

# RETINA Challenge

Daniele Amati, Simone Barbarino, Tomaso Sechi, Alessandro Bartelli

26 febbraio 2024

## 1 Introduzione

L'analisi del fondo oculare, chiamata anche oftalmoscopia o fundoscopia, è un esame delle strutture dell'occhio posizionate posteriormente all'iride e al cristallino. Lo scopo dell'esame è la valutazione delle dimensioni e della morfologia dei vasi sanguigni. Particolarmente interessante per la sua non invasività e per la sua velocità, spesso l'esame viene richiesto nel percorso diagnostico di diverse patologie sistemiche che possono coinvolgere le strutture interne e la vascolarizzazione dell'occhio.

La metodologia di analisi è l'utilizzo di un oftalmoscopio, strumento costituito da uno specchio inclinato, diverse lenti e da una potente fonte luminosa, che permette la visualizzazione e la cattura dell'immagine per un'analisi più approfondita. Nelle immagini sono ben visibili la fovea, la macula e il disco ottico oltre alla vascolarizzazione [Fig. 1], quest'ultima sarà il soggetto principale di questo framework, volto all'implementazione di una rete di intelligenza artificiale per la segmentazione dei vasi sanguigni.

La nostra proposta consiste in un pre-process volto ad aumentare il contrasto e la nitidezza dei vasi sanguigni, seguito da una rete di deep learning UNet3+ per la segmentazione e l'applicazione di operatori morfologici per aumentare l'accuratezza dell'estrazione dell'albero vascolare.

La corretta segmentazione è fondamentale per la valutazione di parametri come la tortuosità e la dimensione dei vasi, le cui anomalie possono fornire informazioni su eventuali patologie come malattie cardiovascolari e disturbi neurologici.

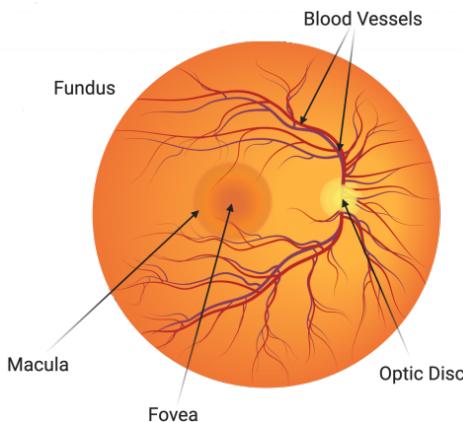


Figura 1: Schema del fondo oculare

## 2 Materiali e Metodi

### 2.1 Database

Per l'allenamento e il tuning dei parametri della rete sono state utilizzate le immagini forniteci per la challenge, in particolare 600 immagini del fondo oculare e 600 maschere manuali con risoluzione 2048x2048 codificate RGB con dinamica 8 bit per canale. Le immagini sono salvate in formato png.

Per la fase di validazione sono stati utilizzati 4 database pubblici utilizzati in svariati studi come set per il calcolo delle performance di reti. In particolare, sono stati utilizzati i set:

STARE (Structured Analysis of the Retina): è un dataset contenente 20 immagini RGB di dimensione 700x605 salvate su 8 bit per canale e 20 maschere di dimensione 700x605 in formato uint8. Le immagini sono state salvate in formato ppm. [1]

ChaseDB1: dataset composto da 28 immagini in formato RGB di dimensione 999x960 salvate su 8 bit per canale e le relative 28 maschere di uguale dimensione in formato uint8. Le immagini del fondo oculare sono salvate in formato jpg e le relative maschere in formato png. Le immagini sono state acquisite su 14 bambini in età scolare (un'immagine per occhio) e sono state salvate in formato png. [2]

DRIVE (Digital Retinal Images for Vessel Extraction): dataset composto da 20 immagini di dimensioni 584x565 in formato RGB salvate su 8 bit per canale e 20 maschere di dimensioni 584x565 in formato uint8. Le immagini sono state salvate in formato tiff. Le immagini sono ottenute da un programma di screening per retinopatia diabetica. [3]

FIVES (Fundus Image Vessel Segmentation dataset): Set composto da 800 immagini divise in training e test set. Nella nostra applicazione abbiamo usato solo il test set composto da 200 immagini a risoluzione 2048x2048 di immagini RGB salvate su 8 bit per canale e 200 maschere di dimensioni 2048x2048 in formato uint8. [4]

Tutti i database non nativamente in png sono stati convertiti in questo formato

## 2.2 Metriche di Valutazione

Le metriche scelte per valutare la bontà della segmentazione della rete sono state il Dice similarity coefficient (*DSC*), l'errore junction to junction ( $E_{J2J}$ ), l'errore junction to end point ( $E_{J2E}$ ), la recall (*Rec*), errore assoluto relativo junction to junction ( $EA_{J2J}$ ) e l'errore assoluto relativo junction to endpoint ( $EA_{J2E}$ ).

Il *DSC* è definito come:

$$DSC = \frac{2|PRED \cap GT|}{|PRED| + |GT|} \quad (1)$$

dove  $|PRED \cap GT|$  è uguale al numero di pixel risultante dall'intersezione della segmentazione della rete e della maschera manuale, e  $|PRED| + |GT|$  è la somma della cardinalità dei pixel nella segmentazione della rete e della cardinalità dei pixel nella maschera manuale. Il *DSC* è la metrica principale per valutare la bontà della segmentazione.

$E_{J2J}$  è definito dal parametro junction to junction  $N(X)_{J2J}$ , che è calcolato dallo skeleton della struttura vascolare, ed è uguale al numero di rami vascolari che collegano due giunzioni nello scheletro  $X$ .

$E_{J2J}$  è definito come:

$$E_{J2J} = \frac{N(GT)_{J2J} - N(PRED)_{J2J}}{N(GT)_{J2J}} \quad (2)$$

dove  $N(GT)_{J2J}$  è il parametro junction to junction della maschera manuale e  $N(PRED)_{J2J}$  è il parametro junction to junction della segmentazione della rete.  $E_{J2J}$  è utile per quantificare il numero di rami vascolari connessi tra loro che non sono rilevabili sulla maschera predetta.

$E_{J2E}$  è definito dal parametro junction to endpoint  $N(X)_{J2E}$ , che è calcolato dallo skeleton della struttura vascolare e rappresenta il numero di rami vascolari che cominciano in una giunzione e terminano in un punto che non è una giunzione.  $E_{J2E}$  è definito come:

$$E_{J2E} = \frac{N(GT)_{J2E} - N(PRED)_{J2E}}{N(GT)_{J2E}} \quad (3)$$

dove  $N(GT)_{J2E}$  è il parametro junction to endpoint della maschera manuale e  $N(PRED)_{J2E}$  è il parametro junction to endpoint della segmentazione della rete. L'utilizzo del parametro junction to endpoint è utile per quantificare il numero di rami vascolari che cominciano in una giunzione e

terminano in un punto che non è una giunzione che non sono rilevabili sulla maschera predetta.

La  $Rec$  definita come:

$$Rec = \frac{|GT \cap PRED|}{|GT|} \quad (4)$$

dove  $|GT \cap PRED|$  è uguale al numero di pixel risultante dall'intersezione della segmentazione della rete e della maschera manuale e  $|GT|$  è la cardinalità dei pixel appartenenti alla maschera manuale. La recall è stata usata per valutare l'eventuale sotto segmentazione.

Il  $EA_{J2J}$  e il  $EA_{J2J}$  sono definiti come:

$$EA_{J2J} = \frac{|N(GT)_{J2J}^{(i)} - N(PRED)_{J2J}^{(i)}|}{N(GT)_{J2J}^{(i)}} \quad EA_{J2E} = \frac{|N(GT)_{J2E}^{(i)} - N(PRED)_{J2E}^{(i)}|}{N(GT)_{J2E}^{(i)}} \quad (5)$$

Il  $EA_{J2J}$  e il  $EA_{J2J}$  sono utilizzati per valutare quanto distano le previsioni del modello rispetto al numero di rami junction to junction e rispetto al numero di vasi junction to endpoint dai valori reali.

### 2.3 Preprocess

Il pre-process adottato consiste in una serie di passaggi volti ad accentuare i vasi sanguigni retinici rispetto al resto dell'immagine, in modo che questi risultino più facilmente identificabili dalla rete di segmentazione successiva [Fig. 2].

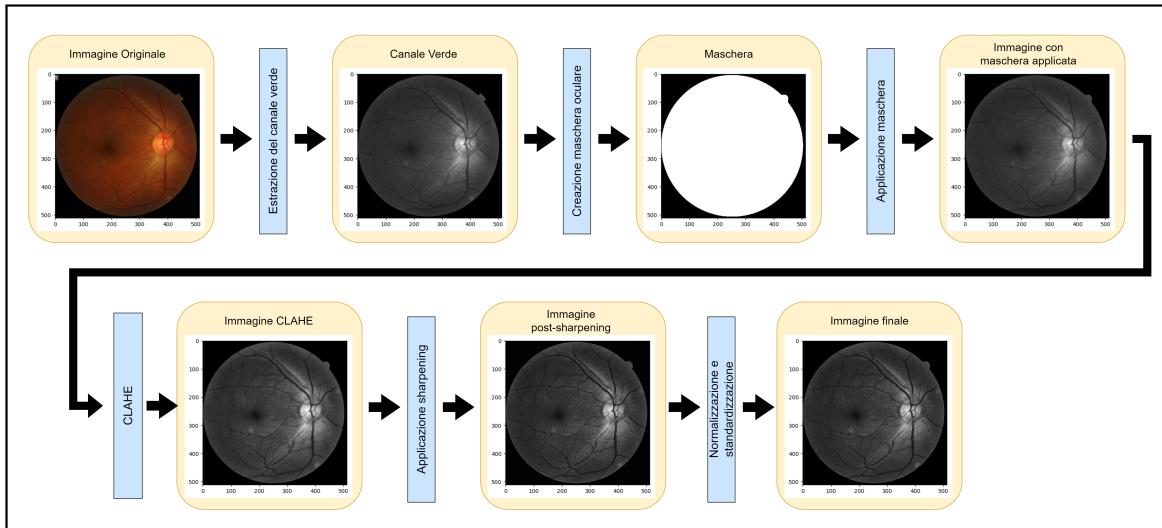


Figura 2: Pipeline completa del pre-processing. L'immagine di esempio è la 329 G del set di train

Si compone dei seguenti passi:

#### 1. Re-size delle immagini

Tutte le immagini vengono ridimensionate a 512x512 pixel. Definire una dimensione fissa per le immagini di ingresso è volto a rendere indipendenti i passaggi successivi dalla risoluzione dell'immagine originale. Inoltre, rende computazionalmente più efficiente l'uso di operatori locali.

#### 2. Estrazione del canale verde

Il canale verde è in letteratura il canale più usato per effettuare la segmentazione, questo perché il coefficiente di assorbimento dell'emoglobina ha un picco in questa regione dello spettro e di conseguenza i vasi appaiono più scuri e contrastati in questo canale [9]. Nella Fig.3 si mostra

un confronto tra i tre canali. Nei passi successivi si è pertanto deciso di utilizzare il solo canale verde.

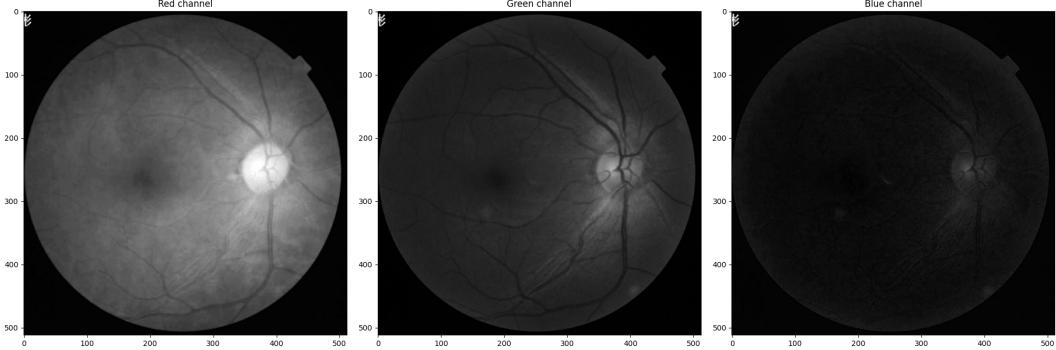


Figura 3: Immagine 329 G del canale rosso (sinistra), del canale verde (centro) e del canale blu (destra)

### 3. Creazione di una maschera oculare

L'immagine è divisibile in due aree: lo sfondo e l'occhio. Per massimizzare questa differenza si è scelto di impostare i pixel dello sfondo a 0. Per farlo si è optato per la definizione di una maschera circolare che comprendesse solo la zona oculare. Il valore di intensità dello sfondo varia da immagine a immagine, ma è tendenzialmente molto inferiore rispetto all'occhio, quindi la maschera è stata definita tramite un thresholding con soglia adattiva, dove si è definito come sfondo i pixel con il valore di intensità di grigio con il maggior numero di occorrenze. La soglia è stata definita come:

$$Threshold = \text{moda}(\text{GreenChannel}) + 2 \quad (6)$$

La maschera ottenuta è una immagine binaria avente 1 nelle zone superiori alla soglia e 0 in quella sottosoglia. Per rimuovere eventuali piccoli punti bianchi residui, si è applicato un operatore morfologico di apertura con footprint quadrato 16x16. La maschera oculare ottenuta è successivamente applicata al canale verde, ponendo a zero, tutti gli elementi non inclusi nella maschera.

### 4. Contrast Limited Adaptive Histogram Equalization (CLAHE)

È una variante dell'equalizzazione dell'istogramma adattiva dove l'amplificazione di contrasto è limitata per ridurre il problema di amplificazione del rumore. Si tratta di uno step di preprocess applicato comunemente al task di segmentazione dei vasi retinici. L'algoritmo opera in regioni delle immagini chiamate tile, proponendosi di risaltare il contrasto locale senza amplificare il rumore in parti di immagine omogenee. In ogni tile il contrasto è risaltato confrontando l'istogramma della regione di output ad una distribuzione uniforme. Le tile vicine sono infine combinate usando bilinear interpolation. Basandoci sul lavoro di James Owlera e Peter Rockett [10], abbiamo applicato l'algoritmo con tile-dimension 8x8 e threshold per contrast-limiting a 1.5.

### 5. Sharpening

Questa tecnica è stata impiegata con l'obiettivo di enfatizzare ulteriormente le repentine variazioni di intensità nell'immagine, evidenziando i margini dei vasi sanguigni e migliorando la loro definizione. Per effettuare lo sharpening, si è utilizzato il kernel:

$$\text{Kernel} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (7)$$

### 6. Standardizzazione e normalizzazione

Questo ultimo passaggio mira ad omologare il range e la distribuzione dei valori delle immagini fornite alla rete. Per prima cosa è stata effettuata una standardizzazione con metodo z-score per

ridurre la varianza tra le immagini e poi una normalizzazione con il metodo del min-max scaling in modo che le immagini siano rappresentate nello stesso range di valori.

## 2.4 Unet3+

### 2.4.1 Descrizione rete

La UNet3+ [Fig. 4] è una deep neural network progettata per la segmentazione di immagini, particolarmente utilizzata nell'ambito della segmentazione in applicazioni mediche [5]. Si basa sull'architettura UNet classica che ha un'architettura encoder-decoder in cui l'immagine, tramite una serie di convoluzioni e operazioni di max pooling, viene rappresentata da diverse feature map di dimensioni progressivamente minori nell'encoding e progressivamente maggiori tramite layer di upsampling nel decoding. Nel corso dell'upsampling, i layer attingono informazioni dalle feature map di pari dimensione estratte durante l'encoding tramite le skip connections per generare una rappresentazione più dettagliata dell'immagine. Nella UNet3+ le feature non vengono solo passate ad un layer di decoding dal layer parallelo di encoding, ma riceve informazioni dai layer di encoding di dimensione superiore ed uguale e da tutti i layer di decoding di dimensione inferiore, questi canali sono chiamati full-scale inter skip connections. La UNet3+ utilizza la full scale deep supervision, quindi produce un output per ogni scala di risoluzione spaziale del decoder che viene concatenato all'output finale e utilizzato nel calcolo della loss function in modo da avere un percorso più immediato nella correzione dei pesi per la backpropagation dei layer intermedi e far convergere più rapidamente il modello.

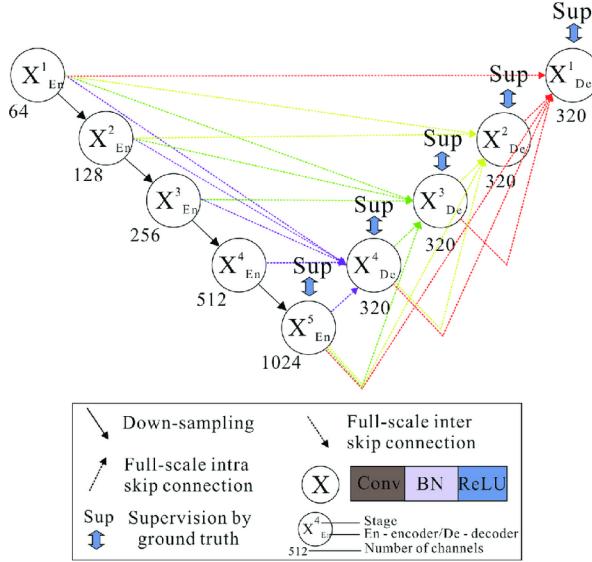


Figura 4: Schema architettura rete

### 2.4.2 Architettura rete

La rete neurale è stata implementata utilizzando il framework Keras, modificando un notebook open source disponibile su kaggle [12].

#### 1. Input Layer

All'input layer sono fornite immagini in scala di grigi con dimensioni 512x512.

#### 2. Encoder Blocks

Questi blocchi consistono in una sequenza di convoluzioni, standardizzazione del batch e attivazioni ReLU, seguite da un max pooling per ridurre la risoluzione spaziale [Fig. 5.a]. Ogni blocco raddoppia il numero di canali della feature map partendo da 64 canali e arrivando a 1024 canali nel latent pathway.

#### 3. Latent Pathway

Il latent pathway [Fig. 5.b] riduce per l'ultima volta la dimensione spaziale con kernel 3x3 e

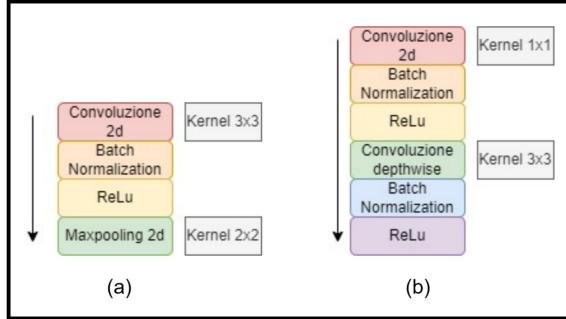


Figura 5: (a): Schema blocchi di encoder, (b): Schema layer latent space

1024 filtri e una convoluzione depthwise [6]. Con questa, ad ogni canale viene applicato un differente kernel in maniera indipendente per evitare il co-adattamento e individuare meglio le hidden features.

#### 4. Decoder Blocks

I blocchi di decoding ricostruiscono la risoluzione spaziale dell'immagine utilizzando layer di upsampling concatenati con le feature dai passaggi di encoding a risoluzione maggiore e dai layer di decoding a risoluzione inferiore tramite le skip connections. Vengono utilizzati 4 layer di decoding in cui sono integrate le skip connection in modo da avere 320 canali per ogni layer.

#### 5. Output Layers

Ogni blocco di decoding produce un output per la deep supervision. Un esempio degli output è riportato nella Fig.6. Ogni output è sottoposto a un processo di convoluzione finale seguito da attivazione ReLU e batch normalization .

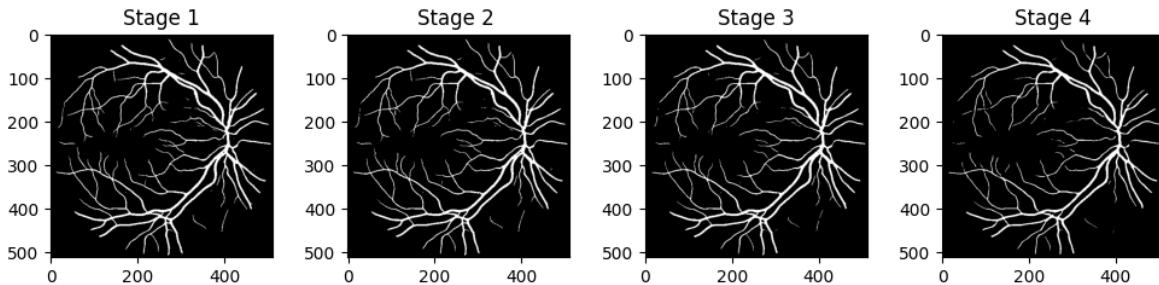


Figura 6: I 4 output della deepsupervision per l'immagine "2 A.png" del test set del FIVES

## 2.5 Costruzione training e validation set

Abbiamo suddiviso il dataset fornito in training set e validation set con proporzioni 90-10 tramite random sampling con un seed fisso per garantire la ripetibilità. Per il testing sono stati utilizzati i test set esterni precedentemente nominati (STARE, DRIVE, CHASEDB1 e FIVES) in modo da verificare come la rete si comportasse su immagini out of distribution.

## 2.6 Data Augmentation

Dalle immagini viene selezionata casualmente una patch 96x96, dopodiché attraversa tre fasi in cui possono essere applicate diverse trasformazioni con diverse probabilità, come mostrato in Fig. 7. Nella prima fase avviene una trasformazione di rotazione di 90° o ribaltamento (flip), nella seconda fase può avvenire un'inversione delle intensità dell'immagine e nella terza fase può essere aggiunto del rumore o può essere coperta una porzione dell'immagine [8].

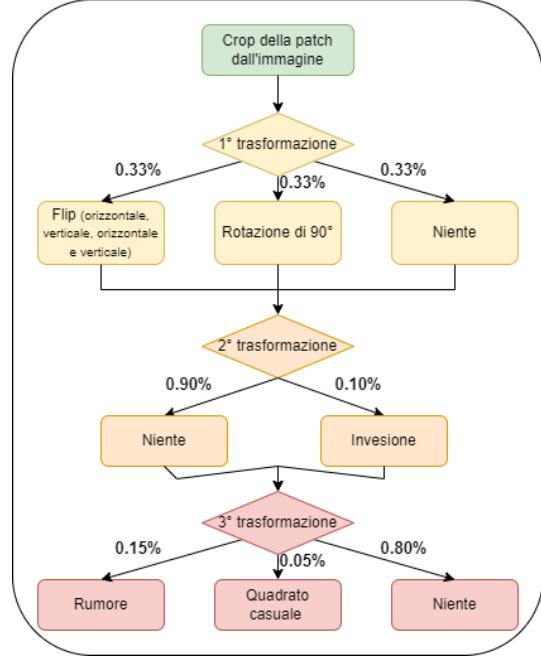


Figura 7: Schema data augmentation

## 2.7 Allenamento

La rete viene allenata su batch da 20 immagini per 30 epoche. Il numero di batch presenti viene aumentato artificialmente applicando ad ogni batch 60 volte una data augmentation randomica. La funzione di loss viene variata ogni 10 epoche con varie combinazioni delle funzioni: dice loss, iou loss, MAE e MSE [eq. 8] [7]. Nelle prime fasi dell’addestramento vengono impiegate MAE e MSE perché favoriscono la stabilità e la convergenza dell’addestramento, in seguito si passa ad una funzione di apprendimento ibrida per la transizione composta da MSE, MAE, Dice loss e Iou loss; Nelle ultime fasi dell’addestramento viene usata una funzione composta solo da Dice loss e IoU loss per enfatizzare la morfologia nella segmentazione. La rete ha un learning rate di  $10^{-3}$  e ottimizzatore ADAM.

$$\begin{aligned}
 MSE &= \frac{1}{n} \sum_{i=1}^n \left( y_{\text{true}}^{(i)} - y_{\text{pred}}^{(i)} \right)^2 & MAE &= \frac{1}{n} \sum_{i=1}^n \left| y_{\text{true}}^{(i)} - y_{\text{pred}}^{(i)} \right| \\
 IoU \ loss &= 1 - \frac{|Y_{\text{true}} \cap Y_{\text{pred}}|}{|Y_{\text{true}} \cup Y_{\text{pred}}|} & Dice \ loss &= 1 - \frac{2 |Y_{\text{true}} \cap Y_{\text{pred}}|}{|Y_{\text{true}}| + |Y_{\text{pred}}|}
 \end{aligned} \tag{8}$$

## 2.8 Post-process albero vascolare

Il post-process è volto al miglioramento dell’estrazione dell’albero vascolare dalle maschere di segmentazione prodotte dalla rete. Si è cercato di ridurre l’impatto negativo sul calcolo delle metriche j2j e j2e causato da discontinuità in vasi contigui nella maschera manuale andando ad applicare un operatore morfologico di dilatazione. Nelle vicinanze del disco ottico, la densità dei vasi è estremamente elevata e vasi distinti si trovano molto vicini tra loro. L’impiego degli operatori morfologici in questa zona ha un impatto negativo sulle metriche di tortuosità. Per mantenere la coerenza nella zona prossima al disco ottico, abbiamo deciso di attuare un post process che non la coinvolgesse.

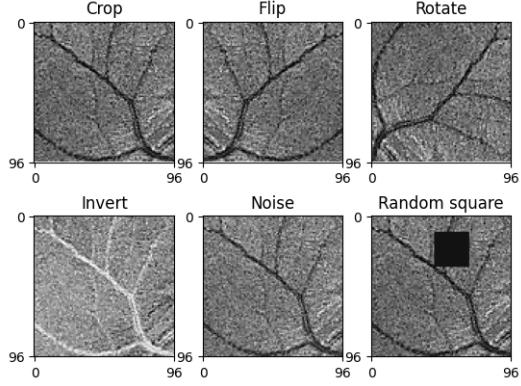


Figura 8: Esempio di data augmentation

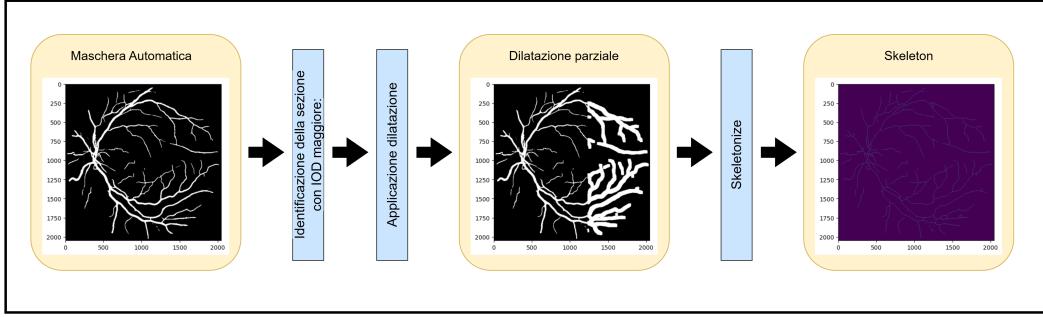


Figura 9: Pipeline del post processing

Il post process proposto si compone delle seguenti fasi [Fig. 9]:

1. **Identificazione sezione di immagine con densità ottica integrata (IOD) maggiore**  
Si calcola separatamente l'IOD della metà destra (IODdx) e metà sinistra (IODsx) della maschera. Il lato con IOD maggiore corrisponde al lato dove è presente il disco ottico.
2. **Applicazione dell'operatore morfologico di dilatazione**  
Nel lato che è risultato avere una minore IOD, solo sui 2/5 di immagine più distali rispetto al centro, si è applicato un operatore morfologico di dilatazione con footprint unitario di dimensione 8x8. I restanti 3/5 di immagine rimangono invariati.
3. **Estrazione dell'albero vascolare**  
Per estrarre l'albero vascolare si è estratto dall'immagine lo skeleton tramite metodo di Zhang [11].

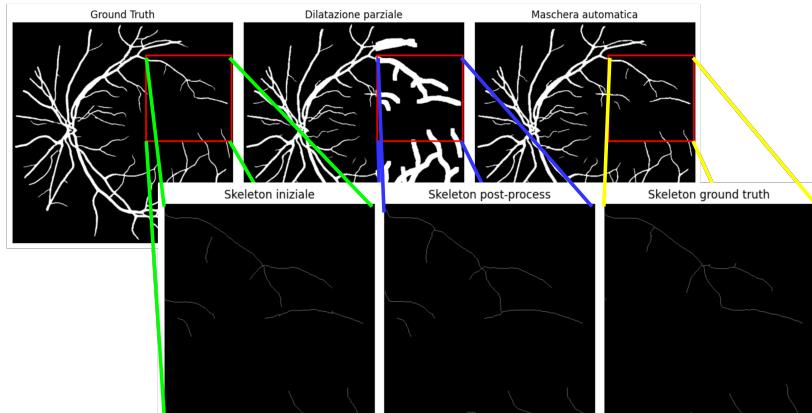


Figura 10: Differenze nell'estrazione dello skeleton senza e con post-processing

### 3 Risultati

Nella tabella 1 sono presenti la numerosità e le performance di segmentazione su training e validation della rete allenata con e senza preprocess. Con immagini senza pre-process si intendono le immagini che sono state processate solo con un resize a 512x512 e con una conversione in scala di grigio. Nella tabella 2 sono riportati i valori prima del post-processing e nella 3 i risultati a seguito del post process. Tutti i valori sono espressi come media  $\pm$  la deviazione standard.

<i>Dataset</i>	<i>Origine</i>	<i>Numerosità</i>	<i>DSC baseline</i>	<i>DSC preprocessed</i>
Training	FIVES	540	$0.870 \pm 0.094$	$0.912 \pm 0.014$
Validation	FIVES	60	$0.843 \pm 0.121$	$0.888 \pm 0.053$

Tabella 1: Construction set con e senza preprocess

<i>Dataset</i>	<i>DSC</i>	<i>Rec</i>	<i>E<sub>J2J</sub></i>	<i>EA<sub>J2J</sub></i>	<i>E<sub>J2E</sub></i>	<i>EA<sub>J2E</sub></i>
Construction	$0.892 \pm 0.058$	$0.933 \pm 0.071$	$0.361 \pm 0.169$	$0.367 \pm 0.155$	$0.388 \pm 0.196$	$0.389 \pm 0.194$
STARE	$0.746 \pm 0.173$	$0.823 \pm 0.230$	$0.000 \pm 0.365$	$0.279 \pm 0.235$	$-0.101 \pm 0.266$	$0.218 \pm 0.184$
DRIVE	$0.794 \pm 0.033$	$0.788 \pm 0.038$	$0.285 \pm 0.260$	$0.347 \pm 0.168$	$0.217 \pm 0.301$	$0.321 \pm 0.187$
CHASE_DB1	$0.777 \pm 0.039$	$0.931 \pm 0.017$	$0.207 \pm 0.159$	$0.232 \pm 0.119$	$0.224 \pm 0.130$	$0.228 \pm 0.122$
FIVES	$0.867 \pm 0.096$	$0.909 \pm 0.111$	$0.391 \pm 0.217$	$0.393 \pm 0.213$	$0.342 \pm 0.188$	$0.342 \pm 0.187$

Tabella 2: Performance della rete pre post processing

<i>Dataset</i>	<i>DSC</i>	<i>Rec</i>	<i>E<sub>J2J</sub></i>	<i>EA<sub>J2J</sub></i>	<i>E<sub>J2E</sub></i>	<i>EA<sub>J2E</sub></i>
Construction	$0.892 \pm 0.058$	$0.933 \pm 0.071$	$0.310 \pm 0.256$	$0.325 \pm 0.207$	$0.294 \pm 0.201$	$0.321 \pm 0.207$
STARE	$0.746 \pm 0.173$	$0.823 \pm 0.230$	$-0.171 \pm 0.291$	$0.348 \pm 0.284$	$-0.123 \pm 0.432$	$0.276 \pm 0.194$
DRIVE	$0.794 \pm 0.033$	$0.788 \pm 0.038$	$0.219 \pm 0.283$	$0.301 \pm 0.193$	$0.150 \pm 0.325$	$0.286 \pm 0.216$
CHASE_DB1	$0.777 \pm 0.039$	$0.931 \pm 0.017$	$0.063 \pm 0.168$	$0.147 \pm 0.103$	$0.147 \pm 0.127$	$0.166 \pm 0.100$
FIVES	$0.867 \pm 0.096$	$0.909 \pm 0.111$	$0.310 \pm 0.256$	$0.326 \pm 0.191$	$0.294 \pm 0.201$	$0.301 \pm 0.191$

Tabella 3: Performance della rete dopo il post processing

Il framework ha conseguito un *DSC* delle maschere di segmentazione superiore al 75% su tutti i set di test adottati, con un valore superiore all’85% nel caso del test set FIVES. Quest’ultimo ha conseguito prestazioni comparabili a quelle del construction set, probabilmente per via della similarità tra questi set. Il set di test con le prestazioni più basse è risultato essere lo STARE, caratterizzato da un taglio dell’immagine e proporzioni differenti rispetto agli altri. Inoltre, le annotazioni in alcuni casi presentano delle differenze, come l’assenza del groviglio di vasi tipico del disco oculare. La recall consente di valutare la tendenza alla sottosegmentazione, è sempre superiore al 75% questo indica che la rete genera pochi falsi negativi. La recall concorre alla precision nel calcolo del *DSC*, nel caso del CHASEDB1 possiamo notare che per un *DSC* molto basso abbiamo una recall anormalmente elevata, questo è indice di una precision bassa, cioè un alto numero di falsi positivi per il set CHASEDB1.

L’utilizzo simultaneo dell’errore relativo e dell’errore assoluto relativo consente di valutare sia la sovrastima che la sottostima dei diversi tipi di vasi, tenendo conto dell’entità dell’errore generale, che potrebbe essere mediato fino ad arrivare a zero. Ad esempio, lo STARE e il DRIVE mostrano errori percentuali inferiori del 10% rispetto all’errore assoluto relativo.

L’errore percentuale indica che il framework tende a sovrastimare il numero dei vasi j2e e j2j rispetto alla ground truth. L’adozione dell’operatore morfologico di dilatazione prima dell’estrazione dello scheletro con il metodo di Zhang [11] riduce mediamente l’errore assoluto j2j del 4% e del j2e del 3%. È importante notare che lo STARE è l’unico set di test in cui l’applicazione del post-process ha determinato un peggioramento delle metriche di tortuosità.

## 4 Discussione

La Rete ha delle buone prestazioni di baseline anche senza un pre-process, tuttavia è stato possibile riscontrare un miglioramento di 4 punti percentuale di DSC sul validation con il pre-process da noi adottato. Data l’elevata correttezza della segmentazione ottenuta, la maggior parte dei metodi di post-process convenzionali come gli operatori morfologici, tendono a peggiorare complessivamente le performance se applicati indiscriminatamente su tutta l’immagine.

Ad ispezione visiva, e confermato dalla scarsa recall sul CHASEDB1 set, uno dei punti deboli del nostro framework sembra essere il riconoscimento dei vasi molto sottili, soprattutto nella loro parte terminale quando tendono ad avere forme più tortuose, questa difficoltà si manifesta particolarmente sui risultati dei set CHASEDB1 e DRIVE. Quando i vasi vengono riconosciuti risultano essere più

spessi dell'annotazione aumentando i falsi positivi [Fig. 11]. Una delle cause di questa problematica potrebbe essere la variabilità tra le annotazioni, in quanto nel training set non sono state utilizzate immagini con vasi sottili quanto quelli presenti nei test. Una divisione in classi di vasi sottili e in classi di vasi spessi prima del training, potrebbe indurre un miglioramento su questa problematica.

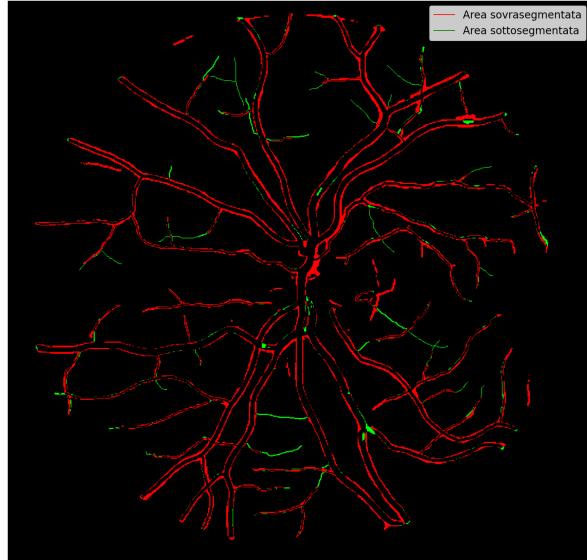


Figura 11: Immagine Image 10L del database CHASEDB1 con contrassegnati in rosso le aree sovra-segmentate e in verde le aree sottosegmentate

Una caratteristica positiva del nostro framework risiede nella sua velocità e semplicità d'uso. Con un approccio che si avvale di poche operazioni e sfrutta il ridimensionamento delle immagini a una risoluzione di 512x512, si ottiene una notevole riduzione dei tempi di elaborazione, senza compromettere l'efficacia delle prestazioni. Per un'immagine 2048x2048 sulla piattaforma Google Colab il preprocess impiegherebbe una media di 40 secondi(s) ridotto a circa 4.0 s di media per l'immagine dopo il resize. Il post-process dell'albero vascolare impiega nel caso di immagini 2048x2048 circa 10 s, molto ridotti per immagini minori.

La rete tende in alcune occasioni a generare dei falsi positivi nella parte esterna dell'immagine, in corrispondenza della cornice circolare attorno alla retina, l'impatto di questi artefatti è trascurabile sulle metriche ed è stato limitato dal pre-process. È complesso applicare un post-process che limiti questo fenomeno, una soluzione possibile potrebbe stabilire un cerchio a priori che isolà la retina, ma nel caso di immagini con formato diverso come il test-set STARE causa un calo delle prestazioni.

Il Construction set usato è caratterizzato da una serie di immagini con diverso rumore, artefatti e luminosità, questo contribuisce alla robustezza della rete nel processare anche immagini molto diverse tra loro. Questa caratteristica è ulteriormente enfatizzata da una data augmentation molto varia.

Il framework realizzato ha generalizzato bene su set molto diversi, in termini di acquisizione e risoluzione, quindi risulta essere molto robusto su immagini out of distribution. Il framework è molto rapido e fedele nella segmentazione, e il metodo per l'estrazione dell'albero vascolare consente una fedele quantificazione delle metriche di tortuosità.

## Riferimenti bibliografici

- [1] A. Hoover et al., "Locating Blood Vessels in Retinal Images by Piece-wise Threhsold Probing of a Matched Filter Response", IEEE Transactions on Medical Imaging , vol. 19 no. 3, pp. 203-210, March 2000
- [2] M. M. Fraz et al., "An Ensemble Classification-Based Approach Applied to Retinal Blood Vessel Segmentation", in IEEE Transactions on Biomedical Engineering, vol. 59, no. 9, pp. 2538-2548, Sept. 2012

- [3] J. Staal et al., "Ridge-based vessel segmentation in color images of the retina", in IEEE Transactions on Medical Imaging, vol. 23, no. 4, pp. 501-509, April 2004
- [4] K. Jin et al., "FIVES: A Fundus Image Dataset for Artificial Intelligence based Vessel Segmentation", Sci Data 9, 475 (2022)
- [5] H. Huang et al., "UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation", arXiv: 2004.08790, Apr. 2020
- [6] Z. Y. Khan et al., "CNN with depthwise separable convolutions and combined kernels for rating prediction", Expert Systems with Applications, Volume 170, 2021
- [7] M. C. Dickson et al., "Hybridised Loss Functions for Improved Neural Network Generalisation", arXiv: 2204.12244, Apr. 2022
- [8] R. Takahashi et al., "Data Augmentation Using Random Image Cropping and Patching for Deep CNNs," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 9, pp. 2917-2931, Sept. 2020
- [9] T. Walter et al., "Automatic detection of microaneurysms in color fundus images", Medical Image Analysis, Volume 11, Issue 6, Pages 555-566, 2007
- [10] J. Owler et al., "Influence of background preprocessing on the performance of deep learning retinal vessel detection", J Med Imaging (Bellingham), Nov. 2021
- [11] T. Y. Zhang and Ching Y. Suen. A fast parallel algorithm for thinning digital patterns. Communications of the ACM, 27(3):236–239, 1984.
- [12] GROLOCH Unet3+ with deep supervision: 83% F1 score. <https://www.kaggle.com/code/groloch/unet3-with-deep-supervision-83-f1-score/output>