Università degli studi di Padova

Dipartimento di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Ingegneria Biomedica

Tesi di Laurea Magistrale

Sviluppo di un algoritmo per il rilevamento del disco ottico su immagini del fondo oculare

Relatore:
Prof. Fabio Scarpa

Correlatrici:
Dott.ssa Chiara Rui
Dott.ssa Silvia Gazzina

Laureanda:
Elena Marzolla
Matricola n. 2023524

Anno accademico 2022-2023
13 Aprile 2023

Indice

Abstract

L'esame del fondo oculare, definito anche oftalmoscopia o fondoscopia, è un test che consente allo specialista di vedere le strutture situate nella porzione posteriore dell'occhio, tra cui la retina, la papilla del nervo ottico e il corpo vitreo. Questo può avvenire manualmente con l'utilizzo di lenti apposite o tramite l'utilizzo di fundus camere/oftalmoscopi. Lo stato fisico della retina determina infatti la qualità della vista e di conseguenza la qualità della vita, è per questo motivo che è importante prendersene cura. Le tecnologie oftalmologiche sono in continua evoluzione, i dispositivi emergenti sono in grado di acquisire in tempo reale e con una maggior qualità immagini del fondo oculare, permettendo ai medici di diagnosticare in tempi molto rapidi l'insorgere di una patologia pericolosa per la vista del paziente, come la retinopatia diabetica, una delle principali cause di cecità nei pazienti diabetici.

Nell'ultimo secolo il settore oftalmologico ha visto lo sviluppo di molte tecniche e modalità di imaging, come l'imaging stereoscopico e l'imaging *ultrawide field*. Nonostante ciò, queste tecniche hanno bisogno dell'interpretazione di un esperto clinico, figura che è scarsamente presente nell'ambiente sanitario. Il *deep learning* è una delle tecniche computerizzate utilizzate nell'ambito dell'*imaging* medico, che ha permesso la riduzione di risorse umane nella sanità, favorendo un'assistenza sanitaria più efficiente. Abbiamo pensato di utilizzare il *deep learning* per sviluppare algoritmi di analisi di immagini retiniche in grado di assistere i medici durante il processo di diagnosi.

In particolare questo progetto di tesi ha l'obiettivo di creare un algoritmo di *deep learning* che, grazie all'impiego di reti neurali convoluzionali, sia in grado di rilevare la presenza o l'assenza del disco ottico su immagini del fondo oculare e, in caso di presenza, di individuare il centro del disco. L'algoritmo consentirà di automatizzare l'individuazione della presenza del disco ottico all'interno di immagini retiniche e di individuarne il centro. Questa fase automatica potrà essere utilizzata come pre-elaborazione per sistemi di AI volti a valutare il disco ottico.

Nel primo capitolo si dà una breve ma accurata introduzione sull'anatomia dell'occhio, un insieme di informazioni di base che consentirà di comprendere meglio come funziona il sistema visivo umano; il capitolo prosegue con la descrizione dell'immagine retinica e le principali tecniche utilizzate nell'*imaging* retinico. Il secondo capitolo invece si focalizza sul *deep learning*, descrivendo le sue funzionalità e caratteristiche, come esso viene implementato e le conoscenze di base per la progettazione e la realizzazione di una rete neurale convoluzionale. Il lavoro si conclude con una descrizione degli algoritmi di *transfer learning* che sono stati usati per la realizzazione della rete neurale convoluzionale. I due capitoli finali sono stati dedicati alla descrizione dello sviluppo e dei risultati delle due reti neurali convoluzionali che sono state create per questo lavoro di tesi. La prima delle due reti si occupa di classificare le immagini valutando la presenza e l'assenza del disco ottico, mentre la seconda si occupa invece di determinare il centro del disco nelle immagini che lo contengono.

Capitolo 1

Introduzione

1.1 Anatomia dell'occhio

L'occhio, o bulbo oculare è l'organo adibito alla vista, esso ha il compito di ricavare informazioni sull'ambiente circostante attraverso la luce. Esso è collocato nella porzione anteriore della cavità orbitaria del cranio, che lo contiene e lo protegge. L'occhio è avvolto da tre membrane chiamate tonache che hanno struttura e funzioni diverse:

- La più esterna è la *tonaca fibrosa*, che protegge e sostiene il bulbo oculare e comprendere la sclera (parte bianca dell'occhio) e la cornea, che permette il passaggio dei raggi luminosi verso le strutture interne dell'occhio e concorre a mettere a fuoco le immagini sulla retina.
- La tonaca intermedia è la *tonaca vascolare*, detta anche uvea, essa è una struttura ricca di vasi e di pigmento con la funzione di dare nutrimento alla retina esterna e di assorbire la luce.
- La più interna è la *tonaca nervosa*, o più comunemente retina, essa è la parte sensoriale che trasforma gli stimoli luminosi in impulsi nervosi.

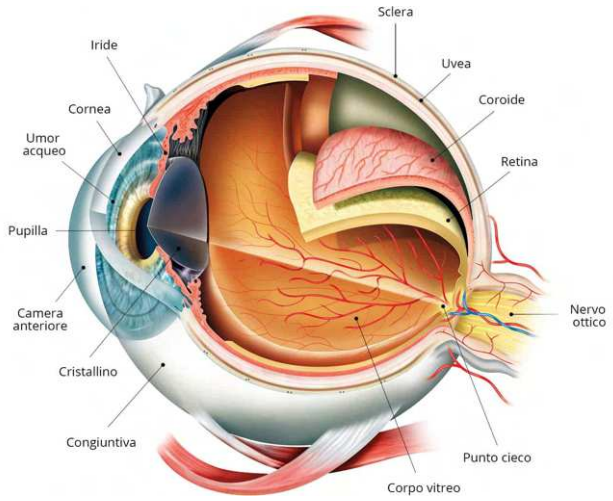


Figura 1.1: Anatomia dell'occhio

La parte esterna è costituita anteriormente dalla cornea e posteriormente dalla sclera. La *sclera* è una membrana fibrosa opaca, che riveste quasi completamente l'occhio. Essa è formata da fibre di collagene ed elastina che le conferiscono la massima resistenza in caso di sollecitazioni meccaniche esterne. La sclera ha quindi funzione di protezione del contenuto oculare.

La *cornea* è una membrana che riveste la parte anteriore dell'occhio esterno aderendo alla sclera. Come abbiamo detto essa funge da lente dell'occhio, è ricca di terminazioni nervose ed è fondamentale per la rifrazione poiché è trasparente, priva di vasi ed ha una struttura concavo-convessa. Inoltre è una struttura che viene continuamente bagnata dal film lacrimale per mantenerne l'idratazione.

Posteriormente alla cornea troviamo l'*iride*, che contiene l'umor acqueo, un liquido salino che bagna e nutre il cristallino e la cornea. L'iride ha una struttura pigmentata, da essa ne dipende il colore dell'occhio. Essa è un disco circolare con un foro centrale chiamato pupilla che, variando il suo diametro grazie a dei muscoli, regola la quantità di raggi luminosi che entrano dentro l'occhio: al sole la pupilla si restringe per proteggere l'occhio dall'eccessiva esposizione alla luce, in penombra la pupilla si allarge per far entrare una maggior quantità di luce per una corretta visione.

Posteriormente all'iride troviamo il *cristallino*, esso è una lente biconvessa, elastica e trasparente che ha la funzione di far convergere e mettere a fuoco sulla retina i raggi provenienti dall'esterno. permettendo la visione di oggetti a diverse distanze.

Tra il cristallino e la retina troviamo il *corpo vitreo* che è un tessuto connettivo, gelatinoso e trasparente che occupa la cavità del globo oculare. É costituito dal 99% d'acqua, la sua viscosità è data dalla presenza di acido ialuronico ed aderisce perfettamente alla retina. Ha diverse funzioni come quella meccanica, infatti, conferisce tono e stabilità al bulbo oculare, e funzione metabolica, ovvero come riserva di metaboliti per i tessuti circostanti e come deposito di scarti.

1.2 La retina

La **retina** o tonaca nervosa, è la membrana più interna dell'occhio. Ha una struttura delicata che contiene i fotorecettori, che sono due tipi di cellule sensibili alle onde luminose: i bastoncelli sono coinvolti nella visione monocromatica in condizioni di luce soffusa o crepuscolare; i coni sono invece responsabili della visione a colori, ma sono attivi soltanto quando la luce è intensa (visione diurna)[Figura 1.2].

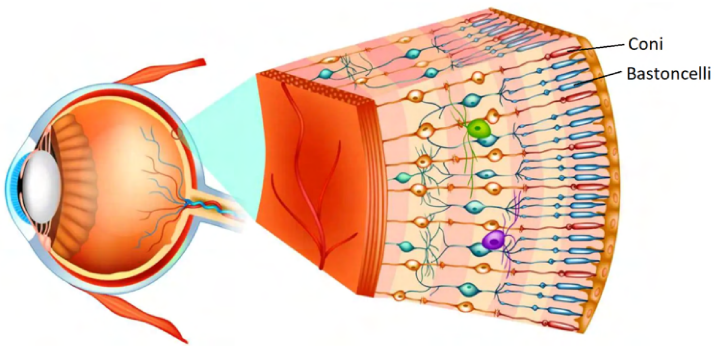


Figura 1.2: Struttura della retina

Al centro della retina troviamo una piccola regione, la *macula*, sensibile alla luce e responsabile della visione nitida e dettagliata. La macula è la regione con più elevata densità di fotorecettori e per questo deputata alla visione distinta e al riconoscimento dei colori. Essa è molto delicata, per questo motivo risulta particolarmente vulnerabile a fenomeni patologici e degenerativi.

Al centro della macula troviamo la *fovea*, essa è una lieve depressione di forma circolare che rappresenta la zona di maggior acuità visiva, in questa regione troviamo la massima concentrazione di coni mentre sono del tutto assenti i bastoncelli.

La retina è ricca di vasi e terminazioni nervose e possiamo dire che ha una funzione di fototrasduttore poiché converte gli stimoli luminosi in segnali bioelettrici, che a loro volta vengono inviati al cervello attraverso le fibre del nervo ottico.

Il *nervo ottico* è considerato una parte del sistema nervoso centrale e costituisce il secondo delle dodici paia di nervi cranici. Esso è un prolungamento delle terminazioni nervose dei fotorecettori della retina, infatti il suo compito è proprio quello di trasmettere al cervello gli impulsi elettrici generati in corrispondenza della retina, permettendo di conseguenza la percezione visiva.

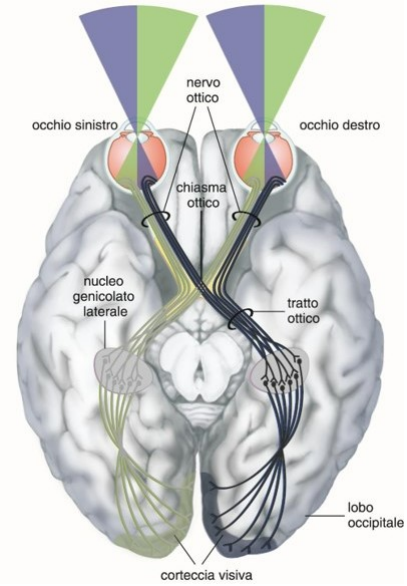


Figura 1.3: Chiasma ottico

Il nervo ottico è rivestito dalle meningi, tre membrane laminari cocentriche con funzione protettiva e che a loro volta avvolgono l'intero cervello. Ogni singola fibra corrisponde a una piccola zona della retina, mentre ogni fascio corrisponde a un'intera area retinica. Dopo essere uscito dal bulbo oculare a livello del disco ottico, il nervo ottico si estende per una lunghezza di circa 5 cm, successivamente il nervo ottico destro e sinistro si incontrano e si suddividono nuovamente, iniziando il tratto chiamato *chiasma*(porzione intracranica) [Figura 1.3].
Nelle immagini retiniche quello che possiamo vedere del nervo ottico è solamente la sua estremità, che viene chiamata disco ottico (o papilla ottica). In queste immagini il disco ottico appare come una piccola area ovale con maggiore diametro verticale, di colore giallo-rosa, per via della sua costituzione fatta da assoni mielinizzati in procinto di uscire dal globo oculare. Il disco ottico si posiziona nella zona bassa rispetto al polo posteriore

dell'occhio, ad una distamza di circa 4 millimetri dalla macula. Dal centro del disco ottico, affiorano i vasi ematici dell'arteria centrale della retina e la vena omonima che irrorano l'intero occhio.

1.3 L'imaging retinico

L'**imaging retinico** è l'integrazione di tutti mezzi che portano all' "informazione visiva" retinica, essa comprende sia la parte hardware e sia la parte software ad essa associata, i report e l'AI che può essere utilizzata. L'imaging retinico ci permette di acquisire in modo non invasivo un insieme d'immagini fotografiche che consentono di analizzare caratteristiche metaboliche, morfologiche e strutturali della retina, al fine di diagnosticare e monitorare patologie retiniche ereditarie e/o acquisiste. Esistono diverse tecniche che ci permettono di fare queste acquisizioni, ciascuna con i suoi vantaggi e svantaggi, per questo motivo molto spesso si ricorre all'*imaging multimodale*, ovvero all'integrazione di due o più metodi di imaging per migliorare la capacità di diagnosi, di prognosi e di terapia.

1.3.1 Fundus Oculi

Uno dei metodi che viene utilizzato per studiare la cavità vitreale del bulbo oculare è il *Fundus Oculi*. Esso consiste nell'utilizzo di una *Fundus Camera* che permette di valutare la presenza di malattie della retina e del nervo ottico; tra le più frequenti troviamo la retinopatia diabetica, la degenerazione maculare legata all'età, il distacco di retina e, in casi meno frequenti, l'influenza di malattie più serie come il tumore al cervello.

Le immagini retiniche si presentano come un'immagine abbastanza semplice dal punto di vista anatomico. Nella Figura 1.4 riusciamo a riconoscere chiaramente il disco ottico, la fovea e i vasi retinici. Chiaramente l'individuazione delle componenti anatomiche è più complessa nei pazienti che presentano patologie come quelle già citate precedentemente.

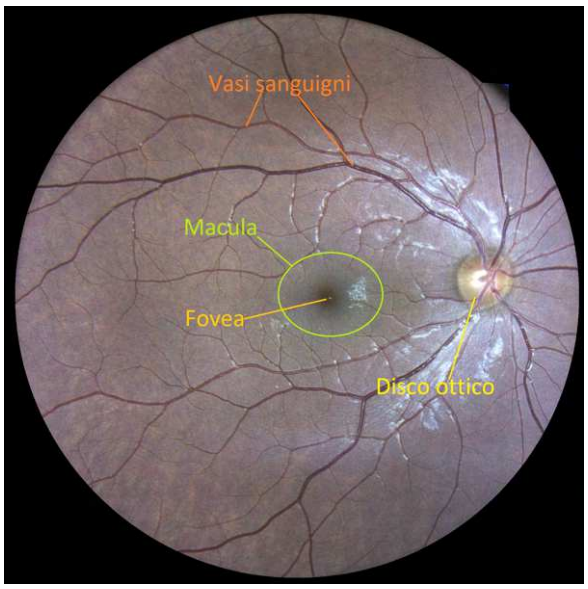


Figura 1.4: Immagine della retina di un paziente sano

L'esame del fundus oculi è di tipo non invasivo, durante il quale al paziente viene chiesto di spostare lo sguardo in varie posizioni, in modo da avere un quadro completo del segmento posteriore dell'occhio. Per facilitare l'esecuzione dell'esame al paziente vengono somministrati alcuni specifici colliri che inducono la dilatazione della pupilla, anche se i dispositivi di nuova generazione riescono ad ottenere un'ottima visualizzazione senza l'utilizzo di colliri. Durante l'esame il paziente non avverte dolore, ma un fastidio dovuto alla sensibilità alla luce. La durata dell'esame varia a seconda delle patologie riscontrate o dalle modalità impiegate per evidenziare il fondo oculare.

Figura 1.5: Esecuzione dell'esame Fundus Oculi

1.3.2 L'OCT

La Tomografia ottica compiuterizzata (OCT), o Tomografia ottica a radiazione coerente, è una tecnica diagnostica di tipo non invasivo. Essa permette di ottenere delle scansioni corneali e retiniche molto precise, in particolare ci permette di analizzare gli strati della cornea, la macula (parte centrale della retina) ed il nervo ottico. Lo scopo è quello di andare a monitorare e diagnosticare numerose patologie corneali e retiniche come la degenerazione maculare senile, la retinopatia diabetica ed il glaucoma; molto utilizzata anche in presenza di edema maculare di diversa origine. Inoltre, l'OCT ha un ruolo essenziale in fase preoperatoria e postoperatoria nelle patologie che necessitano di un intervento chirurgico. Le immagini che vengono ottenute da questo esame sono come le seguenti.

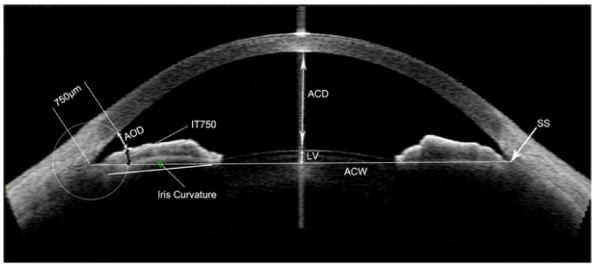


Figura 1.6: Cornea di un paziente sano acquisita con OCT

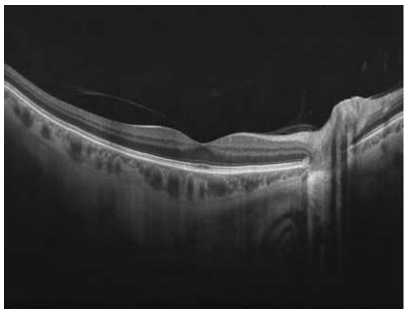


Figura 1.7: Macula di un paziente sano acquisita con OCT

L'OCT è basata sull'interferometria a luce bianca o a bassa coerenza, un fascio laser privo di radiazioni nocive che viene impiegato per analizzare le strutture oculari soprattutto retiniche mediante sezioni ad alta risoluzione. Il paziente viene posizionato davanti lo strumento e gli viene chiesto di fissare una mira luminosa, la scansione inizia dalla messa a fuoco della struttura oculare da analizzare. L'esecuzione è semplice e veloce, con una durata di circa 10-15 minuti. Generalmente richiede la presenza di dilatazione delle

Figura 1.8: Esecuzione di una OCT

pupilla, anche se esistono dispositivi di ultima generazione che non necessitano di questo requisito, previa valutazione da parte del clinico. In un certo numero di condizioni l'OCT permette di sostituire la fluorangiografia, evitando al paziente di sottoporsi ad un esame invasivo.

Possiamo riassumenre i principali vantaggi dell'OCT nei seguenti punti:

- pochi artefatti da movimento;
- visualizzazione diretta della morfologia dei tessuti;
- nessuna necessità di traumatismi;
- nessuna necessità di preparazione delle strutture interne;
- immagini ad elevata risoluzione;
- immagini di sezione con visualizzazione della struttura interna;

- possibilità di ottenere misurazioni oggettive e ripetibili;
- nessuna radiazione ionizzante.

1.3.3 Angiografia oculare

Esistono numerose patologie retiniche che interessano la vascolarizzazione della retina, una di queste è sicuramente il diabete. Nei pazienti affetti da queste patologie è indispensabile conosce con precisione la salute della struttura vascolare retinica e sottoretinica prima di intraprendere un trattamento. L'angiografia oculare è un esame utilizzato per studiare il sistema vascolare della retina e della coroide. L'oculista ricorre a questo tipo di procedura quando non riesce a valutare ad occhio nudo il istema vascolare dell'occhio. Questo esame è minimamente invasivo poiché utilizza un mezzo di contrasto, parliamo quindi dell'*angiografia a fluorescenza* o con *fluoresceina* (*FAG*) o con *verde indocianina* (*ICGA*). La fluoresceina è un mezzo di contrasto organico, quindi ha una tollerabilità elevata, esso viene iniettato in una certa quantità nella vena del braccio e dopo circa 20-25 secondi (tempo retina-braccio) raggiunge la retina. Attraverso un agiografo digitale siamo in grado di lanciare una sorta di lampo/flash e di registrare il passaggio del contrasto attraverso il sistema vascolare della retina.

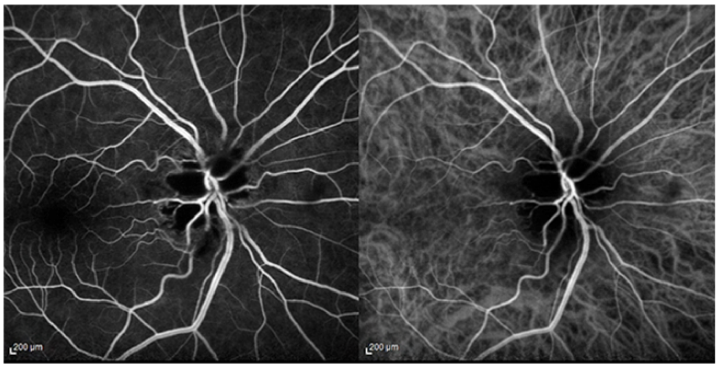


Figura 1.9: Esempio di FAG (a sinistra) e ICGA (a destra)

Tipicamente gli oftalmologi scattano diverse fotografie delle rete vascolare dell'occhio man mano che i vasi si riempioni della sostanza di contrasto, questo perché valutando la tempistica di riempimento dei vasi, possono individuare delle patologie retiniche. Per studiare i vasi che si trovano al di sotto della retina, nella coroide, è necessario cambiare colorante ed utilizzare il verde indocianina. Anche quest'ultimo è un mezzo di contrasto organico e quindi difficilmente causa intolleranze e grazie al quale riusciamo a visualizzare i vasi più profondi, quelli coroideali. Questa diffusione viene usata anche come diagnosi anticipata delle membrane neovascolari di tipo occulto, ovvero di membrane non ancora manifestate ma che hanno una certa probabilità di manifestarsi negli anni.

Nonostante la elevata tollerabilità dei mezzi di contrasto citati è possibile che il paziente mostri delle intollerabilità verso il contrasto, per questo l'esame è controindicato per coloro che sono allergici alla fluoresceina e per le donne in gravidanza. Seppur sia un esame molto comune ed efficiente il mezzo di contrasto è una limitazione per questo esame, di conseguenza negli anni si è sviluppata l'*angiografia OCT* (*Angio-OCT*), che ha proprio la caratteristica di non utilizzare alcun tipo di contrasto.

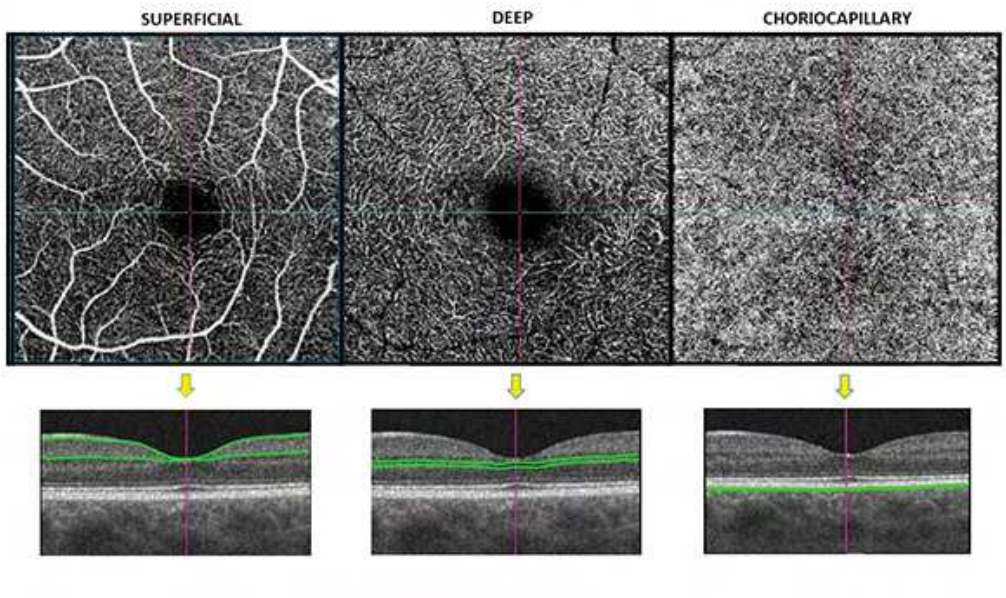


Figura 1.10: Esempio di immagine acquisita con Angio-OCT a diversa profondità

L'Angio-OCT è un esame non invasivo, che sfrutta la tecnologia della tomografia a coerenza ottica per visualizzare i grandi vasi sanguigni e anche i capillari, permette quindi al medico di eseguire l'angiografia della coroide, della retina e del nervo ottico senza l'uso di un mezzo di contrasto. I dispositivi di ultima generazione consentono di mostrare un'immagine ad alta risoluzione anche in tre dimensioni.

1.3.4 Scopo della tesi

Lo scopo della tesi è la realizzazione di reti neurali convoluzionali che siano in grado di analizzare le immagini del fondo oculare che le vengono date in ingresso, più in particolare le reti dovranno essere in grado di riconoscere la presenza e l'assenza del disco ottico in immagini retiniche a colori. La rete si occuperà quindi di risolvere un problema di classificazione delle immagini nelle due classi, Presenza e Assenza. La tesi comprende anche la realizzazione di una seconda rete, che ha lo scopo di individuare il centro del disco ottico in immagini del fondo oculare che lo contengono. Nella realizzazione delle reti andremo a utilizzare le principali tecniche di *deep learning* che ci consentiranno di eseguire i *task* in modo automatico. Ci serviremo inoltre di reti pre-addestrate presenti in letteratura, che offrono un metodo rapido ed efficace alla realizzazione da zero della rete.

In generale la valutazione delle condizioni del nervo ottico è un punto di partenza per lo specialista nella diagnosi di una patologia o nel valutare il decorso di una malattia che

interessa la retina e non solo. Il settore oftalmologico è uno dei rami della medicina che ha assistito ad un forte sviluppo di molte tecniche e modalità di imaging negli ultimi decenni. Nonostante questo, il clinico specialista resta una figura di rilevanza in fase di diagnosi che si trova con scarsità negli ambienti sanitari. Lo sviluppo di tecnologie che utilizzano l'intelligenza artificiale (AI) ha favorito un'assistenza sanitaria più efficiente, colmando in parte la scarsità di personale. Tecniche di *deep learning* hanno dimostrato negli anni un forte potenziale nel risolvere in modo automatico diverse tipologie di *task*, in particolare migliorano la capacità di classificare, riconoscere, rilevare e descrivere utilizzando i dati. É per questo motivo che per risolvere questo *task* di rilevazione del disco ottico abbiamo preso in considerazione l'utilizzo di tecniche di *Deep learning*, in particolare la creazione di due CNN opportunamente costruite per la risoluzione del nostro *task* in modo più accurato e veloce delle tecniche tradizionali.

Capitolo 2

Reti Neurali

2.1 Elementi base: Neuroni e Funzioni di attivazione

Le **reti neurali artificiali** (ANN, artificial neural network) sono dei modelli computazionali che vengono utilizzati nel campo dell'intelligenza artificiale (AI), *machine learning* e *deep learning* per risolvere problemi di diversa natura. I principali campi in cui vengono utilizzate le reti neurali sono: la finanza (con numerosi applicazioni), riconoscimento ed elaborazione delle immagini, analisi del parlato e riconoscimento vocale, simulazione di sistemi biologici, diagnosi mediche, inclusi i referti di TAC e risonanza magnetica, *robot steering*, controllo di qualità su scala industriale, *data mining* e simulazioni di varia natura. Esse simulano il comportamento del cervello umano, ovvero una rete neurale biologica, sono formate infatti da "neuroni" artificiali.
I neuroni che compongono il tessuto nervoso sono caratterizzati da: dendriti, sinapsi e assoni. Il neurone riceve il segnale attraverso i dendriti, che sono delle ramificazioni di forma tubolare. Viene quindi mandato un impulso elettrico lungo l'assone che, grazie alla sua forma permette di distribuire l'informazione in diverse destinazioni nel medesimo istante. L'assone è la struttura fondamentale della conduzione dell'impulso. L'impulso arriva quindi alle sinapsi che sono responsabili del trasferimento dell'impulso nervoso. L'apprendimento avviene quando le sinapsi trasmettono i segnali da un neurone all'altro.

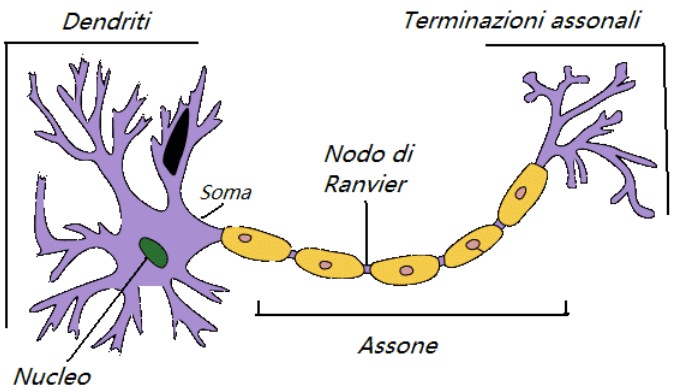


Figura 2.1: Struttura del neurone

I neuroni delle ANN sono detti anche nodi, essi sono delle unità che, proprio come i neuroni biologici, sono interconnessi tra loro e ad ogni connessione viene associato un peso modificabile, che simula l'attività delle sinapsi, e una soglia. La soglia funge da sbarramento per la propagazione dell'output, che viene inviato al nodo del livello successivo solamente se l'output supera la soglia. In caso contrario, non viene passato alcun dato al livello successivo. La struttura base di una rete neurale è costituita generalmente da tre layer principali:

- **Input layer**: che riceve in ingresso i dati numerici che costituiscono gli input della rete;
- **Hidden layer**: chiamati anche livelli nascosti che vengono utilizzati in fase di apprendimento della rete;
- **Output layer**: che fornisce in uscita i risultati predetti dal modello.

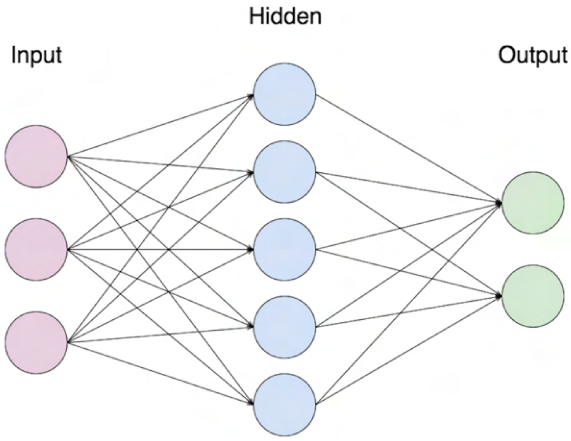


Figura 2.2: Struttura base di una rete neurale

Ad ogni singolo nodo l'output viene calcolato come un modello di regressione lineare, composto da dati di input, pesi, una distorsione(o soglia, il *bias*) e un output. I pesi w_i aiutano a determinare l'importanza o meno di una qualsiasi variabile x_i: pesi più grandi rendono la variabile più significativa nel determinare l'output, viceversa pesi più piccoli rendono la variabile meno significativa nel determinare l'uscita. L'output viene quindi calcolato come segue

$$a = \sum_{k=1}^{m} w_i \cdot x_i + bias \tag{2.1}$$

Successivamente, l'output viene passato attraverso una funzione di attivazione, che determina l'output vero e proprio. Se supera una determinata soglia, tale output attiva il nodo, passando i dati al livello successivo nella rete. L'output di un nodo diventa quindi l'input del nodo successivo. Questo processo di passaggio dei dati da un livello a quello successivo definisce la rete neurale una rete *feedforward*. L'algoritimo di apprendimento si basa sull'aggiornamento dei pesi (w_i) e *bias* con il fine di minimizzare l'errore tra output predetto e output atteso.

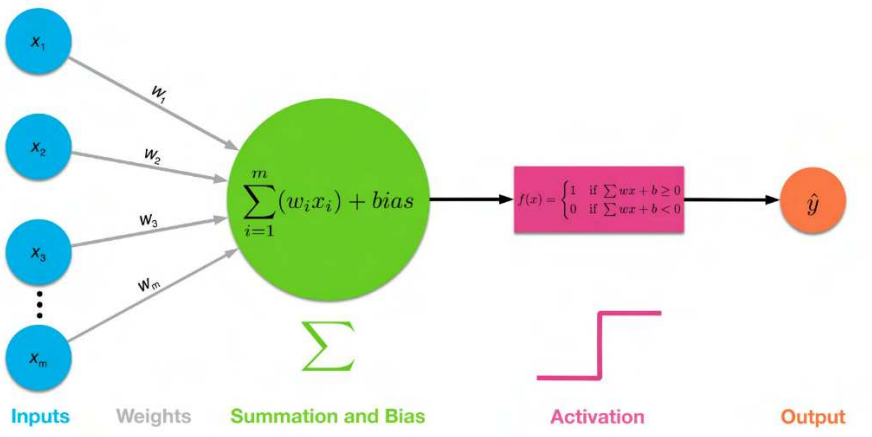


Figura 2.3: Procedura di calcolo di un singolo layer

La funzione di attivazione, *Activation Function*, è una funzione matematica che determina il valore del segnale di uscita, il valore in ingresso è il numero ottenuto dalla sommatoria precedente. La funzione è applicata ad ogni singolo neurone della rete e stabilisce se deve essere attivato o meno, in altre parole, se l'informazione deve essere passata al *layer* successivo. La funzione di attivazione ci permette di simulare il comportamento binario dei neuroni biologici, in quanto l'impulso si propaga solamente al superamento di un certo livello di eccitazione interno.
I neuroni biologici hanno un'attivazione binaria, ma nelle reti neurali artificiali si è visto che è più utile usare funzioni di attivazione come quelle qui riportate (nessuna della quali è binaria).

- **Sigmoid** è una delle funzioni di attivazione più utilizzate, soprattutto per il calcolo delle probabilità poiché ha un codominio compreso tra 0 e 1. É una funzione differenziabile e monotona e viene definita nel seguente modo

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

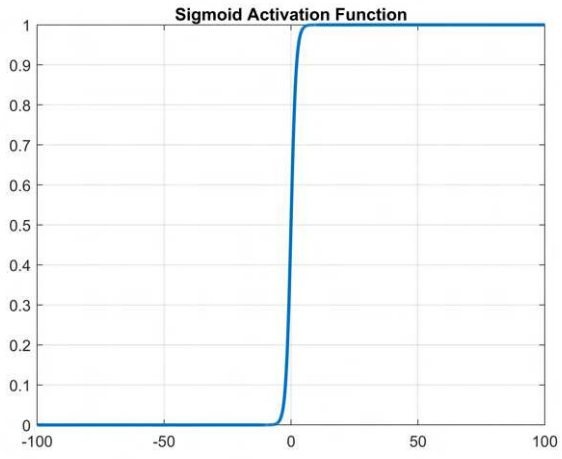


Figura 2.4: Sigmoid activation function

Trova buone applicazioni nel controllo dei segnali, funzioni logiche (restituisce previsioni molto chiare per la classificazione binaria), nelle reti LSTM e nel controllo non lineare, non adatta alle reti di immagini dove viene sostituita dalla ReLU, può causare il problema del gradiente in fuga, non è centrata attorno allo zero ed è computazionalmente costosa.

- **Tangente iperbolica** è una funzione di attivazione molto simile alla precedente Sigmoide con la differenza che il suo codominio va da -1 a 1, questa sua caratteristica può essere utile in varie occasioni.

$$\sigma(x) = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{2.3}$$

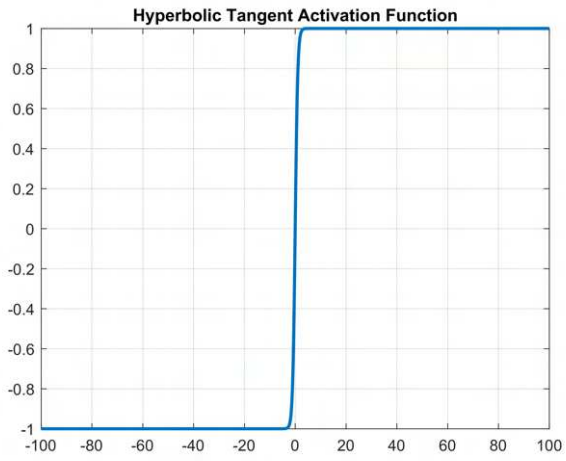


Figura 2.5: Hyperbolic tangent activation function

- **Rectified linear unit (ReLU)** è una funzione di attivazione molto semplice ma considerata la più utilizzata. Essa è definita come

$$f(x) = max(0, x) \tag{2.4}$$

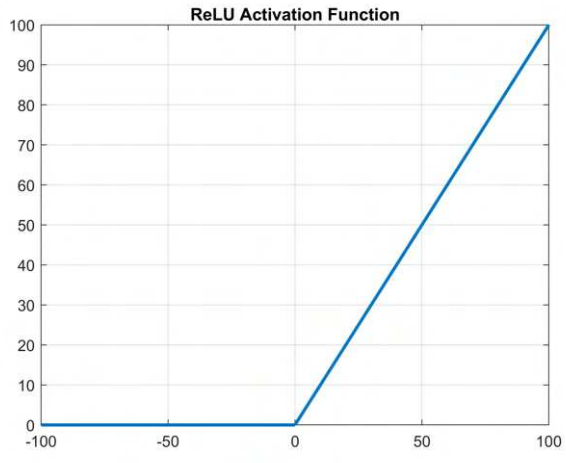


Figura 2.6: ReLU activation function

Questo tipo di funzione di attivazione resta a 0 per i valori negativi, mentre cresce linearmente per i valori positivi. In questo modo non tutti i neuroni sono attivi e questo rende la rete più efficiente a livello computazionale rispetto ad altre funzioni di attivazione. Trova buone applicazioni quando i dati sono delle immagini, ma non è particolarmente indicata per funzioni logiche e controllo nelle reti ricorrenti.

- **Leaky ReLU e PReLU** sono due funzioni di attivazione molto simili alla ReLU, esse hanno una leggera inclinazione per i valori negativi di x, per tale ragione il loro valore medio non è centrato nello zero ma è leggermente slittato a valori positivi, questo slittamento della media nei vari strati della rete crea un rallentamento del processo di apprendimento. Nonostante ciò, sono funzioni di attivazione più veloci di Sigmoid e Tanh. Trovano buona applicazione quando i dati sono delle immagini, vicerversa non sono adatte a funzioni logiche e problemi di controllo. Di seguito le loro formulazioni

$$\begin{aligned} \text{Leaky ReLU:} \quad & f(x) = max(0.01x, x) \\ \text{PReLU:} \quad & f(x) = max(\alpha x, x) \end{aligned} \tag{2.5}$$

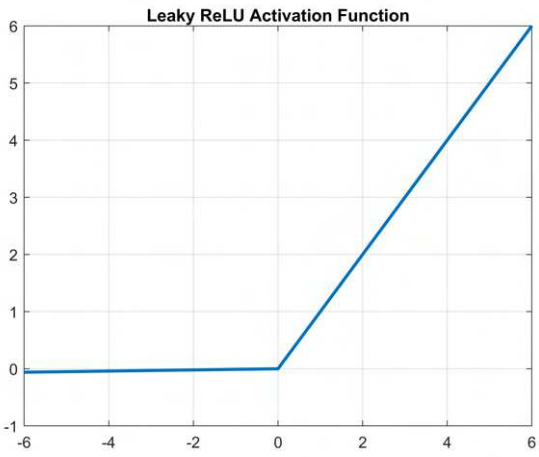


Figura 2.7: Leaky ReLU activation function

- **Softmax** è una generalizzazione della funzione logistica Sigmoide ad una classificazione multi-classe, essa mappa un vettore K-dimensionale x di valori reali arbitrari in un vettore K-dimensionale $\sigma(x)$ di valori compresi tra $(0, 1)$ la cui somma è 1, in poche parole normalizza il vettore iniziale in una distribuzione di probabilità costituita da probabilità K corrispondenti agli esponenziali del numero di input. Softmax si definisce dalla formula

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_i}} \qquad \text{per } i = 1, ..., K \text{ e } z = (x_1, ..., x_K) \in \mathbb{R}^K \tag{2.6}$$

Gli input possono avere valori positivi, negativi, zero o maggiori di 1, se uno degli input è piccolo o negativo, il Softmax lo trasforma in una piccola probabilità, se invece è grande, lo trasforma in una probabilità grande, ma rimarrà sempre compresa tra 0 e 1. Questa funzione di attivazione aiuta il *training* a convergere più velocemente di quanto farebbe altrimenti. Poichè Softmax può essere usata per

la classificazione multi-classe, essa è adatta al *layer* di output, nonostante ciò è computazionalmente costosa in quanto comprende nel suo calcolo molti termini di esponente.

In generale si richiede che le funzioni di attivazione che vengono usate nell'implementazione di una rete neurale, rispettino le caratteristiche di *non linearità*, per garantire la capacità della rete di processare informazioni di input complesse, di *continuità* e di *differenziabilità* per particolari esigenze matematiche di calcolo. La scelta della funzioni di attivazione è quindi strettamente legata al problema che deve affrontare la rete, dalle tipologia di dati e dai requisiti richiesti dal proprio modello, poiché tutte le funzioni di attivazione presentano vantaggi e svantaggi.

2.2 Architettura delle reti

Per architettura di una rete si intende la struttura che la rete neurale assume in termini di numero di neuroni che compongono i vari livelli, la disposizione di tali neuroni nella rete e le connessioni che esistono tra loro. Il modello che viene realizzato e la struttura della rete determinano il suo funzionamento e di conseguenza le *performance* del modello. Il *Deep learning* comprende molteplici algoritmi e differenti architetture di reti neurali applicabili a un grande quantitativo di problemi. É evidente come sia necessario conoscere quale sia l'architettura della rete che meglio si presta alla risoluzione del nostro problema, ed è anche per questo motivo che negli anni sono state sviluppate delle architetture ben distinte, utili ad essere un punto di partenza per la costruzione della rete.

2.2.1 Feed-Forward Neural Networks

La prima architettura creata è la rete neurale *feed-forward*. Essa è composta da uno strato di ingresso, che distribuisce i segnali di input ai neuroni del primo strato, seguito da uno o più strati intermedi, che effettuano una trasformazione non lineare e in fine uno strato di uscita, la cui funzione di attivazione caratterizza le competenze della rete. Per costruzione vengono distinte due tipologie di rete *feed-forward*, la prima è il *percettrone a singolo strato* (SLP), la cui struttura è composta solamente dal *layer* di input e da quello di output, la seconda è il *percettrone multistrato* in cui sono presenti uno o più strati nascosti.

In questa architettura di rete le informazioni fluiscono solamente in una direzione, in avanti, cioè dal primo strato di input via via fino ai nodi di uscita, da questa caratteristica ne deriva il nome della rete. Nella rete non ci sono cicli di *feedback*, infatti le reti *feed-forward* non hanno memoria di input avvenuti a tempi precedenti, per questo motivo l'output è determinato solamente dall'attuale input.

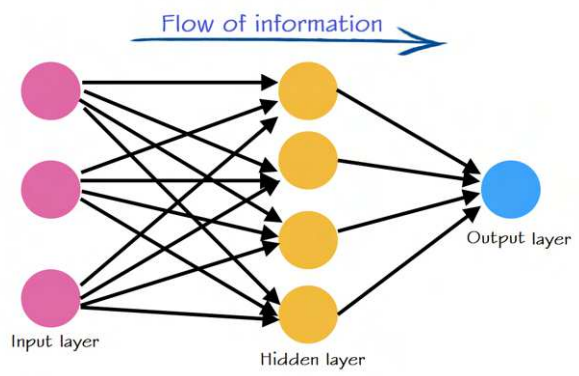


Figura 2.8: Architettura di una *Feed-forward Neural Network*

In questa rete l'output viene calcolato facendo passare in ingresso alla funzione di attivazione ϕ, la somma pesata dei valori di input x_i pesati con i relativi pesi w_i e a cui viene sommato il termine di *bias* b.

$$y_i = \phi\left(\sum_{k=1}^{m} w_i \cdot x_i + b\right) \tag{2.7}$$

Le reti neurali di questo tipo si prestano bene per approssimare una generica funzione f, che prende in ingresso i valori d'ingresso x e i parametri θ e ne restituisce i valori d'uscita $y = f(x;\theta)$. Le reti neurali *feed-forward* sono in grado di apprendere i valori dei parametri θ che meglio approssimano f. Infatti, secondo l'*Universal approximation theorem* le reti *feed-forward*, aventi almeno uno strato intermedio, sono in grado di approssimare qualsiasi funzione continua in $\mathbb{R}^n$.

2.2.2 Recurrent Neural Networks

Le Reti Neurali Ricorrenti o *Recurrent Neural Networks* (RNN) sono reti neurali che a differenza delle *feed-forward*, i neuroni possono ammettere anche dei *loop* e/o possono essere interconnessi anche a neuroni di un livello precedente, quindi il segnale non si propaga solo in avanti, come possiamo vedere dalla Figura 2.9. Per questo motivo i modelli che hanno un'architettura RNN sono molto performanti, in quanto la rete ha una capacità *mnemonica*, ovvero la capacità di tenere memoria dell'esperienza maturata precedentemente ed usarla per prendere decisioni, ecco che entra in gioco il concetto di ricorrenza da cui deriva il nome della rete.

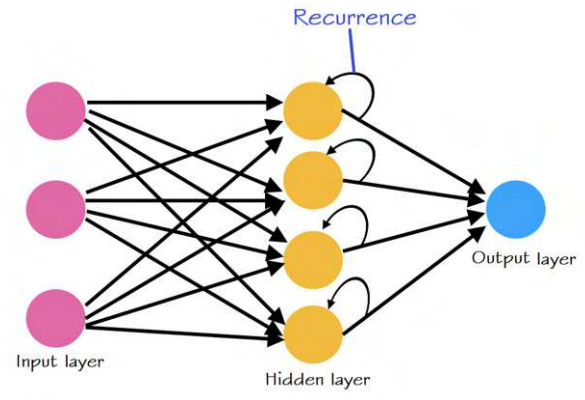


Figura 2.9: Architettura di una *Recurrent Neural Network*

La rete RNN elabora quindi due set di pesi, uno per gli input $x_{(t)}$, che chiameremo W_x, e l'altro per gli output del passo precedente $y_{(t-1)}$, che chiameremo W_y. L'output al tempo t viene quindi calcolato nel seguente modo

$$y_{(t)} = \phi\big(W_x^T \cdot x_{(t)} + W_y^T \cdot y_{(t-1)} + b\big) \tag{2.8}$$

l'input al tempo t e l'output al tempo $(t-1)$ vengono quindi moltiplicati per il relativo set di pesi, alla loro somma aggiungiamo anche il termine di *bias* b, in fine il tutto viene dato in ingresso alla funzione di attivazione ϕ, otteniamo quindi il vettore di output.

Le reti di questo tipo trovano una buona applicazione nei problemi di *Natural Language Processing* (NLP), ovvero in lavori che coinvolgono frasi e/o frammenti di testi, audio e documenti, esse consentono di risolvere problemi come *speech-to-text* (la conversione da voce in testo), *Sentiment Analysis* (estrazione di opinioni dei consumatori, recensioni e commenti social) e *automatic translation*. In fine, sono utilizzante anche in ambito musicale per individuare una sequenza melodica.

2.2.3 Convolutional Neural Network

La Rete Convoluzionale o *Convolutional Neural Network* (CNN) è un tipo di rete *feed-forward*, che si ispira all'organizzazione della corteccia visiva umana, per questo motivo questa tipologia di rete risulta molto diversa dalle altre in termini di architettura e di applicazioni. Questa tipologia di rete infatti lavora soprattutto con immagini, in particolare sono in grado di risolvere problemi di classificazione di immagini e l'identificazione di oggetti in un'immagine. Trova quindi molteplici applicazioni nella guida auto-assistita e il riconoscimento di patologie.

Essendo una rete *feed-forward* essa è costituita da un blocco di input, uno o più blocchi nascosti, nei quali vengono applicate le funzioni di attivazione e in fine dal blocco di uscita.

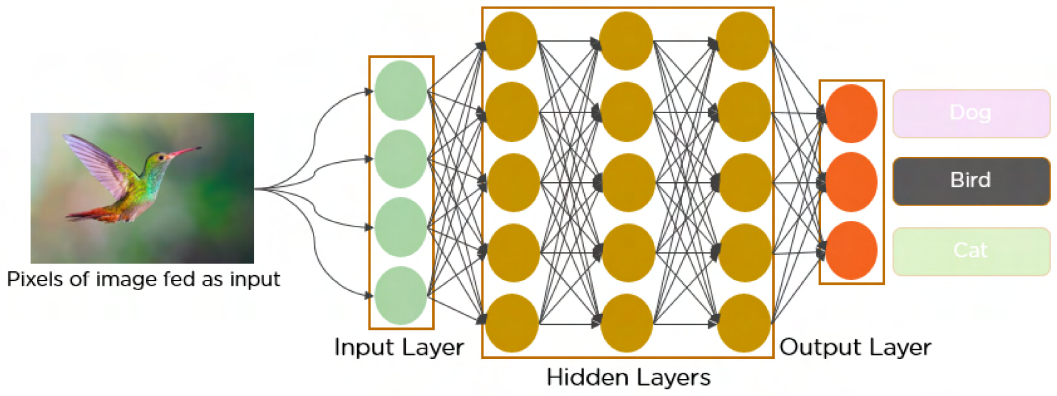


Figura 2.10: CNN applicata al problema di classificazione

Ciò che le differenzia dalle altre tipologie di rete sono i livelli convoluzionali , che hanno la funzione di estrarre le caratteristiche o *features* delle immagini grazie all'utilizzo di filtri che possono variare in numero e dimensione. La presenza dei livelli di convoluzione ci permette di analizzare il contenuto dell'immagini e quindi svolgono un ruolo importantissimo nelle CNN da cui ne deriva il nome.

2.3 Tipologia di layer

Abbiamo già visto come le reti neurali siano formate da uno o più layer nascosti che svolgono diverse funzioni a seconda della loro tipologia e dell'impostazione dei loro parametri interni. Vediamo ora nel dettaglio i layer che maggiormente vegono utilizzati, soprattutto da questo progetto di tesi.

2.3.1 Convolutional

Il layer convoluzionale è lo strato più importante di una rete neurale CNN, come abbiamo già detto, poiché ci permette di estrarre le *features* dell'immagine e di svolgere importanti task.

L'operazione di convoluzione tra due segnali è descritta del seguente modo

$$f(t) * g(t) \triangleq \sum_{\tau=-\infty}^{+\infty} f(\tau) * g(t-\tau) \tag{2.9}$$

Tuttavia, poiché lavoriamo su immagini, la nostra CNN lavorerà nel dominio spaziale anziché nel tempo, per questo l'operazione che viene eseguita è più simile a un prodotto di *cross-correlazione*, nel quale il secondo segnale non viene ribaltato, ma semplicemente traslato, quindi otteniamo

$$f(x,y) * g(x,y) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} f(u,v) * g(x+u, y+v) \tag{2.10}$$

L'operazione di convoluzione che viene applicata coinvolge l'immagine di partenza, sottoforma di una matrice di input, e una matrice di pesi chiamata filtro o *kernel*, di dimensioni inferiori a quelle dell'immagine. Il *kernel* corrisponde al nodo della rete neurale,

ed i valori di ciascun elemento del *kernel* corrispondono ai pesi assegnati alle connessioni. Sfruttando l'operazione di convoluzione, la connessione non è relativa ad un singolo pixel dell'immagine (come succede in una rete neurale standard), ma a tutti.

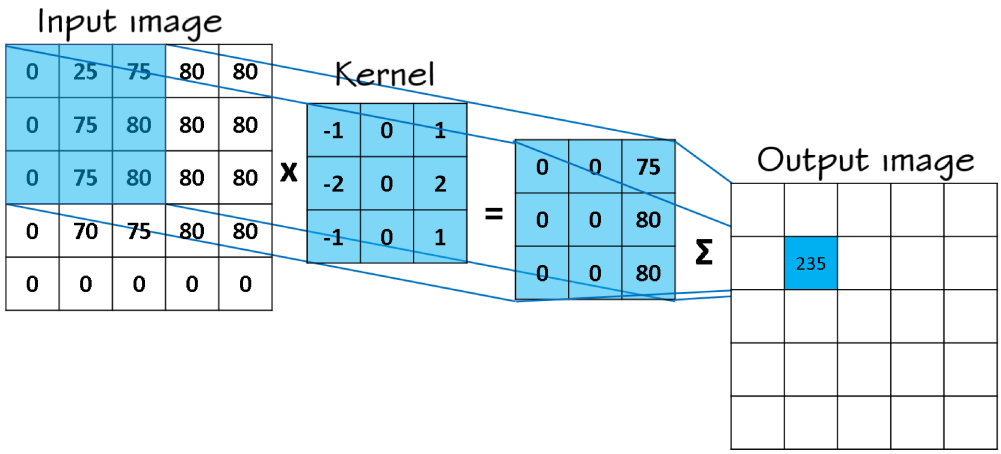


Figura 2.11: Operazione di convoluzione

L'operazione di convoluzione restituisce un'immagine, sotto forma di matrice, in cui i valori dei singoli pixel sono calcolati come somma pesata degli elementi della matrice di input, moltiplicata per i valori contenuti nel filtro. L'operazione viene eseguita per tutti i pixel dell'immagine di partenza facendo traslare il *kernel*, ricoprendo tutta la matrice iniziale. Le dimensioni della matrice finale dipendono da alcuni parametri che definisco il filtro, questi sono chiamati *iperparametri* e sono:

- *size*: definisce le dimensioni del filtro vero e proprio, tipicamente il filtro è scelto di dimensione dispari, in quanto nella convoluzione è importante identificare il centro della matrice kernel, cosa che avviene facilmente con dimensioni dispari, del tipo 3×3, 5×5, 7×7 e così via, in aggiunta le matrici kernel non sono comunemente di grandi dimensioni;

- *stride*: rappresenta la velocità di traslazione del kernel sull'immagine, misurata in pixel orizzontali e verticali, ad esempio uno *stride* 1×1 definisce un filtro che si sposta da sinistra a destra di 1 pixel alla volta e da sopra a sotto di 1 pixel alla volta. Possiamo dire che lo *stride* è il passo di spostamento del filtro, può assumenre un valore minimo di 1 e deve essere tale che il filtro, nel suo spostamento, copra esattamente la dimensione della matrice di input;

- *padding*: rappresenta le dimensioni del bordo esterno applicato all'immagine di input, che si crea quando la posizione del pixel centrale del kernel è occupata da un pixel di bordo dell'immagine. In questo caso i pixel esterni all'area vengono assunti zero, così facendo riusciamo ad applicare il kernel di convoluzione a tutte le posizioni pixel, senza perdere informazione sul contenuto dei bordi dell'immagine. Senza spaziatura interna i kernel vengono spostati solo su posizioni in cui tutti gli input al kernel rientrano ancora all'interno dell'area, di conseguenza la dimensione di output sarà minore della dimensione di input.

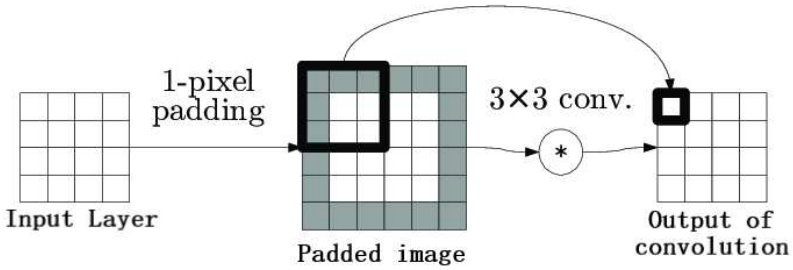


Figura 2.12: Operazione di convoluzione con padding

In generale la formula per il calcolo delle dimensioni $n \times n$ della matrice di output è

$$n = \left\lfloor \frac{m + 2p - f}{s} + 1 \right\rfloor \tag{2.11}$$

dove $m \times m$ è la dimensione della matrice di input, $f \times f$ la dimensione del kernel, s è lo *stride*, p è il *padding* e $\lfloor\ \rfloor$ indica la parte intera del numero risultante.

Per un layer convoluzionale è importante definire il numero di filtri, poiché questo particolare layer è in grado di applicare alla stessa immagine in input un certo numero di filtri contemporaneamente. Tipicamente si sceglie un numero di filtri che aumenta all'aumentare della profondità dei layer, anche se questa non è una vera e propria regola.

2.3.2 Activation

L'*Activation Layer* è il layer dedicato all'applicazione della funzione di attivazione che il costruttore ha scelto di applicare alla rete e le cui tipologie sono state descritte precedentemente nel capitolo 2.1. Tipicamente il blocco di convoluzione comprende anche l'attivazione.

La funzione di attivazione aggiunge non linearità alle trasformazioni eseguite. In assenza delle funzioni di attivazione ci ritroveremo ad applicare funzioni lineari, che difficilmente descrivono la complessità del nostro modello e che quindi non sarebbero in grado di analizzare il contenuto di un'immagine nel nostro caso. Attraverso delle trasformazione non lineari progressive, invece, il livello di astrazione dei dati in ingresso aumenta sino all'ultimo livello, quello di output, che produce una distribuzione di probabilità sulle possibili classi.

2.3.3 Pooling

Il blocco di *Pooling* ha lo scopo di ridurre le dimensioni delle matrici di input, sia in larghezza che in altezza, senza variare la sua profondità (numero di canali). Questo layer segue tipicamente il layer di convoluzione, che fornisce informazioni più pure poiché evidenzia alcune caratteristiche dell'immagie eliminandone altre; il pooling serve dunque ad eliminare la pesantezza di un'immagine di grandi dimensioni, come quella originale e quindi trasmettere ai layer più interni un'immagine alleggerita dalle informazioni inutili ma che

mantiene le caratteristiche principali della stessa. Il *Pooling* utilizza sottocampionamenti di diverso tipo per creare dei *pooling layer* di diversa tipologia.

Il *Max Pooling* è uno dei layer di pooling più utilizzati, nel quale un sottogruppo di valori della matrice di input viene sostituito con il valore massimo contenuto nel sottogruppo. Così facendo si riduce il problema di *overfitting* e si mantengono solo le aree con maggiore attivazione, questo vuol dire che piccoli cambiamenti non cambieranno il risultato finale, perciò questo tipo di ridimensionamento aggiunge robustezza ai dati.

Esiste anche l'*Average Pooling* che, come per il precedente ridimensionamento, divide la matrice di input in sottoblocchi, calcola il valore medio di ogni sottoblocco che verrà sostituito al sottoblocco. In questo modo il ridimensionamento mantiene l'informazione media di ogni sotto-matrice.

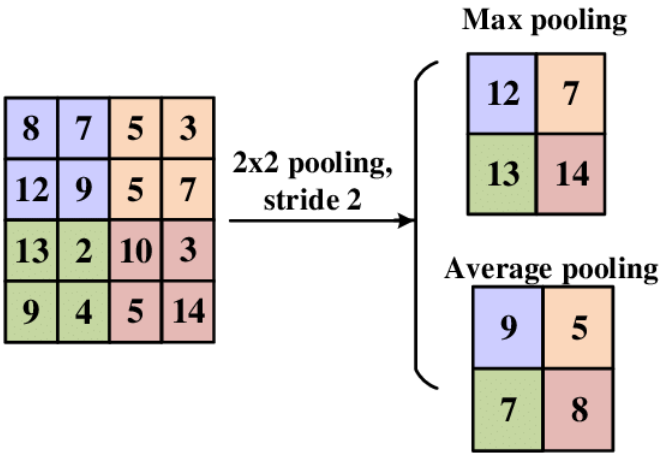


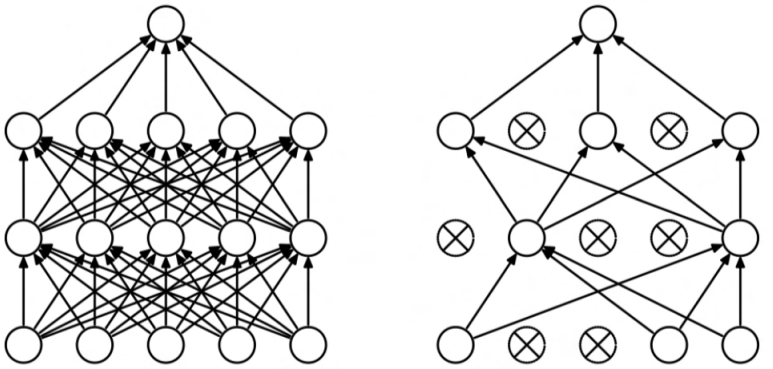
Figura 2.13: Operazione di pooling

Come lo strato convoluzionale anche i layer di pooling sono caratterizzati dagli iperparametri di *size*, *stride* e *padding*, che preservano lo stesso significato del layer convoluzionale.

In generale l'operazione di pooling introduce inevitabilmente una perdita di informazione ma garantisce un minor carico computazionale per gli strati interni della rete e quindi aiuta a prevenire il fenomeno di *overfitting* generalizzando i dati.

2.3.4 Dropout

Il layer di *Dropout* ci permette di "spegnere" o disattivare alcuni neuroni in fase di addestramento. Nello specifico questo layer va a modificare la rete stessa, per ogni epoca vengono scelti casualmente i neuroni da utilizzare per l'addestramento e quali invece da scartare.



(a) Standard Neural Net (b) After applying dropout.

Figura 2.14: Operazione di dropout

La procedura si ripete tenendo e scartando neuroni diversi ad ogni epoca. Questa tecnica ci permette di evitare il fenomeno molto comune dell'*overfitting*.

L'*overfitting* è un problema molto comune quando si va ad allenare una rete neurale, esso accade quando il modello impara o si adatta troppo bene ai dati di *training* e quindi viene definito poco generalizzabile ad altri dati. Nel dettaglio, quando andiamo a creare un modello quello che facciamo è creare tre *subset*: *training set*, contenente i dati utilizzati per allenare il modello, validation set, che contiene i dati utilizzati per validare il modello e infine test set, un insieme di dati completamente nuovi che vengono usati per verificare la capacità di predizione del modello. In fase di *training* il modello utilizzerà il *training set* per imparare a svolgere in modo corretto il task, raggiungerà quindi uno stato in cui sarà in grado di predire gli output per tutti gli altri esempi che ancora non ha visionato. Nel problema di *overfitting* l'apprendimento sembrerebbe restituire buoni risultati, quando invece si passa alla fare di predizione sul *test set* le prestazioni sui dati peggioreranno anche di molto.

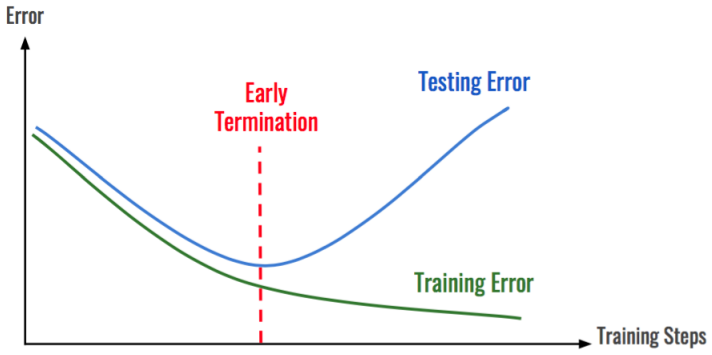


Figura 2.15: Overfitting

Generalmente si utilizza la regolarizzazione per far fronte a questo problema. Esistono diverse tipologie di regolarizzazione ma la più utilizzata è la regolarizzazione L2, in cui viene aggiunto un termine di penalizzazione di ampiezza quadrata alla funzione di perdita che risulta essere

$$J(w,b) = cost(w,b) + \lambda \sum_{j=1}^{M} w_j^2 \tag{2.12}$$

dove λ è la forza di regolarizzazione. Questo metodo tende a penalizzare vettori di pesi con grandi fluttuazioni, privileggiando valori più omogenei. La regolarizzazione, seppur efficace, da sola non basta per mitigare il problema dell'*overfitting*.

Il *Dropout* conferisce alla rete la capacità di imparare informazioni più robuste poiché, grazie al ridotto numero di connessioni, i nodi rimasti dovranno regolare i propri pesi per adattarsi all'assenza dei nodi non connessi, riducendo quindi la dipendenza tra loro che porta il modello ad una condizione di *overfitting*. Applicato ad una CNN il *Dropout* ha effetto solo durante l'addestramento e non durante il test. Il *Dropout* può essere usato con tutti i tipi di layer ma non nel layer di output.

2.3.5 Normalization

La normalizzazione per l'immagine è una tecnica che viene eseguita per migliorla, andando a massimizzare il range di valori che rappresentano tale immagine, migliorando le fasi di calcolo che la utilizzeranno. Anche per le reti neurali esistono le normalizzazioni, che vengono distinte a seconda che queste siano applicate all'immagine di input oppure alle matrici negli strati intermedi.

Nelle CNN viene implementata la *Batch Normalization*, un tipo di normalizzazione che viene applicata negli strati intermedi. Il layer di *Batch Normalization* è capace di normalizzare gli input grazie all'applicazione di una trasformazione, che mantiene la media dell'output vicina a 0 e la relativa deviazione standard vicina a 1. La *Batch Normalization* lavora andando a sottrarre la media e dividendo per la standard deviation di ciascun mini-blocco di dati in modo da normalizzarlo, in questo modo otteniamo una maggiore stabilità della rete neurale, in particolare rende la rete più veloce in fase di training e a raggiungere un'accuratezza maggiore. Inoltre, questa tecnica permette ad ogni layer di imparare autonomamente ed in maniera indipendente rispetto agli altri layer.

2.3.6 Fully Connected

Il layer *Fully Connected* (FC) ci permette di connettere completamente ogni neurone dello strato precedente ad ogni neurone dello strato successivo, utilizzando una rete di connessioni verso tutte le attivazioni del livello precedente. É solitamente posto alla fine della rete, la sua funzione è quella di prendere le immagini filtrate ad alto livello e tradurle in classi, poi utilizza le features per la classificazione delle immagini.

L'ultimo FC layer utilizza la funzione di attivazione che ci permette che la somma delle probabilità tra le classi disponibili sia 1, ovviamente la classe con probabilità maggiore corrisponde alla classe assegnata. In pratica la funzione di attivazione dello strato finale calcola la distribuzione di probabilità condizionata dei suoi valori di ingresso.

2.3.7 Softmax

Il *Softmax layer* viene comunemente visto come uno strato di attivazione seppur viene utilizzando in modo diverso rispetto agli altri. Infatti, trova collocazione nella parte

conclusiva della rete, nell'output, tipicamente subito dopo il FC, poiché restituisce la probabilità di appartenenza dell'input alle diverse classi. In base a tale distribuzione di probabilità tra le classi, la rete restituirà come output finale la classe a cui è attribuita una probabilità maggiore. La funzione di attivazione che maggiormente viene utilizzata è *softmax*, che funziona bene sia in classificatori binari e sia nei problemi multiclasse. Per i classificatori binari viene utilizzata anche *sigmoid*.

2.4 Back-propagation

L'algoritmo di *back-propagation* o retropropagazione è un algoritmo per l'addestramento delle reti neurali, che viene utilizzato nell'apprendimento supervisionato. In passato è stato dimostrato quanto una rete neurale apprenda più velocemenete grazie alla retropropagazione rispetto alle tecniche usate prima della sua introduzione. Oggi esso assume un'importanza notevole al fine dell'apprendimento delle più moderne reti neurali profonde. Questo algoritmo va a migliorare i risultati della rete grazie ad una ottimizzazione continua dei pesi che la compongono. L'algoritmo confronta il valore in uscita del sistema con il valore desiderato, sulla base della differenza così calcolata (l'errore), l'algoritmo modifica i pesi sinaptici della rete neurale, facendo convergere progressivamente il set dei valori di uscita verso quelli desiderati.

La *back-propagation* si compone di due fasi: la propagazione e l'aggiornamento dei pesi. Nella fase di propagazione, nota con il nome di *forward propagation*, i valori di ingresso attraversano tutta la rete fino al livello di output. A questo punto gli output generati vengono utilizzati, insieme alla *loss function*, per il calcolo dell'errore di predizione rispetto agli output attesi. Questo errore viene quindi usato per il calcolo del gradiente della funzione costo, che viene successivamente propagato all'indietro nella rete fino a che ogni neurone viene aggiornato con un valore che dipende dal gradiente e dal *learning rate*, da qui prende il nome di *back-propagation*.

2.5 Loss Function

La *Loss Function* o funzione di perdita non è inserita direttamente nell'architettura della rete, ma viene utilizzata per decodificare l'output. Esse costituiscono un metodo per valutare le prestazioni delle rete in fase di training. Ne esistono di diverse tipologie che vengono scelte dal progettista della rete a seconda del tipo di task che deve eseguire la rete, più in particolare dal numero di classi disponibili; permettendo inoltre di creare una sorta di parametro di regolarizzazione che penalizza alcuni errori e di conseguenza modifica l'apprendimento della rete.

Tra le funzioni di perdita più importanti utilizzate nella classificazione troviamo:

- *Binary Crossentropy*: utilizzata per la classificazione binaria (con due classi);
- *Categorical Crossentropy*: utilizzata per la classificazione multiclasse (due o più classi), l'output che restituisce è un vettore di zeri con un 1 nella posizione indicativa della classe di maggiore probabilità;

- *Sparse Categorical Crossentropy*: utilizzata anch'essa per la classificazione multi-classe ma restituisce un output diverso dalla precedente funzione di perdita, essa restituisce un numero corrispondente alla classe più probabile.

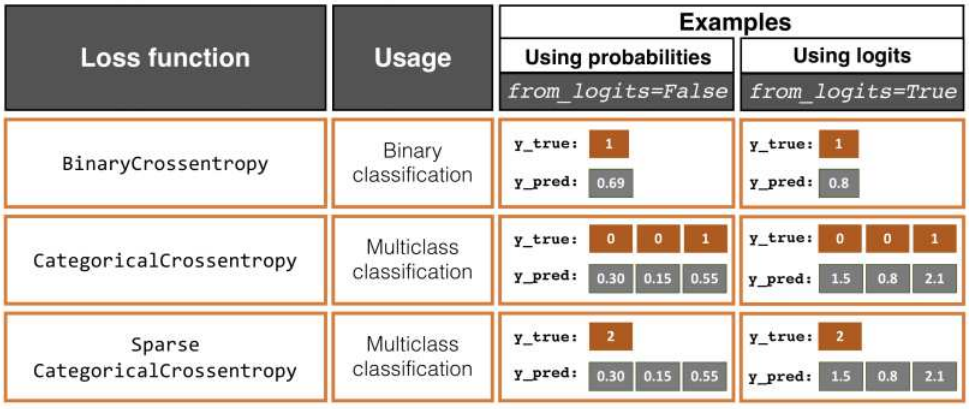


Figura 2.16: Esempi di output ritornati dalle *loss function*

Per problemi di regressione, in cui non sono presenti le classi, la *loss function* che viene comunemente utilizzata è il MSE (*Mean Squared Error*)

$$J(w,b) = \frac{1}{2n}\sum_{i=1}^{n}\|y(n) - a\|^2 \tag{2.13}$$

dove w e b sono i vettori contenenti, rispettivamente, tutti i pesi e i bias della rete, x è il vettore degli input, y è il vettore degli output, n è il numero di elementi totatli del dataset e a è l'output di x stimato dalla rete.

Come abbiamo visto l'algoritmo di *back-propagation* esegue un'ottimizzazione, quest'ultima viene applicata con la discesa del gradiente o *gradient descent*. La discesa del gradiente è un algoritmo basato sul calcolo della derivata per il raggiungimento del minimo della funzione costo, che permette di aggiustare i pesi in modo da ridurre l'errore di predizione. Nella fase di addestramento della rete, ad ogni epoca viene calcolata la funzione costo, essa viene quindi derivata per poterne trovare il minimo rispetto ai pesi e ai bias. Al termine di ogni epoca verranno quindi aggiornati i valori dei pesi e dei bias con i valori minimi.
Un iperparametro di rilevante importanza quando viene utilizzato l'algoritmo del *gradient descendet* è il *learning rate*. Questo iperparametro regola il passo di aggiornamento dei pesi e dei bias durante la fase di addestramento della rete: un *learning rate* elevato può condurre a 'salti' all'indietro troppo elevati della funzione, con una conseguente mancanza di convergenza; viceversa, un valore del parametro troppo piccolo potrebbe rallentare il processo di convergenza, che richiederà un numeri di epoche maggiore per ottenere risultati accettabili.

2.6 Data Augmentation

La *data augmentation*(dall'inglese "dati aumentati" o "dati arricchiti") è comunemente utilizzata per arricchire il *training set* della rete neurale, i dati che se ne derivano vengono

detti *augmented data*, poiché arricchiscono i dati disponibili in partenza incrementandone la diversità. É una tecnica che viene usata per ampliare i dati messi a disposizione per l'addestramento, senza però raccoglierne effettivamente di nuovi, realizzandone delle copie modificate. I dati aumentati non sono altro che copie leggermente modificate di dati già esistenti.
Nel caso di dati composti da immagini le trasformazioni che possiamo utilizzare per creare dei dati aumentati sono le seguenti:

- rotazione
- traslazione
- flip orizzontale e verticale
- scalatura o zoom
- aggiunta di rumore gaussiano
- crop

I dati aumentati vengono usati per risolvere il problema dell'*overfitting*, l'adattamento eccessivo del modello ai dati osservati. Avere a disposizione una quantità di dati maggiore con cui allenare la rete neurale, ci aiuta in fase di classificazione, migliorando le performance e aumentando anche l'accuratezza della rete, in quanto questa processerà le medisime immagini opportunamente modificate come immagini differenti, poiché avranno caratteristiche diverse. Un particolare vantaggio che possiamo trarre da questa tecnica è il risparmio di tempo nella collezione dei dati. Non è sempre facile ottenere nuovi dati da aggiungere per arricchire i *dataset*, inoltre molto spesso i dati raccolti hanno bisogno di essere etichettati e pre-processati prima di essere utilizzati, quindi tutta la fase di collezione e preparazione dei dati in alcuni casi richiede molto tempo e con il *data augmentation* riusciamo ad evitare questo problema.
Non sempre l'aumento dei dati dà un grande miglioramento nel problema dell'*overfitting*.

2.7 Architettura di una CNN e sue applicazioni

Nella costruzione di un modello il primo passo che bisognerà fare è quello di valutare quale tipo di rete andare ad usare, in base al tipo di dati, alla tipologia di task da svolgere e alla potenza di calcolo di cui si dispone. Modelli troppo semplici possono portare al problema dell'*underfitting*, viceversa modelli troppo complessi possono portare al problema dell'*overfitting*. Come abbiamo già detto le reti convoluzionali sono di fondamentale importanza nell'analisi di immagini e per questo motivo avranno un ruolo di rilevanza in questo progetto di tesi.

La struttura della CNN è composta principalmente da tre tipologie di layer, il layer convoluzionale e attivazione, quello di pooling e in fine dal *fully connected*. Tipicamente il layer di attivazione è contenuto in quello convoluzionale e per le CNN si preferisce scegliere la ReLU come funzione di attivazione. Al layer convoluzionale segue poi quello di pooling, i due possono essere ripetuti apportando alcune modifiche ai loro iperparametri; più si vuole rendere profonda la rete e più coppie di layer conv e pooling saranno presenti l'una a

seguito dell'altra. Le reti profonde sono in gran parte destinate ad un dataset di immagini più complesse, dove è necessaria un'analisi più dettagliata per avere una classificazione abbastanza accurata. Il layer di connessione viene posizionato nella parte finale della rete, più in prossimità del layer di output, anch'esso un *fully connected layer*.

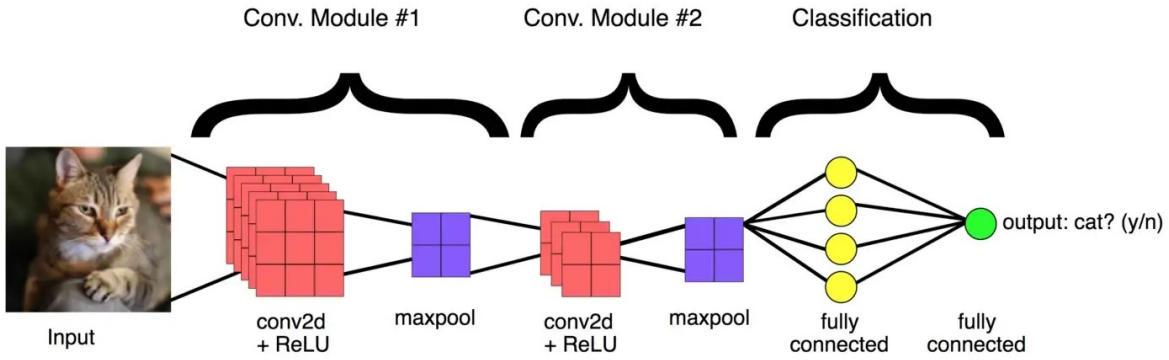


Figura 2.17: Tipica struttura di una CNN

In generale la costruzione di una rete neurale convoluzione richiede molto tempo, poiché non esistono precise regole di costruzione, che determinino con precisione il numero di layer da utilizzare, la loro tipologia e i valori che i loro iperparametri assumono in relazione al tipo di task che la rete deve eseguire. Dunque tutte le reti neurali convoluzionali che vengono presentate in letteratura sono frutto di sperimentazione, poiché presentano un corposo numero di parametri da addestrare e richiedono un elevato tempo computazionale. Negli anni alcune architetture sono divenute rilevanti come punto di partenza per la creazione di nuove CNN, l'utilizzo e l'adattamento di queste architetture prende il nome di *transfer learning*.

2.8 Transfer learning

Il *transfer learning* è un metodo di *machine learning* che utilizza un modello pre-addestrato sviluppato per svolgere una precisa attività di apprendimento, come punto di partenza per lo sviluppo di un modello destinato all'esecuzione di una diversa attività. Il problema del dispendio di tempo per la realizzazione di una rete neurale viene quindi ovviato dal riutilizzo di questi modelli già impostati.

Il *transfer learning* nasce intorno al 1976 e negli anni è aumentato notevolmente il suo sviluppo e utilizzo, in particolare dal 2012 abbiamo assistito ad un rapido progresso delle reti convoluzionali. Il miglioramento delle architetture ha permesso di raggiungere performance sempre più elevate, passando da una accuratezza del 63.3% della rete AlexNet arrivando ad una accuratezza del 90% delle reti EfficientNet.

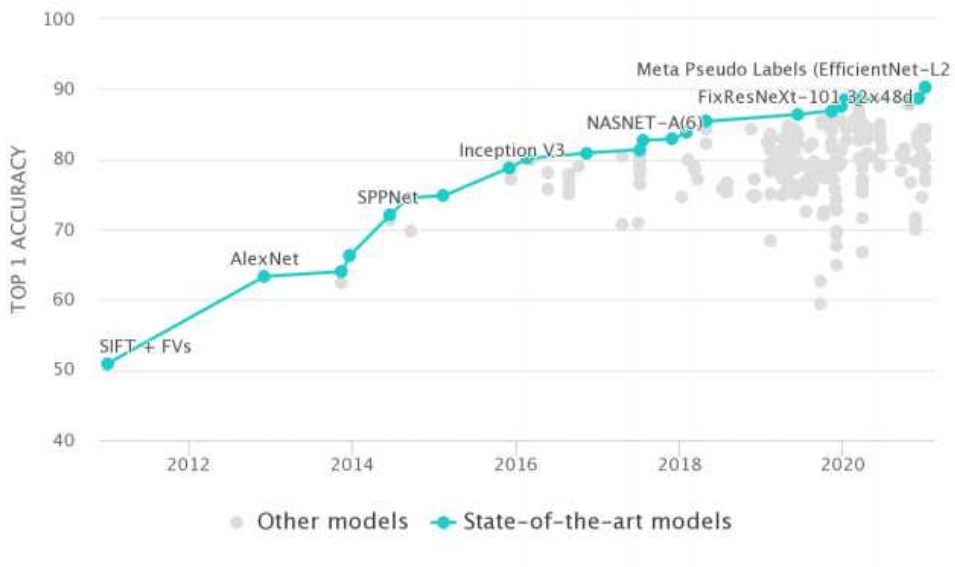


Figura 2.18: Evoluzione dei modelli CNN

Il *transfer learning* richiede un apprendimento supervisionato, in particolare esso consiste in un riaddestramento di una rete neurale. La rete che si vuole utilizzare infatti è già stata addestrata con un set di dati di grandi dimensioni, tipicamente con lo scopo di classificare delle immagini su larga scala. Il modello pre-addestrato può essere utilizzato interamente o parzialmente, in quest'ultimo caso il modello viene personalizzato in base alle esigenze del costruttore, che quindi trae beneficio da una porzione del modello pre-addestrato. Il *transfer learning* viene principalmente utilizzato per la classificazione di immagini, l'idea che sta alla base del suo utilizzo è che se un modello viene addestrato su un set di dati sufficientemente ampio e generico, allo stesso modo il modello riuscirà a sfruttare la generalità appresa per risolvere un ulteriore problema di classificazione. In questo modo riusciamo ad ottenere dei risultati ottimali senza dover addestrare da zero un nuovo modello di reti neurali, disperdendo risorse e tempo nel processo di apprendimento.

Esistono due metodi implementativi del *transfer learning* e sono:

- Estrazione di funzionalità: consiste nel riutilizzo della rete neurale per estrarre delle *features* significative ulteriori. La rete convoluzionale conterrà tutte le informazioni utili per la classificazione generica delle immagini, bisognerà aggiungere la parte finale relativa alla classificazione vera e propria del modello, che servirà per adattare il modello generico pre-addestrato al nostro specifico task di classificazione.

- *Fine-tuning*: in questo metodo vengono bloccati o "congelati" alcuni livelli della rete, tipicamente quelli superiori del modello pre-addestrato, questo ha lo scopo di limitare l'addestramento e l'elaborazione a un numero ridotto di parametri. Questa tecnica ci permette quindi di perfezionare la rete, in modo da renderla più conforme all'esecuzione del nuovo task. Gli strati riutilizzati sono etichettati come "read-only", velocizzando i tempi di addestramento, diminuendo la potenza di elaborazione richiesta e incrementando l'accuratezza della rete.

La maggior parte dei modelli pre-addestrati sono disponibili in rete e accessibili pubblicamente. Vediamo di seguito alcuni modelli pre-addestrati.

2.8.1 VGG16

VGG16 è una rete neurale convoluzionale (CNN), il suo nome è un acronimo e sta per *Visual Geometry Group*, il dipartimento di scienze ingegneristiche dell'Università di Oxford, che ha definito la sua architettura. É stata pensata per task di classificazione e identificazione, la sua principale caratteristica è quella di utilizzare filtri convoluzionali(ricettivi) piccoli. Le reti VGG furono tra le prime che ebbero l'idea che usare filtri 3×3, quindi filtri più piccoli, ma ripetuti in sequenza, permettano di ottenere gli stessi risultati ottenuti usando filtri ricettivi molto più grandi. In questo modo si ottenne un grande risparmio computazionale vista la diminuzione del numero di parametri, portando ad una mappatura migliore tra le immagini.
VGG16 ha 16 livelli e 138 milioni di parametri, nonostante ciò ha una struttura semplice e uniforme. Esiste anche la sua versione con 11 e 19 livelli che prendono il nome di VGG11 e VGG19 rispettivamente.

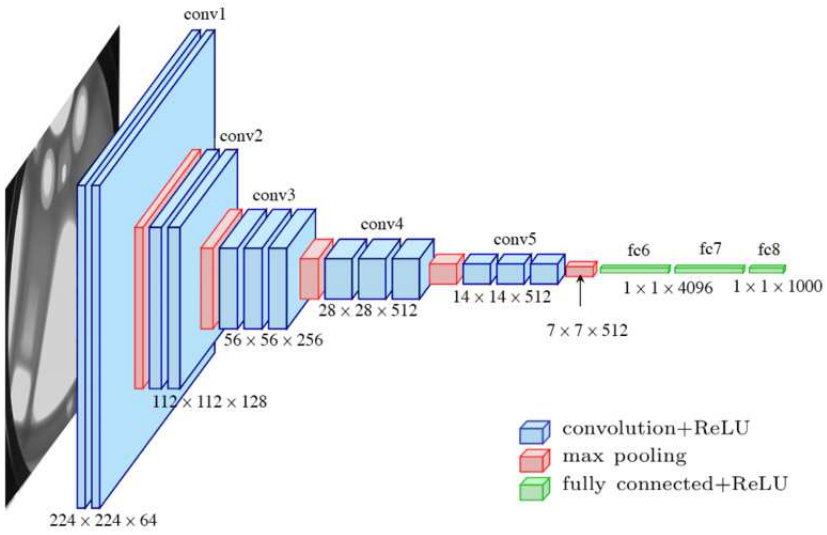


Figura 2.19: Architettura di VGG16

Nel dettaglio essa è formata da 13 strati convoluzionali, 5 di *Max Pooling* e 3 *layer Dense*, di questi 21 solamente 16 sono pesati. Richiede immagini di input di dimensione $224 \times 224 \times 3$, dove la terza dimensione indica la presenza dei canali RGB dell'immagine (immagine a colori). Invece di avere un grande numero di iper-parametri essa è composta principalmente da filtri di piccole dimensioni: 3×3 ma ne esistono anche di dimensioni 2×2 e 1×1. *Padding* e *stride* sono settati in modo tale che la risoluzione spaziale sia la stessa prima e dopo l'applicazione del filtro. VGG16 è stata addestrata per il riconoscimento di oggeti e classificazione, essa è in grado di classificare 1000 immagini di 1000 categorie diverse con una accuratezza del 92.7%. É uno degli algoritmi più popolari per la classificazione di immagini ed è facile da utilizzare.

2.8.2 ResNet50

ResNet sta per Residual Network ed è un particolare tipo di CNN introdotta nel 2015 ed è comunemente usata nell'ambito del *Computer Vision*. É una rete neurale profonda,

infatti è composta da 50 *layer*(48 strati convoluzionali, uno strato *MaxPool* e uno strato *pool* medio). Le reti neurali residue sono un tipo di rete neurale artificiale che formano reti impilando blocchi residui. La rete ResNet originale comprendeva 34 strati, in seguito sono stati aggiunti più *layer* convoluzionali. Ad oggi esiste una versione di ResNet50 ancora più profonda la ResNet101.

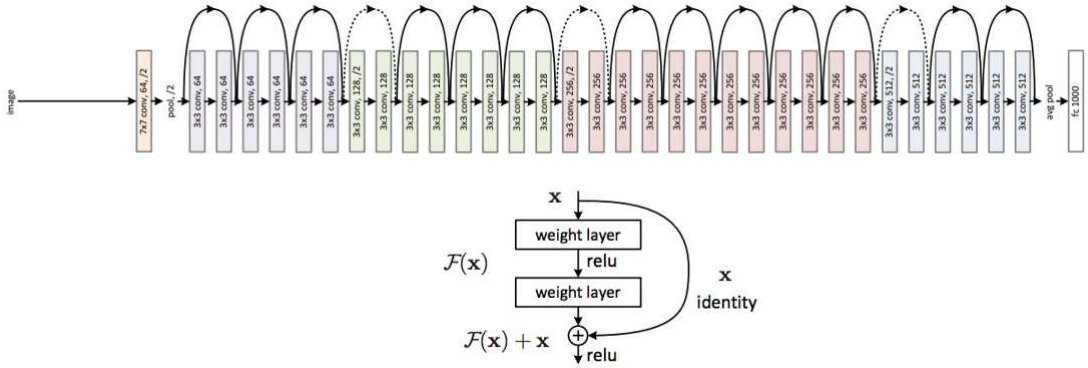


Figura 2.20: Architettura di ResNet50

L'architettura di una rete residuale ResNet segue due regole base: la prima è riferita al numero di filtri, che deve essere uguale in ogni layer e deve tener conto della dimensione della mappa delle caratteristiche in uscita; la seconda è che se la dimensione della mappa delle carteristiche è dimezzata, allora il numero di filtri deve raddoppiare, questo per mantenere la complessità temporale di ogni livello. Una caratteristica importante che differenzia la ResNet50 dalle altre tipologie di reti residuali è la presenza di un blocco di costruzione. Il blocco residuo costituisce un filtro convoluzionale di dimensioni 1×1, comunemente noto con il nome di "bottleneck", letteralmente collo di bottiglia, che riduce il numero di parametri e il numero di moltiplicazioni di matrice. Questo blocco permette quindi un addestramento molto più veloce in ogni layer.

2.8.3 GoogLeNet

GoogLeNet o Inception è una rete neurale proposta da Google nel 2014, vincitrice della competizione ILSVRC sulla classificazione delle immagini. Questa architettura è nata con lo scopo di migliorare le prestazioni, in termini di *error rate*, delle più note architetture che la precedono come AlexNet e ZF-Net.

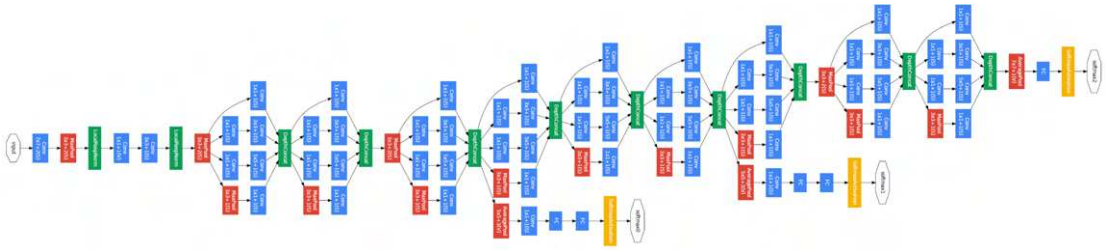


Figura 2.21: Architettura di GoogLeNet

La GoogLeNet è composta da 22 strati ed ha una struttura molto diversa dalle architetture appena citate, essa utilizza diversi tipi di metodi come la convoluzione con filtri

molto piccoli di dimensione 1×1 e *average pooling* globale che rendono la rete un'architettura profonda. Nel dettaglio l'utilizzo di filtri convoluzionali 1×1 è stato utilizzato con lo scopo di ridurre notevolmente il numero di parametri che vengono utilizzati dalla rete. Tale riduzione permette ai costruttori di realizzare una rete più pronfonda, evitando l'impiego di un numero immane di parametri che possano portare al classico problema di *overfitting* della rete. Per quanto riguarda la scelta del *average pooling* globale, invece del comune *fully connected* alla fine della rete, è stata fatta anch'essa con lo scopo di decrementare il numero di parametri trainabili a zero e migliorare l'accuratezza del circa 0.6%.

In aggiunta a queste caratteristiche, la rete neurale GoogLeNet utilizza un modulo iniziale di struttura diversa, che implementa una serie di convoluzioni con filtri di varie dimensioni, seguite dal *maxpooling*. Tali convoluzioni vengono eseguite parallelamente all'ingresso; l'output di queste operazioni viene impilato insieme per generale l'output finale. L'idea di base è che i filtri di convoluzioni di dimensioni diverse gestiscano meglio gli oggetti in scala multipla. Questo blocco iniziale viene definito dai costruttori come *Inception layer*, la sua applicazione permise grossi miglioramenti prestazionali e del numero di parametri e portò alla riduzione dell'*overfitting*.

La GoogLeNet ha dimostrato di essere migliore rispetto alle architetture che la precedono sia nei task di classificazione che nel task di rilevazione. In seguito vennero create sue versioni modificate che migliorarono la struttura del blocco di *inception*, riducendono l'impatto computazionale e adottando una architettura molto più semplice e con moduli più piccoli.

2.8.4 Unet

Unet è una *Fully Convolutional Network* (FCN) sviluppata per applicazioni in campo medico, come l'individuazione di tumori nei polmoni e nel cervello attraverso la segmentazione delle immagini. L'architettura della Unet la rende infatti particolarmente adatta a risolvere problemi di segmentazione delle immaggini, essa è in grado di lavorare anche un piccoli frammenti.

La struttura di questa rete si compone du due elementi fondamentali: il *encoder* e il *decoder*. Il *encoder* prende l'immagine d'ingresso e la trasforma di una *feature map*, estraendo gli elementi chiave dell'immagine necessari per catturare il contesto della stessa, lo fa grazie agli strati di convoluzione e *pooling* della rete. Il *decoder* è utilizzato dalla rete per amplificare la *feature map* in una immagine, grazie ai livelli di deconvoluzione (o convoluzioni trasposte), che permettono la localizzazione degli elementi. La Unet assume quindi una struttura a U [Figura 2.22], da cui ne prende il nome. I due percorsi fatti da *encoder* e *decoder* sono simmetrici, e rappresentano le fasi dette *down sampling* e *up sampling*, che comunicano tra loro condividendo informazioni e formando le così dette connessioni scorciatoia (*short connection*). Sono proprio queste connessioni che permettono alla rete di catturare più informazioni contenendo la complessità computazionale.

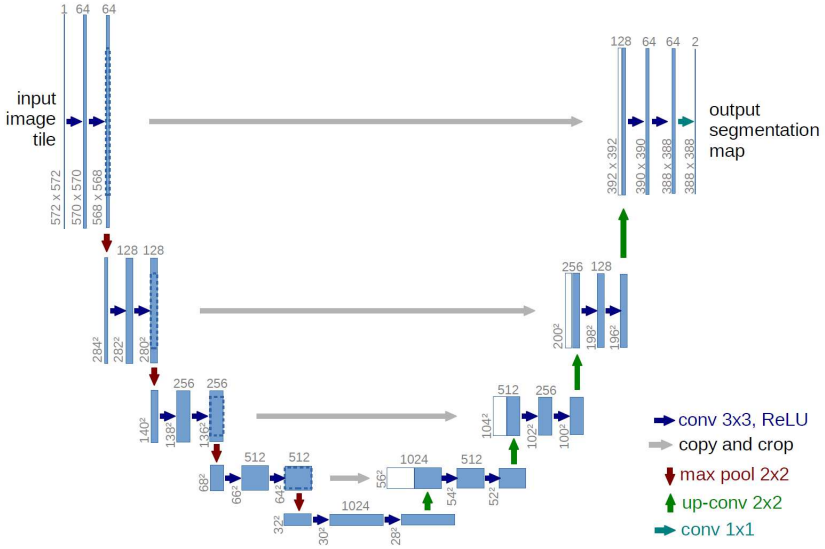


Figura 2.22: Architettura di Unet

Un'importante caratteristica di questa rete e in particolare della fase di *up sampling* è che il numero di canali presenti sono più grandi, consentendo di propagare maggiori informazioni ai livelli di risoluzione più alta. Un'altra cosa da tenere inconsiderazione quando si utilizza la Unet è la creazione di maschere binarie, che vengono utilizzate dalla rete come *labels* per l'individuazione degli elementi. Le maschere vengono realizzate impostando a 1 i pixel che costituiscono l'oggetto da segmentare e impostando a 0 i pixel restanti.

2.9 Procedure di apprendimento

Nei paragrafi precedenti abbiamo visto come i termini "addestramento" e "apprendimento" siano comunemente usati poiché strettamente legati al *machine learning*, *deep learning* e più in particolare alle reti neurali. Grazie all'intelligenza artificiale una macchina è in grado di imparare a svolgere numerosi compiti e di migliorare le proprie capacità e prestazioni nel tempo. Esistono differenti modalità di apprendimento, tutte efficaci, che differiscono non solo per gli algoritmi utilizzati, ma anche per lo scopo per cui sono realizzate le macchine stesse.
L'apprendimento si divide in tre tipologie:

- Apprendimento supervisionato;
- Apprendimento non supervisionato;
- Apprendimento semi-supervisionato;
- Apprendimento con rinforzo.

L'apprendimento supervisionato o *supervised learning* è basato sull'osservazione di un numero di esempi di cui è nota con precisione l'appartenenza ad una specifica classe o output. Vengono quindi presentati alla macchina degli input e i relativi output desiderati o *label*. L'idea è quella di utilizzare questi dati come esempi, tramite i quali la macchina

può apprendere; successivamente nuovi dati verranno dati all'algoritmo, questa volta i dati non conterranno l'output desiderato, che verrà utilizzato solamente per verificare la correttezza delle predizioni della macchina. Gli algoritmi che fanno uso di apprendimento supervisionato vengono utilizzati in molti settori, da quello medico a quello di identificazione vocale: essi, infatti, hanno la capacità di effettuare ipotesi induttive, ossia ipotesi che possono essere ottenute scansionando una serie di problemi specifici per ottenere una soluzione idonea ad un problema di tipo generale.

Diversamente dal precedente, nell'apprendimento non supervisionato o *unsupervised learning* si dispone di un numero di esempi di cui però non è nota la classe di appartenenza o output. In questa tipologia di apprendimento si dà una maggior libertà alla macchina, che dovrà organizzare le informazioni in maniera intelligente e imparare quali siano i risultati migliori, trovare cioè la suddivisione in classi o gruppi (cluster) che massimizza la varianza inter-classe e minimizza la varianza intra-classe. L'azione di raggruppamento di dati viene definita *clustering* e costituisce il punto di partenza per la comprensione dei dati.

L'apprendimento semi-supervisionato è un approccio che coinvolge i due approcci appena descritti. Esso utilizza un training set in cui solo una parte di dati sono etichettati, la restante parte è composta da dati non etichettati.

L'ultimo e più complesso metodo di apprendimento è l'apprendimento per rinforzo. Esso prevede che il sistema sia in grado di migliorare il proprio apprendimento e di comprendere le caratteristiche dell'ambiente circostante, questo grazie all'utilizzo di appositi strumenti. La macchina messa a disposizione viene dotata di elementi di supporto, come sensori, telecamere, GPS e altri, che le consentono di ricavare informazioni sull'ambiente. Le scelte che poi la macchina andrà a fare saranno effettuate in base a ciò che riesce a rilevare, infatti un esempio di utilizzo di questo tipo di tecnologia è il pilota automatico, che grazie a un complesso sistema di sensori di supporto, è in grado di riconoscere ostacoli, strade e di comprendere le indicazioni stradali.

Capitolo 3

CNN 1 : Rilevazione della presenza e assenza del disco ottico

3.1 Raccolta dati e labeling

L'obiettivo della prima CNN è la classificazione delle immagini retiniche in due classi: Presenza del disco ottico e Assenza del disco ottico. I dati che utilizza la rete neurale convoluzionale per risolvere il problema di classificazione binaria sono stati forniti da Centervue S.p.A, un'azienda di Padova specializzata nell'ambito delle teconologie oftalmologiche. Centervue SpA è parte di Revenio Group e, insieme a Icare Finland Oy, Icare USA Inc. e iCare World Australia Pty Ltd rappresenta il brand iCare. ICare offre dispositivi medici per lo screening e la diagnosi di numerose patologie oculari, come ad esempio il glaucoma, la retinopatia diabetica e la degenerazione maculare (AMD). La linea di prodotti iCare consiste di sistemi di imaging automatizzati del fondo dell'occhio, perimetri, tonometri con tecnologia *rebound* e soluzioni software.

3.1.1 Strumenti

Le immagini sono state acquisite dai *device* Eidon, Compass e DRSplus.

Eidon è un dispositivo in grado di acquisire immagini retiniche confocali ad alta risoluzione (3680×3288 pixel). Le immagini possono essere acquisite in diversi *field of view* (central, nasal, temporal, central-nasal, inferior, inferior-nasal, inferior-temporal, superior, superior-nasal e superior-temporal.) e con diverse modalità (*TrueColor* e *Infrared*). Eidon combina infatti le migliori caratteristiche dei sistemi SLO (*Scanning Laser Ophthalmoscopy*) con quelle dei tradizionali sistemi di *fundus imaging*, garantendo degli altissimi standard di prestazione nell'ambito dell'imaging retinico. É un *device* dal design compatto che non richiede un PC aggiuntivo, ha flessibilità nel passare dalla modalità completamente automatizzata a quella completamente manuale. Dal momento che è un dispositivo di imaging confocale, ha la capacita' di acquisire immagini di alta qualità anche in presenza di eventuali opacità o di cataratta. La pupilla minima del dispositivo è 2.5 mm, pertanto le operazioni di acquisizione delle immagini possono essere effettuate anche senza necessita' di dilatazione della pupilla. Grazie alla tecnologia confocale, Eidon garantisce dunque una maggiore nitidezza, una migliore risoluzione, maggiori dettagli e contrasto rispetto alle fundus camera tradizionali. Tutte queste caratteristiche, unite

alla facilità di utilizzo, rendono Eidon uno strumento prezioso ed efficiente in qualsiasi ambiente clinico.

Figura 3.1: iCare Eidon

Compass è un perimetro automatizzato che permette di effettuare esami del campo visivo compensando in tempo reale i movimenti dell'occhio del paziente (*retinal tracking*), massimizzando l'affidabilità dell'esame perimetrico. Inoltre, Compass permette di acquisire immagini del fondo dell'occhio in alta qualità sia in infrarosso sia a colori. Compass si può quindi definire come "fundus perimeter", dal momento che permette di effettuare esami del campo visivo e allo stesso tempo di fornire immagini retiniche confocali in altissima qualità. Il funzionamento è automatico (grazie alle funzioni di auto-allineamento e autofocus) e facile da utilizzare per il clinico. Le immagini acquisite hanno risoluzione di 1920×1920 pixel.

Figura 3.2: iCare Compass

DRSplus è un dispositivo di imaging confocale, che permette di acquisire immagini *TrueColor* in altissima risoluzione in maniera completamente automatizzata. La tecnologia *TrueColor Confocal*, che è considerata uno standard di alta qualità dell'immagine, fornisce immagini ricche di dettagli con maggiore nitidezza (3600×2910 pixel) e contrasto rispetto alle fundus camera tradizionali. Il DRSplus è una fundus camera non midriatica, ovvero consente di acquisire immagini del fondo oculare senza necessità di dilatazione farmacologica della pupilla. La pupilla minima è di 2.5 mm. Il DRSplus è un dispositivo

di facile utilizzo e permette di ridurre al minimo il tempo di esecuzione degli esami. Di conseguenza, permette di effettuare screening in maniera facile e veloce.

Figura 3.3: iCare DRSplus

3.1.2 Dataset

I dati a disposizione si compongono di immagini acquisite con Eidon, Compass e DRSplus. Le immagini sono state acquisite con diversi *field of view* e per questo motivo non sempre contengono il disco ottico. Inoltre, esse appartengono a soggetti sani e pazienti con varie patologie come il glaucoma, dunque non sempre il disco ottico è facilmente individuabile anche manualmente. Alcune immagini presentano aree atrofiche, ovvero porzioni di retina più chiara e di varie dimensioni che rendono difficile la localizzazione del disco ottico, dunque le aree atrofiche influenzano l'aspetto del disco ottico che assume diverse forme e dimensioni.

Figura 3.4: Esempi di retine con aree atrofiche

La totalità delle immagini è stata raccolta dall'azienda in diversi momenti di acquisizione. A seguito di questa procedura le immagini sono state rese a disposizione per la realizzazione di questo progetto di tesi. La classificazione delle immagini è stata eseguita da un *grader* umano e ha portato alla seguente distribuzione dei dati.

DEVICE	TOTALE	ASSENZA	PRESENZA
Eidon	12283	880	11403
Compass	1949	11	1938
DRSplus	4273	324	3949
TOTALE	18505	1215	17290

Tabella 3.1: Tabella della distribuzione dei dati

Il totale delle immagini disponibili è più di 18000. Si può osservare come la cardinalità della classe Assenza sia nettamente inferiore a quella della classe Presenza. Ciò è spiegabile dal fatto che uno degli elementi più importanti che il clinico valuta durante un esame del fondo oculare è il disco ottico, per cui risulta di fondamentale importanza la sua individuazione. Tale sbilanciamento e la sua risoluzione verranno trattati nel dettaglio nel paragrafo 3.2.

Un ostacolo che abbiamo dovuto affrontare deriva dai dati: tra le immagini fornite ne sono presenti alcune in cui il disco ottico viene contenuto parzialmente, per questa ragione si è scelto di classificare le immagini come Presenza solo quelle in cui il disco ottico compare per circa l'80% e Assenza le restanti.
La scelta è stata fatta valutando anche la realizzazione della seconda rete convoluzionale, che abbiamo già anticipato che si occuperà di trovare il centro del disco ottico. Come possiamo vedere dalle due immagini riportate come esempio, nella prima (Figura 3.5)

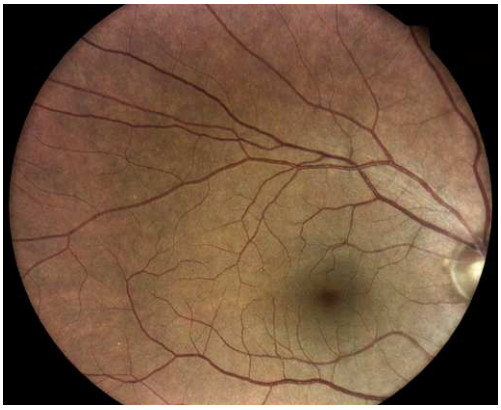


Figura 3.5: Immagine classificata come Assenza

Figura 3.6: Immagine classificata come Presenza

risulterà molto più complesso il calcolo del centro del disco ottico, inoltre il risultato potrebbe non essere veritiero per mancanza di informazioni. L'aggiunta di questa tipologia di immagini nella classe Presenza potrebbe portare in errore la seconda rete neurale convoluzionale descritta nel capitolo 4. Nella seconda immagine invece (Figura 3.6), sebbene il disco non sia completamente visualizzato, il centro del disco può essere individuato chiaramente.

Dopo aver diviso le immagini nelle due classi siamo passati alla procedura di *labeling* che è stata fatta con l'utilizzo di Matlab. Abbiamo creato un *dataframe* che ad ogni immagine ne estrae il nome ed altre informazioni, quali: l'ID del paziente, il campo di esecuzione dell'immagine, occhio destro e sinistro, *label* pari a 0 per l'Assenza e 1 per la Presenza. Il tutto è stato quindi caricato sulla piattaforma online di Google Colab che è stata utilizzata per il *pre-processing* dei dati e la realizzazione della rete neurale convoluzionale.

3.2 Preparazione dei dati

3.2.1 Unsampling

Il primo problema che abbiamo dovuto affrontare è stato quello delle classi fortemente sbilanciate, come indicato nel capitolo 3.1 e che viene reso più evidente dal seguente istogramma. Sull'asse verticale abbiamo riportato il numero di immagini e sull'asse orizzontale i *device* a cui appartengono, differenziando le immagini secondo la classe Presenza e Assenza.

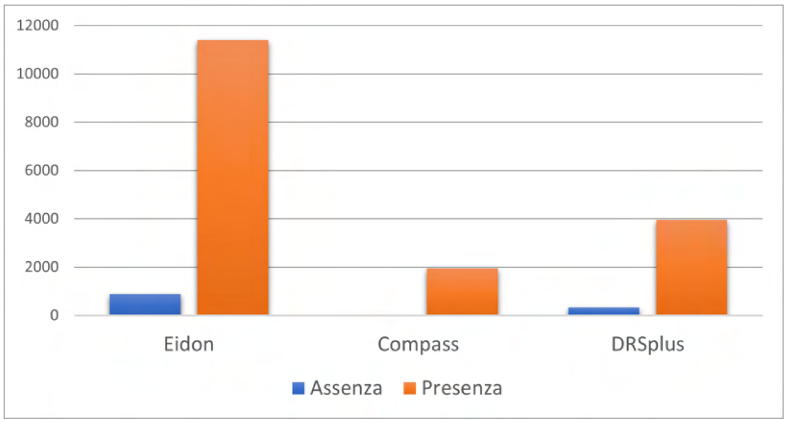


Figura 3.7: Istogramma della distribuzione delle immagini

Il problema delle classi sbilanciate è abbastanza comune nei *task* di classificazione e deve essere opportunamente trattato, poiché lo sbilanciamento introduce un bias nella previsione della classe maggioritaria, nel nostro caso la Presenza. Per questo motivo abbiamo deciso di risolvedere l'ostacolo delle classi *unbalanced* utilizzando un metodo che si chiama *unsampling*.

L'*unsampling* è una tecnica che bilancia i dataset riducendo la dimensione della classe maggioritaria. Per realizzare questo metodo siamo quindi partiti dal numero di immagini classificate come Assenza (1215 immagini) e abbiamo selezionato in ugual misura le immagini dalla classe Presenza (1215 immagini). Durante questa procedura abbiamo ritenuto opportuno prendere un'ulteriore accortezza, ovvero quella di mantenere la stessa proporzione di presenza delle immagini dai tre diversi dispositivi. Per questo motivo, abbiamo applicato la tecnica di campionamento singolarmente per ognuno dei diversi *device*. Al fine di rendere più comprensibile la procedura ne riportiamo un esempio: prendiamo in considerazione il primo *device* Eidon a cui appartengono 12283 immagini, di queste solamente 880 sono assenze; l'*unsampling* ci consentirà quindi di prendere 880 immagini casualmente dall'insieme delle 12283 immagini classificate come presenze, lo stesso procedimento viene applicato alle immagini di Compass e DRSplus.

Un'altra accortezza che è stata presa durante il campionamento è la distinzione degli occhi. Nei gruppi sono presenti immagini dello stesso occhio acquisite in diversi *field of view*, il campionamento è stato quindi eseguito cercando di prendere immagini di occhi distinti l'uno dall'altra. Questo accorgimento è necessario per avere un corretto funzionamento della rete, è fondamentale infatti che la rete utilizzi dati completamente nuovi in fase di test. Per dati nuovi intendiamo che la rete non deve riconoscere in una immagine di test qualcosa che ha già visto in fase di training. Sebbene due immagini dello stesso occhio, acquisite su campi diversi, possano risultare a prima vista due immagini diverse, esse contengono in realtà porzioni della stessa retina che la rete è potenzialmente in grado di riconoscere. In questo modo la rete vedrà in fase di test immagini completamente nuove perché appartenenti a occhi non inclusi nel training e nel validation set. Questo è di fondamentale importanza per avere delle stime realistiche delle performance del modello, infatti se il test set contenesse delle porzioni di retina comuni al training set o al validation set, la rete sarebbe portata a classificare correttamente l'immagine. L'esito corretto della predizione sarebbe però da attribuire al fatto che l'immagine è stata già vista in fase di training della rete. Le performance sarebbero di fatto più alte ma non rispecchierebbero la capacità di apprendimento reale della rete, nella realtà il modello dovrà essere in grado

di classificare correttamente immagini di occhi completamente nuovi.

Dopo il procedimento di bilanciamento quello che otteniamo è un dataset di 1215 immagini appartenenti alla classe Assenza e ulteriori 1215 immagini appartenenti alla classe Presenza, abbiamo dunque bilanciato le due classi. Non possiamo parlare invece di bilanciamento se andiamo ad analizzare le immagini dal punto di vista dei *device*. Notiamo infatti che le immagini che appartengono a ciascun dispositivo compaiono nel *dataset* finale con diverse percentuali, come possiamo vedere in Figura 3.8

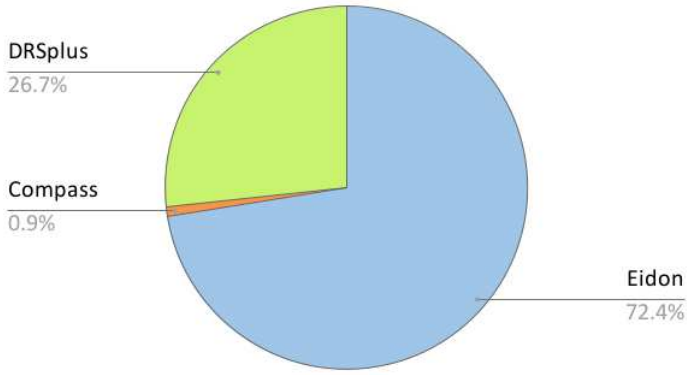


Figura 3.8: Distribuzione delle immagini tra i device

Il dataset finale è dunque composto dal 72.4% da immagini acquisite con Eidon, dal 26.7% da immagini acquisite con DRSplus e dal 0.9% da immagini acquisite con Compass. In questo progetto di tesi si è ritenuta più importante il bilanciamento delle classi relative all'output della rete, indipendentemente che queste derivassero da un *device* rispetto che ad un altro, l'obiettivo infatti è quello di classificare le immagini retiniche a prescindere dalla loro derivazione. Per tale ragione si è scelto di trascurare la provenienza delle immagini, cercando di sviluppare un modello che funzioni indipendentemente dal *device* con cui sono state acquisite le immagini.

3.2.2 Padding delle immagini

Dalle specifiche tecniche dei *device* osserviamo che questi acquisiscono immagini in diverse dimensioni:

- Eidon: 3288×3680 pixel
- DRSplus: 3600×2910 pixel
- Compass: 1920×1920 pixel

La differenza di dimensioni tra le immagini costituisce un problema, poichè la rete neurale convoluzionale richiede di avere in input dati della stessa dimensione. L'individuazione di un fattore di scala comune a tutte le immagini non è la soluzione più adatta per affrontare questo problema, perchè, in previsione della costruzione della CNN e di conseguenza dell'utilizzo di appositi filtri che verranno applicati all'immagine, è indispensabile rendere omogenee le dimensioni delle immagini. Per dare una dimostrazione

dell'effetto della riscalatura osserviamo cosa accade a una immagine di DRSplus, che viene riscalata a una dimensione di 224×224 pixel, dimensione di riscalatura che abbiamo preso da esempio e che abbiamo pensato sia una dimensione appropriata in relazione alla numerosità delle immagini e allo spazio disponibile nell'ambiente Colab, in cui abbiamo realizzato la rete.

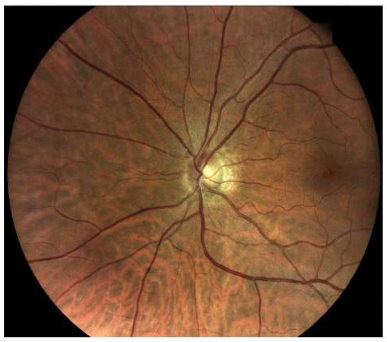


Immagine originale (DRSplus)

Immagine riscalata a 224 x 224 pixel

Possiamo osservare chiaramente che una semplice riscalatura dell'immagini ha portato a una deformazione della stessa, che risulta quindi allungata. Deduciamo quindi che sarà necessario ovviare a questo problema per preservare le informazioni originali della nostra immagine. Ecco che si rende quindi necessaria l'operazione di *padding* come risoluzione a questo problema.

Al fine di non tagliare le immagini, perdendo così possibile informazione utile in fase di addestramento, si è scelto di rendere le immagini di dimensione 3680×3680 pixel, grazie all'aggiunta di un bordo nero. La scelta delle dimensioni 3680×3680 pixel è stata fatta prendendo come riferimento la grandezza maggiore che troviamo tra le dimensioni delle immagini. Per semplicità si è scelto di rendere le immagini di forma quadrata.

Mostriamo di seguito i risultati della procedura di *padding*, prendendo tre immagini, una per ogni *device*:

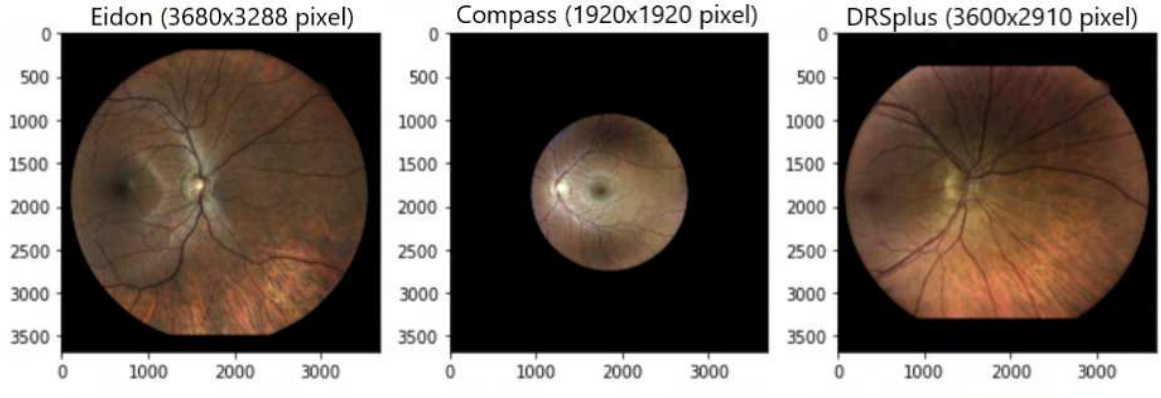


Figura 3.9: Risultato del padding

dai risultati possiamo vedere che l'aggiunta dei bordi neri di *padding* ha influito pesantemente nelle immagini di Compass per via della loro ridotta dimensione rispetto gli altri due *device*. Per quando riguarda DRSplus e Eidon l'aggiunta di bordi neri mantiene l'immagine molto è più simile all'originale. Notiamo inoltre che, sebbene le immagini siano state modificate l'informazione in esse contenuta non viene alterata. L'immagine della

retina non è stata in alcun modo modificata. A conclusione della procedura di *padding* abbiamo ottenuto un *dataset* di immagini tutte di dimensione 3680×3680 pixel.

3.2.3 Split dei dati

Dopo la preparazione dei dati segue la loro divisione nei set di training, test e validation. Per lo sviluppo di un modello di *deep learning* è importante dividere i dati in questi tre set, poiché ognuno di essi ha un compito specifico e diverso. Il training set verrà utilizzato esclusivamente in fase di addestramento del modello, le immagini contenute nel train serviranno infatti alla rete ad imparare quando il disco è presente nelle immagini oppure quando non è presente. Il validation set viene anch'esso usato in fase di addestramento per validare i risultati ottenuti dal modello e per evitare l'*overfitting*. Nel corso dell'addestramento della rete neurale convoluzionale, alla fine di ogni epoca vengono valutate le metriche scelte in fase di progettazione (e.g. *accuracy*) e vengono aggiustati gli iperparametri per l'epoca successiva. Questo algoritmo prende il nome di *backpropagation*, esso utilizza una ottimizzazione continua dei pesi che la compongono. In questo modo gli iperparametri del modello verranno modificati ad ogni epoca cercando di minimizzare l'errore. Infine, il test set è un set di dati completamente indipendente che viene utilizzato per verificare le performance della rete su dati nuovi e diversi rispetto a quelli imparati in fase di addestramento. La cosa importante è che le immagini che compongono il test set siano delle immagini completamente nuove per la rete.

Nel nostro caso abbiamo eseguito la procedura di divisione dei dati con estrema attenzione a comporre i set destinando al training set l'80% delle immagini, al test set il 10% e il restante 10% al validation set, ognuno dei quali conterrà immagini di occhi diversi. Per semplicità l'operazione di split nei tre set è stata fatta dividendo singolarmente le immagini di Eidon, Compass e DRSplus. In questo modo sarà più semplice gestire anche le immagini che contengono porzioni di retina dello stesso occhio. Dopo la divisione, i tre set contengono il seguente numero di immagini.

TRAINING SET		
DEVICE	**ASSENZA**	**PRESENZA**
Eidon	706	706
Compass	9	9
DRSplus	257	257
TOTALE	972	972

Tabella 3.2: Tabella della distribuzione dei dati nel training set

TEST SET		
DEVICE	**ASSENZA**	**PRESENZA**
Eidon	86	86
Compass	1	1
DRSplus	33	33
TOTALE	120	120

Tabella 3.3: Tabella della distribuzione dei dati nel test set

VALIDATION SET		
DEVICE	**ASSENZA**	**PRESENZA**
Eidon	88	88
Compass	1	1
DRSplus	34	34
TOTALE	123	123

Tabella 3.4: Tabella della distribuzione dei dati nel validation set

3.3 Caricamento dei dati

Dopo aver omogeneizzato le dimensioni delle immagini con l'aggiunta del bordo nero e dopo averle divise nei tre set(training, test e validation) siamo giunti alla fase di caricamento dei dati. I dati vengono cariati in Colab utilizzando la funzione *ImageDataGenerator* di *Keras*, con questa funzione siamo in grado di ridimensionare le immagini alle dimensioni di 224×224 pixel e di normalizzarle. La tecnica di normalizzazione è una tecnica necessaria durante la fase di pre-elaborazione dei dati, viene utilizzata per ridimensionale le funzionalità nello stesso intervallo, in modo che possano essere elaborate in modo più accurato dalla rete neurale convoluzionale. Nel nostro caso il range dei dati in ingresso sono stati riportati in [0,1].

Il nostro set di immagini comprende un buon numero di immagini di bassa qualità, questo perché le immagini retiniche non sono sempre facili da acquisire, soprattutto in pazienti anziani, poco collaborativi e/o con difficoltà di fissazione. Inoltre, come sappiamo tra le immagini sono presenti anche retine con atrofie, in cui le componenti anatomiche della retina, come il disco ottico di nostro interesse, non sono facilmente distinguibili dallo sfondo.

Alla fine di questa procedura viene prodotta una matrice (X) con le immagini ridimensionate e scalate, di dimensione ($N \times 224 \times 224 \times 3$), dove N è il numero delle immagini rispettivamente di train, validation e test set, un vettore (y) con la *ground truth* (0 = assenza disco ottico, 1 = presenza disco ottico), di dimensione ($N \times 1$).

3.4 Realizzazione dei modelli

Per la realizzazione dei modelli si è deciso di partire da zero, prendendo spunto da modelli già noti in letteratura per affrontare problemi simili, e di applicare ad essi opportune modifiche per adattarli al nostro caso. In un secondo momento siamo passati al *transfer learning*. I risultati di *sensitivity*, *specificity* e *accuracy* sono stati calcolati prendendo come classe positiva la classe Presenza.

3.4.1 Modello 1

Il primo modello che è stato preso in considerazione è quello proposto in [3], realizzato con lo scopo di individuare il disco ottico in modo automatico in diverse immagini che lo contengono. A sua volta l'architettura di questo modello è stata realizzata sulla base di un algoritmo ideato per la *face detection*. Il modello è una CNN, composto da 13 *layer*

convoluzionali, 5 di *pooling*, 3 *fully-connected* e 1 *softmax*. Il modello è stato realizzato in Python, scegliendo come iperparametri: un *learning rate* di 0.001; la dimensione dei *batch* è di 64, che abbiamo scelto perché uno dei valori che comunemente viene utilizzato, siamo quindi partiti da questa impostazione. Questo modello risulta particolarmente complesso per il nostro *task*, dovuto alla presenza di una quantità enorme di parametri trainabili che risultano essere più di 36 milioni.

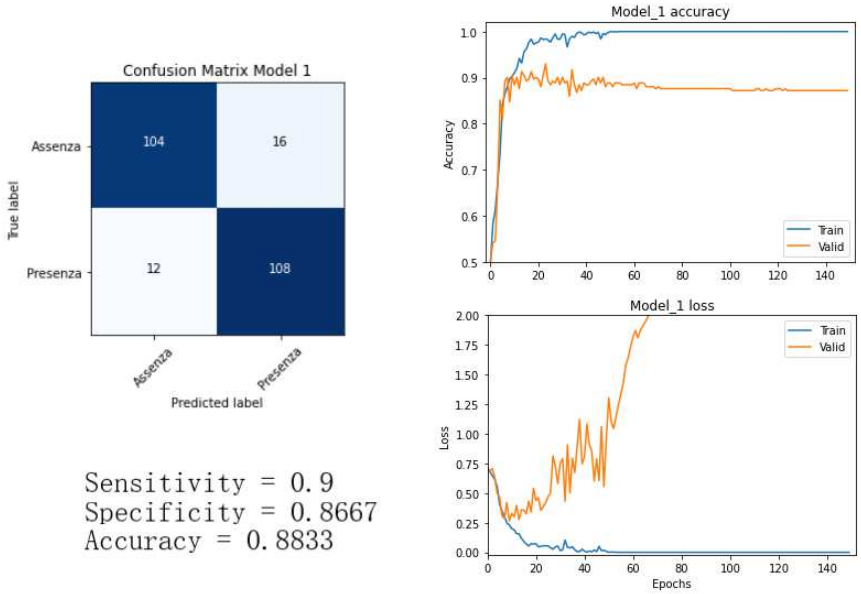


Figura 3.10: Risultati del modello 1

In Figura 3.8 abbiamo riportato le curve di *accuracy* e *loss* del training set e del validation set che ci consentono di valutare le prestazioni del modello durante l'addestramento. La curva di colore blu identifica i valori di *accuracy* e *loss* che ha ottenuto il training set ad ogni epoca, la curva arancio invece è identificativa dei valori di *accuracy* e *loss* del validation set in funzione delle epoche. Abbiamo fissato il numero di epoche a 140 perché già dalle prime 40 epoche le curve mostravano una situazione di *overfitting*, che abbiamo voluto verificare nelle epoche successive. Dai grafici di accuratezza e *loss* possiamo vedere un forte problema di *overfitting*, che si manifesta già dalle prime venti epoche con la divergenza delle curve del validation set. In particolare la curva *loss* del validation set assume una forma esponenziale, arrivando ad assumere valori di perdita molto elevati rispetto a quelli assunti dalla curva *loss* del training set. Anche la curva di *accuracy* del validation set presenta un andamento diverso da quello del training set, essa infatti ha un andamento crescente nelle prime epoche e poi decrescente, non in linea con dei buoni risultati di validazione della rete. Otteniamo un'*accuracy* inferiore allo 0.9. Con lo scopo di migliorare la situazione di divergenza abbiamo provato tecniche di regolarizzazione come il *dropout* ottenendo scarsi risultati. Concludiamo che il modello in esame risulta essere troppo complesso per il nostro task e la nostra tipologia di dati e per questo motivo tale modello viene abbandonato.

3.4.2 Modello 2

Il secondo modello utilizzato è stato realizzato partendo dall'architettura del primo modello di [1]. Tale architettura risulta notevolmente più semplice rispetto al primo modello, anche il numero di parametri risulta nettamente inferiore, nel modello 2 il numero di parametri trainabili è circa 1 milione e mezzo. La rete è composta da un totale di 6 *layer* convoluzionali, 3 *layer* di *maxpooling*, due dei quali si alternano ai primi due *layer* convoluzionali e il terzo segue 3 *layer* convoluzionali messi in serie, la rete conclude con 3 *dense layer*. I valori degli iperparametri sono stati impostati scegliendo un *learning rate* di 0.0001 e valore di *batch size* a 64. Il training del modello è stato fatto per N= 140 epoche, che abbiamo scelto per verificare la presenza del problema dell'*overfitting* e per rendere i risultati del modello confrontabili con quelli del modello 1.

Presentiamo di seguito i risultati che abbiamo ottenuto allenando il modello con i nostri dati

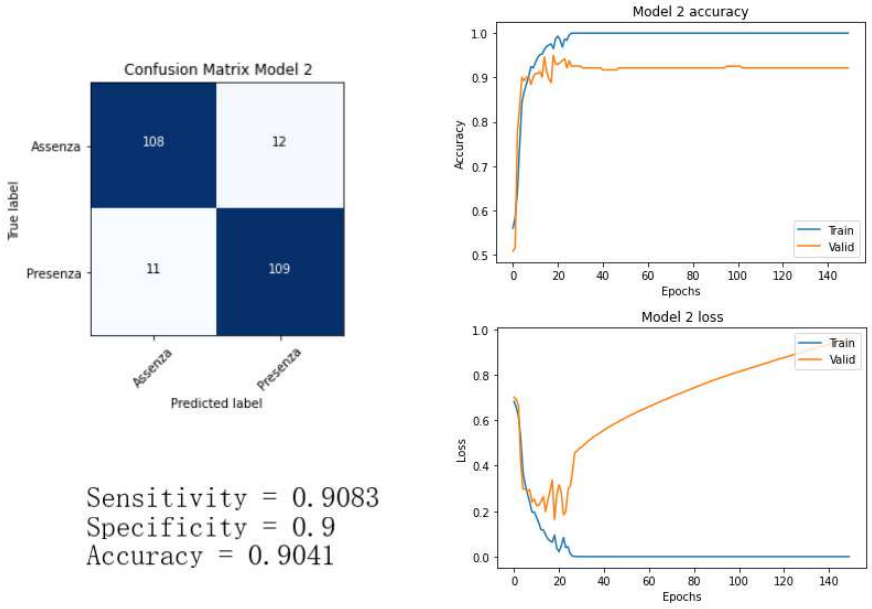


Figura 3.11: Risultati del modello 2

Come possiamo vedere dai grafici riportati la divergenza della funzione loss è notevolmente mitigata, probabilmente per via della ridotta complessità del modello presentato. La curva dell'*accuracy* si stabilisce a un valore leggermente più alto rispetto al modello 1, pur non raggiungendo risultati migliori per quanto riguarda la matrice di confusione, che rimane pressochè invariata. Il modello 2 ottiene quindi valori migliori rispetto al modello 1. A seguito della fase di addestramento della rete, i pesi migliori risultano essere quelli dell'epoca 22.

3.4.3 Modelli di transfer learning

In seguito ai risultati riscontrati nei precedenti modelli si è deciso di passare ad un approccio diverso, optando per l'utilizzo di modelli di *transfer learning* preaddestrati. Abbiamo già visto come questa tecnica sia estremamente utile per la realizzazione di modelli, soprattutto quando non si hanno abbastanza dati in ingresso per un training da zero.

I modelli che sono stati utlizzati sono VGG16, ResNet50, ResNet101 e GoogLeNet. Per ognuno di questi modelli abbiamo deciso di utilizzare un *learning rate* dinamico, che parte da un valore di 0.001 e che si riduce di un fattore 0.5 al raggiungimento del plateau di accuratezza della curva della validation; inoltre sono stati provati diversi valori di *batch size* e diversi metodi di ottimizzazione. Abbiamo testato tre diversi *optimizer*: Adam, SGD e Adagrad, che soni tra i più comunemente utilizzati nei problemi di classificazione.

Uno dei modelli che abbiamo utilizzato è stato VGG16, al modello sono stati modificati gli ultimi livelli a cui sono stati aggiunti 3 *layer dense* con 50, 20 e 2 filtri per adattare il modello agli output di questo *task*. Il modello ottenuto si compone di un totale di quasi 16 milioni di parametri di cui poco più di un milione trainabili. Per questo modello abbiamo utilizzato tre diversi otimizzatori partendo da Adam, proseguendo con SGD e Adagrad. Presentiamo i risultati ottenuti dall'*optimizer* Adam

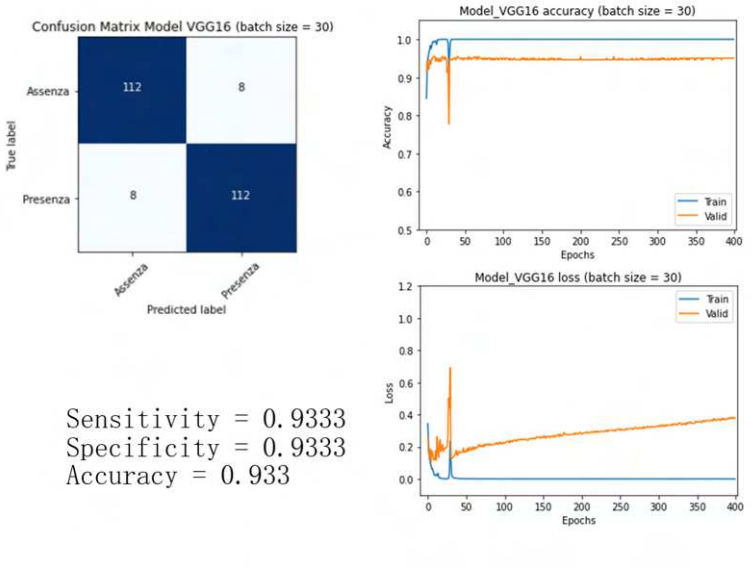


Figura 3.12: Risultati del modello VGG16 con optimizer Adam

Con il primo *optimizer* abbiamo ottenuto dei buoni risultati per la matrice di confusione, mentre per le curve di accuratezza è presente la divergenza della funzione di perdita, seppur in maniera molto attenuata rispetto ai primi due modelli. A tal proposito abbiamo provato l'inserimento di uno strato di *Dropout* per cercare di eliminare la divergenza ma con scarsi risultati.

Siamo passati quindi ad un altro tipo di ottimizzatore, SGD, mantenendo gli stessi strati finali della rete e gli stessi iperparametri. Per questo modello vogliamo presentare i risultati che sono stati ottenuti valutando diversi valori di dimensione dei *batch*.

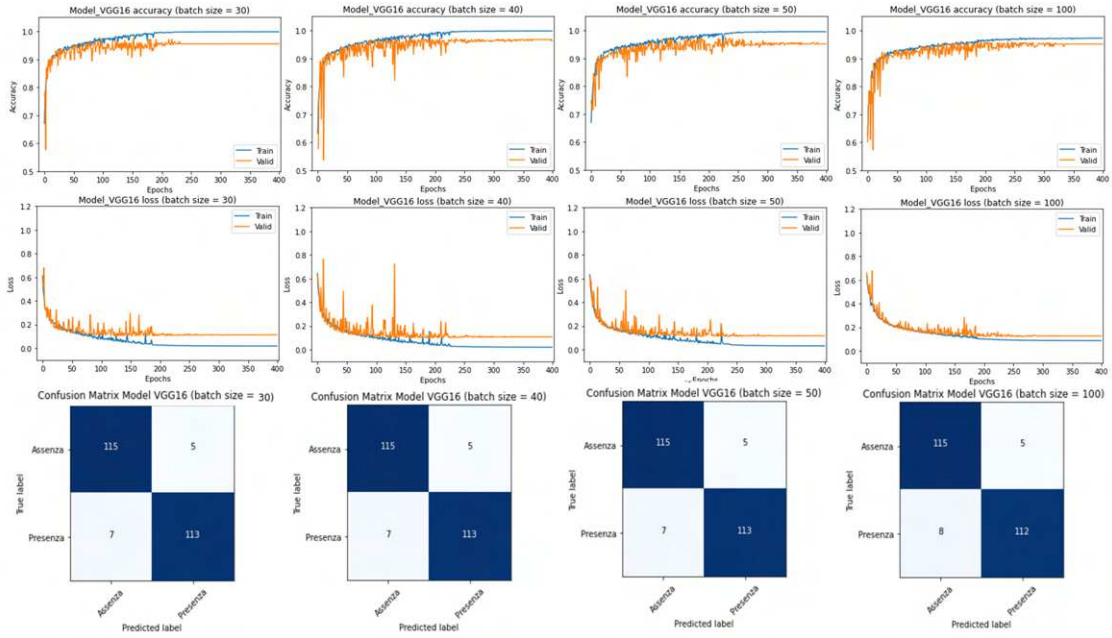


Figura 3.13: Risultati del modello VGG16 con optimizer SGD

Possiamo vedere dalle curve riportate che il valore della dimensione dei *batch* influenza molto l'andamento delle curve, esse infatti raggiungono valori di accuratezza maggiori durante l'addestramento con un *batch size* inferiore. In generale i valori assunti dalle curve del validation non sembrano avere andamento diverso nelle diverse casistiche, seppur avendo una variabile rumorosità delle curve.

Infine abbiamo implementato il modello con *optimizer* Adagrad, riportiamo anche in questo caso i risultati ottenuti assumendo valori diversi di *batch size*

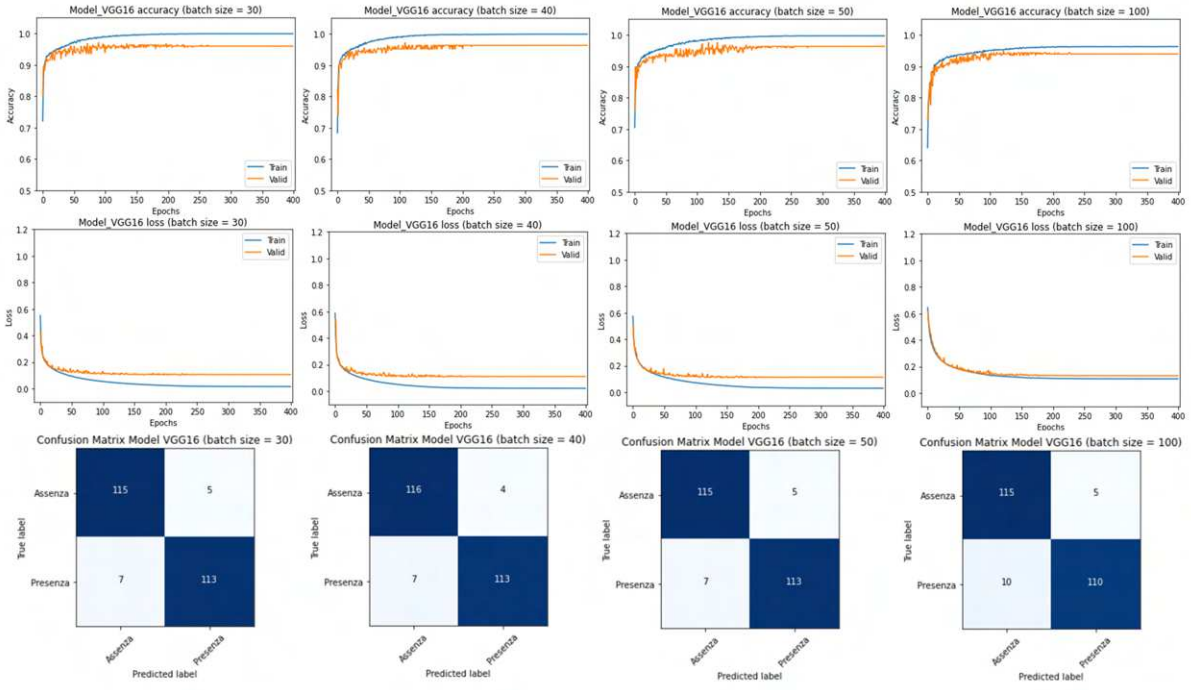


Figura 3.14: Risultati del modello VGG16 con optimizer Adagrad

Le curve sono molto meno rumorose rispetto a quelle ottenute con il precedente *optimizer*,

tuttavia le curve di accuratezza e perdita del validation non sembrano avere andamento diverso. Tale risultato viene ulteriormente confermato in fase di predizione sul test set, le matrici di confusione assumono risultati analoghi alle prove precedenti, con la differenza che a un valore di *batch size* di 100 la rete sbaglia di più in fase di predizione.

Prendiamo ora in considerazione i modelli che hanno ottenuto i risultati migliori, che sono i modelli VGG16 con *optimizer* SGD e Adagrad. Riassumiamo nella seguente tabella i valori di sensitività, specificità e accuratezza ottenuti con *batch size* pari a 40, che abbiamo voluto scegliere come parametro migliore tra i diversi valori provati

	VGG16 con SGD	**VGG16 con Adagrad**
Sensitivity	0.9583	0.9666
Specificy	0.9416	0.9416
Accuracy	0.9500	0.9541

Tabella 3.5: Tabella dei risultati (CNN1)

Esaminiamo nel dettaglio i falsi negativi e falsi positivi del modello VGG16 con *optimizer* SGD, riportati nelle Figure 3.15 e 3.16.

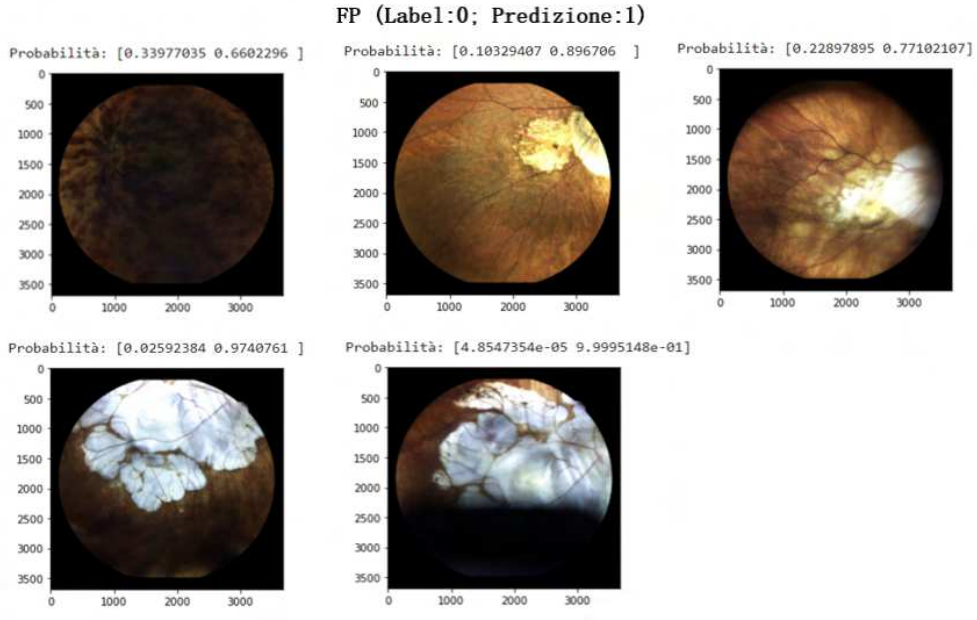


Figura 3.15: Falsi positivi del modello VGG16 con optimizer SGD

Gli errori che la rete commette nella classificazione sono dettati dalla presenza di immagini di retine patologiche, la presenza di macchie di colore chiaro potrebbero dare l'idea della presenza del disco ottico anche se questo non è effettivamente visualizzato nell'immagine.

Viceversa se osserviamo nel dettaglio i falsi negativi (Figura 3.16), possiamo vedere che non tutte le immagini possiedono una luminosità ottima che ci consenta di distinguere chiaramente il disco ottico dallo sfondo. In questi casi in cui non abbiamo un netto contrasto tra il disco ottico e lo sfondo la rete sbaglia la classificazione. Le immagini dove il disco ottico compare solo parzialmente vengono classificate per la maggior parte in modo corretto.

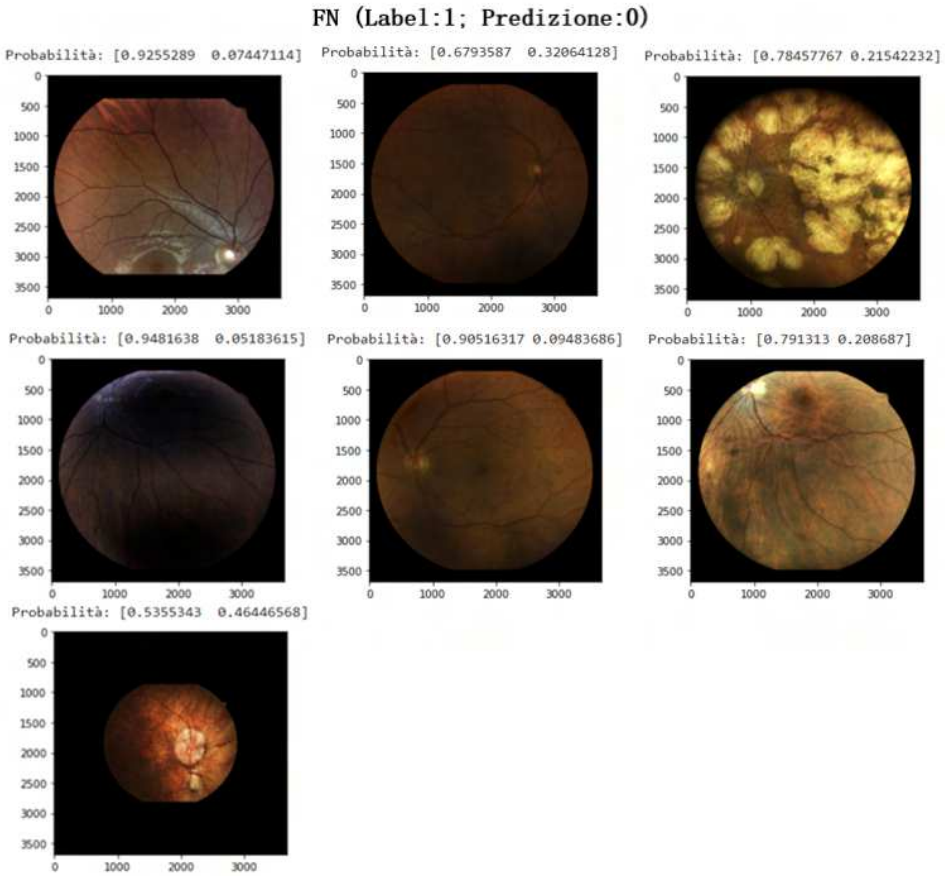


Figura 3.16: Falsi negativi del modello VGG16 con optimizer SGD

Riportiamo di seguito i risultati della predizione ottenuti con modello VGG16 con ottimizzatore Adagrad

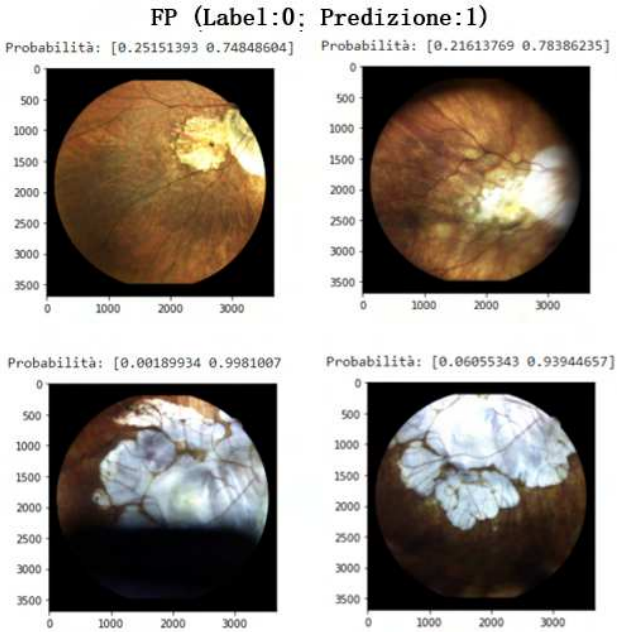


Figura 3.17: Falsi positivi del modello VGG16 con optimizer Adagrad

Notiamo che i falsi positivi di questo modello, pur diminuiti di una immagine, presen-

tano le stesse problematiche del modello precedente. Le immagini in cui la rete sbaglia la classificazione sono causate dalla presenza di retine patologiche e quindi sono considerati errori accettabili.

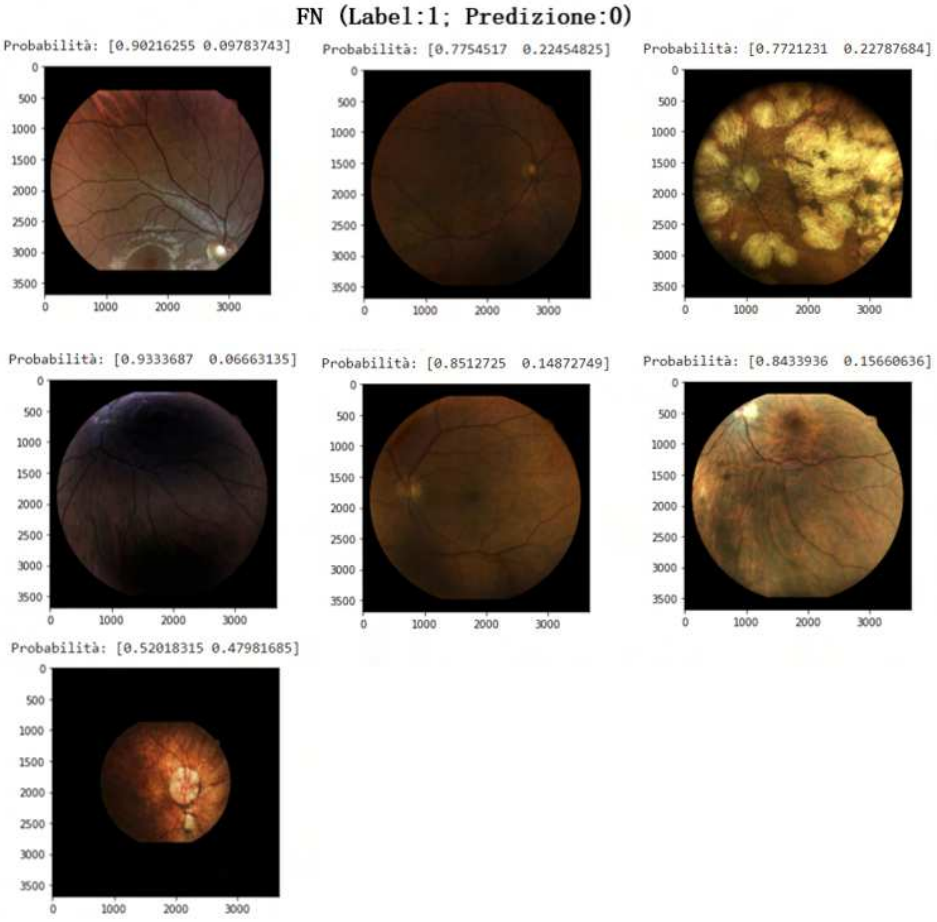


Figura 3.18: Falsi negativi del modello VGG16 con optimizer Adagrad

I falsi negativi rimangono invariati, sono esattamente le stesse immagini nel modello precedente. Notiamo una leggera modifica delle probabilità appartenenti alle due classi, che in questo caso diminuiscono leggermente per la classe di corretta classificazione.

3.4.4 Conclusioni

Il training di una rete è una procedura complessa in termini di numero di parametri da trainare, che tipicamente è molto grande (sopra a 1 milione per i modelli 1 e 2 e modelli di transfer learning presentati) e in termini computazionali di tempo e di spazio. Nel dettaglio, per la realizzazione di questa prima rete è stata di fondamentale importanza l'utilizzo di modelli *transfer learning*, che consentono di ridurre il numero di parametri da calcolare e quindi riducono la possibilità di *overfitting*. Altri aspetti che ci hanno spinto a utilizzare modelli di *transfer learning* sono: la necessità di ridurre il tempo di training set, che diventa piuttosto oneroso (diverse ore) specialmente quando i dati da trainare sono delle immagini e la necessità di gestire un numero considerevole di dati che occupano uno spazio di memoria non indifferente. É indispensabile provare diversi modelli e diverse impostazioni degli iperparametri, poiché non esistono delle regole fisse. Non esiste un

modello migliore in senso assoluto nè esistono delle combinazioni di iperparametri che vanno bene per tutti i problemi. E' necessario individuare l'architettura della rete e il set di iperparametri che meglio si adattano al problema in esame. La concretizzazione del modello che meglio esegue il *task* deriva da sperimentazione.

I risultati di errata classificazione (Figure 3.15, 3.16, 3.17 e 3.18) sono da attribuire alla presenza di aree atrofiche che rendono il disco ottico difficilmente localizzabile, inoltre anche la qualità delle immagini ha influito sui risultati di classificazione. Il dataset è composto anche da immagini di qualità non ottima, che presentano delle opacità e zone molto scure che possono portare ad una errata classificazione dell'immagine.

Nel nostro caso il modello che meglio realizza la classificazione è VGG16 con ottimizzatore Adagrad.

Per migliorare ulteriormente i risultati bisognerebbe aumentare il numero di immagini del training set con caratteristiche simili a quelle che ora vengono classificate erroneamente.

Capitolo 4

CNN 2 : Individuazione del centro del disco ottico

4.1 Raccolta dati e labeling

La seconda rete convoluzione ha l'obiettivo di individuare il centro del disco ottico su immagini retiniche a colori che lo contengono. Per la realizzazione di questa CNN abbiamo preso in considerazione tutte le immagini che abbiamo classificato come Presenza. Prima di procedere al loro utilizzo, vengono eliminate dal dataset le immagini di bassa qualità (individuate da un *grader* umano) per ottenere un dataset di training di buona qualità che permetta di apprendere al meglio il task di individuazione del centro del disco.

Al termine di questa procedura abbiamo ottenuto la seguente distribuzione delle immagini

DEVICE	IMMAGINI PRESENZA
Eidon	10996
Compass	1918
DRSplus	3745
TOTALE	16659

Tabella 4.1: Tabella dei dati disponibili

Dalla tabella 4.1 osserviamo che le immagini che abbiamo a disposizione per l'addestramento e la predizione della rete sono di molto superiori a quelle che avevamo a disposizione per la CNN1, per la quale abbiamo dovuto eseguire l'operazione di *unsampling* per bilanciare le classi di Presenza e Assenza.

Passiamo alla fase di etichettatura. Essa comprende l'individuazione del centro del disco ottico, questa procedura è stata eseguita manualmente per ogni immagine con l'utilizzo del software Matlab. Il centro del disco viene individuato grazie alle sue coordinate x ed y del pixel relativo al centro, che sono state salvate man mano in un dataframe. L'individuazione del centro però non basta, pensiamo infatti alla fase di predizione della rete, quando la rete ci restituirà come predizione un nuovo punto ritenuto il centro del disco, occorrerà disporre di una variabile che ci aiuti a valutare la predizione. Abbiamo quindi pensato di utilizzare il raggio di un cerchio che ha come centro il punto individuato

come il centro del disco, per valutare le predizioni della rete. Il cerchio è stato individuato manualmente per ogni immagine, cercando di costruirlo in modo approssimativo perché contenga il maggior numero di area del disco ottico. É importante sottolineare che l'obiettivo della seconda rete convoluzionale non è la segmentazione del disco ottico, quest'ultimo richiede infatti un *labeling* più preciso che deve essere necessariamente fatto da un clinico. Per il nostro progetto il cerchio è una misura che permetterà di valutare la predizione della rete neurale convoluzionale e verificare se tale predizione sia o meno contenuta in tale area. Sarebbe stato più semplice determinare un raggio costante applicabile ad ogni disco, ma non sarebbe stato veritiero per via della variabilità di dimensione e forma del disco ottico che troviamo nelle diverse immagini, ne riportiamo alcuni esempi

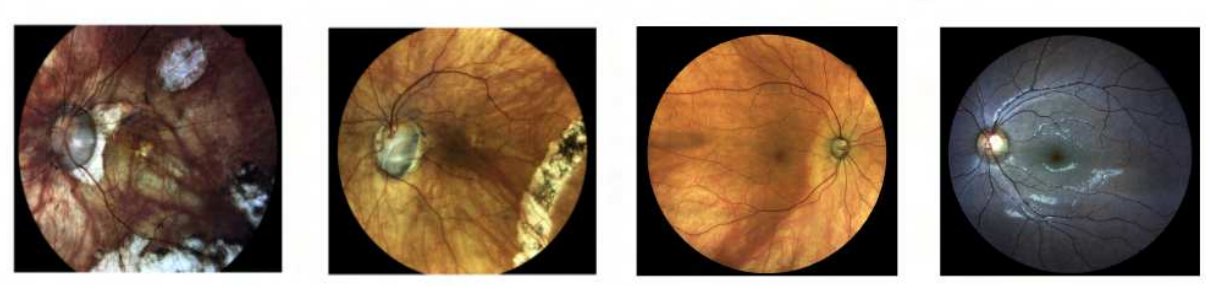


Figura 4.1: Esempi di variazioni del disco per forma e dimensione

Da queste immagini possiamo vedere che il disco ottico non assume sempre una forma univoca, possiamo trovare infatti forme ovali più o meno allungate e di diverse dimensioni, questo perché la morfologia della papilla ottica dipende fortemente da caratteristiche genetiche e dalla condizione sanitaria del paziente. Avendo a disposizione un dataset di immagini di molti pazienti e non tutti sani, è allora indispensabile definire per ogni immagine un cerchio personalizzato centrato con il centro del disco ottico.

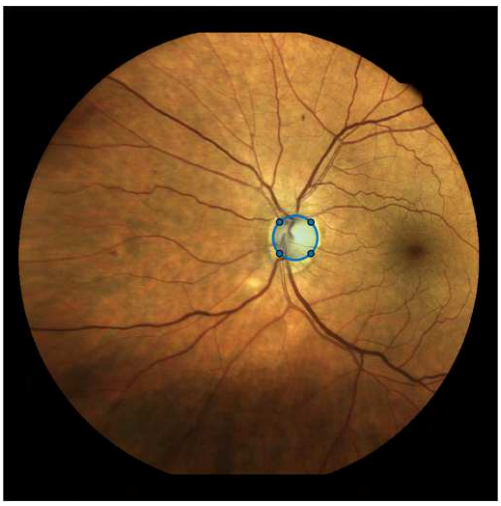


Figura 4.2: Labeling del disco

Per semplicità l'operazione di etichettatura è stata fatta su immagini a cui abbiamo già applicato l'operazione di bordatura per omogeneizzare le dimensioni delle immagini, come abbiamo fatto per la rete convoluzionale precedente. Le immagini in fase di *labeling* non sono riscalate, questo perché la riscalatura avrebbe reso le immagini molto più piccole dell'originale e quindi sarebbe stato molto più difficile tracciare il cerchio che contiene il

disco ottico. Sarà necessario utilizzare un fattore di scala per adattare le coordinate alla riscalatura dell'immagine.

Alla fine di questa procedura otteniamo un dataframe dove ogni immagine è stata salvata nel seguente modo

ID_EYE	DEVICE	IMG	PATIENT	EYE	FIELD	OUTPUT	XPIX	YPIX	RADIUS
1	Eidon	Study_001- ...-.jpg	1	right	central	1	2774,47	1738,01	136,30
2	Eidon	Study_001- ...-.jpg	2	right	nasal	1	1672,80	1754,32	151,03
...	...	...	...	...	...	...	...	...	...
140	Compass	exam_1535- ...-.jpg	1535	Na	central	1	2440,63	1807,89	78,57
...	...	...	...	...	...	...	...	...	...
842	DRSplus	ffc424f4- ...-.jpg	ffc42	right	nasal	1	1749,66	1854,47	176,04

Tabella 4.2: Dataframe per la CNN2

Come possiamo vedere dalla tabella abbiamo ritenuto opportuno tenere traccia del dispositivo di derivazione delle immagini, questo per valutare in seguito i risultati della rete rapportati anche al *device* di provenienza. Inoltre le immagini hanno anche associato un numero identificativo del paziente a cui appartiene l'immagine (PATIENT) e un numero identificativo dell'occhio (ID_EYE), queste informazioni ci serviranno nella fase di *split* dei dati.

4.2 Split dei dati

A seguito della pulizia dei dati e della loro etichettatura, troviamo l'operazione di *split*, in cui, come per la rete precedente, abbiamo diviso le immagini a disposizione nei tre set: train, test e validation.É importante far notare che questa operazione deve essere fatta rispettando l'unicità dei tre diversi set, cioè le immagini che compaiono in un *subfolder* non possono comparire in un altro. Ricordiamo infatti che le immagini non sono del tutto uniche, alcune di esse sono immagini dello stesso occhio acquisito in campi diversi. Per questa ragione è necessario che le immagini appartenenti allo stesso occhio siano contenute nello stesso set e quindi è importante tenere traccia dell'occhio di provenienza delle immagini durante la fase di *labeling*. Per farlo abbiamo lavorato singolarmente ogni set, assegnando ad ogni occhio un ID, memorizzato nel dataframe del *labeling* sotto la colonna titolata ID_EYE. In questo modo sarà più facile nella divisione delle immagini riconoscere se due immagini appartengono o meno allo stesso occhio. Abbiamo quindi diviso le immagini cercando di rispettare la proporzione 80:10:10 e di assegnare immagini appartenenti allo stesso occhio in un set comune.

Al termine di questa procedura abbiamo ottenuto la seguente distribuzione dei dati

DEVICE	TRAINING	TEST	VALIDATION
Eidon	8891	1122	983
Compass	1538	190	190
DRSplus	3055	324	366

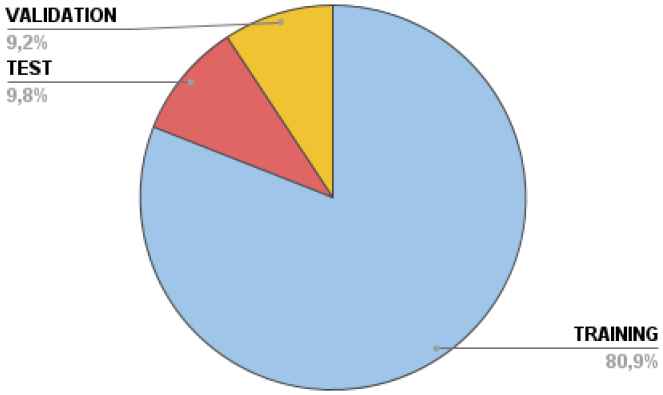


Figura 4.3: Distribuzione delle immagini nei set (training, test e validation)

Riportiamo inoltre gli aerogrammi che ci fanno capire la distribuzione delle immagini nei tre set (training, test e validation) secondo il dispositivo di appartenenza, questa distribuzione ci sarà utile in fase di valutazione dei risultati

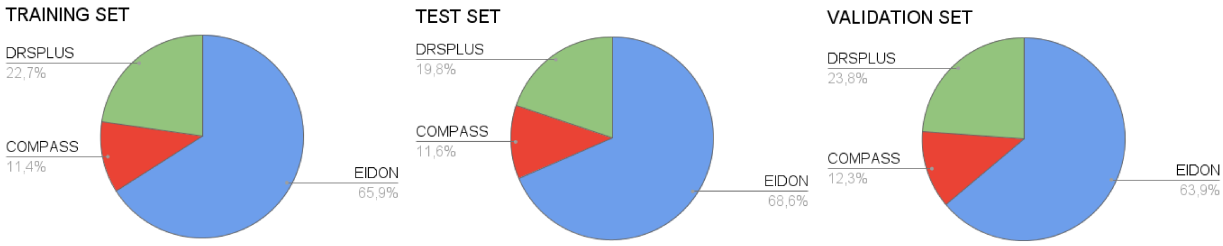


Figura 4.4: Distribuzione delle immagini per device

La presenza di immagini appartenenti ad ogni dispositivo è omogenea in training, test e validation set.

4.3 Caricamento dei dati

La procedura di caricamento dei dati è stata fatta in Colab, utilizzando come per la prima rete convoluzionale, la funzione *ImageDataGenerator* di Keras, che ci permette di ridimensionare le immagini alle dimensioni di 224×224 pixel e di normalizzarle. Alla funzione viene dato in ingresso anche la colonna del dataframe relativa alle coordinate del centro del disco, che provengono della fase di etichettatura delle immagini. Ricordiamo inoltre che il valore del raggio ci servirà solamente in fase di analisi dei risultati e quindi non fa parte dei *labels* delle immagini.

Alla fine di questa procedura viene prodotta una matrice (X) con le immagini ridimensionate e scalate, di dimensione $(N \times 224 \times 224 \times 3)$, dove N è il numero delle immagini rispettivamente di training, test e validation set, un vettore (y) con la *ground truth* [x_pix, y_pix], corrispondenti alla localizzazione del pixel centrale del disco.

4.4 Realizzazione dei modelli

Per questo task abbiamo pensato di procedere con la realizzazione di modelli *"from scratch"* per poi passare a modelli presenti in letteratura.

Poiché la CNN1 esegue un task di classificazione mentre la CNN2 esegue un task di regressione, per i modelli che seguono abbiamo utilizzato una metrica diversa da quella utilizzata per la CNN1. Nel dettaglio parleremo del coefficiente di determinazione (R^2), utilizzato in statistica per misurare il legame tra la variabilità dei dati e la correttezza del modello statistico utilizzato. Il coefficiente di determinazione è espresso nel seguente modo

$$R^2 = 1 - \frac{\sum_{m=1}^{M}(\hat{y}_m - y_m)^2}{\sum_{m=1}^{M}(\bar{y}_m - y_m)^2} \tag{4.1}$$

dove $\hat{y}_m$ è il centro predetto del disco sottoforma di vettore $[x_{pix}, y_{pix}]$, $\bar{y}_m$ è il valore medio, e y_m rappresenta la *ground truth* del centro del disco. Valori di R^2 vicino a 1 indicano piccoli errori di predizione del centro del disco.

4.4.1 Modello 1

Il primo modello che ha dato dei risultati significati è quello estratto dall'articolo [2] in bibliografia. Il modello presentato dall'articolo è stato realizzato con lo scopo di rilevare le coordinate del centro e il relativo raggio del disco solare usando una rete convoluzionale. Poiché il nostro interesse principale è quello di individuare il centro del disco ottico abbiamo preso in considerazione tale modello, escludendo però dalla rete l'informazione sulla predizione del raggio.

L'architettura proposta trae ispirazione dalla VGGNet, dunque composta da tre componenti: livelli di convoluzione, livelli di *pooling* e *fully connected layer*. I livelli di convoluzione sono principalmente utilizzati per l'estrazione delle caratteristiche dell'immagine, mentre i livelli di *max-pooling* vengono usati per ridurre le dimensioni delle *feature map* e ridurre il numero di calcoli. L'*optimizer* scelto per questa rete è Adam.

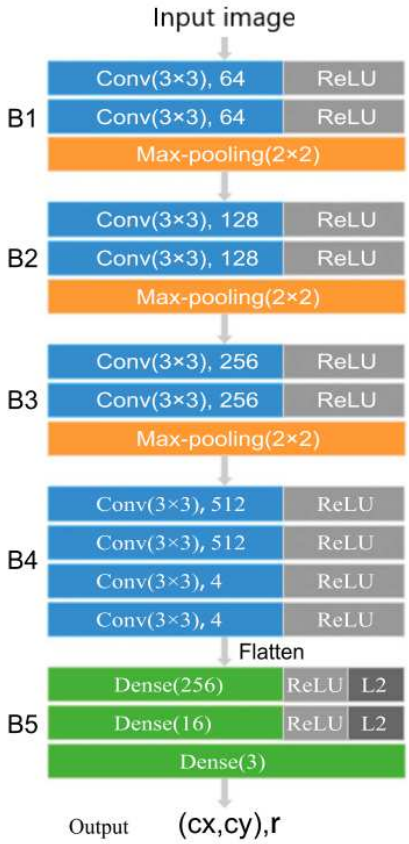


Figura 4.5: Architettura del primo modello

L'architettura della rete convoluzionale proposta comprende una regolarizzazione L2 nei livelli più profondi della rete, con lo scopo di attenuare eventuali effetti dell'*overfitting* dei dati.

L'applicazione di questa rete convoluzionale ha dato i seguenti risultati per le nostre immagini retiniche

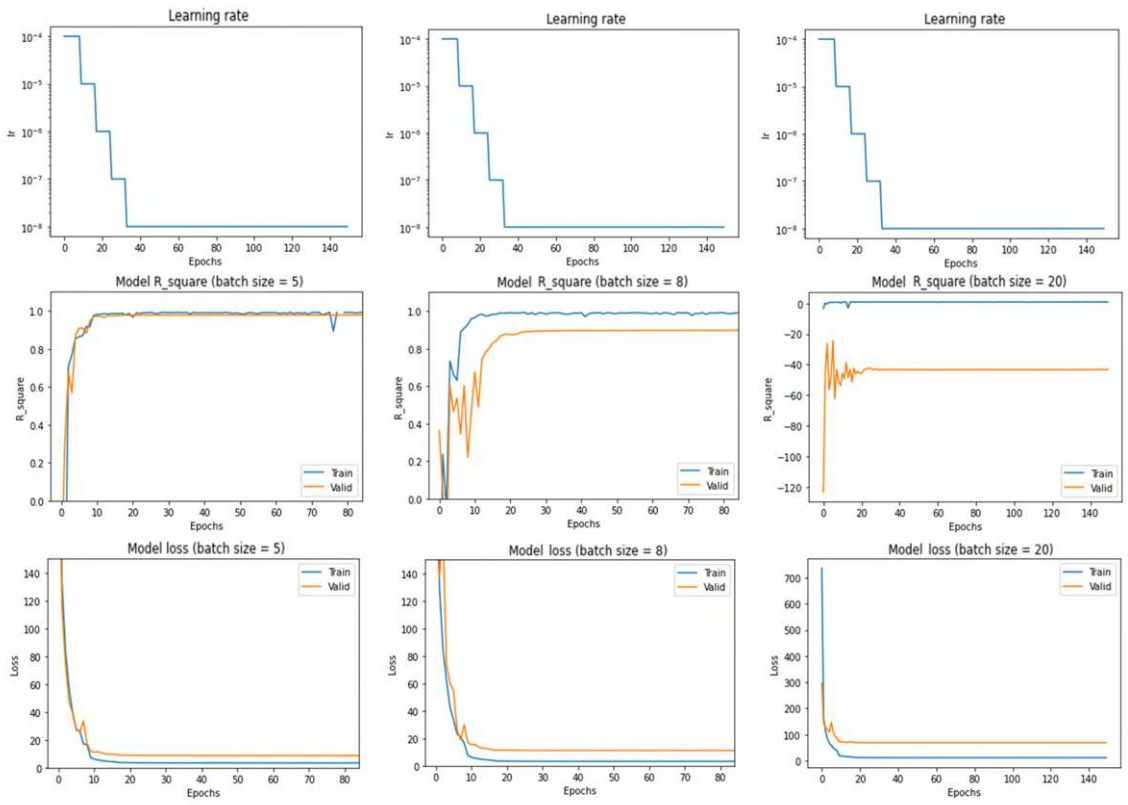


Figura 4.6: Risultati del modello 1

Poiché per la CNN1 avevamo notato una variabilità dei risultati in funzione del valore di *batch size*, abbiamo pensato di testare il modello 1 impostando diverse dimensioni dei *batch*. In Figura 4.6 abbiamo riportato per valori di *batch size* pari a 5, 8 e 20, le curve di training relative a: i valori assunti dal *learning rate*, le curve di R^2 e le relative funzioni *loss*. Per ogni situazione sono state graficate con il colore blu le curve relative al training set, mentre in arancione abbiamo definito le curve del validation set. La variazione tra le curve nei tre valori testati di *batch size* hanno mostrato una veloce variabilità dei risultati all'aumentare della dimensione dei *batch*. La rete è in grado di predire con buona precisione i valori delle coordinate dei centri, ottenendo risultati migliori con valore di *batch size* più piccoli, per quali otteniamo anche delle curve meno rumorose. Il termine di regolarizzazione L2 non elimina del tutto il problema dell'*overfitting*, che si presenta con una saturazione della curva di *loss* e di R^2 del validation set a valori inferiori rispetto a quelli raggiunti dalla curva del training set. La predizione sul test set ha portato i seguenti risultati

	Batch size=5	Batch size=8	Batch size=20
Test loss	11.591	11.720	54.157
Test R^2	0.985	0.984	0.906

Tabella 4.3: Tabella dei risultati di predizione del modello 1

Nonostante le curve di addestramento presentino il problema dell'*overfitting*, i risultati di predizione sul test set hanno raggiunto un valore piuttosto alto per R^2 per il modello con *batch size* pari a 5, confermando le buone prestazioni della rete.

4.4.2 Modello 2

Seguentemente ai buoni risultati raggiunti dal modello precedente, ci siamo chiesti se questi possano essere migliorati apportando delle modifiche all'architettura del primo modello. A tal fine abbiamo pensato di valutare l'influenza del termine di regolarizzazione L2, che è presente negli ultimi livelli della rete del modello precedente. Abbiamo quindi preso la prima rete proposta e abbiamo eliminato il termine di regolarizzazione L2, la restante parte dell'achitettura della rete è rimasta invariata, comprese le impostazioni degli iperparametri del modello.

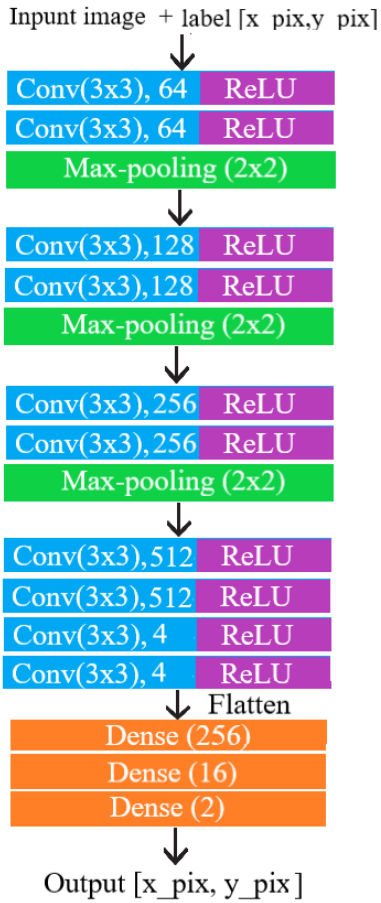


Figura 4.7: Architettura del modello 2

Osserviamo cosa accade con l'assenza del termine di regolarizzazione

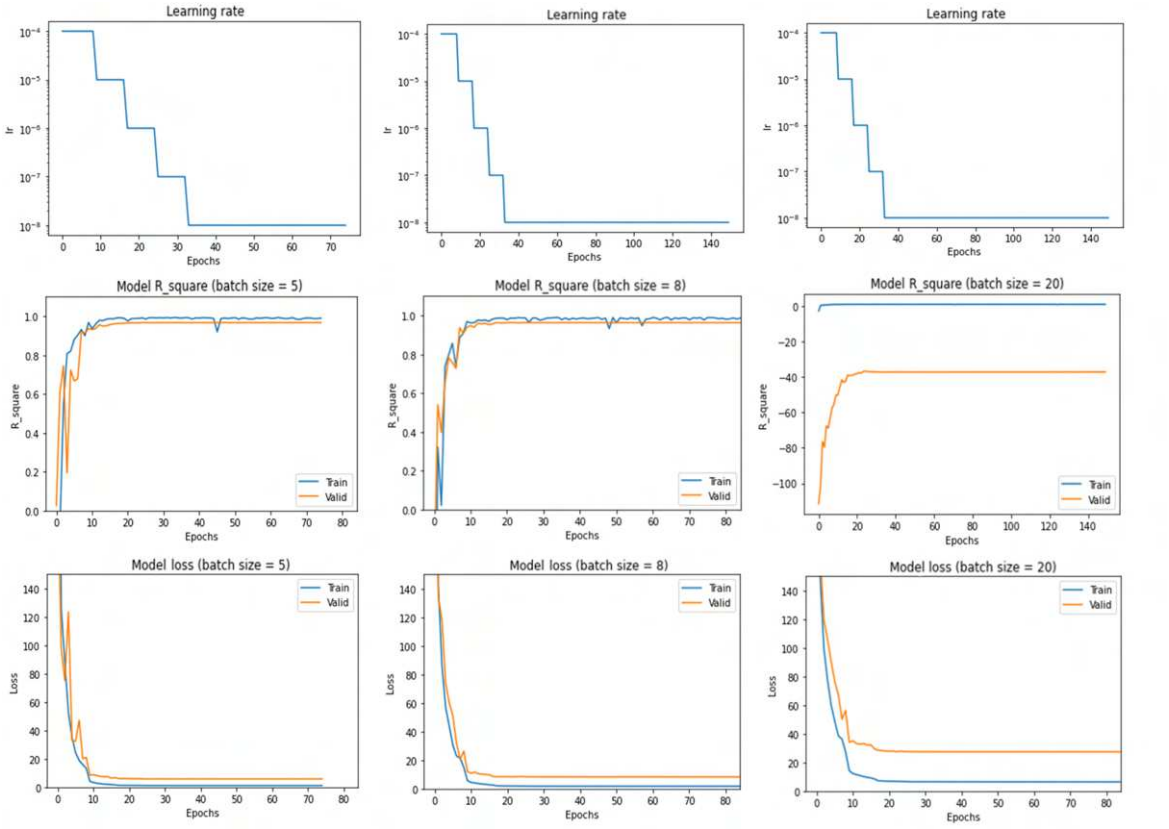


Figura 4.8: Risultati del modello 2

In Figura 4.8 sono stati riportati i risultati ottenuti in fase di training per valori di *batch size* pari a 5, 8 e 20. Le curve in blu definiscono i valori ottenuti per il training set mentre in arancione troviamo i valori relativi al validation set. Visibilmente possiamo notare un peggioramento delle curve all'aumentare del valore di *batch size*, quindi possiamo ritenere che i valori di *batch* da preferire sono i più piccoli. Abbiamo inolte utilizzato, come per la CNN1, un valore di *learning rate* dinamico, che diminuisce man mano che le curve di addestramento raggiungono la saturazione. La scelta del decadimento del *learning rate* durante il training della rete è stata fatta con lo scopo di verificare la presenza di piccoli miglioramenti delle curve di training, quando queste raggiungono una situazione di saturazione. In particolare, abbiamo voluto modificare il valore del *learning rate* monitorando la curva loss del validation set. Il valore di *learning rate* decade di un fattore 10 ogni 8 epoche, fino al raggiungimento del valore minimo di 10^{-8}, come suggerito dal modello originario proposto nell'articolo [2] in bibliografia. Questa assunzione è uno dei metodi comunemente utilizzati per la realizzazione di un *adaptive learning rate*, in grado di aiutare la rete a convergere a un minimo locale della funzione *loss*, evitando oscillazioni. Dai risultati ne segue che l'assenza del termine di regolarizzazione porta ad un miglioramento delle curve di training. Esse infatti raggiungono valori più elevati di R^2 e valori più bassi per la curva *loss*, seppur con la presenza di qualche picco rumoroso.

Applicando tale modello al test set otteniamo i seguenti risultati

	Batch size = 5	Batch size = 8	Batch size = 20
Test loss	6.369	10.250	35.492
Test R^2	0.989	0.979	0.927
Immagini predizione uscente	34 (2,078%)	59 (3,606%)	132 (8,068%)

Tabella 4.4: Tabella dei risultati di predizione del modello 2

I risultati della predizione confermano che i valori di *batch size* da preferire sono i più piccoli poiché ad essi corrispondono dei risultati migliori anche a livello predittivo. Scegliamo quindi il modello con *batch size* pari a 5 e riportiamo alcuni dei risultati di predizione, in cui il centro di *ground truth* è indicato dal simbolo + verde, il centro predetto dalla rete è definito dal simbolo × rosso e il cerchio giallo indica il cerchio individuato durante la fase di labelling.

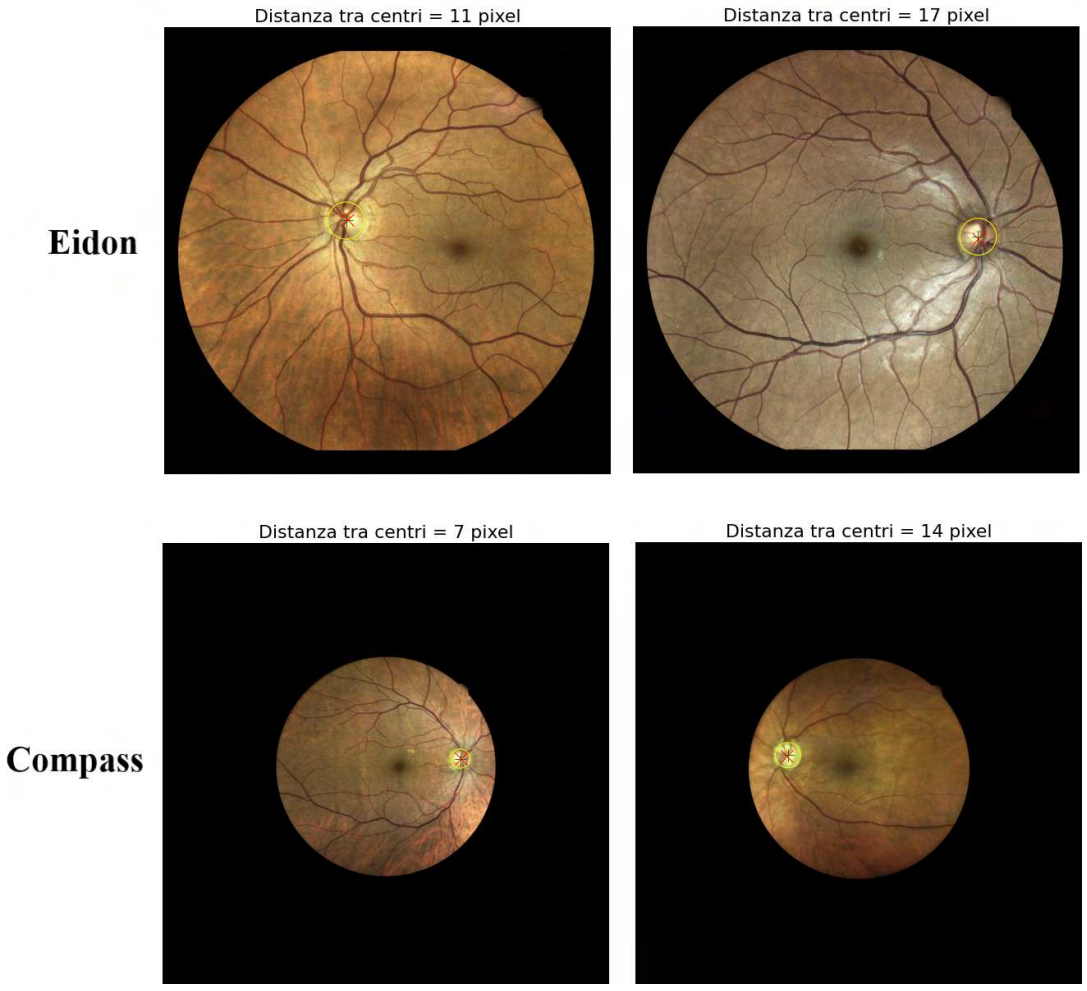


Figura 4.9: Risultati del modello 2

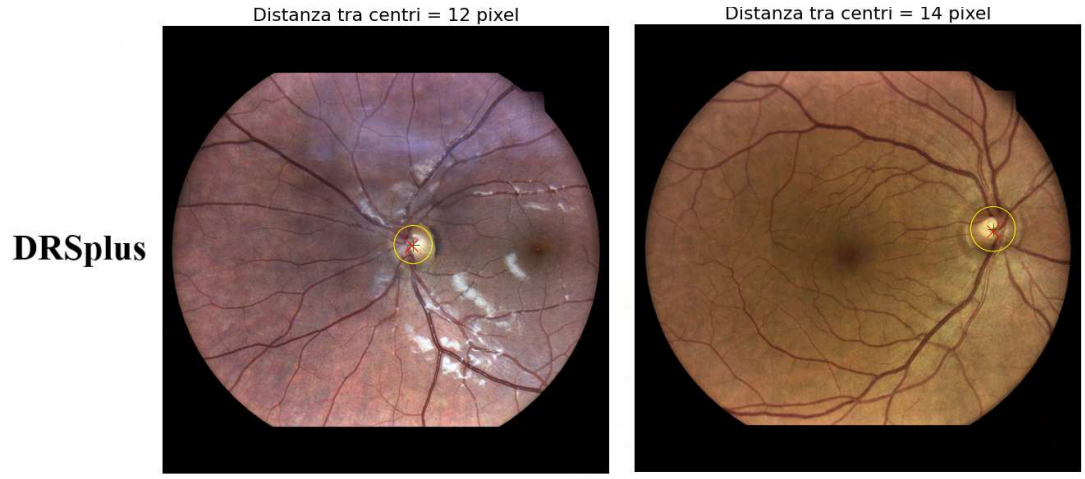


Figura 4.10: Risultati del modello 2

In generale il modello riesce a predire bene le coordinate del centro del disco ottico sulle immagini di tutti i *device*, risultando particolarmente preciso nelle immagini in cui le componenti anatomiche sono ben distinguibili. Osservando i risultati di predizione possiamo dire anche che il modello sembrerebbe aver compreso che il disco sia localizzato nella porzione di retina più chiara e dove si addensano i vasi sanguigni più spessi.

Sebbene il modello risulti soddisfacente, non significa che sia privo di errori. I risultati delle curve di Figura 4.8 vengono rispecchiati anche nell'ultima riga della Tabella 4.4, in cui abbiamo inserito una riga relativa al numero di immagini in cui la predizione è "uscente", cioè immagini in cui il punto predetto come centro del disco ottico esce dal cerchio definito in fase di *labeling* delle immagini. Il modello che ci restituisce il numero minore di queste immagini è proprio il modello con valore di *batch size* pari a 5, in cui la percentuale di immagini con predizione uscente costituisce solamente il 2% delle immagini del test set.

Vediamo nel dettaglio le immagini in cui il centro predetto esce dal raggio contenente il disco. Prendiamo dunque in considerazione il modello con *batch size* pari a 5, che ci ha dato le prestazioni migliori e analizziamone i risultati. Facendo una analisi dettagliata delle immagini con predizione uscente osserviamo che esse si distribuiscono nel seguente modo tra i dispositivi.

IMMAGINI PREDIZIONE USCENTE		
Eidon	Compass	DRsplus
21	6	7
1,87 %	3,15 %	2,16 %
TOTALE IMMAGINI		
34		

Tabella 4.5: Tabella delle immagini con predizione uscente

Durante questa analisi riteniamo sia importante ricordare che le immagini utilizzate sono costituite anche da retine patologiche. In generale tra le immagini a nostra disposizione abbiamo individuato la presenza di aree atrofiche, dove con questo termine vogliamo indicare porzioni della retina caratterizzate da macchie di colore più chiaro, che possono essere facilmente scambiate dalla rete per il disco ottico, oppure aree che limitano

la definizione precisa del disco ottico. Nella totalità del test set abbiamo individuato la seguente composizione di immagini con aree atrofiche

Totale immagini con aree atrofiche nel test set		
69 (4.22% sul totale del test set)		
Eidon	Compass	DRsplus
53	4	12

Tabella 4.6: Tabella delle immagini con aree atrofiche nel test set

Le immagini con aree atrofiche costituiscono solamente il 4.22% del totale delle immagini del test set. Sebbene questa percentuale sia molto bassa, vedremo di seguito, che sono proprio queste immagini a occupare la maggior parte delle immagini che abbiamo definito nella Tabella 4.5.

Nel dettaglio osserviamo che le immagini con predizione uscente (Tabella 4.5) sono composte maggiormente da immagini derivati dal dispositivo Eidon, ricordiamo però che il nostro set di dati è composto dal 65% di immagini di Eidon, ne segue che se rapportiamo il numero di immagini con predizione errata al numero totale delle immagini di ogni dispositivo otteniamo le percentuali riportate in Tabella 4.5. Le immagini di Compass sono quindi le immagini in cui la rete tende a sbagliare maggiormente la predizione del centro del disco, posizionandolo fuori dal cerchio che definisce il disco. Tale risultato potrebbe essere attribuito alla variazione di dimensione delle immagini di Compass rispetto alle immagini degli altri dispotivi. Ricordiamo infatti che le immagini di Compass originali sono immagini molto piccole rispetto a quelle degli altri due dispositivi, ne segue che in fase di *padding* dell'immagine, proprio le immagini di Compass sono quelle a cui viene aggiunto un bordo nero maggiore. Inoltre le immagini Compass sono presenti in numero molto inferiore, costituiscono infatti solamente l'11,4% del training set, contro il 65,9% di Eidon e il 22,7% di DRSplus.

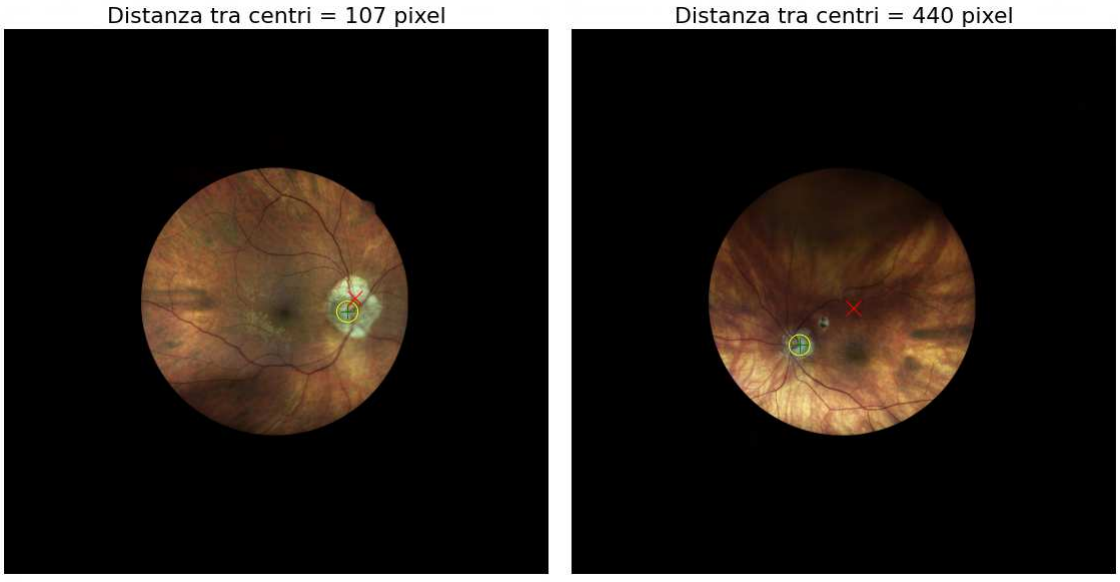


Figura 4.11: Risultati Compass del modello 2

Queste sono alcune delle immagini in cui il centro predetto (× rossa) esce dal cerchio che definisce il disco (cerchio giallo). Possiamo ritenere che tale errore sia attribuito,

come abbiamo detto, alla forte presenza del bordo nero, che ha influenzato la capacità della rete di apprendere da queste immagini la posizione corretta del disco ottico.

Passiamo alle immagini con predizione uscente di Eidon, dove abbiamo identificato il centro di *ground truth* e il centro predetto come abbiamo fatto per le immagini di Compass. Esaminando queste immagini possiamo notare che esse sono costituite per la maggior parte da immagini che presentano aree atrofiche.

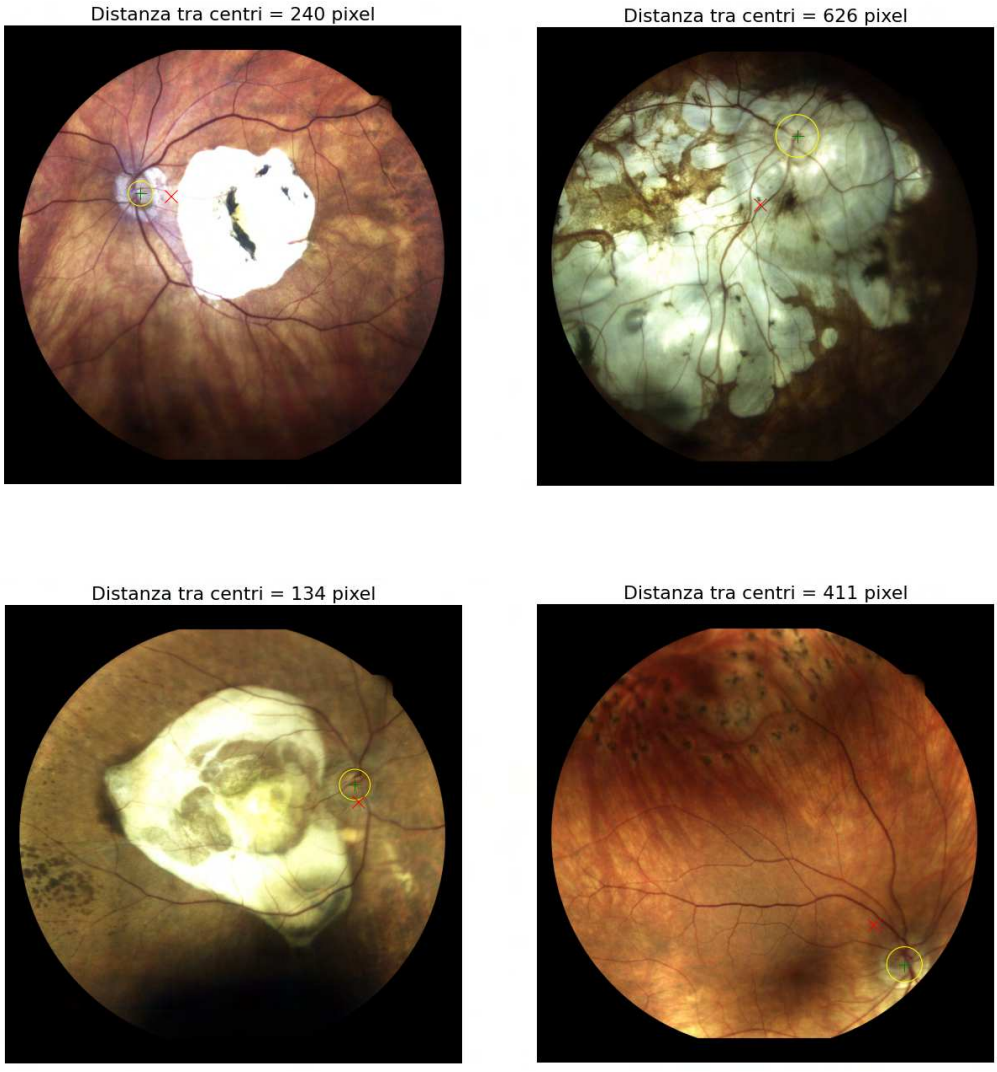


Figura 4.12: Risultati Eidon del modello 2

L'area atrofica non rende il disco ottico facilmente riconoscibile, dunque possiamo attribuire ad esse l'errata predizione del centro del disco nelle immagini del dispositivo Eidon. Riportiamo in fine alcuni esempi di immagini di DRSplus, osservando che anche in queste immagini la presenza di aree atrofiche ha facilmente influenzato la predizione della rete.

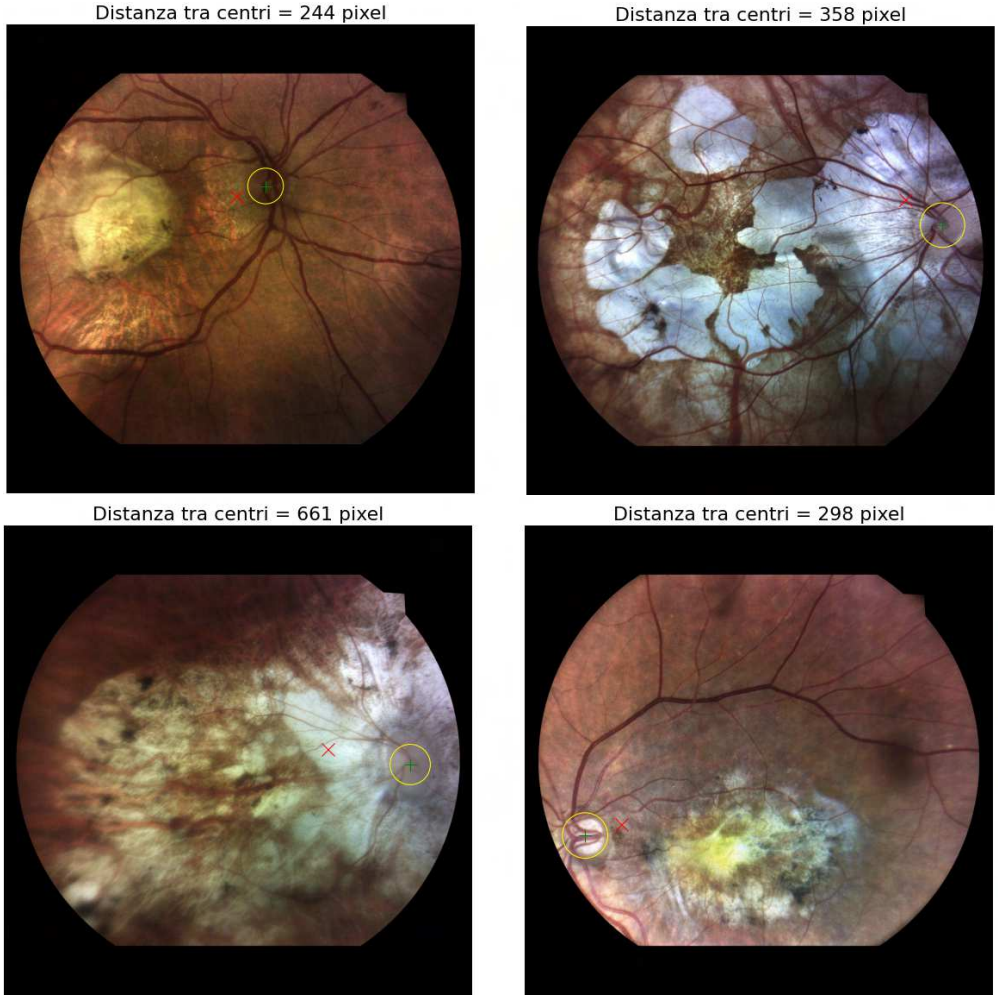


Figura 4.13: Risultati DRSplus del modello 2

In generale possiamo dire che gli errori commessi dal modello 2 (immagini con predizione uscente) sono attribuibili alla presenza di immagini scure e di aree atrofiche che rendono difficilmente riconoscibile il disco ottico. Questo risultato è associabile alla scarsa presenza di immagini con aree atrofiche nel training set.

Totale immagini con aree atrofiche nel training set		
811 (6% sul totale del trainining set)		
Eidon	Compass	DRsplus
702	50	59

Tabella 4.7: Tabella delle immagini con aree atrofiche nel training set

Le immagini con aree atrofiche costituiscono solamente il 6% dell'intero training set (Tabella 4.7), ne segue probabilmente che il modello 2 non ha sufficienti esempi di immagini con aree atrofiche per localizzare precisamente il disco ottico anche in questi casi particolari. Inoltre, tra le immagini con aree atrofiche il 90% dei casi è costituito da immagini con aree atrofiche poco estese, di conseguenza le immagini con aree atrofiche importanti, che coinvolgono quasi la totalità dell'immagine, sono soggette ad una predizione con errore maggiore rispetto alle immagini in cui le aree atrofiche sono poco estese.

Per comprendere meglio dove la rete sbaglia abbiamo cercato di ragionare sui singoli valori delle coordinate del centro del disco (x e y) predette. Abbiamo quindi realizzato i seguenti plot, i quali riportano sull'asse delle ascisse le coordinate del centro del disco (x e y) di *ground truth* di ogni *device* rapportata all'ampiezza e all'altezza rispettivamente dell'immagine del device di appartenenza; sull'asse delle ordinata invece sono state riportate le coordinate x e y predette rapportate anch'esse alle dimensioni delle immagini di appartenenza e differenziandole con colori diversi a seconda del dispositivo di appartenenza. Al fine di rendere confrontabili le misure delle coordinate del centro abbiamo applicato una procedura di normalizzazione sulle coordinate in funzione delle dimensioni originali dell'immagine di appartenenza. Ricordiamo infatti che le immagini acquisite con i *device* Eidon, Compass e DRSplus hanno dimensioni diverse.

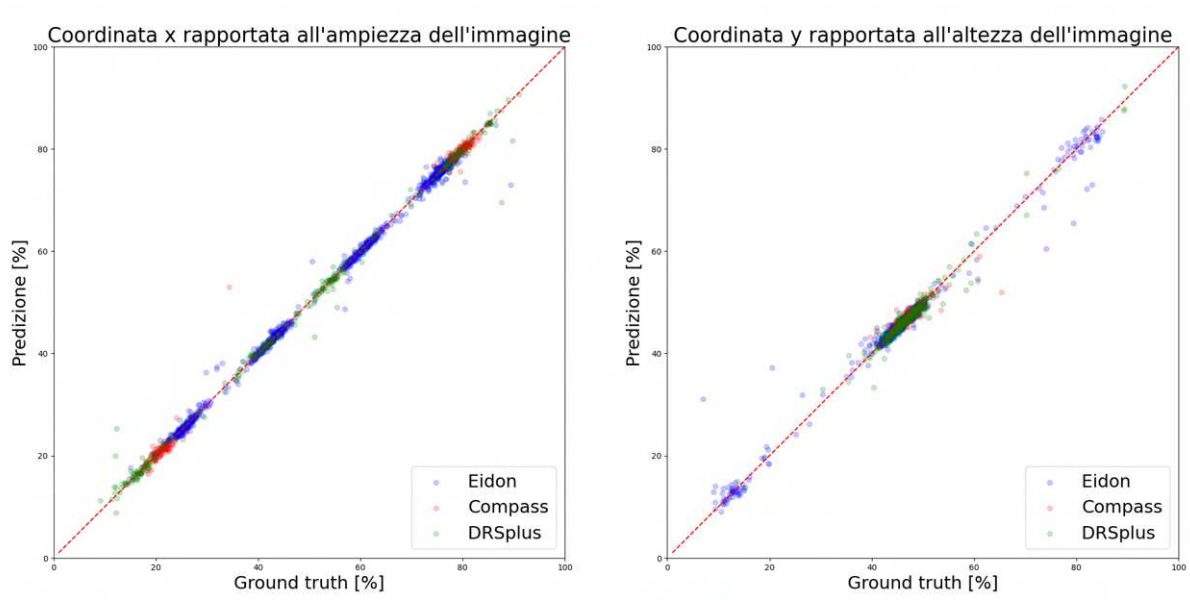


Figura 4.14: Plot delle coordinate a confronto

Se mettiamo a confronto i valori veri della coordinata e la sua predizione (in termini percentuali) notiamo che la maggior parte dei valori si colloca attorno alla linea centrale, che definisce le predizioni che si avvicinano maggiormente ai valori di *ground truth*. I punti collocati lontano dalla linea centrale determinano gli errori della predizione di ciascuna coordinata. Per questi punti possiamo dire che la coordinata y influenza maggiormente la predizione errata del punto che definisce il centro, poiché il suo grafico presenta un maggior numero di punti che si allontanano dalla linea centrale. Un'altra osservazione che possiamo fare è relativa ai dispositivi, i punti infatti sono stati plottati con colori diversi a seconda del dispositivo di appartenenza. In generale sembrerebbero di Eidon i punti che si posizionano lontani dalla linea diagonale, ma associamo questo risultato al numero elevato di immagini Eidon nella totalità dei dati.

Una seconda analisi può essere fatta osservando gli scatter plot delle predizioni e dei punti di *ground truth* che presentiamo di seguito. Anche in questo caso abbiamo apportato una procedura di normalizzazione per i valori x e y di localizzazione del centro predetti e di *ground truth*. Le coordinate x sono state quindi normalizzate sull'ampiezza delle immagini originali, distinguendo i *device* poiché essi acquisiscono immagini con diverse dimensioni l'uno dall'altro. Le coordinate y sono state invece normalizzate sull'altezza

delle immagini originali, distinguendoli in base al *device* di appartenenza dell'immagine. Nei seguenti scatter plot sono stati cerchiati i punti relativi alle immagini per le quali la rete ha predetto, come centro del disco ottico, punti che non sono contenuti nel cerchio di etichetta, dunque i punti che abbiamo definito "uscenti" in fase di analisi dei risultati.

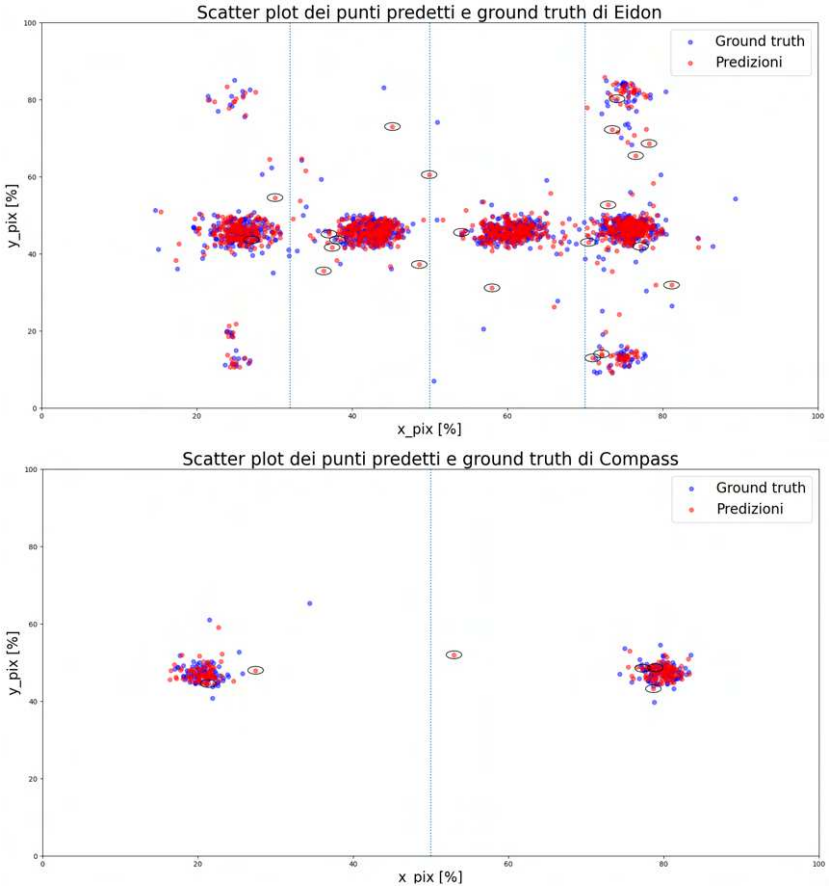


Figura 4.15: Scatter plot dei punti predetti e di ground truth

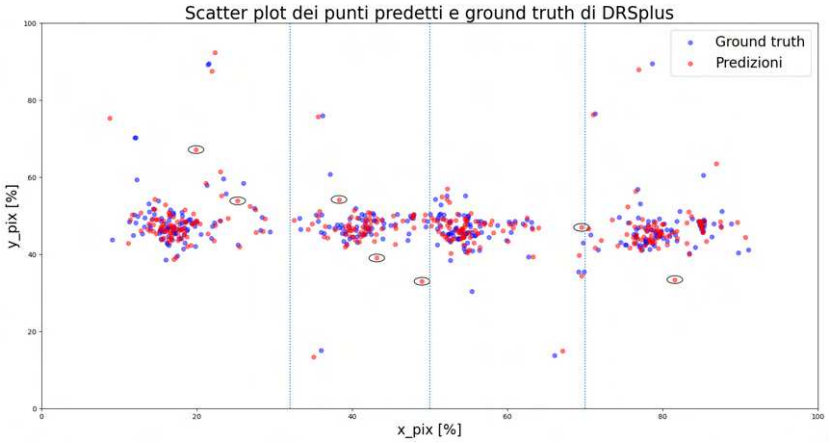


Figura 4.16: Scatter plot dei punti predetti e di ground truth

É interessante notare che negli scatter plot si distinguono dei *cluster* che, in generale, sono attribuibili a immagini dell'occhio destro e sinistro dei pazienti. In Eidon e DRSplus si distinguono un maggior numero di gruppi, attribuibili non univocamente alla diversità di *field* di acquisizione delle immagini. Per Compass invece compaiono solamente due

grandi gruppi, poiché le immagini di tale dispositivo sono acquisibili sono nel campo *central*. Per quando riguarda i punti cerchiati, possiamo vedere come vengano distribuiti in modo abbastanza omogeneo tra occhio destro e sinistro e tra i campi di acquisizione, ne segue dunque che il campo di acquisizione e la posizione dell'occhio non siano correlati in alcun modo con l'errata predizione della rete. Osserviamo inoltre come al di fuori dei grandi *cluster* siano presenti anche punti isolati e gruppi molto più piccoli, essi sono relativi a immagini in cui il disco ottico è posizionato nella zona periferica dell'immagine, nonostante ciò essi non contengono punti con predizione uscente.

I Bland-Altman plot seguenti sono stati realizzati con lo scopo di valutare quantitativamente gli errori che vengono fatti nella predizione del centro del disco, dividendo tale analisi a seconda del *device*. L'analisi del Bland-Altman plot viene comunemente utilizzato per valutare l'*agreement* tra due misure, calcolando le differenza media delle due misure (il bias). L'intervallo di confidenza è calcolato al 95%, quindi contiene il 95% delle differenza tra i due metodi di misurazione. I limiti dell'*agreement* sono collocati a $1.96SD$ dalla differenza media. L'ampiezza dell'*agreement* è indice di quanto siano concordati i due metodi di misurazione, minore è la sua ampiezza e mmigliore è la concordanza tra i due metodi.

I seguenti grafici sono stati fatti rapportando le coordinate x e y alla dimensione delle immagini, rispettivamente ampiezza e altezza di ogni *device*. Sull'asse verticale troviamo dunque le differenze tra la coordinata di *ground truth* e di predizione in termini percentuali, mentre sull'asse orizzontale le medie aritmetiche delle due misure in termini percentuali. In questo modo i Bland-Altman dei diversi dispositivi sono tra loro confrontabili, poiché abbiamo già visto che i *device* restituiscono immagini con proporzioni diverse e dimensioni diverse, dunque si rende necessario la procedura di normalizzazione altrimenti gli errori commessi dalla rete sulle immagini dei tre *device* non potrebbero essere tra loro confrontabili.

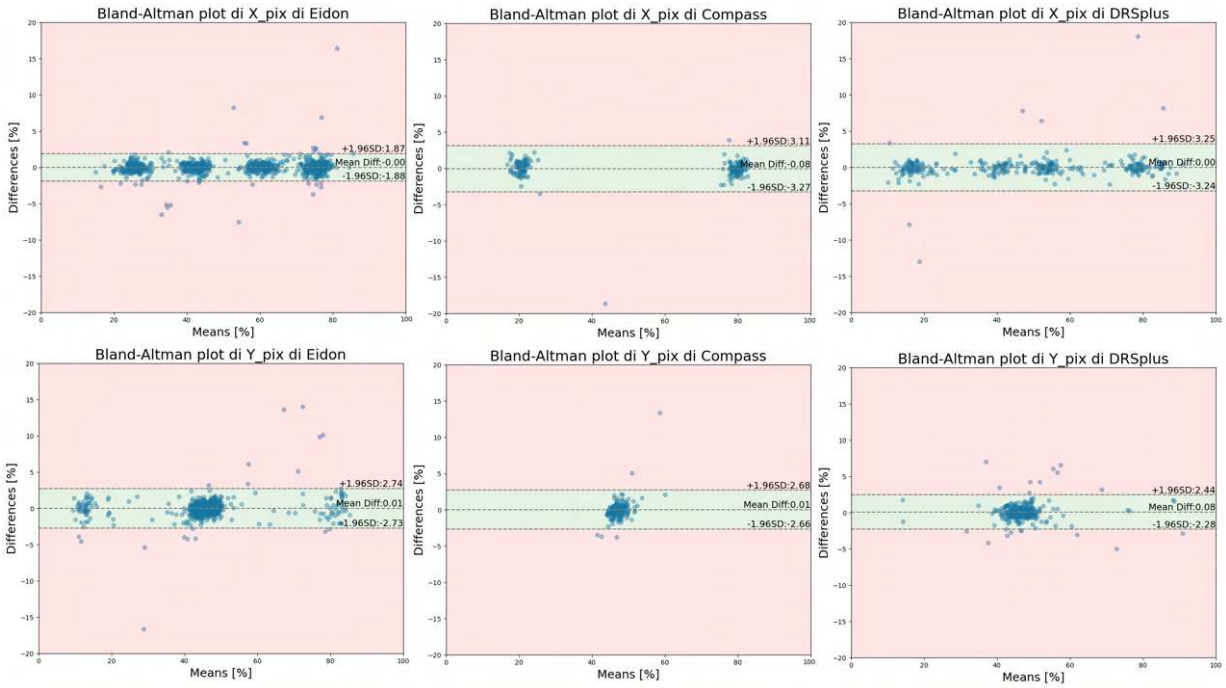


Figura 4.17: Bland-Altman plot del modello 2

Mettendo a confronto i risultati ottenuti possiamo notare come gli *agreement* risultino piuttosto omogeni per la coordinata y. Possiamo dire che l'errore di predizione di tale coordinata non è associabile ad immagini di un particolare *device*. Considerando invece la coordinata x possiamo notare come Compass e DRSplus abbiano l'*agreement* di ampiezza molto simile; per Eidon l'*agreement* della coordinata x risulta più piccolo rispetto agli altri appena citati, ne segue che con immagini di Eidon la rete commette un errore di predizione inferiore per la coordinata x.

4.4.3 Modello 3

Un'altro modello che abbiamo voluto sviluppare in questo progetto di tesi è il modello originariamente proposto dall'articolo [2] a cui abbiamo fatto riferimento per la realizzazione dei modelli 1 e 2. Ci siamo chiesti se l'informazione sul raggio del disco ottico potesse in qualche modo migliorare la predizione del centro. Per rispondere a questa domanda abbiamo pensato di realizzare il modello 3 utilizzando il modello 2 e fornendo alla rete in ingresso l'informazione del raggio del disco in aggiunta alle coordinate del centro, che finora abbiamo utilizzato solo in fase di valutazione dei risultati. Il modello proposto riceverà come *label di ground truth* delle immagini un array numpy di vettori così composti: $[x\_pix, y\_pix, radius]$, i primi due elementi localizzano il pixel del centro del disco e l'ultima componente definisce il raggio che contiene il disco ottico.

Riportiamo di seguito i risultati ottenuti per diversi valori di *batch size*, come abbiamo fatto anche per gli altri modelli

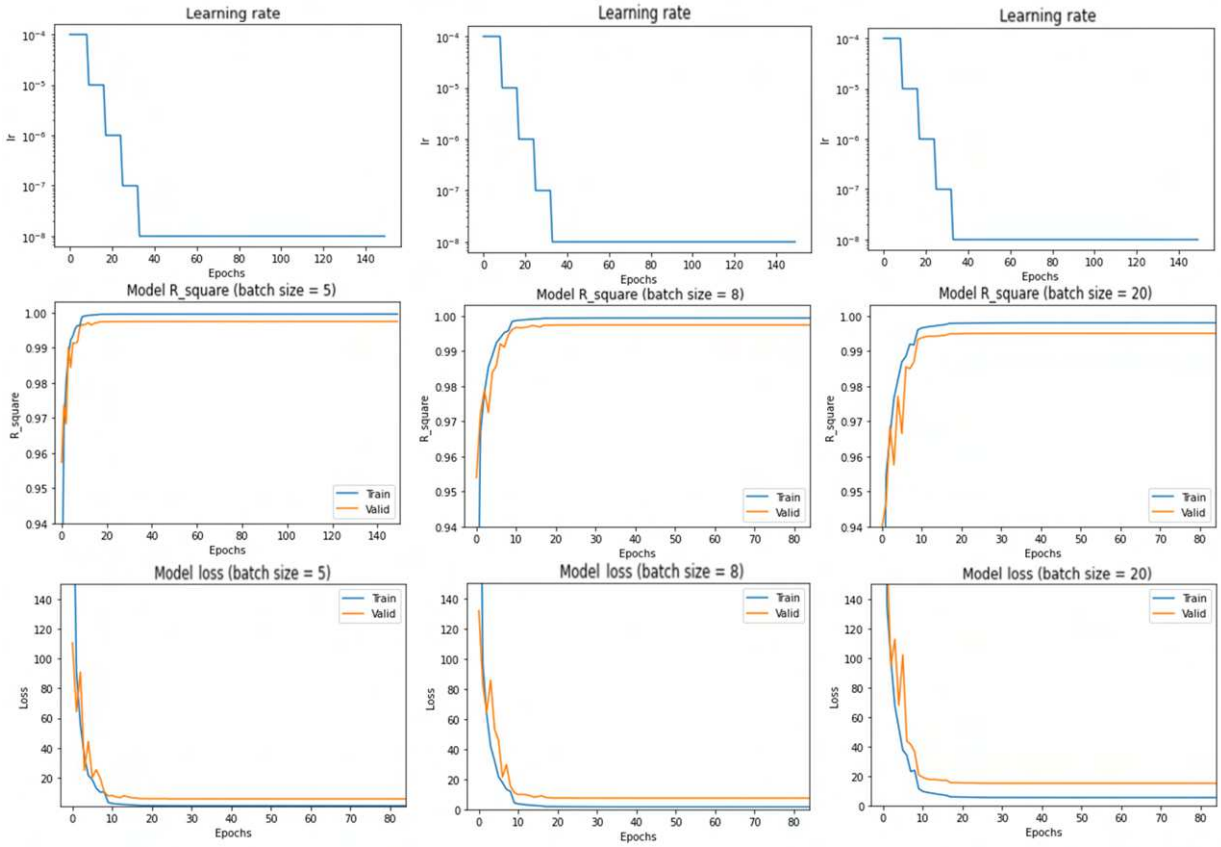


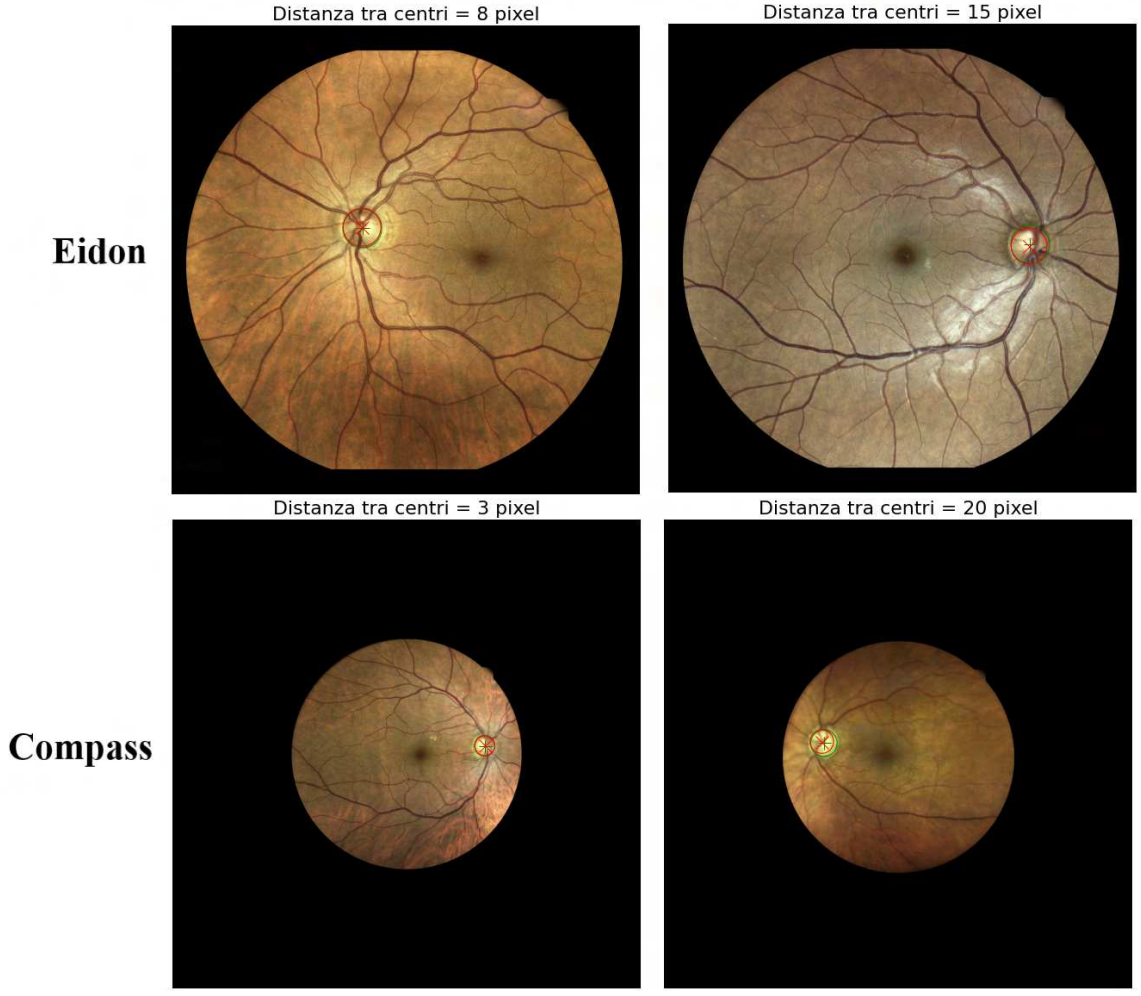
Figura 4.18: Risultati del modello 3

I piccoli picchi rumorosi che comparivano nelle curve del modello 2 sembrerebbero mitigati, le curve del modello 3 risultano quindi più omogenee, mentre i valori di plateau delle curve sono molto simili ai valori ottenuti con il modello 2. Una differenza la troviamo nei valori di predizione sul test set

	Batch size = 5	Batch size = 8	Batch size = 20
Test loss	5.418	8.569	13.333
Test R^2	0.998	0.997	0.995
Immagini predizione uscente	32 (1,956%)	56 (3,423%)	106 (6,479%)

Tabella 4.8: Tabella dei risultati di predizione del modello 3

Confrontando i valori ottenuti in Tabella 4.8 con quelli della Tabella 4.4 è facile notare che per tutte le *batch size*, il modello 3 ha restituito risultati di predizione migliore rispetto al modello precedente. Il modello 3 risulta quindi essere quello che ha restituito i risultati migliori, inoltre si conferma essere la *batch size* pari a 5 la migliore tra quelle testate . In generale questo modello sembra essere un buon predittore della posizione e dimensione del disco ottico. Le immagini che seguono sono alcuni esempi della predizione che restituisce la rete per ogni *device*; nelle immagini il cerchio di *ground truth* è definito da un cerchio verde, il cui centro è evidenziato dal simbolo + verde, il cerchio predetto dalla rete con il relativo centro sono di colore rosso.



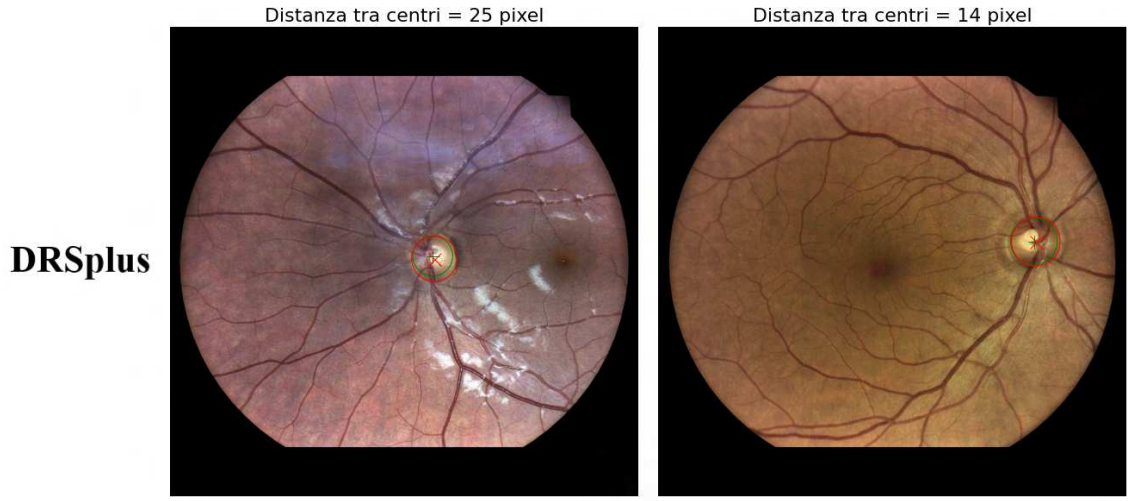


Figura 4.19: Risultati del modello 3

Osservando attentamente la immagini possiamo vedere come il modello riesca a predire molto bene il cerchio che definisce il disco e il relativo centro, soprattutto nelle immagini in cui le componenti anatomiche risultano ben distinte. Anche questo modello sembrerebbe sia in grado di utilizzare l'informazione dei vasi sanguigni principali per la localizzazione del disco. L'informazione del raggio sembra portare un miglioramento nella predizione delle coordinate del disco, infatti se facciamo un confronto con la Figura 4.20, che mostra le stesse immagini, possiamo vedere come ci sia un miglioramento della predizione del centro in alcune immagini. Questi risultati vengono rispecchiati anche dalla quantità di immagini con predizione uscente dal cerchio di *labeling*, che risulta minore rispetto al modello precedente.

Analizziamo nel dettaglio la composizione delle immagini in cui la predizione esce dal cerchio di etichetta

IMMAGINI PREDIZIONE USCENTE		
Eidon	Compass	DRsplus
24	3	5
2,14 %	1,57 %	1,54 %
TOTALE IMMAGINI		
32		

Tabella 4.9: Tabella delle immagini con predizione uscente

L'informazione del raggio ha permesso alla rete di diminuire significativamente gli errori che la rete commette nella predizione: il numero di immagini di Compass e DRSplus con predizione uscente si dimezza, la stessa cosa non si può dire per Eidon il cui numero di immagini con predizione uscente è leggermente maggiore.

Esaminando nel dettaglio le immagini di Eidon in cui il centro predetto esce dal cerchio di etichetta ci accorgiamo che sono per la maggior parte le immagini con predizione uscente del modello 2, vediamone alcuni esempi.

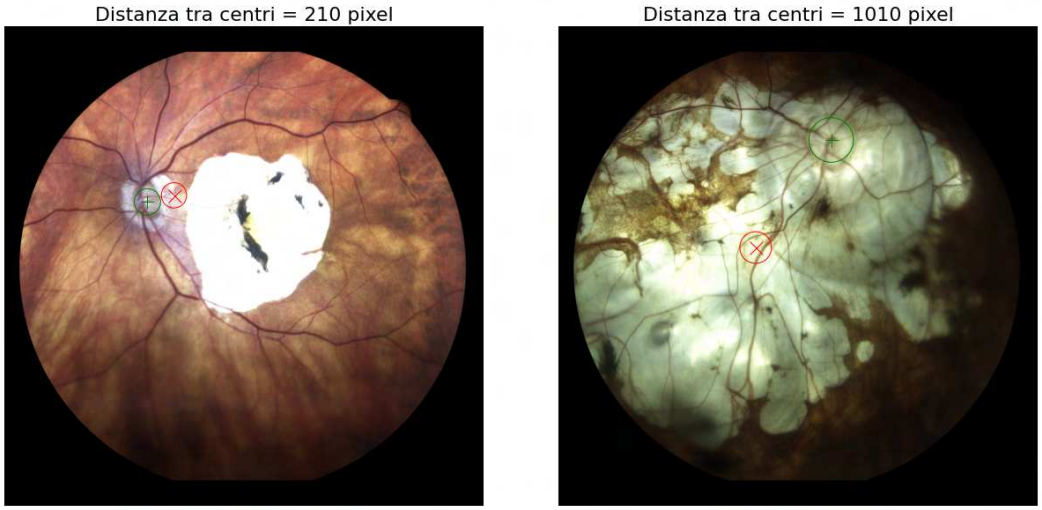


Figura 4.20: Risultati Eidon del modello 3

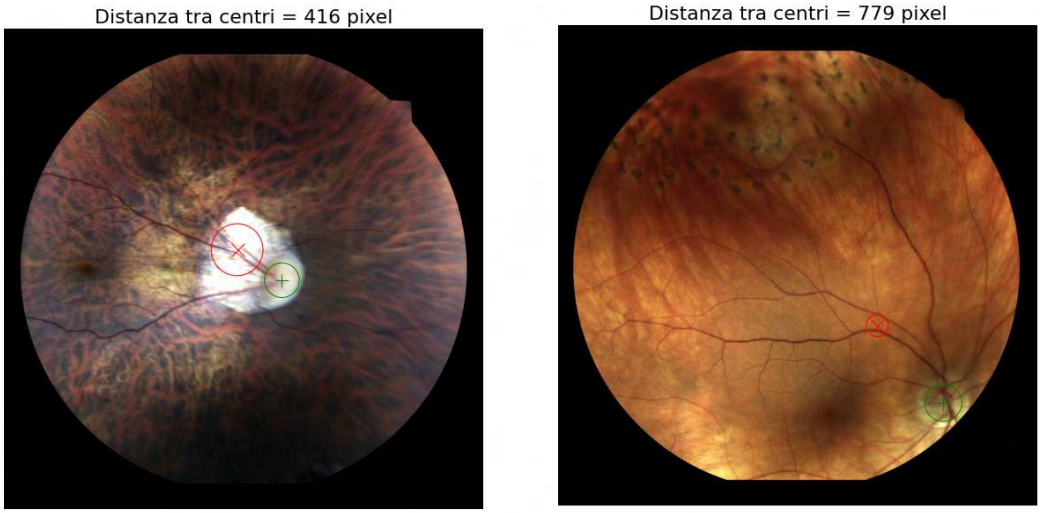


Figura 4.21: Risultati Eidon del modello 3

Anche in questo modello la presenza di aree atrofiche ha condizionato fortemente la predizione della rete. L'informazione aggiuntiva sul raggio del disco ottico non ha aiutato nella predizione, in alcuni casi peggiorandola rispetto al modello precedente.

Le immagini di Compass sono quelle che hanno subito il maggior miglioramento a livello predittivo della rete, in cui, come abbiamo visto dai risultati di Tabella 4.9, le immagini con predizione uscente sono dimezzate. Le 3 immagini con predizione uscente di Compass sono costituite da: 2 immagini che appartenevano anche alle immagini con centro uscente del modello precedente ed uno che differisce tra i due modelli.

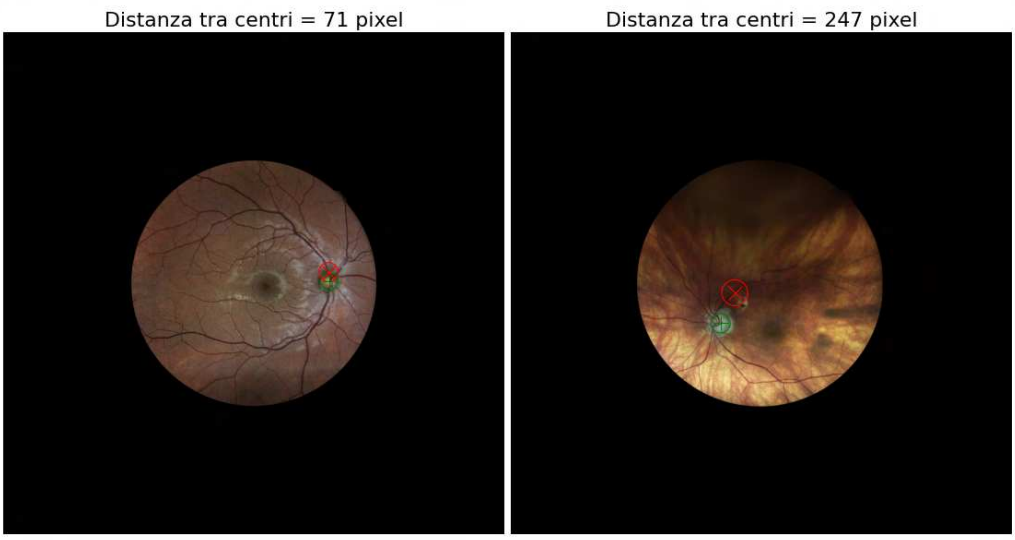


Figura 4.22: Risultati Compass del modello 3

Nelle immagini di DRSplus osserviamo lo stesso fenomeno, l'informazione sul raggio ha permesso alla rete di migliorare la predizione di alcune immagini rispetto al modello precedente.

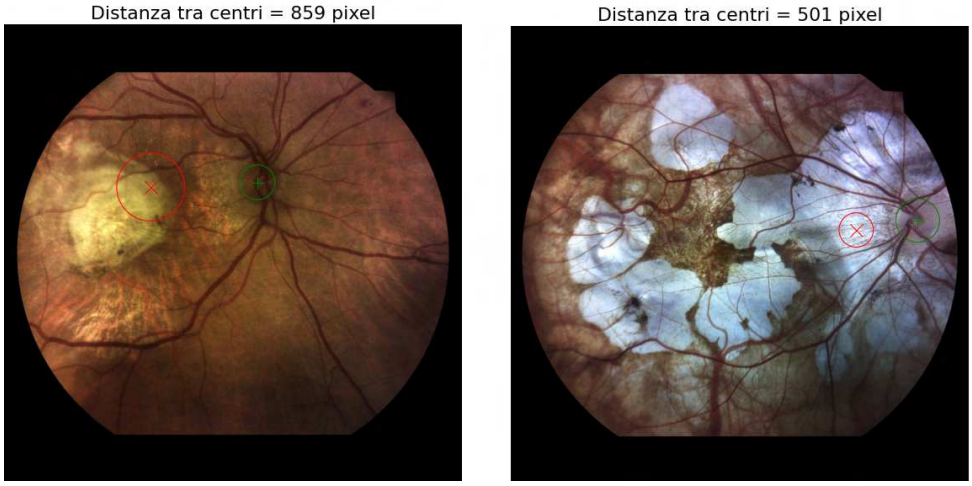


Figura 4.23: Risultati DRsplus del modello 3

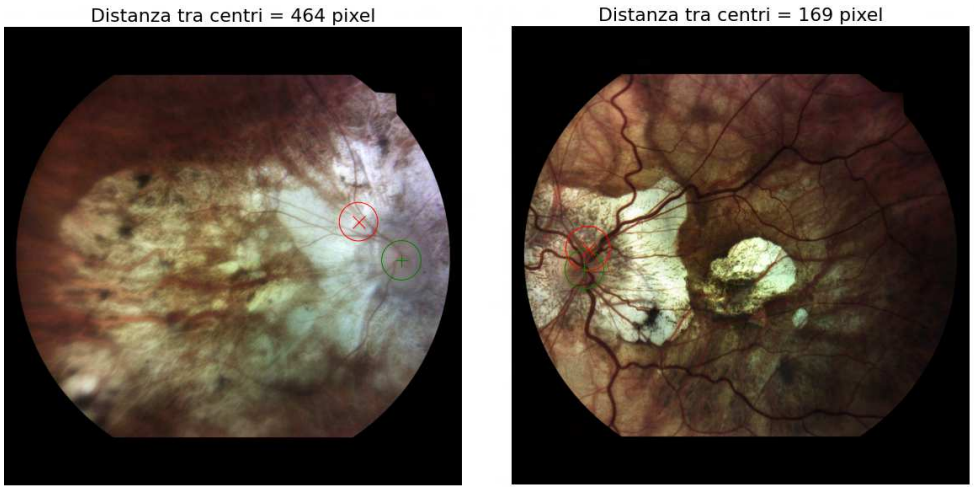


Figura 4.24: Risultati DRsplus del modello 3

Poiché le immagini di DRSplus con predizione uscente sono costituite principalmente da immagini con aree atrofiche, ciò suggerisce che questa caratteristica ha condizionato la predizione della rete, che risulta quindi essere errata, nonostante alcune immagini hanno mostrato un miglioramento di predizione delle coordinate del centro.

A conclusione dell'analisi visiva dei risultati, potremmo dire che molto probabilmente gli errori commessi dal modelle 3 sono correlati alla presenza di aree atrofiche nelle immagini. Analogamente al modello 2, questi risultati sono da attribuire alla scarsa presenza di immagini con aree atrofiche nel training set (Tabella 4.7), esse infatti costituiscono solamente il 6% delle immagini del training set. Dunque la rete non ha sufficienti esempi per dare delle predizioni più accurate anche quando è presente una area atrofica. Inoltre i risultati con errore maggiore sono attribuibili alle immagini in cui l'area atrofica si estende per più dell'80 % dell'immagine; delle immagini atrofiche del training set solamente il 10% è costituito da immagini con area atrofica che si estende per più dell'80% dell'immagine.

Al fine di valutare l'errore di prezione del centro, si mettono a confronto i valori *ground truth* e di predizione delle singole coordinate, che sono state opportunatamente normalizzate. Anche in questo caso è necessaria una procedura di normalizzazione tra i valori di x ed y, poiché le immagini originali hanno dimensioni diverse dipendentemente dal *device* di appartenenza. Le coordinate x sono state quindi normalizzate rispetto all'ampiezza dell'immagine originale di appartenenza, mentre la coordinata y è stata normalizzata sull'altezza dell'immagine originale. Visualizzando i grafici delle singole coordinate di *ground truth* e predette a confronto (Figura 4.26) possiamo affermare che gli errori sulle coordinate x ed y del modello 3 sono minori rispetto al modello 2. Si riconferma essere la coordinata y la principale causa di errore sulla localizzazione del disco, poiché presenta un numero maggiore di punti che si discostano dalla diagonale, inoltre sono posizionati ad una maggior distanza da essa rispetto alla coordinata x.

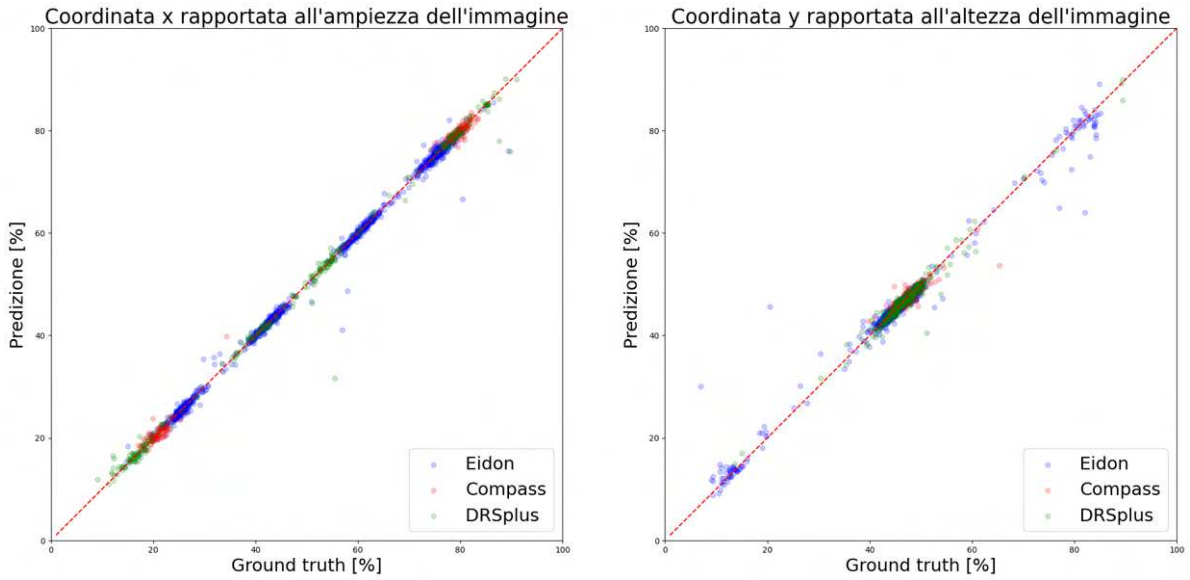


Figura 4.25: Plot delle coordinate a confronto

In generale, sembrerebbe che il raggio riesca a migliorare in piccola parte la predizione del centro seppur presentando con una forte distanza dei punti dalla diagonale, indice che i valori del raggio predetti dalla rete si discostano molto dai valori di *ground truth* (Figura 4.26).

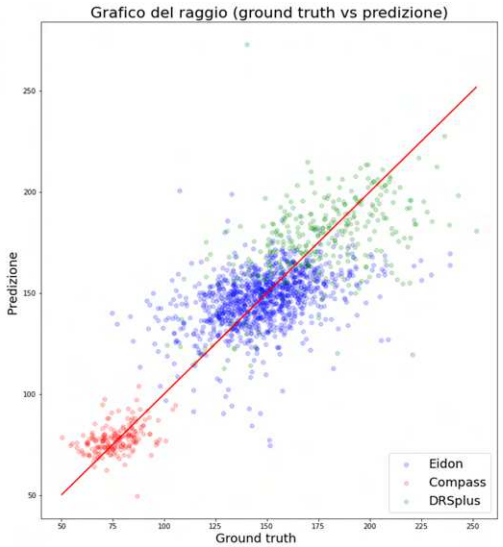


Figura 4.26: Plot del raggio a confronto

Per congruenza con il modello precedente riportiamo di seguito gli scatter plot (Figura 4.27) che abbiamo ottenuto plottando, per ogni *device*, i punti predetti e quelli veri, normalizzando le singole coordinate x e y ai valori di ampiezza e altezza rispettivamente come abbiamo fatto per i grafici in Figura 4.23. Come in precedenza abbiamo evidenziato le predizioni uscenti cerchiandole di nero.

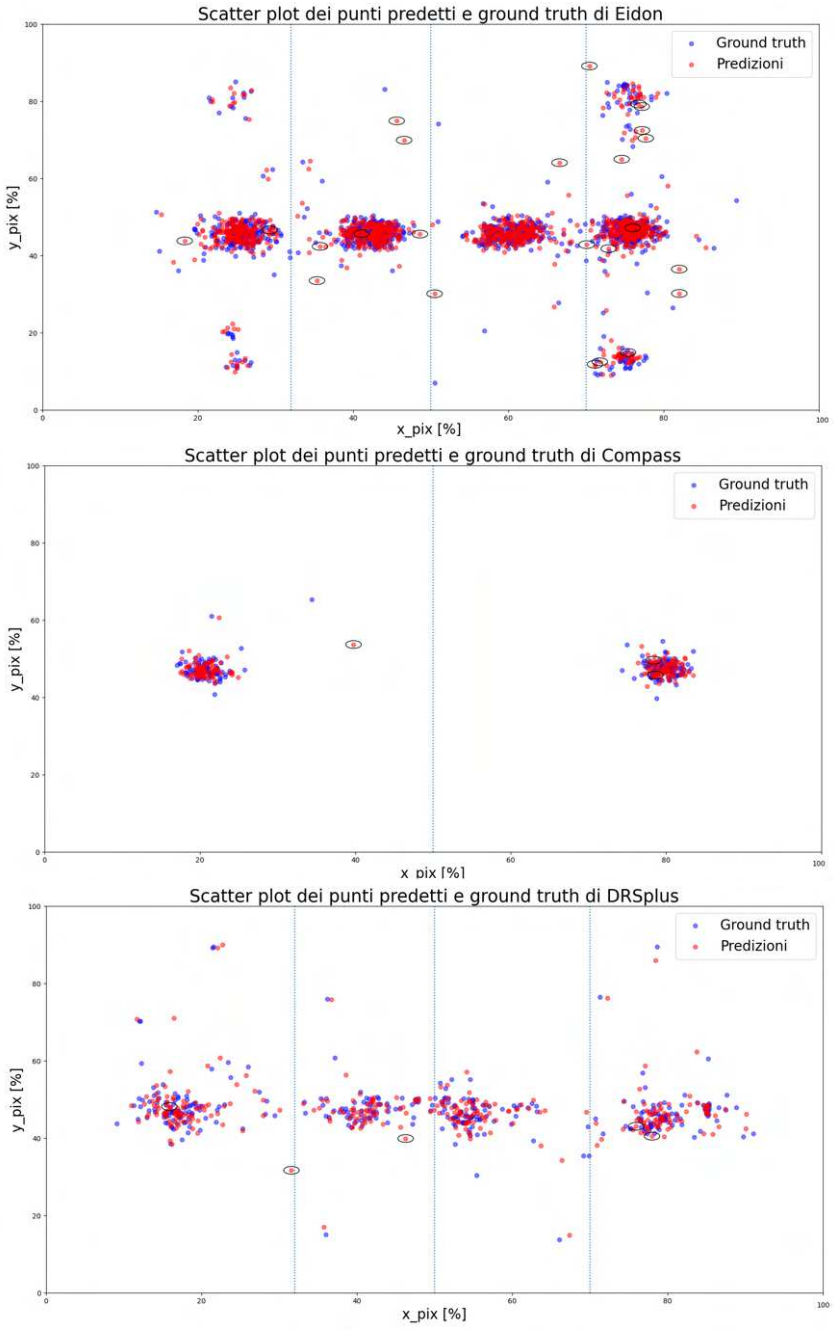


Figura 4.27: Scatter plot dei punti predetti e di ground truth

Come per i punti di *ground truth* anche le predizioni si distribuiscono in *cluster* più o meno grandi, identificativi di appartenenza a retine di occhio destro e sinistro. Qualitativamente parlando possiamo dire c'è un buon *agreement* tra la *ground truth* e la predizione dei centri dei dischi. Le predizioni rispecchiano in modo molto buona la distribuzione dei punti dei centri veri. I punti cerchiati, che definisco le immagini con centro predetto uscente dal cerchio di *ground truth*, sembrano non avere una relazione con la tipologia di occhio e del campo di acquisizione dell'immagine, infatti si distribuiscono in modo omogeneo in tutte le porzione del grafico a cui possiamo attribuire approssimativamente l'appartenenza a immagini di occhio destro e sinistro. In generale, sembrerebbe che le immagini in cui il disco ottico è collocato perifericamente non sia fonte di errore per la predizione della sua

localizzazione.

In ultima analisi, riportiamo i grafici Bland-Altman, che hanno l'obiettivo di valutare l'*agreement* tra i valori delle coordinate di predizione e di *ground truth*. Per farlo abbiamo dovuto normalizzare le singole coordinate con le dimensioni delle immagini a cui esse appartengono, questo per rendere i grafici Bland-Altman tra loro confrontabili. Sull'asse verticale troviamo dunque le differenze tra la coordinata di ground truth e di predizione in termini percentiali, mentre sull'asse orizzontale le medie aritmetiche delle due misure in termini percentuali. Nei grafici Bland-Altman minore è l'ampiezza dell'intervallo di confidenza (area verde in cui per costruzione si collocano il 95% delle differenze medie) e migliore è l'*agreement* tra le due variabili.

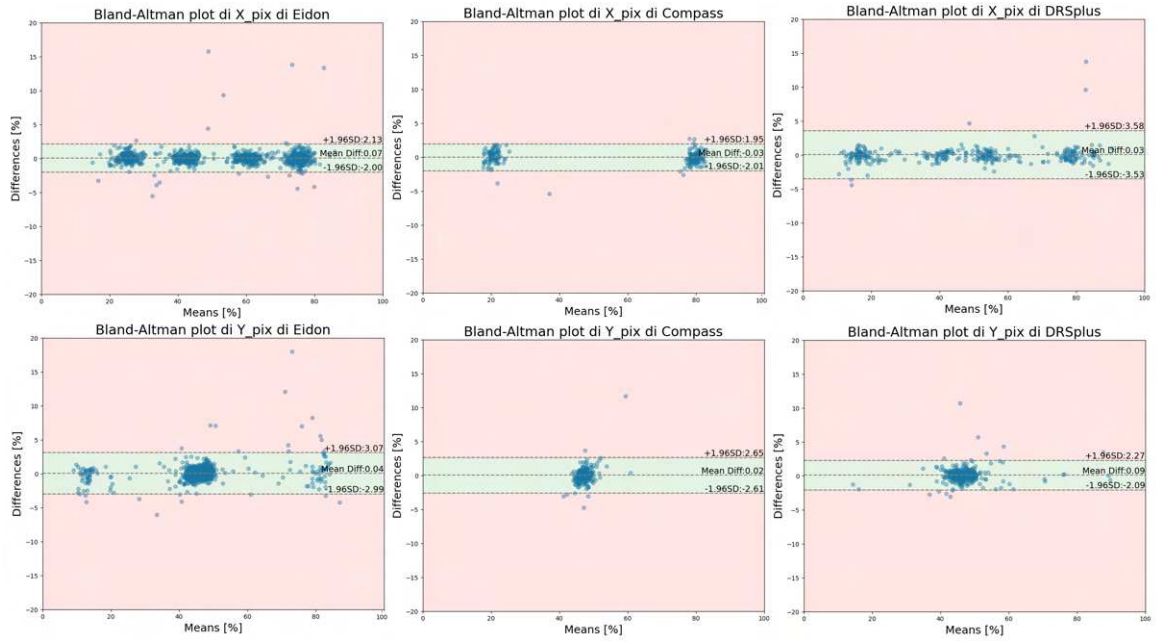


Figura 4.28: Bland-Altman plot del modello 3

I grafici mostrano una certa variabilità delle ampiezze dell'*agreement* della coordinata x,

Device	Intervallo di agreement per la coordinata x
Eidon	2.13+2.00=4.13%
Compass	1.95+2.01=3.96%
DRSplus	3.58+3.53=7.11%

Tabella 4.10: Tabella degli intervalli di agreement della coordinata x

in particolare l'errore di predizione di questa coordinata risulta maggiore per DRSplus, rispetto a Eidon e Compass che hanno ampiezza dell'*agreement* abbastanza simile. Per quanto riguarda la coordinata y è Eidon ad avere l'*agreement* più ampio

Device	Intervallo di agreement per la coordinata y
Eidon	3.07+2.99=6.06%
Compass	2.65+2.61=5.26%
DRSplus	2.27+2.09=4.36%

Tabella 4.11: Tabella degli intervalli di agreement della coordinata y

quindi le immagini Eidon sono quelle con maggior errore di predizione sulla coordinata y del centro.

I risultati appena presentati sono leggermente variati rispetto a quelli del modello 2, in cui i grafici Bland-Altaman (Figura 4.28) non evidenziavano una variabilità significativa tra gli *agreement* delle singole coordinate. In generale, possiamo dire che la variazione tra le ampiezze degli *agreement* della coordinata y non è significativa per differenziare le performance del modello in funzione dei *device*.

4.4.4 Conclusioni

La semplice architettura delle reti che abbiamo presentato ha permesso la realizzazione di modelli che predicono con buoni risultati il centro del disco ottico sulle immagini a nostra disposizione. L'assenza del termine di regolarizzazione migliora il tempo di esecuzione del modello. Tutti i modelli proposti danno buoni risultati con valori di *batch size* piccoli, mentre valori più elevati peggiorano notevolmente le curve di *training*. Per questo motivo abbiamo considerato il valore migliore testato di *batch size* pari a 5, con cui abbiamo ottenuto i migliori risultati. I modelli hanno restituito valori buoni anche in fase di predizione. Paragonando i modelli 2 e 3 con *batch size* 5, vediamo che i valori delle immagini in cui la predizione esce dal cerchio di etichetta (predizione considerata non accettabile) risultano di poco superiori al 2% per il modello 2 e inferiori al 2% per il modello 3, calcolate sul totale del test set. Ne segue che l'informazione sul raggio del disco ottico migliora le performace di predizione del centro del disco. Tale risultato trova un riscontro nel numero di immagini con predizione uscente, che diminuisce notevolmente per le immagini Compass e DRSplus nel modello 3.
L'analisi sugli errori di predizione ci porta a concludere che le immagini di Compass sono quelle in cui la rete del modello 2 trova maggiore difficoltà di localizzazione del centro, probabilmente dovuto alla forte presenza del bordo nero nelle immagini e alla ridotta presenza delle immagini di Compass nel training set(11,4%) rispetto a quella di Eidon (65,9%) e DRSplus (22,7%). Questi risultati migliorano significativamente con l'aggiunta dell'informazione del raggio (modello 3). Un minore ma ugualmente significativo miglioramento viene riscontrato anche per le immagini di DRSplus, passando dal modello 2 al modello 3. Non si può dire la stessa cosa delle immagini di Eidon che, nel passare al modello 2 al modello 3, subiscono un peggioramento delle performance.

Esaminando quantitativamente gli errori di predizione rapportati alle dimensioni delle immagini, ci porta a concludere che tali errori non siano attribuibili a immagini di uno specifico *device*. L'errore di predizione sulle singole coordinate sembrerebbe comparire in uguale quantità per ogni *device*, poiché l'ampiezza degli intervalli di *agreement* nei Bland-Altamn plot del modello 2 risultano omogenei tra i *device*. I Bland-Altman plot del modello 3 hanno invece evidenziato una leggera variabilità dell'*agreement* della coordinata x, la predizione sembrerebbe avere un errore maggiore (in termini percentuali) su immagini di DRSplus. La stessa cosa non accade per la coordinata y, i cui grafici di Bland-Altman non hanno mostrato una significativa variabilità tra le *agreement* dei diversi *device*, questi risultati sono in accordo con quelli del modello 2.

In generale, gli errori della rete possono essere attribuiti in parte alla presenza di aree atrofiche, che rendono difficile la localizzazione precisa del centro del disco. Tale risultati trova una relazione con la scarsa presenza di immagini con aree atrofiche nel training set.

4.5 Sviluppi futuri

I modelli presentati possono costituire una solida base per eventuali sviluppi futuri. In particolare ricordiamo che la fase di *labeling* dei dati è stata realizzata manualmente e senza revisione clinica, quindi potrebbe presentare degli errori. Ne segue che una supervisione clinica in fase di *labeling* dei dati potrebbe rendere più accurate le predizioni delle reti che abbiamo presentato. Sempre con supervisione clinica sarebbe interessante analizzare quanto la presenza di retine patologiche e la loro tipologia influenza la predizione della rete.

Le predizioni dei modelli sono probabilmente influenzate dalla presenza di aree atrofiche nelle immagini retiniche, che abbiamo visto essere presenti nel training set con una ridotto percentuale (6%). Un consistente aumento delle immagini con aree atrofiche nel training set, potrebbe migliorare le predizioni della rete.

Nelle curve di training dei modelli presentati e visibile l'overfitting, che sembra peggiore con l'assunzione di valori di *batch size* crescenti. Ssarebbe interessante verificare se questo problema può essere risolto con l'inserimento di *layer* di *dropout*, *batch normalization* e regolarizzazioni.

In fine, per la CNN2 potrebbe essere interessante utilizzare reti convoluzionali per la segmentazione come la Unet, come metodo alternativo per la localizzazione del centro del disco.

Bibliografia

[1] Calimeri C., Marzullo A., Stamile C. e Terracina G. (2016) *Optic Disc Detection using Fine Tuned Convolutional Neural Networks*, 12th International Conference on Signal-Image Technology & Internet-Based Systems.

[2] Gaofei Z., Ganghua L., Dongguang W. e Xiao Y. (2020), *A New Approach for the Regression of the Center Coordinates and Radius of the Solar Disk Using a Deep Convolutional Neural Network*, University of Chinese Academy of Sciences, Beijing, Republic of China.

[3] Huang W., Dunwei W., M. Ali Akber Dewan, Yan Y. e Wang K. (2018), *Automatic Detection of Optic Disc in Retina Image Using CNN and CRF*, IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovations

[4] I Gede Pande Darma Suardika, I Md. Dendi Maysanjaya e Made Windu Antara Kesiman(2022) *Optic Disc Segmentation Based on Mask R-CNN in Retinal Fundus Images*, Faculty of Engineering and Vocational, Universitas Pendidikan Ganesha, Bali, 4th International Conference on Biomedical Engineering (IBIOMED), Yogyakarta, Indonesia.

[5] Lim G., Cheng Y., Hsu W., Li Lee M. (2015) *Integrated Optic Disc and Cup Segmentation with Deep Learnings*, School of Computing National University of Singapore, IEEE 27th International Conference on Tools with Artificial Intelligence.

[6] Liu R., Lehman J., Molino P., Petroski Such F., Frank E., Sergeev A. e Yosinski J. (2018) *An intriguing failing of convolutional neural networks and the CoordConv solution*, 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

[7] Wejdan L. Alyoubi , Wafaa M. Shalash, Maysoon F. Abulkhair (2020), *Diabetic retinopathy detection through deep learning techniques: A review*, Information Technology Department, University of King Abdul Aziz, Jeddah, Saudi Arabia.

[8] Biochem Med online (2015), *Understanding Bland Altman analysis*, `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4470095/` [data ultima consultazione Marzo 2023]

[9] Clinica baviera online (2022), *Salute degli occhi*, `https://www.clinicabaviera.it/blog/salute-degli-occhi/`

angiografia-oculare-con-o-senza-contrasto-a-cosa-serve/ [data ultima consultazione Settembre 2022]

[10] Datagen online, *ResNet-50: The Basics and a Quick Tutorial*, https://datagen.tech/guides/computer-vision/resnet-50/ [data ultima consultazione Dicembre 2022]

[11] Depends on the definition online (2018), *U-Net for segmenting seismic images with keras*, https://www.depends-on-the-definition.com/unet-keras-segmenting-images/ [data ultima consultazione Marzo 2023]

[12] Fondazione Raimondi Francesco online, *Angio OCT Oculare*, https://fondazioneraimondi.it/poliambulatori/medicina-diagnostica/angio-oct-oculare/ [data ultima consultazione Settembre 2022]

[13] Humanitas reasearch hospital online, *Tomografia ottica computerizzata (OCT)*, https://www.humanitas.it/visite-ed-esami/tomografia-ottica-computerizzata-oct// [data ultima consultazione Ottobre 2022]

[14] Icare Centervue spa online, https://www.icare-world.com/ [data ultima consultazione Febbraio 2023]

[15] IRCCS Fondazione G.B. Bietti online (2019), *Patologie del nervo ottico e patologie delle vie ottiche*, https://www.fondazionebietti.it/patologie/patologie-del-nervo-ottico-e-patologie-delle-vie-ottiche/ [data ultima consultazione 09/03/2020]

[16] Intelligenza artificiale online, *Machine learning*, https://www.intelligenzaartificiale.it/machine-learning/ [data ultima consultazione Novembre 2022]

[17] Machine learning mastery online, *How to use Learning Curves to Diagnose Machine Learning Model Performance*, https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/ [data ultima consultazione Ottobre 2022]

[18] Mathsly reasearch online (2021), *Plot di Bland-Altman: come misurare l'agreement*, https://www.mathsly.it/wordpress/plot-di-bland-altman-come-misurare-lagreement/#:~:text=Il\%20metodo\%20pi\%C3\%B9\%20noto\%20per,due\%20misurazioni\%20sono\%20in\%20accordo. [data ultima consultazione Marzo 2023]

[19] NetAi online (2021), *Guida rapida alle funzioni di attivazione nel Deep Learning*, https://netai.it/guida-rapida-alle-funzioni-di-attivazione-nel-deep-learning/#page-content [data ultima consultazione Novembre 2022]

[20] Network Digital 360 online, *Addestramento reti neurali*, https://www.ai4business.it/intelligenza-artificiale/transfer-learning-cose-come-funziona-e-applicazioni/\#:~:text=Il\%20transfer\%20learning\%20\%C3\%A8\%20un,di\%20una\%20seconda\%20differente\%20attivit\%C3\%A0 [data ultima consultazione Settembre 2022]

[21] xford academic, BJA: British Journal of Anaesthesia online (2007), *I. Using the Bland–Altman method to measure agreement with repeated measures*,https://academic.oup.com/bja/article/99/3/309/355972 [data ultima consultazione Marzo 2023]

Ringraziamenti

Alla conclusione di questo importante percorso mi sembra sia importante ringraziare le persone, che in modi diversi, sono state presenti durante questi anni di studio.

Prima di tutto desidero ringraziare Chiara e Silvia e l'azienda Centervue di Padova, che mi hanno accolto proponendomi questo progetto di tesi; da voi ho imparato molto, grazie per avermi dato fiducia e la possibilità di realizzare questo progetto insieme.

Un grazie va anche al relatore di questa tesi, il professor Fabio Scarpa che è stato un attento supervisore, grazie per i suoi consigli e per il suo lavoro.

I ringraziamenti più sentiti vanno ai miei genitori, senza di loro non sarei arrivata fin qui, grazie per avermi dato la possibilità di scegliere il mio futuro intraprendendo questo percorso, siete un modello di vita per me, spero di avervi reso almeno un pò orgogliosi.

Un grazie va anche ai miei fratelli, in particolare a Giulia, sempre al mio fianco in ogni situazione, grazie per i tuoi consigli, perché sai sempre cosa dire nel momento giusto e per avermi dato la forza di affrontare ogni difficoltà.

Grazie anche a Riccardo, che da diversi anni mi accompagna nella vita e in questo percorso, grazie per aver sopportato tutti i miei malumori e i miei colpi di testa, grazie anche per farmi sentire capace di poter fare ogni cosa.

Desidero ringraziare anche le mie amiche Fabiola, Giulia, Giulia e Alice perché sempre presenti nonostante la vita ci porti a intraprendere cammini diversi.

In fine, un grazie a tutti coloro che in qualche modo mi sono stati vicini in questi anni, che hanno saputo festeggiare con me le mie vittorie e che mi hanno dato sostegno durante le mie cadute. Devo riconoscere che non è stato un percorso semplice, che mi ha portato molte volte a dubitare di me e delle mie capacità, ma credo anche che il bello sia questo, lottare per ciò che ci sta a cuore e per ciò che ci appassiona. «Non tutti possiamo fare grandi cose, ma possiamo fare piccole cose con grande amore», con queste parole spero, non in un futuro sempre facile, ma uno per cui valga la pena lottare con accanto le persone che mi vogliono bene e anche con un pò di fede.