

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Biomedica

Elaborazione di Immagini Mediche



Nuclei & Cytoplasm segmentation

279565

Brenciani Giacomo

292207

Filippi Edoardo

279241

Marini Cecilia

282579

Mengoni Alessia

A.A. 2021/2022

SOMMARIO

Introduzione.....	3
Organizzazione dei dati.....	3
Caricamento dei dati.....	4
Preprocessing.....	4
Data augmentation	5
Implementazione rete neurale	5
Gestione delle sovrapposizioni	6
Post-processing.....	7
Segmentazione delle singole cellule	7
Algoritmo di watershed	8
Costruzione maschere e verifica cellule individuate	9
Metriche di valutazione	9
Valutazione performance: Fase 1	10
Valutazione performance: Fase 2	11
Valutazione performance: Fase 3	11
Conclusioni e performance finali	12

INTRODUZIONE

L'algoritmo prevede la segmentazione automatica di plasmacellule nella condizione patologica di mieloma multiplo.

Il **mieloma multiplo** è un tumore che colpisce alcune cellule del midollo osseo, coinvolte nella produzione di anticorpi necessari a combattere le infezioni, le **plasmacellule**. In condizioni fisiologiche plasmacellule rappresentano meno dell'1% delle cellule presenti nel midollo osseo, ma nel mieloma multiplo se ne ha una produzione incontrollata (plasmacellule tumorali). Questa sovrapproduzione può portare allo sviluppo di patologie quali **anemia**, **leucopenia**, **piastrinopenia**¹, dovute alla ridotta produzione cellule emopoietiche, e ad un generale aumento della fragilità ossea, unitamente ad un abbassamento della difesa immunitaria². Dal momento che la diagnosi avviene di norma previa biopsia (di qui la produzione di immagini citologiche), risulta essere vantaggioso lo sviluppo di un algoritmo che possa assistere l'anatomopatologo nella valutazione e stadiazione della neoplasia.

Questo lavoro ha come obiettivo la realizzazione di un **sistema di segmentazione automatico** delle plasmacellule tumorali a partire da un Dataset di immagini e di maschere manuali già fornite. La segmentazione prevede non solo la generazione di maschere per isolare singolarmente le cellule anomale, ma anche la divisione della segmentazione cellulare nelle sue componenti interne, citoplasma e nucleo.

ORGANIZZAZIONE DEI DATI

Il Dataset fornito è composto da un totale di 450 immagini in formato .tiff così suddivise: 300 immagini per il **Training Set**, 50 immagini per il **Validation Set** e 100 immagini per il **Test Set**. Per il Training e Validation Set sono state inoltre fornite le segmentazioni manuali, effettuate da un patologo esperto. Le immagini e le maschere di input possono avere una dimensione pari a 1536x2040x3 pixels o 1920x2560x3; affinché possano essere trattate in modo equivalente all'interno dell'algoritmo queste saranno sottoposte a un primo resize nella fase di caricamento, per essere poi riportate nella dimensione originale dopo la segmentazione. Le immagini fornite sono di tipo RGB, mentre le maschere delle singole cellule sono delle immagini multilayer a canale singolo, dove il numero dei layer è pari al numero delle cellule segmentate, ed in ogni layer le componenti di interesse rispettano la seguente codifica colore a 8 bit: nuclei [255], citoplasmi [128] e sfondo [0].

Pipeline di riferimento

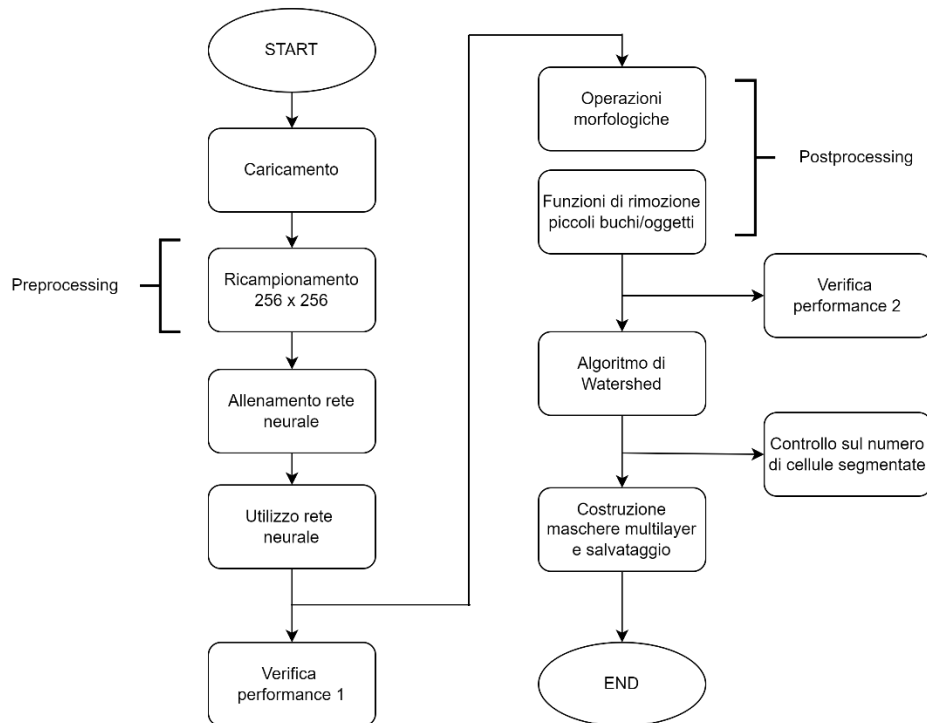


Figure 1: Diagramma di flusso dell'algoritmo di segmentazione

¹ <https://www.humanitas.it/malattie/mieloma-multiplo/>

² <https://www.msmanuals.com/it-it/professionale/ematologia-e-oncologia/disturbi-plasmacellulari/mieloma-multiplo>

CARICAMENTO DEI DATI

Inizialmente le immagini e le maschere sono state caricate mantenendo conservata la proporzionalità tra larghezza e altezza dell'immagine, ma riducendone le dimensioni (resize a 512 x 680). I layer relativi a cellule differenti sono stati sovrapposti per creare la maschera completa per ogni immagine fornita, sia per il Training Set che per il Validation Set. Le immagini sono state caricate in formato uint8, mentre le maschere in formato float32. Entrambe sono state poi visualizzate per verificare l'effettiva correttezza della corrispondenza.

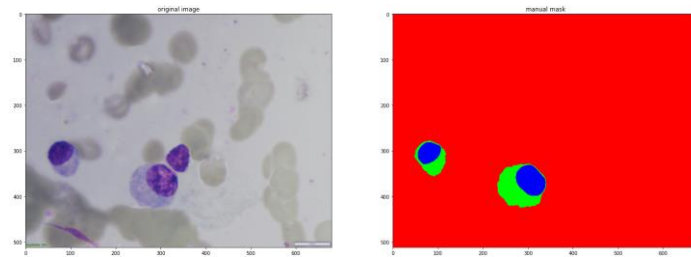


Figure 2: Esempio di immagine citologica estratta dal Validation Set e corrispettiva maschera

Nella maggior parte dei casi le annotazioni manuali risultano corrispondere alle rispettive immagini; sono state però individuate 2 maschere manuali (la numero 17 e 18) che non risultano corrispondere alle immagini fornite (vengono segmentate cellule che non sono presenti nell'immagine). Tuttavia, non si esclude la presenza di altri outliers.

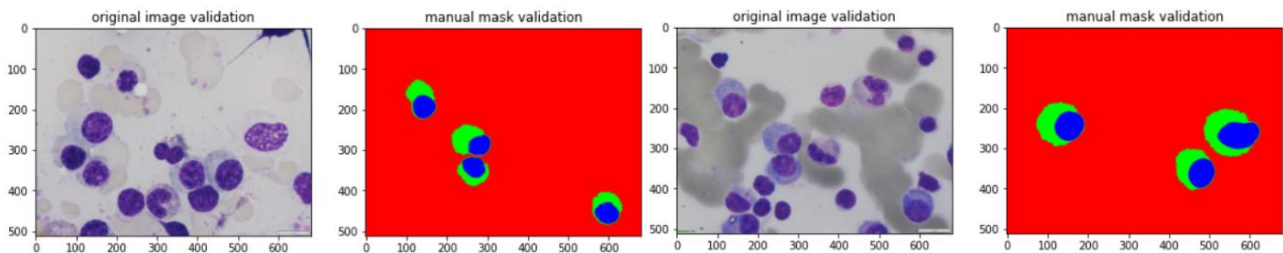


Figure 3: Illustrazione delle immagini citate del Validation Set non appaiate

Le immagini sono state comunque considerate nel calcolo delle performance nel Validation Set e naturalmente entrambe incidono sui risultati ottenuti con le metriche di validazione, così come la rete stessa durante l'apprendimento.

Dal confronto tra le maschere fornite e le immagini, non è stato identificato a primo impatto un criterio di selezione delle plasmacellule tumorali; caratteristiche fisiche, come forma, dimensioni o caratteristiche colorimetriche, prese singolarmente, non risultano distintive della plasmacellula tumorale, se non agli occhi di un anatomopatologo esperto. Di conseguenza la soluzione più immediata al problema di segmentazione è risultata l'implementazione di una rete neurale, in grado di estrarre in maniera automatica le features necessarie all'individuazione delle plasmacellule tumorali.

PREPROCESSING

Le immagini sono state sottoposte ad un ridimensionamento iniziale di 256x256 al fine di:

- ridurre i tempi computazionali dell'algoritmo di segmentazione
- uniformare il formato delle immagini in ingresso
- ottenere immagini di input quadrate da fornire alla rete neurale
- evitare di saturare la RAM che Colab rende disponibile.

Questo è stato fatto sia per le maschere che per le immagini attraverso una funzione apposita. Sono state create delle matrici separate denominate 'processed', mentre non sono state modificate le matrici originali per permettere di fare eventuali confronti e controlli.

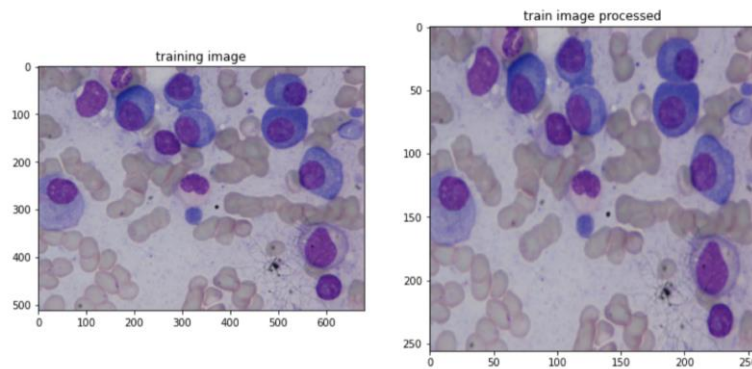


Figure 4: Immagine 27 Training, prima e dopo il preprocessing

DATA AUGMENTATION

Dato che è stato scelto di implementare una rete neurale sono necessarie molte immagini, quindi si è deciso di fare del data augmentation. La libreria utilizzata per implementare la rete (Keras) offre la possibilità di farlo in tempo reale grazie a ImageDataGenerator, che permette di fare data augmentation senza occupare memoria con le nuove immagini. Le operazioni fatte sulle immagini sono le seguenti:

- **Rotazione:** il modello viene addestrato per riconoscere un oggetto indipendentemente dall'orientazione;
- **Flipping:** le immagini vengono specchiate sia verticalmente che orizzontalmente;
- **Shift:** le immagini potrebbero contenere l'oggetto non esattamente al centro dell'immagine. Ciò che si fa è spostare i pixel sia orizzontalmente che verticalmente;
- **Fill mode:** nel momento in cui viene applicata una rotazione dell'immagine, alcuni pixel tendono a spostarsi, lasciando delle zone vuote. Per tal motivo, queste zone vengono riempite (modalità **reflect**).

IMPLEMENTAZIONE RETE NEURALE

In seguito al pre-processing, si è scelto quindi di implementare una rete neurale con lo scopo di segmentare le cellule tumorali ed avere quindi in uscita la suddivisione dei pixel in tre differenti classi (0=sfondo, 1=citoplasma, 2= nucleo). Sulla base della letteratura riguardo la segmentazione di cellule in immagini istologiche³, si è deciso di testare 3 differenti reti (Densenet121, Resnet34 e InceptionV3), scegliendo il modello migliore sulla base dell'output della rete. Per scegliere la rete da utilizzare e valutarne le prestazioni unicamente sulla base dell'output, non si effettua alcun tipo di post-processing. Le prestazioni dell'output delle reti testate saranno valutate sulla base delle metriche descritte successivamente. Di seguito vengono riportati i parametri scelti per effettuare due prove per ogni rete, che si distinguono per il numero di epoche e batch size.

	Modello	Pesi encoder	Encoder freeze	Decoder block type	Funzione di attivazione	Algoritmo di ottimizzazione	Funzione di loss	Batch size	Numero di epoche	Early stopping
Prova 1	Densenet121 Resnet34 InceptionV3	Imagenet	True	upsampling	softmax	Adam	binary cross entropy	8	35	No
Prova 2	Densenet121 Resnet34 InceptionV4	Imagenet	True	upsampling	softmax	Adam	binary cross entropy	16	25	No

Tabella 1. Reti e parametri impostati per la definizione dei modelli

Breve descrizione dei parametri:

- **Pesi encoder:** i pesi *imagenet* corrispondono ad un ampio database di immagini contenenti più di 14 milioni di immagini riguardanti diversi campi; tra questi, anche il campo relativo alle immagini mediche.
- **Encoder freeze:** settando "TRUE" facciamo in modo che i pesi non vengano modificati durante l'allenamento della rete per arrivare ad un'ottimizzazione.
- **Decoder block type:** si è selezionato il parametro di upsampling (col quale vengono aumentate le dimensioni dell'input (sovracampionamento)).

³ Gehlot S, Gupta A, Gupta R. A CNN-based unified framework utilizing projection loss in unison with label noise handling for multiple Myeloma cancer diagnosis. Med Image Anal. 2021 Aug;72:102099. doi: 10.1016/j.media.2021.102099. Epub 2021 May 13. PMID: 34098240.

- **Funzione di attivazione:** permettono di valutare l'output della rete, mappando l'ingresso in uscita. Utilizzando funzioni di attivazione **lineari**⁴ su reti neurali profonde (come tutte quelle utilizzate), indipendentemente dalla profondità della rete, sarà equivalente ad usare una semplice rete neurale senza layer nascosti. In tal modo si perde tutta la complessità della rete. Si focalizza quindi l'attenzione sull'utilizzo della funzione di attivazione **softmax** in entrambe le prove.
- **Algoritmo di ottimizzazione:** è stato scelto l'algoritmo di ottimizzazione di Adam.
- **Funzione di loss:** la funzione di loss scelta è quella che valuta la crossentropia binaria.
- **Batch size:** La scelta della batch fa riferimento alla ricerca dell'ottimo globale della rete. Tale parametro è legato all'algoritmo della discesa del gradiente: un batch size piccolo (ad esempio 8) indica spostamenti più piccoli che potrebbero garantire una maggiore precisione ma maggior tempo computazionale per arrivare ad un ottimo globale. Un batch size più grande (ad esempio 16) potrebbe far raggiungere la convergenza più facilmente ma il rischio è che l'ottimo trovato sia locale e non globale. Per questi motivi le due prove realizzate vedono le combinazioni di un batch size piccolo (8) con un numero di epoche rilevante (35) e un batch size leggermente maggiore (16) unito ad un numero di epoche minore (25).
- **Epoche:** il numero di epoche è stato deciso sulle basi delle considerazioni fatte sopra.
- **Early stopping:** si è deciso di non inserire early stopping nell'allenamento della rete.

Si sono così utilizzati 3 modelli di reti ma si descrive accuratamente soltanto la rete scelta (vedere "Valutazione performance – fase1"):

INCEPTIONV3⁵: sono reti pre-allenate di tipo convoluzionale costituite da 48 layers. Rispetto alle VGGNet, le Inceptionv3 risultano essere più efficienti dal punto di vista computazionale, sia in termini di numero di parametri generati dalla rete che di costo sostenuto (in termini di memoria e altre risorse). Di seguito è riportato lo schema di un **modulo Inception**:

1. **CONVOLUZIONI FATTORIZZATE:** ciò aiuta a ridurre l'efficienza computazionale, riducendo i parametri utilizzati dalla rete;
2. **CONVOLUZIONI PIU' PICCOLE:** permettono di realizzare un addestramento più veloce della rete; quindi la convoluzione 5x5 classica viene rimpiazzata con 2 convoluzioni 3x3 (si passa da 25 a 18 parametri);
3. **CONVOLUZIONI ASIMMETRICHE:** la classica convoluzione 3x3 è rimpiazzata da una convoluzione 1x3 seguita da quella 3x1;
4. **CLASSIFICATORE AUSILIARIO:** è inserita un'ulteriore CNN tra i layers della rete (durante l'allenamento) per aumentarne la profondità;
5. **RIDUZIONE DELLA DIMENSIONALITA' DELLA GRIGLIA;**

Di seguito viene presentata l'architettura di una rete InceptionV3:

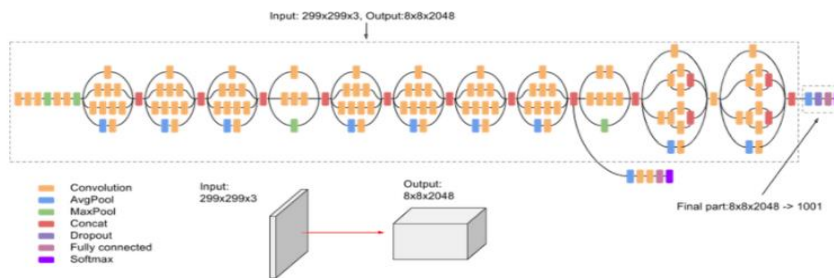


Figure 5: Architettura di un modulo InceptionV3

GESTIONE DELLE SOVRAPPOSIZIONI

All'uscita della rete è stata implementata una funzione che ha permesso di eliminare tutte le ambiguità di assegnazione ad una specifica classe. Questa funzione confronta due canali alla volta e assegna univocamente ad un pixel una sola classe in caso di conflitti. Dopo alcune prove si è deciso di assegnare al nucleo i pixel che erano stati assegnati sia al nucleo che allo sfondo, e sia al nucleo che al citoplasma, mentre al citoplasma i pixel che erano stati assegnati sia al citoplasma che allo sfondo. Inoltre, in caso di pixel non assegnati si è deciso di assegnarli allo sfondo.

- Pixel ∈ nucleo & sfondo → Pixel ∈ nucleo
- Pixel ∈ nucleo & citoplasma → Pixel ∈ nucleo
- Pixel ∈ sfondo & citoplasma → Pixel ∈ citoplasma
- Pixel non classificati → Pixel ∈ sfondo

⁴ <https://netai.it/guida-rapida-alle-funzioni-di-attivazione-nel-deep-learning/#page-content>

⁵ <https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/>

POST-PROCESSING

In uscita dalla rete neurale sono state applicate delle funzioni di post-processing per poter migliorare i risultati della rete:

- **Chiusura** effettuata con elemento strutturale di forma circolare in quanto si sta segmentando oggetti circolari; si è scelto di utilizzare la chiusura e non l'apertura perché ciò che ne deriva è un aumento delle prestazioni, a discapito però di una suddivisione non propriamente corretta delle cellule. Con l'apertura si sarebbe infatti ottenuta una suddivisione più corretta dal punto di vista delle cellule ma una diminuzione delle prestazioni.
- **Remove small objects** (con `min_size` a 80);
- **Remove small holes** (con `area_threshold` pari a 80);

VALIDAZIONE

È stata creata una funzione di validazione che itera su tutte le immagini in uscita dalla rete e per ogni immagine fornisce in uscita le segmentazioni finali. La funzione segue queste fasi:

1. Applicazione di una **threshold** per creare una maschera binaria a partire dall'output della rete. Questa operazione è fatta separatamente per ogni livello così da rendere possibile applicare soglie indipendenti.
2. Applicazione della funzione **post-processing**.
3. Applicazione della funzione di **controllo della sovrapposizione**.

Come output della funzione si ha una matrice di dimensioni $[N \times \text{dim} \times \text{dim} \times 3]$ dove N indica il numero di immagini del dataset considerato, dim la dimensione delle immagini in pixel (256 nel nostro caso) e 3 indica il numero delle classi. Inoltre, in output a questa funzione ci sono anche tutte le metriche di valutazione sulla bontà di segmentazione, che verranno descritte successivamente.

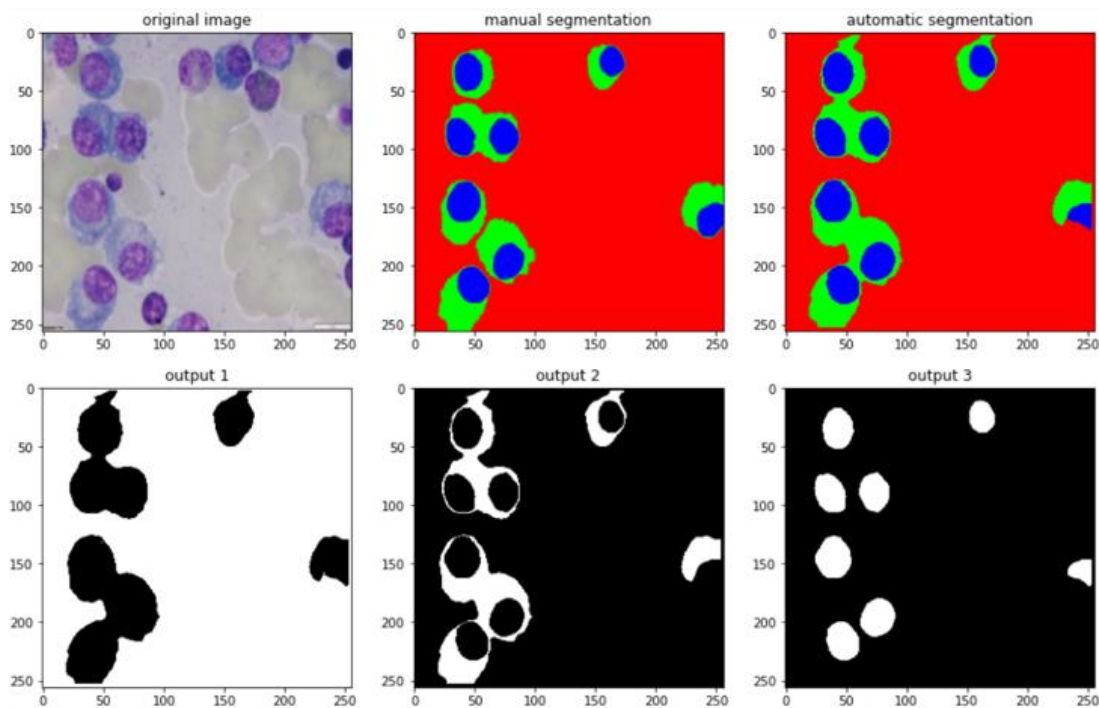


Figure 6: Esempio di output della rete dopo post-processing e analisi sovrapposizioni (immagine 27 del Training Set)

SEGMENTAZIONE DELLE SINGOLE CELLULE

Dopo aver ottenuto le maschere in uscita dal post-processing, l'obiettivo è ora quello di provvedere alla corretta segmentazione delle singole cellule. L'operazione risulta abbastanza complessa, specialmente nel caso di cellule vicine, dove i citoplasmici delle cellule risultano indistinguibili gli uni dagli altri.

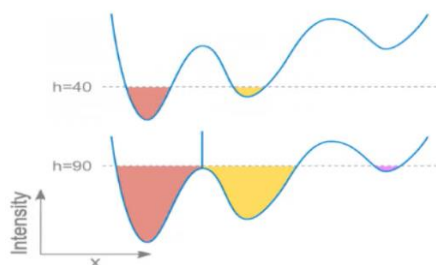


Figure 7: Esempio di profilo di intensità dei pixel di un'immagine

Algoritmo di watershed

A tale scopo è stato utilizzato l'**algoritmo di watershed**, una funzione in grado di segmentare specifiche aree di un'immagine sulla base della similarità dei valori assunti dai pixel. Immaginando infatti di osservare un'immagine in scala di grigi come una superficie tridimensionale, watershed "riempie" le valli della superficie associando loro una segmentazione. Nel momento in cui due valli si uniscono, l'algoritmo impone automaticamente una barriera, in modo tale da non unire la segmentazione di due oggetti distinti.

L'algoritmo di watershed di OpenCV è un algoritmo marker-based, per cui sono necessarie delle operazioni preliminari utili a distinguere tutto ciò che è nostro interesse segmentare (le cellule) dallo sfondo e assegnare loro delle etichette (marker).

Si riassumono di seguito i passi seguiti nell'implementazione dell'algoritmo di segmentazione:

- **Creazione delle maschere cellulari:** Si generano nuove maschere che identificano le cellule, date dalla sovrapposizione dei layer di nucleo e citoplasma;
- **Identificazione dello sfondo:** Si sfrutta un'operazione di dilatazione della maschera, per determinare con certezza tutto ciò che è sfondo;
- **Calcolo della Distance transform:** La Distance Transform calcola la distanza in pixel di ogni pixel dal punto più vicino di colore opposto. Vengono usati i nuclei in quanto identificano quasi con certezza la presenza di una cellula (a meno di errore da parte della rete neurale) e vengono scartate automaticamente segmentazioni di puro citoplasma;
- **Identificazione degli oggetti di interesse:** Utilizzando una soglia basata sulla Distance Transform (0.5 del suo valore massimo) si è in grado di isolare le parti più interne dei nuclei e scartare quelli troppo piccoli, che probabilmente sono stati individuati erroneamente. A partire dalle zone individuate viene effettuata la segmentazione e queste zone vengono denominate come 'sicuramente appartenenti alla cellula';
- **Definizione della regione sconosciuta:** Tutto ciò che non appartiene a nessuna delle due aree sopra definite (ciò che è sicuramente sfondo e ciò che appartiene sicuramente alle cellule) è classificato come regione ignota;
- **Assegnazione delle etichette:** Le regioni note con certezza sono etichettate con numeri interi positivi diversi, mentre la regione ignota viene etichettata come zero. Per evitare conflitti, i marker delle cellule partono dal valore di 10.
- **Utilizzo dell'algoritmo di watershed:** A questo punto si effettua la segmentazione utilizzando l'algoritmo di watershed implementato nella libreria OpenCV, assegnando porzioni dell'area ignota alle cellule o allo sfondo.

Sono state eseguite diverse prove al fine di ottenere dei parametri ottimali per l'individuazione dello sfondo (che varia in base alla dimensione del kernel di dilatazione utilizzato) e delle regioni di interesse (determinate principalmente in base alla soglia scelta, calcolata a partire dalla Distance Transform); la valutazione è stata fatta a partire da un'analisi visiva dei risultati e confrontando il numero di cellule segmentate con l'algoritmo e quelle segmentate da parte dell'operatore.

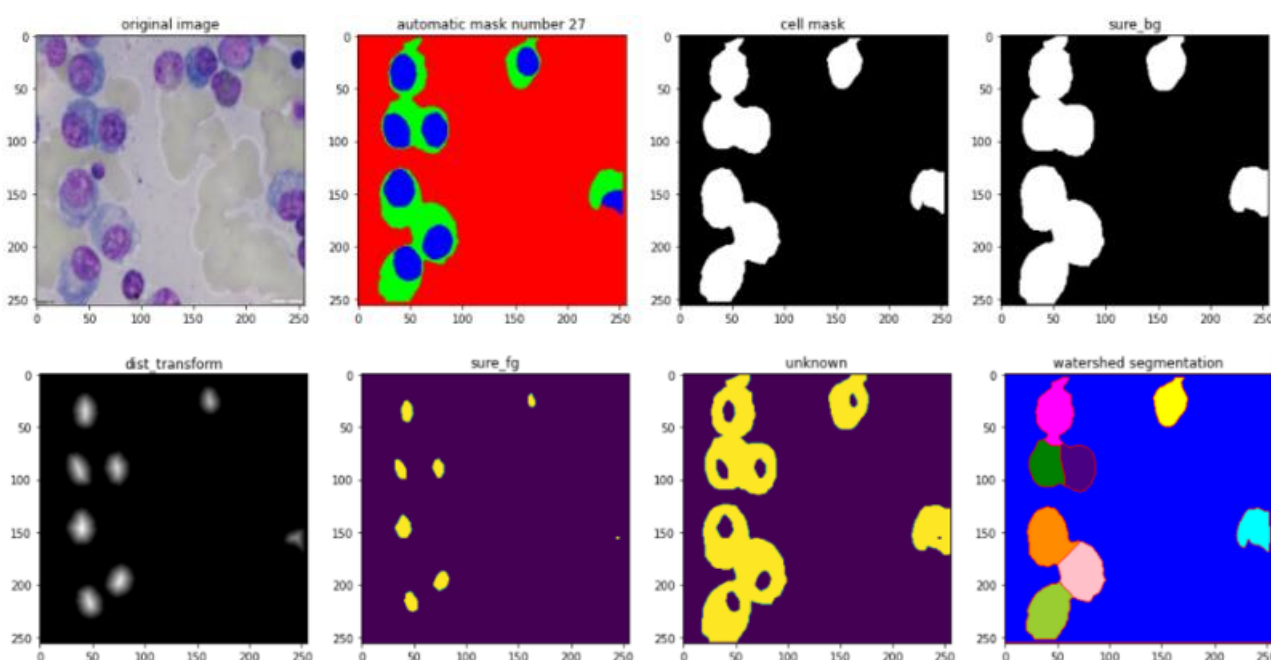


Figura 8: Output della funzione di watershed con passaggi intermedi: (1) immagine originale, (2) maschera automatica (immagine 27 del Training Set), (3) maschere cellule senza operazione di dilatazione, (4) maschere cellule con operatore di dilatazione, (5) distance transform dei nuclei, (6) aree di interesse (nuclei), (7) regione sconosciuta, (8) segmentazione watershed.

COSTRUZIONE MASCHERE E VERIFICA CELLULE INDIVIDUATE

Preliminarmente al salvataggio sono state costruite delle maschere in scala di grigio per le cellule singole. Per fare ciò, è stata scritta una funzione che prende in input la maschera con i markers ottenuta dall'algoritmo di watershed e itera su tutti i differenti numeri salvando ogni cellula singolarmente. Con la conversione in scala di grigi si ottengono immagini in formato uint8 di dimensioni [dim x dim x 1] in cui il nucleo ha valore 255, il citoplasma 128 e lo sfondo 0.

Nella stessa funzione viene anche confrontato il numero di cellule trovato con il numero di cellule delle segmentazioni manuali (che era stato salvato per ogni immagine durante il caricamento) e viene stampato a video.

L'output della funzione è una lista di matrici di formato [dim x dim x N] dove N è il numero di cellule. Queste immagini sono quelle che poi verranno salvate in formato tiff, opportunamente ridimensionate.

Per completezza è stato contato il numero totale di cellule individuate per ogni set (sia Training che Validation) e il numero di cellule effettivamente presenti, in modo da dare un'idea dell'accuratezza del sistema.

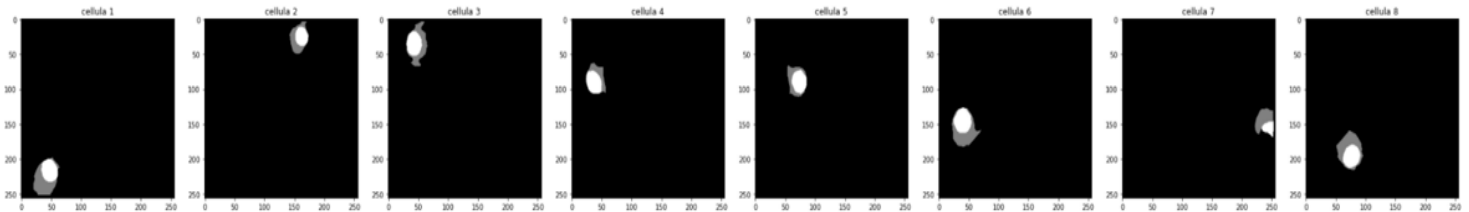


Figure 9: Esempio di output con cellule divise (immagine 27 del Training Set)

METRICHE DI VALUTAZIONE

Le metriche di valutazione utilizzate per valutare le performance del modello (con algoritmi di post-processing annessi), sia sul Training Set che sul Validation Set, mirano a quantificare la similarità tra la maschera manuale ("disegnata" e fornita dal medico) e la maschera automatica che viene restituita come output dalla rete. Tutti i parametri che verranno descritti in seguito sono stati calcolati sulla cellula (intesa come sovrapposizione delle maschere riferite a nucleo e citoplasma), citoplasma e nucleo. Tutti i termini sono stati espressi in maniera percentuale essendo valori compresi nell'intervallo [0 1]. Di seguito vengono presentate le metriche utilizzate:

Intersection Over Union (IoU)

$$IoU(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

Viene utilizzata per valutare numericamente la percentuale di sovrapposizione tra la maschera di segmentazione manuale (X) e la maschera predetta in output dal modello (Y). Parlando di immagini, essa rappresenta il numero di pixel che risultano essere in comune tra le 2 maschere rapportato al numero totale di pixel delle 2 maschere.

Dice similarity coefficient (DSC):

$$DSC(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|}$$

Tale coefficiente (anche chiamato F1-score) è utilizzato per quantificare la similarità tra 2 immagini (che in tal caso sono la maschera di segmentazione manuale (X) e quella automatica (Y)). Essendo anche questo coefficiente pixel-based, esso rappresenta il numero di pixel in comune tra le 2 maschere, moltiplicato per 2, rapportato alla somma dei pixel della maschera manuale e di quella automatica.

Precision:

$$Precision(X, Y) = \frac{|X \cap Y|}{|Y|}$$

Oppure:

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

In imaging, la Precision è calcolata come il rapporto tra la sovrapposizione delle maschere e la maschera segmentata. E' una misura legata all'errore di sovra-segmentazione della maschera, in quanto un valore basso corrisponde ad un numero alto di pixel individuati che non sono effettivamente appartenenti alla cellula.

Recall:

$$\text{Recall}(X,Y) = \frac{|X \cap Y|}{|X|}$$

Oppure:

$$\text{Recall} = \frac{TP}{TP + FN}$$

In imaging, la Recall si traduce nel rapporto tra la sovrapposizione delle due maschere e la maschera di riferimento. È una misura legata all'errore di sotto-segmentazione della maschera, in quanto un valore basso significa che una percentuale dei pixel appartenenti alla cellula non sono stati individuati.

VALUTAZIONE PERFORMANCE: FASE 1

Performance output rete neurale

Per prima cosa sono state valutate le performance della rete basandosi unicamente sul risultato dell'output della rete neurale: per far ciò, all'interno della funzione di validazione è stata utilizzata la funzione "sovrapposizione" ma non quella di post-processing, per valutare l'effetto sulle performance unicamente rispetto alla scelta della rete. Sempre per questa ragione le threshold per la creazione delle maschere sono state inizialmente imposte pari a 0.5 per tutti i livelli (sfondo, citoplasma, nucleo).

Di seguito vengono presentate le performance sulla base delle 2 prove (vedi sezione IMPLEMENTAZIONE RETE NEURALE) sul Validation Set:

Segmentazione cellula		IoU (%)	Dice (%)	Precision (%)	Recall (%)
PROVA 1	Inception v3	47.06 ±23.83	60.14 ±27.36	72.27 ±32.03	54.57 ±26.30
	Resnet34	47.10 ±32.32	60.02 ±25.80	75.07 ±30.55	52.24 ±24.72
	Densenet121	33.27 ±30.91	42.29 ±33.62	71.25 ±32.60	35.05 ±31.93
PROVA 2	Inception v3	42.06 ±23.62	54.76 ±27.22	71.56 ±31.19	47.07 ±25.42
	Resnet34	34.83 ±22.69	47.35 ±26.14	73.53 ±31.19	38.01 ±24.51
	Densenet121	21.73 ±28.37	28.61 ±31.23	61.87 ±45.63	21.91 ±28.36

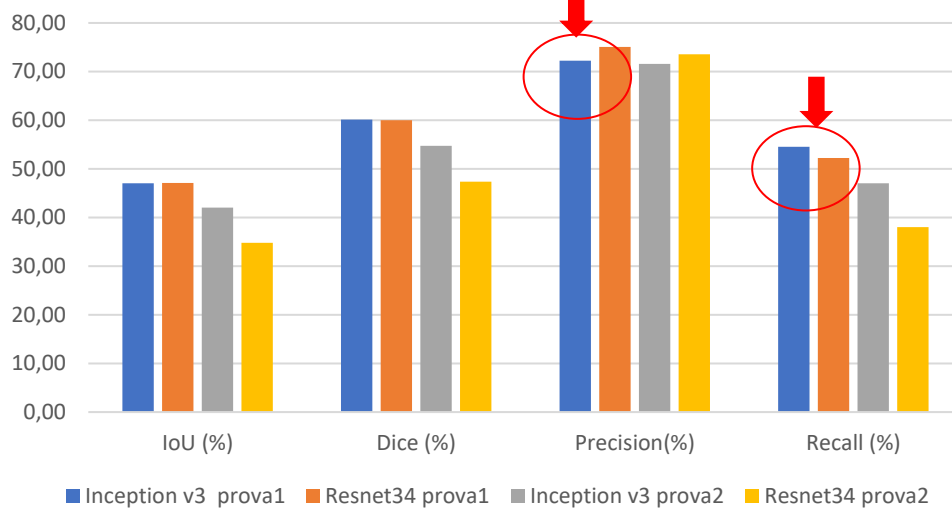
Tabella 2. Performance della segmentazione della cellula (somma dei nuclei e citoplasma) nel Validation Set.

Il problema a cui facciamo fronte è un problema di segmentazione di cellule tumorali e come tale bisogna fare delle scelte logiche, considerando sempre il caso peggiore. La priorità è quella di far sì che tutte le cellule che hanno una minima probabilità di essere plasmacellule tumorali vengano segmentate: si predilige quindi la sovra-segmentazione piuttosto che la sotto-segmentazione, in altri termini si predilige avere un numero più alto di Totali Positivi (e quindi anche di Falsi Positivi), piuttosto che un numero minore di Falsi Negativi.

Idealmente, la rete migliore è quella che gode, oltre che di parametri IoU e Dice elevati, anche di buone Recall e Precision; in particolare, questi ultimi due coefficienti sono indici rispettivamente di errori di sotto-segmentazione e sovra-segmentazione, ed idealmente dovrebbero essere pari al 100%. Tra le due reti che hanno in generale performance migliori (Inception v3 e Resnet34 nella prova1), si è deciso in base alla premessa fatta, di prediligere la rete con Precision minore piuttosto che quella con Recall minore: ovvero si è preferito avere una sovra-segmentazione piuttosto che una sotto-segmentazione.

Sulla base dei concetti appena discussi, la rete migliore analizzata è l'**Inceptionv3** con i parametri della prima prova. Di conseguenza, per valutare le performance nella fase 2 (post-processing) e per il watershed algorithm, verrà utilizzata unicamente tale rete.

Performance delle reti Resnet34 e Inceptionv3



VALUTAZIONE PERFORMANCE: FASE 2

Performance output rete neurale con post-processing

In questa seconda fase si utilizza il parametro di IoU a rappresentazione della bontà dell'algoritmo di segmentazione.

PRIMA VERIFICA SULLE PERFORMANCE (IoU): La prima prova che è stata effettuata a seguito della scelta della rete riguarda le performance dell'output della rete neurale al variare dei valori della threshold, per i canali di nucleo e citoplasma. La soglia dello sfondo è stata mantenuta a 0.5, questo perché oltre a non influire nel calcolo delle metriche, grazie al controllo delle sovrapposizioni tutto quello che non è assegnato alle altre due classi viene automaticamente assegnato allo sfondo.

TRAINING

threshold citoplasma	threshold nuclei	Cellula (%)	Citoplasma (%)	Nucleo (%)
0.5	0.5	53.35 ± 22.39	42.85 ± 18.88	52.27 ± 23.26
0.2	0.2	61.47 ± 24.48	49.50 ± 20.24	59.34 ± 25.06
0.1	0.1	61.68 ± 25.05	48.70 ± 20.18	59.78 ± 25.33
0.1	0.2	62.11 ± 25.35	50.85 ± 21.06	59.34 ± 25.06
0.2	0.1	60.87 ± 24.32	47.49 ± 19.54	59.78 ± 25.33

VALIDATION

threshold citoplasma	threshold nuclei	Cellula (%)	Citoplasma (%)	Nucleo (%)
0.5	0.5	47.60 ± 23.83	39.30 ± 19.98	45.76 ± 23.71
0.2	0.2	55.77 ± 26.03	45.29 ± 21.11	54.79 ± 25.50
0.1	0.1	56.62 ± 26.19	45.78 ± 21.08	55.32 ± 25.42
0.1	0.2	57.35 ± 26.63	47.81 ± 22.01	54.79 ± 25.50
0.2	0.1	55.21 ± 25.53	43.46 ± 20.30	55.32 ± 25.42

Le soglie sono state scelte sulla base del valore IoU più elevato per l'intera cellula, e sono rispettivamente: per lo sfondo **0.5**, per il citoplasma **0.1**, per i nuclei **0.2**.

SECONDA VERIFICA SULLE PERFORMANCE (IoU): Nella seconda prova è stato variato il valore del footprint, cambiando il raggio dell'elemento strutturale.

TRAINING

footprint	Cellula (%)	Citoplasma (%)	Nucleo (%)
3	63.47 ± 25.28	52.16 ± 21.18	60.05 ± 25.24
5	63.48 ± 25.24	51.77 ± 21.05	60.33 ± 25.23
7	63.32 ± 25.14	51.18 ± 20.85	60.34 ± 25.19

VALIDATION

footprint	Cellula (%)	Citoplasma (%)	Nucleo (%)
3	58.19 ± 26.95	48.37 ± 22.19	55.35 ± 25.92
5	58.24 ± 26.88	48.09 ± 22.08	55.63 ± 25.96
7	58.22 ± 26.81	47.84 ± 22.01	55.55 ± 25.88

Anche qui il raggio dell'elemento di footprint è stato scelto sulla base del valore IoU più elevato per l'intera cellula; il raggio scelto ha dimensione **3**.

VALUTAZIONE PERFORMANCE: FASE 3

Controllo finale sul numero di cellule segmentate

Infine, considerate tutte le operazioni, le cellule totali segmentate in maniera automatica risultano essere:

- 1812 su 1655 (dalle segmentazioni manuali) nel Training Set
- 243 su 224 (dalle segmentazioni manuali) nel Validation Set

- 422 in totale nel Test Set.

Come atteso, il numero di cellule segmentate in maniera automatica è superiore al numero di cellule segmentate dall'operatore. E' una conferma della scelta sulla sovra-segmentazione fatta in precedenza.

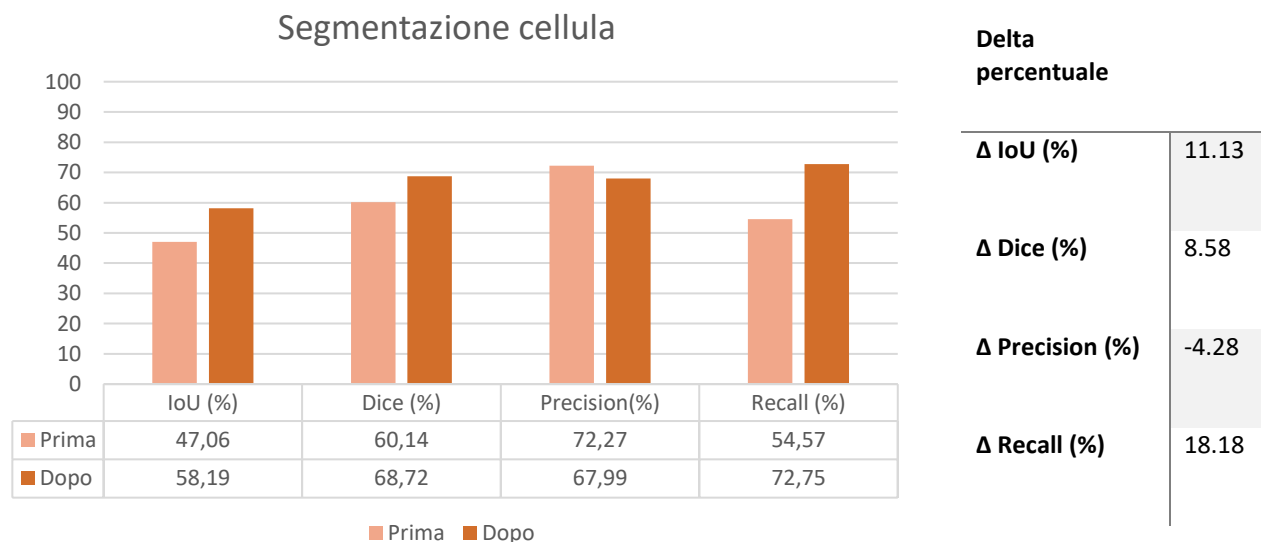
CONCLUSIONI E PERFORMANCE FINALI

Di seguito vengono riportate le performance finali dell'algoritmo di segmentazione in seguito all'impostazione dei parametri di rete, soglia, footprint come sopra descritto.

TrainingSet	IoU (%)	Dice (%)	Precision (%)	Recall (%)	ValidationSet	IoU(%)	Dice (%)	Precision (%)	Recall(%)
cellula	63.47±25.28	73.69±26.08	71.90±26.85	77.47±27.18	cellula	58.19±26.95	68.72±28.95	67.99±30.90	72.75±30.78
citoplasma	52.16±21.18	65.29±23.81	69.44±27.04	63.07±22.79	citoplasma	48.37±22.19	61.33±26.21	66.48±31.07	59.35±25.36
nucleo	60.05±25.24	70.86±26.88	74.02±29.16	69.79±26.97	nucleo	55.35±25.93	66.73±27.65	71.06±30.07	66.34±28.57

Dal confronto tra Training Set e Validation Set, le prestazioni risultano minori per il secondo; questo a conferma che l'algoritmo è più performante nel processare le immagini che gli sono state fornite durante l'allenamento.

Come si evince dal grafico sottostante, la combinazione scelta di parametri di soglia e operazioni morfologiche (post-processing) ha influito positivamente sulle metriche di valutazione, ad eccezione della metrica di Precision. Un valore più basso della metrica di Precision indica che la sovra-segmentazione è aumentata nel momento in cui sono state applicate le soglie ed è stato effettuato il post-processing. Questo decremento non è un problema ai fini dell'obiettivo prefissato in quanto si è preferito sovra-segmentare piuttosto che sotto-segmentare. Le altre performance rimangono comunque soddisfacenti, attestandosi ad un valore superiore al 58%. L'algoritmo, pertanto, si può reputare sufficientemente performante.



In termini di segmentazione cellulare, di conseguenza, il numero delle cellule segmentate con l'algoritmo sono maggiori rispetto quelle segmentate dal patologo. Tuttavia, sebbene il numero di cellule sia maggiore, non è possibile affermare con certezza che tutte le plasmacellule tumorali vengano correttamente segmentate.