kristen, Molinari Filippo, and Marzola Francesco, “Sviluppo di un metodo automatico basato sul deep learning per la valutazione automatica del nervo ottico in immagini ultrasonografiche,” Politecnico di Torino, Torino, 2021.

[45] “Peak Signal-to-Noise Ratio as an Image Quality Metric.” [Online]. Available: https://www.ni.com/it-it/innovations/white-papers/11/peak-signal-to-noise-ratio-as-an-image-quality-metric.html

[46] P. Datta, “All about Structural Similarity Index (SSIM),” Sep. 03, 2020. https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e

[47] “Root-mean-square deviation.” https://en.wikipedia.org/wiki/Root-mean-square_deviation

[48] R. Kumar and V. Moyal, “Visual Image Quality Assessment Technique using FSIM,” *International Journal of Computer Applications Technology and Research*, vol. 2, no. 3, pp. 250–254, 2013.

[49] C. H. Chen, L. F. Pau, P. S. P Wang, M. Tuceryan, and A. K. Jain, “Handbook of Pattern Recognition and Computer Vision, pp. 235-276 Eds CHAPTER 2.1 TEXTURE ANALYSIS,” 1993. [Online]. Available: www.worldscientific.com

[50] F. Molinari, C. Caresio, U. R. Acharya, M. R. K. Mookiah, and M. A. Minetto, “Advances in Quantitative Muscle Ultrasonography Using Texture Analysis of Ultrasound Images,” *Ultrasound Med Biol*, vol. 41, no. 9, pp. 2520–2532, Sep. 2015, doi: 10.1016/j.ultrasmedbio.2015.04.021.

[51] S. Jadon, “A survey of loss functions for semantic segmentation,” Jun. 2020, doi: 10.1109/CIBCB48159.2020.9277638.