



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**



**DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE**

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

**CORSO DI LAUREA MAGISTRALE IN
BIOINGEGNERIA**

**APPLICAZIONE DI TECNICHE DI DEEP LEARNING PER LA
SEGMENTAZIONE DI CELLULE DENDRITICHE CORNEALI**

Relatore: Prof. Scarpa Fabio

Laureanda: Rappo Elena

Correlatore: Dott.ssa Colonna Alessia

ANNO ACCADEMICO 2021 – 2022

Data di laurea 13 dicembre 2022

Abstract

La microscopia confocale corneale (CCM) è una modalità di *imaging* non invasiva in grado di fornire una visualizzazione diretta, *in vivo* ed in tempo reale dello stato fisiologico presente, se sano o malato. In particolare, è possibile osservare, a livello del plesso nervoso sub-basale, la presenza di cellule dendritiche (DCs): le loro differenti condizioni di numerosità e morfologia consentono di diagnosticare patologie quali occhio secco, neuropatia diabetica, cheratiti fungine, ecc. L'applicazione dell'intelligenza artificiale per effettuare diagnosi si è nel tempo proposta come una metodologia per superare il dispendio temporale e la soggettività dovuti all'analisi condotta dal medico.

In questa tesi è stato implementato un algoritmo di *Deep Learning* per la segmentazione automatica delle cellule dendritiche corneali, facendo uso di un network neurale convoluzionale (CNN) di tipo U-Net costituito da un percorso di codifica e da uno di decodifica.

L'algoritmo sviluppato consente di individuare le cellule dendritiche con un *True Positive Rate* dell'80.7 % ed un *False Discovery Rate* del 21 %. Questi risultati dimostrano la possibilità di individuare in automatico le cellule dendritiche ed incoraggiano l'ulteriore sviluppo dell'algoritmo proposto.

Indice

Introduzione	7
Capitolo 1: La cornea	9
1.1 Epitelio corneale	11
1.1.1 Strato profondo	11
1.1.2 Strato intermedio	12
1.1.3 Strato superficiale	12
1.2 Membrana di Bowman	12
1.3 Stroma corneale	13
1.4 Membrana di Descemet	13
1.5 Endotelio corneale	13
1.6 Plesso sub basale e patologie	14
1.7 Cellule dendritiche corneali	17
1.7.1 Condizioni patologiche tramite CCM	19
1.7.1.1 Trattamento del cancro	20
1.7.1.2 Cheratocono	21
1.7.1.3 Occhio secco	22
Capitolo 2: Deep learning	24
2.1 Deep learning e limiti del machine learning	24
2.2 Networks neurali artificiali	26
2.3 Deep feed-forward networks	28
2.4 Funzioni d'attivazione	29
2.5 Addestramento tramite back-propagation	32
2.5.1 SGDM	35
2.6 Loss functions	37
2.6.1 Cross-Entropy	38
2.7Regolarizzazione	39
2.7.1 Overfitting	39
2.7.2 Tecniche di regolarizzazione	40

2.7.2.1	Weight decay	40
2.7.2.1.1	L^2 Regularization	41
2.7.2.2	Data augmentation	41
2.7.2.3	Dropout	42
2.7.2.4	Batch Normalization	43
2.8	CNNs	45
2.8.1	Layer di convoluzione	46
2.8.2	Layer di pooling	48
2.8.2.1	Max-pooling	48
2.8.2.2	Average-pooling	49
2.8.3	Fully connected layer	49
2.9	Segmentazione	50
2.9.1	Encoder-decoder	51
2.9.1.1	U-Net	52
2.10	Deep learning in medicina	54
2.10.1	Segmentazione di nervi corneali	55
2.10.2	Segmentazione di cellule dendritiche corneali	56
Capitolo 3: Segmentazione automatica delle cellule dendritiche corneali		58
3.1	Dataset	59
3.1.1	Immagini pixel-label	60
3.2	Data augmentation	61
3.2.1	Rotazione antioraria di 90°	61
3.2.2	Variazione di luminosità	62
3.2.3	Flip casuale	63
3.3	Addestramento della rete	64
3.3.1	Loss functions per la segmentazione semantica	64
3.3.1.1	Weighted Cross-Entropy	64
3.3.1.2	Generalized Dice Loss	65
3.3.2	Iper-parametri	65
3.3.3	Leave-One-Out-Cross-Validation	66
3.4	Post-processing	67

Capitolo 4: Risultati	70
4.1 Analisi statistica	70
Conclusioni	78
Bibliografia	80

Introduzione

La cornea è un tessuto altamente innervato con un'elevata sensibilità, importante per la sua protezione da agenti esterni. In precedenza, si era infatti osservato che ad una profondità di 35-60 μm , preferenzialmente tra l'epitelio basale ed il plesso nervoso sotto-basale della cornea, si trovano delle particolari cellule dette di Langerhans (LCs), o dendritiche (DCs), responsabili della risposta immunitaria adattativa dell'occhio.

Esse fanno parte del gruppo delle APC (*antigen presenting cell*): riconoscono i microbi che hanno superato la barriera epiteliale, in questo caso la cornea, distinguendosi tra cellule immature e mature. Le prime hanno il ruolo di catturare e fagocitare l'antigene presente, che viene riconosciuto per mezzo di particolari recettori specifici espressi da tali cellule, come PRR (*pattern recognition receptor*) e TLR (*toll-like receptor*), e mediante proteine specializzate chiamate MHC (*major histocompatibility complex*). Le seconde (quelle mature) derivano dalle cellule immature una volta incontrato l'antigene: iniziano ad esprimere i recettori per le *chemochine* ed hanno una funzione di regolamento della morfologia dei nervi corneali in base al grado di attivazione del sistema immunitario ed alla condizione fisiologica presente. Dunque, il ruolo delle cellule dendritiche corneali è molto importante e significativo: in presenza di una condizione anomala per la cornea, cambiano la loro densità e morfologia.

Grazie alla microscopia confocale corneale *in vivo* (IVCCM), è possibile visualizzare in maniera non invasiva la morfologia della cornea, osservando così la numerosità e la condizione delle cellule dendritiche presenti, e fornendo di conseguenza importanti informazioni sullo stato di salute del paziente e sul grado di severità della patologia, in modo tale da suggerire la cura più adatta.

In questo lavoro di tesi è stato progettato un algoritmo per la segmentazione automatica delle cellule dendritiche corneali, a partire da 100 immagini ottenute con la tecnica di IVCCM, a livello del plesso nervoso sub-basale della cornea di 100 differenti soggetti. Si è utilizzata la rete neurale convoluzionale (CCN) di tipo U-Net, particolarmente efficiente nella segmentazione di immagini mediche: essa è costituita da un percorso di codifica dei dati presenti in input e da un rispettivo percorso di decodifica, in modo tale da ottenere la segmentazione dell'oggetto d'interesse.

Nel primo capitolo viene descritta l'anatomia della cornea, dunque gli strati che la compongono e le principali caratteristiche, approfondendo, in particolare, il plesso nervoso sub-basale e le cellule dendritiche presenti, anche in condizioni patologiche.

Il secondo capitolo riguarda il *deep learning*, che viene inizialmente confrontato rispetto ai precedenti algoritmi di *machine learning*, per poi descrivere i networks neurali artificiali, in particolare i *feed-forward networks* e le loro proprietà, anche in fase di addestramento. Vengono poi approfonditi i CNNs e la segmentazione semantica con la rete U-Net, fornendo infine esempi di applicazione delle tecniche di *deep learning* in medicina, in particolare nella segmentazione delle cellule dendritiche e dei nervi corneali.

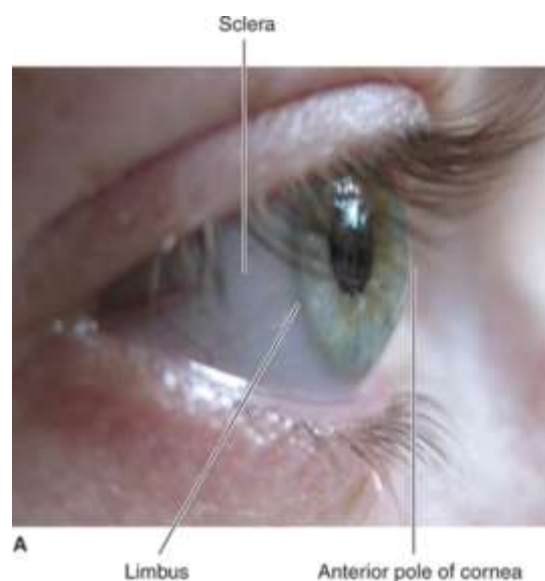
Nel terzo capitolo viene presentato l'algoritmo proposto per la segmentazione automatica delle cellule dendritiche corneali, descrivendo il materiale ed il procedimento impiegati.

Nell'ultimo capitolo, il quarto, vengono presentati i risultati ottenuti e, successivamente, si apportano delle considerazioni finali a riguardo.

Capitolo 1

La cornea

La cornea è un tessuto connettivo avascolare trasparente che agisce come prima barriera infettiva e strutturale dell'occhio [4, 9, 10]: è una lamina trasparente, inserita tra la sclera ed il *limbus* (Figura 1.1), ed è situata nella porzione anteriore della tonaca fibrosa. La componente strutturale principale della cornea è rappresentata dal tessuto connettivo collageno dello stroma. Nei soggetti adulti lo spessore varia da 0.5 mm nella sezione centrale a 0.8-1 mm in quella periferica, orizzontalmente il suo diametro è circa di 11.7 mm e verticalmente di 10.6 mm circa. Le funzioni principali della cornea sono intimamente legate alla sue proprietà di trasparenza e resistenza. La forma della cornea crea un sistema ottico non sferico.



Riordan-Eva P. *Anatomy & Embryology of the Eye*. In: Riordan-Eva P, Augsburger JJ. eds. *Vaughan & Asbury's General Ophthalmology*. 19e: McGraw Hill; 2017.

<https://accessmedicine.mhmedical.com/content.aspx?bookid=2186§ionid=165515771>

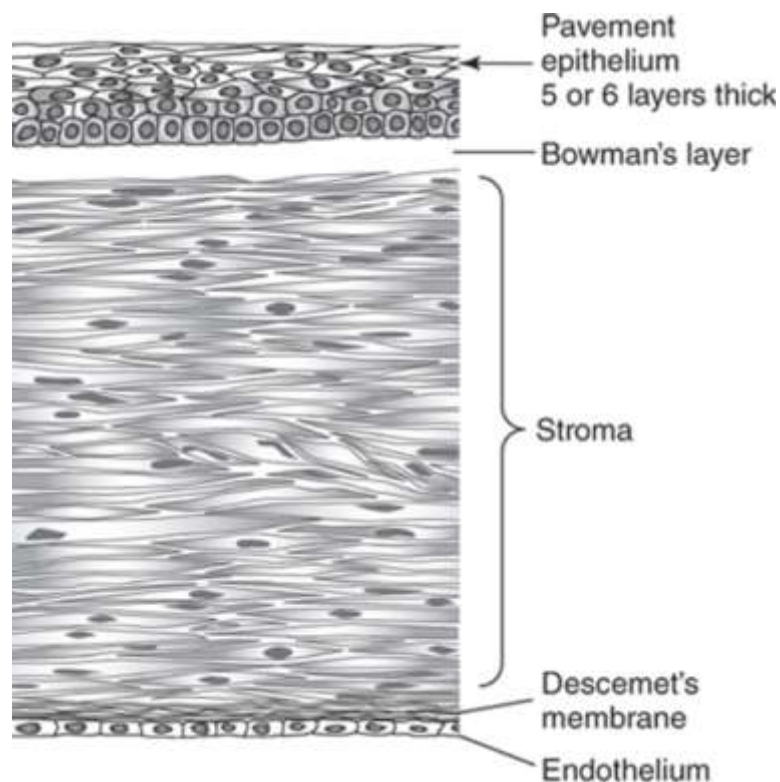
Figura 1.1: La cornea.

Nella cornea si distinguono due facce (interna ed esterna) ed una circonferenza. La faccia esterna, convessa, è a contatto con il liquido lacrimale e presenta un aspetto lievemente ovale con un

maggiore asse trasversale. La faccia interna, concava, a contatto con l'umor acqueo, ha aspetto circolare. La circonferenza della cornea è in rapporto di continuità con la sclera e prende il nome di orlo sclero-corneale o *limbus*. Lo spessore della cornea non è uniforme: è maggiore a livello del margine sclero-corneale (0.7 mm), superando quello della sclera, per poi ridursi verso il centro (0.5 mm) [8].

La cornea si compone di cinque strati (Figura 1.2) che in senso antero-posteriore sono:

1. Epitelio corneale;
2. Membrana di Bowman (o membrana limitante anteriore);
3. Stroma corneale;
4. Membrana di Descemet (o membrana limitante interna);
5. Endotelio corneale.



Riordan-Eva P. *Anatomy & Embryology of the Eye*. In: Riordan-Eva P, Augsburger JJ. eds. *Vaughan & Asbury's General Ophthalmology*. 19e: McGraw Hill; 2017.

<https://accessmedicine.mhmedical.com/content.aspx?bookid=2186§ionid=165515771>

Figura 1.2: Sezione trasversale della cornea.

1.1 Epitelio corneale

Esso crea la prima barriera nei confronti del mondo esterno, ed è parte dell'interfaccia tra il film lacrimale e la cornea, cruciale per il potere refrattivo dell'occhio. Il film lacrimale è il primo protettore della superficie corneale dall'invasione microbica, fornendo fattori di crescita e componenti immunologiche [4, 10]. Nello specifico, l'epitelio corneale è di tipo pavimentoso, pluristratificato e non cheratinizzato di 50 μm di spessore; è composto da 5 o 6 strati cellulari e di 3 citotipi: basali o colonnari, intermedi o poligonali e piatti o superficiali. Le cellule epiteliali progenitrici si trovano a ridosso del limbo basale.

Si possono distinguere tre strati differenti:

1. Lo strato profondo, che è composto da cellule basali;
2. Uno strato intermedio, il quale è formato da cellule poligonali;
3. Uno strato superficiale costituito da cellule appiattite.

1.1.1 Strato profondo

Gli elementi basali formano un unico strato di cellule batiprismatiche, alte 18-20 μm , e con diametro trasverso di 8-10 μm . Esse costituiscono la riserva germinativa dell'epitelio. Le cellule basali possiedono numerose giunzioni intercellulari (*gap-junctions*) e zonule aderenti, e risultano ancorate alla sottostante membrana basale per mezzo di un ampio sistema di emidesmosomi [4, 8, 10]. Con il trascorrere del tempo, per deposizione successiva di materiale da parte delle cellule basali, lo strato profondo aumenta di circa 3 nm all'anno e si raddoppia intorno ai 60 anni.

Quindi, le cellule basali hanno una notevole attività metabolica, sono soggette ad un processo di maturazione fino a desquamarsi una volta raggiunta la superficie. Le cellule basali sono le uniche (tra quelle appartenenti all'epitelio corneale) ad essere mitotiche; inoltre, sono attaccate alla sottostante membrana basale tramite un sistema emidesmosomiale. In questo strato, oltre ai linfociti si trovano le cellule di Langerhans.

1.1.2 Strato intermedio

Le cellule basali vanno incontro a modificazioni morfologiche e biochimiche che ne determinano la trasformazione in elementi di forma poligonale [4, 8]. Lo strato intermedio risulta infatti composto da cellule disposte su 3 o 5 file: le più profonde hanno forma poliedrica, mentre quelle più superficiali sono allungate orizzontalmente (cellule alari). Tali cellule sono provviste di numerose interdigitazioni e desmosomi. Nel citoplasma sono presenti discrete quantità di filamenti e microtubuli.

1.1.3 Strato superficiale

Migrando verso la superficie, gli elementi poligonali si trasformano in cellule squamose [4, 8]: le cellule superficiali sono disposte generalmente su due strati, hanno un aspetto pavimentoso con diametro maggiore di 40-60 μm . Hanno due funzioni principali: quella di mantenere ancorato il film lacrimale all'epitelio corneale, e quella di costituire una barriera anatomica tra quest'ultimo e l'ambiente extracellulare. Si possono distinguere cellule chiare e scure: le prime più giovani, mentre le seconde in via di desquamazione. Si possono trovare numerosi microvilli e micropieghe sulle membrane cellulari apicali delle cellule più superficiali, e giunzioni serrate sulle superfici laterali degli spazi intercellulari epiteliali.

1.2 Membrana di Bowman

È una membrana acellulare, uniforme, con uno spessore di 8-16 μm posta tra epitelio e stroma, interrompendosi a livello del *limbus*. È costituita da un sottile reticolo di fibrille di collagene distribuite in maniera dispersa. Alla microscopia ottica la sua superficie anteriore è omogenea e regolare, quella posteriore invece è irregolare: i suoi fasci di fibrille si intersecano con quelli dello stroma, da cui non può essere separata. Per la sua resistenza e compattezza tale struttura riveste un ruolo non trascurabile per il mantenimento e la stabilità della curvatura corneale [8]; se distrutta, non si può rigenerare e ciò che ne risulta è una cicatrice.

1.3 Stroma corneale

Lo stroma compone il 90% dello spessore corneale, è costituito da tessuto connettivo fibroso a fibre incrociate [4, 10] ed è avascolare. Le fibre di collagene rappresentano l'80% circa del peso secco della cornea, hanno un diametro uniforme di 25-30 nm e sono organizzate in lamelle. Le fibre appartenenti alla singola lamella sono raccolte in fasci paralleli, e risultano disposte ortogonalmente rispetto alle fibre delle lamelle adiacenti: questo consente l'azzeramento della rifrazione della luce incidente, conferendo alla cornea la sua proprietà di trasparenza. La struttura entro la quale le lamelle sono disposte varia in base alla profondità dello stroma, risultando più organizzate in profondità e meno in superficie (particolarmente evidente in presenza di *edema corneale*). E' presente inoltre un sottile reticolo di fibre elastiche. La matrice amorfa extracellulare (15% del peso corneale) circonda le fibre di collagene ed è ricca di glicosamminoglicani (GAGs); svolge un ruolo costante nel mantenere la distanza tra le fibre pari a 60 nm. La componente cellulare è formata da due citotipi: cellule fisse e mobili. Le prime (cellule fisse) sono i cheratociti (5% del peso corneale): si trovano nell'interstizio tra le lamelle, costituiscono la maggioranza tra le cellule dello stroma e sono coinvolte nel mantenimento della matrice extracellulare. Le seconde, quelle mobili, riguardano quegli elementi che provengono, per diapedesi, dai vasi sanguigni (monociti e leucociti polimorfonucleati).

1.4 Membrana di Descemet

È una spessa membrana basale trasparente situata tra lo stroma e l'endotelio, ed è sintetizzata da cellule endoteliali e cheratociti [4, 8]. Ha uno spessore variabile nell'adulto, pari a 5 µm al centro della cornea e 10 µm alla periferia. Tale membrana è composta da collagene di tipo IV e da fibronectina. In prossimità del contorno si osservano dei rilievi (corpi di Hassan-Henle), risultando elastica e molto resistente ai traumi: infatti, una volta lesa può rigenerarsi.

1.5 Endotelio corneale

È lo strato più interno della cornea, la sua faccia esterna è addossata alla membrana di Descemet, quella interna è a contatto con l'umore acqueo della camera anteriore [4, 8, 10]. È un epitelio pavimentoso monostratificato formato da elementi poligonali: la densità cellulare si modifica anche

nel corso della vita: da 3500-4000 cell/mm² alla nascita a 2500-3000 cell/mm² in età adulta. Col tempo, queste cellule si appiattiscono e diventano strettamente aderenti l'una all'altra, stabilizzandosi ad uno spessore di 4 µm. Le cellule adulte hanno perso la capacità mitotica: aumenta così l'area cellulare media e si riduce la densità cellulare. Gli endotelioцитi corneali sono dotati di un imponente sistema citoplasmatico finalizzato ad un'intensa produzione di energia. Anche qui, in corrispondenza della superficie laterale delle cellule sono presenti diversi tipi di giunzioni intercellulari (zone *occludens* e desmosomi).

1.6 Plesso sub-basale e patologie

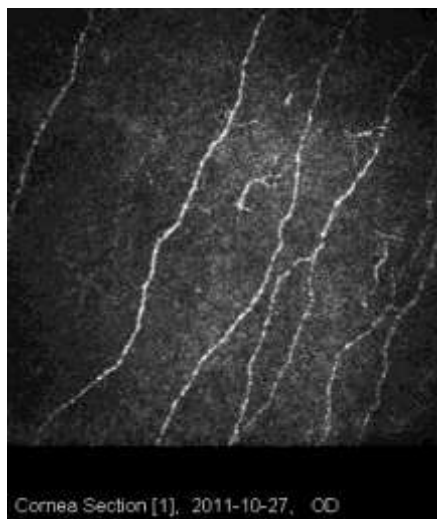
La cornea è un tessuto altamente innervato: la maggior parte dei nervi ha un'elevata sensibilità, importante per un'adeguata protezione da agenti infettivi ed irritanti. L'innervazione sensoriale è costituita dalle fibre terminali dei nervi ciliari lunghi che derivano dal naso-ciliare della branca oftalmica del trigemino, e sono presenti nello strato epiteliale ed in quello superficiale [8, 11].

Prima di raggiungere la cornea, le fibre nervose perdono la guaina mielinica e, giunte all'epitelio, mancano anche della guaina di Schwann. L'endotelio e la membrana di Descemet non sono innervati.

Il plesso del nervo sotto-basale (SNP) è localizzato appena sotto l'epitelio basale, anteriormente alla membrana di Bowman; il suo spessore lo rende facilmente rappresentabile e visibile. La tecnica di microscopia confocale corneale (CCM), minimamente invasiva ed *in vivo*, ha un'alta risoluzione e permette la valutazione dell'innervazione corneale; fornisce così un importante strumento di indagine clinica. In presenza di patologie o in conseguenza di interventi chirurgici o farmacologici, la cornea evidenzia una mancanza della sua normale innervazione. Lo SNP mostra una forma a spirale, all'interno del quadrante nasale basso. E' orientato per lo più verticalmente presso la parte centrale della cornea e forma un network comunicante (Figura 1.3).

Lo SNP è una struttura molto dinamica: i suoi punti d'innervazione sono soggetti ad un movimento centripeto superiore a 26 µm ogni settimana, con un periodo complessivo di sei settimane, dopo le quali i cambiamenti ottenuti sono molto evidenti. Le cellule epiteliali corneali e lo SNP si spostano in serie. La densità di SNP e la sensibilità corneale sono maggiori nella cornea centrale, proprio perché sono queste le regioni con il più alto rischio di traumi.

La CCM permette quindi di investigare sullo stato dei vari strati corneali in determinate condizioni patologiche. Una di queste è la cheratite infettiva causata da funghi e batteri, dove si osserva una riduzione del 70% dell'innervazione corneale rispetto ad un soggetto sano; la *cheratite* da *acanthamoeba* è la più severa e debilitante. Indipendentemente dalla tipologia di cheratite, il livello di rigenerazione nervosa non riuscirà a raggiungere le normali condizioni.

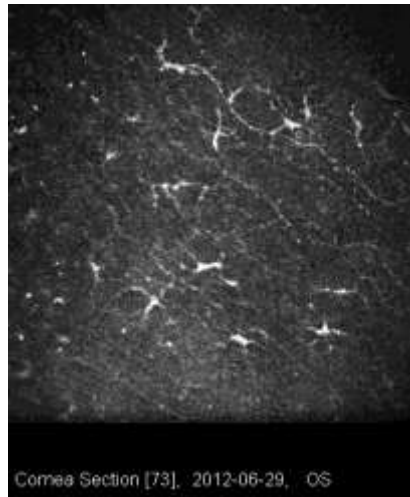


Kokot J, Wylęgała A, Wowra B, Wójcik Ł, Dobrowolski D, Wylęgała E. *Corneal confocal sub-basal nerve plexus evaluation: a review*. Acta Ophthalmol; 2018. 96(3): 233 p. doi: 10.1111/aos.13518. Epub 2017. PMID: 28741902.

Figura 1.3 : SNP in soggetti sani.

Anche in presenza di *cheratiti* da *herpes simplex* (HSK) i parametri medi nervosi sono più bassi rispetto alle condizioni normali (Figura 1.4).

La diminuzione di SNP dipende, ad esempio, dall'età del paziente e dalla durata della malattia, a cui conseguono significativi mutamenti di morfologia e densità nervosa, e sono permanenti. In particolare, una situazione anormale è data da una densità di SNP $\leq 835 \mu\text{m}/\text{frame}$. I soggetti con *cheratiti acute*, *fungine* o da *acanthamoeba* manifestano una significativa decrescita della densità di SNP, NFL (*nerve fibre length*) e diramazione nervosa.



Kokot J, Wylęgała A, Wowra B, Wójcik Ł, Dobrowolski D, Wylęgała E. *Corneal confocal sub-basal nerve plexus evaluation: a review*. Acta Ophthalmol; 2018. 96(3): 234 p. doi: 10.1111/aos.13518. Epub 2017. PMID: 28741902.

Figura 1.4: SNP in pazienti con cheratite da *herpes simplex*.

Ancora, una diminuzione di SNP ed una maggiore tortuosità dei nervi è presente in occhi affetti da HZO (*herpes zoster oftalmico*). Una volta che l'innervazione decresce, si evidenzia una maggiore risposta immunitaria. In presenza di *cheratocongiuntivite cronica vernal* (VKC) la microscopia confocale riesce ad evidenziare che l'epitelio superficiale e basale, lo stroma anteriore, il nervo dello stroma e quello sotto-basale subiscono cambiamenti. Oltre alle conseguenze già evidenziate, si osserva una maggiore percentuale di cellule infiammatorie.

Anche a seguito di *cheratocono*, già dopo poco tempo la densità media di SNP risulta ridotta ed il decorso dei nervi è più tortuoso; questo aiuta a diagnosticare precocemente la presenza di questa malattia. Ciò nonostante, nei pazienti con *cheratocono unilaterale* i fattori elencati in precedenza risultano anche nell'occhio sano.

Inoltre, in soggetti affetti da *occhio secco* (Sjögren e non) è possibile notare, con la tecnica di CCM, una decrescita di sensibilità corneale, numero e densità di SNP, ed un aumento della tortuosità nervosa. La perdita di sensibilità corneale può essere una conseguenza di un processo infiammatorio. Anche in pazienti con *glaucoma* si è osservato un aumento di tortuosità ed una decrescita di fibre nervose, come in soggetti con *diabete mellito* (DM), che ha un effetto di ispessimento e tortuosità nervosa. In soggetti con *neuropatia diabetica* decresce anche la riflettività.

I cambiamenti dello stato dei nervi a seguito dell'insorgenza di una malattia sono maggiormente evidenti in prossimità della parte inferiore della cornea, che ha la più alta predisposizione ad un danno nervoso. In particolare, la tipologia di *diabete* influisce sull'estensione del danno nervoso nella cornea, poiché le differenze nella produzione di insulina sono legate alla funzione nervosa. Anche a seguito di un intervento chirurgico la densità corneale di SNP si riduce, esponendo un soggetto diabetico all'insorgenza di *cheratopatia diabetica*.

1.7 Cellule dendritiche corneali

Le cellule dendritiche (DCs) nascono dalle cellule staminali emopoietiche nel midollo osseo e sono le più importanti tra le APC (*antigen presenting cell*), insieme ai macrofagi e ai linfociti B [1]. Il loro nome deriva dalla particolare forma ramificata che assumono poiché sono caratterizzate da lunghe protrusioni della membrana, simili ai dendriti di una cellula nervosa. Le cellule dendritiche esercitano la funzione di sentinelle nei confronti delle infezioni microbiche, riconoscendo gli agenti patogeni che hanno superato la barriera epiteliale. Sono sia cellule residenti nei tessuti sia circolanti, che percepiscono la presenza dei microrganismi, iniziano reazioni di immunità innata e presentano proteine microbiche ai linfociti T per generare una risposta immunitaria adattativa. Inoltre, le cellule dendritiche rendono le cellule T tolleranti agli antigeni innati nel corpo (antigeni *self*), minimizzando in questo modo le reazioni autoimmuni; dunque, il loro ruolo è molto importante.

All'interno dell'organismo e in assenza di stimoli, le cellule dendritiche sono distribuite in tutti i tessuti, dove sono in uno stato immaturo o di quiescenza: in questa fase, sono presenti nei tessuti periferici, e in particolare dov'è più comune incontrare antigeni (cute e mucose). Il riconoscimento dell'antigene avviene per mezzo di particolari recettori specifici espressi dalle DCs, come PRR (*pattern recognition receptor*) e TLR (*toll-like receptor*), e mediante proteine specializzate chiamate MHC (*major histocompatibility complex*) [3]. L'incontro con l'antigene dà inizio alla maturazione delle cellule dendritiche: una volta attivate, iniziano ad esprimere i recettori per le *chemochine* che ne facilitano la migrazione ai siti dove sono presenti i linfociti T. Il riconoscimento di microbi causa la secrezione di *citochine* pro-infiammatorie che reclutano e attivano cellule dell'immunità innata nei siti d'infezione, come i leucociti. Le molecole di membrana delle APC vengono chiamate *costimolatorie*. La cute, gli epiteli mucosi e gli organi parenchimatosi possiedono numerosi capillari linfatici, che drenano la linfa e convogliano gli antigeni da questi siti verso i linfonodi regionali. Le cellule dendritiche, una volta mature, hanno inoltre una funzione di regolamento della morfologia dei nervi corneali sulla base dell'attività del sistema immunitario presente.

Per quanto riguarda l'occhio, la sua difesa fisiologica è dovuta all'azione di cellule e sostanze normalmente presenti sulla superficie corneale e congiuntivale (dove appaiono più addensate), ed in prossimità della periferia della cornea, specialmente nello strato basale dell'epitelio. Nello specifico, all'interno dell'epitelio congiuntivale (in particolare nel *limbus*) sono state rinvenute le cellule di Langerhans (LCs), caratterizzate da numerosi e sottili prolungamenti che le consente di aderire alle cellule adiacenti. Le LCs solitamente si trovano vicine alle fibre nervose circostanti, poiché impiegano il plesso nervoso per il proprio movimento intraepiteliale. I processi dendritici a livello dello strato basale possono perforare anche la membrana basale.

Inoltre, le cellule di Langerhans possiedono recettori superficiali particolari e siti recettoriali per i *proantigeni*, come i determinanti antigenici (Ia-HLA), rendendole adatte a presentare nuovi antigeni alle cellule T *helper* ed innescando, in questo modo, il meccanismo della difesa immunitaria specifica [8].

Nella cornea e nella congiuntiva, in condizioni normali, non esistono altri elementi dotati di determinanti Ia-HLA oltre alle cellule di Langerhans; questo significa che tali cellule sono una componente decisiva della risposta immunitaria specifica oculare (le cellule dendritiche sono più efficienti rispetto ai macrofagi nell'induzione della risposta mediata dalle cellule T).

La fase iniziale del meccanismo di difesa della superficie oculare è basata sia sulle sostanze protettive che sui contributi cellulari ed umorali. Tali fattori proteggono l'occhio dalle infezioni ed elaborano, per la distruzione di batteri, virus o altri agenti patogeni, alcune sostanze di natura potenzialmente antigenica. Questi *proantigeni* possono aderire sia alla superficie delle cellule di Langerhans che a quella dei macrofagi. Ne sono un esempio i determinanti HLA-DR che consentono di elaborare gli antigeni da presentare alle cellule T *helper* innescando il meccanismo di difesa immunologica specifica.

Sulla superficie oculare, dunque, si verificano due distinti meccanismi di elaborazione degli antigeni: il primo, usufruendo dell'azione dei costituenti del film lacrimale, degrada il materiale estraneo ed elabora i *proantigeni* mediante le cellule di Langerhans (difesa fisiologica), il secondo avviene tramite i leucociti ed i fattori umorali, elabora i *proantigeni* determinando una reazione infiammatoria (difesa patologica).

Le cellule immunitarie che controllano l'immunità innata nella cornea sono: cellule dendritiche (DCs), mastociti (MCs), macrofagi, cellule *natural killer* (NKs), cellule T- $\gamma\delta$ e cellule linfoidi innate (ILCs). Sono necessarie all'omeostasi corneale ed a far fronte alla presenza di patologie. Il

limbus corneale ha una ricca distribuzione di capillari e vasi linfatici che servono come entrata ed uscita delle cellule immunitarie.

Nel corso dello sviluppo (periodo embrionale), le LCs provengono da cellule progenitrici dei macrofagi del sacco vitellino e del fegato. Hanno un livello di proliferazione pari a 5% a volta, poi migrano verso i linfonodi per completare il processo di difesa fisiologica. La capacità di mantenere il network di LCs dipende da: fattori di crescita (TGF- β 1) e di trascrizione (*Runx3*, *ID2*, *PU.1* e *PI4*). Lo sviluppo delle cellule di Langerhans dipende anche da segnali come IL-34, mediati dal recettore CSF-1 (*colony stimulating factor*). Quando le LCs sono danneggiate, vengono sostituite da monociti provenienti dal midollo osseo, che producono una crescita nella concentrazione di *chemochine*.

Uno studio recente [7] ha mostrato che la langerina (CD207) è un marcatore altamente selettivo per le LCs. È un recettore impiegato per la cattura dell'antigene: una volta catturato, la langerina lo internalizza formando così un organulo specifico (granulo di Birbeck) che diventa un compartimento specifico per il *processing* dell'antigene. Si è evidenziato, inoltre, che le cellule di Langerhans producono delle cellule T regolatorie (Tregs) quando la pelle è esposta alla radiazione ionizzante.

1.7.1 Condizioni patologiche tramite CCM

La cornea è facilmente accessibile tramite microscopia confocale, che consente di evidenziare, ad alta risoluzione ed in modo non invasivo, le cellule dendritiche presenti nel plesso nervoso sottobasale, le quali possono essere distinte tra immature e mature. La presenza di cellule dendritiche immature è legata alla capacità di far fronte ad una potenziale risposta infiammatoria, con abilità di cattura e fagocitosi dell'antigene; le cellule dendritiche mature, invece, derivano da quelle immature una volta che vengono espresse molecole *costimolatorie* come CD80/CD86 e MHC-II, stimolando così le cellule T.

In condizioni normali, le cellule dendritiche immature risiedono nell'epitelio e nello stroma anteriore della cornea centrale, una posizione strategica che permette loro di rispondere agli agenti patogeni che invadono la superficie della cornea [13].

Utilizzando la CCM è possibile rilevare che, in seguito ad una situazione di infiammazione oculare dovuta alla presenza di una condizione anomala per l'occhio, c'è un'alterazione della densità e della morfologia delle cellule di Langerhans, proporzionale allo stato di gravità della patologia.

Vengono riportati in seguito tre esempi di patologie della cornea quali *cancro*, *cheratocono* e *occhio secco*, sottolineando come, grazie alla tecnica di microscopia confocale corneale, risulta possibile riscontrare e diagnosticare tali patologie tramite una modifica di densità e morfologia delle cellule dendritiche.

1.7.1.1 *Trattamento del cancro*

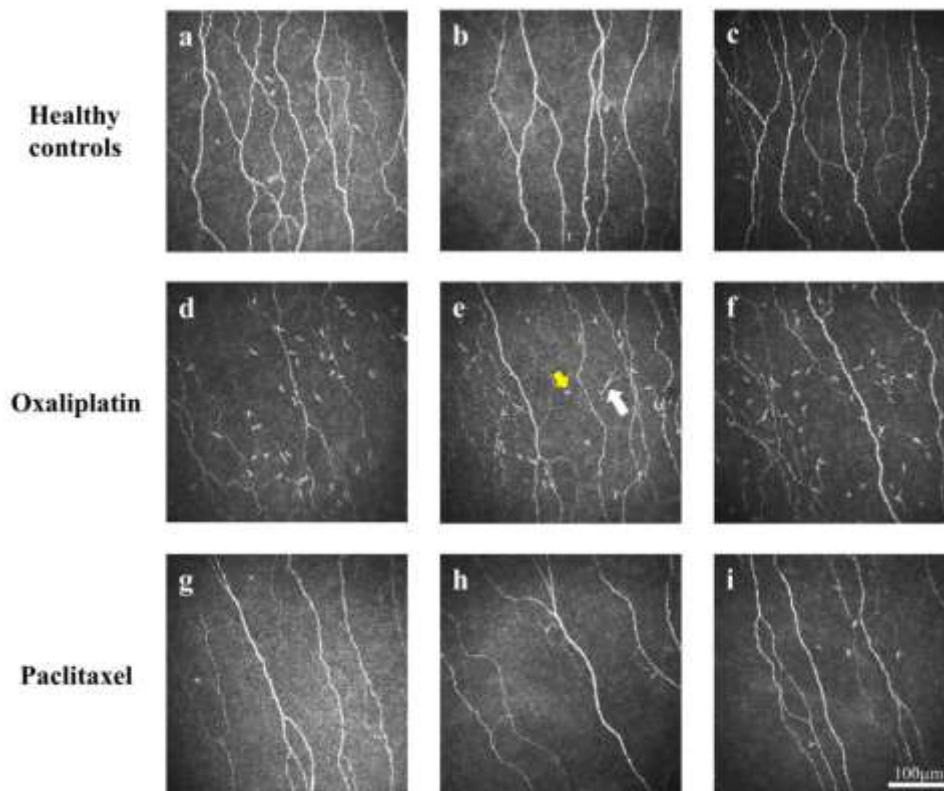
Come trattamento per il *cancro* è comune la chemioterapia, che comporta l'impiego di farmaci potenzialmente tossici per il sistema nervoso periferico: *oxaliplatin* e *paclitaxel* sono gli agenti citotossici più comuni associati alla *neuropatia periferica*.

Dal momento che la cornea è facilmente accessibile tramite microscopia confocale, è possibile evidenziare, ad alta risoluzione e in modo non invasivo, le cellule immunitarie presenti nel plesso nervoso sotto-basale; con riferimento ad uno studio [6] in cui 39 pazienti sono stati trattati con *oxaliplatin*, 48 con *paclitaxel* e 40 i soggetti di controllo, si sono potute identificare le densità di cellule dendritiche immature (ImDCs), mature (MDCs) e totali (TotalDCs).

Si può notare in Figura 1.5 la differenza tra le cellule dendritiche immature (freccia gialla) e mature (freccia bianca). Le prime sono piccole, in qualità di corpi cellulari riflettenti, con dendriti ancora piccoli (<25 µm) e terminale conico. Le seconde hanno corpi cellulari riflettenti e snelli, spesso con processi dendriformi multipli che si estendono all'infuori dal principale corpo cellulare (>25 µm).

Ad esempio, è interessante notare che il valore di ImDCs è più alto in soggetti trattati con *oxaliplatin* (22.7 cellule/mm²) se paragonato con i soggetti di controllo (10.1 cellule/mm²), mentre non c'è molta differenza tra i soggetti trattati con *paclitaxel* (10.6 cellule/mm²) e quelli di controllo. Se la densità di cellule dendritiche immature è più alta, significa che, potenzialmente, la capacità di catturare e processare l'antigene è elevata.

Il valore di MDCs invece risulta simile tra i tre gruppi, indicando che l'assunzione di tali farmaci non ha prodotto o stimolato una forte reazione infiammatoria. Risulta così evidente la correlazione tra densità di cellule dendritiche e reazione infiammatoria.



Chiang JCB, Goldstein D, Tavakoli A, Trinh T, Klisser J, Lewis CR, Friedlander M, Naduvilath TJ, Au K, Park SB, Krishnan AV, Markoulli M. *Corneal dendritic cells and the subbasal nerve plexus following neurotoxic treatment with oxaliplatin or paclitaxel*. Sci Rep; 2021. 11(1): 3 p. doi: 10.1038/s41598-021-02439-0. PMID: 34819589; PMCID: PMC8613280.

Figura 1.5 : Microscopia confocale *in-vivo*. Le sezioni a-c) si riferiscono a soggetti sani, le d-f) a quelli trattati con *oxaliplatin* e le restanti g-i) ai soggetti con *paclitaxel*.

1.7.1.2 Cheratocono

In un altro studio [5], si sono valutate le condizioni della cornea centrale in 21 soggetti affetti da cheratocono (lieve o severo), dai 18 ai 65 anni, riscontrando, anche qui, una correlazione tra tale patologia e le caratteristiche di densità e morfologia delle cellule di Langerhans. In particolare, sono state considerate quattro situazioni (da 0 a 3): 0 per l'assenza di cellule dendritiche, 1 per le cellule senza dendriti, 2 per le cellule di Langerhans con dendriti corti e 3 per le cellule di Langerhans con dendriti lunghi, visibili.

Da notare che le cellule di Langerhans sono state trovate nella cornea centrale per il 91% dei soggetti con *cheratocono*: quelle di grado 1 nel 19%, mentre di grado 2 o 3 nel 71%. Il valore mediano della densità di tali cellule è risultato pari a 15 cell/mm².

Le cellule di Langerhans immature hanno una bassa capacità di stimolare le cellule T, invece quelle mature hanno una bassa capacità di catturare l'antigene ma sono potenti stimolatrici di cellule T. Nella cornea centrale le cellule di Langerhans sono per la maggior parte immature: la presenza di cellule mature suggerisce uno stato infiammatorio; in soggetti affetti da *cheratocono*, si è dunque visto che le LCs mature si trovano anche nella cornea centrale, oltre a quella periferica dove sono solitamente presenti: è così possibile diagnosticare tramite CCM tale patologia, che si è riscontrato avere conseguenze sul sistema nervoso circostante.

1.7.1.3 Occhio secco

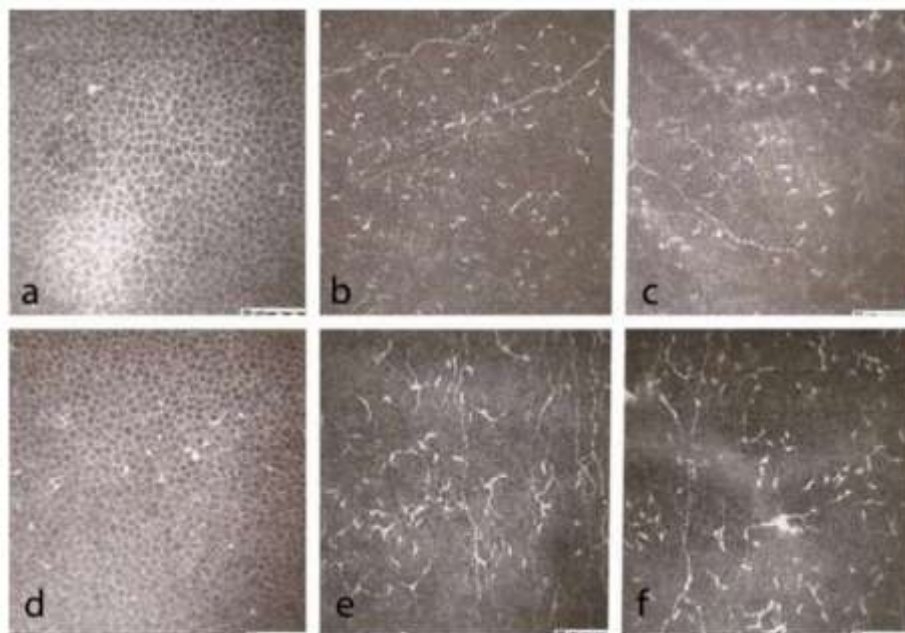
L'*occhio secco* comporta un'infiammazione della superficie oculare a cui sono talvolta associati *disturbi reumatologici*. Si è visto che più del 6% della popolazione al di sopra dei 40 anni ne soffre: aumenta prevalentemente in base all'età ed interessa maggiormente il sesso femminile.

In uno studio [12] sono stati considerati 3 gruppi di soggetti: 18 affetti da SSDE (*Sjögren syndrome dry eye*), 17 con NSSDE (*non Sjögren syndrome dry eye*) e 14 volontari sani (NDE, *non dry eye*), laddove con *Sjogren syndrome* si intende una patologia autoimmune multi-sistemica che comporta un'ipo-funzione delle ghiandole lacrimali. In particolare, sono state valutate le cellule appartenenti a tre differenti aree: la fascia intermedia dell'epitelio (tra lo strato superficiale e basale), il plesso nervoso sotto-basale e quello nervoso sotto-epiteliale, che compone i 2/3 dello stroma, in cui penetrano le fibre nervose di ampia dimensione.

Ciò che si riscontra è una crescita della densità delle cellule di Langerhans della cornea centrale (valori mediani) del plesso nervoso sotto-basale rispetto a quello sotto-epiteliale, ed un maggior numero di LCs nella periferia della cornea rispetto alla parte centrale, la cui differenza tra periferia e centro è più grande nei soggetti NDE, suggerendo in questo modo che la patologia aumenta il numero di cellule di Langerhans sia nella cornea centrale che in quella periferica. Calcolando la media delle densità di LCs nei tre differenti strati epiteliali della cornea centrale e di quella periferica dei tre gruppi di soggetti, si è inoltre osservato che il cambiamento dovuto alla condizione patologica di *occhio secco* non ha prodotto un evidente cambiamento nella cornea periferica.

Grazie alla microscopia confocale, con quest'analisi si può accertare che, per notare la presenza di patologia di *occhio secco* in termini di cellule di Langerhans, è sufficiente concentrarsi sulla cornea centrale [12].

Si può notare in Figura 1.6 la presenza di cellule di Langerhans corneali tramite CCM in soggetti SSDE: nell'*occhio secco*, le LCs con i dendriti sono più frequenti nella cornea periferica, mentre quelle senza i dendriti sono più frequenti in quella centrale.



Machetta F, Fea AM, Actis AG, de Sanctis U, Dalmaso P, Grignolo FM. *In vivo confocal microscopic evaluation of corneal langerhans cells in dry eye patients*. Open Ophthalmol J.; 2014. 8: 58 p. Doi: 10.2174/1874364101408010051. PMID: 25317216; PMCID: PMC4195179.

Figura 1.6: Immagini tramite CCM di soggetti con SSDE. Cornea centrale: a) epitelio intermedio, b) plesso nervoso sotto-basale, c) plesso nervoso sotto-epiteliale; cornea periferica: d) epitelio intermedio, e) plesso nervoso sotto-basale, f) plesso nervoso sotto-epiteliale.

Capitolo 2

Deep learning

2.1 Deep learning e limiti del machine learning

Il *machine learning* ed il *deep learning* appartengono al ramo dell'intelligenza artificiale (AI), che si prefigge di conferire ai computer le capacità di apprendimento e soluzione ai problemi proprie dell'uomo, con applicazioni, ad esempio, nel campo della *computer vision* [14, 15, 16].

Gli algoritmi di *machine learning* sono una tecnica di apprendimento automatico per classificare ed identificare dati “grezzi” quali immagini o video, che costituiscono l'*input*: analizzando e prendendo decisioni a partire da tali dati, viene approssimata una determinata funzione f (che soddisfa particolari caratteristiche) in modo da ottenere il valore di *output* desiderato. Questi algoritmi hanno però una capacità limitata nel processare i dati presenti in input nella loro forma originale, poiché richiedono anche l'intervento di un'appropriata figura professionale ed esperta.

Sia nel *machine learning* che nel *deep learning* si impiegano due tipologie di apprendimento, *supervisionato* (vengono fornite le *etichette* dei dati presenti in input, rendendo noto l'output desiderato), e *non supervisionato* (caso contrario). Ad esempio, l'apprendimento supervisionato viene utilizzato per svolgere *task* specifici quali la classificazione di oggetti presenti in un'immagine.

Uno dei motivi per il quale si è spostato l'interesse dal *machine learning* al *deep learning* è principalmente perché gli algoritmi del primo non si sono mostrati sufficientemente capaci di condurre funzioni complesse in spazi di elevate dimensioni; i dati presenti in input, infatti, molte volte possono contenere informazioni che risultano ridondanti, rendendo così necessaria una struttura computazionale più complessa. Il *deep learning* è una tecnica di *machine learning* più avanzata, che consente ai computer di condurre in maniera completamente automatica l'apprendimento di informazioni utili a partire da dati grezzi, migliorando l'accuratezza ottenuta.

Come è stato osservato, i sistemi di intelligenza artificiale acquisiscono conoscenza estraendo *categorie* a partire da dati iniziali grezzi; tuttavia, a differenza del *machine learning*, il *deep*

learning impiega modelli che coinvolgono un maggior numero di composizioni, richiedendo pertanto un maggior numero di dati per addestrare la rete.

Nello specifico, il termine *deep learning* si riferisce a dei livelli multipli di composizione, ispirandosi all'attività cerebrale e neurale umana in cui più unità computazionali forniscono informazioni che interagiscono le une con le altre. L'idea principale del *deep learning* è di assumere che i dati siano stati generati da una composizione di fattori/caratteristiche, o *features*, che seguono dei livelli multipli gerarchici.

L'impiego del *deep learning* risponde all'esigenza di risolvere problemi difficili per l'uomo ma che possono risultare automatici per i computer, come riconoscere parole o facce in un'immagine: si impara dall'esperienza e si cerca di comprendere la realtà presente in termini di concetti gerarchici. Si apprendono così concetti complessi a partire da quelli più semplici: a tale scopo, la struttura (*network*) è composta da molti *layers*, risultando, quindi, "profonda".

L'input è un *layer* "visibile", ossia contenente variabili che si è in grado di osservare; successivamente si trovano dei *layers* "nascosti" (*hidden*): ciascuno di essi estrae le *features* dall'immagine di input, come viene rappresentato in Figura 2.1. I *deep neural networks*, infatti, per apprendere le complesse relazioni presenti tra il *layer* di input e quello di output impiegano modelli di tipo non lineare.

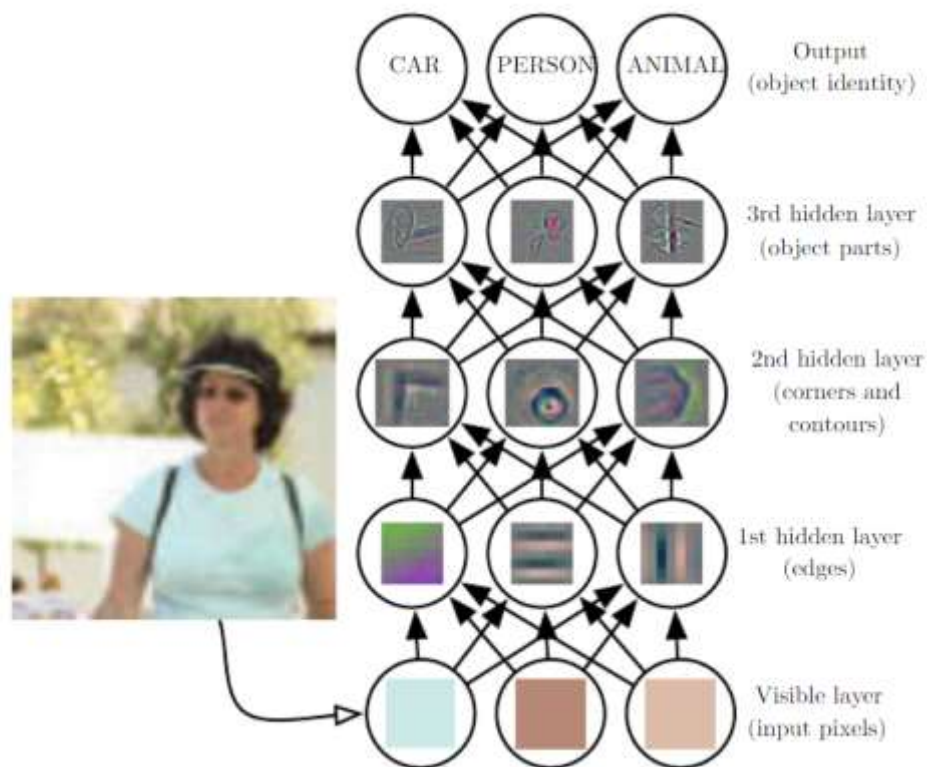
Non tutte le informazioni elaborate da un *layer* codificano fattori che spiegano l'oggetto in input; alcune sono solamente delle informazioni di stato che aiutano ad eseguire un programma utile per la rappresentazione di oggetti.

I concetti centrali che riguardano il *deep learning* sono:

- L'input di un sistema, il quale è rappresentato da *features*;
- L'impiego della retro-propagazione per addestrare i *networks* neurali;
- L'affidamento all'esperienza ed la correzione di eventuali errori.

Il *deep learning* ha così superato i limiti del *machine learning*, in cui le *features* venivano estratte manualmente: tramite il *network* composto da multipli *hidden layers* si è capaci di correggere tali caratteristiche in maniera automatica, *layer* dopo *layer*, eseguendo classificazione e predizione con un'elevata accuratezza e robustezza; le *features* sono quindi apprese dai dati non grazie all'operatore, ma al *network*.

Grazie alla loro struttura, le tecniche di *deep learning* consentono di eseguire classificazione, riconoscimento e segmentazione di oggetti, a partire da immagini, sequenze audio o testi, con applicazioni nel campo medico, socio-economico e politico.



Goodfellow, Ian. *Deep learning*. Massachusetts: Cambridge; 2016. 6 p.

Figura 2.1: Esempio di un modello di *deep learning*. Il primo *hidden layer* identifica i bordi, il secondo cerca angoli e contorni estesi, il terzo rileva parti intere di oggetti specifici, permettendo il riconoscimento di oggetti in un'immagine.

2.2 Artificial neural networks

Ciò che il *deep learning* si prefigge di imitare è l'attività neuronale del cervello umano, realizzando così dei networks neurali artificiali [16].

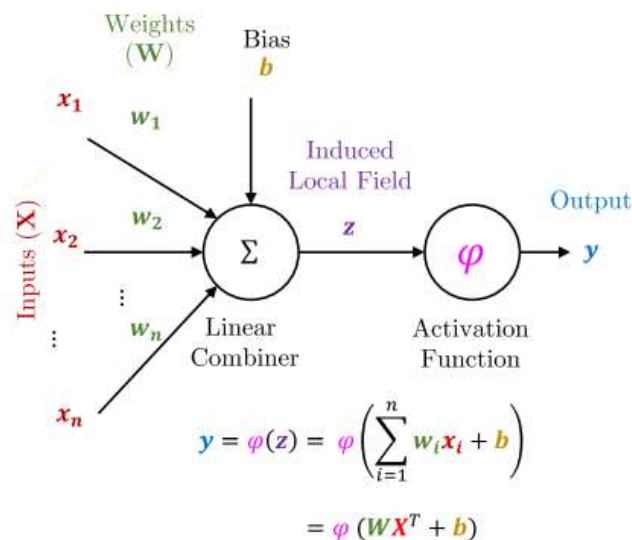
L'unità fondamentale di questi networks è il neurone artificiale, considerato come un singolo elemento decisionale. Esso può essere rappresentato graficamente come in Figura 2.2:

successivamente ad un vettore di input \mathbf{x} composto da n segnali, il cui contributo viene controllato da n rispettivi pesi (vettore \mathbf{w}), si aggiunge un termine di *bias* \mathbf{b} , inducendo in questo modo un campo locale z (il potenziale d'azione specifico per il neurone). Tale fenomeno può essere rappresentato dalla seguente formula:

$$z = \sum_{i=1}^m w_i x_i + b \quad (2.1)$$

Il compito di dimostrare l'attivazione del neurone viene svolto dalla funzione d'attivazione φ , che agisce quindi come una funzione di decisione, tale per cui $y = \varphi(z)$.

Per poter sostenere operazioni complesse, più neuroni sono portati a lavorare insieme, realizzando così una rete neurale di tipo *multi-layers*.



K.C. Santosh, Nibaran Das, Swarnendu Ghosh. Deep learning models for medical imaging. San Diego: Elsevier Science & Technology; 2021, 31 p. <https://doi.org/10.1016/C2020-0-00344-0>

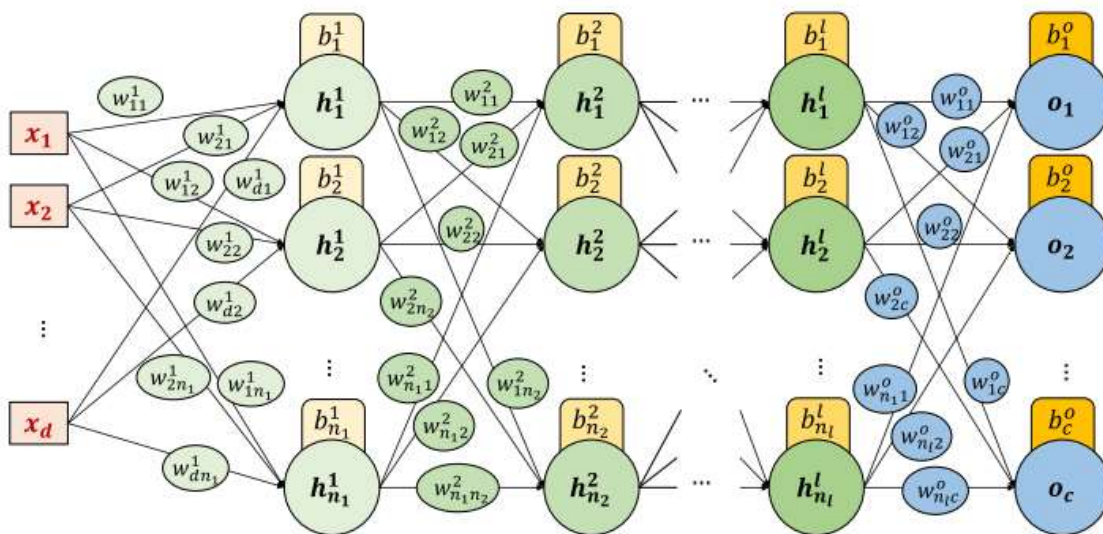
Figura 2.2: Modello matematico del neurone artificiale.

2.3 Deep Feed-forward networks

I *deep feed-forward networks* hanno come obiettivo comune quello di approssimare una funzione f^* : definiscono la mappatura $y = f(x; \theta)$ ed apprendono il valore dei parametri θ che approssimano al meglio la funzione. Si dicono *feed-forward* perché l'informazione viene trasferita lungo la rete, a partire dall'input x verso il *layer* di output finale e sono privi di retro-connessioni [15].

Questa struttura a catena è una delle più comuni impiegate a formare i networks neurali, come mostrato in Figura 2.3: tramite una successione di *layers* si riesce, a partire da un determinato input, a giungere all'output desiderato, il cui valore è noto (apprendimento supervisionato); infatti, ogni elemento di x è unito ad una *label* tale per cui $y \approx f^*(x)$.

Tuttavia, il comportamento degli altri *layers* (quelli posti tra il *layer* di input e quello di output) non è direttamente specificato dai dati di *training*, da cui il nome *hidden layers*, che comunque risultano composti da molte unità agenti in parallelo, con un comportamento simile a quello di un neurone.



K.C. Santosh, Nibaran Das, Swarnendu Ghosh. *Deep learning models for medical imaging*. San Diego: Elsevier Science & Technology; 2021. <https://doi.org/10.1016/C2020-0-00344-0>

Figura 2.3: Struttura di un *multilayer feed forward network*.

Per poter rappresentare funzioni non lineari dell'input \mathbf{x} è risultato necessario applicare ad esso una sua trasformazione, indicata dalla mappatura ϕ , che fornisce una serie di *features*. Inizialmente i valori dei parametri di questa funzione venivano corretti manualmente, mentre ora, grazie alle tecniche di *deep learning*, vengono appresi dalla rete.

Ne è un esempio il modello pari a:

$$y = f(\mathbf{x}; \boldsymbol{\theta}, \mathbf{w}) = \phi(\mathbf{x}; \boldsymbol{\theta})^T \mathbf{w} \quad (2.2)$$

Dove $\boldsymbol{\theta}$ sono i parametri che si impiegano per apprendere la mappatura ϕ e \mathbf{w} sono i pesi. L'obiettivo è trovare il miglior *set* di parametri $\boldsymbol{\theta}$ tali da ottenere l'output desiderato y .

Dunque, i *feed-forward networks* hanno introdotto il concetto di *hidden layer*, e questo richiede l'utilizzo di appropriate funzioni d'attivazione, necessarie per l'addestramento della rete.

2.4 Funzioni d'attivazione

La presenza della funzione d'attivazione $\varphi(z)$, in cui z si riferisce al campo locale indotto, consente di rendere il segnale di output diverso da una semplice funzione lineare, che quindi renderebbe l'azione del neurone alla pari di un modello di regressione lineare.

Vengono ora elencate le più importanti funzioni d'attivazione [16, 17]:

- funzione a gradino: è la funzione d'attivazione più semplice, definita dal seguente sistema di valori:

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

Essa fornisce solamente due possibilità di output perdendo buona parte dell'informazione; risulta quindi difficilmente addestrabile in un network neurale. Ne è riportato un esempio in Figura 2.4.

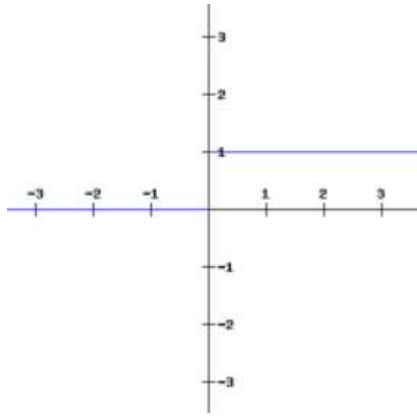


Figura 2.4: Funzione a gradino.

- funzione sigmoidea: è un esempio di funzione d'attivazione non lineare, definita dalla seguente espressione:

$$\varphi(z) = \sigma(z) = \frac{1}{1+e^{-z}} \quad (2.4)$$

Essa, a differenza della precedente funzione binaria a gradino, presenta in uscita anche i valori compresi nell'intervallo [0-1] (per valori di z prossimi all'origine), come si può notare in Figura 2.5 [17].

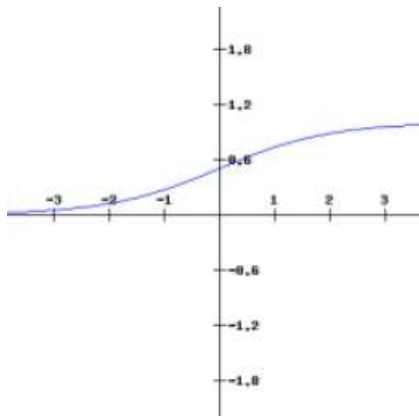


Figura 2.5: Funzione sigmoidea.

Tuttavia, l'impiego della funzione d'attivazione sigmoidea in un network con dimensioni piuttosto elevate può risultare difficile (vi si presentano problemi di vanificazione del gradiente, che saranno più chiari in seguito).

- Funzione ReLU (*rectified linear unit*): essa è capace di gestire in modo migliore, rispetto alla precedente funzione, il problema dovuto alle dimensioni elevate di una rete e viene descritta dalla formula seguente:

$$\text{relu}(z) = \max(0, z) \quad (2.5)$$

Come si può notare in Figura 2.6 [17], le attivazioni per valori di z negativi si annullano e questo diminuisce il carico computazionale del sistema complessivo presente.

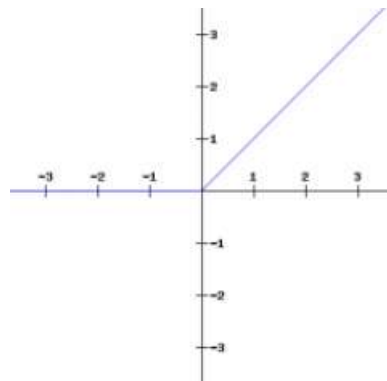


Figura 2.6: Funzione ReLU.

- Funzione Softmax: è una generalizzazione della funzione sigmoideale, e viene utilizzata per rappresentare la distribuzione di probabilità per variabili discrete aventi n possibili valori. È impiegata come ultima funzione d'attivazione di un classificatore multi-classe e, a tale proposito, verrà ripresa nel prossimo capitolo (paragrafo 2.8.3).

In particolare, per il neurone i -esimo e per j classi differenti vale [15]:

$$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (2.6)$$

2.5 Addestramento tramite back-propagation

La fase di addestramento dei *feed-forward neural networks* procede in due direzioni [15]:

- *Forward-propagation*: l'informazione si propaga a partire da un input \mathbf{x} per arrivare all'output predetto $\hat{\mathbf{y}}$. Confrontando l'output predetto con l'output desiderato è possibile calcolare una funzione costo che tenga conto dell'errore commesso ($E(w)$) nella predizione dell'output da parte della rete rispetto al valore desiderato.
- *Backward-propagation*: una volta calcolato l'errore e la relativa funzione costo, è possibile ottenere un valore per l'aggiornamento dei parametri della rete, con lo scopo di minimizzare l'errore. Questa operazione è resa possibile dal percorso opposto a quello della *forward-propagation* (dall'output si procede verso l'input, calcolando ad ogni step l'aggiornamento del peso in analisi).

La tecnica di calcolo del gradiente dell'errore rispetto ai pesi w della rete prende il nome di algoritmo di *back-propagation*, che viene spiegato in seguito [18].

Innanzitutto, si considera l'errore ($E(w)$) come la sommatoria N -esima dei contributi degli errori di ciascun dato presente in input, per cui vale:

$$E(w) = \sum_{n=1}^N E_n(w) \quad (2.7)$$

Si è visto precedentemente (2.1) come l'unità di un generico *feed-forward network* venga matematicamente espressa in funzione dei pesi, ciascuno relativo al segnale che giunge da un input i -esimo, e di un termine di *bias*. Di conseguenza, data la funzione d'attivazione $\varphi(\cdot)$ ed omettendo il termine di *bias*, è possibile esprimere il segnale di output complessivo di un'unità generica j come:

$$a_j = \sum_i w_{ji} z_i \quad (2.8)$$

$$z_j = \varphi(a_j) \quad (2.9)$$

Una volta ottenuti, per ogni elemento di input (in totale n elementi), i valori delle attivazioni di tutte le unità che appartengono ai *layers* di output e di *hidden* (*forward-propagation*), si può valutare il gradiente dell'errore E_n rispetto ai pesi w_{ji} presenti.

Applicando la regola delle derivate parziali, è possibile scomporre il valore del gradiente $\partial E_n / \partial w_{ji}$ con la modalità seguente:

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} \quad (2.10)$$

Assegnando poi il primo termine del secondo membro dell'equazione precedente ad un coefficiente δ_j e sfruttando la (2.8), è possibile scrivere l'espressione del gradiente dell'errore rispetto ai pesi in funzione delle unità di input ed output tramite:

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i \quad (2.11)$$

Quindi, dal momento che nel *feed-forward network* ci sono una successione di connessioni, le quali a partire da un input ed attraverso gli *hidden layers* presenti giungono all'output, per ottenere il valore del gradiente complessivo è sufficiente calcolare il coefficiente δ per ognuna delle unità di output e degli *hidden layers*, e poi applicare la (2.11).

Per quanto riguarda il coefficiente δ applicato alla k -esima unità di output, esso è pari alla differenza tra il valore di output predetto (\hat{y}) e quello desiderato (y), dunque pari a:

$$\delta_k = \hat{y}_k - y_k \quad (2.12)$$

Successivamente, si calcola il coefficiente δ riferito all'unità di *hidden layer* j -esima sulla base della successione di connessioni che essa, durante la *forward-propagation*, ha stabilito con la k -esima

unità di output o di *hidden* (com'è indicato in Figura 2.7); anche qui la scomposizione con le derivate parziali risulta utile:

$$\delta_j = \sum_k \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial a_j} \quad (2.13)$$

Ora, se si sostituisce a $\frac{\partial E_n}{\partial a_k}$ il coefficiente δ_k e utilizzando le definizioni (2.9) e (2.10), si ottiene la formula di *back-propagation*, secondo la quale:

$$\delta_j = \varphi'(a_j) \sum_k w_{kj} \delta_k \quad (2.14)$$

Questa formula suggerisce che, una volta noti il valore delle attivazioni delle unità di output e di *hidden* (*forward-propagation*) e dunque il valore di δ_k , il valore di δ_j per ognuna delle unità di *hidden* che si ottiene è dato dalla retro-propagazione dell'errore. Sostituendo poi la (2.14) nella (2.11) si ottiene il gradiente dell'errore rispetto ai pesi richiesto.

Quindi, risulta importante per l'algoritmo di retro-propagazione del gradiente, che tutte le funzioni utilizzate siano derivabili. Un fattore che potrebbe influenzare la catena delle derivate necessarie alla computazione del gradiente è la profondità della rete: se molto profonda, un prodotto eccessivamente lungo potrebbe causare problemi di vanificazione del gradiente, come accennato nel paragrafo precedente.

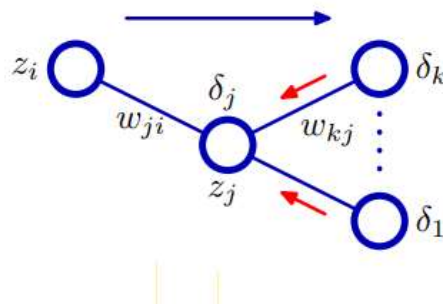


Figura 2.7: Retro-propagazione dell'errore per l'unità j -esima di *hidden* (freccia rossa) a partire dalla k -esima unità di output (o di *hidden*) ad essa connessa. La freccia blu indica la direzione della *forward-propagation*.

2.5.1 Stochastic Gradient Descent with Momentum (SGDM)

Una volta ottenuta l'informazione fornita dal gradiente tramite l'algoritmo di *back-propagation*, spostandosi nella direzione opposta si riduce al massimo il valore dell'errore ($E(w)$) ed i pesi vengono così aggiornati (*gradient descent*). Dati i parametri w (pesi), E (errore) ed η (*learning rate*), è possibile rappresentare quanto detto con la seguente espressione [18]:

$$w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E(w^{(\tau)}) \quad (2.15)$$

Il *learning rate* ($\eta > 0$) è un parametro molto importante per l'addestramento dei networks neurali, poiché identifica la velocità con la quale i pesi vengono aggiornati; un valore troppo basso causa lentezza nel raggiungimento del minimo globale della funzione d'errore o potrebbe portare al raggiungimento di un minimo locale; mentre un valore troppo alto potrebbe non raggiungerlo mai. Un esempio dell'andamento del *learning rate* rispetto alla funzione d'errore, o *loss function*, è mostrato in Figura 2.8.

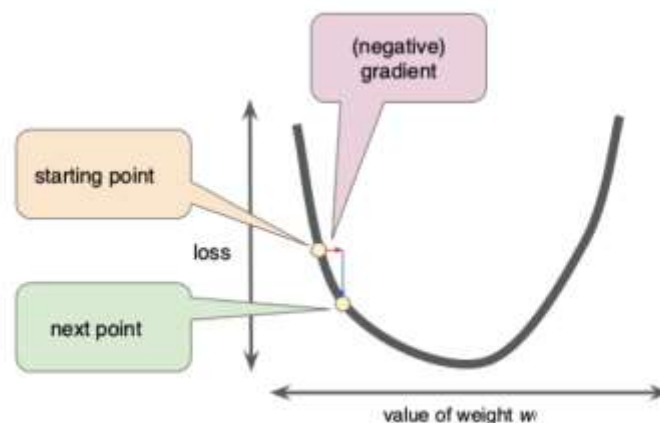


Figura 2.8: Andamento del *learning rate* rispetto alla *loss function*.

La funzione d'errore solitamente viene definita sulla base del dataset di *training*, ossia quell'insieme di dati che vengono utilizzati per l'addestramento del network neurale; osservando la formula (2.15), questo significa che ad ogni step τ , per poter ottenere il valore del gradiente dell'errore bisogna elaborare l'intero *training set*. Le tecniche che per aggiornare il valore dei pesi impiegano, di volta in volta, l'intero dataset iniziale sono indicate con il termine *batch*, risultando onerose se il dataset è molto grande. Una versione differente, che prende il nome di *stochastic gradient descent* (SGD), supera questo problema: i pesi vengono infatti aggiornati basandosi su un sottoinsieme di dati di *training* (*mini-batch*), che viene così elaborato ad ogni iterazione. Quando l'intero dataset di *training* è stato processato, si è raggiunta un'epoca. Oltre ad essere particolarmente adatto per dataset di ampie dimensioni, il metodo SGD risulta migliore rispetto a quelli *batch* nel sostenere il problema della ridondanza dei dati presenti in input.

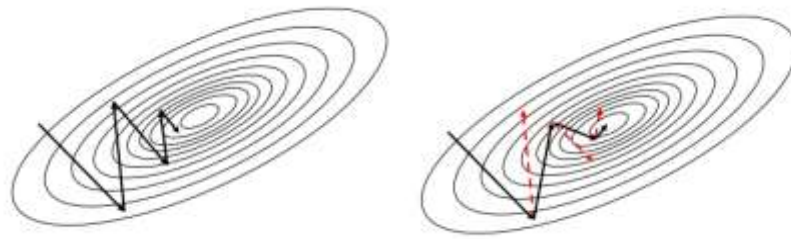
L'andamento del gradiente ha un ruolo molto importante nell'efficienza del metodo SGD; è per questo che a tale algoritmo è stato introdotto il concetto di momento [16]. Esso prende in considerazione il fatto che il gradiente di una certa iterazione aumenta la pendenza di quello presente all'iterazione successiva (durante la discesa del gradiente), forzando così i pesi a muoversi più velocemente lungo la pendenza negativa del gradiente, com'è evidente in Figura 2.9.

Con la precedente osservazione, l'aggiornamento dei pesi risulta pari a:

$$\mathbf{v} = \alpha \mathbf{v} - \eta \frac{\partial E}{\partial \mathbf{w}} \quad (2.16)$$

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^\tau + \mathbf{v} \quad (2.17)$$

Il coefficiente α controlla l'impatto che il gradiente precedente ha sull'andamento del gradiente successivo, il quale, una volta aggiornato, è indicato con il coefficiente \mathbf{v} .



K.C. Santosh, Nibaran Das, Swarnendu Ghosh. *Deep learning models for medical imaging*. San Diego: Elsevier Science & Technology; 2021. 42 p. <https://doi.org/10.1016/C2020-0-00344-0>

Figura 2.9: Effetto del momento (freccie nere) sulla direzione di aggiornamento dei pesi (freccie rosse).

2.6 Loss functions

Come osservato in precedenza, la *loss function* viene impiegata per l'addestramento dei networks neurali: per poter aggiornare il valore dei pesi di una rete nella fase di retro-propagazione, si vuole raggiungere il minimo assoluto della funzione costo, spostandosi nella direzione opposta alla crescita del gradiente.

Precedentemente, negli algoritmi di *machine learning* si era soliti utilizzare come funzione costo la funzione *mean squared error* (MSE): essa è data dalla media della sommatoria del quadrato degli errori ottenuti dalla differenza tra il valore di output predetto \hat{y} e quello desiderato y , dove n sono gli elementi presenti nel *layer* di output.

La formula è la seguente [16]:

$$E_{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (2.18)$$

Tuttavia, se si calcola il valore del gradiente della *mean squared error* rispetto ai pesi $\left(\frac{\partial E}{\partial w}\right)$ si ottiene un valore indipendente dal parametro y (output desiderato), e questo rende la MSE inadatta per problemi di classificazione, causando una riduzione in termini di accuratezza ed efficienza. Nelle

tecniche di *deep learning*, infatti, la maggior parte delle funzioni d'attivazione applicate nell'ultimo *layer* sono la sigmoidea e la Softmax, in base ai *task* che vengono eseguiti [19].

Scegliere un *loss function* adeguata è dunque cruciale per l'apprendimento di una rete neurale e dipende dal compito richiesto; viene descritta in seguito la funzione costo *Cross-Entropy*, ampiamente utilizzata sia nel caso della classificazione sia in quello della segmentazione semantica (ossia una classificazione a livello dei pixel presenti in un'immagine).

Nel capitolo successivo, verrà ripreso il concetto di *loss functions* nel caso specifico del problema della segmentazione semantica delle cellule dendritiche corneali, apportando delle correzioni alla definizione classica di *Cross-Entropy* e presentando un'ulteriore funzione costo.

2.6.1 Cross-Entropy

In caso di classificazione binaria (BCE, cioè *binary cross entropy*) e richiamando i valori definiti precedentemente di output predetto \hat{y} e desiderato y , la funzione costo definita dalla *Cross-Entropy* risulta pari a [20]:

$$E_{BCE}(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \quad (2.19)$$

Rispetto alla MSE, se si calcola il gradiente della funzione costo rispetto ai pesi, ciò che si ottiene è un valore che dipende dalla differenza tra y ed \hat{y} , rendendo, per questo motivo, tale funzione molto efficace per i problemi di classificazione. Nel caso in cui siano presenti più di due classi (*categorical cross entropy*-CCE), la formula precedente viene così modificata [16]:

$$E_{CCE} = - \sum_{i=1}^C y_i \log \hat{y}_i \quad (2.20)$$

Dove C indica il numero di classi.

2.7 Regolarizzazione

Con regolarizzazione si intende qualsiasi metodo che migliori la capacità di generalizzazione di un network, ossia l'abilità di apprendimento e di riconoscimento di *features* in presenza di dati non ancora analizzati o visti. È infatti possibile distinguere i dati di *training*, con i quali si addestra la rete (*training set*), ed i dati di *test*, con i quali si osserva la performance della rete (*test set*). Condurre regolarizzazione significa quindi ridurre l'errore presente sui dati di test, eventualmente a spese dell'errore dei dati di *training* [15,16].

2.7.1 Overfitting

La rete neurale, come osservato finora, viene dunque addestrata a partire dal *training set*, ed i pesi vengono di volta in volta aggiornati per minimizzare la funzione costo impiegata. Le prestazioni della rete vengono poi valutate sul *validation set*, composto da elementi differenti rispetto a quelli impiegati per addestrare la rete; in questo modo, si modificano alcuni parametri del network (iper-parametri) per perfezionare l'abilità di apprendimento della rete. Gli iper-parametri possono essere, ad esempio, il numero di neuroni, *layers*, iterazioni e la tipologia delle funzioni d'attivazione [16]. *Validation set* e *test set* sono differenti perché il primo viene impiegato per validare i parametri della rete, mentre il secondo viene utilizzato per osservarne la performance finale; entrambi, però, sono costituiti da elementi che non vengono utilizzati per addestrare la rete.

La condizione in cui si verifica un miglioramento delle prestazioni della rete nel *training set* rispetto a quelle nel *validation set* prende il nome di *overfitting*, ed è rappresentato in Figura 2.10.

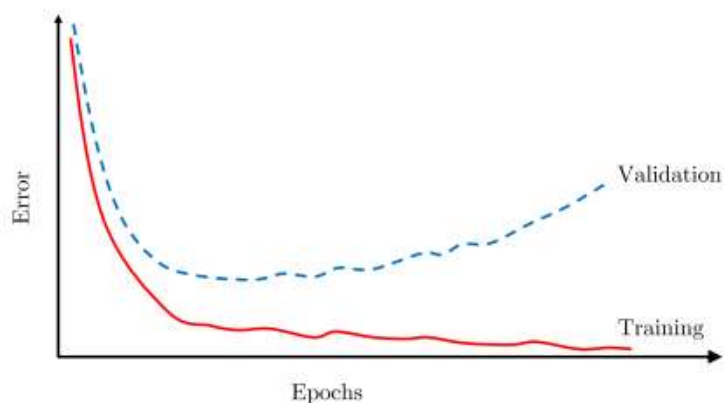


Figura 2.10: Overfitting.

L'errore di generalizzazione del network è dato dalla differenza tra la performance di *training* e quella di *validation*, e l'obiettivo della procedura di regolarizzazione è dunque ridurne il valore.

2.7.2 Tecniche di regolarizzazione

Solitamente, nelle tecniche di regolarizzazione (impiegate anche negli algoritmi di *machine learning*), si modifica la funzione costo aggiungendo dei termini, che spesso si presentano come costanti. L'aggiunta di questi termini cerca di prevenire che l'andamento dell'errore della funzione costo, calcolata nella fase di *training*, sia distante dal punto di minimo assoluto. Questo potrebbe aumentare la bontà di apprendimento della rete nello specifico del dataset di *train*, ma non su dati differenti che la rete non ha ancora osservato (scarsa capacità di generalizzazione e condizione di *overfitting*) [16].

Vengono ora presentate le tecniche di regolarizzazione più comuni.

2.7.2.1 Weight decay

È un approccio che aggiunge una penalità ($\Omega(\mathbf{w})$) alla funzione costo E con la modalità seguente [15]:

$$E_R(\mathbf{w}, \mathbf{X}, \mathbf{y}) = E(\mathbf{w}, \mathbf{X}, \mathbf{y}) + \alpha\Omega(\mathbf{w}) \quad (2.21)$$

Dove α (positivo) è un iper-parametro che rappresenta il peso da attribuire alla penalità Ω da dare alla rete addestrata (più è grande, più c'è regolarizzazione), \mathbf{X} rappresenta i dati di *training* mentre \mathbf{y} i valori di output desiderati.

Il parametro Ω può assumere diverse tipologie. Da notare che, per i networks neurali, si penalizzano solamente i pesi e si lasciano invariati i *bias*.

2.7.2.1.1 L^2 Regularization

Essa prende il nome anche di *ridge regression* ed è definita da [15, 16]:

$$E_{L2}(\mathbf{w}, \mathbf{X}, \mathbf{y}) = E(\mathbf{w}, \mathbf{X}, \mathbf{y}) + \lambda \sum_i w_i^2 \quad (2.22)$$

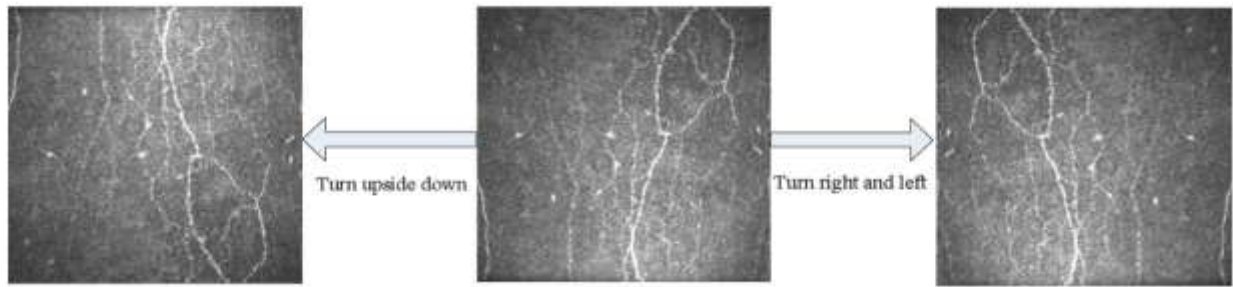
Dove λ è un coefficiente che controlla il peso da attribuire alla regolarizzazione.

2.7.2.2 Data Augmentation

Il *deep learning* differisce dal *machine learning* per la grande mole di dati richiesti per addestrare la rete, quindi un'ulteriore tecnica per migliorare la generalizzazione è quella di amplificare il dataset a disposizione, che spesso non presenta dimensioni sufficienti.

In generale, la procedura di *data augmentation* è ottenuta alterando, in maniera controllata, le condizioni dei campioni in input, aggiungendoli poi al *training set*, tramite: *flip* orizzontale o verticale, rotazione angolare, variazioni di luminosità, aggiunta di rumore [16, 21].

Ciò nonostante, è di rilevante importanza che le operazioni eseguite siano accettabili, rispettando le naturali caratteristiche e circostanze di ciò che si vuole rappresentare, specialmente nel campo medico. Esso richiede infatti che le operazioni di *data augmentation* preservino integrità e autenticità delle immagini. In Figura 2.11, è mostrato un esempio di operazioni di *data augmentation* accettabili per le immagini acquisite tramite microscopia confocale corneale (relative ad un soggetto affetto da *cheratite fungina*), quali il flip orizzontale e verticale.



Zhi Liu, Yankun Cao, Yujun Li, Xiaoyan Xiao, Qingchen Qiu, Meijun Yang et al. *Automatic diagnosis of fungal keratitis using data augmentation and image fusion with deep convolutional neural network*. Computer Methods and Programs in Biomedicine; 2020. 3 p. <https://doi.org/10.1016/j.cmpb.2019.105019>.

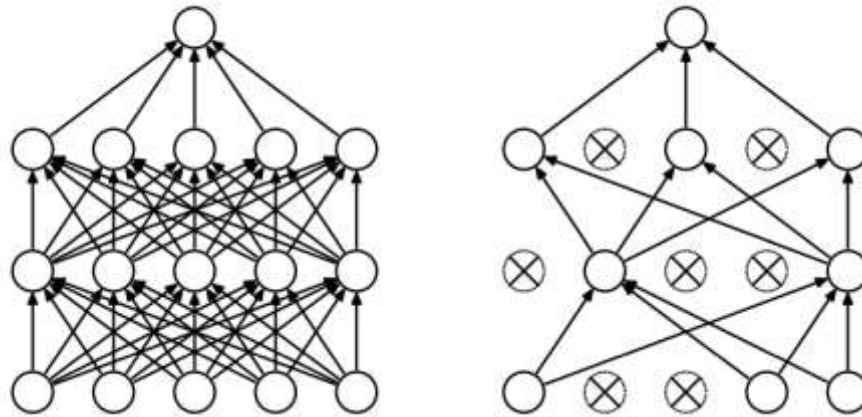
Figura 2.11: *Data augmentation* per immagini acquisite tramite CCM: a sinistra, flip verticale, al centro immagine originale, a destra flip orizzontale.

2.7.2.3 Dropout

Il termine *dropout* richiama il concetto di escludere in maniera casuale con probabilità $1-p$ alcune unità di *hidden* e di input, con le relative connessioni in ingresso e in uscita, moltiplicandone l'output per un valore pari a 0, com'è mostrato in Figura 2.12. Il valore della probabilità p viene solitamente fissato al 50%, oppure può essere scelto utilizzando il *validation set*; per le unità di input è invece prossimo a 1 (minore probabilità di essere escluse dalla rete).

Ogni qual volta si esegue l'addestramento della rete, dunque, il network si presenta con una configurazione differente rispetto alla precedente, permettendo ai singoli neuroni del network di avere una buona *performance* anche in assenza dei loro "vicini", riducendone l'adattamento e migliorando la loro capacità di apprendimento e generalizzazione [16, 22].

In fase di *test*, viene utilizzato l'intero network modificando il valore di alcuni pesi: se, durante il *training* della rete, una generica unità è stata esclusa con probabilità $1-p$, i pesi relativi ai suoi segnali in uscita vengono moltiplicati per p , assicurando che il contributo dell'output dei neuroni di *hidden* esclusi in fase di addestramento sia lo stesso di quello presente in fase di *test*.



Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, Journal of Machine Learning Research; 2014. 15, 56, 1930 p. <http://jmlr.org/papers/v15/srivastava14a.html>

Figura 2.12: *Dropout* di un network neurale. A sinistra è rappresentata la rete in condizioni iniziali, mentre a destra il risultato del *dropout* (le croci indicano le unità che sono state escluse).

2.7.2.4 Batch Normalization

In fase di addestramento, gli input di ciascun *layer* risentono dei parametri presenti nei *layers* precedenti: se la distribuzione dei nodi interni al network viene modificata, la rete deve adattarsi alla nuova configurazione. Questi cambiamenti causano un fenomeno nominato *internal covariate shift*; poterli eliminare consente di velocizzare il processo di addestramento [23].

Il meccanismo di *Batch Normalization* consente di ridurre l'entità del fenomeno precedente facendo uso di uno step di normalizzazione: vengono fissati i valori di media e deviazione standard dei *layers* di input e si regolarizza la rete (per questo può essere considerata come una tecnica di regolarizzazione). È così possibile ridurre l'utilizzo di *dropout*, permettendo l'impiego di valori di *learning rate* più elevati.

Considerando un *layer* di input con dimensione pari a d ($x = x^{(1)} \dots x^{(d)}$), si normalizza la dimensione dell'input x nella maniera seguente, in cui i valori di E (aspettazione) e varianza sono calcolati nell'intero *training set*:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}} \quad (2.23)$$

Questa normalizzazione dev'essere però corretta assicurandosi che la trasformazione apportata al network rappresenti l'identità (una semplice normalizzazione potrebbe modificare ciò che il *layer* rappresenta). Vengono così introdotti due ulteriori parametri $\gamma^{(k)}$ e $\beta^{(k)}$, che scalano e spostano il valore normalizzato $\hat{x}^{(k)}$, nella seguente maniera:

$$y^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)} \quad (2.24)$$

Tuttavia, nel caso di algoritmi come lo *stochastic gradient descent with momentum*, che utilizzano, di volta in volta, un *mini-batch* per addestrare la rete (non l'intero *training set*), bisogna apporre un'ulteriore modifica alla formulazione precedente.

Si può dunque considerare che ogni *mini-batch*, in fase di addestramento, produce le stime di media e varianza di ogni attivazione, ed impiega questi valori per la fase di retro-propagazione del gradiente.

Considerando un *mini-batch* di dimensione pari a m , i valori precedentemente formulati vengono modificati secondo:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (2.25)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (2.26)$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (2.27)$$

$$y_i = \gamma \hat{x}_i + \beta = BN_{\gamma, \beta}(x_i) \quad (2.28)$$

Dove i parametri γ e β vengono appresi dalla rete, ed il termine BN indica la trasformazione di *Batch Normalization* inerente all'attivazione i -esima di input applicata su un *mini-batch*.

L'obiettivo, come indicato precedentemente, è fissare i valori di media e varianza degli input di ciascun *mini-batch*, accelerando l'addestramento del network e permettendo, attraverso l'introduzione dei parametri γ e β , di preservare la capacità della rete anche a seguito della procedura di normalizzazione delle attivazioni x .

2.8 CNNs

I CNNs (*convolutional neural networks*) sono una tipologia specifica di *deep neural network* per il *processing* dei dati che presentano una struttura a forma di griglia. Ne sono un esempio i dati di serie-temporali, che possono essere pensati come una griglia monodimensionale con campioni ad intervalli di tempo regolari, ed i dati di immagini, che sono rappresentati come una griglia bidimensionale di pixel.

Il termine “convoluzionale” deriva dal fatto che il network impiega l'operazione matematica lineare di convoluzione: in almeno uno dei *layers* viene utilizzata la convoluzione al posto della tradizionale moltiplicazione matriciale: considerando come input un'immagine I bidimensionale, e K il *kernel*, la convoluzione discreta tra i due elementi risulta pari a [15]:

$$S(i,j) = \sum_m \sum_n I(m,n) K(i-m,j-n) \quad (2.29)$$

La convoluzione riguarda tre concetti importanti, che possono migliorare un sistema di *machine learning*: l'interazione sparsa, la condivisione dei parametri e la rappresentazione equivariante. Grazie a tali aspetti, il metodo si presenta come uno strumento adatto per lavorare con un input di dimensione variabile. La prima caratteristica, ovvero l'interazione sparsa, si riferisce al fatto che la dimensione del *kernel* è relativamente piccola rispetto all'immagine intera: non c'è un'interazione esatta di ogni unità di output con ognuna dell'input, bensì solo una parte di esso, come potrebbero essere i bordi di un'immagine. In questo modo, si aumenta l'efficienza e si riduce la memoria impiegata. La seconda caratteristica (condivisione dei parametri) si riferisce all'utilizzo dello stesso parametro in più di una funzione del modello: dal momento che il valore di un parametro può essere

utile per l'analisi dei *layers* successivi, esso viene visitato almeno una volta, non una soltanto. Per quanto riguarda la terza caratteristica, ovvero la rappresentazione equivariante, essa sottolinea il concetto per il quale se l'input cambia, l'output cambia nella stessa maniera, in modo tale che, date $f(x)$ e g , valga $f(g(x)) = g(f(x))$. La convoluzione non è però equivariante a trasformazioni quali cambiamenti di scala o rotazioni dell'immagine, per i quali sono necessari altri meccanismi. Ciò nonostante, la convoluzione rende possibili dei processi di dati non consentiti dai networks neurali tradizionali.

I CNNs sono i networks di tipo supervisionato (c'è un output predefinito) più comunemente usati nella classificazione e segmentazione d'immagini: tramite regolarizzazione e retro-propagazione, si addestra il network in modo tale che la differenza tra il valore di output ottenuto e quello desiderato sia minima. Generalmente, non si usano solamente operazioni lineari per passare dal *layer* di input a quello di output; prima di un'eventuale operazione non lineare, viene aggiunto un termine di *bias*.

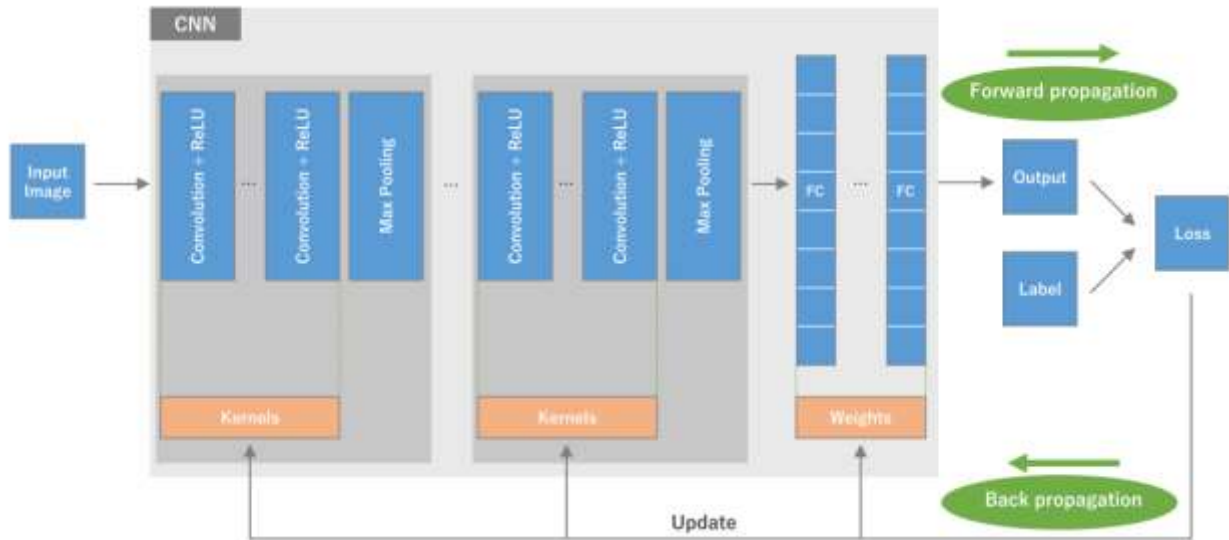
La convoluzione che si applica nel contesto dei networks neurali non è esattamente pari alla comune definizione matematica, ci sono alcune differenze. Prima di tutto, la convoluzione impiegata nel *deep learning* fa riferimento a molte convoluzioni in parallelo; successivamente, l'input non è una griglia di valori reali, spesso lo è di vettori: ad esempio, nel caso delle immagini, l'input e l'output sono dei tensori tridimensionali. Inoltre, la proprietà di commutatività viene persa nel momento in cui i networks convoluzionali sono di tipo multicanale, viene invece preservata se il numero di canali in input è lo stesso dell'output.

I CNNs sono realizzati da tre tipologie di *layers*, o blocchi: convoluzione, *pooling* e *fully connected layer* (*layer* completamente connesso); i primi due eseguono l'estrazione delle *features*, mentre il terzo mappa le *features* appena estratte sull'output finale. Solitamente vi è una ripetizione dei *layers* di convoluzione e di *pooling*, seguiti poi da uno o più *layers* completamente connessi, com'è rappresentato in Figura 2.13. La *performance* della rete viene valutata aggiornando i valori dei *kernels* e dei pesi durante la fase di retro-propagazione, minimizzando una funzione costo [24].

2.8.1 Layer di convoluzione

Il *layer* di convoluzione rappresenta un elemento fondamentale per i CNNs dal momento che esegue l'estrazione delle *features* [24]. Durante l'operazione di convoluzione e considerando come input un'immagine bidimensionale, viene eseguito il prodotto elemento per elemento tra il *kernel*

utilizzato (di dimensioni minori rispetto all'immagine iniziale) ed i rispettivi elementi presenti in input, per poi sommarne i valori: il risultato ottenuto viene salvato in una *feature map*.



Yamashita R., Nishio M., Do R.K.G., Togashi K. *Convolutional neural networks: an overview and application in radiology*. Insights Imaging 9; 2018. 613 p. <https://doi.org/10.1007/s13244-018-0639-9>

Figura 2.13: Architettura di un CNNs.

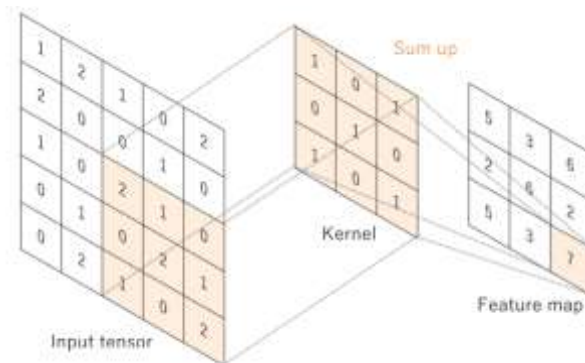
Questa procedura viene ripetuta spostando il *kernel* lungo l'immagine iniziale, e salvando di volta in volta il valore ottenuto nella *feature map*, com'è rappresentato in Figura 2.14.

Più *kernels* possono essere applicati e di conseguenza possono essere presenti più *feature maps*. Spesso, all'operazione di convoluzione appena descritta si associa talvolta quella di *zero-padding*: essa permette di aggiungere delle righe e/o colonne di zeri all'estremità dell'immagine iniziale. In particolare, la tecnica di *zero-padding* viene eseguita per prevenire che, per determinate dimensioni del *kernel*, non ci sia corrispondenza tra gli elementi del *kernel* situati all'estremità dell'immagine iniziale ed il tensore di input.

Solitamente, le dimensioni del *kernel* utilizzate sono pari a 3x3, 5x5 o 7x7, ed il passo (*stride*) tra due posizioni successive è pari a 1; questi valori, uniti al numero dei *kernels* impiegati ed alla tipologia del *padding*, devono essere determinati prima di addestrare la rete.

Una volta che si ottiene la *feature map*, vengono trasferiti i suoi valori ad una funzione d'attivazione non lineare; la più utilizzata è la ReLU (paragrafo 2.4).

L'utilizzo del *kernel* durante l'operazione di convoluzione consente l'estrazione delle *features* appartenenti al tensore di input. Di conseguenza, utilizzare più *kernels* permette di estrarre più caratteristiche (ad esempio bordi orizzontali o verticali nel caso di un'immagine).



Yamashita R., Nishio M., Do R.K.G., Togashi K. *Convolutional neural networks: an overview and application in radiology*. Insights Imaging 9; 2018. 614 p. <https://doi.org/10.1007/s13244-018-0639-9>

Figura 2.14: Esempio di operazione di convoluzione. Il *kernel* viene applicato sul tensore di input moltiplicando i rispettivi elementi costituenti e sommandone i valori; il risultato viene poi salvato in una determinata posizione della *feature map*.

2.8.2 Layer di pooling

Il *layer* di *pooling* esegue un'operazione di sotto-campionamento a partire dal risultato ottenuto nel precedente *layer* (il più delle volte corrisponde ad un *layer* di convoluzione), permettendo di migliorare la capacità della rete ad apprendere *features* più complesse [16, 24]. La dimensione della *feature map* viene dunque ridotta e questo conferisce la proprietà di invarianza alle traslazioni (relative a piccoli spostamenti) ed alle distorsioni presenti.

L'operazione di *pooling* può essere di vario tipo: le più comuni sono il *max-pooling* o l'*average-pooling*, e vengono di seguito spiegate.

2.8.2.1 Max-pooling

L'operazione di *max-pooling* estrae il massimo valore presente a partire da un sotto-gruppo della *feature map* definito dalla dimensione del parametro *filter size*. Comunemente, quest'ultimo è pari a

2x2 e, accompagnato ad un valore di stride pari a 2, permette di eseguire un sotto-campionamento di un fattore pari a 2. Un esempio della procedura di *max-pooling* è indicato in Figura 2.15.

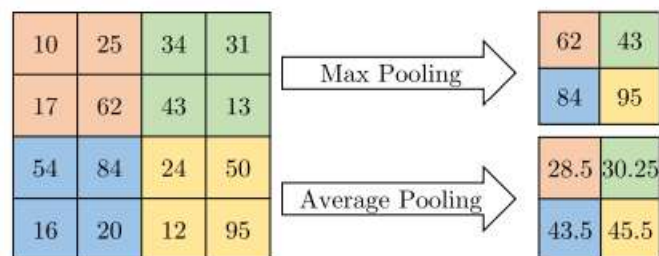
Esso viene comunemente utilizzato nei *layers* intermedi, in cui l'informazione fornita dalle attivazioni con valore massimo domina sulle altre.

Come per l'operazione di convoluzione, i valori di *filter-size*, *stride* e la tipologia di *padding* (se utilizzato) sono iper-parametri che devono essere definiti poiché non vengono appresi dalla rete.

2.8.2.2 Average-pooling

L'operazione di *average-pooling* estrae la media degli elementi all'interno del sotto-gruppo della *feature map*, definito anche qui dal parametro *filter size*. Essa viene utilizzata nel momento in cui le attivazioni ottenute dal valore massimo non risultano significative.

Ne è mostrato un esempio in Figura 2.15.



K.C. Santosh, Nibaran Das, Swarnendu Ghosh. *Deep learning models for medical imaging*. San Diego: Elsevier Science & Technology; 2021. 53 p. <https://doi.org/10.1016/C2020-0-00344-0>

Figura 2.15: *Max-pooling vs average-pooling.*

2.8.3 Fully connected layer

Il *fully connected layer*, presente successivamente all'ultimo *layer* di *pooling* del CNNs, come indicato in Figura 2.13 con l'acronimo FC, è un *array* mono-dimensionale di numeri, e può essere connesso ad uno o più *fully connected layers* (*dense layers*), in cui ogni input è connesso ad ogni output mediante dei pesi che vengono appresi dalla rete. La lunghezza del *fully connected layer* è

pari al numero di classi; ogni *layer fully connected* è seguito da una funzione d'attivazione non lineare, come quelle descritte al paragrafo 2.4 (ReLU, sigmoideale e *softmax*) [24].

Tuttavia, la funzione d'attivazione applicata all'ultimo *layer* è differente rispetto a quelle applicate precedentemente perché varia in base al problema. Nel caso, ad esempio, della classificazione con un numero di classi superiore a due, la funzione d'attivazione che si utilizza è la *softmax*, che normalizza i valori reali in uscita dall'ultimo FC in probabilità delle classi, in modo che la somma degli elementi ottenuti sia pari ad 1, e ciascun valore sia compreso tra 0 e 1, come mostrato in Figura 2.16. Se invece si tratta di una classificazione binaria, può essere sufficiente l'utilizzo della funzione d'attivazione sigmoideale.

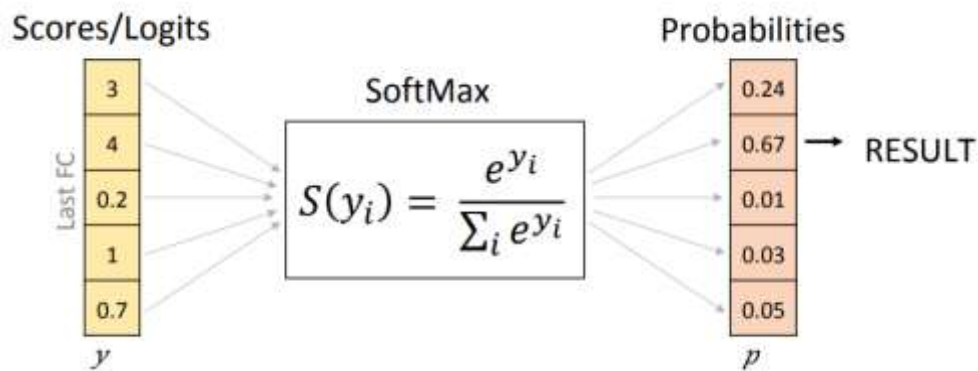


Figura 2.16: Applicazione della funzione d'attivazione *softmax* per un problema classificazione di tipo multi-classe.

2.9 Segmentazione

La segmentazione è un aspetto molto importante nel campo della *computer vision* e risponde al problema di suddividere un'immagine o dei *video frames* in oggetti d'interesse. Viene applicata per: l'analisi di immagini mediche (estrazione di componenti, misurazione del volume dei tessuti), la guida autonoma dei veicoli (rilevamento di pedoni, strade, ecc.), la sorveglianza tramite video ed il campo della realtà aumentata [16, 25, 26].

Essa risponde al problema di classificare i pixel di un'immagine tramite *labels* (segmentazione semantica), o di suddividere l'immagine in ogni oggetto d'interesse presente (*instance segmentation*). La segmentazione semantica si occupa di etichettare ciascun pixel dell'immagine in

categorie (es. cellula sana o malata). Dapprima, per eseguire la segmentazione di un'immagine, si applicavano metodi quali *thresholding*, *region-growing*, *active contours*; successivamente, si sono utilizzate le tecniche di *deep learning* che hanno migliorato l'accuratezza.

Infatti, il successo dei *convolutional neural networks* (paragrafo 2.8) nel classificare oggetti ha suggerito il loro utilizzo anche nella segmentazione. Tuttavia, la presenza del *fully connected layer* (paragrafo 2.8.3) per la classificazione finale non consente di preservare l'informazione spaziale codificata nelle *feature maps*, e l'utilizzo dei *layers* di *pooling* (paragrafo 2.8.2) riduce la nitidezza del risultato di segmentazione in uscita. Per risolvere questo problema, sono stati dunque proposti dei nuovi algoritmi, come quelli appartenenti al gruppo dei *DeepLab*, che hanno introdotto ai *layers* di convoluzione un ulteriore parametro, il *dilation rate*: per esempio, un *kernel* 3x3 con *dilation rate* pari a 2 recepisce le stesse informazioni di un *kernel* con dimensione 5x5, senza aumentare il costo computazionale.

Un'ulteriore architettura si è rivelata adatta ai problemi di segmentazione semantica delle immagini, ossia quella di tipo *encoder-decoder*, che viene di seguito approfondita.

2.9.1 Encoder-decoder

I modelli di tipo *encoder-decoder* generalmente mappano determinati dati da un input ad un output, mediante un network costituito da due fasi:

- la parte di codifica (*encoder*) con la relativa funzione $z = g(x)$, che comprime l'input x in una sua rappresentazione z ;
- la parte di decodifica (*decoder*), che permette di ottenere il valore di output finale predetto $\hat{y} = f(z)$.

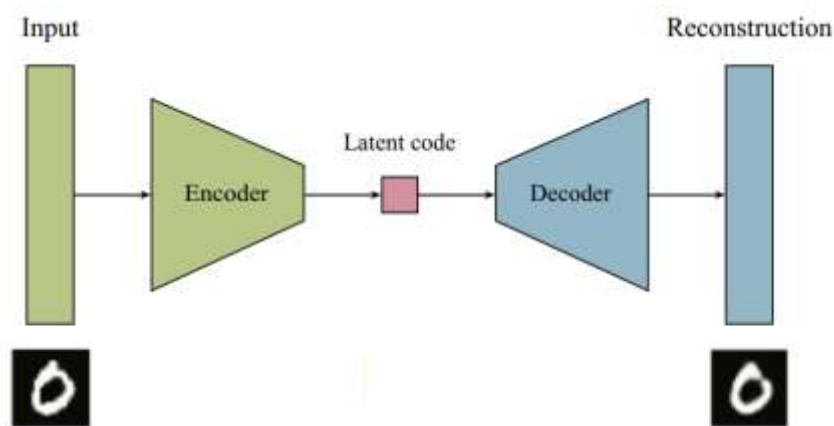
Tra la parte di codifica e quella di decodifica, viene considerata l'informazione utile al risultato di segmentazione finale e prende il nome di rappresentazione latente. Nel caso in cui l'output che viene ricostruito sia lo stesso dell'input, si parla di *autoencoder*, rappresentato in Figura 2.17.

Il dato presente in input (x) può essere un'immagine (2D o 3D), una sequenza audio/video o un testo, e la struttura delle architetture di tipo *encoder-decoder* è versatile: può essere più o meno profonda, in base alla complessità della *features* che vengono estratte [25, 27].

L'obiettivo degli *autoencoders* è comunque quello di ricostruire l'output nella maniera più accurata possibile, minimizzando la funzione costo durante la fase di addestramento della rete, ossia la differenza tra la distribuzione della classe ricostruita e quella desiderata, a livello dei pixel.

La rappresentazione latente (interposta tra *encoder* e *decoder*) ha una dimensione minore rispetto al dato di input iniziale, riducendo la correlazione e la ridondanza delle *features* presenti; questo consente di ridurre il tempo di *processing* e la presenza di *overfitting*.

Ci sono varie implementazioni che seguono questa architettura; un esempio è dato dalla rete U-Net.



Walter Hugo, Lopez Pinaya, Sandra Vieira, Rafael Garcia-Dias, Andrea Mechelli. *Autoencoders*. Academic Press; 2020. 11: 195 p. <https://doi.org/10.1016/B978-0-12-815739-8.00011-0>

Figura 2.17: Struttura di un *autoencoder*. Due networks neurali, l'*encoder* ed il *decoder*, sono posti in successione, in maniera simmetrica, per ricostruire il dato iniziale (in questo caso uno 0).

2.9.1.1 U-Net

Vari modelli, tra cui quelli con struttura di tipo *encoder-decoder*, sono stati proposti per la segmentazione in campo medico. Ronnenberger *et al* [28] propose l'architettura U-Net per la segmentazione di immagine mediche rilevate al microscopio: la tradizionale struttura di un CNNs è stata modificata ed estesa, migliorando il risultato di segmentazione finale.

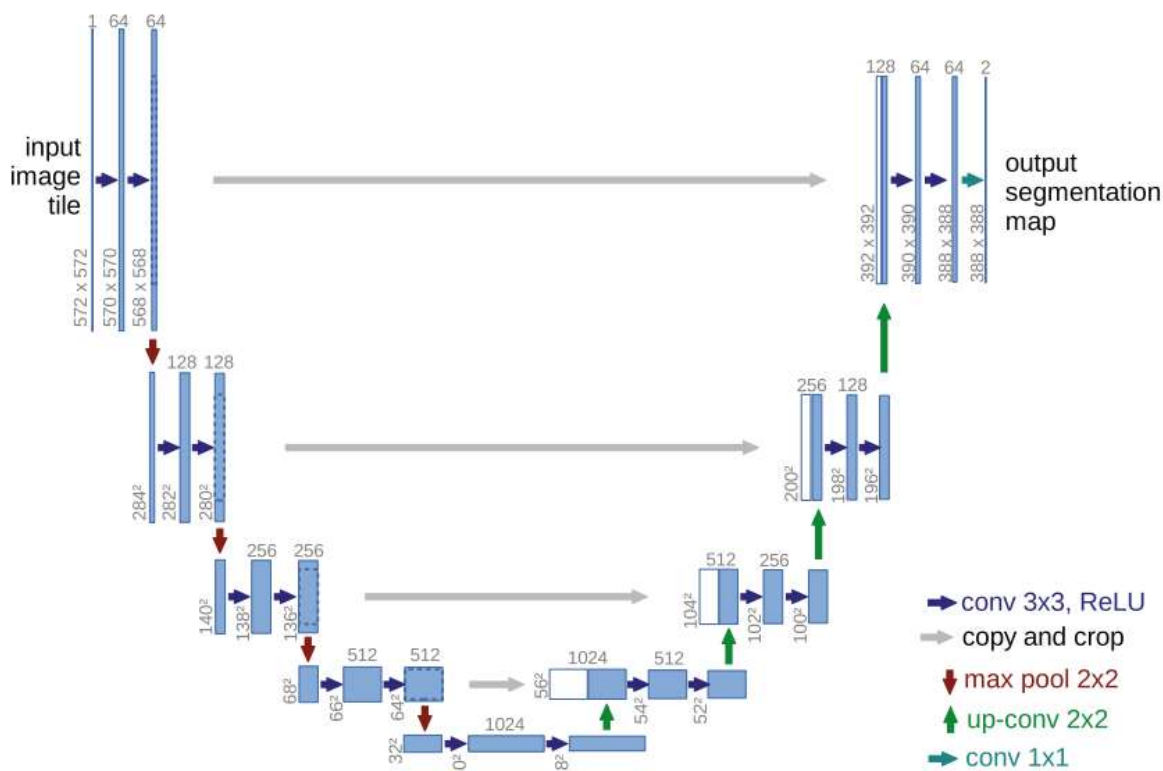
L'idea principale è quella di integrare il network di contrazione ponendo in successione vari *layers*, in cui si sostituiscono gli operatori di *pooling* con quelli di sovra-campionamento (*upsampling*), migliorando la risoluzione in uscita. Per poter localizzare le strutture da segmentare, le *features* che hanno ottenuto un'elevata risoluzione nel percorso di contrazione sono combinate con l'output ottenuto nel percorso di sovra-campionamento, ed un successivo *layer* di convoluzione impiega l'informazione finora ricavata per giungere all'output finale, come si può notare in Figura 2.18. Nella parte di sovra-campionamento è presente un ampio numero di *features channels*, che consentono al network di propagare l'informazione presente ai *layers* con un'elevata risoluzione. Il percorso di espansione (a destra), è simmetrico al percorso di contrazione (a sinistra), da cui il nome U-Net. Il network è privo di *fully connected layers*.

Nello specifico, il percorso di contrazione (a sinistra) segue la tipica architettura di un network di convoluzione: vi sono due convoluzioni 3x3, seguite dalla funzione d'attivazione ReLU e dall'operazione di *max-pooling* 2x2, con *stride* pari a 2. Ad ogni step di sotto-campionamento il numero di *feature channels* viene raddoppiato.

Per quanto riguarda il percorso di espansione (a destra), vengono eseguiti, in successione: un sovra-campionamento della *feature map* presente, una convoluzione 2x2 che dimezza il numero di *feature channels*, due ulteriori convoluzioni di tipo 3x3 e la funzione d'attivazione ReLU.

Nel *layer* finale, si utilizza una convoluzione 1x1 per mappare ogni *feature vector* verso il numero di classi desiderato.

Una caratteristica importante dell'architettura U-Net è la diretta connettività tra i *layers* di codificazione e decodificazione, che consente il riuso delle *features* estratte ed il rafforzamento della propagazione delle *features* [30], oltre che ridurre la vanificazione del gradiente durante la retro-propagazione dell'errore. Questa struttura viene comunemente impiegata per la segmentazione di immagini mediche.



Olaf Ronneberger, Philipp Fischer, Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Switzerland: Springer; 2015. 235 p.

Figura 2.18: Architettura U-Net. Ogni riquadro blu corrisponde ad una *feature map* di tipo multi-canale: il numero indicato sopra corrisponde al numero di canali, mentre quello presente in basso a sinistra si riferisce alla dimensione *x-y*. I riquadri bianchi corrispondono alle *feature maps* copiate, mentre le frecce colorate alle differenti operazioni, descritte nella legenda in basso a destra.

2.10 Deep learning in medicina

Con l'avvento dell'intelligenza artificiale (AI), le tecniche di *deep learning* (DL) hanno rivoluzionato il campo dell'*imaging* medico: con un adeguato dataset ed un appropriato addestramento, i modelli di DL, quali ad esempio i CNNs, sono risultati molto validi per la classificazione ed il *processing* d'immagini, riuscendo a rilevare ad alta precisione un danno o un'anomalia tessutale in maniera non invasiva.

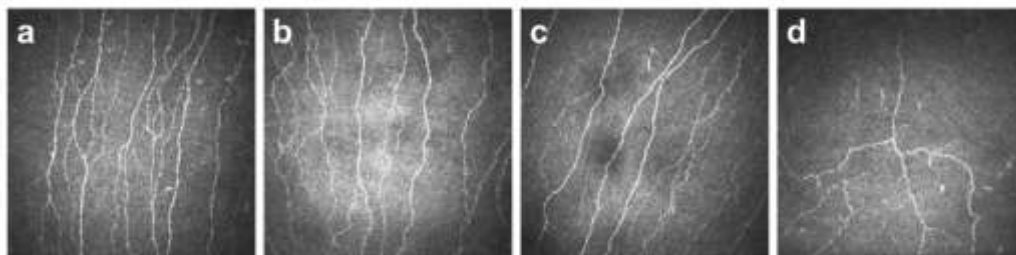
Vengono di seguito presentati due esempi di applicazione della rete U-Net per la segmentazione di nervi e cellule dendritiche corneali in soggetti sani o patologici tramite microscopia confocale, presso il plesso nervoso sotto-basale.

2.10.1 Segmentazione di nervi corneali

La quantificazione della morfologia del plesso nervoso sotto-basale richiederebbe maggior dispendio di tempo se non fosse condotta da algoritmi di intelligenza artificiale: tramite un CNN e l'impiego di *data augmentation* si riesce a quantificare in modo automatico il plesso nervoso corneale sotto-basale, in modo tale da diagnosticare, ad esempio, la presenza di neuropatia diabetica [30].

Le piccole fibre nervose sono le prime a risentire dell'incidenza di neuropatia diabetica periferica (DPN), com'è evidente in Figura 2.19. Per poter quantificare la morfologia di tali fibre, i nervi devono essere contraddistinti dal *background* o da altre strutture cellulari in maniera accurata; per questo è necessaria una procedura automatica.

Come spiegato precedentemente (sezione 2.8), le reti neurali convoluzionali sono indicate ad eseguire classificazione d'immagini, alleviando dal computare manualmente le *features* (passaggio necessario negli algoritmi di *machine learning* utilizzati in precedenza). Grazie alla loro architettura, infatti, estraggono le *features* e classificano le immagini. Quest'ultima operazione (la classificazione) viene svolta a livello dei pixel per completare la procedura di segmentazione, conferendo, nel complesso, le caratteristiche di obiettività, efficienza e riproducibilità della relativa diagnosi.



Williams BM, Borroni D, Liu R, Zhao Y, Zhang J, Lim J et al. *An artificial intelligence-based deep learning algorithm for the diagnosis of diabetic neuropathy using corneal confocal microscopy: a development and validation study*. Epub; 2019. 421 p. PMID: 31720728; PMCID: PMC6946763.

Figura 2.19: Immagini CCM del plesso nervoso sotto-basale. Le prime due (a-b) si riferiscono ad individui sani, le seconde due (c-d) ad individui con neuropatia diabetica.

Nello specifico, in uno studio è stato impiegato un dataset composto da 3835 immagini confocali (dimensione 384x384) del plesso nervoso sotto-basale corneale, distinguendo 3 gruppi: il primo, composto da 120 immagini, il secondo, costituito da 1578 immagini, ed il terzo dalle restanti 2137. Il 90% delle immagini del secondo gruppo è stato impiegato per la fase di *train*, la rimanente parte, insieme a tutte le immagini del primo gruppo, sono state impiegate per il *testing*.

È stato calcolato un *ground-truth* per ogni immagine tramite un programma Matlab® ed è stata impiegata una procedura di *data augmentation* per accrescere il dataset.

Addestrando la rete U-Net con il dataset descritto precedentemente, è stato evidenziato che la lunghezza di CNF nei volontari sani ($2695.2 \pm 606.8 \mu\text{m}$) presenta un valore più alto rispetto ai soggetti malati; il valore più basso è infatti dato da soggetti con diabete di tipo 1 ($1229.0 \pm 710.4 \mu\text{m}$) con neuropatia, rispetto a quelli senza o con diabete di tipo 2.

I risultati ottenuti grazie alla precedente tecnica di *deep learning* hanno dunque consentito di classificare gli individui con e senza neuropatia. Il valore della lunghezza e della densità di CNF, infatti, è la misura più affidabile nel diagnosticare la neuropatia diabetica, che comporta una riduzione delle fibre nervose corneali.

2.10.2 Segmentazione delle cellule dendritiche corneali

Come osservato nel capitolo 1, le cellule dendritiche nella cornea hanno un ruolo centrale in alcune patologie infiammatorie; pertanto riuscire ad individuarle ne agevola la diagnosi.

In uno studio [31] sono state considerate 100 immagini (384 x 384 pixel) del plesso nervoso corneale sotto-basale, sia della parte centrale che semi-periferica della cornea, di 20 soggetti sani e 20 post-LASIK. Il contorno di ogni cellula dendritica è stato tracciato manualmente. Per poter aumentare il dataset in fase di *training*, ogni immagine è stata flippata orizzontalmente e verticalmente (300 immagini) ed in qualità di procedura di validazione è stata usata la *leave-one-out-cross-validation*.

La segmentazione automatica è stata confrontata sia a livello del *pixel* che cellulare (regione connessa): nel primo caso è stato ottenuto un valore di TPR (*true positive rate*) pari al 92% e di

FDR (*false discovery rate*) al 37.6%, mentre a livello cellulare il TPR è risultato uguale al 96.3% ed il parametro FDR al 20%.

In quest'ultimo caso, i valori ricavati evidenziano la capacità dell'architettura U-Net di rilevare e misurare efficacemente le cellule dendritiche corneali, superando i limiti di variabilità intra/inter osservatore, dispendio temporale e non riproducibilità legati ai metodi manuali.

Capitolo 3

Segmentazione automatica delle cellule dendritiche corneali

L'obiettivo di questa tesi è investigare il miglior settaggio dei parametri di una rete U-Net per ottenere la segmentazione delle cellule dendritiche corneali, a partire da immagini acquisite mediante IVCCM (*in vivo corneal confocal microscopy*). Si sono inoltre confrontate le prestazioni ottenute da due *loss functions* differenti.

Come osservato nel primo capitolo, le cellule dendritiche svolgono un ruolo di sentinelle per le infezioni microbiche dell'occhio: in presenza di una patologia, modificano la loro densità e morfologia. Osservare ed analizzare tali cellule è dunque uno strumento di diagnosi.

Tuttavia, le analisi condotte dal medico richiedono molto tempo e sono soggettive, poiché dipendono dall'esperienza e dalla preparazione a riguardo. Con il passare degli anni, per superare questo problema, sono state proposte varie tecniche in grado di rendere automatica la segmentazione di oggetti d'interesse in immagini mediche.

Una rivoluzione nel campo dell'*image processing* e della classificazione d'immagini è stata ottenuta dai modelli di *deep learning*, che con un adeguato dataset ed un addestramento appropriato, consentono di migliorare notevolmente i valori di precisione ed accuratezza. In particolare, la rete neurale di tipo U-Net si è rivelata particolarmente adatta al problema di segmentazione semantica delle immagini acquisite al microscopio. Per questo motivo è stata impiegata in questo progetto di tesi.

L'algoritmo è stato interamente implementato in ambiente Matlab®, ed il flusso di lavoro ha previsto i seguenti passaggi:

1. Operazioni di *data augmentation* sul dataset a disposizione per migliorare la capacità di generalizzazione della rete;
2. Utilizzo della rete U-Net per la segmentazione delle cellule dendritiche corneali;
3. Definizione delle *loss functions* ed impostazione degli iper-parametri;
4. Addestramento e validazione della rete tramite *Leave-One-Out-Cross-Validation*;

5. Fase di *post-processing* e calcolo degli indici statistici TPR (*true positive rate*) e FDR (*false discovery rate*) a livello di regione.

Nei prossimi paragrafi verranno descritti in dettaglio:

- il dataset utilizzato (immagini e *labels*) per l'addestramento della rete;
- le *loss functions* e gli iper-parametri impiegati;
- la validazione della rete tramite *Leave-One-Out-Cross-Validation*;
- la procedura di *post-processing* utilizzata per migliorare i risultati ottenuti.

3.1 Dataset

Si è utilizzato un dataset composto da 100 immagini del plesso nervoso sub-basale della cornea di 100 soggetti differenti, acquisite mediante microscopia confocale corneale *in vivo*. Lo strumento di riferimento è *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)*. Le immagini sono state anonimizzate per eliminare qualsiasi informazione riferita ai pazienti. Ne è mostrato un esempio in figura 3.1.

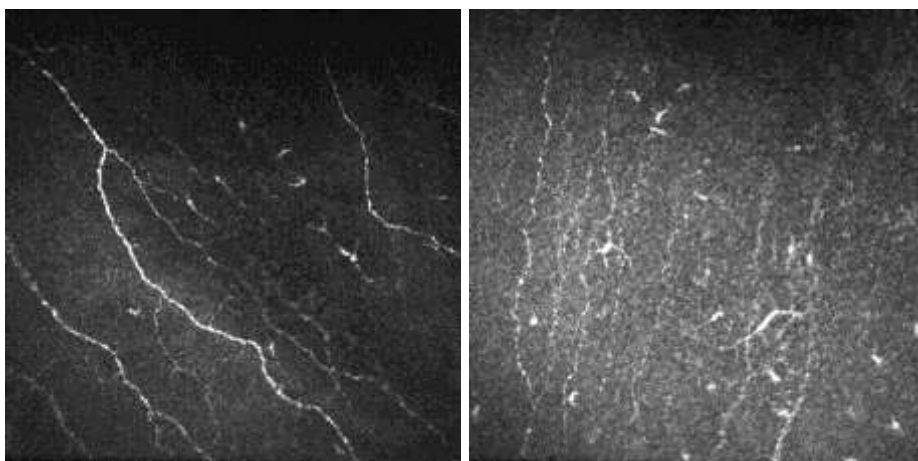


Figura 3.1: Esempio di immagini acquisite con IVCCM a livello del plesso nervoso sub-basale della cornea di due soggetti differenti.

3.1.1 Immagini pixel-label

Dal momento che la tipologia di apprendimento impiegata in questo progetto di tesi è di tipo supervisionato, per la fase di addestramento è necessario unire alle 100 immagini acquisite con IVCCM le rispettive etichette/*labels*, in cui ogni immagine viene etichettata, a livello del pixel, sulla base della categorie di appartenenza.

Come si può notare in Figura 3.2, le categorie d'interesse sono tre (tre distinti livelli di grigio): “cellula dendritica”, “nervo” e “sfondo”, il quale comprende tutto ciò che non rientra né nella prima né nella seconda categoria. Sarà più chiaro in seguito che l'aggiunta della categoria “nervi”, rispetto alla sola suddivisione tra DC e sfondo (problema di classificazione binario), migliorerà i risultati di segmentazione.

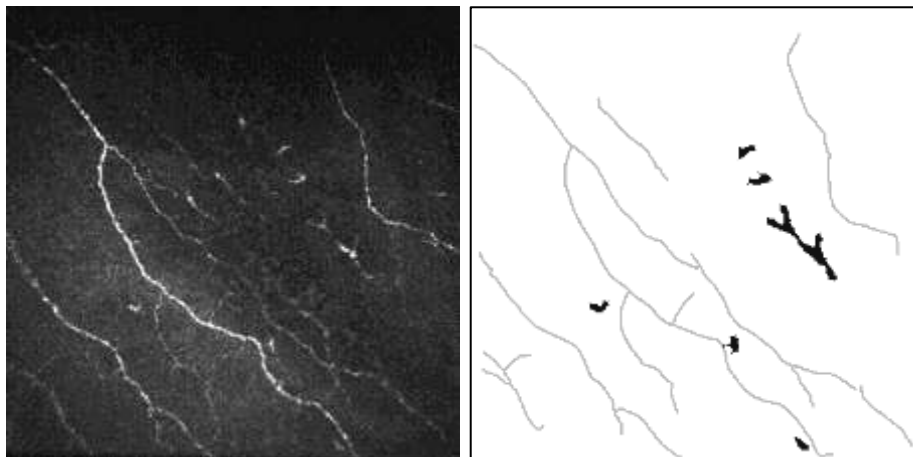


Figura 3.2: A sinistra, immagine originale IVCCM del plesso nervoso sotto-basale della cornea, a destra la rispettiva *label*: il colore bianco indica lo sfondo, il grigio i nervi ed il nero le cellule dendritiche.

Come indicato in un lavoro precedente [33], per l'individuazione della categoria “cellula dendritica” sono stati dapprima considerati i contorni individuati manualmente dai clinici, e, successivamente, sono state eseguite operazioni di sogliatura e morfologiche, sottoponendo ulteriormente il risultato ad un controllo manuale. Per identificare, a livello del pixel, la categoria “nervi”, invece, è stato utilizzato un algoritmo proposto da *Guimarães et al* [32], giungendo così infine alle immagini *pixel-label* distinte nei tre livelli di grigio.

3.2 Data Augmentation

Come osservato nel paragrafo 2.7.2.2, dal momento che il *deep learning* richiede un gran numero di dati in fase di addestramento, la tecnica di regolarizzazione di *data augmentation* permette di migliorare la capacità di generalizzazione della rete.

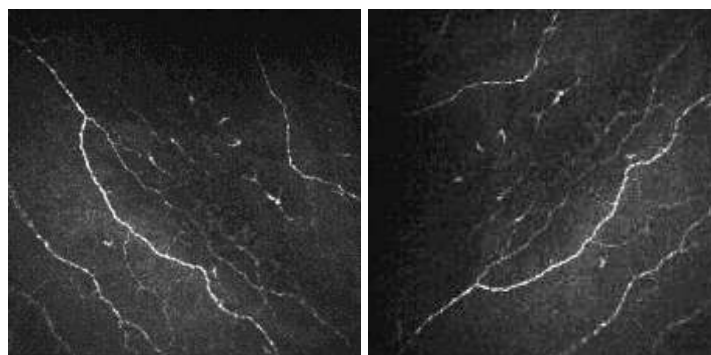
Tuttavia, risulta molto importante che le operazioni apportate su ciascuna immagine rispettino le naturali caratteristiche del dataset iniziale. Osservando la Figura 3.1, si può notare che le due immagini presentano due differenze principali: la prima è legata ad un diverso orientamento delle fibre nervose e delle cellule dendritiche, la seconda ad una differente luminosità (l'immagine a sinistra è più scura dell'immagine a destra), la quale è in parte dovuta alla curvatura corneale presente (parametro soggetto-dipendente), in parte ad un artefatto luminoso causato dallo strumento in fase d'acquisizione.

Di conseguenza, tra le possibili operazioni di *data augmentation*, si è preferito impiegare su ciascuna immagine: una rotazione antioraria di 90° , una variazione di luminosità ed un flip casuale, che verranno descritte in seguito.

È importante notare che le operazioni di *data augmentation* riguardano anche le *labels*, poiché costituiscono l'intero *training set* impiegato per addestrare la rete insieme alle immagini IVCCM. Dunque, per ogni coppia del dataset iniziale costituita dall'immagine IVCCM e dalla rispettiva *label* ne sono state ottenute altre tre, quadruplicando così il dataset di partenza.

3.2.1 Rotazione antioraria di 90°

La rotazione antioraria di 90° rende possibile ottenere delle immagini che potrebbero essere effettivamente acquisite tramite la tecnica IVCCM, com'è rappresentato in Figura 3.3.



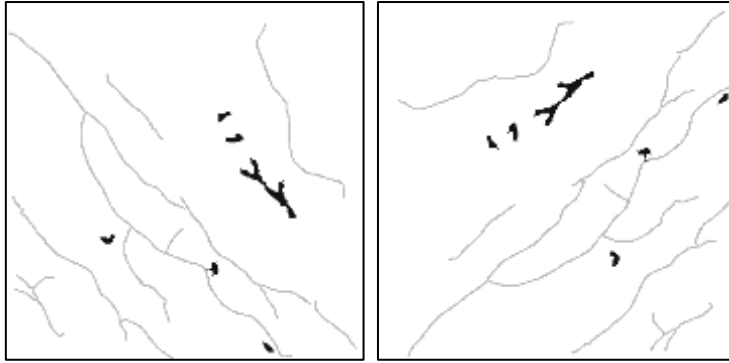


Figura 3.3: A sinistra, immagine IVCCM (in alto) e rispettiva *label* (in basso) nella posizione iniziale, a destra rotazione antioraria di 90° .

3.2.2 Variazione di luminosità

L'operazione di cambiamento di luminosità è molto delicata, poiché vengono modificati i valori dei pixel dell'immagine originale [21]. In questo caso, come spiegato precedentemente, il dataset iniziale presenta un'ampia variabilità in termini di luminosità. Rendere un'immagine più chiara o più scura di quella di partenza si presenta come un'operazione accettabile che, se svolta nella maniera corretta, non interferisce sull'integrità del dataset originale, anzi ne produce una versione aumentata da cui acquisire maggiore informazione nella fase di *train*.

Nello specifico, la luminosità è stata modificata sulla base di una soglia valutata empiricamente: se l'intensità media dell'immagine IVCCM originale superava il livello di grigio pari a 70 veniva scurita, schiarita in caso contrario. In particolare, sono state eseguite delle trasformazioni puntuali dei livelli di grigio delle immagini originali secondo l'equazione:

$$I_m = I^\gamma \quad (3.1)$$

Dove I è l'immagine originale, I_m quella modificata e γ è il coefficiente impiegato per rendere più chiara o più scura l'immagine (< 1 per schiarire, > 1 per scurire). Il coefficiente γ è stato dunque applicato in base all'intensità media dei livelli di grigio di ciascuna immagine, prestando attenzione a non interferire con la proprietà di verosimiglianza del dataset di partenza. È mostrato un esempio del risultato ottenuto in Figura 3.4.

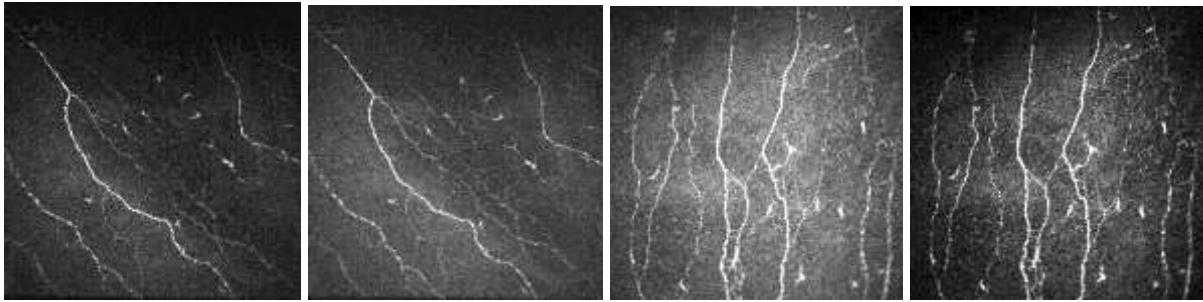


Figura 3.4: A partire da sinistra, rispettivamente: esempio di immagine originale IVCCM scura e successiva operazione di schiarimento ($\gamma = 0.75$), ulteriore esempio di immagine originale IVCCM chiara e successiva operazione di oscuramento ($\gamma = 1.55$).

3.2.3 Flip casuale

Come ultima operazione di *data augmentation* è stato impiegato il flip casuale: per ogni coppia definita dall'immagine IVCCM del *training set* e dalla rispettiva *label*, si è applicato il flip orizzontale o verticale sulla base del valore di un numero n , compreso nell'intervallo $[0,1]$, generato in maniera *random* da una distribuzione uniforme. Se n risultava inferiore o uguale a 0.5 è stato eseguito il flip orizzontale, verticale altrimenti, come mostrato in Figura 3.5.

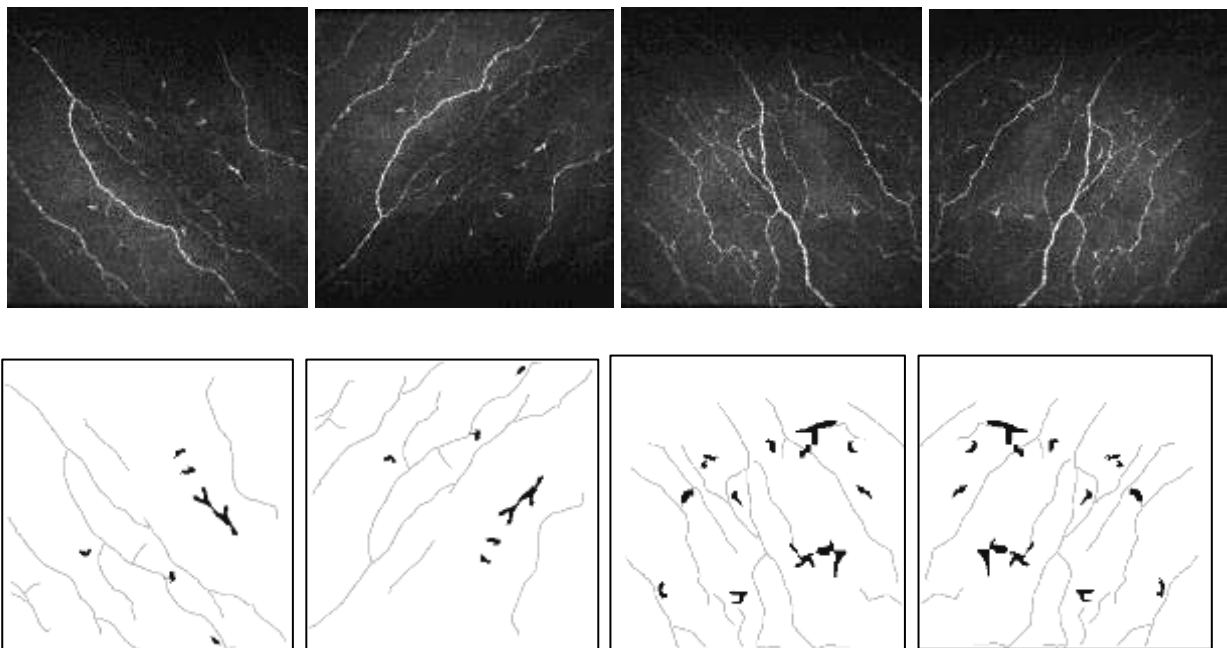


Figura 3.5: A partire da sinistra, rispettivamente: immagine originale IVCCM (in alto) e *label* corrispondente (in basso) nella posizione iniziale, successiva operazione di flip verticale ($n > 0.5$),

ulteriore esempio di immagine originale IVCCM (in alto) e *label* corrispondente (in basso) e successiva operazione di flip orizzontale ($n \leq 0.5$).

3.3 Addestramento della rete

3.3.1 Loss Functions per la segmentazione semantica

Nella segmentazione semantica ogni pixel viene classificato sulla base della categoria di appartenenza; tuttavia le classi non occupano lo stesso spazio in un'immagine: come si può notare in Figura 3.2, la categoria “sfondo” (colore bianco) risulta preponderante rispetto a “nervi” (colore grigio) o “cellula dendritica” (colore nero). È dunque presente un problema di sbilanciamento tra classi, molto frequente nel caso della segmentazione di immagini biomediche, il quale dev'essere preso in considerazione. A tale proposito, vengono di seguito descritte due *loss functions*: *Weighted Cross Entropy* e *Generalized Dice Loss*.

3.3.1.1 Weighted Cross Entropy

Nel paragrafo 2.6.1 era stata spiegata la funzione costo *Cross Entropy* nella sua versione tradizionale, sia per un problema di classificazione binario (es. “DC”, “sfondo”) sia multi-classe (es. “DC”, “nervi”, “sfondo”). Tuttavia, per correggere lo sbilanciamento tra classi, è necessario introdurre una penalità alle classi più frequenti, applicando ad esse dei pesi inversamente proporzionali alla frequenza della rispettiva classe, da cui il termine *Weighted Cross Entropy*. La formula è la seguente (per un problema di classificazione binario) [19]:

$$E_{WCE} = -\frac{1}{N} \sum_{i=1}^N (w g_i \log(p_i) + (1 - g_i) \log(1 - p_i)) \quad (3.2)$$

Dove p_i si riferisce alla probabilità di appartenere alla classe c predetta per il pixel i -esimo, g_i al valore del pixel i -esimo desiderato (*ground-truth*) e w al peso della classe c . Questa formula può inoltre essere riportata per un problema di classificazione multi-classe, come svolto per la tradizionale *Cross-Entropy* (2.20), sommando il contributo di ogni classe.

3.3.1.2 Generalized Dice Loss

La funzione *Generalized Dice Loss* consente di valutare la performance della segmentazione semantica misurando la corrispondenza tra l'immagine di segmentazione ottenuta dalla rete e quella desiderata (*ground-truth*). Viene corretto il problema dello sbilanciamento tra classi mediante dei pesi, i quali sono inversamente proporzionali alla regione occupata da ciascuna classe nelle immagini di *ground-truth*, in modo da evitare che le classi dominanti prevalgano sul contributo alla funzione costo. La formula è la seguente (per un problema di classificazione binario) [19]:

$$E_{GDice} = 1 - 2 \frac{\sum_{c=1}^2 w_c \sum_{i=1}^N p_{ci} g_{ci}}{\sum_{c=1}^2 w_c \sum_{i=1}^N p_{ci} + g_{ci}} \quad (3.3)$$

Dove g_{ci} rappresenta il valore del pixel i -esimo desiderato e p_{ci} la probabilità predetta del pixel i -esimo per la classe c . Come spiegato precedentemente, il contributo dei pesi w_c riferiti alla classe c è dato da:

$$w_c = \frac{1}{(\sum_{i=1}^N g_{ci})^2} \quad (3.4)$$

3.3.2 Iper-parametri

In fase di addestramento, la selezione degli iper-parametri è molto importante poiché definisce il comportamento della rete neurale. Il valore degli iper-parametri è stato corretto mediante *cross-validation*, che verrà discussa in seguito, ottenendo così il set tale da garantire la migliore *performance*.

Vengono riportati in seguito gli iper-parametri impiegati in questo lavoro di tesi per l'addestramento della rete U-Net:

- Metodo di discesa del gradiente: è stato impiegato lo *stochastic gradient descent with momentum*, frequentemente utilizzato negli algoritmi di *deep learning*. Per quanto riguarda il valore del *momentum*, è stato settato al valore di *default* pari a 0.9;

- *Max epochs*: il numero massimo di epoche, che si riferisce al numero di volte che l'algoritmo di addestramento della rete prende in esame l'intero dataset di *training*, è stato impostato a 100. Inizialmente era stato settato a 76, tuttavia, mediante ispezione visiva dei risultati di segmentazione, si è preferito aumentarlo;
- *Mini-batch size*: la dimensione del *mini-batch* corrisponde al sottoinsieme delle immagini del *training set* che viene impiegato per valutare il gradiente della funzione costo, aggiornando di conseguenza il valore dei pesi; nel momento in cui l'algoritmo di discesa del gradiente percorre l'intero sottoinsieme, si completa un'iterazione. In questo caso, si avevano a disposizione, per ogni ciclo di *training*, 396 immagini (sarà più chiaro in seguito), dunque si è impostato il valore del *mini-batch* a 33 per permettere alle rete di analizzare tutte le immagini in un'epoca;
- *Learning rate*: esso determina l'ampiezza degli step percorsi dall'algoritmo di discesa del gradiente per raggiungere il minimo assoluto della funzione costo; valori troppo bassi renderebbero troppo lento il processo di addestramento, mentre valori troppo elevati potrebbero impedire il raggiungimento del minimo assoluto. Si è deciso dunque di variare il valore del *learning rate* durante la fase di *training* della rete nella seguente maniera: a partire da un valore iniziale pari a 0.01, è stato ridotto di un fattore 0.1 ogni 25 epoche; in questo modo, si sono impiegati inizialmente passi più ampi nella discesa del gradiente, e sono stati poi ridotti progressivamente;
- *L2 Regularization*: esso corrisponde al coefficiente di regolarizzazione λ (2.22) impiegato per migliorare la capacità di generalizzazione della rete e ridurre la possibilità di *overfitting*. È stato impostato al valore di *default* pari a 0.0001.

3.3.3 Leave-One-Out-Cross-Validation

Dal momento che il dataset a disposizione ha cardinalità limitata, anziché suddividere in *training set*, *validation set* e *test set*, si è preferito impiegare la tecnica di *Leave-One-Out-Cross-Validation*.

A partire dall'intero dataset composto dalle 400 immagini ottenute a seguito di *data augmentation*, ad ogni ciclo di *training* (per un totale di 100 poiché tale è il numero dei soggetti), il *validation set* era costituito dal sotto-gruppo delle immagini (4 in totale) riferite ad un rispettivo soggetto, mentre il *training set* era dato dalle 396 immagini rimanenti. La rete veniva dunque addestrata, di volta in volta, su un diverso sottoinsieme di immagini.

In questo modo, la tecnica di *Leave-One-Out-Cross-Validation* è stata applicata 100 volte, in maniera tale da validare la rete sul totale dei 100 soggetti. Tuttavia, la valutazione degli iperparametri è stata eseguita osservando, mediante ispezione visiva, le prestazioni di segmentazione semantica della rete solamente sull'immagine originale (riferita al soggetto) del *validation set*, non su quelle aumentate.

3.4 Post-Processing

Le immagini di segmentazione delle cellule dendritiche ottenute dalla rete neurale presentano risultati differenti in base al numero di classi impiegato (se due o tre), soprattutto per quanto riguarda le zone di concentrazione dei falsi positivi. L'obiettivo della procedura di *post-processing* è dunque migliorare i risultati ottenuti dalla rete neurale.

Nei risultati di segmentazione semantica impiegando tre classi (es. "DC"/ "nervi"/ "sfondo") la presenza dei falsi positivi interessava le zone circostanti le DCs ed alcune aree luminose erroneamente identificate come cellule dendritiche; mentre, nelle immagini ottenute con la classificazione binaria "DC"/ "sfondo", si è potuto notare un elevato numero di falsi positivi anche nei tratti nervosi, com'è rappresentato in Figura 3.6. Per quanto riguarda le immagini ottenute dalle due *loss functions*, non si sono osservate particolari differenze, come sarà evidente nei risultati indicati nel capitolo successivo.

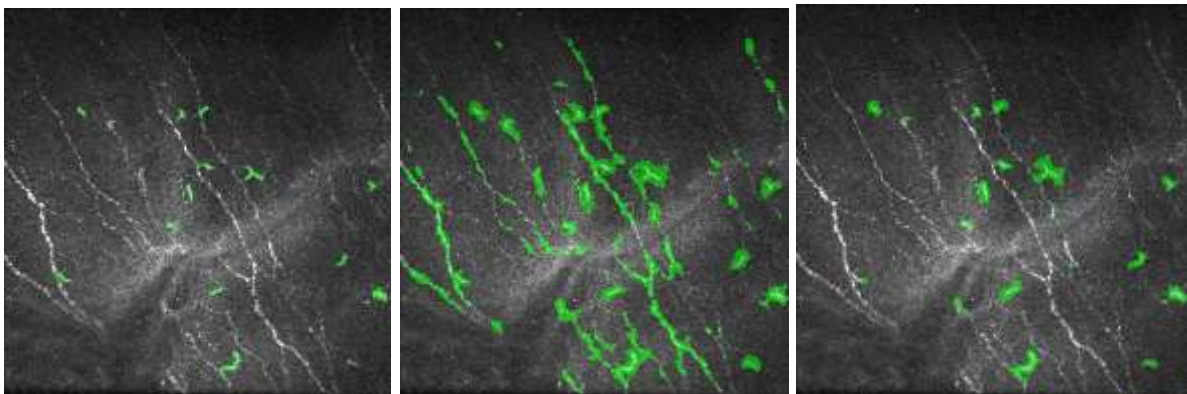


Figura 3.6: A sinistra, segmentazione manuale delle cellule dendritiche; al centro e a destra risultato di segmentazione semantica ottenuto dalla rete U-Net, rispettivamente con due e tre classi.

Per riconoscere i falsi positivi concentrati nei nervi (segmentazione con due classi) si è impiegata la trasformata di Hough. Nello specifico, per ogni immagine di segmentazione ottenuta dalla rete, è stata realizzata una maschera in modo da eliminare le regioni rettilinee individuate dalla trasformata, che corrispondevano a buona parte dei falsi positivi concentrati lungo i tratti nervosi. Ne è mostrato un esempio in Figura 3.7.

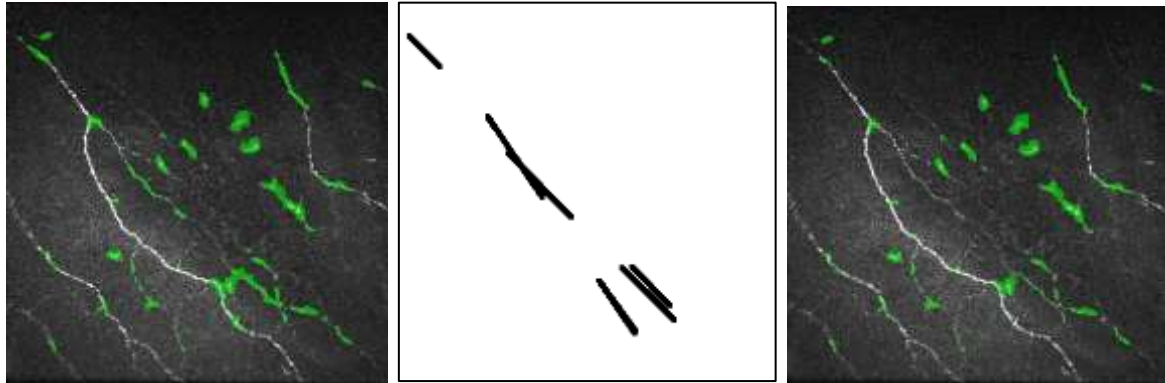


Figura 3.7: A sinistra, risultato di segmentazione ottenuto dalla rete, al centro maschera realizzata mediante l'utilizzo della trasformata di Hough, a destra risultato finale ottenuto.

Successivamente, per scartare i falsi positivi concentrati nelle zone circostanti le cellule dendritiche, presenti in entrambi i casi di segmentazione con due e tre classi, si è eseguita un'operazione globale di sogliatura sulla base del valore medio dell'intensità dell'immagine originale, in maniera tale da rendere la forma delle cellule dendritiche simile a quella ottenuta con la segmentazione manuale. Nel caso della segmentazione con due classi, è stato inoltre necessario eseguire un'operazione morfologica di erosione per migliorare ulteriormente i risultati. Come ultima operazione, sono state filtrate le regioni di area molto piccola in modo da eliminare i falsi positivi rimasti.

In figura 3.8 è riportato un esempio del risultato ottenuto a seguito delle precedenti operazioni per la segmentazione con due classi.

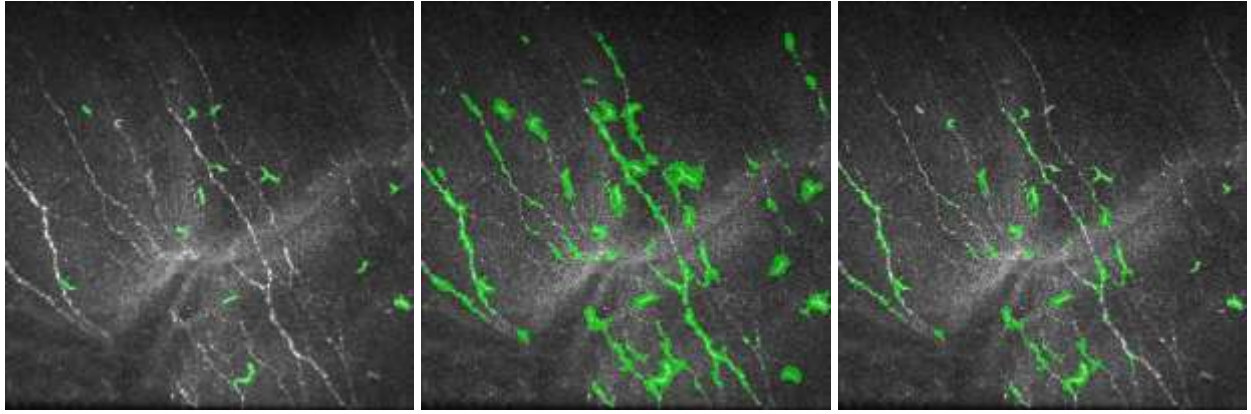


Figura 3.8: A sinistra, segmentazione manuale delle cellule dendritiche, al centro segmentazione con due classi ottenuta dalla rete U-Net ed a destra risultato successivo al *post-processing*.

La fase *post-processing* ha dunque consentito di eliminare buona parte dei falsi positivi; tuttavia, le restrizioni apportate dalle procedure di erosione e di filtraggio dell'area hanno scartato anche alcune regioni correttamente identificate dalla rete come cellule dendritiche, riducendo il valore della sensibilità finale.

Di seguito è riportato un esempio (Figura 3.9) delle operazioni precedentemente descritte nel caso della segmentazione con tre classi.

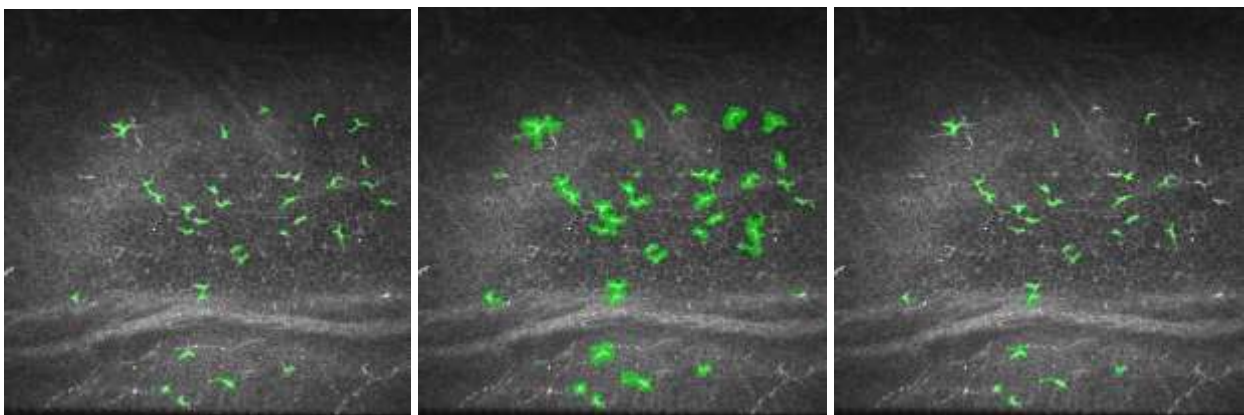


Figura 3.9: A sinistra è mostrata la segmentazione manuale delle cellule dendritiche, al centro la segmentazione con tre classi ottenuta dalla rete U-Net ed a destra il risultato finale ottenuto dalla procedura di *post-processing*.

Capitolo 4

Risultati

In questo capitolo vengono riportati i risultati ottenuti per quanto riguarda la segmentazione semantica delle cellule dendritiche corneali. I risultati sono frutto dell'analisi automatica descritta al capitolo precedente, che prevedeva l'utilizzo della rete U-Net (addestrata sia con sole due classi, sia con tre), ed il confronto di due differenti *loss functions*. L'analisi dei risultati verrà effettuata sia nel momento precedente la fase di *post-processing*, sia nel momento successivo. Questa doppia analisi ci permetterà non solo di indagare quali siano le variazioni legate al solo cambio di *loss function* o del numero di classi, ma anche come queste differenze possano portare ad una fase di *post-processing* più o meno efficace (e conseguentemente a risultati più o meno buoni).

4.1 Analisi statistica

Per la fase di test sono stati presi in analisi i risultati di segmentazione semantica ottenuti per le immagini originali escluse dall'addestramento in fase di *cross-validation*. Le prestazioni della rete sono state dapprima valutate mediante ispezione visiva e, successivamente, sono stati calcolati gli indici statistici sulla base della corrispondenza tra le regioni di segmentazione (intese come pixel connessi) ottenute in maniera automatica, e quelle presenti nel *ground-truth* di riferimento (segmentazione manuale).

In particolare, le performance della rete sono state valutate in termini di TPR (*true positive rate*) e FDR (*false discovery rate*).

La sensibilità, o TPR, indica la percentuale delle cellule dendritiche correttamente individuate dalla rete. Essa viene espressa in funzione di ulteriori indici statistici: TP (*true positive*) e FN (*false negative*). Il primo (TP) si riferisce agli oggetti individuati dalla rete come cellule dendritiche che risultano tali anche nel manuale; il secondo, ossia il valore di FN, corrisponde agli oggetti che risultano cellule dendritiche nel *ground-truth*, ma che non sono stati identificati come tali dalla rete. È dunque possibile definire il valore di TPR nella seguente maniera:

$$\text{TPR} = \frac{TP}{TP+FN} \quad (4.1)$$

Il *false discovery rate* (FDR) definisce la percentuale delle cellule dendritiche erroneamente identificate dalla rete. Esso è funzione del valore di TP e FP (*false positive*), il quale si riferisce agli oggetti identificati dalla rete come cellule dendritiche ma che non risultano tali nel manuale di riferimento. Il *false discovery rate* viene calcolato nella maniera seguente:

$$\text{FDR} = \frac{FP}{FP+TP} \quad (4.2)$$

La Figura 4.1 mostra sulla sinistra l'analisi manuale, e sulla destra il risultato del confronto dell'analisi manuale con l'analisi della segmentazione dopo l'utilizzo della rete U-Net seguita dalla fase di *post-processing*. Nell'immagine di destra, è evidenziata la differenza tra l'analisi manuale e quella automatica dalla suddivisione tra TP (colore verde), FP (colore rosso) e FN (colore blu).

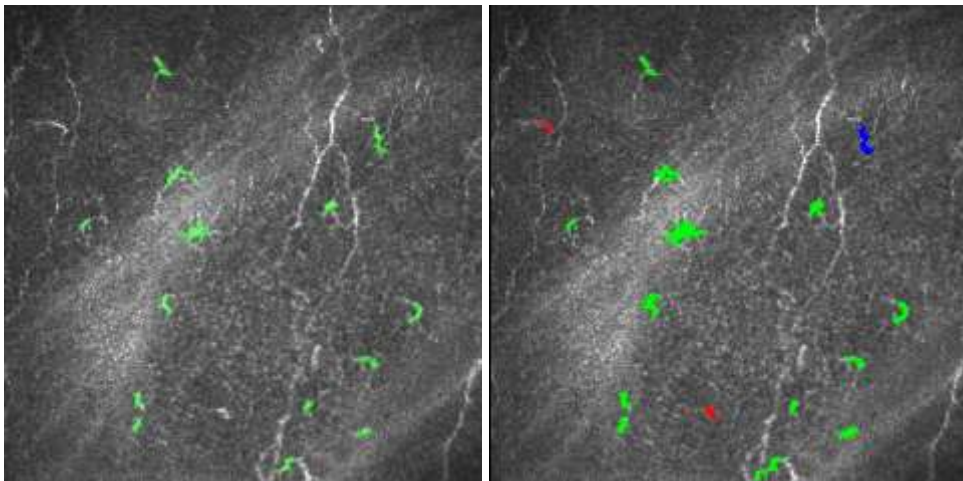


Figura 4.1: A sinistra, segmentazione manuale delle cellule dendritiche (colore verde), a destra risultato ottenuto dalla segmentazione semantica con la rete U-Net, dove in verde sono indicati i TP, in rosso i FP ed in blu i FN.

Vengono riportati in seguito i valori di TPR e FDR ottenuti, in termini di valore medio \pm deviazione standard, per le due differenti *loss functions*, sia nel caso della segmentazione con due classi sia in quello con tre (rispettivamente Tabella 4.1 e 4.2), precedentemente alla fase di *post-processing*.

	TPR (media \pm sd)	FDR (media \pm sd)
<i>Weighted Cross Entropy</i>	99.28 \pm 3.45	81.11 \pm 13.30
<i>Generalized Dice Loss</i>	99.28 \pm 3.45	81.31 \pm 12.78

Tabella 4.1: Valori di TPR e FDR ottenuti dalla segmentazione automatica con **due** classi precedentemente al *post-processing*.

	TPR (media \pm sd)	FDR (media \pm sd)
<i>Weighted Cross Entropy</i>	94.66 \pm 10.29	58.05 \pm 19.37
<i>Generalized Dice Loss</i>	94.99 \pm 10.24	57.57 \pm 19.43

Tabella 4.2: Valori di TPR e FDR ottenuti dalla segmentazione automatica con **tre** classi precedentemente al *post-processing*.

Com'è possibile notare dalle Tabelle 4.1 e 4.2, le due *loss functions* hanno ottenuto prestazioni molto simili, dal momento che la differenza tra i valori medi di TPR e di FDR è minore dello 0.5 %. La segmentazione con due classi ha ottenuto ottimi risultati in termini di sensibilità, ma il valore di FDR è risultato troppo alto, poiché, come osservato nel paragrafo 3.4, si sono riscontrati numerosi falsi positivi concentrati lungo i nervi, aumentando il valore di FDR. Infatti, aggiungendo la classe “nervo” alla semplice suddivisione “DC”/“sfondo”, si è potuto notare un valore ancora ottimo in termini di sensibilità ed una diminuzione (23 % in media) del valore di FDR, il quale, tuttavia, rimane ancora elevato.

Nello specifico, un alto valore di FDR è stato riscontrato in determinate immagini, per la presenza di: aree piccole e bianche potenzialmente riconducibili per forma e luminosità alle cellule dendritiche, alcuni tratti nervosi, artefatti luminosi ed alcuni strati corneali differenti dal plesso nervoso sub-basale. Ne è mostrato un esempio in Figura 4.2.

Di conseguenza, per ridurre il valore di FDR si è eseguita la fase di *post-processing*, della quale vengono riportati in seguito gli indici statistici ottenuti sia per la segmentazione con due classi sia in quella con tre, per le due *loss functions* differenti, in termini di valore medio \pm deviazione standard.

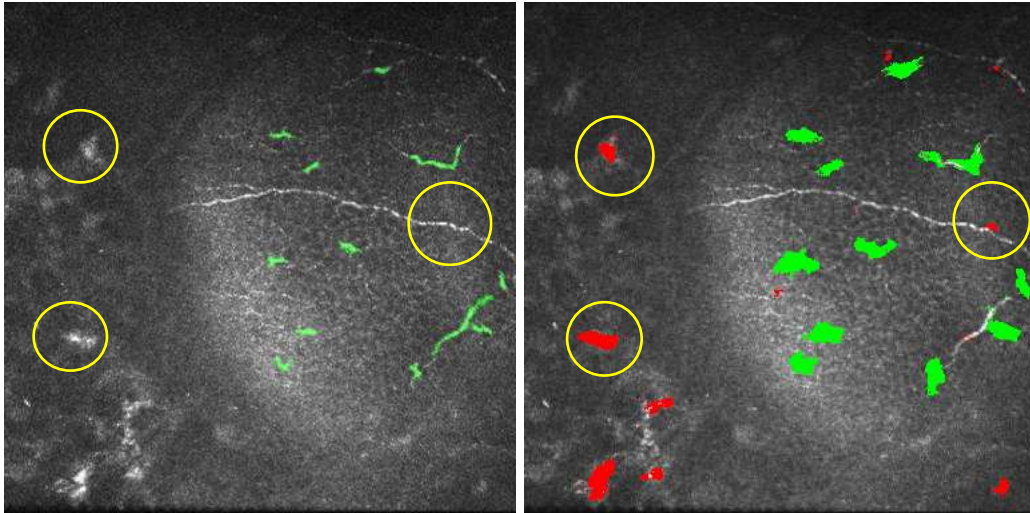


Figura 4.2: Zone di concentrazione dei falsi positivi in particolari aree dell'immagine (cerchio giallo) quali macchie luminose o tratti nervosi.

	TPR (media \pm sd)	FDR (media \pm sd)
<i>Weighted Cross Entropy</i>	77.60 \pm 19.27	45.61 \pm 23.05
<i>Generalized Dice Loss</i>	77.16 \pm 19.37	45.38 \pm 23.80

Tabella 4.3: Valori di TPR e FDR ottenuti dalla segmentazione automatica con **due** classi successivamente al *post-processing*.

	TPR (media \pm sd)	FDR (media \pm sd)
<i>Weighted Cross Entropy</i>	80.89 \pm 18.29	21.34 \pm 18.60
<i>Generalized Dice Loss</i>	80.49 \pm 19.01	21.24 \pm 19.12

Tabella 4.4: Valori di TPR e FDR ottenuti dalla segmentazione automatica con **tre** classi successivamente al *post-processing*.

Com'è possibile notare dalle Tabelle 4.3 e 4.4, anche in questo caso le due *loss functions* hanno ottenuto prestazioni molto simili: in termini di valore medio, la *Weighted Cross Entropy* ha ottenuto

prestazioni leggermente migliori della *Generalized Dice Loss* per quanto riguarda la sensibilità, sia nella segmentazione con due classi sia in quella con tre.

Rispetto alle Tabelle 4.1 e 4.2, nelle Tabelle 4.3 e 4.4 è importante osservare che il valore della sensibilità è diminuito del 14 % circa per la segmentazione con tre classi, e del 22 % circa per la segmentazione con due classi. Questo è dovuto alle operazioni di *post processing*. Per quanto riguarda la segmentazione con tre classi, l'operazione di soglia globale ed il successivo filtraggio dell'area hanno scartato, oltre a molti falsi positivi, anche alcune regioni di dimensione ridotta, o con area non ben definita, correttamente identificate dalla rete come cellule dendritiche. Ne è un esempio la Figura 4.3: nell'immagine raffigurata a destra, che rappresenta il risultato di segmentazione semantica ottenuto successivamente al *post-processing*, il numero di falsi negativi è aumentato (diminuendo, di conseguenza, la sensibilità) rispetto all'immagine raffigurata al centro, su cui non è stata eseguita alcuna operazione di *post-processing*.

Il motivo per il quale la sensibilità, nella segmentazione con due classi, è maggiormente diminuita (rispetto al caso con tre) risiede nel fatto che, come spiegato nel paragrafo 3.4, è stata eseguita un'ulteriore operazione morfologica di erosione per migliorare il risultato finale (dal momento che la concentrazione di falsi positivi era molto alta), a discapito della sensibilità.

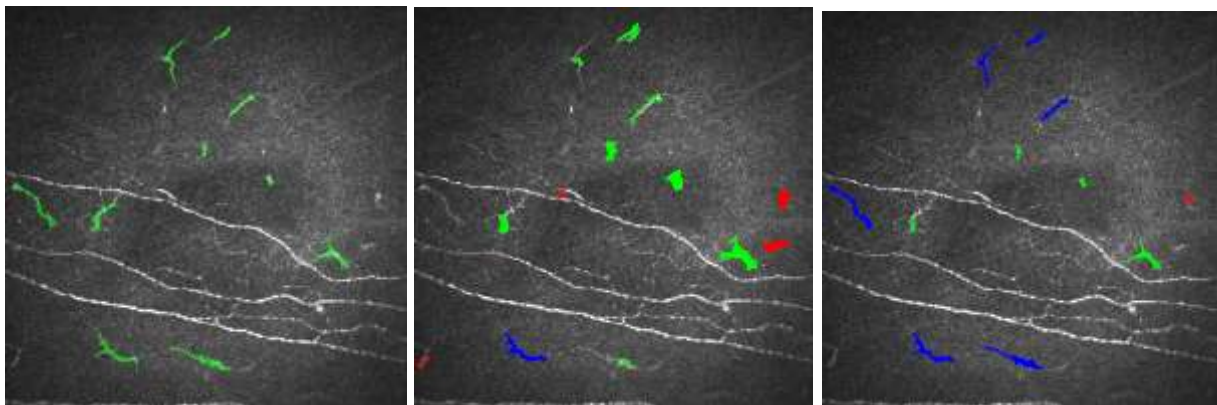


Figura 4.3: A sinistra, segmentazione manuale delle cellule dendritiche, al centro ed a destra, rispettivamente, segmentazione automatica delle cellule dendritiche prima e dopo *post-processing*.

Dunque, considerando la segmentazione con tre classi, i valori medi finali di sensibilità e FDR ottenuti sono risultati, rispettivamente, dell'80.7 % e del 21 %. È importante sottolineare che, a fronte di una riduzione di sensibilità, del 14 % rispetto alla fase precedente al *post-processing*, il

FDR risulta ridotto del 36.5 %: il valore di sensibilità ottenuto, infatti, non è particolarmente elevato, ma può essere spiegato per le operazioni di *post-processing* eseguite per ridurre il numero di falsi positivi (inizialmente molto alto).

Talvolta, alcuni falsi positivi potrebbero essere dovuti a cellule dendritiche realmente presenti a livello del plesso nervoso sub-basale e dunque correttamente identificate dalla rete, ma non individuate dalla segmentazione manuale, soggetta al parere del clinico. Ne è mostrato un esempio in Figura 4.4, nella quale con il cerchio verde e con quello giallo vengono identificati, rispettivamente, un TP ed un FP.

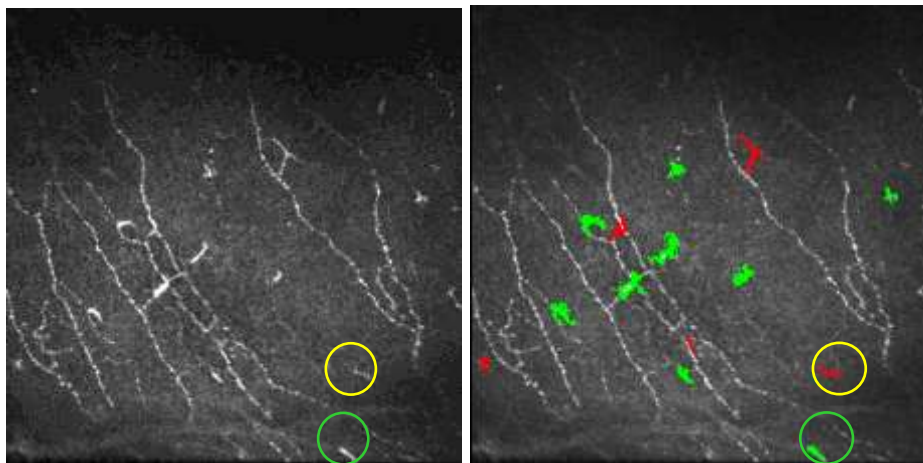


Figura 4.4: A sinistra, immagine originale IVCM, a destra, risultato di segmentazione automatica successivamente al *post-processing*. Il cerchio giallo indica l'individuazione di un FP, il cerchio verde un TP.

Inoltre, in alcune immagini si è potuto osservare un valore di FDR molto al di sopra della media. Ciò può essere spiegato per il fatto che, in fase d'acquisizione, si sono presentati alcuni "errori": in alcune immagini, infatti, compaiono *pattern* appartenenti a strati corneali differenti dal plesso nervoso sub-basale, in altre *pattern* legati ad artefatti luminosi. Queste caratteristiche si presentano come strutture simili per forma e luminosità alle cellule dendritiche e, per tal motivo, vengono classificate erroneamente dalla rete (Figura 4.5). In una sola immagine, invece, i falsi positivi sono risultati concentrati lungo il tratto nervoso anche nel caso della segmentazione con tre classi, aumentando, di conseguenza, il valore di FDR.

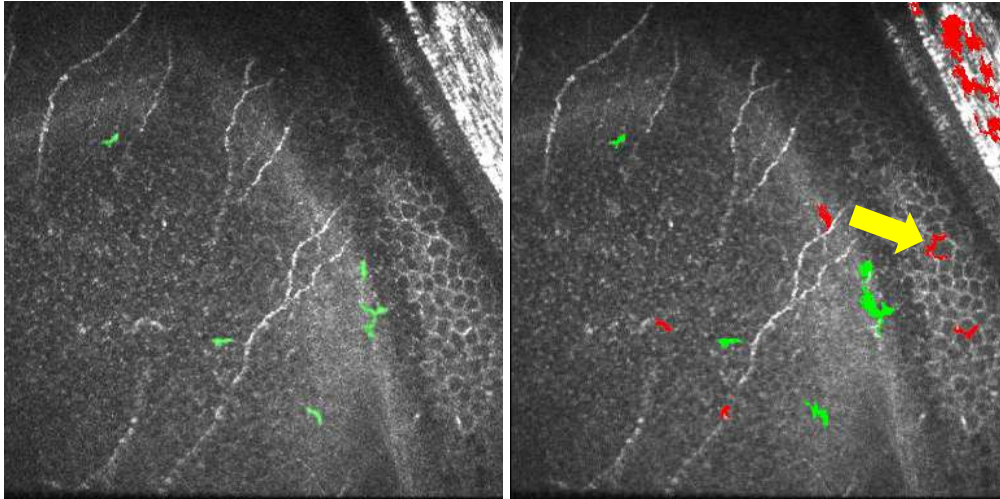


Figura 4.5: A sinistra, segmentazione manuale, a destra segmentazione automatica. Con la freccia gialla sono indicati i FP rilevati dalla rete in strati epiteliali differenti dal plesso nervoso sotto-basale

I risultati ottenuti, dunque, indicano che l'utilizzo di una rete neurale convoluzionale permette di ottenere buoni valori di sensibilità e di FDR, che possono comunque essere migliorati.

In questo progetto di tesi, l'obiettivo era quello di identificare il miglior settaggio per ottenere la segmentazione delle cellule dendritiche corneali, applicando operazioni di *data augmentation* differenti dal flip orizzontale e verticale; sono state, infatti, introdotte le operazioni di rotazione angolare di 90° e di variazione di luminosità.

Tuttavia, nella fase precedente al *post-processing*, si è riscontrato un valore elevato di falsi positivi, specialmente nella segmentazione con due classi, in cui le zone di concentrazione interessavano per lo più il tratto nervoso. E' dunque possibile ottenere migliori risultati, in termini di sensibilità e di FDR, modificando, ad esempio, il dataset di addestramento della rete e l'impostazione di alcuni iper-parametri.

Per quanto riguarda il dataset di addestramento, esso dipende sia dal numero di immagini inizialmente a disposizione, sia dalle operazioni di *data augmentation*. Nell'algoritmo proposto, per aumentare il dataset di *train*, si sono utilizzati, oltre alla rotazione antioraria di 90° , il flip casuale e la variazione di luminosità sulla base, rispettivamente, di un numero generato casualmente e di una soglia basata sulla luminosità. La modifica della luminosità è stata eseguita per ottenere immagini che potrebbero essere realmente acquisite dalla tecnica di IVCCM; tuttavia, rimane un'operazione delicata poiché è di tipo non lineare, ed i pixel delle immagini vengono modificati. Di conseguenza,

è possibile che l'applicazione di operazioni di *data augmentation* differenti dalla variazione di luminosità, unita all'acquisizione di ulteriori immagini IVCCM, migliori il valore finale di sensibilità.

Conclusioni

La cornea è il tessuto maggiormente innervato del corpo umano ed ha un'elevata sensibilità: in particolare, a livello del plesso nervoso sub-basale, sono state rilevate le cellule dendritiche, che proteggono l'occhio da agenti esterni e partecipano alla risposta immunitaria adattativa.

La microscopia confocale corneale *in vivo* (IVCCM) consente la visualizzazione diretta ed in tempo reale della numerosità e della morfologia delle cellule dendritiche in maniera non invasiva. In numerosi studi, è stata infatti riportata la correlazione tra le condizioni di densità e morfologia delle cellule dendritiche e la presenza di patologie. Solitamente, l'analisi delle cellule dendritiche viene eseguita dai medici, risultando estremamente laboriosa ed incline all'esperienza; rendere automatico questo processo può quindi agevolare la diagnosi di particolari patologie.

In questo progetto di tesi, è stato sviluppato un algoritmo, mediante tecniche di *deep learning*, in grado di segmentare in maniera automatica le cellule dendritiche a partire da 100 immagini (100 differenti soggetti) acquisite con la tecnica IVCCM presso la *School of Optometry and Vision Science* della *University of New South Wales* (UNSW, Sydney). Nello specifico, è stata utilizzata una rete neurale convoluzionale (CCN) di tipo U-Net, particolarmente efficiente nella segmentazione di oggetti a partire da immagini mediche rilevate al microscopio.

L'algoritmo sviluppato ha consentito di ottenere valori di TPR e FDR pari, rispettivamente, all'80.7 % ed al 21%. Il valore di TPR fornito inizialmente dalla rete è risultato ottimo ma, allo stesso tempo, il numero di falsi positivi era molto alto. L'elevata presenza di FP nelle immagini in uscita dalla rete dipende sia dal dataset impiegato per addestrare il network (è importante sottolineare la limitatezza del numero di immagini a disposizione), sia dall'impostazione degli iper-parametri, per i quali si suggerisce un *tuning* più accurato. Inoltre, vi sono alcune immagini in cui è stato possibile riscontrare la presenza di artefatti luminosi ed altri strati della cornea differenti dal plesso nervoso sub-basale, aumentando il numero di falsi positivi. A seguito delle operazioni di *post-processing*, è stato possibile diminuire notevolmente il valore di FDR, a discapito, però, della sensibilità.

I risultati ottenuti in questa tesi sono incoraggianti: sarà interessante indagare ulteriormente sui parametri da settare e sulla fase di *post-processing* per migliorare le prestazioni dell'algoritmo proposto, con il fine ultimo di aumentare il TPR e ridurre ulteriormente il valore di FDR.

Oltre ai parametri, dato che la segmentazione con tre classi ha ottenuto migliori risultati rispetto a quella con due, è lecito supporre che aggiungendo ulteriori categorie, sia possibile incrementare la *performance* finale, con lo scopo di migliorare l'abilità di diagnosi. Le categorie da inserire potrebbero risultare da una distinzione dello stato di maturazione delle cellule stesse (esse

presentano una struttura morfologica differente se si trovano in uno stato maturo o immaturo), oppure dall'analisi di altre strutture presenti nell'immagine, quali, ad esempio, i neuromi (com'è stato fatto per i nervi).

Bibliografia

1. *Enciclopedia della Scienza e della Tecnica*. Treccani; 2008.
2. S. Adamo, M. De Felici, A. Dolfi, A. Filippini, M. Grano, A. Musarò et al. *Istologia di Monesi*. Padova: Piccin; 2018 (7^a edizione). 633 p.
3. Abul K. Abbas, Andrew H. Lichtman, Shiv Pillai, Silvano Sozzani et al. *Immunologia cellulare e molecolare*. Milano: Edra; 2018 (9^a edizione).
4. Giuseppe Ferreri, Anna M. Roszkowska et al. *Semeiotica della cornea*. Palermo: MEDICAL BOOKS; 2007. 19-28 p.
5. Mandathara PS, Stapleton FJ, Kokkinakis J, Willcox MDP. *A pilot study on corneal Langerhans cells in keratoconus*. *Cont Lens Anterior Eye*. 2018. 41(2): 219-223 p. doi: 10.1016/j.clae.2017.10.005. Epub 2017 Oct 21. PMID: 29066264.
6. Chiang JCB, Goldstein D, Tavakoli A, Trinh T, Klisser J, Lewis CR, Friedlander M, Naduvilath TJ, Au K, Park SB, Krishnan AV, Markoulli M. *Corneal dendritic cells and the subbasal nerve plexus following neurotoxic treatment with oxaliplatin or paclitaxel*. *Sci Rep*; 2021. 11(1). doi: 10.1038/s41598-021-02439-0. PMID: 34819589; PMCID: PMC8613280.
7. Liu J, Li Z. *Resident Innate Immune Cells in the Cornea*. *Front Immunol*. 2021. doi: 10.3389/fimmu.2021.620284. PMID: 33717118; PMCID: PMC7953153.
8. Giuseppe Scuderi. *La cornea*. Masson; 1998. 59-70 p.
9. Riordan-Eva P. *Anatomy & Embryology of the Eye*. In: Riordan-Eva P, Augsburger JJ. eds. *Vaughan & Asbury's General Ophthalmology. 19e*: McGraw Hill; 2017. <https://accessmedicine.mhmedical.com/content.aspx?bookid=2186§ionid=165515771>
10. Del Monte DW, Kim T. *Anatomy and physiology of the cornea*. *J Cataract Refract Surg*. 2011. 37(3): 588-98 p. doi: 10.1016/j.jcrs.2010.12.037. PMID: 21333881.
11. Kokot J, Wylęgała A, Wowra B, Wójcik Ł, Dobrowolski D, Wylęgała E. *Corneal confocal sub-basal nerve plexus evaluation: a review*. *Acta Ophthalmol*; 2018. 96(3): 232-242 p. doi: 10.1111/aos.13518. Epub 2017. PMID: 28741902.
12. Machetta F, Fea AM, Actis AG, de Sanctis U, Dalmaso P, Grignolo FM. *In vivo confocal microscopic evaluation of corneal langerhans cells in dry eye patients*. *Open Ophthalmol J*; 2014. 8: 51-9 p. doi: 10.2174/1874364101408010051. PMID: 25317216; PMCID: PMC4195179.

13. Yamaguchi T, Hamrah P, Shimazaki J. *Bilateral Alterations in Corneal Nerves, Dendritic Cells, and Tear Cytokine Levels in Ocular Surface Disease*. *Cornea*; 2016. 35 Suppl 1: S65-S70. doi: 10.1097/ICO.0000000000000989. PMID: 27617877; PMCID: PMC5658781.
14. N. K. Chauhan and K. Singh, *A Review on Conventional Machine Learning vs Deep Learning*. International Conference on Computing, Power and Communication Technologies (GUCON); 2018. 347-352 p. , doi: 10.1109/GUCON.2018.8675097.
15. Goodfellow, Ian. *Deep learning*. Massachusetts: Cambridge; 2016.
16. K.C. Santosh, Nibaran Das, Swarnendu Ghosh. *Deep learning models for medical imaging*. San Diego: Elsevier Science & Technology; 2021. <https://doi.org/10.1016/C2020-0-00344-0>
17. Sharma S., Sharma S., *Activation functions in neural networks*, International Journal of Engineering Applied Sciences and Technology; 2020. Vol. 4, Issue 12, ISSN No. 2455-2143, 310-316 p. <http://www.ijeast.com>
18. M. Bishop. *Pattern Recognition and Machine Learning*. Springer; 2006.
19. Yingjie Tian, Duo Su, Stanislao Lauria, Xiaohui Liu, *Recent advances on loss functions in deep learning for computer vision*. *Neurocomputing*; 2022. 497: 129-158 p. ISSN 0925-2312. <https://doi.org/10.1016/j.neucom.2022.04.127>
20. S. Jadon, *A survey of loss functions for semantic segmentation*. IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB); 2020. 1-7 p. doi: 10.1109/CIBCB48159.2020.9277638.
21. Zhi Liu, Yankun Cao, Yujun Li, Xiaoyan Xiao, Qingchen Qiu, Meijun Yang et al. *Automatic diagnosis of fungal keratitis using data augmentation and image fusion with deep convolutional neural network*. *Computer Methods and Programs in Biomedicine*; 2020. <https://doi.org/10.1016/j.cmpb.2019.105019>.
22. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, *Journal of Machine Learning Research*; 2014. 15, 56, 1929-1958 p. <http://jmlr.org/papers/v15/srivastava14a.html>
23. Ioffe, Sergey, Szegedy, Christian, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. arXiv; 2015. <https://arxiv.org/abs/1502.03167>
24. Yamashita R., Nishio M., Do R.K.G., Togashi K. *Convolutional neural networks: an overview and application in radiology*. *Insights Imaging* 9; 2018. 611–629 p. <https://doi.org/10.1007/s13244-018-0639-9>

25. S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, D. Terzopoulos, *Image Segmentation Using Deep Learning: A Survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence; 2022. 44, 7: 3523-3542 p. doi: 10.1109/TPAMI.2021.3059968.
26. V. Badrinarayanan, A. Kendall, R. Cipolla. *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence; 2017. 39, 12, 2481-2495 p. doi: 10.1109/TPAMI.2016.2644615.
27. Walter Hugo, Lopez Pinaya, Sandra Vieira, Rafael Garcia-Dias, Andrea Mechelli. *Autoencoders*. Academic Press; 2020. 11: 193-208 p. <https://doi.org/10.1016/B978-0-12-815739-8.00011-0>
28. Olaf Ronneberger, Philipp Fischer, Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Switzerland: Springer; 2015. 234-241 p.
29. Zhi Liu, Yankun Cao, Yujun Li, Xiaoyan Xiao, Qingchen Qiu, Meijun Yang et al. *Automatic diagnosis of fungal keratitis using data augmentation and image fusion with deep convolutional neural network*. Computer Methods and Programs in Biomedicine; 2020. <https://doi.org/10.1016/j.cmpb.2019.105019>
30. Williams BM, Borroni D, Liu R, Zhao Y, Zhang J, Lim J et al. *An artificial intelligence-based deep learning algorithm for the diagnosis of diabetic neuropathy using corneal confocal microscopy: a development and validation study*. Epub; 2019. PMID: 31720728; PMCID: PMC6946763.
31. Alessia Colonna, Fabio Scarpa, Riccardo Zorzan, Cecilia Chao, Rabia Mobeen, Blanka Golebiowski et al. *Automatic identification of corneal epithelial dendritic cells via deep learning*. Invest. Ophthalmol. Vis. Sci. ; 2020. 61(7): 4309.
32. 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.; 2016. 5(5), 7. doi: 10.1167/tvst.5.5.7. PMID: 27730007; PMCID: PMC5054765.
33. Zorzan R. *Sviluppo di un algoritmo basato su tecniche di deep learning per la segmentazione di cellule dendritiche in immagini della cornea*. Tesi di laurea in Bioingegneria, DEI, Università di Padova; 2020.

