



UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Bioingegneria

Sviluppo di un algoritmo basato su tecniche di deep learning per la segmentazione di cellule dendritiche in immagini della cornea

Laureando:

Riccardo Zorzan

Relatore:

Prof.ssa Maria Pia Saccomani

Correlatori:

Alessia Colonna

Fabio Scarpa

A.A. 2019/2020

Questa tesi è stata fatta in collaborazione con:

The School of Optometry and Vision Science at University of New South Wales
(UNSW, Sydney).



UNSW
SYDNEY

Abstract

La microscopia confocale è una tecnologia che consente di visualizzare in-vivo, in modo rapido e non invasivo, l'architettura cellulare e le fibre nervose della cornea. Le immagini acquisite tramite tale strumento, permettono di ottenere importanti informazioni cliniche sullo stato di salute del tessuto corneale. In particolare, le immagini acquisite a una specifica profondità, ovvero in corrispondenza del plesso nervoso sub-basale, consentono la visualizzazione dei nervi e delle cellule dendritiche corneali (*Dendritic Cells*, DCs).

La correlazione tra densità e morfologia delle cellule dendritiche e patologie oculari o sistemiche è stata dimostrata in diversi studi.

In questo lavoro, è stato progettato un algoritmo di *Deep Learning* per la segmentazione automatica delle cellule dendritiche corneali. Il modello proposto si basa su una rete neurale convoluzionale (CNN), con architettura *U-Net*, in cui un percorso di codifica, per catturare i descrittori delle DCs, è seguito da un percorso di decodifica simmetrico per la localizzazione spaziale delle cellule dendritiche.

Il modello proposto ha permesso una segmentazione automatica delle cellule dendritiche corneali con una sensibilità superiore al 90%, rispetto alla segmentazione manuale, migliorando inoltre le prestazioni di un algoritmo automatico sviluppato precedentemente.

Indice

Abstract	pagina 6
Introduzione	pagina 7
Capitolo 1: La cornea	pagina 8
1.1 Anatomia macroscopica della cornea	pagina 8
1.2 Anatomia microscopica della cornea	pagina 9
1.2.1 Epitelio	pagina 9
1.2.1.1 Cellule superficiali	pagina 10
1.2.1.2 Cellule intermedie o alari (<i>wing cells</i>)	pagina 11
1.2.1.3 Cellule basali	pagina 11
1.2.1.4 Lamina basale	pagina 11
1.2.2 Membrana di Bowman	pagina 12
1.2.3 Stroma	pagina 12
1.2.4 Membrana di Descemet	pagina 13
1.2.5 Endotelio	pagina 13
1.3 Innervazione della cornea	pagina 14
1.4 Cellule dendritiche corneali	pagina 15
1.5 Patologie cliniche di interesse	pagina 18
1.5.1 Cheratite infettiva	pagina 18
1.5.2 Malattia di <i>Behcet</i>	pagina 19
1.5.3 Malattia di <i>Fabry</i>	pagina 19
1.5.4 Sindrome dell'occhio secco	pagina 20
1.5.5 Diabete mellito di tipo 2	pagina 21
1.6 Microscopia confocale in vivo (<i>In-vivo confocal microscopy</i> – IVCN)	pagina 22
Capitolo 2: Deep Learning	pagina 26
2.1 Introduzione	pagina 26
2.2 Neurone artificiale	pagina 28
2.3 Architettura di rete neurale <i>Feed-Forward</i>	pagina 30
2.4 Addestramento supervisionato di una rete neurale	pagina 32
2.4.1 Classificazione multi-classe	pagina 32
2.4.2 <i>Stochastic Gradient Descent</i> (SGD)	pagina 34
2.4.2.1 ADAM (<i>Adaptative Moment Estimation</i>)	pagina 36

2.4.3 <i>Back-propagation</i>	pagina 37
2.4.4 <i>Overfitting</i>	pagina 40
2.4.4.1 <i>Regolarizzazione</i>	pagina 40
2.4.4.2 <i>Dropout</i>	pagina 41
2.4.4.3 <i>Batch Normalization</i>	pagina 41
2.4.4.4 <i>Data augmentation</i>	pagina 42
2.4.4.5 <i>Validazione</i>	pagina 43
2.5 Reti Neurali Convoluzionali	pagina 43
2.5.1 Architettura di una Rete Neurale Convoluzionale	pagina 45
2.5.1.1 <i>Layer di estrazione delle features</i>	pagina 45
2.5.1.1.1 <i>Strati di convoluzione (Convolutional layer)</i>	pagina 45
2.5.1.1.2 <i>Funzione di attivazione – ReLU layer</i>	pagina 48
2.5.1.1.3 <i>Pooling</i>	pagina 48
2.5.1.2 <i>Layer di classificazione</i>	pagina 49
2.5.1.2.1 <i>Fully connected layer</i>	pagina 49
2.5.1.2.2 <i>SoftMax layer</i>	pagina 49
2.5.1.2.3 <i>Classification layer</i>	pagina 50
2.6 Segmentazione semantica	pagina 50
2.7 Applicazioni cliniche del Deep Learning	pagina 52
2.7.1 Applicazioni in oftalmologia	pagina 54
Capitolo 3: Segmentazione automatica delle cellule dendritiche corneali	pagina 55
3.1 Dataset	pagina 56
3.2 <i>Pre-processing</i>	pagina 56
3.2.1 <i>Creazione immagini pixel-label</i>	pagina 56
3.2.2 <i>Data augmentation</i>	pagina 59
3.3 Rete Neurale Convoluzionale: architettura <i>U-Net</i>	pagina 60
3.4 Addestramento della rete	pagina 62
3.4.1 <i>Iper-parametri</i>	pagina 62
3.4.2 <i>100-fold cross-validation</i>	pagina 64
3.5 <i>Post-processing</i>	pagina 64
Capitolo 4: Risultati	pagina 67
4.1 <i>Analisi statistica</i>	pagina 67
Conclusioni	pagina 73
Bibliografia	pagina 75

Introduzione

La cornea è un tessuto trasparente multi-stratificato che adempie ad una duplice funzione: agisce infatti sia come barriera meccanicamente resistente e chimicamente impermeabile tra l'occhio e l'ambiente esterno, sia come maggiore superficie rifrattiva.

Fino a tempi recenti si pensava che la cornea fosse un tessuto dotato di privilegio immunitario, a causa dell'assenza di cellule presentanti l'antigene (*Antigen Presenting Cells*, APCs). Tuttavia, studi recenti hanno dimostrato l'esistenza di cellule dendritiche (*Dendritic Cells*, DCs) inattive nello stroma anteriore e nell'epitelio corneale, soprattutto tra lo strato basale e sub-basale.

Grazie alla microscopia confocale in vivo (IVCM), è possibile acquisire, in modo rapido e non invasivo, immagini dei vari strati della cornea. In particolare, le immagini acquisite a una specifica profondità, ovvero in corrispondenza del plesso nervoso sub-basale, consentono la visualizzazione delle cellule dendritiche corneali.

La correlazione tra densità e morfologia delle DCs e alcune patologie oculari (come, ad esempio, la sindrome dell'occhio secco) o sistemiche (come il diabete) è stata dimostrata in diversi studi. Per studiare la correlazione tra questi fattori, dunque, si rende necessario identificare e segmentare le cellule dendritiche presenti nella cornea. Questa analisi, sino ad ora, viene effettuata manualmente dai clinici: ciò comporta che l'analisi sia *time-consuming* e influenzata dalla soggettività.

Questa tesi si prefigge lo scopo di fornire descrittori robusti della cornea di forma automatica, con il fine ultimo di identificare e segmentare le cellule dendritiche in maniera oggettiva e più rapida rispetto all'analisi manuale.

È stato, dunque, progettato un algoritmo di *Deep Learning* per la segmentazione completamente automatica delle cellule dendritiche corneali in immagini IVCM del plesso nervoso sub-basale. Il modello proposto si basa su una rete neurale convoluzionale (CNN), con architettura *U-Net*, in cui un percorso di codifica, per catturare i descrittori delle DCs, è seguito da un percorso di decodifica simmetrico per la localizzazione spaziale delle cellule dendritiche.

Outline della tesi

Nel primo capitolo viene trattata l'anatomia macro- e microscopica della cornea. Vengono inoltre descritte le cellule dendritiche, nonché alcune patologie, oculari e sistemiche, correlate alla presenza e allo stato di maturazione delle DCs stesse. Vengono infine presentati principi e tecnologie dello strumento utilizzato per l'acquisizione delle immagini: la microscopia confocale.

Nel secondo capitolo vengono presentate le basi degli algoritmi di *Deep Learning*, soffermandosi in particolare sulle tecniche utilizzate in questo lavoro. Viene descritto il paradigma di apprendimento supervisionato di una rete neurale per la classificazione multi-classe. Vengono poi trattate in dettaglio le reti neurali convoluzionali, e le architetture di codifica/decodifica per la segmentazione semantica.

Nel terzo capitolo viene presentato l'algoritmo proposto per la segmentazione automatica delle cellule dendritiche corneali. Sono quindi descritti materiali e metodi utilizzati, con particolare dettaglio sull'architettura di rete neurale convoluzionale implementata.

Infine, nel quarto capitolo vengono presentati, attraverso un'analisi statistica, i risultati ottenuti.

Capitolo 1.

La cornea

La cornea è un tessuto trasparente avascolare collocato nella parte anteriore della tonaca fibrosa esterna del bulbo oculare, di cui occupa un sesto del totale (i rimanenti cinque sestimi di tonaca fibrosa posteriore sono occupati dalla sclera, unita alla cornea in corrispondenza del *limbus* corneo-sclerale).

Si tratta di una struttura anatomica che adempie ad una duplice funzione: agisce infatti sia come barriera meccanicamente resistente e chimicamente impermeabile tra l'occhio e l'ambiente esterno, sia come maggiore superficie rifrattiva, in grado di fornire circa 48 diottrie di potenza e focalizzare i raggi luminosi sulla retina con degradazione e dispersione ottica minima.

La cornea si è specializzata sia funzionalmente che strutturalmente per raggiungere le proprietà ottiche richieste, soddisfacendo i criteri ricercati per le lenti ottiche: adeguata regolarità della superficie, forma sferica e trasparenza interna. Per soddisfare quest'ultimo requisito, caratteristiche strutturali dello stroma (di cui si parlerà in seguito) e meccanismi fisiologici di idratazione si uniscono alla struttura avascolare del tessuto corneale: la cornea è infatti priva di vasi sanguigni; trae ossigeno e nutrimenti dall'ambiente, nella parte anteriore, e dall'umore acqueo nella parte posteriore.

1.1 Anatomia macroscopica della cornea

La superficie anteriore della cornea ha forma convessa ed orizzontalmente ellittica, a causa della maggiore prominente del *limbus* corneo-sclerale nella parte superiore e inferiore; il diametro orizzontale misura circa 12.6 mm mentre quello verticale 11.7 mm. Presenta inoltre una curvatura non uniforme, risultando approssimativamente sferica nella regione centrale, fondamentale per il conseguimento di una buona acuità visiva, con raggio di curvatura di circa 7.8 mm.

La superficie posteriore della cornea ha forma maggiormente sferica rispetto a quella anteriore, con raggio di curvatura di circa 6.5 mm.

È possibile inoltre osservare un progressivo ispessimento della cornea dalla regione centrale, dove lo spessore è di circa 0.52 mm, alla regione esterna, dove lo spessore è di circa 0.65 mm, dovuto ad un incremento nella quantità di collagene nello stroma periferico.

1.2 Anatomia microscopica della cornea

Dal punto di vista microscopico, la cornea può essere suddivisa in cinque strati, che comprendono (dalla superficie anteriore a quella posteriore):

- Epitelio
- Membrana di Bowman
- Stroma
- Membrana di Descemet
- Endotelio

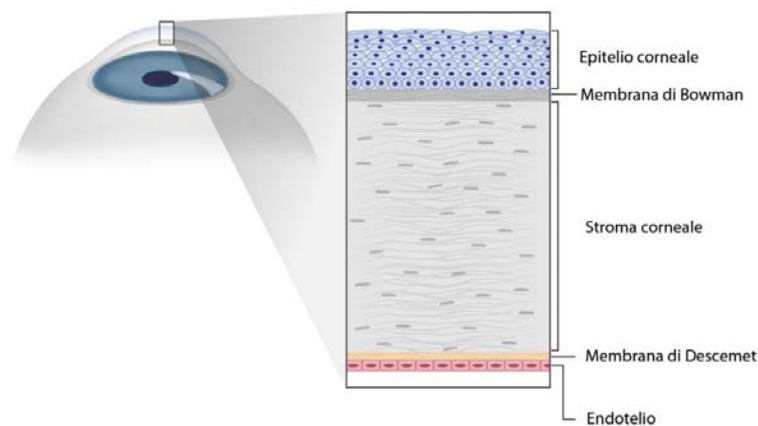


Figura 1.1. **Struttura stratificata della cornea.**

1.2.1 Epitelio

L'epitelio corneale è di tipo pavimentoso pluristratificato, non cheratinizzato, e rappresenta il 10 % dello spessore corneale (è spesso circa 50 micron). La sua funzione primaria è di agire da barriera contro agenti patogeni, sostanze chimiche, acqua e altri corpi estranei. Contribuisce inoltre al potere refrattivo dell'occhio in quanto costituisce la parte interna dell'interfaccia film lacrimale-cornea, di cui è importante sottolineare la relazione simbiotica: affinché la cornea funzioni come lente ottica, è necessario un processo di *smoothing* della superficie corneale esterna, a causa della presenza sull'epitelio di reticolazioni che potrebbero agire da potenziali fonti di degradazione dell'immagine. La presenza del film lacrimale preoculare consente di ottenere una superficie ottica regolare.

L'epitelio corneale consiste di 5-6 strati di tre tipi di cellule: cellule superficiali, cellule intermedie o alari e cellule basali. Esse rappresentano diversi stadi maturativi di una stessa unità

cellulare: le cellule basali sono infatti le uniche cellule epiteliali che subiscono mitosi; le cellule figlie così formate spingono anteriormente e modificano la loro forma in modo da conformarsi alle cellule alari contigue. Continuando a muoversi verso l'esterno diventano infine cellule superficiali, che rappresentano dunque il più alto livello di differenziazione, per infine disintegrarsi nel processo di desquamazione. L'epitelio corneale si rinnova approssimativamente ogni 7 giorni.

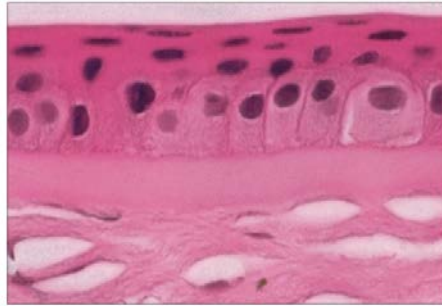


Figura 1.2. **Istologia dell'epitelio corneale.** È possibile apprezzare anche la membrana di Bowman e lo stroma anteriore.

1.2.1.1 Cellule superficiali

Le cellule superficiali dell'epitelio corneale sono disposte tipicamente in due strati e hanno forma poligonale, con un diametro variabile tra i 40 e i 60 micron. In corrispondenza del nucleo hanno uno spessore che varia dai 4 ai 6 micron, mentre perifericamente sono spesse 2 micron. Le cellule superficiali hanno inoltre pochi organelli e un complesso di Golgi poco sviluppato. La membrana superficiale di queste cellule è caratterizzata da due aspetti distintivi, dati dalla presenza di proiezioni microscopiche (microvilli, reticolazioni, micropliche) e di un esteso glicocalice. Entrambi i fattori giocano un ruolo importante nel meccanismo di aderenza al film lacrimale. Interessante inoltre come la presenza di un cuore di filamenti di actina nei microvilli e micropliche potrebbe consentire il movimento delle proiezioni superficiali.

Un'altra proprietà fondamentale dello strato cellulare superficiale è data dalla presenza di un complesso di giunzioni tra cellule adiacenti lateralmente, rappresentazione anatomica della capacità dell'epitelio di fungere da barriera al movimento di sostanze dal film lacrimale allo spazio intercellulare dell'epitelio. Il sistema giunzionale consiste, in particolare, di giunzioni occludenti (*tight junctions*), che fungono inoltre da barriera permeabile per il controllo dei meccanismi di trasporto ionico all'interno dell'epitelio, e di numerosi attaccamenti desmosomiali, in grado di conferire stabilità meccanica laterale.

1.2.1.2 Cellule intermedie o alari (*wing cells*)

Le cellule intermedie, dette anche alari in virtù della loro forma, sono tipicamente disposte in due o tre strati e hanno nuclei arrotondati o lievemente allungati in direzione parallela alla superficie. Perifericamente risultano estensivamente interdigitate e congiunte da giunzioni desmosomiali e giunzioni serrate (*gap junctions*), ossia canali permeabili che consentono la diffusione di piccole molecole. Aggregati di tonofilamenti e microtubuli riempiono il citoplasma delle cellule alari, consentendone il mantenimento della struttura tridimensionale. Spesso i tonofilamenti convergono verso il placode interno dei desmosomi, contribuendo alle loro proprietà strutturali.

1.2.1.3 Cellule basali

Le cellule basali, caratterizzate da un diametro tra gli 8 e i 10 micron, costituiscono un singolo strato cellulare, con altezza variabile tra i 18 e i 20 micron. Hanno forma colonnare nella cornea centrale, cubica nella regione periferica.

Come già accennato, rappresentano l'unico strato germinativo dell'epitelio e possiedono molti più organelli citoplasmatici rispetto agli strati più anteriori. La presenza di filamenti di actina nel lato citoplasmatico della membrana consente loro capacità di migrazione nel processo di guarigione di abrasioni.

Sui bordi laterali le cellule basali sono estensivamente interdigitate e congiunte per mezzo di giunzioni aderenti (*zonula adherens*) e giunzioni serrate. Sulla superficie posteriore, un complesso emidesmosomiale consente loro di aderire saldamente alla lamina basale sottostante.

1.2.1.4 Lamina basale

La lamina basale è una sottile membrana spessa 40-60 nm posizionata tra lo strato cellulare basale dell'epitelio e la membrana di Bowman. È in grado di rigenerarsi se danneggiata ed è dunque coinvolta nei processi di ri-epitelizzazione della cornea. Dal punto di vista strutturale è possibile distinguere una lamina lucida e una lamina densa.

Dal punto di vista biochimico rappresenta invece un prodotto secretorio extracellulare delle cellule basali, nella cui composizione è possibile distinguere collagene di Tipo IV (nella lamina densa), laminina (lamina densa e lucida), fibronectina e fibrina. Fibrille di ancoraggio,

composte da collagene di Tipo VII, ancorano le laminine al tessuto connettivo sottostante terminando in placche di ancoraggio, costituite da collagene di Tipo I.

La funzione principale della lamina basale è dunque quella di formare uno *scaffold* per l'organizzazione dell'epitelio, fungendo in particolare da intermediaria per l'ancoraggio dell'epitelio corneale allo stroma sottostante.

1.2.2 Membrana di Bowman

La membrana di Bowman è uno strato spesso 12 micron di materiale extracellulare quale collagene (di Tipo I e V) e proteoglicani. Le fibrille di collagene, arrangiate in modo irregolare, attraversano la membrana e la ancorano allo stroma anteriore dove si fondono con le lamelle di collagene. Svolge un ruolo di protezione dello stroma e aiuta la cornea nel mantenimento della propria forma.

1.2.3 Stroma

Lo stroma rappresenta circa l'80 % dell'intero spessore corneale. All'interno della sua struttura è possibile riconoscere una componente cellulare e una extracellulare.

La componente extra-cellulare è caratterizzata da una struttura stratificata di fibrille di collagene (predominatamente di tipo I), organizzate in 300-500 lamelle, e circondate da una sostanza amorfa di glicosamminoglicani (GAGs), soprattutto cheratansolfato e condroitinsolfato.

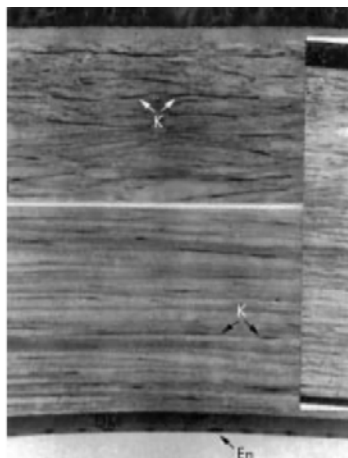


Figura 1.3. **Istologia dello stroma.** Rappresentazione dell'organizzazione delle lamelle di collagene nello stroma. Nella porzione intermedia e posteriore le lamelle sono più compatte. I cheratociti, indicati con la lettera K, sono collocati principalmente all'interfaccia di lamelle adiacenti.

Il collagene non solo conferisce al tessuto resistenza meccanica alla pressione intraoculare, ma è responsabile anche della sua trasparenza. Una delle teorie fisiche alla base della trasparenza dello stroma risiede nell'arrangiamento delle fibrille collagene in un reticolo cristallino perfetto: la luce dispersa dalle singole fibre sarebbe eliminata dall'interferenza distruttiva con la luce dispersa dalle fibre adiacenti. Tuttavia, più recentemente è stato proposto un criterio basato semplicemente sulle dimensioni delle fibrille, in cui l'arrangiamento ordinato delle stesse sembrerebbe non essere un requisito necessario per la trasparenza: lo stroma disperderebbe minimamente la luce perché le fibrille di collagene hanno dimensioni limitate (diametro di circa 300 Å) rispetto alla lunghezza d'onda della luce incidente.

La componente cellulare è rappresentata principalmente dai cheratociti, coinvolti nel mantenimento della matrice extra-cellulare attraverso una costante attività di sintesi di molecole di collagene e glicosamminoglicani. I cheratociti occupano dal 3 al 5 % del volume stromale e si collocano tipicamente tra le lamelle di collagene (Figura 1.3).

1.2.4 Membrana di Descemet

La membrana di Descemet è una lamina basale secreta dall'endotelio composta da collagene di tipo IV e laminina. È costituita da una porzione anteriore fasciata e una posteriore non fasciata. Lo spessore della membrana di Descemet aumenta negli anni, anche se questo incremento sembra interessare solo la porzione posteriore: mentre lo strato anteriore mantiene uno spessore di circa 3 micron, lo spessore di quello posteriore aumenta con l'età da 2 a 10 micron.

Le cellule endoteliali giacciono sulla membrana di Descemet senza alcuna giunzione meccanica specifica.

1.2.5 Endotelio

L'endotelio corneale forma un singolo strato di circa 400 mila cellule, spesso dai 4 ai 6 micron, sulla superficie posteriore della cornea. Viste posteriormente, queste cellule hanno forma principalmente esagonale e larghezza di circa 20 micron.

Le cellule endoteliali, estensivamente interdigitate e congiunte lateralmente da giunzioni occludenti e serrate, hanno un citoplasma ricco di mitocondri e caratterizzato da un reticolo endoplasmatico e da un apparato del Golgi ben sviluppati, sinonimo di un'intensa attività metabolica.

Le cellule endoteliali giocano un ruolo fondamentale nel controllo dell'idratazione dello stroma: grazie alla presenza di meccanismi di trasporto ionico di soluti dallo stroma all'umore acqueo, l'endotelio corneale deidrata lo stroma mantenendone la trasparenza.

Le cellule endoteliali sono essenzialmente amitotiche. Alla nascita la densità cellulare varia da 3500 a 4000 cell/mm², mentre una cornea adulta presenta una densità variabile dalle 1400 alle 2500 cell/mm². A questo decremento nella densità corrispondono processi di pleomorfismo e polimegatismo cellulare.

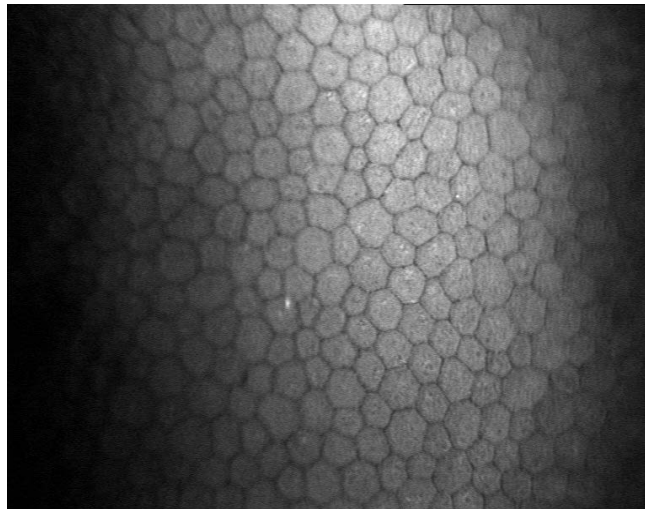


Figura 1.4. **Struttura regolare delle cellule endoteliali della cornea.**

1.3 Innervazione della cornea

La cornea è uno dei tessuti che presentano la maggiore densità di fibre nervose nel corpo umano. È innervata da processi neurali provenienti principalmente da due gangli: il ganglio trigeminale (semilunare), attraverso l'innervazione sensoriale della divisione oftalmica del nervo trigemino (V nervo cranico), e il ganglio cervicale superiore, attraverso l'innervazione di fibre nervose simpatiche.

In termini quantitativi, il numero di assoni sensoriali corneali è maggiore rispetto al numero di assoni simpatici. La cornea è infatti uno dei tessuti più sensibili del corpo; i nervi sensoriali sono responsabili della sensazione di secchezza, temperatura, tatto, dolore, e giocano un ruolo fondamentale nel riflesso palpebrale, nella guarigione delle ferite e nella produzione lacrimale.

Il nervo nasociliare, una branca della divisione oftalmica del nervo trigemino, entra nell'orbita attraverso la fessura orbitaria superiore e si ramifica in modo che da uno a tre nervi ciliari possano penetrare la sclera per diversi millimetri dal nervo ottico.

Nella loro corsa intraoculare, i nervi viaggiano nello spazio sovracoroideale dove ramificano diverse volte prima di raggiungere il limbo corneo-sclerale, dove le fibre nervose risulteranno arrangiate in un plesso circolare.

Una volta entrate nella cornea, le fibre nervose corrono in direzione quasi-radiale attraverso lo stroma intermedio, ramificando anteriormente a formare un denso plesso sub-epiteliale. Perforando la membrana di Bowman formano, a livello della lamina basale, il plesso nervoso sub-basale.

Dal plesso nervoso sub-basale le fibre nervose decorrono infine superficialmente, innervando lo strato basale dell'epitelio e terminando a livello superficiale.

La microscopia confocale, che verrà approfondita nel paragrafo 1.6, consente di studiare la morfologia dell'innervazione corneale.

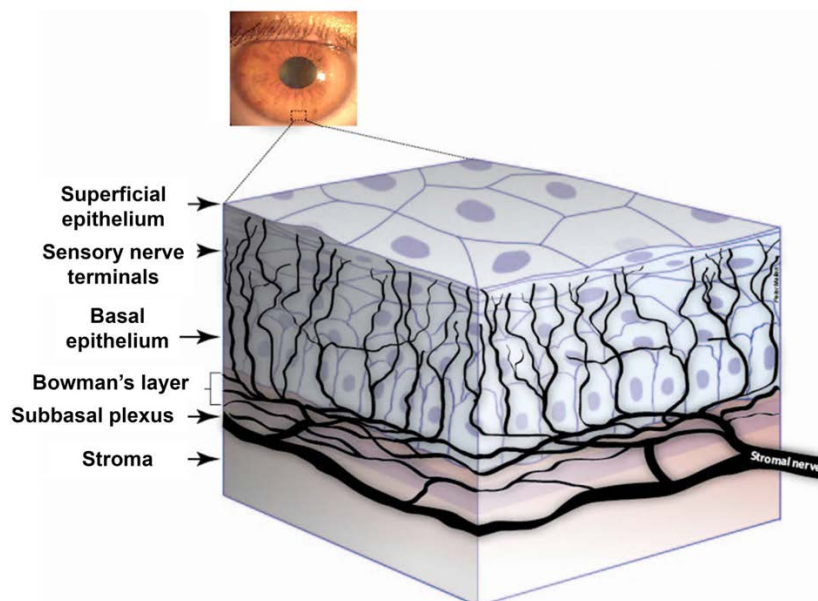


Figura 1.5. Rappresentazione schematica dell'innervazione corneale.

1.4 Cellule dendritiche corneali

Fino a tempi recenti si pensava che la cornea fosse un tessuto dotato di privilegio immunitario, a causa dell'assenza di cellule presentanti l'antigene (*Antigen Presenting Cells*, APCs). Tuttavia, studi recenti hanno dimostrato l'esistenza di APCs anche nella cornea, rivelando, in

particolare, un numero significativo di macrofagi nello stoma posteriore e di cellule dendritiche inattive nello stroma anteriore e nell'epitelio corneale, soprattutto tra lo strato basale e sub-basale [4].

Nell'epitelio, le cellule dendritiche residenti nello strato basale sono più numerose in periferia rispetto alla regione centrale, dove alcune di esse inseriscono i caratteristici processi membranosi tra le cellule epiteliali in modo simile alle terminazioni nervose sensoriali orientate verticalmente. Questo processo serve a campionare gli antigeni presenti nell'ambiente [5].

Le cellule dendritiche (*Dendritic Cells*, DCs) sono cellule presentanti l'antigene professionista, ovvero cellule caratterizzate da un'elevata espressione di molecole MHC II (complesso maggiore di istocompatibilità di classe II, *Major Histocompatibility Complex*) in fase infiammatoria.

Le DCs rappresentano un collegamento essenziale tra immunità innata e immunità acquisita: le APCs riconoscono l'antigene attraverso particolari recettori di membrana, chiamati *Pattern Recognition Receptors* (PRRs) (inclusi soprattutto i *Toll-like Receptors*, TLRs), dopodiché migrano nei linfonodi dove presentano l'antigene catturato e processato ai linfociti T. Le cellule dendritiche attivate up-regolano dunque le molecole costimolatorie e producono citochine, guidando l'attivazione e il *priming* dei linfociti T ancora vergini [6].

La maturazione delle DCs è legata sia ad un incremento dell'attività metabolica, dell'espressione di proteine di membrana e della secrezione di citochine pro-infiammatorie, sia ad un cambiamento nella morfologia (Figura 1.6): le cellule dendritiche attivate perdono le strutture adesive e riorganizzano il proprio citoscheletro attraverso la proiezione di processi membranosi in grado di aumentarne la motilità e consentirne dunque la migrazione negli organi linfoidi secondari [7].

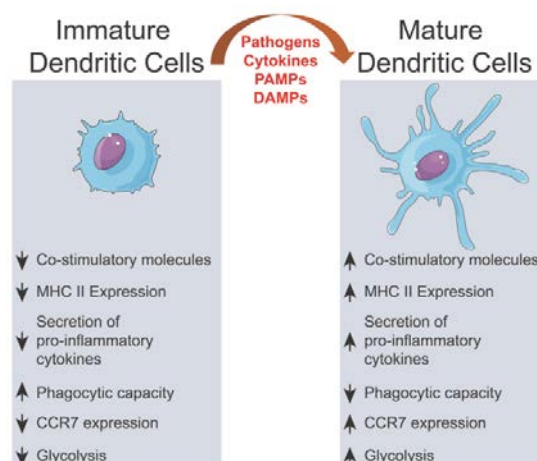


Figura 1.6. **Attivazione delle cellule dendritiche.** Segnali extra-cellulari innescano l'attivazione metabolica e strutturale delle cellule dendritiche.

Le cellule dendritiche derivano dal processo di ematopoiesi linfo-mieloide. Sulla base dei marcatori espressi, è possibile individuarne 2 tipi fondamentali, ognuno specializzato per rispondere a particolari patogeni e interagire con specifici gruppi di linfociti T [8]:

1. Cellule dendritiche plasmacitoidi (pDCs).

Le pDCs sono specializzate per riconoscere e rispondere a infezioni virali attraverso la rapida produzione di interferone di tipo I e III e la secrezione di citochine.

Sono caratterizzate da un array di recettori di membrana strettamente collegati nella regolazione della produzione di interferone di tipo I, come i marcatori CD303 e CD304.

2. Cellule dendritiche mieloidi o convenzionali (cDCs).

Le cDCs possiedono diversi meccanismi in grado di mediare un efficiente riconoscimento di antigeni virali e intracellulari. Possiedono infatti la capacità intrinseca di cross-presentare gli antigeni esprimendo molecole MHC di classe I in grado attivare i linfociti cito-tossici CD8⁺ e stimolare la secrezione di interleuchine da parte delle cellule T *helper*.

Studi recenti hanno riportato un'interessante correlazione tra densità e modificazioni morfologiche delle cellule dendritiche e densità nervosa nel plesso sub-basale corneale, suggerendo una potenziale interazione tra sistema immunitario e nervoso nella cornea [9].

In particolare, DCs e nervi sensoriali corneali sembrerebbero essere interconnessi sia funzionalmente che strutturalmente. Dal punto di vista funzionale, le cellule dendritiche rilasciano fattori neurotrofici, come il CNTF (fattore neurotrofico ciliare), in grado di promuovere la rigenerazione degli assoni e la preservazione dei nervi sensoriali. Dal punto di vista strutturale, si parla in termini di co-localizzazione nel plesso nervoso sub-basale: mentre le DCs con i caratteristici dendriti sono interconnesse con le terminazioni nervose, le DCs a forma arrotondata sono principalmente localizzate nei punti di biforcazione dei nervi e lungo le fibre verticali che attraversano la membrana basale terminando nell'epitelio [5].

La microscopia confocale in vivo consente di monitorare densità e morfologia delle cellule dendritiche, anche attraverso lo studio sistematico dell'interazione bidirezionale descritta precedentemente a livello del plesso nervoso sub-basale [9].

Nel seguito sono presentate alcune patologie cliniche di interesse, la cui relazione con le DCs è stata esaminata mediante microscopia confocale a livello del plesso nervoso sub-basale.

1.5 Patologie cliniche di interesse

1.5.1 Cheratite infettiva

La cheratite infettiva è un'infezione della cornea provocata dall'azione di agenti biologici come batteri, funghi e virus. L'opacità corneale, una delle principali conseguenze della cheratite infettiva, rappresenta la quarta causa a livello mondiale in grado di condurre a cecità, soprattutto nelle popolazioni più marginalizzate. È tuttavia riscontrabile anche nei paesi più sviluppati, spesso associata a traumi oculari o ad un uso scorretto delle lenti a contatto [10].

Uno studio pubblicato da *Cruzat et al* ha esaminato, tramite microscopia confocale in vivo, la cornea centrale di quattro gruppi di pazienti (controllo, cheratite batterica, cheratite fungina, cheratite da *Acanthamoeba*), dimostrando una diminuzione della densità nervosa nel plesso sub-basale e un incremento nel numero di cellule dendritiche nei soggetti affetti da cheratite infettiva. Sono stati inoltre evidenziati cambiamenti morfologici nelle DCs: mentre nel gruppo di controllo le cellule immature erano caratterizzate da pochi se non zero dendriti, nei pazienti patologici si è osservato un incremento nella dimensione e nel numero di processi membranosi, sinonimo di maturazione delle cellule (Figura 1.7) [11].

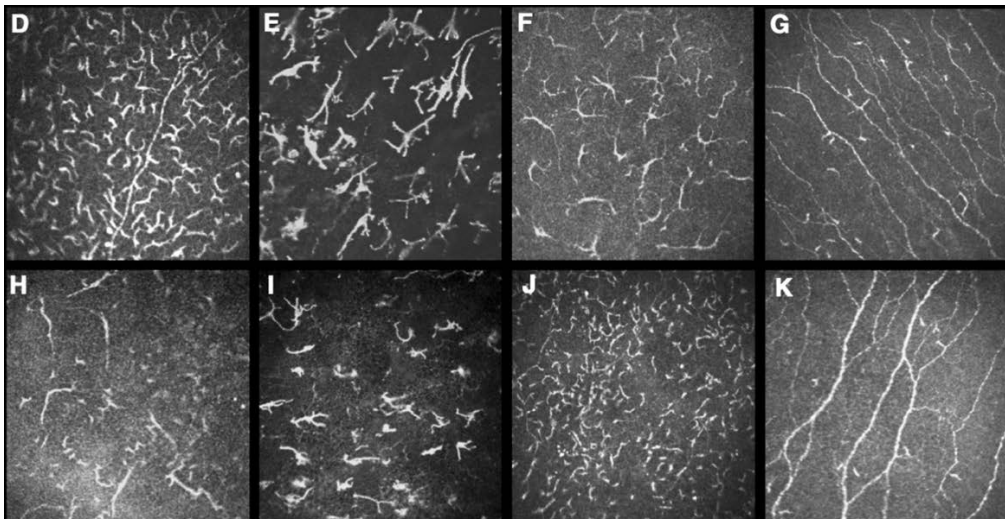


Figura 1.7. Immagini di microscopia confocale in vivo delle cellule dendritiche. (D,H) Cheratite batterica. (E,I) Cheratite fungina. (F,J) Cheratite da *Acanthamoeba*. (G,K) Controllo normale. In tutti e tre i gruppi patologici è possibile osservare un incremento della densità di cellule dendritiche.

1.5.2 Malattia di Behcet

La sindrome di *Behcet* è una vasculite cronica multisistemica recidivante. Benché originariamente descritta come malattia dermatologica, le principali cause di mortalità e morbilità risiedono nel coinvolgimento del sistema vascolare, neurologico e oculare [12].

In relazione all'ambiente oculare, molti studi hanno descritto neuropatie ottiche e patologie retiniche come principali conseguenze della malattia di *Behcet*. È stata inoltre riportata un'alterazione delle proprietà meccaniche della cornea, con ispessimento della regione centrale causato dal processo infiammatorio.

Nello studio pubblicato da *Bitirgen et al*, l'utilizzo della microscopia confocale ha consentito di valutare alterazioni nei nervi e nelle cellule immunitarie corneali in soggetti affetti dalla malattia di *Behcet*. La ricerca ha dimostrato un decremento nel numero e nella lunghezza delle fibre nervose e un incremento della densità delle cellule dendritiche (Figura 1.8) [13].

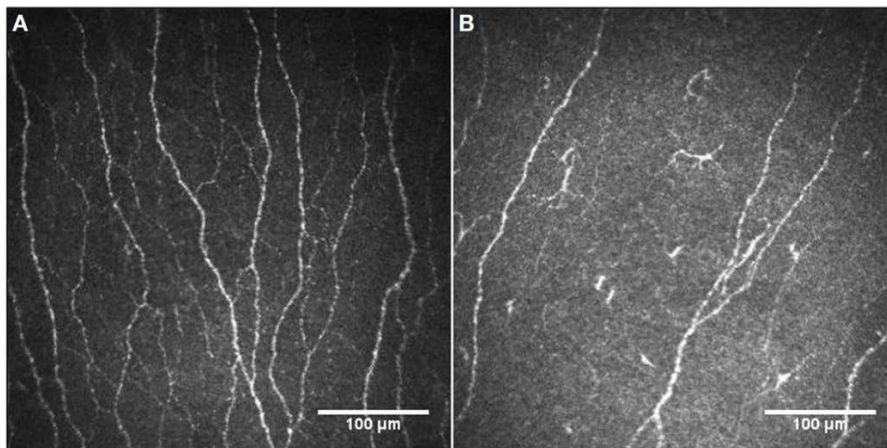


Figura 1.8. Immagini di microscopia confocale corneale del plesso nervoso sub-basale di un soggetto sano (A) e un soggetto affetto da sindrome di *Behcet* (B). Si osserva una riduzione delle fibre nervose e un aumento del numero di cellule dendritiche.

1.5.3 Malattia di Fabry

La malattia di *Fabry* è dovuta alla carenza dell'enzima α -galattosidasi A, che porta all'accumulo di glicosfingolipidi nei tessuti viscerali e nell'endotelio vascolare di tutto l'organismo, con danni a livello renale, cardiaco e del sistema nervoso centrale. Il sintomo principale è il dolore neuropatico, relazionato alla perdita di piccole fibre nervose. Nei soggetti

patologici, la quantificazione dei danni a carico dei nervi è dunque un aspetto importante per una diagnosi iniziale e per la valutazione della progressione della neuropatia. Studi di citometria a flusso hanno, inoltre, riportato una riduzione nel numero di cellule dendritiche circolanti nei vasi periferici, suggerendo una componente immunitaria nella patofisiologia della sindrome di *Fabry* [14].

Bitirgen et al hanno dimostrato, mediante microscopia confocale della cornea centrale, un decremento nei parametri dei nervi (lunghezza e densità) e un aumento del numero di cellule dendritiche (Figura 1.9), che potrebbe riflettere il meccanismo di extravasazione e migrazione delle DCs nei tessuti periferici [14].

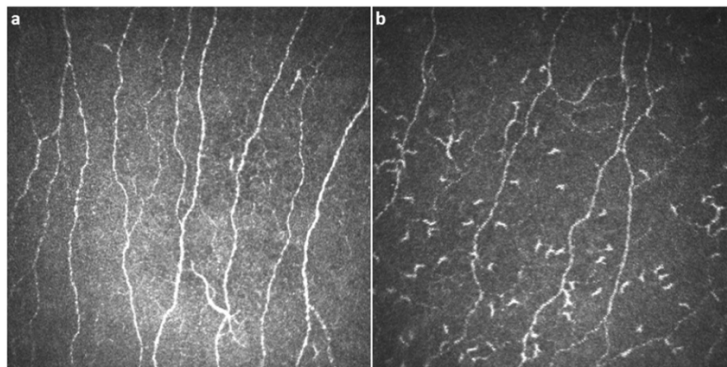


Figura 1.9. Immagini di microscopia confocale corneale del plesso nervoso sub-basale di un soggetto sano (A) e un soggetto affetto da sindrome di *Fabry* (B). Si osserva una riduzione delle fibre nervose e un aumento del numero di cellule dendritiche.

1.5.4 Sindrome dell'occhio secco

La sindrome dell'occhio secco è una malattia infiammatoria cronica che può essere innescata da numerosi fattori estrinseci ed intrinseci in grado di promuovere l'instabilità e l'iperosmolarità del film lacrimale. Modificazioni a carico della composizione lacrimale, combinate in alcuni casi con fattori sistemici, conducono a un ciclo infiammatorio che causa patologie epiteliali a carico della superficie oculare e anomalie neurosensoriali. Tutto ciò attiva il reclutamento di cellule infiammatorie e la maturazione delle cellule dendritiche [15].

Choi et al hanno dimostrato, mediante microscopia confocale della cornea centrale, un incremento significativo non solo nel numero di cellule dendritiche, ma anche delle dimensioni cellulari e della lunghezza dei processi dendritici nei soggetti patologici. In alcuni pazienti è stato inoltre riportato la vicinanza strutturale tra DCs e fibre nervose. Contrariamente, nel

gruppo di controllo tutte le DCs avevano corpo piccolo e arrotondato, e sono state tipicamente osservate separatamente dai nervi (Figura 1.10) [16].

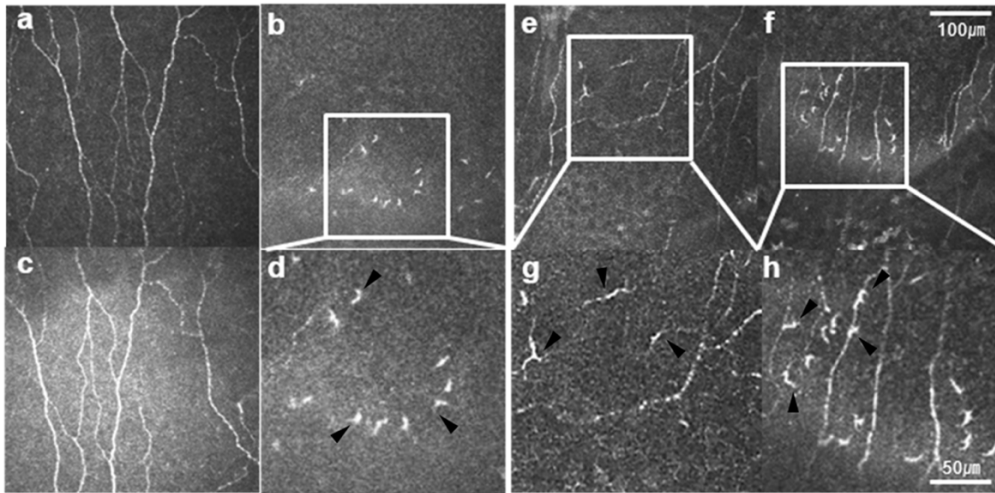


Figura 1.10. **Immagini di microscopia confocale corneale del plesso nervoso sub-basale e delle cellule dendritiche in soggetto di controllo (a sinistra) e soggetto patologico (a destra).** È possibile apprezzare un cambiamento non solo del numero ma anche della morfologia delle cellule dendritiche.

1.5.5 Diabete mellito di tipo 2

Il diabete mellito di tipo 2 è una malattia cronica su base multifattoriale caratterizzata dalla combinazione di resistenza all'insulina periferica e alterata secrezione di insulina da parte delle cellule beta pancreatiche.

Tra le complicazioni del diabete, quelle a carico dell'occhio sono spesso insidiose e progressive. Manifestazioni oculari di questa patologia riguardano non solo retinopatia o edema maculare, ma si estendono anche alla cornea, con particolare riferimento all'insorgenza di neuropatia diabetica periferica a carico del plesso nervoso sub-basale.

In studi recenti è stato suggerito un ruolo importante del processo infiammatorio nella patogenesi del diabete mellito di tipo 2, con coinvolgimento del sistema immunitario innato e acquisito.

Data questa associazione, *Lagali et al* hanno investigato la possibile associazione tra cellule dendritiche e sviluppo del diabete di tipo 2. Mediante microscopia confocale del plesso nervoso sub-basale, hanno dimostrato come la proporzione di cellule dendritiche mature nella cornea centrale di soggetti sani sia più che raddoppiata con l'insorgenza della patologia, incremento accompagnato da un proporzionale decremento nel numero di DCs immature, suggerendo

dunque un processo di maturazione delle stesse con lo sviluppo della malattia. È stato inoltre riportato un processo di arrangiamento delle DCs mature in *cluster* di cellule (Figura 1.11), suggerendo una mobilitazione delle stesse in seguito all'alterazione del metabolismo del glucosio [17].

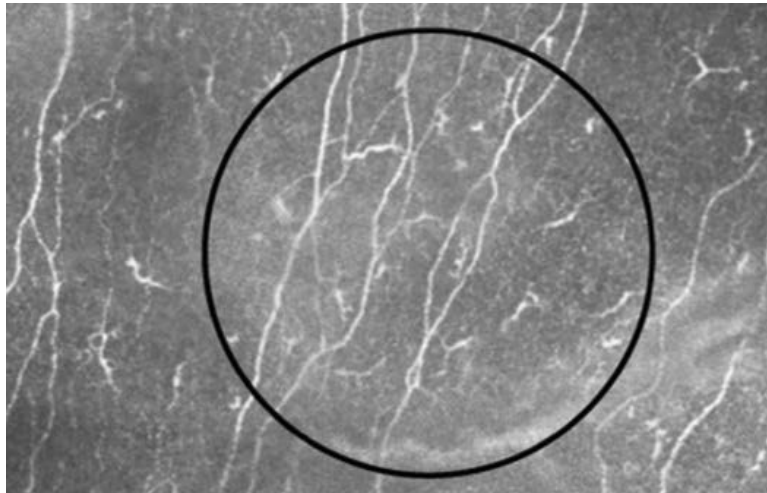


Figura 1.11. **Immagine di microscopia confocale corneale del plesso nervoso sub-basale di un soggetto patologico.** All'interno del cerchio nero è possibile apprezzare un *clustering* di DCs mature.

1.6 Microscopia confocale in vivo (*In-vivo confocal microscopy - IVC*M)

La microscopia confocale sta diventando un *tool* indispensabile per lo studio non-invasivo della patofisiologia corneale. Si tratta di una tecnologia che consente la visualizzazione in-vivo dell'architettura cellulare e delle fibre nervose della cornea, con tempi di acquisizione molto rapidi e livelli di magnificazione fino a 800 volte.

Il principio alla base della microscopia confocale consiste nel coniugare l'allineamento dei raggi luminosi focalizzati sul tessuto attraverso le lenti condensatrici con quelli riflessi dal tessuto e catturati dall'obiettivo. In altri termini, il sistema di illuminazione e quello di osservazione hanno punti focali comuni (da qui il termine "confocale") [18].

La microscopia ottica convenzionale produce immagini di bassa qualità a causa della luce riflessa e dispersa dalle strutture che non appartengono al piano focale.

Il principale vantaggio della microscopia confocale risiede invece nella capacità di ottenere immagini a profondità selezionate mediante il sezionamento ottico, rimuovendo le interferenze provenienti dai piani adiacenti a quello focale.

In figura 1.12 viene riportata una rappresentazione diagrammatica dei principi ottici della microscopia confocale: la luce bianca passa attraverso il primo diaframma (*pinhole*) e viene focalizzata nel piano focale selezionato attraverso le lenti condensatrici. La luce emessa dal tessuto viene dunque catturata dalla lente dell'obiettivo e deviata attraverso un secondo *pinhole*. In questo modo, viene impedito alla luce proveniente dai piani fuori fuoco di raggiungere il sistema di acquisizione dell'immagine.

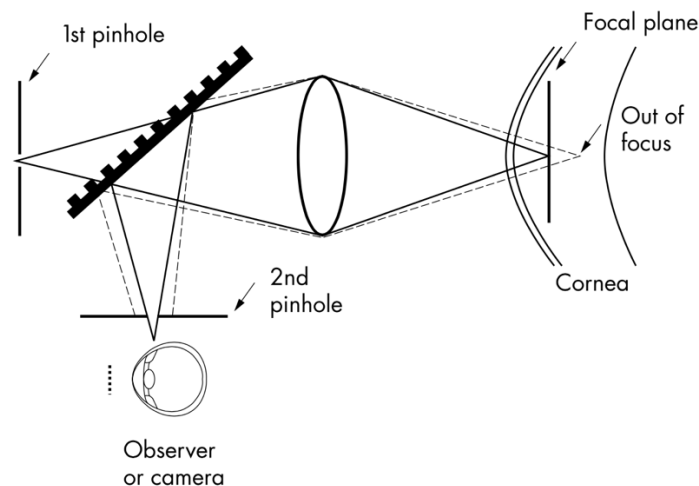


Figura 1.12. **Rappresentazione schematica del principio di funzionamento ottico del microscopio confocale.**

Il sistema consente di ottenere una risoluzione assiale di 5-10 micron e una risoluzione laterale dell'ordine di 1-2 micron [19]. Inoltre, l'acquisizione sequenziale di punti focali di profondità crescente consente di ottenere una ricostruzione tridimensionale delle strutture corneali.

A causa del limitato campo visivo fornito dalla tecnica di *imaging* confocale (la sorgente luminosa scansiona l'oggetto punto per punto), è necessario scansionare rapidamente il punto focale su tutto il campione al fine di ottenere la ricostruzione di un'immagine bidimensionale.

In base al meccanismo di scansione è possibile definire tre tipi di microscopi confocali [18]:

1. ***Tandem-Scanning Confocal Microscope (TSCM).***

Il TSCM utilizza un disco di *Nipkow* rotante caratterizzato da un insieme di *pinhole* microscopici arrangiati in una spirale archimedeica. Questo sistema confocale consente di ottenere un'ottima risoluzione laterale e assiale. Tuttavia, a causa delle numerose aperture, si osserva un incremento nella dispersione della luce: meno dell'1% raggiunge la cornea. Si rende dunque necessario l'utilizzo di sorgenti luminose molto forti.

2. *Slit-Scanning Confocal Microscope (SSCM).*

Il *Confoscan* (*Nidek Technologies*, Giappone) è un esempio di microscopio confocale *Slit-Scanning*.

Uno SSCM utilizza fenditure multiple verticali sia per l'illuminazione che per l'osservazione del campione. Il sistema, scansionando simultaneamente e parallelamente aree più grandi di tessuto lungo l'asse delle fenditure, consente di ridurre i tempi di acquisizione. In confronto con un TSCM, poiché la fenditura consente di raggiungere il campione con più luce, uno SSCM consente di ridurre l'intensità della sorgente luminosa e di ottenere immagini con una messa a fuoco e una luminosità migliore.

Il limite di questa tecnologia risiede in una risoluzione trasversale e assiale inferiore.

3. *Laser-Scanning Confocal Microscope (LSCM).*

Il LSCM utilizza una sorgente di luce coerente (un fascio laser) muovendola mediante specchi mobili per scansionare il campione.

Il dispositivo *Heidelberg Retinal Tomograph* (HRT) con il modulo *Rostock Cornea Module* prodotto dalla *Heidelberg Engineering*, in Germania, è l'unico LSCM disponibile a livello commerciale per lo studio in vivo della cornea.

Questo strumento consente di ottenere immagini con contrasto, messa a fuoco e qualità migliore rispetto allo SSCM. Permette in particolare la visualizzazione del plesso nervoso sub-basale con un'eccellente risoluzione. Inoltre, l'utilizzo di un laser con lunghezza d'onda di 670 nm consente la visualizzazione in vivo delle cellule dendritiche.

In Figura 1.13 è possibile visualizzare immagini di strutture corneali ottenute mediante SSCM (*Confoscan 4*, *NIDEK*, *Gamagori*, Giappone), nella colonna di sinistra, e mediante LSCM (*Heidelberg Retina Tomography II Rostock Corneal Module*, *Heidelberg Engineering*, *Heidelberg*, Germania) nella colonna di destra [20].

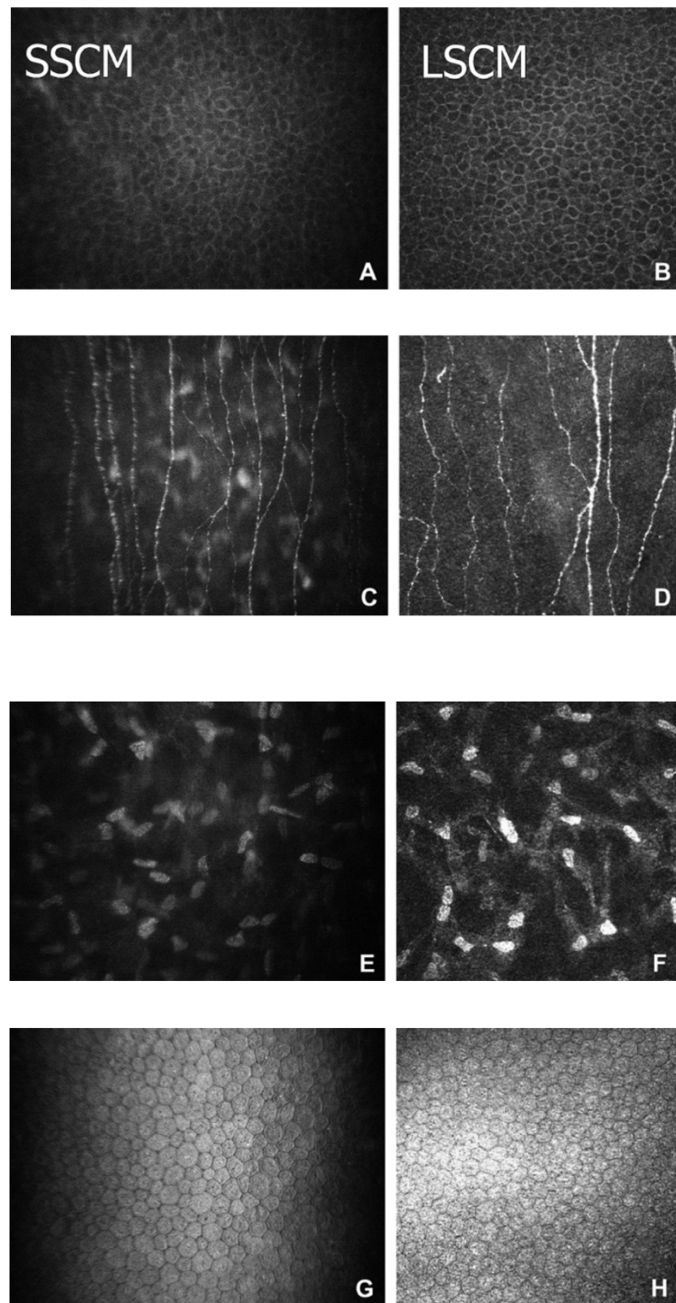


Figura 1.13. Immagini IVCM della cornea ottenute mediante SSCM (Confoscan 4, NIDEK, Gamagori, Giappone), nella colonna di sinistra, e LSCM (Heidelberg Retina Tomography II Rostock Corneal Module, Heidelberg Engineering, Heidelberg, Germany) nella colonna di destra. (A,B) Epitelio corneale. (C,D) Plesso nervoso sub-basale. (E,F) Stroma corneale. (G,H) Endotelio corneale.

CAPITOLO 2.

Deep Learning

2.1 Introduzione

Il termine *Machine Learning* (ML) fa riferimento al riconoscimento automatico di *pattern* significativi in dati grezzi, un *tool* diventato ad oggi comune in quasi tutti i compiti che richiedono l'estrazione di informazioni da grandi data set.

La società moderna è permeata da tecnologia basata sull'apprendimento automatico: basti pensare al *software antispy*, che impara come filtrare le mail, alle fotocamere digitali, che apprendono come rilevare i volti delle persone, alle automobili equipaggiate di sistemi di prevenzione degli incidenti basati su algoritmi di ML.

Spesso, la complessità e la variabilità dei *task* da automatizzare (es. il riconoscimento di determinate strutture all'interno delle immagini) fa sì che per un programmatore umano risulti complicato fornire una specificazione esplicita e dettagliata del *task* stesso.

Il *Machine Learning*, dunque, concerne l'implementazione di algoritmi dotati della capacità di apprendere dall'esperienza ed adattarsi alle situazioni, in un tentativo di imitazione degli esseri intelligenti, che, difatti, sono in grado di acquisire abilità attraverso un processo di apprendimento esperienziale piuttosto che attraverso un'aderenza a specifiche regole. In tal senso, quindi, si tratta di un campo di ricerca riconducibile ad un sottoinsieme dell'Intelligenza Artificiale, ramo delle scienze computazionali che concerne la progettazione e lo sviluppo di macchine intelligenti in grado di mimare le funzioni cognitive umane [21].

Le performance degli algoritmi di *Machine Learning* convenzionali dipendono strettamente dalla rappresentazione dei dati che viene loro fornita nella fase di addestramento, ovvero, dall'insieme di *features* estratte per lo specifico compito richiesto. Si tratta chiaramente di un'operazione non banale: per decenni, la costruzione di sistemi di *pattern-recognition* ha richiesto la progettazione esperta da parte di ingegneri di un sistema di estrazione delle *features*. Una soluzione a questo tipo di problema è stata data dal cosiddetto *Representation Learning* (RL), un approccio di apprendimento che tenta di identificare non solo la mappatura da rappresentazione ad output, ma anche la rappresentazione stessa dei dati grezzi.

In fase di progettazione di un algoritmo per l'apprendimento automatico, l'obiettivo fondamentale è dunque quello di separare i fattori di variazione che descrivono i dati osservati.

Tuttavia, si tratta spesso di caratteristiche di alto livello, astratte, esistenti più sotto forma di costrutti mentali che di dati oggettivi osservabili direttamente, e dunque difficilmente estraibili da dati grezzi.

Il *Deep Learning* (DL) si configura come soluzione a questo problema: si tratta di una forma specifica di ML, e in particolare di RL, caratterizzata da molteplici livelli di rappresentazione, ottenuti dalla combinazione di moduli semplici ma non lineari ciascuno dei quali trasforma la rappresentazione da un livello più basso e interpretabile, ad uno più alto ed astratto, consentendo quindi l'apprendimento di caratteristiche complesse partendo da elementi semplici [22].

Il *Deep Learning* esegue dunque un "apprendimento *end-to-end*", in cui un modello impara automaticamente come elaborare dati grezzi e svolgere un'attività (es. una classificazione).

La maggior parte dei metodi di DL utilizza le architetture di reti neurali. La ricerca nel campo del ML, infatti, ha trovato la principale fonte di ispirazione nel funzionamento del cervello umano: il tentativo è, quindi, quello di imitare la struttura neurale del sistema nervoso attraverso l'utilizzo di reti neurali artificiali (*Artificial Neural Networks*, ANNs), ovvero modelli di calcolo costituiti da unità fondamentali, dette neuroni artificiali. Tali unità operano in parallelo, organizzate in una struttura stratificata di nodi interconnessi, combinando diversi livelli di elaborazione.

I modelli di *Deep Learning* sono spesso denominati Reti Neurali Profonde (*Deep Neural Networks*, DNNs) in virtù dell'utilizzo di algoritmi basati su ANNs caratterizzate da diversi e numerosi strati (generalmente più di 5) [23].

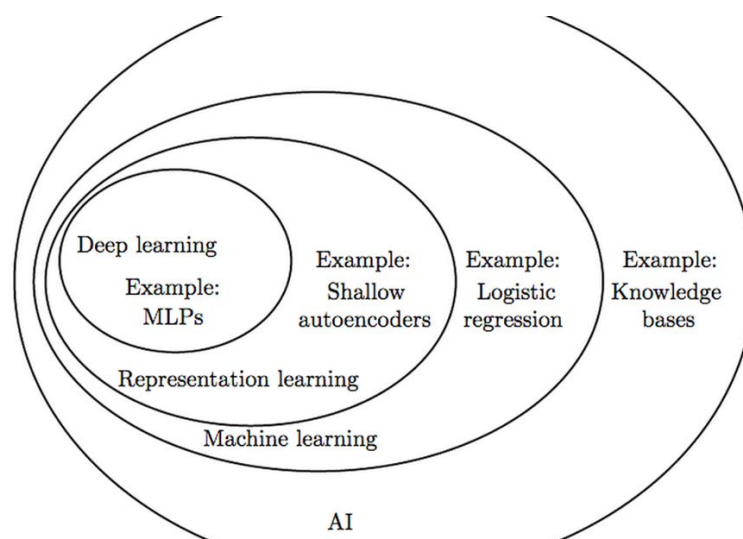


Figura 2.1. Relazione tra *Deep Learning* (DL), *Representation Learning* (RL) e *Machine Learning* (ML).

2.2 Neurone artificiale

Un neurone artificiale rappresenta l'unità fondamentale di elaborazione delle informazioni all'interno di una *Artificial Neural Network*.

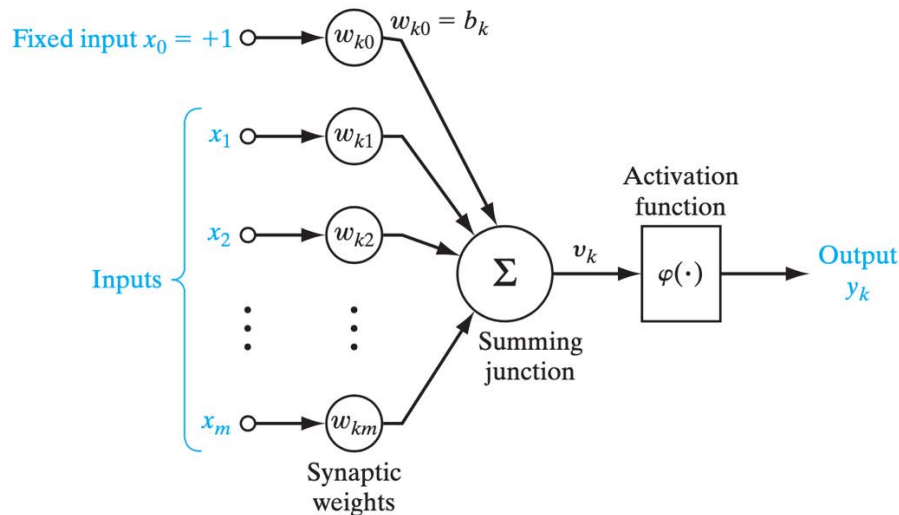


Figura 2.2. Modello di neurone artificiale.

La Figura 2.2 mostra il modello non lineare di un neurone artificiale, in cui è possibile identificare 3 elementi fondamentali [24]:

1. Un insieme di sinapsi, o connessioni sinaptiche, ognuna delle quali caratterizzata da un peso che determina l'importanza della sinapsi stessa nel *task*.
In particolare, un segnale x_j in ingresso alla sinapsi j connessa al neurone k viene moltiplicato per il peso sinaptico w_{kj} .
2. Un sommatore, un nodo che esegue una somma dei segnali di input pesata dalla forza sinaptica di ciascun neurone (w_{kj}).
3. Una funzione di attivazione $\varphi(\cdot)$, il cui scopo è limitare l'ampiezza del segnale in uscita da un neurone, determinandone dunque il comportamento in base al livello di eccitazione.

Il modello di neurone artificiale proposto, inoltre, include un ulteriore parametro detto *bias* (nel seguito indicato con b_k o w_{k0}). Il *bias* si considera collegato a un ingresso fittizio x_0 con valore unitario utile per tarare il punto di lavoro ottimale del neurone.

In termini matematici, è possibile descrivere il neurone k mediante la coppia di equazioni:

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (2.1)$$

$$y_k = \varphi(v_k) \quad (2.2)$$

dove il termine v_k indica il potenziale di attivazione del neurone (livello di eccitazione globale). Come accennato, la funzione di attivazione $\varphi(v_k)$ definisce l'output di un neurone in termini di campo locale indotto.

È possibile identificare due tipi fondamentali di funzione di attivazione:

1. Funzione soglia o Funzione di Heaviside (Figura 2.3), per cui vale:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \quad (2.3)$$

L'output del neurone k è espresso dunque come:

$$y_k = \begin{cases} 1 & \text{se } v_k \geq 0 \\ 0 & \text{se } v_k < 0 \end{cases} \quad (2.4)$$

Nella computazione neuronale si fa riferimento a tale neurone come al modello McCulloch-Pitts.

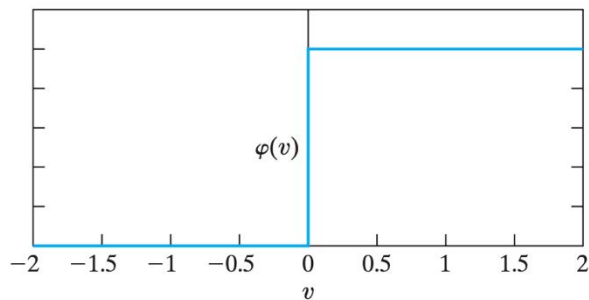


Figura 2.3. **Funzione soglia o Funzione di Heaviside.**

2. Funzione sigmoide, la forma più comune di funzione di attivazione utilizzata per la costruzione di una rete neurale, un cui esempio importante è dato dalla funzione logistica, definita da:

$$\varphi(v) = \sigma(v) = \frac{1}{1 + \exp(-av)} \quad (2.5)$$

in cui il parametro di forma a determina la pendenza della funzione (Figura 2.4).

Solitamente è tuttavia desiderabile avere una funzione di attivazione il cui *range* sia compreso nell'intervallo $[-1,1]$. Si utilizza quindi la funzione tangente iperbolica, ottenuta dalla funzione logistica a seguito di una trasformazione di scala e di una traslazione:

$$\varphi(v) = \tanh(v) = 2 \sigma(v) - 1 \quad (2.6)$$

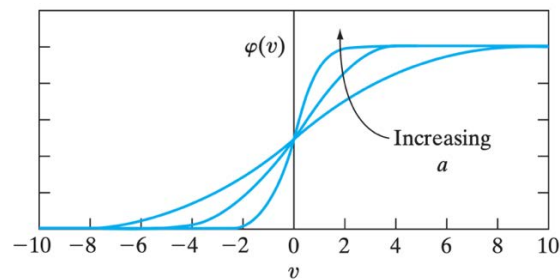


Figura 2.4. Funzione di attivazione logistica.

2.3 Architettura di rete neurale *Feed-Forward*

Le reti neurali artificiali sono costituite da gruppi di neuroni artificiali interconnessi tra loro e organizzati in una struttura stratificata.

L'architettura *Feed-Forward Neural Network* (FFNN) è la tipologia di rete neurale più diffusa e utilizzata, in cui i neuroni sono disposti in un livello di input, un livello di output, e uno o più livelli nascosti (*hidden*).

Si consideri per semplicità la topologia di rete proposta in Figura 2.5 [25].

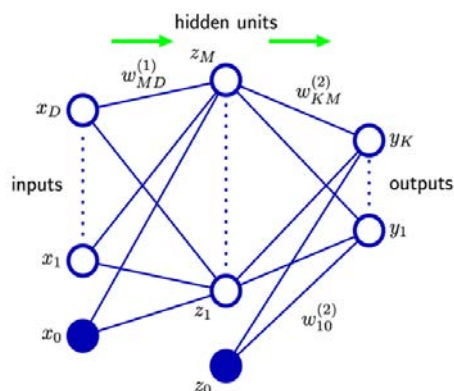


Figura 2.5. Architettura *feed-forward* di una rete neurale *fully-connected*.

Nel primo strato del modello, la combinazione lineare delle D variabili di ingresso x_1, \dots, x_D porta a definire il potenziale di attivazione per ciascuno degli M neuroni dello strato intermedio:

$$v_j = \sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)}, \quad j = 1, \dots, M \quad (2.7)$$

dove, continuando a seguire la notazione del paragrafo precedente, $w_{ji}^{(1)}$ indica il peso della connessione sinaptica tra il neurone i e il neurone j mentre $w_{j0}^{(1)}$ il termine di *bias*.

Ciascuna attivazione è quindi trasformata attraverso una funzione di attivazione $h(\cdot)$ non lineare, differenziabile, definita generalmente da una funzione sigmoidea:

$$z_j = h(v_j) \quad (2.8)$$

Questi valori sono nuovamente combinati linearmente per ottenere il livello di eccitazione globale di ciascuno dei K neuroni di uscita:

$$v_k = \sum_{j=1}^M w_{kj}^{(2)} z_j + w_{k0}^{(2)}, \quad k = 1, \dots, K \quad (2.9)$$

Infine, è possibile ottenere le uscite della rete:

$$y_k = \varphi(v_k), \quad k = 1, \dots, K \quad (2.10)$$

La funzione di attivazione dell'ultimo livello dipenderà in generale dallo scopo di utilizzo della rete neurale. Ad esempio, come si vedrà nel paragrafo successivo, per un problema di classificazione multi-classe viene tipicamente utilizzata la funzione *softmax*.

È possibile quindi costruire la funzione complessiva della rete:

$$y_k(\mathbf{x}, \boldsymbol{\theta}) = \varphi\left(\sum_{j=1}^M w_{kj}^{(2)} h\left(\sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \quad (2.11)$$

dove $\boldsymbol{\theta}$ indica l'insieme di tutti i parametri addestrabili (pesi sinaptici e *bias*).

Il modello di rete neurale proposto è quindi semplicemente rappresentato da una funzione non lineare da un set di variabili di ingresso $\{x_i\}$ a un set di variabili di uscita $\{y_k\}$ controllata da un vettore $\boldsymbol{\theta}$ di parametri adattabili.

Si parla in particolare di FFNN in quanto il processo di valutazione di $y_k(\mathbf{x}, \boldsymbol{\theta})$ può essere interpretato come la propagazione in avanti delle informazioni attraverso il *network*, struttura in cui non sono consentite connessioni all'indietro o verso lo stesso livello in modo tale da assicurare che gli output siano funzioni deterministiche degli input.

Naturalmente, quanto esposto può essere esteso al caso in cui siano presenti più livelli nascosti nei quali ciascuna unità intermedia k computa una funzione data:

$$z_k = h(\sum_j w_{kj} z_j) \quad (2.12)$$

dove la sommatoria riguarda tutti i neuroni che inviano connessioni al nodo k .

Per un dato insieme di valori applicato all'ingresso della rete, successive applicazioni dell'equazione (2.12) permettono l'attivazione di tutti i neuroni del modello [25].

La rete neurale riportata in Figura 2.5, inoltre, è detta *fully connected* poiché ciascun nodo, in ciascun livello, è connesso ad ogni altro nodo appartenente al livello immediatamente successivo. Si parla invece di rete parzialmente connessa in caso di assenza di una o più connessioni sinaptiche tra unità di strati adiacenti.

Nel paragrafo 2.5 verrà descritta nel dettaglio una particolare tipologia di FFNN, ovvero la Rete Neurale Convoluzionale (CNN).

2.4 Addestramento supervisionato di una rete neurale

L'obiettivo fondamentale nell'addestramento di una rete neurale è l'ottimizzazione dei pesi sinaptici di ogni strato, ovvero, la determinazione dei parametri che consentono di ottenere la mappatura desiderata tra ingresso e uscita della rete.

In questo paragrafo verrà fatto esplicito riferimento allo scenario di apprendimento considerato per lo svolgimento di questa tesi, ovvero all'addestramento supervisionato per la classificazione multi-classe.

2.4.1 Classificazione multi-classe

Nel paradigma di addestramento supervisionato, il compito dell'algoritmo di apprendimento è quello di determinare, dato un insieme di esempi (*training set*), i parametri del modello che meglio consentono di ottenere la relazione desiderata tra input e output. Il *Training Set* è quindi

l'esperienza dalla quale si vuole apprendere: consiste in un insieme di dati di ingresso (rappresentati da vettori o matrici), ciascuno dei quali è etichettato con l'uscita reale che si desidera mappare (anche in questo caso è possibile avere singoli numeri, vettori o matrici).

Nel contesto specifico di addestramento supervisionato di una rete neurale, il problema può essere formalizzato come un processo di ottimizzazione numerica il cui obiettivo è quello di determinare i pesi e *bias* delle connessioni sinaptiche in grado di minimizzare l'errore tra output desiderato (output reale) e output predetto dal modello. L'errore viene espresso attraverso una funzione costo, denominata funzione di perdita (*loss function*), che verrà minimizzata durante la fase di *training*.

Si parla in particolare di classificazione multi-classe quando l'obiettivo è quello di assegnare un vettore di ingresso \mathbf{x} ad una di K classi discrete C_k , con $k = 1, \dots, K$ e $K > 2$. Una volta fornito alla rete il *training set* $D = \{\mathbf{x}, \mathbf{t}\}_{n=1}^N$, dove \mathbf{t} rappresenta il vettore *target* definito per rappresentare l'etichetta associata ad ogni classe, si dovrà minimizzare la *loss function* con l'obiettivo di trovare la rete che meglio mappi l'input \mathbf{x} sull'output \mathbf{t} [25].

In particolare, \mathbf{t} è un vettore di lunghezza K per il quale è conveniente utilizzare lo schema di codifica *1-of-K*, ovvero una rappresentazione binaria tale che se la classe di appartenenza del *pattern* di ingresso è C_j , allora tutti gli elementi t_k di \mathbf{t} sono settati a zero tranne l'elemento t_j , che assume valore unitario.

L'approccio più semplice alla classificazione consiste nel costruire una funzione discriminante $f: \mathbb{R}^N \rightarrow \{1, \dots, K\}$ in grado di assegnare ogni vettore di ingresso \mathbf{x} direttamente alla specifica classe.

È tuttavia possibile adottare una più potente prospettiva probabilistica per il problema in esame, modellando la distribuzione di probabilità condizionata $p(C_k / \mathbf{x})$ in una fase inferenziale, per poi usare tale distribuzione per prendere la decisione di assegnazione ottima.

Tra i vari metodi per determinare $p(C_k / \mathbf{x})$, una tecnica consiste nel modellare la stessa attraverso un modello parametrico e ottimizzarne i parametri utilizzando il *training set*.

La predizione del modello viene dunque espressa attraverso il seguente modello lineare generalizzato:

$$p(C_k | \mathbf{x}, \boldsymbol{\theta}) = y(\mathbf{x}) = \varphi(\mathbf{w}^T \mathbf{x} + b) \quad (2.13)$$

dove, nel contesto di una rete neurale, \mathbf{w} rappresenta il vettore dei pesi sinaptici, b il termine di *bias* e $\varphi(\cdot)$ la funzione di attivazione non lineare.

Nella discussione di modelli generativi per la classificazione multi-classe, per un'ampia classe di distribuzioni le probabilità a posteriori sono date dalla funzione di attivazione *softmax*:

$$p(C_k|\mathbf{x}, \boldsymbol{\theta}) = y_k(\mathbf{x}) = \frac{e^{(\mathbf{w}_k^T \mathbf{x} + b_k)}}{\sum_{j=1}^K e^{(\mathbf{w}_j^T \mathbf{x} + b_j)}} \quad (2.14)$$

$$0 \leq y_k \leq 1; \quad \sum_{k=1}^K y_k = 1$$

dove \mathbf{w}_k indica il vettore dei pesi che conduce al nodo di uscita associato alla classe k [25,26]. I parametri $\boldsymbol{\theta}$ sono fittati attraverso la procedura di ottimizzazione mediante massimizzazione della *likelihood* o, equivalentemente, mediante minimizzazione della *negative-log-likelihood*:

$$\boldsymbol{\theta}_{ML} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} - \sum_{n=1}^N \sum_{k=1}^K \log [p(C_k|\mathbf{x}_n, \boldsymbol{\theta})] \quad (2.15)$$

che risulta nella definizione della funzione costo *cross-entropy*, ovvero, utilizzando per \mathbf{t} la convenzione descritta in precedenza, nella seguente formulazione:

$$\operatorname{loss}(\boldsymbol{\theta}) = E(\boldsymbol{\theta}) = - \sum_{n=1}^N \sum_{k=1}^K t_{nk} \ln y_{nk} \quad (2.16)$$

dove N è il numero di campioni del *training set*, K il numero di classi, t_{nk} l'indicatore che il campione n -esimo appartiene alla classe k -esima, y_{nk} l'output per il campione n appartenente alla classe k , valore derivante dalla funzione *softmax* [25].

2.4.2 Stochastic Gradient Descent (SGD)

Come già esposto, l'obiettivo fondamentale nell'addestramento supervisionato di una rete neurale è quello di determinare i parametri del modello che consentano di minimizzare la funzione costo scelta.

Poiché tipicamente in un sistema di *Deep Learning* sono presenti migliaia di neuroni, è necessario interfacciarsi con procedure di ottimizzazione numerica iterative.

Nello scenario di apprendimento supervisionato la tecnica di ottimizzazione più utilizzata è la discesa del gradiente (*gradient descent* GD o *steepest descent*).

L'algoritmo di discesa del gradiente è una tecnica di ottimizzazione di tipo locale, e consiste nel cercare il minimo di una funzione data, in questo caso la funzione di *loss*, muovendo nella direzione del gradiente (anzi l'antigradiente), come riportato in Figura 2.6. Il metodo del gradiente, quindi, prevede di partire da una soluzione iniziale (arbitraria o no) e di aggiornare i pesi della rete iterativamente secondo la regola seguente:

$$\boldsymbol{\theta}^{(\tau+1)} = \boldsymbol{\theta}^{(\tau)} - \eta \nabla E(\boldsymbol{\theta}^{(\tau)}) \quad (2.17)$$

dove τ indica il numero di iterazione, $\eta > 0$ è il *learning rate* (ovvero lo *step* che si intende fare ad ogni iterazione), $\boldsymbol{\theta}$ è il vettore dei parametri ed $E(\boldsymbol{\theta})$ indica la loss function.

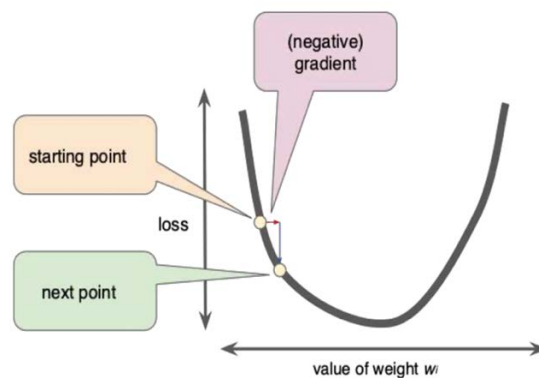


Figura 2.6. **Ottimizzazione iterativa mediante discesa del gradiente della funzione costo.**

Il *learning rate*, come anticipato, controlla lo *step* d'aggiornamento, ovvero, di quanto spostarsi lungo l'antigradiente ad ogni iterazione, ed è probabilmente il parametro più interessante e delicato in fase di addestramento di una rete. La scelta di un valore troppo elevato, infatti, potrebbe portare l'algoritmo a non convergere mai, mentre un valore troppo ridotto potrebbe allungare drasticamente i tempi computazionali richiesti per raggiungere la soluzione, o portare l'algoritmo ad una soluzione sub-ottima (minimo locale)[25,27,28].

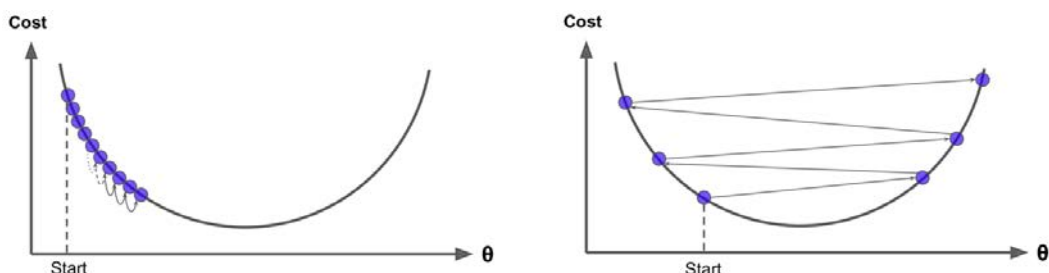


Figura 2.7. **Learning rate.** A sinistra, l'utilizzo di un *learning rate* troppo basso richiede molti aggiornamenti dei parametri per raggiungere un punto di minimo. A destra, un *learning rate* troppo alto porta l'algoritmo a divergere.

Negli algoritmi di discesa del gradiente convenzionali, il gradiente della funzione costo viene valutato utilizzando l'intero *training set*. Tuttavia, nella pratica, una procedura molto utilizzata è la cosiddetta discesa stocastica del gradiente (*Stochastic Gradient Descent, SGD*): ad ogni iterazione, l'algoritmo SGD valuta il gradiente ed aggiorna i parametri della rete utilizzando un sottoinsieme dei dati di addestramento (*mini-batch*). In ciascuna iterazione viene utilizzato un diverso sottoinsieme.

Il passaggio completo dell'algoritmo sull'intero *training set* rappresenta un'epoca.

Il termine stocastico deriva dal fatto che l'utilizzo di un set ridotto di dati di addestramento produce una stima rumorosa del gradiente altrimenti ottenuto utilizzando tutti gli esempi [29]. Diversi algoritmi di ottimizzazione derivano dalla SGD, tra cui uno dei principali è il metodo Adam.

2.4.2.1 ADAM (*Adaptive Moment Estimation*)

Il metodo ADAM è una tecnica di ottimizzazione stocastica efficiente che calcola *learning rate* individuali adattativi per diversi parametri attraverso le stime dei momenti di primo e secondo ordine dei gradienti. In altri termini, è un algoritmo di ottimizzazione che tenta di migliorare l'addestramento della rete utilizzando *learning rate* in grado di adattarsi automaticamente e singolarmente, parametro per parametro, in base alla *loss function* da minimizzare. Per farlo l'algoritmo memorizza la media con decadimento esponenziale del gradiente (m) e del quadrato del gradiente (v) [30].

Di seguito le equazioni che descrivono il metodo:

$$m^{(\tau)} = \beta_1 m^{(\tau-1)} + (1 - \beta_1) \nabla E(\boldsymbol{\theta}) \quad (2.18)$$

$$v^{(\tau)} = \beta_2 v^{(\tau-1)} + (1 - \beta_2) [\nabla E(\boldsymbol{\theta})]^2 \quad (2.19)$$

dove gli iperparametri β_1 e $\beta_2 \in [0, 1)$ controllano le velocità di decadimento esponenziale.

Le medie mobili calcolate vengono poi utilizzate per aggiornare i parametri della rete:

$$\boldsymbol{\theta}^{(\tau+1)} = \boldsymbol{\theta}^{(\tau)} - \frac{\eta m^{(\tau)}}{\sqrt{v^{(\tau)} + \epsilon}} \quad (2.20)$$

dove ϵ rappresenta un *offset* aggiunto per evitare che il denominatore si annulli.

Il metodo di discesa del gradiente trova nell'algoritmo di *back-propagation* un'efficiente implementazione per il calcolo del gradiente della funzione costo.

2.4.3 *Back-propagation*

Lo sviluppo dell'algoritmo di retro-propagazione dell'errore, a metà degli anni '80, rappresenta una pietra miliare nella storia delle reti neurali, introducendo un metodo computazionalmente efficiente per l'addestramento delle reti multistrato, in particolare per il calcolo del gradiente della funzione costo rispetto ai parametri della rete.

L'addestramento con la tecnica di *back-propagation* procede in due fasi [27,28]:

1. ***Forward phase.***

In questa fase i pesi delle connessioni sinaptiche della rete rimangono inalterati ed il segnale di input viene propagato in avanti attraverso il *network*, strato per strato, fino a raggiungere il livello di output. In questo passo le modifiche sono dunque confinate al potenziale di attivazione e all'uscita di ciascun neurone.

Nello strato finale, la funzione costo misura l'errore commesso negli output predetti dalla rete rispetto a quelli desiderati.

2. ***Backward phase.***

In questa fase, il segnale errore calcolato nello strato finale viene propagato all'indietro, e, utilizzando la regola di derivazione a catena, viene computato il gradiente della *loss function* per ogni parametro della rete.

Infine, i pesi sinaptici della rete sono aggiornati attraverso il metodo di discesa del gradiente.

Viene ora esposto con maggior dettaglio il metodo di retro-propagazione dell'errore [25].

Si parte anzitutto dalla considerazione che molte funzioni costo di interesse, come quella definita dalla *maximum likelihood*, comprendono una somma di termini, uno per ogni dato nel *training set*, in modo tale che:

$$E(\boldsymbol{\theta}) = \sum_{n=1}^N E_n(\boldsymbol{\theta}) \quad (2.21)$$

Di seguito verrà dunque considerato il problema di valutare $\nabla E_n(\boldsymbol{\theta})$ per uno di tali termini, procedura che può essere utilizzata direttamente per metodi di ottimizzazione sequenziale come la discesa stocastica del gradiente.

Richiamando le equazioni del paragrafo precedente, in una rete neurale *feed-forward* ogni unità j calcola una somma pesata dei suoi ingressi nella forma:

$$v_j = \sum_i w_{ji} z_i \quad (2.22)$$

dove z_i rappresenta l'attivazione di un neurone che invia una connessione all'unità j , e w_{ji} è il peso associato a tale collegamento.

L'attivazione dell'unità j è data dunque da:

$$z_j = h(v_j) \quad (2.23)$$

Dove, al solito, $h(\cdot)$ rappresenta una funzione di attivazione non lineare e differenziabile.

Si supponga quindi che, attraverso successive applicazioni della (2.22) e (2.23), l'informazione sia stata propagata in avanti attraverso la rete con conseguente attivazione di tutte le unità nascoste e di uscita (*forward propagation*).

Si consideri ora la valutazione della derivata di E_n rispetto al peso w_{ji} attraverso l'applicazione della regola di derivazione a catena per le derivate parziali:

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial v_j} \frac{\partial v_j}{\partial w_{ji}} \quad (2.24)$$

È possibile utilizzare la seguente notazione:

$$\delta_j = \frac{\partial E_n}{\partial v_j} \quad (2.25)$$

dove δ_j viene detto gradiente locale o semplicemente errore.

Utilizzando la (2.22) è possibile scrivere:

$$\frac{\partial v_j}{\partial w_{ji}} = z_i \quad (2.26)$$

e quindi, sostituendo la (2.25) e la (2.26) nella (2.24) si ottiene l'equazione:

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i \quad (2.27)$$

che afferma come la derivata richiesta possa essere ottenuta semplicemente moltiplicando il valore di δ corrispondente all'unità al terminale di uscita del peso in esame per il neurone al terminale di ingresso della connessione sinaptica ($z = 1$ nel caso in cui il parametro da addestrare sia il *bias*).

In particolare, per le unità di output vale:

$$\delta_k = y_k - t_k \quad (2.28)$$

mentre per i nodi nascosti è possibile utilizzare nuovamente la regola di derivazione a catena:

$$\delta_j = \frac{\partial E_n}{\partial v_j} = \sum_k \frac{\partial E_n}{\partial v_k} \frac{\partial v_k}{\partial v_j} \quad (2.29)$$

dove la sommatoria è estesa a tutte le unità k a cui il neurone j invia delle connessioni (Figura 2.7).

Sostituendo infine la definizione di δ data da (2.25) in (2.29) e facendo uso di (2.22) e (2.23) si ottiene la formula di *back-propagation*:

$$\delta_j = h'(v_j) \sum_k w_{kj} \delta_k \quad (2.30)$$

che afferma come il valore di δ_j per un determinato neurone nascosto possa essere ottenuto propagando all'indietro i valori di δ_k delle unità k a cui j invia connessioni (Figura 2.7).

Per concludere, è utile riassumere l'algoritmo di retro-propagazione dell'errore:

1. Dato un vettore di ingresso x_n alla rete, propaga in avanti l'informazione utilizzando la (2.22) e (2.23).
2. Valuta δ_k per tutti i nodi di uscita utilizzando la (2.28).
3. Retro-propaga i gradienti locali utilizzando la (2.30) per ottenere l'errore δ_j per ogni neurone nascosto della rete.
4. Utilizza la (2.27) per ottenere le derivate richieste.

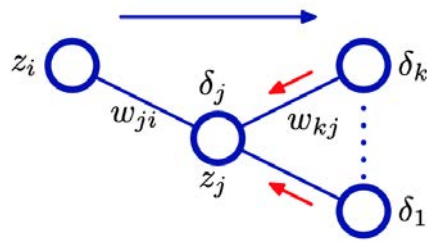


Figura 2.7. Calcolo del gradiente locale per l'unità nascosta j mediante retro-propagazione degli errori dai neuroni k a cui j invia delle connessioni.

2.4.4 Overfitting

L'*overfitting* (“sovr-adattamento”) rappresenta un problema comune nell’addestramento supervisionato di modelli complessi come, ad esempio, le reti neurali profonde. Si tratta della situazione in cui la rete “memorizza” i dati di addestramento piuttosto che apprendere da essi alcune caratteristiche generali, che consentano di predire in modo accurato input mai visionati. Viene dunque a mancare la capacità di generalizzazione del modello.

Di seguito vengono proposte alcune contromisure volte a ridurre la situazione di *overfitting*.

2.4.4.1 Regolarizzazione

La tecnica di regolarizzazione consiste nell’intervenire nel processo di ottimizzazione dei parametri della rete aggiungendo un termine di penalità alla funzione costo da minimizzare:

$$E_R(\boldsymbol{\theta}) = E_D(\boldsymbol{\theta}) + \lambda \Omega(\mathbf{w}) \quad (2.31)$$

dove \mathbf{w} indica il vettore dei pesi e λ il coefficiente di regolarizzazione, il quale controlla la relativa importanza dell’errore dipendente dai dati di addestramento $E_D(\boldsymbol{\theta})$ e della funzione di regolarizzazione $\Omega(\mathbf{w})$.

Una delle forme più semplici ed utilizzate per la funzione $\Omega(\mathbf{w})$ definisce la cosiddetta regolarizzazione L2:

$$\Omega(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} \quad (2.32)$$

$$E_R(\boldsymbol{\theta}) = E_D(\boldsymbol{\theta}) + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \quad (2.33)$$

Questa particolare scelta del regolarizzatore è nota nella letteratura del *Machine Learning* come *weight decay* (decadimento del peso) in quanto introduce un termine di penalità alla funzione costo in modo che i pesi della rete convergano a valori assoluti vicini allo zero [25,31]. (Da notare come i termini di *bias* non vengano coinvolti nel processo di regolarizzazione).

2.4.4.2 Dropout

Con il termine *dropout* si intende lo spegnimento casuale dei neuroni nascosti durante la fase di addestramento della rete: le unità disattivate non contribuiscono alla fase di propagazione in avanti e non partecipano alla *back-propagation*.

Si tratta di un approccio alla regolarizzazione in quanto consente di ridurre il co-adattamento tra neuroni, impedendo loro di fare affidamento sulla presenza di altre specifiche unità; ogni neurone è perciò forzato ad apprendere *features* funzionali più robuste con conseguente incremento della propria potenza individuale [32,33].

2.4.4.3 Batch Normalization

L'addestramento di una DNN è reso ulteriormente complicato dal fatto che la distribuzione degli *input*, che alimentano ogni strato della rete, cambia durante il processo di apprendimento, in corrispondenza della modifica dei parametri dei livelli precedenti. Il fenomeno viene chiamato *internal covariate shift* [34].

Ciò rallenta l'allenamento della rete, richiedendo *learning rates* più bassi e un'attenta inizializzazione dei parametri.

Un approccio efficace a tale problematica è rappresentato dall'algoritmo di *batch-normalization* (BN), che consiste nell'includere all'interno dell'architettura del modello un processo di normalizzazione dei *mini-batch* di allenamento: la BN opera dunque sottraendo la media (μ_B) e dividendo per la deviazione standard (σ_B^2) di ciascun mini-gruppo di dati in modo da normalizzarlo.

Considerando di prendere come input i valori di \mathbf{x} su un mini-gruppo di dati $B = (x_{i...m})$, si ottiene:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (2.34)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (2.35)$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (2.36)$$

Una semplice operazione di normalizzazione potrebbe modificare ciò che un *layer* è in grado di rappresentare: per questo motivo, per ogni attivazione \hat{x}_i viene eseguita la seguente trasformazione

$$y_i = \gamma \hat{x}_i + \beta = BN_{\gamma, \beta}(x_i) \quad (2.37)$$

dove γ, β sono parametri addestrabili in grado dunque di ristabilire il potere di rappresentazione di ogni strato della rete [34].

La *batch-normalization* è un processo che consente di accelerare il processo di addestramento, abilitando l'utilizzo di *learning rates* più elevati, e può essere visto come contromisura all'*overfitting* in quanto permette a ciascun livello di apprendere autonomamente in maniera più indipendente dagli altri livelli.

2.4.4.4 Data augmentation

Al fine di evitare l'*overfitting* e incrementare la capacità di generalizzazione di un modello è importante collezionare più dati di addestramento possibile.

Nel contesto della classificazione di immagini, quando il *training set* a disposizione ha una cardinalità ristretta, un metodo semplice ma efficace è rappresentato dalla *data augmentation*, una tecnica che consiste nel processare i dati a disposizione elaborandoli mediante trasformazioni cosiddette *label-preserving* [33].

Le più comuni consistono nel tagliare, ruotare, riflettere o aggiungere rumore gaussiano alle immagini.

Le reti convoluzionali sono invarianti alla traslazione, perciò non avrebbe senso fare *data augmentation* con operazioni di traslazione

2.4.4.5 Validazione

Per stabilire la capacità di generalizzazione di un modello è necessario valutarne la *performance* su dati non inclusi nel set di addestramento e dunque mai visionati dalla rete.

In caso di data set di cardinalità elevata, un approccio comune è quello di suddividerlo in 3 parti:

- una (tipicamente l'80%) per l'addestramento della rete, il *training set*,
- un'altra, detta *test set*, indipendente dalla precedente (ma che segua la stessa distribuzione di probabilità), che permette di individuare i casi di *overfitting* e valutare le *performance* della rete addestrata
- Il *validation set*, utilizzato per la ricerca degli iperparametri migliori

Quando i dati di addestramento hanno numerosità ridotta (non si riescono a formare i tre set con abbastanza variabilità), una tecnica popolare è la *K-fold cross-validation*: consiste nel suddividere l'intero dataset a disposizione in K sottoinsiemi di uguale dimensione ed eseguire K diversi cicli di *training*, in ognuno dei quali si esclude un diverso sottoinsieme tra i K e lo si usa per la validazione. Le prestazioni del modello vengono misurate facendo la media degli errori quadrati sul *validation set* per ognuna delle K ripetizioni.

Questa tecnica permette di testare modelli simili tra loro su insiemi di dati diversi e validare quindi un modello generale addestrato su tutti i dati a disposizione.

Se $K = N$, dove N indica il numero di esempi nel *training set*, il metodo viene detto *leave-one-out-cross-validation*: nel K-esimo ciclo di addestramento, vengono utilizzati tutti i dati ad eccezione del K-esimo, che viene invece usato per la validazione [35].

2.5 Reti Neurali Convolutionali

La rete neurale convoluzionale (*Convolutional Neural Network*, CNN) è uno degli algoritmi più comuni di *Deep learning*. È utilizzato per l'elaborazione di dati caratterizzati da una particolare topologia a griglia: una CNN è in grado di enfatizzare relazioni locali a partire da strutture di adiacenza presenti nei dati, attraverso un apprendimento automatico e adattativo di *pattern* da basso ad alto livello [36].

Le CNNs rappresentano dunque il principale modello utilizzato nel campo della *computer vision* e in generale nelle applicazioni che richiedono il riconoscimento di oggetti e la visione artificiale.

Si tratta di un'architettura ispirata dalla struttura biologica della corteccia visiva, in cui è presente una gerarchia di due tipi fondamentali di cellule: cellule semplici e complesse. Le cellule semplici reagiscono a *pattern* primitivi presenti in sotto regioni del campo visivo, chiamate campi recettivi, mentre le cellule complesse sintetizzano le informazioni provenienti dalle prime in modo da identificare strutture più complesse [29]. Allo stesso modo, i neuroni presenti in uno strato convoluzionale sono connessi a sotto-regioni del livello precedente, e non risentono dei segnali localizzati al di fuori di quell'area. I campi recettivi possono inoltre sovrapporsi: i neuroni di una CNN producono dunque risultati spazialmente correlati.

È possibile dunque identificare la principale differenza con una FFNN: mentre in una rete neurale *fully connected* il numero di parametri da apprendere aumenta con l'aumentare delle dimensioni dell'input, una rete neurale convoluzionale riduce il numero di parametri grazie al ridotto numero di connessioni, pesi condivisi e sotto-campionamento.

Una rete neurale convoluzionale può avere decine o centinaia di *layer*, ognuno dei quali è costituito da diversi filtri adibiti a rilevare diverse *features* e costruire le corrispondenti *feature maps*.

Nel caso delle immagini, infatti, a ciascun ingresso vengono applicati filtri (detti *kernel*) e, tramite un'operazione di convoluzione, vengono generate le *feature maps*. Quest'ultime saranno utilizzate come input per il *layer* successivo.

I filtri dei primi *layer* cercano *features* molto semplici (es. bordi), per assumere forme via via più complesse, in grado di definire in modo univoco l'oggetto.

Nel seguito vengono descritti i principali strati che caratterizzano l'architettura di una rete neurale convoluzionale, riportata in Figura 2.8.

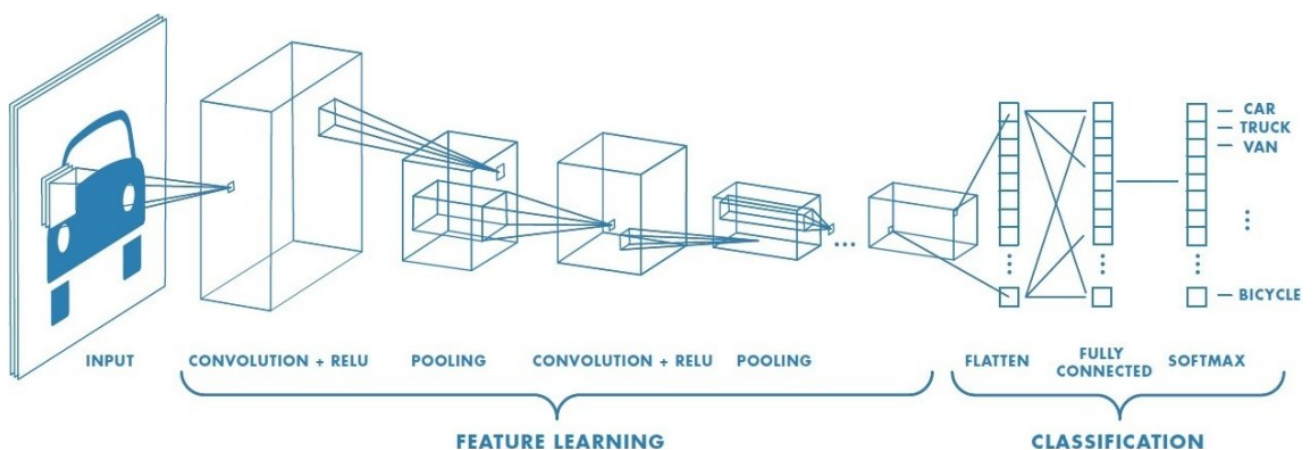


Figura 2.8. Architettura di una rete neurale convoluzionale per la classificazione.

2.5.1 Architettura di una Rete Neurale Convolutionale

L'architettura di una rete neurale convoluzione include diversi blocchi, come, ad esempio, strati convoluzionali, strati di *pooling* e strati completamente connessi (*fully connected*) [32].

L'architettura della rete neurale convoluzione può essere suddivisa in due parti: una parte di *feature detection* (che si occupa quindi dell'estrazione delle *features* attraverso operazioni come la convoluzione, il *pooling* e la *ReLU*), e una parte di classificazione che genera l'output predetto tramite l'utilizzo di *fully-connected layer* e operazioni, ad esempio, di *softmax*.

L'ottimizzazione dei parametri della rete avviene con le stesse modalità descritte in precedenza: il segnale funzione viene propagato in avanti nella rete fino allo strato finale, dove il gradiente della *loss function* viene computato e retro-propagato per consentire l'aggiornamento dei pesi tramite un algoritmo di discesa del gradiente [36].

2.5.1.1 Layer di estrazione delle features

Questi *layer* eseguono operazioni che alterano i dati con l'intento di apprendere le caratteristiche specifiche dei dati. Tre dei *layer* più comuni sono: convoluzione, attivazione o *ReLU* e *pooling*.

2.5.1.1.1 Strati di convoluzione (*Convolutional layers*)

La convoluzione è un particolare tipo di operazione lineare in cui ad un array multidimensionale di numeri, detto tensore di ingresso, viene applicato un filtro, o *kernel*, ossia una maschera di parametri multidimensionale limitata in altezza e larghezza ma che si estende per tutta la profondità del volume di input.

Il nucleo scansiona il tensore di ingresso in altezza e larghezza: in ogni posizione, viene calcolato il prodotto scalare tra il *kernel* e la porzione di input coperta, in modo da ottenere il valore di output nella posizione corrente. Il volume in uscita ottenuto una volta terminato il processo viene detto mappa di attivazione o *feature map* [36].

È utile in questo contesto formalizzare quanto descritto nel caso in cui l'ingresso della rete sia un'immagine.

Nel caso più semplice di immagine a livelli di grigio, descritta dalla funzione $I(x,y)$, verrà utilizzato un *kernel* K bidimensionale per implementare l'operazione di convoluzione [22]:

$$S(x, y) = (K * I)(x, y) = \sum_m \sum_n I(m, n)K(x - m, y - n) \quad (2.38)$$

In realtà, molte librerie per la modellazione delle reti neurali implementano la funzione di cross-correlazione, in cui il filtro non è flippato:

$$S(x, y) = (K * I)(x, y) = \sum_m \sum_n I(x + m, y + n)K(m, n) \quad (2.39)$$

In caso di immagine a colori, codificata dalla combinazione dei tre colori di base RGB (*Red-Green-Blue*) e rappresentata dunque da un tensore tri-dimensionale, anche il filtro sarà tri-dimensionale e lo strato di convoluzione si comporterà allo stesso modo, tenendo in considerazione le tre dimensioni. Infine, la somma elemento per elemento eseguita lungo la profondità del volume ottenuto consentirà di ottenere una mappa di attivazione bi-dimensionale (Figura 2.9 a destra) [37].

La procedura può essere ripetuta utilizzando diversi filtri in modo da ottenere il numero desiderato di mappe di attivazione, ciascuna delle quali rappresenterà diverse caratteristiche del tensore di input estratte dallo specifico filtro.

È ora possibile comprendere maggiormente come la caratteristica chiave dell'operazione di convoluzione risieda nella condivisione dei pesi dei filtri lungo le posizioni del tensore di ingresso: ciò consente non solo di incrementare l'efficienza del modello neurale, riducendo il numero di pesi da apprendere, ma rende il *pattern* di caratteristiche locali estratte dalla maschera invariante alla traslazione [36].

Riprendendo inoltre l'interessante analogia con il sistema visivo, il risultato dell'operazione di convoluzione può essere interpretato come una struttura tridimensionale di neuroni in cui ciascuna unità è connessa a un campo recettivo del livello precedente: lungo la profondità del volume, ciascun neurone coprirà il medesimo campo visivo ma imparerà a rilevare caratteristiche dell'ingresso differenti.

I parametri dei filtri (i pesi e il *bias* caratterizzanti ciascun *kernel*) rappresentano i parametri apprendibili nello strato convoluzionale durante la fase di addestramento della rete.

I seguenti iper-parametri devono essere invece impostati dal programmatore prima della fase di *training*:

- Dimensione dei filtri: altezza (h) e larghezza (w) dei kernel, che definiscono dunque la dimensione del campo ricettivo di ciascun neurone.
- Numero di filtri: corrisponde con la profondità di ciascun *hidden layer*, definendo dunque il numero di *feature maps*.

- *Stride* (passo): controlla la distanza tra due localizzazioni successive del *kernel*, ovvero il passo con cui il filtro muove scansionando l'input in altezza e larghezza. Determina quindi la sovrapposizione dei campi ricettivi.
- *Padding*: mediante l'aggiunta di un bordo al tensore d'ingresso con diverse modalità (*padding* a zero, a 1, con *flip*, ...) permette di regolare altezza e larghezza del volume in uscita dalla convoluzione. Nel caso dello *zero padding*, viene aggiunto un bordo di zeri tale da ottenere in output un volume che abbia dimensioni uguali a quelle del tensore di ingresso (Figura 2.9 a sinistra).

Attraverso gli iper-parametri è possibile definire:

- Numero di parametri apprendibili in uno strato convoluzionale:

$$(h * w * c + 1) * \text{Numero di filtri} \quad (2.40)$$

dove h, w indicano rispettivamente altezza e larghezza del *kernel*, c numero di canali dell'input in ingresso al filtro. Il termine "+1" corrisponde al *bias*.

Questo valore permette di fare una rapida approssimazione della memoria necessaria durante la fase di addestramento, oltre a permettere di indagare a priori la probabilità di *overfitting*.

- Dimensione (altezza e larghezza) dell'uscita del filtro:

$$\frac{\text{Input size} - \text{Filter size} + 2 * \text{Padding}}{\text{Stride}} + 1 \quad (2.41)$$

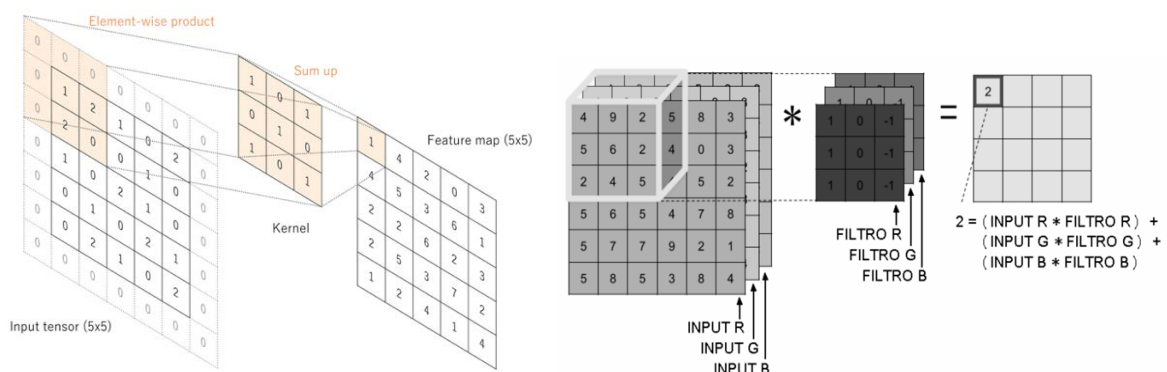


Figura 2.8. **Convolutional Layer**. A sinistra, operazione di convoluzione con zero padding in modo da preservare altezza e larghezza del tensore di ingresso. A destra, operazione di convoluzione per un'immagine a colori.

2.5.1.1.2 Funzione di attivazione – ReLU layer

Lo strato di convoluzione è tipicamente seguito da una funzione di attivazione non lineare: solo le *features* attivate vengono fatte passare al *layer* successivo.

La funzione più utilizzata risulta essere l'unità lineare rettificata (ReLU), la quale esegue un'operazione di soglia su ogni elemento mappando i valori negativi a zero e mantenendo i valori positivi:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.42)$$

È stato dimostrato come l'utilizzo della funzione ReLU consenta un addestramento più rapido ed efficace rispetto alle tradizionali unità come la tangente iperbolica [33].

2.5.1.1.3 Pooling

Lo strato di *pooling* esegue un'operazione di sotto-campionamento non lineare che riduce la dimensione trasversale delle mappe di attivazione, lasciandone inalterata la profondità, con l'obiettivo non solo di semplificare (riducendone i parametri) l'output dello strato convoluzionale precedente, ma anche di introdurre invarianza traslazionale a piccoli spostamenti e distorsioni e, conseguentemente, apportando maggiore robustezza rispetto alla localizzazione delle *features* [36].

Riprendendo l'analogia con la corteccia visiva, l'unità di *pooling* è stata ispirata dal comportamento delle cellule complesse: catturando un campo visivo crescente, esse sono in grado di apprendere gerarchie spaziali di *feature-pattern* risultando meno sensibili a lievi spostamenti nella posizione delle caratteristiche salienti [22].

La forma più popolare dell'operazione di *pooling* è il *max pooling*, che esegue un sotto-campionamento non lineare dividendo l'input in regioni rettangolari e restituendo il valore massimo all'interno di ciascuna finestra (Figura 2.10).

Si tratta di uno strato che non presenta pesi apprendibili ma per il quale deve essere settato un insieme di iper-parametri prima dell'addestramento, quali dimensione del filtro, passo (*stride*) e *padding*. In particolare, se dimensione del filtro r e *stride* sono uguali, l'operazione riduce le dimensioni del tensore di ingresso, per ogni canale, di un fattore r .

La tecnica *max-pooling* conduce a una convergenza più rapida e consente inoltre di migliorare il processo di generalizzazione del modello [38].

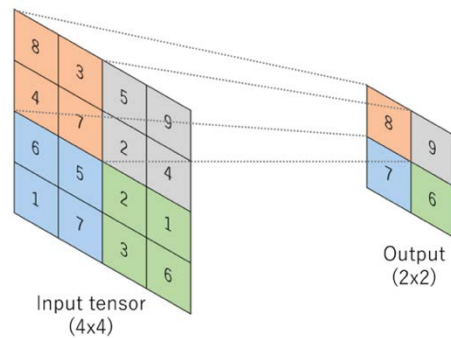


Figura 2.10. L'operazione di *max-pooling* esegue un sotto campionamento non lineare del tensore di ingresso.

2.5.1.2 Layer di classificazione

2.5.1.2.1 Fully connected layer

Le mappe di attivazione in uscita dall'ultimo blocco di estrazione delle *features* sono trasformate in un vettore e connesse ad uno o più strati completamente connessi (*fully connected layers, FC layers*) in cui ogni valore del vettore in input è collegato a tutti i valori dello strato successivo.

L'ultimo strato completamente connesso genererà un vettore di dimensione K pari al numero di classi in cui l'input deve essere classificato.

2.5.1.2.2 SoftMax layer

La funzione di attivazione applicata all'ultimo strato FC differisce generalmente da quelle utilizzate nei livelli precedenti e sarà selezionata sulla base del compito richiesto alla rete.

Come descritto nel paragrafo 2.4.1, nel paradigma di apprendimento multi-classe si utilizza la funzione *softmax*, la quale normalizza i K valori reali ottenuti dall'ultimo layer FC in probabilità di appartenenza alle K classi in esame [36].

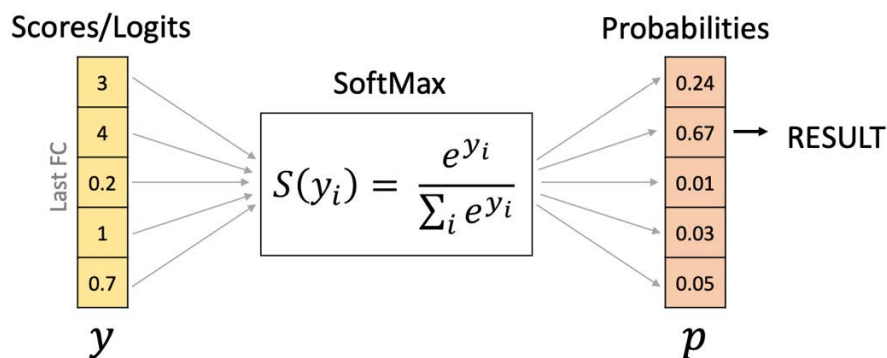


Figura 2.11. La funzione softmax trasforma i valori dello strato FC in probabilità.

2.5.1.2.3 Classification layer

L'ultimo strato della rete neurale convoluzionale assegna la classe di appartenenza. Nell'approccio di classificazione multi-classe, le probabilità calcolate nello strato *softmax* alimentano la funzione di perdita *cross-entropy*, come descritto nel paragrafo 2.4.1 [36].

2.6 Segmentazione semantica

La segmentazione semantica è un algoritmo di *Deep Learning* molto importante nell'analisi delle immagini e nella *computer vision*.

Si tratta di un processo che associa un'etichetta o una categoria a ogni pixel di un'immagine, utile in molteplici applicazioni come, ad esempio, la guida autonoma (per identificare un percorso percorribile dalle auto separando la strada da ostacoli come pedoni, marciapiedi, pali e altre auto), la visione robotica (per identificare e muoversi tra oggetti e terreni) e l'*imaging* medicale (per esempio, per analizzare e rilevare anomalie cancerose nelle cellule).

È un metodo più informativo rispetto alla classificazione di immagini o alla localizzazione di oggetti: queste applicazioni consentono rispettivamente di rilevare la presenza di un oggetto nell'immagine o di indicarne la posizione in un rettangolo di delimitazione. La segmentazione semantica, invece, consente la localizzazione spaziale pixel per pixel di ogni categoria di oggetto presente nella scena [39].

La struttura *SegNet* (Figura 2.12) proposta da *B. et al* rappresenta un approccio comune alla segmentazione semantica [40].

Si tratta di un modello di rete neurale convoluzionale in cui la parte di *features detection* è basata su un'architettura di codifica/decodifica. In tale architettura, ad un percorso di contrazione viene fatto corrispondere un percorso di espansione simmetrica. Come nei casi precedentemente presentati, la parte di estrazione delle *features* è seguita da una parte di classificazione finale, in questo caso, pixel per pixel.

Più in dettaglio:

- *Encoder path*: la parte di codifica è topologicamente identica alla rete neurale convoluzionale descritta nel paragrafo precedente (esclusi i *layers* di classificazione). Il sottocampionamento in successione dell'immagine originale porta ad una perdita di risoluzione spaziale: il percorso di contrazione è utilizzato dunque per acquisire un contesto nell'immagine per ogni pixel della *feature map*.

- *Decoder path*: nel percorso di espansione simmetrica, ogni blocco di decodifica utilizza gli indici di *max pooling* ricevuti dal corrispondente blocco di contrazione (vengono cioè memorizzate le posizioni dei valori massimi in ogni finestra di raggruppamento per ogni *feature map*) per eseguire un sovra-campionamento non lineare delle mappe di attivazione ricevute in ingresso, al fine dunque di recuperarne la risoluzione persa in fase di sotto-campionamento.

Poiché l'operazione di *up-sampling* porta ad avere *feature maps* rare, queste vengono convolute con filtri di decodifica addestrabili in modo da ottenere mappe di attivazione dense.

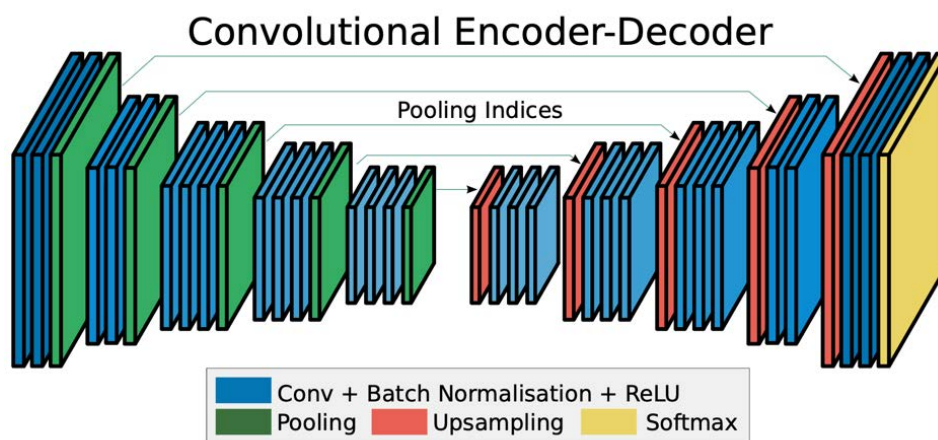


Figura 2.12. Architettura di rete neurale convoluzione SegNet per la segmentazione semantica.

Una struttura simile al modello *SegNet* è rappresentata dalla cosiddetta *U-Net* proposta da *R. et al.* (Figura 2.13) [41].

L'architettura consiste ancora di un percorso di contrazione, utile per catturare il contesto nell'immagine e caratterizzato dalla tipica struttura di una CNN, seguito da un percorso di espansione simmetrico per abilitare una precisa localizzazione.

A differenza però della *SegNet*, la *U-Net* non riutilizza gli indici di raggruppamento, ma memorizza le mappe di attivazione ottenute nei blocchi di codifica prima del sotto-campionamento, le trasferisce ai corrispondenti blocchi di decodifica e le concatena con le rispettive mappe sovra-campionate attraverso un'operazione di up-convoluzione (ottenuta, in questo caso, tramite una convoluzione trasposta).

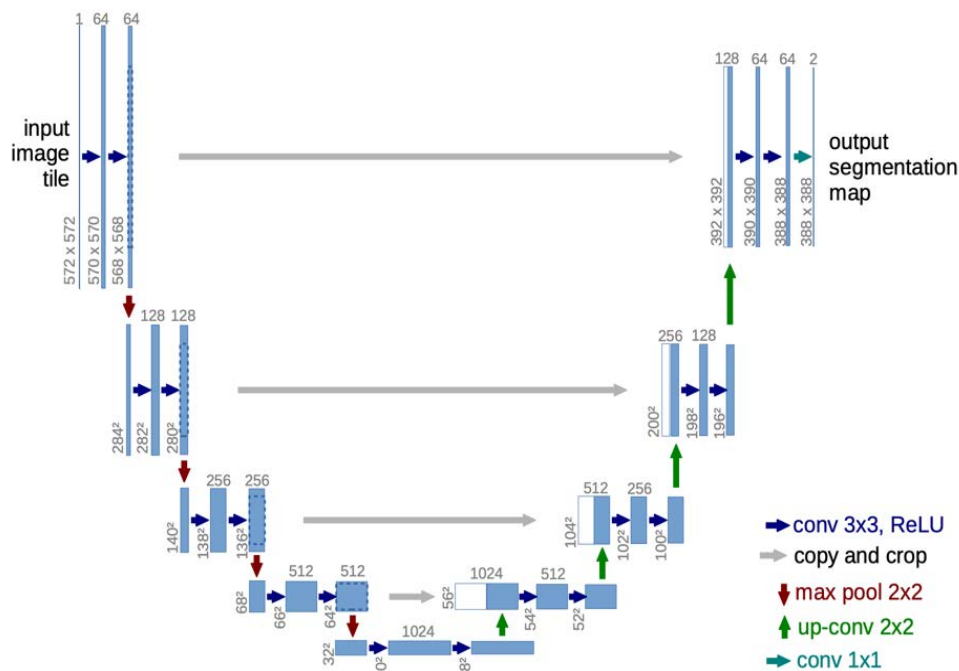


Figura 2.13. *U-Net*. Architettura di rete neurale convoluzionale *U-Net* per la segmentazione semantica.

2.7 Applicazioni cliniche del Deep Learning

Gli algoritmi di *Deep Learning*, e in modo particolare le reti neurali convoluzionali, sono diventati rapidamente una metodologia molto importante nella pratica clinica, soprattutto nel campo dell'analisi delle bioimmagini.

Uno studio pubblicato nel 2017 ha indicato più di 300 contributi del DL nel campo dell'*imaging* biomedicale (Figura 2.12), nel quale è possibile rilevare le seguenti applicazioni [26]:

- Classificazione, in cui si fa riferimento non solo al determinare la presenza o meno di una malattia nell'ambito di un esame diagnostico (si parla, per esempio, della classificazione dei pazienti affetti da *Alzheimer* sulla base di immagini di risonanza magnetica del cervello), ma anche alla classificazione di oggetti o lesioni nell'immagine, come noduli nella tomografia computerizzata del torace o masse nella mammografia.
- *Detection*, in cui si fa riferimento alla localizzazione di organi, regioni, oggetti o lesioni anatomiche. Un esempio è dato dal riconoscimento di noduli nei polmoni o di metastasi cancerose nei linfonodi del seno.
- Segmentazione di organi, sottostrutture anatomiche o lesioni, un'applicazione molto importante in quanto consente l'analisi quantitativa di parametri clinici come volume o forma degli oggetti anatomici.
- Registrazione, in cui gli algoritmi di DL possono consentire, ed esempio, di predire direttamente i parametri di trasformazione geometrica da applicare a due immagini in modo da allinearle nello spazio.

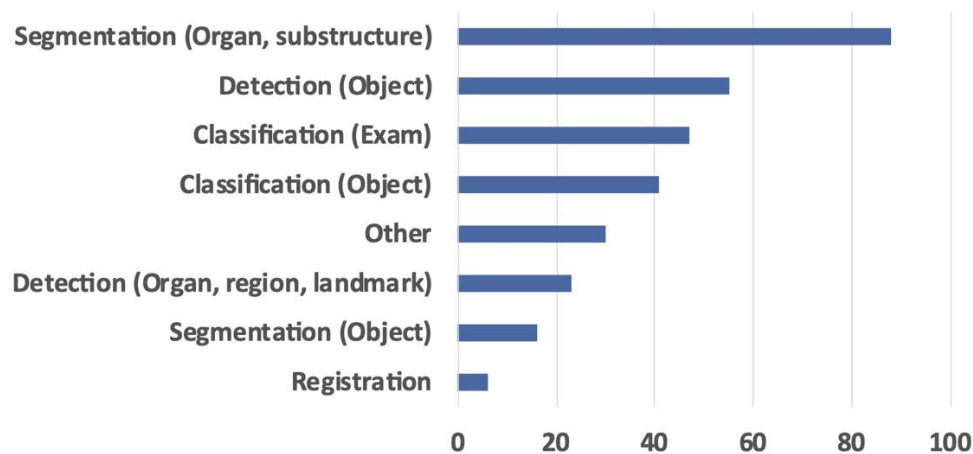


Figura 2.12. Numero di pubblicazioni, riportato nel 2017, per ogni applicazione del *Deep Learning* nell'analisi di immagini biomediche.

2.7.1 Applicazioni in oftalmologia

Gli oftalmologi si interfacciano al problema del *pattern recognition* attraverso la visualizzazione diretta o indiretta dell'occhio e delle sue strutture limitrofe per la diagnosi di determinate patologie; si tratta dunque di un campo che presenta tutte le potenzialità per una vincente incorporazione degli algoritmi di *Deep Learning* [42,43].

Nella pratica oftalmologica metodi di apprendimento automatico sono stati applicati a diverse modalità diagnostiche, come:

- Fotografia del *fundus*, per lo studio della retina.
- Tomografia a coerenza ottica, per lo studio di retina e cornea.
- Microscopia confocale, per lo studio della cornea.

Le applicazioni in questo ambito includono:

- Diagnosi di patologie oculari, come cataratta, glaucoma, disordini retinici (retinopatia diabetica, degenerazione maculare correlata all'età).
- Segmentazione di strutture anatomiche.
- Segmentazione e riconoscimento di anomalie retiniche, come segmentazione di cisti intra-retiniche o di edema maculare diabetico.

CAPITOLO 3.

Segmentazione automatica delle cellule dendritiche corneali

Lo scopo di questa tesi consiste nello sviluppo di un algoritmo, basato su tecniche di *Deep Learning*, per la segmentazione automatica delle cellule dendritiche corneali. Come già anticipato nel capitolo 1.5, la presenza di cellule dendritiche e il loro stato di maturazione sono correlate alla presenza e/o al progredire di alcune patologie, sia oculari sia sistemiche. Per studiare la correlazione tra questi fattori, dunque, si rende necessario identificare e segmentare le cellule presenti nella cornea. Questo processo, sino ad ora, viene effettuato manualmente dai clinici: ciò comporta che l'analisi sia *time-consuming* e influenzata dalla soggettività. Questa tesi, quindi, si prefigge lo scopo di fornire descrittori della cornea di forma automatica e che siano robusti, con il fine ultimo di identificare e segmentare le cellule dendritiche.

Per raggiungere tale obiettivo, è stata sviluppata una rete neurale convoluzionale (CNN), con un'architettura basata sul tipo *U-Net*, per la segmentazione semantica di immagini del plesso nervoso sub-basale della cornea, ottenute mediante microscopia confocale in vivo.

Ad ogni pixel di ciascuna immagine è stato assegnato un *label* attraverso la definizione, inizialmente, di sole due categorie (cellula dendritica e "altro"); successivamente, si è ritenuto opportuno aggiungere una terza etichetta (nervo). L'aggiunta di un'ulteriore categoria è stata necessaria per poter addestrare al meglio il modello: maggiori infatti sono le informazioni che vengono passate in ingresso alla rete neurale, migliori sono i risultati.

L'algoritmo è stato interamente sviluppato in ambiente Matlab®.

Il flusso di lavoro ha previsto fondamentalmente 4 passaggi:

1. Creazione delle immagini *pixel-label* per l'addestramento della rete, in cui ogni pixel è stato etichettato manualmente con la categoria di appartenenza.
2. Creazione dell'architettura di rete neurale *U-net*, in cui è stata definita la struttura di codifica/decodifica necessaria per la classificazione a livello di pixel.
3. Impostazione degli iper-parametri, addestramento e validazione della rete mediante *cross-validation*.
4. Fase di *post-processing*.

Nei prossimi paragrafi verranno esposti con maggior dettaglio "materiali e metodi" utilizzati per il progetto. Anzitutto saranno descritti il dataset utilizzato e il lavoro di *pre-processing* svolto sulle immagini. Saranno poi definiti i diversi strati implementati nell'architettura di rete

neurale convoluzionale; verranno quindi riportati gli iper-parametri impostati e altri aspetti riguardanti l'addestramento della rete. Infine, saranno presentate le operazioni di *post processing* utilizzate per ottenere una migliore corrispondenza *pixel-a-pixel*.

3.1 Dataset

In questo studio, è stato utilizzato un dataset di 100 immagini confocali del plesso nervoso sub-basale corneale di 100 soggetti. Le immagini sono state acquisite con lo strumento *Heidelberg Retina Tomograph (HRT-II)* con il modulo *Rostock Cornea Module (Heidelberg Engineering GmbH, Heidelberg, Germania)*. Ciascuna immagine copre un campo di $400 \times 400 \mu\text{m}^2$ (384×384 pixel). L'acquisizione è stata eseguita presso la *School of Optometry and Vision Science* della *University of New South Wales (UNSW, Sydney)*. Ogni immagine è stata anonimizzata eliminando ogni informazione sui pazienti.

Ciascuna immagine è stata analizzata da clinici esperti, i quali hanno indicato la localizzazione delle cellule dendritiche, racchiuse all'interno di un cerchio rosso, e ne hanno delineato la morfologia attraverso un contorno rosso (Figura 3.1).

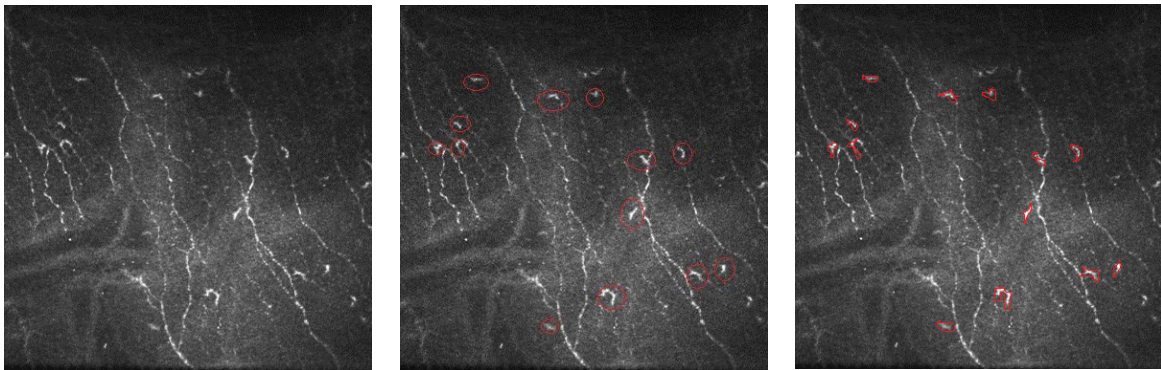


Figura 3.1. Immagine IVCM del plesso nervoso sub-basale originale (a sinistra), con localizzazione delle cellule dendritiche (in centro) e con delineamento della morfologia delle DCs (a destra).

3.2 Pre-processing

3.2.1 Creazione immagini *pixel-label*

Come sottolineato nel capitolo precedente, il paradigma di addestramento considerato in questo lavoro è l'apprendimento supervisionato. È stato quindi necessario creare un *training set* in cui

ogni immagine originale fosse etichettata, a livello di pixel, con la categoria di appartenenza. In particolare, come anticipato, sono stati considerati tre *label*: cellula dendritica, nervo e altro (dove con “altro” si intende tutto ciò che non è né cellula dendritica né fibra nervosa).

Per la creazione del *ground truth*, da utilizzare in fase di *training*, sono state utilizzate le immagini in cui le cellule dendritiche sono state delineate dai clinici.

All'interno di ciascuna ROI, definita dal bordo rosso tracciato manualmente, i pixel appartenenti a una DC sono stati etichettati mediante operazioni di sogliatura e operazioni morfologiche. Il risultato di tali operazioni, infine, è stato sottoposto ad un controllo manuale, per assicurare che il numero di cellule segmentate fosse uguale a quello di partenza.

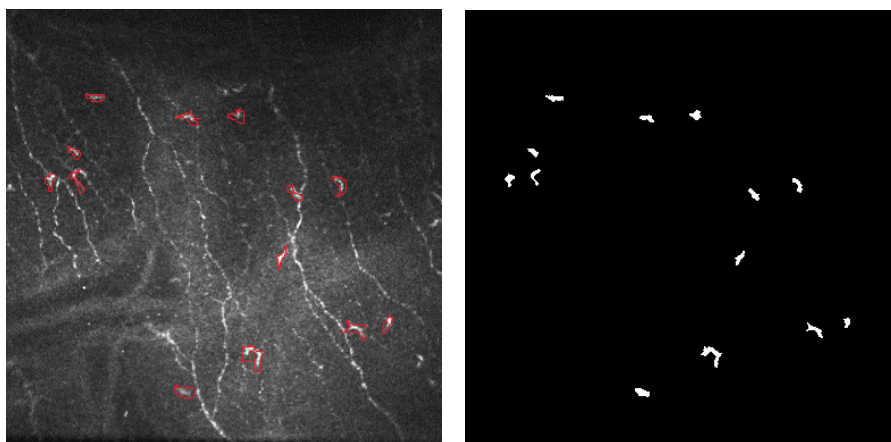


Figura 3.2. Immagini IVCM del plesso nervoso sub-basale con delineata la morfologia delle cellule dendritiche (a sinistra) e segmentazione manuale delle DCs (a destra, in bianco le cellule, in nero lo sfondo).

Come anticipato precedentemente, per migliorare le *performance* della rete è stato ritenuto necessario introdurre anche la classe “nervi” nella definizione del *ground truth*. I nervi, infatti, si presentano come un *pattern* importante nelle immagini che si vogliono analizzare, e tale informazione è sicuramente d’aiuto per discriminare ciò che si presenta come “oggetto luminoso” nelle immagini di IVCM.

Poiché anche nel caso dell’analisi dei nervi, il tracciamento manuale risulta dispendioso a livello di tempo e influenzato dalla soggettività, si è deciso di utilizzare il risultato di un lavoro precedente. In *Guimarães et al.*, è stato proposto un metodo basato su algoritmi di *support vector machine* (SVM) capace di tracciare in maniera automatica i nervi presenti nelle immagini confocali, presentando una sensibilità attorno al 90% [1]. Questo tracciamento è già stato dimostrato essere valido per il *training* di una CNN per il *tracing* dei nervi. In *Colonna et al.*, infatti, è stato utilizzato come *ground truth* per l’addestramento di una rete neurale

convoluzionale, con architettura *U-Net*. Il modello ha permesso di ottenere ottimi risultati, con una sensibilità superiore al 95% [2]. In entrambi gli algoritmi, però, si presentava un valore di *false discovery rate* (FDR) abbastanza elevato: tali valori erano dovuti, in parte, ad una inaccurata analisi manuale e, in parte, alla classificazione errata. In questo secondo caso, parte delle classificazioni errate erano dovute alla presenza di cellule dendritiche erroneamente etichettate come nervo.

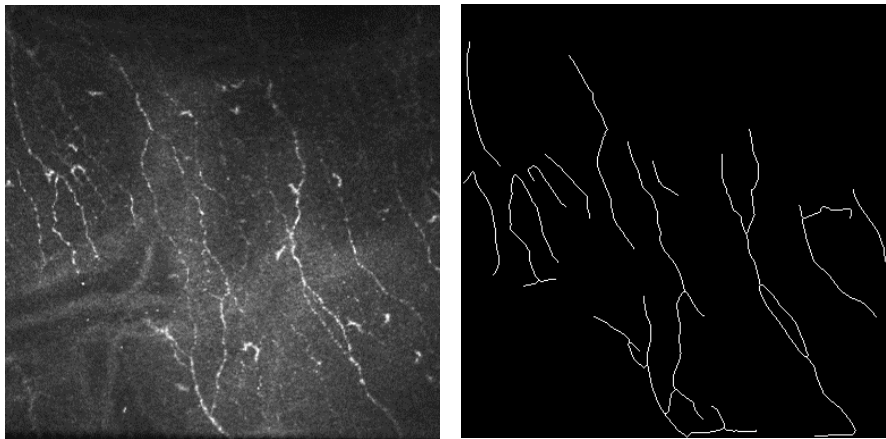


Figura 3.3. Immagine IVCM del plesso nervoso sub-basale originale (a sinistra) e segmentazione manuale delle fibre nervose (a destra, in bianco i nervi, in nero lo sfondo).

Per ottenere le immagini *pixel-label* finali è stato necessario unire le maschere binarie ottenute dall'analisi manuale delle cellule dendritiche e quelle ottenute per i nervi dal precedente algoritmo automatico. In particolare, poiché l'algoritmo per il tracciamento dei nervi può presentare alcuni errori (come spiegato precedentemente, si ha un FDR elevato), i pixel associati alla categoria "nervo" e contemporaneamente indicati dai clinici come DCs, sono stati etichettati con la classe "cellula dendritica".

Inoltre, per discriminare ulteriormente i nervi dai dendriti, sono state attuate alcune operazioni morfologiche: i tracciati dei nervi segmentati sono stati dilatati e filtrati in base all'area di ciascuna regione connessa (eliminando i segmenti minori formati, ad esempio, dalle precedenti operazioni di assegnazione dei *label*).

Per una corretta visualizzazione delle immagini usate come *ground truth*, ad ogni pixel è stato assegnato un livello di grigio sulla base della categoria di appartenenza.

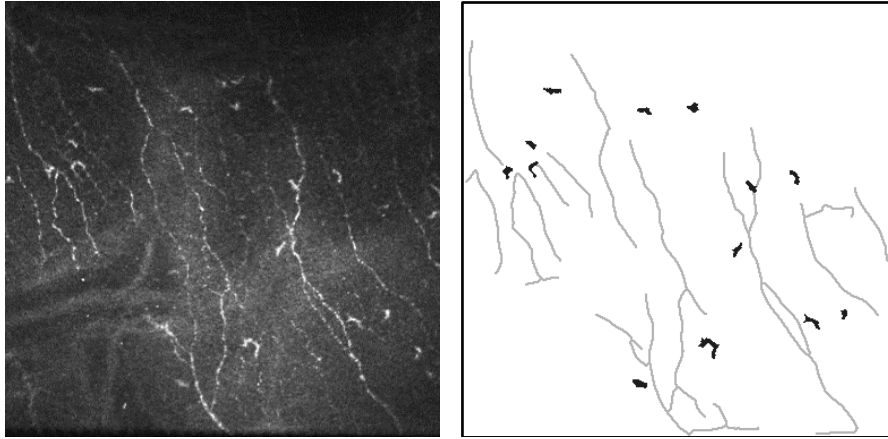


Figura 3.4. Immagine IVCM del plesso nervoso sub-basale originale (a sinistra), e immagine *pixel-label* corrispondente (a destra), in bianco lo sfondo, in nero le cellule dendritiche, in grigio i nervi.

3.2.2 Data augmentation

Al fine di incrementare la capacità di generalizzazione di un modello di *Deep Learning*, è importante collezionare più dati di addestramento possibile.

In particolare, le reti neurali convoluzionali hanno, tipicamente, un numero di parametri apprendibili dell'ordine del milione, data la complessità dei *task* che viene loro richiesto di eseguire. Per ottenere buone performance, sarebbe dunque necessario avere un numero di esempi proporzionale.

Poiché nel progetto in esame il *training set* aveva cardinalità molto limitata, per incrementarne la numerosità è stata utilizzata la tecnica di *data augmentation*.

Per ciascuna immagine, quindi, ne sono state ottenute altre due, mediante un'operazione di riflessione (*flip*) in direzione verticale e orizzontale.

Nella segmentazione semantica, i dati di addestramento sono costituiti dalle coppie “immagine originale – immagine *pixel-label*”: l'elaborazione ha riguardato quindi sia dati originali sia dati etichettati (Figura 3.5).

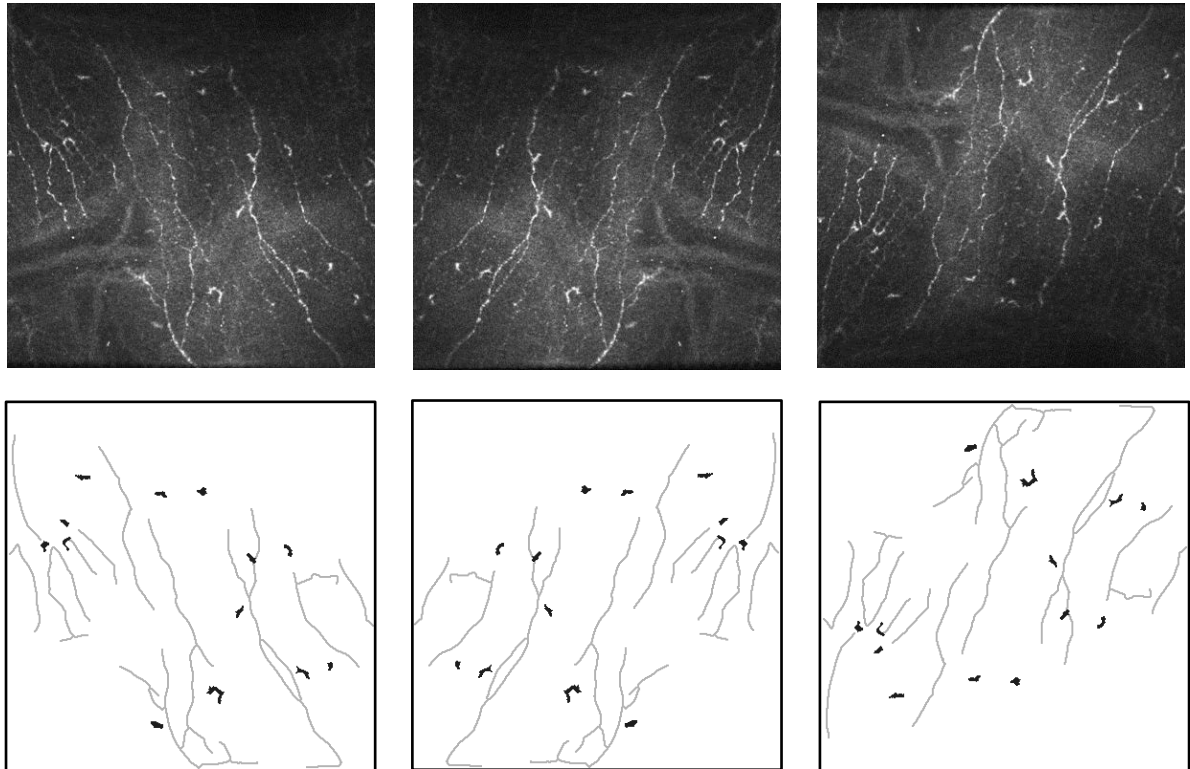


Figura 3.5. **Data augmentation.** In alto, immagine originale (a sinistra), immagine riflessa in direzione orizzontale (in centro) e immagine riflessa in direzione verticale (a destra). In basso, immagine *pixel-label* originale (a sinistra), immagine *pixel-label* riflessa in direzione orizzontale (in centro) e immagine *pixel-label* riflessa in direzione verticale (a destra).

3.3 Rete Neurale Convolutionale: architettura *U-Net*

In figura 3.6 è riportata la rete neurale convoluzionale progettata, basata sull'architettura *U-Net*.

Consiste dunque di un percorso di codifica/contrazione, per catturare il contesto nell'immagine, seguito da un percorso di decodifica/espansione simmetrica per ottenere la classificazione di ogni singolo pixel.

La parte contraente riceve in ingresso un'immagine 384x384 a livelli di grigio, ed è caratterizzata da 4 blocchi. Ogni blocco consiste di 2 unità convoluzionali (strato convoluzionale, strato di *batch normalization*, funzione di attivazione *ReLU*) seguite da uno strato di *max pooling*.

In particolare, ogni strato convoluzionale applica n filtri $3 \times 3 \times c$, scansionando l'input con passo (*stride*) unitario e *zero padding*, in modo tale che la dimensione trasversale (altezza e larghezza) del tensore di uscita sia uguale alla dimensione trasversale del tensore di ingresso. Per ogni *kernel*, il numero di canali c è sempre uguale al numero di canali dell'input. Inoltre, per ogni

blocco, il numero di filtri n applicati dagli strati convoluzionali sarà il doppio di quello precedente: si parte dunque da 16 *feature maps* prodotte in uscita dal primo blocco, per arrivare alle 128 mappe di attivazione del quarto.

Gli iperparametri degli strati di *max pooling* (dimensione dei filtri 2x2, passo 2) sono impostati in modo tale da sotto-campionare l'ingresso di un fattore 2.

Nel quarto blocco, prima dell'operazione di sotto-campionamento, è inserito uno strato di *dropout*: in fase di addestramento della rete, ad ogni iterazione ciascun neurone verrà disattivato con probabilità 0.5. Ciò incrementa le capacità di generalizzazione del modello neurale.

Un ulteriore livello di due unità convoluzionali, seguite da uno strato di *dropout*, funge da "ponte" tra il percorso di codifica e il percorso di decodifica.

La parte di espansione simmetrica consiste anch'essa di 4 blocchi, ciascuno dei quali concatena l'immagine sovra-campionata con quella corrispondente nel percorso di contrazione, per poi applicare due unità convoluzionali (strato convoluzionale, strato di *batch normalization*, funzione di attivazione *ReLU*) impostate come descritto precedentemente.

L'*up-sampling* dell'immagine è eseguito utilizzando uno strato di convoluzione trasposta seguito da uno strato *ReLU*. In particolare, l'operazione di convoluzione trasposta esegue sia un sovra-campionamento dell'input, raddoppiandone la dimensione trasversale (altezza e larghezza), sia un filtraggio, utilizzando ogni volta un numero di *kernels* dimezzato rispetto al blocco precedente.

L'uscita dell'ultimo livello di espansione avrà dunque le stesse dimensioni trasversali (altezza e larghezza, 384x384) dell'immagine in ingresso alla rete. Tuttavia, il numero di canali (terza dimensione) sarà maggiore, pari al numero di filtri dell'ultima convoluzione (16).

Questa terza dimensione deve quindi essere "ristretta" al numero di categorie in esame per la segmentazione (ovvero 3). Ciò viene fatto utilizzando uno strato convoluzionale 1x1 il cui numero di filtri è pari al numero di classi.

Infine, lo strato *softmax* e lo strato di classificazione assegnano l'etichetta categorica ad ogni pixel dell'immagine. In particolare, la funzione *softmax* produce 384x384x3 attivazioni, che rappresentano, per ognuna delle tre categorie, la probabilità di ciascun pixel di appartenere a quella specifica classe.

La funzione costo *cross-entropy* assegna infine ogni pixel alla categoria di appartenenza.

Poiché le classi sono sbilanciate (i pixel di nervi e cellule sono sottorappresentati rispetto ai pixel etichettati come "altro"), è stata utilizzata una versione pesata della *cross-entropy*: nella computazione della funzione costo, ogni categoria è stata ponderata secondo l'inverso della frequenza della classe.

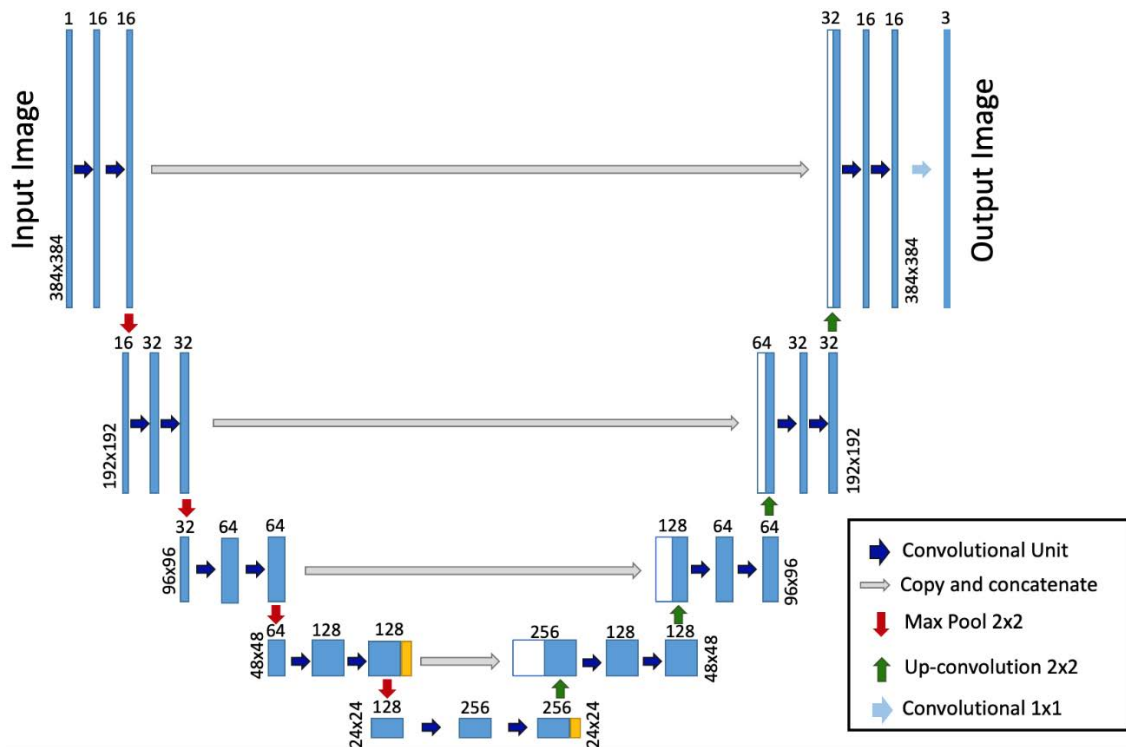


Figura 3.6. **Architettura U-Net della rete neurale convoluzionale.** Ogni rettangolo blu corrisponde a un tensore 3D. Il numero riportato sopra ogni rettangolo denota il numero di canali (numero di filtri). Il numero nell'angolo in basso a sinistra del rettangolo fornisce la dimensione trasversale (altezza e larghezza). I rettangoli bianchi nel percorso di decodifica rappresentano i tensori concatenati corrispondenti nel percorso di codifica. I rettangoli arancioni corrispondono agli strati di *dropout*. Le frecce rappresentano le diverse operazioni, come riportato nella legenda.

3.4 Addestramento della rete

3.4.1 Iper-parametri

Nel processo di addestramento di una rete neurale, l'impostazione degli iper-parametri rappresenta una fase molto delicata, che andrà a influenzare tempi computazionali richiesti e *performance* del modello.

Per selezionare il set migliore di iper-parametri, è necessario testare diversi valori e combinazioni, valutando di volta in volta l'accuratezza raggiunta dall'algoritmo con quello specifico insieme.

Dopo diverse sperimentazioni, sono stati impostati i seguenti iper-parametri, la cui validità è stata poi valutata mediante *cross-validation*, come descritto nel paragrafo successivo.

1. **Metodo di discesa del gradiente:** per l'addestramento dei parametri della rete, è stato utilizzato il metodo di discesa stocastica del gradiente *Adam* (si è infatti osservato un

miglioramento rispetto al metodo più usato: *stochastic gradient descent with momentum*). Gli iper-parametri dell'algoritmo di ottimizzazione sono stati impostati usando i valori di default:

- $\beta_1 = 0.9$ (*Gradient decay factor*, velocità di decadimento esponenziale del gradiente).
 - $\beta_2 = 0.999$ (*Squared gradient decay factor*, velocità di decadimento esponenziale del quadrato del gradiente).
 - $\varepsilon = 10^{-8}$ (*offset*).
2. **Mini-batch size:** la dimensione del *mini-batch*, ovvero del sottoinsieme del *training set* utilizzato per valutare il gradiente della *loss*, è stata impostata a 33 (scelta in parte legata alle risorse limitate di memoria a disposizione). Di conseguenza, ad ogni iterazione, l'algoritmo di discesa del gradiente utilizza 33 immagini per aggiornare i parametri della rete. È stata inoltre utilizzata l'opzione di rimescolamento (*shuffle*) del *training set* prima di ogni epoca: in questo modo, ad ogni iterazione verrà presentato alla rete un sottoinsieme di immagini diverso rispetto alla corrispondente iterazione dell'epoca precedente.
 3. **Max epoch:** il numero massimo di epoche è stato settato a 66. Si tratta di un iper-parametro molto importante: da un lato, un valore di epoche troppo ridotto potrebbe compromettere le prestazioni del modello sui dati di addestramento (*underfitting*), dall'altro, un numero troppo elevato potrebbe condurre l'algoritmo a memorizzare i dati di esempio, perdendo la capacità di generalizzazione (*overfitting*). In questo lavoro, il numero di epoche è stato impostato tramite ispezione visiva del *plot* dell'accuratezza e della *loss*: a 66 epoche, il raggiungimento di una sorta di *plateau* indicava che la rete non era più in grado di acquisire informazioni utili al miglioramento delle *performance*.
 4. **Learning rate:** il *learning rate* non è stato impostato a un valore costante, ma è stato ridotto di un fattore 0.1 ogni 22 epoche, partendo da un valore iniziale pari a 0.01. In questo modo è possibile ridurre i tempi di addestramento, utilizzando inizialmente passi più ampi nella discesa del gradiente, quando la soluzione è lontana dal minimo della *loss*, per poi ridurli progressivamente durante il processo di ottimizzazione.
 5. **Regolarizzazione L2:** per incrementare la capacità di generalizzazione del modello, è stata utilizzata la tecnica di regolarizzazione L2 (*weight decay*). Per il coefficiente di regolarizzazione λ è stato utilizzato il valore di default 0.0001.

3.4.2 100-fold cross-validation

Per stabilire la capacità di generalizzazione di un modello è necessario valutarne la *performance* su dati non inclusi nel set di addestramento, e dunque mai visionati dalla rete.

Data la cardinalità molto ridotta del *training set*, nonostante la *data augmentation*, non è stato possibile suddividere i dati di addestramento in un *test set* e in un *validation set*.

Di conseguenza, per validare la scelta degli iper-parametri, è stata utilizzata la tecnica di cross-validazione *100-fold*: le 300 immagini del *training set* sono state suddivise in 100 sottoinsiemi di tre elementi, ciascuno costituito da un'immagine originale e dalle due corrispondenti ottenute con la *data augmentation*. Sono stati quindi eseguiti 100 diversi cicli di addestramento, utilizzando per ognuno 297 immagini di allenamento ed escludendo un diverso sottoinsieme tra i 100, utilizzato invece per la validazione. In particolare, le prestazioni del modello sono state valutate di volta in volta mediante ispezione visiva di una sola delle tre immagini del *validation set*, ossia quella originale.

3.5 Post-processing

Nel percorso di decodifica, la *U-Net* cerca di recuperare la risoluzione dell'immagine originale persa nella parte di contrazione, attraverso strati di *up-sampling*. Tuttavia, è molto probabile che il processo di recupero delle informazioni spaziali non sia perfettamente preciso. Di conseguenza, la segmentazione *pixel a pixel* porta ad avere molti falsi positivi, ovvero pixel classificati dalla rete come "cellula dendritica" ma non etichettati come tali nel manuale. Ad ogni modo, i falsi positivi non sono distribuiti su tutta l'immagine, ma sono tipicamente localizzati nelle zone circostanti le DCs.

Un'adeguata fase di *post-processing* consente dunque di migliorare i risultati.

La figura 3.7 consente di visualizzare quanto esposto.

L'immagine in basso a sinistra costituisce una rappresentazione del risultato prodotto dalla rete neurale. Definire un'operazione di sogliatura all'interno di ciascuna ROI, definita dalla DC ricostruita dalla *U-Net* (in azzurro), ha permesso di ottenere una ricostruzione spaziale più fedele alla segmentazione manuale (raffigurata nell'immagine in alto a destra).

Successivamente, ciascuna ROI è stata filtrata in base all'area. Questa operazione è stata necessaria per scartare molti falsi positivi, intesi in questo caso a livello di regione connessa, ovvero come cellule dendritiche riconosciute dall'algoritmo ma non etichettate come tali nel

manuale. Come verrà maggiormente descritto nel prossimo capitolo, questa procedura ha permesso di ridurre il *False Discovery Rate*, a scapito però della sensibilità.

L'immagine in basso a destra rappresenta il risultato finale della fase di *post-processing*.

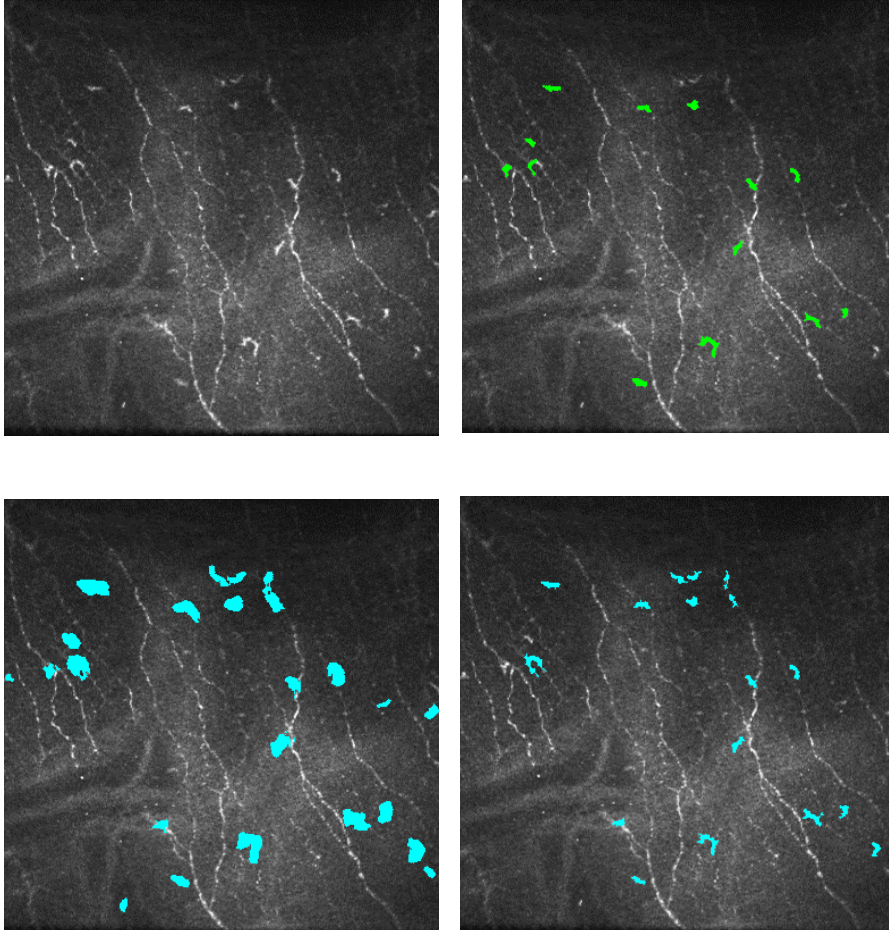


Figura 3.7. **Post-processing.** In alto a sinistra, immagine originale. In alto a destra, segmentazione manuale delle cellule dendritiche. In basso a sinistra, segmentazione semantica delle cellule dendritiche in uscita dalla *U-Net*. In basso a destra, risultato finale *post-processing*.

Infine, tre immagini sono state sottoposte ad un ulteriore step di elaborazione manuale. Queste presentavano, oltre a cellule dendritiche e nervi, strutture epiteliali o neuromi, ovvero proliferazioni iperplastiche di cellule e fibre nervose, che venivano riconosciute dall'algoritmo come aggregati di DCs. Così come è stato fatto per i nervi, anche in questo caso sarebbe stato giusto etichettare i *pattern* rappresentati l'epitelio o i neuromi. In questo caso, però, non si avevano a disposizione tracciamenti manuali o algoritmi in grado di riconoscere tali strutture. Si è deciso, dunque, di creare manualmente delle maschere binarie con cui eliminare le regioni presentanti tali *pattern* (Figura 3.8).

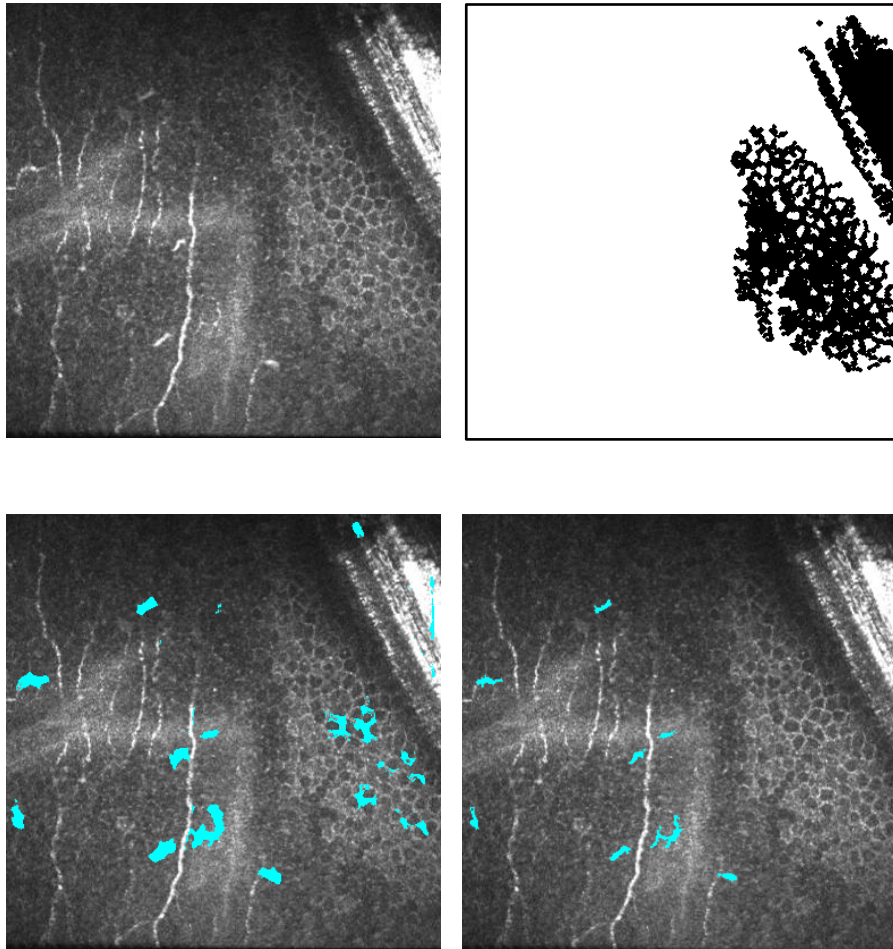


Figura 3.8. **Segmentazione manuale dell'epitelio.** In alto a sinistra, immagine originale in cui è visibile l'epitelio. In alto a destra, maschera binaria dell'epitelio. In basso a sinistra, segmentazione semantica della *U-Net*, in cui sono stati erroneamente localizzati aggregati di DCs nell'epitelio. In basso a destra, risultato *post-processing*.

Capitolo 4.

Risultati

In questo capitolo vengono presentati i risultati ottenuti dall'algoritmo proposto per quanto riguarda la segmentazione automatica delle cellule dendritiche corneali.

Le prestazioni della rete neurale convoluzionale sono state valutate sui *leave-out sets* creati in fase di cross-validazione del modello. In particolare, tra le tre immagini escluse di volta in volta dall'addestramento, è stata considerata solo quella originale (ovvero, l'unica non ottenuta con tecniche di *data augmentation*).

Come verrà esposto nel paragrafo seguente, i risultati sono stati espressi in termini di indici statistici, utilizzando la segmentazione manuale come *ground-truth* di riferimento.

Infine, per ottenere un confronto con un'altra tecnica, le stesse immagini sono state segmentate con un algoritmo automatico sviluppato in un lavoro precedente, il cui obiettivo era il medesimo: la segmentazione delle cellule dendritiche presenti nello strato sub-basale della cornea.

4.1 Analisi statistica

Le immagini segmentate, ottenute come risultato dell'algoritmo proposto, sono state dapprima sottoposte ad ispezione visiva (analisi qualitativa) e, successivamente, sono state analizzate in maniera automatica, estraendone alcuni parametri statistici quantitativi (analisi quantitativa).

Poiché, come ripetuto più volte, l'analisi manuale può essere soggetta ad errori, specialmente se si tratta di valutare la segmentazione pixel-a-pixel, e poiché il primo step nello studio della correlazione DCs-patologie è legato alla sola individuazione delle cellule dendritiche, le prestazioni del modello sono state valutate a livello di cellule (regioni di pixel connessi).

Per l'analisi quantitativa, sono stati considerati i seguenti indici statistici, ottenuti dal confronto con il tracciamento manuale, utilizzato come *ground-truth* durante la fase di *training*:

- TP (True Positive): oggetti correttamente classificati dalla rete come cellule dendritiche, e dunque etichettati come tali anche nel manuale.
- FP (False Positive): oggetti etichettati dalla rete come cellule dendritiche, ma classificati come “nervi” o “altro” nella segmentazione manuale.

- FN (False Negative): oggetti etichettati dalla rete come “nervi” o “altro”, ma classificati come cellule dendritiche nella segmentazione manuale.

La Figura 4.1 riporta graficamente gli indici statistici sopracitati. In particolare, in verde sono rappresentate le regioni corrispondenti con i TP, in rosso quelle corrispondenti con i FP e in blu i FN.

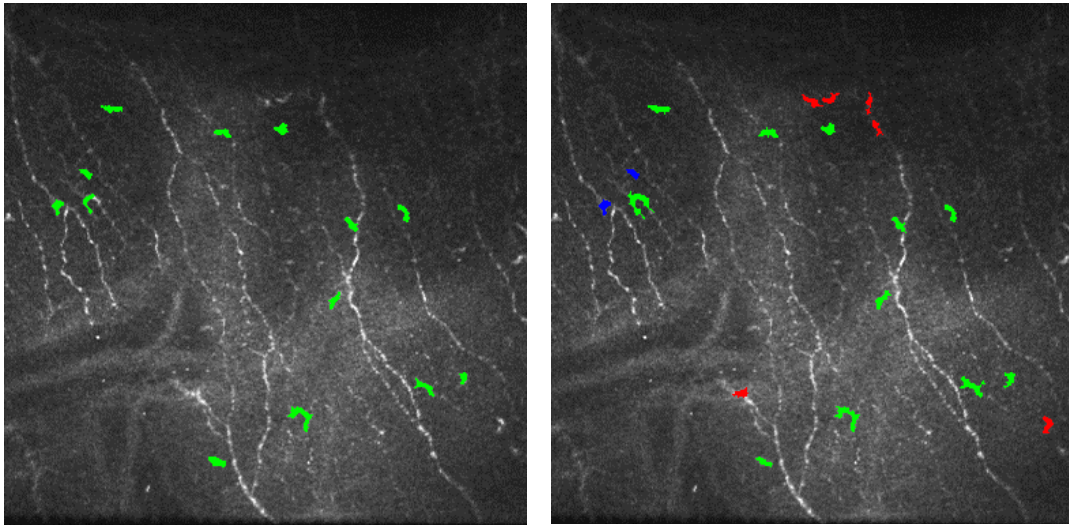


Figura 4.1. **Ispezione visiva dei risultati.** A sinistra, segmentazione manuale delle cellule dendritiche (in verde). A destra, segmentazione automatica delle DCs, in cui è possibile visualizzare: TP (in verde), FP (in rosso), e FN (in blu).

A causa di alcune operazioni di sogliatura applicate durante la fase di *post-processing*, è possibile che alcune regioni, appartenenti ad una stessa cellula dendritica, vengano suddivise in due o più componenti (*splitting region*). L’analisi statistica condotta tiene conto di questa divisione: due regioni ottenute dall’analisi automatica e corrispondenti ad un’unica regione nella segmentazione manuale vengono considerate come un unico TP.

In Figura 4.2 viene riportato un esempio della situazione di *splitting* appena presentata.

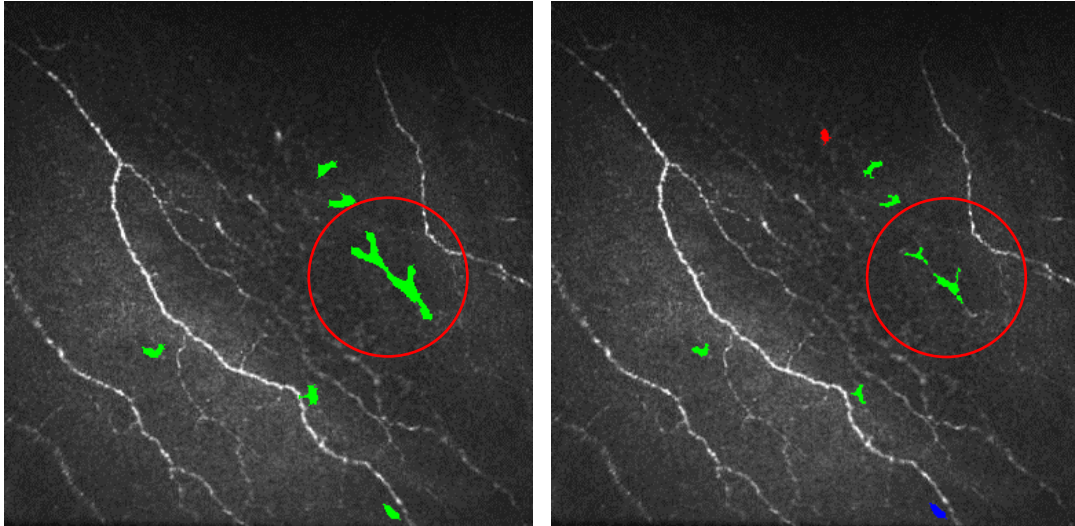


Figura 4.2. *Splitting di una cellula dendritica (cerchio rosso)*. A sinistra, segmentazione manuale delle cellule dendritiche. A destra, risultato del *post-processing*.

A partire dagli indici statistici appena presentati, sono stati calcolati due ulteriori indici:

- La sensibilità, o TPR (*True Positive Rate*): la percentuale di oggetti correttamente classificati come cellule dendritiche dall' algoritmo proposto.

Viene calcolato utilizzando TP e FN:

$$TPR = \frac{TP}{TP + FN}$$

- Il FDR (*False Discovery Rate*): la percentuale di oggetti erroneamente classificati come cellule dendritiche dall' algoritmo proposto.

Viene calcolato utilizzando FP e TP:

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

Gli indici ottenuti sono riportati nella seguente tabella:

	Sensibilità	FDR
Algoritmo proposto	91.02 ± 12.33	22.68 ± 16.73

Tabella 4.1. **Risultati in termini di sensibilità e FDR (media ± deviazione standard di 100 immagini).**

Il modello proposto, dunque, ha permesso di ottenere ottimi risultati in termini di sensibilità. Tuttavia, presenta un valore ancora alto per quanto riguarda il FDR.

Ad ogni modo, tramite un'ispezione visiva è possibile ricercare alcune spiegazioni plausibili circa il numero elevato di FP, riscontrato soprattutto in determinate immagini.

Anzitutto, alcune immagini acquisite risultano essere molto rumorose, caratterizzate da una luminosità non uniforme e dalla presenza di “macchie” di pixel luminosi di cui non è chiara la natura, ma che la rete ha classificato come DCs (Figura 4.3). Tali regioni hanno spesso forma arrotondata e dimensioni ridotte, ma la difficoltà di eliminarle nella fase di *post-processing* risiede nell'eterogeneità morfologica delle cellule dendritiche: nella loro forma immatura, le DCs possono assumere una conformazione piccola e globulare, impedendo quindi la definizione di un'operazione di elaborazione morfologica in grado di rimuovere i falsi positivi, preservando al tempo stesso il riconoscimento automatico delle cellule dendritiche. Infatti, un problema osservato nella fase di *post-processing* di molte immagini è legato al filtraggio degli oggetti, riconosciuti dalla rete come DCs, sulla base dell'area: se da un lato questa operazione ha permesso di scartare molti FP, a favore del FDR, dall'altro ha portato a perdere alcuni TP, a scapito del TPR (Figura 4.4).

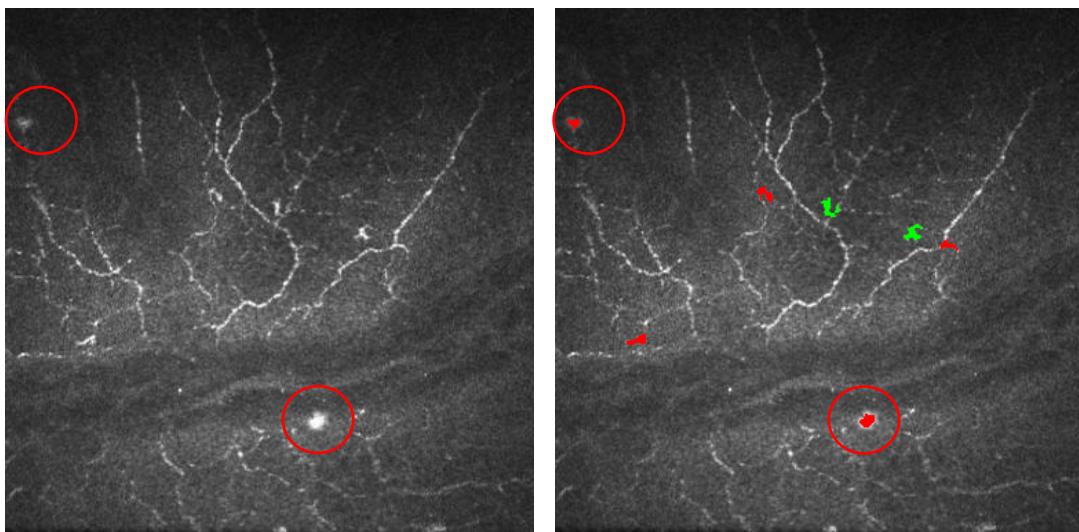


Figura 4.3. “Macchie” di pixel luminosi (indicate nel cerchio rosso) erroneamente classificati dall'algoritmo come cellule dendritiche, probabilmente a causa della somiglianza con la tipica morfologia di una DC immatura.

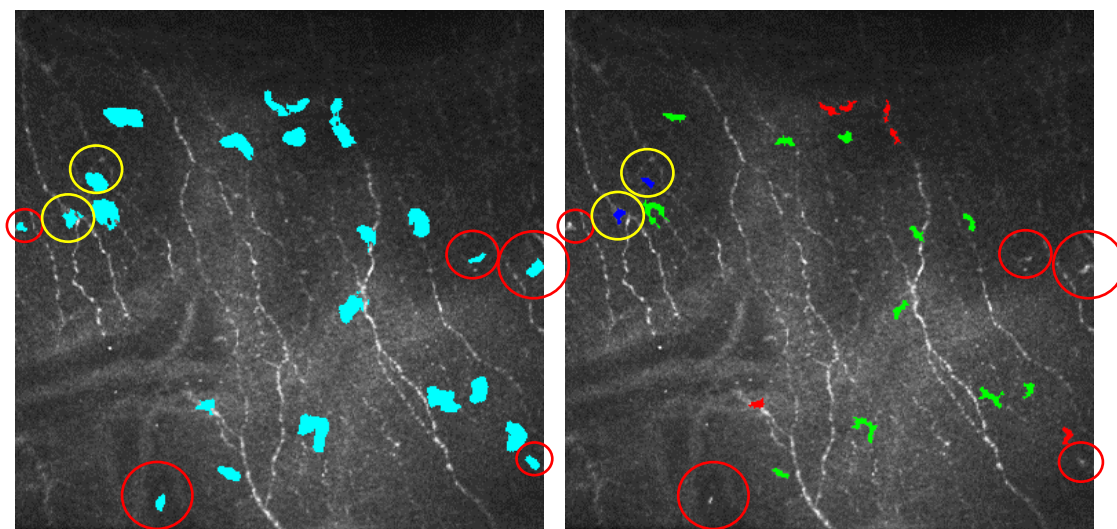


Figura 4.4. **Il *post-processing* porta a scartare falsi positivi perdendo allo stesso tempo sensibilità.** A sinistra, visualizzazione dell'output prodotto dalla rete. A destra, risultato del *post-processing*, in cui è possibile notare come l'operazione di filtraggio dell'area abbia portato a scartare diversi FP (cerchi rossi) perdendo tuttavia TP (cerchi gialli) che erano stati correttamente riconosciuti dall'algoritmo e che, dopo la soglia, sono diventati FN.

Altri FP sono invece legati all'acquisizione, nell'immagine di microscopia confocale, di altri strati della cornea, al cui interno la rete riconosce erroneamente delle cellule dendritiche.

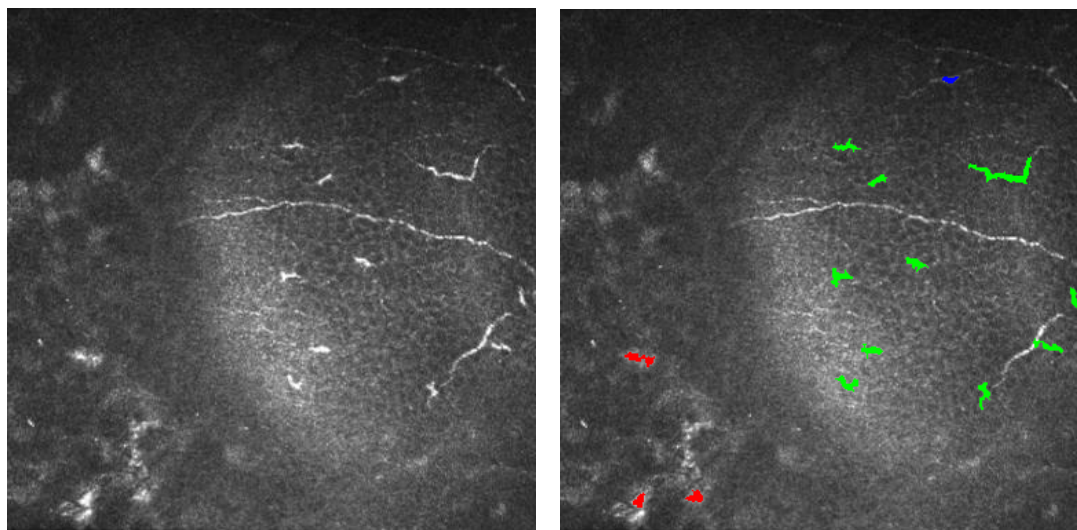


Figura 4.5. **Localizzazione di falsi positivi all'interno di uno strato della cornea diverso dal plesso nervoso sub-basale.**

Infine, è importante sottolineare come, attraverso un'accurata valutazione dei risultati dell'analisi automatica da parte dei clinici (fase di "revisione"), alcuni falsi positivi potrebbero essere ricondotti a cellule dendritiche precedentemente non riconosciute nell'analisi manuale.

Per concludere, viene riportato il confronto tra i risultati ottenuti in questa tesi, e quelli ottenuti utilizzando un algoritmo sviluppato in un lavoro di tesi precedente. Anche in tale caso, infatti, lo scopo consisteva nella segmentazione automatica di cellule dendritiche corneali presenti nelle immagini del plesso nervoso sub-basale della cornea, acquisite tramite microscopia confocale.

L'idea proposta in *Faoro* [46] prevedeva fondamentalmente tre passaggi:

1. Operazione di sogliatura, per evidenziare le strutture presenti nelle immagini originali.
2. Applicazione dell'algoritmo proposto in *Guimarães et al* per il *tracing* automatico dei nervi corneali [44]. In particolare, era stata creata una funzione per eliminare piccole diramazioni nervose laterali, corrispondenti a cellule dendritiche ma che l'algoritmo aveva classificato erroneamente come nervi.
3. Riconoscimento delle cellule dendritiche mediante sottrazione, dalle immagini sogliate, delle fibre nervose. L'idea di base prevedeva che, una volta eliminate le strutture nervose, gli oggetti rimasti rappresentassero le cellule dendritiche.

Per ottenere un confronto, le stesse cento immagini, utilizzate per la validazione del modello neurale proposto in questa tesi, sono state analizzate con l'algoritmo proposto nel precedente lavoro di tesi.

In tabella 4.2 sono riportati gli indici statistici ottenuti, che dimostrano come lo sviluppo di una rete neurale convoluzionale abbia portato a risultati decisamente migliori, specialmente in termini di FDR.

	Sensibilità	FDR
Algoritmo proposto	91.02 ± 12.33	22.68 ± 16.73
<i>Faoro</i> [1]	82.83 ± 18.03	49.60 ± 19.37

Tabella 4.2. **Confronto tra i risultati, in termini di sensibilità e FDR (media ± deviazione standard sulle stesse 100 immagini) ottenuti con l'algoritmo sviluppato in questa tesi e quello proposto in *Faoro* [46].**

Conclusioni

La microscopia confocale in vivo (IVCM) consente l'acquisizione, rapida e non invasiva, di immagini dei vari strati della cornea. In particolare, le immagini prese a livello del plesso nervoso sub-basale, permettono di visualizzare, oltre alle fibre nervose che lo caratterizzano, anche le cellule dendritiche (*Dendritic Cells*, DCs), la cui densità e morfologia è stata correlata con patologie oculari e/o sistemiche in diversi studi. Per poter studiare come forma e numero di cellule influenzino il manifestarsi e il progredire di alcune patologie risulta prima necessario identificare e segmentare accuratamente le DCs. Questo processo viene ad oggi effettuato manualmente dai clinici, con conseguenti difficoltà legate ai tempi di esecuzione, nonché alla soggettività di ogni analisi.

In questo lavoro, quindi, si è voluto automatizzare tale processo e, a tale scopo, si è sviluppato un algoritmo, basato su tecniche di *Deep Learning*.

In particolare, è stata implementata una rete neurale convoluzionale (*Convolutional Neural Network*, CNN) con architettura *U-Net*, in cui ad un percorso di codifica, usato per catturare descrittori robusti delle DCs, segue un percorso di decodifica simmetrico per la localizzazione spaziale delle cellule dendritiche.

La CNN è stata addestrata utilizzando, per la segmentazione manuale, le immagini IVCM fornite dai clinici della *School of Optometry and Vision Science* della *University of New South Wales* (UNSW, Sydney), in cui le cellule dendritiche erano state accuratamente identificate e i contorni delineati.

L'algoritmo proposto ha permesso di ottenere una sensibilità superiore al 90%, dimostrando la capacità della CNN di estrarre *features* in grado di descrivere correttamente le cellule dendritiche.

Tuttavia, nonostante il notevole miglioramento rispetto a un lavoro di tesi precedente, i risultati ottenuti mostrano un valore ancora elevato per il FDR. Tramite un'ispezione visiva è possibile ricercare alcune spiegazioni plausibili circa l'alto numero di FP, legate principalmente al livello di rumore presente in determinate immagini. Per esempio, a causa di difficoltà durante l'acquisizione, nelle immagini possono essere presenti altri strati della cornea, che si presentano come dei *pattern* classificati erroneamente dalla rete come cellule dendritiche.

Questi risultati incoraggiano ulteriori sviluppi volti, in particolar modo, a ridurre il numero di falsi positivi e incrementare la sensibilità. Sicuramente si possono migliorare le *performance* dell'algoritmo acquisendo più immagini per l'addestramento della rete neurale: come noto,

infatti, gli algoritmi di *Deep Learning* richiederebbero migliaia di immagini durante la fase di addestramento, mentre in questo caso ne sono state utilizzate solamente 100.

Sempre per l'incremento delle *performance*, sarà sicuramente interessante considerare la definizione di più *pattern*:

- sia legati all'eterogeneità morfologica delle cellule dendritiche, con l'introduzione di categorie che distinguano la "cellula dendritica matura" dalla "cellula dendritica immatura".
- sia legati alla presenza di altre strutture, come, ad esempio, i neuromi.

Come sopra riportato, l'identificazione e la segmentazione delle cellule dendritiche, presenti nello strato sub-basale della cornea, è il primo passo per l'automatizzazione dello studio della correlazione che può intercorrere tra DCs e alcune patologie oculari e/o sistemiche. Sarà quindi interessante lo sviluppo di un algoritmo in grado di classificare le immagini in sane e patologiche, a partire da alcuni parametri (come, ad esempio, la densità delle DCs) estratti tramite il lavoro presentato in questa tesi.

Bibliografia

1. Kaufman H. E., Barron B. A., McDonald M. B., Waltman S. R., *The Cornea*, Churchill Livingstone, 1988.
2. Sridhar MS., *Anatomy of cornea and ocular surface*, Indian J Ophthalmol, 2018; 66:190-4.
3. Yang A. Y., Chow J., Liu J., *Corneal innervation and sensation: the eye and beyond*, Yale Journal of biology and medicine, 2018; 13-21.
4. Takaaki H., Hiroki T., Reza D., *Novel insights into the immunoregulatory function and localization of dendritic cells*, Cornea, 2016; S49–S54.
5. Gao N., Lee P., Yu F.-S., *Intraepithelial dendritic cells and sensory nerves are structurally associated and functional interdependent in the cornea*, Scientific Reports, 2016.
6. Ganguly D., Haak S., Sisirak V., Reizis B., *The role of dendritic cells in autoimmunity*, Nat Rev Immunol, 2013; 13(8): 566–577.
7. Patente T.A., Pinho M.P., Oliveira A.A., Evangelista G.C.M., Bergami-Santos P.C., Barbuto J.A.M, *Human Dendritic Cells: Their Heterogeneity and Clinical Application Potential in Cancer Immunotherapy*, Front. Immunol., 2019; 9:3176.
8. Collin M, Bigley V., *Human dendritic cell subsets: an update*, Immunology, 2017; 154:3–20.
9. Yamaguchi T., Hamrah P., Shimazaki J., *Bilateral Alterations in Corneal Nerves, Dendritic Cells and Tear Cytokine Levels in Ocular Surface Disease*, Cornea, 2016; S65–S70.
10. Austin A., Lietman T., Rose-Nussbaumer J., *Update on the Management of Infectious Keratitis*, Ophthalmology, 2017; 124(11): 1678–1689.
11. Cruzat A., Witkin D., Baniyadi N., Zheng L., Ciolino J.B., Jurkunas U.V., Chodosh J., Pavan-Langston D., Dana R., Hamrah P., *Inflammation and the Nervous System: The Connection in the Cornea in Patients with Infectious Keratitis*, Investigative Ophthalmology & Visual Science, 2011.
12. Aksel S., Orhun H. K.S.S.A.A., Vedat H.C.I., Naci K., Hasan Y., *Behçet's disease: diagnostic and prognostic aspects of neurological involvement*, J Neurol, 2001; 248:95–103.

13. Bitirgen G., Tinkir Kayitmazbatir E., Satirtav G., Malik R. A., Ozkagnici A., *In Vivo Confocal Microscopic Evaluation of Corneal Nerve Fibers and Dendritic Cells in Patients With Behçet's Disease*, *Front. Neurol.*, 2018; 9:204.
14. Bitirgen G., Turkmen K., Malik R. A., Ozkagnici A., Zengin N., *Corneal Confocal Microscopy Detects Corneal Nerve Damage and Increased Dendritic Cells in Fabry Disease*, *Scientific Reports*, 2018.
15. Stephen C. P., Cintia S. de Paiva, *The pathophysiology of dry eye disease: What we know and future directions for research*, *Ophthalmology*, 2017; S4–S13.
16. Choi E.Y., Kang H.G., Lee C.H., Yeo A., Noh H.M., Gu N., et al., *Langerhans cells prevent subbasal nerve damage and upregulate neurotrophic factors in dry eye disease*, *PLoS ONE*, 2017;
17. Neil S. L., Reza A.B., Xu L., Tobias R. F., Johan Ä., Tor P.U., Lars B. D., Olov R., *Dendritic cell maturation in the corneal epithelium with onset of type 2 diabetes is associated with tumor necrosis factor receptor superfamily member 9*, *Scientific Reports*, 2018.
18. Cruzat A., Qazi Y., Hamrah P., *In Vivo Confocal Microscopy of Corneal Nerves in Health and Disease*, *Ocul Surf.*, 2017; 15(1): 15–47.
19. I. Jalbert, F. Stapleton, E. Papas, D. F. Sweeney, M. Coroneo, *In vivo confocal microscopy of the human cornea*, *Br J Ophthalmol*, 2003; 87:225–236.
20. You Y.J., Botelho J.P., *Corneal In Vivo Confocal Microscopy: Clinical Applications*, *Rhode Island Medical Journal*, 2016.
21. Shai Shalev-Shwartz, Shai Ben-David, *Understanding Machine Learning from theory to algorithms*, Cambridge University Press, 2014.
22. Goodfellow I., Bengio Y., Courville A., *Deep Learning*, MIT Press, 2016.
23. Ahuja AS, *The impact of artificial intelligence in medicine on the future role of the physician*, *PeerJ*, 2019, 7:e7702.
24. Haykin S., *Neural Networks and Learning Machines*, Pearson Prentice Hall, 2008.
25. Bishop C.M., *Pattern Recognition and Machine Learning*, Springer, 2006.
26. Litjens G., Kooi T., Bejnordi, BE, et al, *A survey on deep Learning in Medical Image Analysis*, *Medical Image Analysis*, 2017, 42: 60-88.
27. LeCun Y., Bengio Y., Hinton G., *Deep learning*, *Nature*, 2015, 521: 436- 444.
28. Watt J., Borhani R., Katsaggelos A. K., *Machine Learning Refined*, Cambridge University Press, 2016.

29. Seonwoo M., Byunghan L., Sungroh Y., *Deep learning in bioinformatics*, Briefings in Bioinformatics, 2017, 18(5):851–869.
30. Kingma, Diederik, Jimmy Ba, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980 (2014).
31. Bengio Y., *Practical Recommendations for Gradient-Based Training of Deep Architectures*, arXiv:1206.5533 (2012).
32. Koichiro Y., Hiroyuki A., Akira K., Shigeru K., Osamu A., *Deep learning with convolutional neural network in radiology*, Japanese Journal of Radiology , 2018, 36:257–272.
33. Krizhevsky A., Sutskever I., Hinton G. E., *ImageNet Classification with Deep Convolutional Neural Networks*, *Advances in Neural Information Processing Systems*, 2012, 25.
34. Ioffe, Sergey, Christian Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, preprint, arXiv:1502.03167 (2015).
35. Murphy K. P., *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012
36. Yamashita R., Nishio M., Do RKG, et al., *Convolutional neural networks: an overview and application in radiology*, Insights Imaging, 2018, 9:611-629.
37. Dumoulin V. , Visin F., *A guide to convolution arithmetic for deep learning*, arXiv:1603.07285v2 [stat.ML] 11 Jan (2018).
38. Nagi, J., F. Ducatelle, G. A. Di Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, L. M. Gambardella, *Max-Pooling Convolutional Neural Networks for Vision-based Hand Gesture Recognition*, IEEE International Conference on Signal and Image Processing Applications (ICSIPA2011), 2011.
39. Singh S, Kaushik S, Vats R, et al, *Semantic Segmentation Using Deep Learning*, IRJET. Volume: 06 Issue: 04| Apr 2019.
40. Badrinarayanan V., Kendall A., Cipolla R., *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*, arXiv:1511.00561v3 [cs.CV] 10 Oct 2016.
41. Ronneberger O, Fischer P, Brox T. *U-Net: convolutional networks for biomedical image segmentation*, In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, 9351:234-241. Springer, Cham, 2015.
42. Parampal et al., *Deep learning in ophthalmology: a review*, Canadian J Ophthalmol. 2018.

43. Ting et al., *Artificial intelligence and deep learning in ophthalmology*, Br J Ophthalmol. 2019.
44. Guimarães, P., Wigdahl, J., Ruggeri, A.: *A fast and efficient technique for the automatic tracing of corneal nerves in confocal microscopy*. Transl. Vis. Sci. Technol. 5(5), 7 (2016)
45. Colonna A, Scarpa F, Ruggeri A., *Segmentation of Corneal Nerves Using a U-Net-Based Convolutional Neural Network*, In: Stoyanov D. et al. (eds) OMIA 2018, COMPAY 2018. LNCS, 11039. Springer, Cham, 2018.
46. Faoro G., *Segmentazione automatica di cellule dendritiche in immagini corneali*, Tesi di Laurea in Ingegneria Biomedica, DEI, Università di Padova.