

POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Biomedica



**Politecnico
di Torino**

Tesi di Laurea Magistrale

**Sviluppo di un algoritmo basato su reti
neurali convoluzionali per la predizione di
positività al Covid-19 e relativa severità
in scansioni TC**

Relatore

Ing. MASSIMO SALVI

Correlatore

Ing. FILIPPO MOLINARI

Candidato

SIMONE AZZONE

Marzo 2023

*Al nonno Carmelo, al nonno Giovanni, alla nonna Tina,
al piccolo Filippo ed al nonno Peppe,
grazie della vostra protezione. Siete i miei angeli custodi.
Vi avrei voluti tutti qui.
Vi amo*

Sommario

Dalla sua rilevazione iniziale nel 2019 a Wuhan, in Cina, il Covid-19 si è diffuso rapidamente in tutto il mondo, modificando l'ordine socio-economico e demografico mondiale. I metodi tradizionalmente utilizzati per la diagnosi come il test RT-PCR o il tampone antigenico rapido, richiedono abbastanza tempo per essere processati e forniscono talvolta risultati con bassa precisione. Per tale motivo, l'analisi di immagini radiografiche, ed in particolare di immagini di tomografia computerizzata (TC), forniscono un quadro dettagliato delle condizioni polmonari ed assistono i radiologi nella diagnosi di Covid-19. Tuttavia, le scansioni TC di una persona contengono centinaia di slices e, in tal senso, le tecniche di intelligenza artificiale di classificazione dei volumi polmonari potrebbero assistere i radiologi per una più rapida e precisa rilevazione dell'infezione.

Questo progetto di tesi utilizza algoritmi di deep learning basati su reti neurali convoluzionali (CNN), addestrate per identificare i pazienti positivi all'infezione e con forma grave di Covid-19, definito come intubazione o morte entro un mese dall'acquisizione della TC. Il dataset sul quale vengono addestrate le reti è costituito da un campione pubblicamente accessibile di scansioni TC dello STOIC Dataset, il più grande dataset di immagini di tomografia computerizzata di pazienti con COVID-19 e pazienti sani. Il dataset prevede la distinzione in tre classi: pazienti Covid-19 positivi in forma lieve, pazienti Covid-19 positivi in forma grave e pazienti sani.

Sono state addestrate, mediante il metodo del transfer learning, diverse reti di classificazione, fornendo loro in input le immagini opportunamente pre-processate. In particolare, dal dataset iniziale, denominato Dataset 1, è stato ottenuto un secondo dataset, denominato Dataset 2, costituito dalle slices con il polmone segmentato e su ciascuno di questi sono stati condotti gli allenamenti delle reti. Il classificatore addestrato definisce, per ciascuna slice delle scansioni, delle previsioni che vengono quindi combinate, mediante tecniche di post-processing, per classificare i pazienti in una delle tre classi sopracitate.

Per la valutazione delle performance, così come utilizzato dalla Challenge, si utilizza come metrica primaria l'AUC severity che quantifica la capacità discriminativa della forma grave dell'infezione; come metrica secondaria si utilizza

l'AUC positivity per definire la capacità della rete di discriminare i pazienti positivi all'infezione da quelli sani. È stata poi testata una soluzione che ha ottenuto risultati migliori rispetto ai singoli modelli, la quale prevede la media delle softmax, ottenute dalla fase di inference, di alcuni dei modelli di classificazione addestrati.

Dunque, sono state utilizzate le softmax delle reti HRNet e ResNet101, addestrate e testate sul Dataset 1, e le softmax delle reti HRNet e W-ResNet50, addestrate e testate sul Dataset 2.

La combinazione di più reti ha permesso di raggiungere valori di AUC severity pari al 60% sui pazienti di Test ed a circa il 75% sui pazienti di Training, conseguendo performance superiori rispetto alle singole reti. Anche in termini di AUC positivity, si raggiungono risultati superiori pari a circa il 76% e 84% rispettivamente sui pazienti di Test e sui pazienti di Training.

Summary

Since its initial survey in 2019 in Wuhan, China, Covid-19 has spread rapidly around the world, changing the world's socio-economic and demographic order. Traditional diagnostic methods include RT-PCR testing or the antigenic rapid swab but require time to process and sometimes provide results with low precision. For this reason, the analysis of radiographic images, especially computed tomography, provide a detailed picture of lung conditions and assist radiologists in diagnosing Covid-19. However, a CT scan contain hundreds of slices and, which is why artificial intelligence (IA) techniques for classifying lung volumes could assist radiologists for faster and more accurate detection of infection.

This thesis project uses deep learning algorithms based on convolutional neural networks (CNN), trained to identify patients infected with Covid-19 and to predict the severe outcome of Covid-19, defined as intubation or death within one month from the acquisition of the CT scan. The CNN are trained on a dataset consisting of a publicly available sample of the STOIC Dataset, the largest dataset of computed tomography images of COVID-19 patients and healthy patients. The dataset is divided into three classes: mild positive Covid-19 patients, severe positive Covid-19 patients and healthy patients.

Different classification networks have been trained, using the transfer learning method, by providing them with pre-processed images as input. In particular, from the initial dataset, called Dataset 1, was obtained a second dataset, called Dataset 2, consisting of slices with the segmented lung and on each of them the training of the networks was conducted. The trained classifier defines, for each slice of the scans, predictions which are then combined, using post-processing techniques, to classify patients into one of the three aforementioned classes. For performance evaluation, as used by the Challenge, AUC severity is used as the primary metric, which quantifies the discriminatory ability of patients' severity; AUC for COVID19 RT-PCR test positivity is used as a secondary metric, to define the network's ability to classify infection-positive patients from healthy ones.

A solution was then tested that obtained better results than the single models, which averages the softmax, obtained in the inference phase, of some of the classification models trained. Therefore, the softmax of the HRNet and ResNet101

networks, trained and tested on Dataset 1, and the softmax of the HRNet and W-ResNet50 networks, trained and tested on Dataset 2. The combination of multiple networks has achieved AUC severity values equal to 60% on Test patients and approximately 75% on Training patients, achieving higher performances than single networks. Also in terms of AUC positivity, higher results of approximately 76% and 84% were achieved on Test and Training patients, respectively.

Table of Contents

Elenco delle tabelle	IX
Elenco delle figure	X
1 Introduzione	1
1.1 Anatomia del sistema respiratorio	1
1.1.1 Componenti dell'apparato respiratorio	1
1.1.2 I polmoni	2
1.1.3 Ventilazione meccanica	4
1.2 Patologie dell'apparato respiratorio	4
1.3 Covid-19	5
1.3.1 L'origine della pandemia e le conseguenze a livello globale	5
1.3.2 Modalità di trasmissione	6
1.3.3 Manifestazioni cliniche	6
1.3.4 Manifestazioni cliniche a livello polmonare	7
2 Metodi di diagnostica	8
2.1 Test molecolare	8
2.2 Test antigenico rapido	9
2.3 Tecniche di imaging radiologiche	9
2.3.1 Radiografia del torace	9
2.3.2 Tomografia computerizzata del torace	10
2.4 Descrizione delle principali lesioni associate al Covid-19 osservabili con TC	13
3 Apprendimento automatico	15
3.1 Intelligenza Artificiale (IA)	15
3.2 Machine Learning e Deep Learning	16
3.3 Reti neurali	16
3.4 Il neurone artificiale ed il suo modello matematico	18
3.5 Rete Neurale Convolutionale (CNN)	20

3.5.1	Architettura delle reti neurali convoluzionali	21
3.5.2	Addestramento delle reti	24
3.5.3	Transfer Learning	26
3.5.4	Overfitting	26
4	Stato dell'arte	28
5	Materiali e metodi	39
5.1	STOIC Dataset	39
5.2	Data pre-processing	42
5.2.1	Split dataset	43
5.2.2	Data rescaling	43
5.2.3	Data selection	45
5.2.4	Data balancing	52
5.3	Training delle reti CNN	53
5.3.1	Architettura delle reti	54
5.3.2	Parametri ed iperparametri delle reti	57
5.3.3	Salvataggio del modello addestrato	63
5.4	Fase di testing e post-processing	64
5.4.1	Ensemble di più reti e dei loro output	66
5.5	Metodi di valutazione	67
5.5.1	Confusion Matrix	67
5.5.2	Metriche di validazione	68
5.5.3	ROC Curve e AUC	71
6	Risultati	73
6.1	Risultati sul Training set	74
6.1.1	Risultati ResNet50	74
6.1.2	Risultati ResNet101	75
6.1.3	Risultati W-ResNet50	76
6.1.4	Risultati HRNet	77
6.2	Risultati sul Test set	78
6.2.1	Risultati ResNet50	78
6.2.2	Risultati ResNet101	79
6.2.3	Risultati W-ResNet50	80
6.2.4	Risultati HRNet	81
6.3	Risultati ensemble di più reti sui due dataset	82
6.4	Confronto finale tra classificazioni	84
7	Conclusioni e sviluppi futuri	87
	Bibliografia	89

Elenco delle tabelle

5.1	Parametri ed iperparametri di training delle reti	57
6.1	Tabella metriche di valutazione ResNet50 sui Training set dei due dataset	75
6.2	Tabella metriche di valutazione ResNet101 sui Training set dei due dataset	76
6.3	Tabella metriche di valutazione W-ResNet50 sui Training set dei due dataset	77
6.4	Tabella metriche di valutazione HRNet sui Training set dei due dataset	78
6.5	Tabella metriche di valutazione ResNet50 sui Test set dei due dataset	78
6.6	Tabella metriche di valutazione ResNet101 sui Test set dei due dataset	80
6.7	Tabella metriche di valutazione W-ResNet50 sui Test set dei due dataset	81
6.8	Tabella metriche di valutazione HRNet sui Test set dei due dataset	82
6.9	Tabella confronti singole reti sui due Dataset	84
6.10	Tabella confronti singole reti sui due Dataset, mediate sui tre subsets	85
6.11	Tabella confronti performance	86

Elenco delle figure

1.1	Rappresentazione polmoni	2
2.1	Dispositivo per tomografia computerizzata (TC)	10
2.2	Scala Hounsfield relativa a diversi tessuti	12
2.3	Tipiche TC di due pazienti con polmonite da COVID-19, che mostrano aree bilaterali di opacità a vetro smerigliato [12]	13
3.1	Rappresentazione neurone biologico e neurone artificiale [25]	18
3.2	Tipica architettura CNN di classificazione [30]	21
3.3	Operazioni di convoluzione [28]	22
3.4	Esempi di underfitting o overfitting [35]	27
4.1	Architettura della soluzione proposta da Aswathy [37]	29
4.2	Struttura del meccanismo di predizione basato sul decision fusion e relativo esempio proposto nello studio di Mishra et al. [38]	30
4.3	Andamenti del: a) learning rate ciclico b) learning rate ridotto sul plateau [39]	31
4.4	Architettura di deep learning proposta da Nguyen et al. [40]	32
4.5	Architettura che definisce la presenza di Covid-19 sulla base di scansioni TC proposta da Serte et al. [41]	33
4.6	Architetture proposte da Shah et al.: a) modello CTnet-10; b) modello VGG-19. Entrambe vengono utilizzate per la classificazione e per effettuare un confronto di performance tra architetture diverse [42]	34
4.7	Architettura generale del modello CNN proposto da Perumal et al. [43]	36
4.8	Operazioni di pre-processing delle immagini in Rajawat [44]	37
4.9	Architettura C-COVIDNet proposta da Rajawat et al. [44]	38
5.1	Slice di una scansione TC del Dataset STOIC	41
5.2	Flowchart pre-processing	42
5.3	Effetto del rescaling sulle immagini della scansione TC	45

5.4	Esempi di immagini appartenenti ai due dataset costruiti	46
5.5	Flowchart data selection	47
5.6	5.6a è il risultato dell'operazione di Otsu thresholding; 5.6b è il risultato dell'utilizzo del filtro di Sobel; 5.6c è il risultato del comando ' <i>clear_border</i> '	49
5.7	5.7a è l'immagine originale con la presenza dell'artefatto del supporto paziente; 5.7b è la maschera binaria con la presenza dell'artefatto; 5.7c è il risultato della rimozione dell'artefatto del supporto paziente dalla maschera binaria	51
5.8	Architettura delle reti ResNet [47]	55
5.9	Architettura delle reti Wide ResNet[47]	56
5.10	Architettura delle reti HRNet [50]	56
5.11	Effetto delle operazioni di CutMix e MixUp su immagini generiche [59]	63
5.12	Andamento loss function sul Training set e sul Validation set durante l'allenamento della rete [61]	64
5.13	Andamenti delle predizioni di un classificatore	65
5.14	Schema generale dell'algoritmo di fusione degli output dei classificatori	67
5.15	Confusion Matrix per classificazione binaria [64]	68
5.16	Confusion Matrix a 3 classi	70
5.17	Esempi di curve ROC [70]	72
6.1	Confusion matrix sul Training set con rete ResNet50 su ambo i dataset	74
6.2	Confusion matrix sul Training set con rete ResNet101 su ambo i dataset	75
6.3	Confusion matrix sul Training set con rete W-ResNet50 su ambo i dataset	76
6.4	Confusion matrix sul Training set con rete HRNet su ambo i dataset	77
6.5	Confusion matrix sul Test set con rete ResNet50 su ambo i dataset	79
6.6	Confusion matrix sul Test set con rete ResNet101 su ambo i dataset	79
6.7	Confusion matrix sul Test set con rete W-ResNet50 su ambo i dataset	80
6.8	Confusion matrix sul Test set con rete HRNet su ambo i dataset . .	81
6.9	Schema fusione delle decisioni mediante utilizzo delle softmax . . .	82
6.10	AUC severity ed AUC positivity per l'ensemble delle reti, calcolate per i cinque approcci di aggregazione delle predizioni.	83
6.11	Confusion Matrix ottenute dall'ensemble di più classificatori	85

Capitolo 1

Introduzione

Il primo capitolo di questo lavoro di tesi presenta le principali caratteristiche fisiologiche ed anatomiche dell'apparato respiratorio, con un focus specifico sul polmone, oggetto delle immagini di tomografia computerizzata. Segue poi una breve esposizione delle principali patologie del sistema respiratorio, introducendo così l'infezione da Covid-19. Vengono quindi riportate le modalità di trasmissione e soprattutto le manifestazioni cliniche a livello polmonare ed a livello dell'intero sistema respiratorio, al fine di mostrare un quadro completo di tale infezione.

1.1 Anatomia del sistema respiratorio

L'apparato respiratorio, o sistema respiratorio, è l'insieme di organi e tessuti che provvede alla respirazione, ossia all'introduzione dell'ossigeno atmosferico nell'organismo e, al contempo, all'espulsione dell'anidride carbonica prodotta dal metabolismo cellulare.

L'apparato è normalmente suddiviso in due tratti principali: le vie aeree superiori (o tratto respiratorio superiore) e le vie aeree inferiori (o tratto respiratorio inferiore). Il naso, la bocca, la faringe, la nasofaringe e la laringe appartengono al tratto respiratorio superiore; del tratto respiratorio inferiore, invece, fanno parte la trachea, i bronchi, i bronchioli, i polmoni ed i muscoli respiratori quali diaframma e muscoli intercostali [1].

1.1.1 Componenti dell'apparato respiratorio

In questa tesi, maggiore attenzione è rivolta alla descrizione del tratto respiratorio inferiore, rappresentato in Figura 1.1, e per tale motivo viene mostrato un prospetto delle sue caratteristiche a partire dalla trachea.

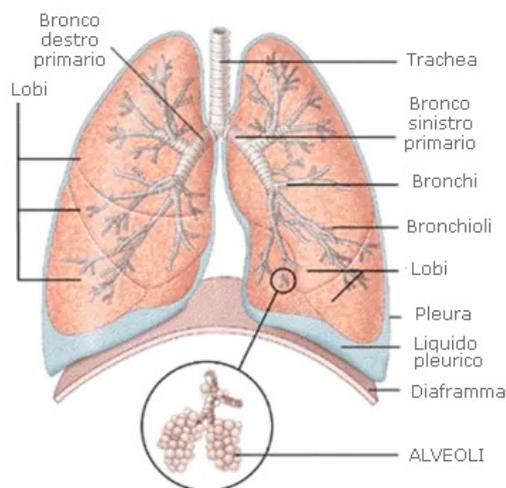


Figura 1.1: Rappresentazione polmoni

La trachea, costituita anelli cartilaginei, è un canale mediano estensibile e flessibile che ha inizio nel collo e termina nel torace, biforcandosi nei due bronchi primari che, a loro volta, si suddividono nei bronchi lobari (o bronchi secondari) che raggiungono i lobi polmonari [2][3]. Ciascun bronco primario penetra all'interno del rispettivo polmone, dando origine ad un'arborizzazione costituita da ulteriori e numerose ramificazioni, chiamate bronchioli [4]. In anatomia si definisce albero bronchiale il tratto delle vie aeree inferiori costituito dai bronchi e bronchioli che, a livello funzionale, sono responsabili del trasporto dell'aria dalla trachea ai polmoni. A loro volta, anche i bronchioli si suddividono fino a formare, nel tratto terminale, piccole vescicole chiamate alveoli (ciascun polmone contiene all'incirca 150-200 milioni di alveoli che nell'insieme raggiungono un'estensione totale di circa 75 m^2).

L'unità fondamentale per la ventilazione è l'alveolo, organizzato in strutture alveolari (sacchi alveolari) a forma di grappolo, che presenta pareti estremamente sottili per permettere gli scambi gassosi di ossigeno ed anidride carbonica con i capillari sanguigni che lo circondano. Dal momento che le pareti alveolari sono prive di tessuto muscolare, il polmone non può contrarsi ma è parimenti costretto ad adattarsi ai movimenti della cassa toracica che partecipa attivamente alla respirazione tramite variazioni di pressione intratoracica [3][5].

1.1.2 I polmoni

Anatomia polmonare

Le unità funzionali del sistema respiratorio sono i polmoni, i due organi deputati

alla ventilazione, ossia all'introduzione di ossigeno nell'organismo ed all'eliminazione dell'anidride carbonica dal sangue, la cui frequenza è adattata alle esigenze metaboliche dell'organismo da parte di stimoli chimici e nervosi. Viene fornita una loro rappresentazione in Figura 1.1.

Anatomicamente appartenenti alle vie aeree inferiori, i polmoni sono situati all'interno delle logge polmonari, le quali sono delimitate superiormente dalle strutture che si trovano nella parte alta del torace (come il plesso brachiale), inferiormente dal diaframma, esternamente dalla gabbia toracica e dai muscoli intercostali e internamente dal mediastino [6]. Il polmone è costituito da una base (o faccia diaframmatica), un apice, una faccia laterale (la più estesa) e una faccia mediale (o mediastinica). Profonde scissure percorrono la superficie polmonare, dividendo l'organo in lobi. Il polmone destro presenta due scissure - una principale ed una secondaria, rispettivamente in posizione obliqua ed orizzontale - che suddividono il polmone in tre lobi: superiore, medio e inferiore. Il polmone sinistro, meno voluminoso, presenta una sola scissione - equivalente alla scissura obliqua del polmone destro - che divide il polmone in due lobi: superiore e inferiore [7]. La minore grandezza del polmone sinistro, rispetto al polmone destro, è utile, a livello anatomico, a garantire uno spazio adeguato al cuore.

Due membrane sierose dette pleure polmonari -fondamentali nella ventilazione in quanto consentono la trasmissione del movimento respiratorio dal torace ai polmoni- rivestono la superficie interna della parete toracica e quella esterna dei polmoni; esse prendono i nomi di pleura parietale, più robusta e ricca di fibre, ed pleura viscerale, più sottile e trasparente [8]. In condizioni normali, queste scivolano l'una sull'altra nel corso della respirazione e vengono mantenute adese per mezzo di un sottile velo di fluido pleurico lubrificante, situato nella cavità pleurica.

I polmoni sono costituiti da un tessuto elastico spugnoso che risulta fondamentale per le variazioni di volume indotte dai movimenti respiratori, permettendo ai polmoni di espandersi durante l'inspirazione grazie ad una pressione sub-atmosferica (negativa), presente nella cavità pleurica e di tornare in condizioni normali grazie al ritorno elastico, durante l'espirazione [1].

Tale pressione è generata da due forze opposte che tendono a dilatare lo spazio pleurico: una verso l'interno esercitata dal polmone, espanso rispetto alla sua condizione di equilibrio, ed una verso l'esterno esercitata dalla gabbia toracica, compressa rispetto alla sua condizione di equilibrio.

La respirazione

Come accennato, i polmoni sono i due organi responsabili della respirazione, la quale si può suddividere in fase inspiratoria e fase espiratoria.

Durante l'inspirazione, l'aria attraversa l'intero sistema e perviene agli alveoli polmonari, luogo degli scambi gassosi sangue-alveoli, durante i quali i capillari delle

arterie polmonari rilasciano negli alveoli l'anidride carbonica ed i prodotti di scarto della circolazione periferica e si caricano dell'ossigeno presente nelle cavità alveolari dopo l'inspirazione.

La fase espiratoria inizia dopo gli scambi di gas e consente all'aria, ricca di anidride carbonica e povera di ossigeno, di eseguire in direzione opposta il percorso che l'ha portata nei polmoni ed uscire.

A questo punto, il sangue ossigenato presente nelle vene polmonari viene trasportato fino al cuore in modo da poter raggiungere e fornire ossigeno, per mezzo della circolazione sanguigna, agli organi ed ai tessuti di tutto l'organismo [1].

In una respirazione tranquilla, l'inspirazione è un processo attivo, avviato dalla stimolazione nervosa dei muscoli inspiratori quali il diaframma ed i muscoli intercostali. Al contrario, l'espirazione è un processo passivo in cui l'aria viene espulsa dalle vie aeree [3]. Tuttavia, l'espirazione può anche essere un processo attivo (espirazione forzata) se avviene con il contributo dei muscoli espiratori.

L'espirazione è seguita da una breve pausa (condizione di riposo) che precede il successivo atto inspiratorio. Durante questa pausa, in cui non vi è movimento d'aria da e per gli alveoli, tutti i muscoli respiratori sono rilasciati e le forze che agiscono sui polmoni e sulla parete toracica sono in equilibrio tra loro.

1.1.3 Ventilazione meccanica

La ventilazione meccanica, adottata quando il paziente è incapace a respirare adeguatamente ed in modo autonomo, aumenta la pressione delle vie aeree forzando l'aria nei polmoni per produrre l'inspirazione. Il ventilatore è invece inutile nella fase espiratoria che avviene passivamente poiché il sistema, nel momento in cui si introduce aria, viene caricato di energia meccanica/elastica, consentendo dunque un'espirazione spontanea (in questa fase il ventilatore è disattivato).

1.2 Patologie dell'apparato respiratorio

I medici patologi classificano le patologie che possono colpire l'apparato respiratorio in:

- Malattie ostruttive → Il sintomo principale è il restringimento delle vie aeree che si traduce in occlusione nei casi più gravi. Rientrano in questa categoria patologie quali: la bronchite, l'asma e l'enfisema polmonare;
- Malattie restrittive → Il sintomo principale è la ridotta capacità d'espansione del tessuto polmonare con un conseguente minor flusso d'aria in fase d'inspirazione. Rientrano in tale categoria: l'interstiziopatia polmonare, la fibrosi polmonare ed il versamento pleurico;

- Malattie vascolari → È tipicamente caratterizzata dalla presenza di problemi sia a livello dei capillari alveolari che a livello delle diramazioni delle arterie polmonari. Tra le principali patologie vi sono: l'embolia polmonare, l'edema polmonare e l'ipertensione polmonare;
- Malattie infettive/ambientali → L'elemento caratterizzante è l'origine virale, batterica o fungina della condizione. Le infezioni più comuni dell'apparato respiratorio sono: la polmonite, la tubercolosi e la tracheite [1].

1.3 Covid-19

1.3.1 L'origine della pandemia e le conseguenze a livello globale

Il 31 dicembre 2019, i funzionari della Commissione di Sanità Pubblica della Provincia di Hubei, Cina, hanno informato l'Organizzazione Mondiale della Sanità (OMS) circa un focolaio epidemico di polmonite atipica severa, ad eziologia sconosciuta verificatisi nell'area urbana di Wuhan e caratterizzata da sintomi quali febbre, polmonite, tosse secca, dispnea ed insufficienza respiratoria.

Il 7 gennaio 2020, le autorità hanno identificato un nuovo coronavirus, il SARS-CoV-2, quale responsabile dell'infezione polmonare, oggi denominata Covid-19. Da allora, si è verificata una rapida diffusione del virus in tutto il mondo, causando contagi e decessi che hanno colpito dapprima i paesi limitrofi come Giappone e Corea del Sud, fino ad interessare, da gennaio e febbraio 2020, anche Europa, Stati Uniti e Africa.

L'11 marzo 2020 l'Organizzazione mondiale della sanità (OMS) ha dichiarato di poter considerare l'epidemia di Covid-19 una pandemia [9].

Inizialmente, a causa della mancanza di un trattamento specifico disponibile, l'OMS e le autorità si affidarono a trattamenti comportamentali per interrompere la catena di diffusione del virus, pubblicando linee guida sull'igiene personale e sociale e sull'esercizio del distanziamento sociale.

Dalla sua rilevazione iniziale, il virus si è diffuso rapidamente in tutto il mondo, infettando oltre 660 milioni di persone, uccidendone oltre 6.7 milioni e modificando l'ordine socio-economico mondiale.

La pandemia di Covid-19 ha anche aperto nuovi scenari anche dal punto di vista dell'industria farmaceutica e delle pratiche vaccinali. È cambiata infatti la velocità di realizzazione di una grande numero di candidati vaccini, resa possibile sia dalla diffusione di dati open access relativi alla sequenza genomica del virus Sars-Cov-2, dalla disponibilità di nuove tecnologie e dalla semplificazione (senza trascurare le verifiche di efficacia e sicurezza) delle procedure di sperimentazione clinica e di immissione in commercio dei preparati.

Grazie a tutto ciò, nel giro di poco più di un anno sono nati i seguenti vaccini: il Covid-19 Vaccine Jansen della Johnson & Johnson, il Vaxzevria di Astra-Zeneca, il CoronaVac e lo Sputnik V. Pertanto, il continuo incremento di persone vaccinate e guarite da Covid-19 ha determinato una pressione selettiva sul SARS-CoV-2, favorendo la comparsa e la diffusione di nuove varianti virali [10].

In un tale scenario, il rilevamento precoce e l'utilizzo efficiente della tecnologia possono far risparmiare tempo e denaro.

1.3.2 Modalità di trasmissione

Secondo i Centers for Disease Control (CDC) il coronavirus può essere trasmesso e diffuso principalmente attraverso uno stretto contatto tra persone vicine. Come altri virus respiratori, la trasmissione avviene principalmente quando una persona infetta da SARS-CoV-2 tossisce, canta, parla o semplicemente respira, a causa dell'emissione di particelle denominate droplet (goccioline) di dimensioni variabili [11]. Pertanto, le principali fonti di infezione sono quindi le stesse persone infette da SARS-CoV-2 [9].

1.3.3 Manifestazioni cliniche

Il Covid-19 è causa di una sindrome respiratoria abbastanza simile alla polmonite. Il quadro clinico risulta ampio e molto eterogeneo, ma le principali manifestazioni sono: febbre, tosse secca, mancanza di respiro, costrizione toracica, indolenzimento muscolare e affaticamento. Anche i disturbi dell'olfatto e del gusto sembrano essere sintomi relativamente frequenti. Altri sintomi segnalati meno frequentemente includono mal di testa, diarrea, nausea, vomito e dolore addominale [12]. La malattia generalmente colpisce i polmoni negativamente, con effetti che tendono a permanere anche per mesi.

Circa l'80% delle persone infette sviluppa la patologia in forma moderata o lieve. Nel rimanente 20% dei casi più severi, solo il 5% è in condizioni critiche e sviluppa una polmonite molto grave ed altre complicazioni gravi che includono sindrome da distress respiratorio acuto (ARDS), acidosi metabolica difficile da trattare, shock settico, coagulopatia ed infine insufficienza multiorgano [13]. Nei pazienti con forma severa della malattia, ma non necessariamente in condizioni critiche, si manifestano dispnea e/o ipossiemia.

Inoltre, come evidenziato in numerosi articoli presenti in letteratura, l'infezione da Covid-19 può generare complicanze cardiache [9].

Dunque, una diagnosi tempestiva della positività al Covid-19 ed il successivo utilizzo di farmaci e trattamenti possono aiutare a prevenire in larga misura il peggioramento delle condizioni del paziente che, in casi estremi, ne determinano il decesso [11].

I pazienti più fragili come gli anziani, i bambini piccoli, i pazienti immunocompromessi ed i pazienti con condizioni mediche preesistenti (inclusi diabete, patologie cardiovascolari, malattie respiratorie, obesità e malattie renali ed epatiche croniche) possono essere più vulnerabili a questo tipo di infezione, così come avviene per la normale polmonite [12].

1.3.4 Manifestazioni cliniche a livello polmonare

La principale manifestazione clinica del Covid-19 è la polmonite interstiziale (polmonite da Covid-19), un'inflammatione acuta degli alveoli polmonari la cui severità può compromettere la respirazione fino a generare forme gravi di ARDS che richiedono supporto esterno (intubazione o ventilazione meccanica) e, in casi estremi, mettere in pericolo la vita dei soggetti colpiti [14][13]. Quando si parla di interstizio, si intende un'unità particolare situata fra alveoli e capillari che viene esaminata principalmente mediante tecniche radiologiche. Le polmoniti interstiziali sono caratterizzate da un accumulo dell'edema e dell'infiltrato infiammatorio cellulare negli spazi interstiziali e solo negli stadi più avanzati della malattia iniziano a riempire gli spazi cavi, prima in maniera subtotale ("ground glass") e poi totale (consolidamento) [9].

Nelle forme più lievi, la guarigione dalla polmonite da Covid-19 richiede da 7 a 10 giorni circa senza danni a livello polmonare. Tuttavia, nelle forme più gravi, la prognosi è più lenta -dalle 3 alle 6 settimane- e con possibili effetti a lungo termine.

In alcuni soggetti, soprattutto anziani di sesso maschile con patologie pregresse, le cui condizioni mediche hanno richiesto il ricovero in terapia intensiva o un lungo periodo di supporto ventilatorio, è stato documentato un incremento del 20% dello sviluppo di fibrosi polmonare, una patologia polmonare grave ed irreversibile.

Inoltre, in pazienti con forme più severe di polmonite da Covid-19, gli effetti sono vari e possono comprendere tosse cronica, fibrosi polmonare e alterazioni vascolari del distretto polmonare associate alla dispnea ed al perdurare di un'insufficienza respiratoria che richiede il continuo ricorso all'ossigenoterapia [15].

Capitolo 2

Metodi di diagnostica

La diagnosi precoce di Covid-19 è uno dei punti chiave nei tentativi di prevenire la diffusione del virus e mantenere la popolazione sana, limitando dunque la mortalità dovuta alla pandemia. Si analizzano le diverse tipologie di test sviluppati per la diagnosi dell'infezione.

2.1 Test molecolare

Il test molecolare Covid-19, comunemente noto come tampone molecolare, è l'esame diagnostico di riferimento internazionale per la rilevazione del coronavirus SARS-CoV-2. Questo test rileva tracce specifiche del genoma del coronavirus da tamponi nasofaringeo od orofaringei, confermando (o non confermando) la positività di soggetti sintomatici, pre-sintomatici o asintomatici, anche in caso di bassa carica virale.

I campioni prelevati dalle secrezioni respiratorie di naso e bocca vengono processati utilizzando una tecnica di biologia molecolare chiamata Real Time reverse transcription PCR (RT-PCR), che combina due PCR: prima una PCR e, successivamente, una Real Time PCR per il rilevamento del genoma (RNA) del virus SARS-CoV-2. La PCR ("Polymerase Chain Reaction" ovvero Reazione a Catena della Polimerasi) è una tecnica che permette di clonare ed amplificare in provetta segmenti selezionati di DNA attraverso ripetuti cicli di sintesi, utilizzando inneschi specifici e la DNA polimerasi, senza che siano alterati i rapporti di concentrazione relativi dell'RNA virale [16].

Tuttavia, l'elevato livello di specificità del test RT-PCR non è affiancato da un altrettanto alto livello di sensibilità, che potrebbe essere relativamente bassa a causa dell'alto tasso di falsi negativi. Il test è più accurato quando la carica virale è elevata durante la fase sintomatica. Diversi fattori possono portare a un risultato falso negativo, tra i quali:

- scarsa qualità del campione;
- il campione viene raccolto troppo presto (per esempio tra l'esposizione a SARS-CoV-2 e l'insorgenza dei sintomi) o in maniera tardiva durante l'infezione;
- manipolazione e spedizione inadeguate del campione.

2.2 Test antigenico rapido

I tamponi antigenici rapidi – indicati anche come test rapidi Covid-19 – sono test che ricercano nel muco nasale o nella faringe la presenza di proteine virali (antigeni) espressi sulla superficie di SARS-CoV-2.

È quindi possibile rilevare se è in corso l'infezione da Covid-19 al momento dell'esame. Inoltre, tale test è probabilmente l'esame più conosciuto, perché effettuato anche in farmacia.

Tuttavia, a differenza del test molecolare (esame "gold standard" per la diagnosi di Covid-19), il tampone rapido non rileva tracce del genoma del virus SARS-CoV-2, ma rileva alcune componenti del virus (proteine Spike e Nucleocapside) che indicano l'avvenuta esposizione all'agente virale [17].

2.3 Tecniche di imaging radiologiche

In un contesto di sintomatologia aspecifica e di quadri clinici molto eterogenei in termini di gravità e progressione, il contributo dell'imaging integrato può essere importante non solo per la diagnosi di polmonite, ma anche per monitorare l'efficacia della terapia e per ottenere informazioni prognostiche.

2.3.1 Radiografia del torace

L'esame radiografico standard del torace è caratterizzato da una bassa sensibilità nel rilevamento delle alterazioni polmonari più precoci causate dal Covid-19, caratterizzate da opacità a "vetro smerigliato". Pertanto, a parte la diagnosi iniziale per escludere altre polmoniti alveolari batteriche, non è l'esame radiologico più indicato.

Negli stadi avanzati dell'infezione, l'esame radiologico mostra opacità alveolari multifocali bilaterali con tendenza a convergere fino all'opacamento completo del polmone [9].



Figura 2.1: Dispositivo per tomografia computerizzata (TC)

2.3.2 Tomografia computerizzata del torace

In radiologia, la tomografia computerizzata (TC o, in inglese, Computer Tomography) è una tecnica diagnostica di imaging che, mediante l'utilizzo di radiazioni ionizzanti (i raggi X), consente la scansione di sezioni corporee del paziente in esame, per la diagnosi e lo studio di numerose patologie. Si tratta di un esame radiologico in cui i dati sono raccolti facendo passare vari fasci di raggi X nell'area interessata e si rielaborano i dati sfruttando la potenza di calcolo di un computer (motivo per cui si definisce "computerizzata") al fine di ricostruire immagini volumetriche, tridimensionali, dei vari tipi di tessuto [18].

La radiografia tradizionale ha un importante limite in quanto si basa su una geometria di tipo proiettivo (un oggetto tridimensionale viene proiettato sul piano su cui si trovano i sensori), comportando la perdita di una dimensione che in alcuni casi può causare degli errori di valutazione. Ciò è particolarmente rilevante, ad esempio, quando si ha un'interfaccia tra un tessuto più assorbente e uno meno in quanto il primo tende a mascherare tutti gli altri tessuti.

Un'altra problematica è la dinamica limitata delle immagini di radiografia tradizionale. Il dispositivo di tomografia computerizzata è stato progettato per porre rimedio a queste 2 limitazioni: la dinamica dell'immagine è molto maggiore rispetto a quella di un'immagine radiologica, migliorando, di conseguenza, anche il contrasto tra tessuti diversi. Inoltre, vi è la possibilità di ottenere una rappresentazione tridimensionale di un volume, in modo da evitare la perdita di una delle dimensioni e poter eseguire anche calcoli sull'intero volume che sull'immagine bidimensionale non era possibile svolgere. Come riportato in Figura 2.1, un dispositivo per TC è composto da un tomografo (o scanner), da un sistema computerizzato di calcolo

dotato di processori ausiliari dedicati per ridurre i tempi di calcolo necessari agli algoritmi ricostruttivi (fino a meno di 1 secondo per immagine), e da una stazione di consultazione dotata di display e di dispositivi interattivi che consentono una manipolazione delle immagini. Il tomografo è costituito da un gantry contenente il tubo radiogeno che ruota attorno al paziente emettendo uno o più fasci collimati di raggi X (RX) sui sensori opposti. Una schiera di detettori (sensori) rileva il fascio di RX, attenuato in seguito al passaggio nei tessuti, e trasmettono all'elettronica l'indicazione sulla differenza fra i raggi X rilevati e quelli emessi (definendo dunque i raggi assorbiti dai tessuti del corpo). A questo punto si procede alla ricostruzione della singola fetta utilizzando l'algoritmo di retroproiezione filtrata; impilando tutte le sezioni si riesce a ricostruire un volume che, in seguito, può essere visualizzato. Contestualmente, il paziente è disteso su un lettino che si muove orizzontalmente, passando attraverso un'apertura circolare di diametro di circa 50 cm.

Esistono almeno 2 modalità di acquisizione differenti:

1. Scansione assiale: è la scansione tradizionale (anche definita stop and shoot) in cui il lettino si muove ad intervalli di tempo regolari; durante l'acquisizione il lettino rimane fermo e, una volta terminata la scansione di una fetta, si sposta per acquisire la slice successiva.
2. Scansione a spirale (anche definita spiral-CT): è la scansione in cui il lettino si muove a velocità costante ed il tubo radiogeno è in rotazione continua sul suo supporto acquisendo in modo continuo i dati dal paziente, disegnando una sorta di spirale intorno al paziente.

Poiché l'esame prevede l'utilizzo di raggi X, è opportuno non abusarne e per questo motivo i fasci di raggi X sono dosati in base alle esigenze ed alle caratteristiche del paziente. Infatti, il rischio di tumori e leucemie incrementa in relazione alla dose di radiazioni a cui si è esposti, soprattutto nei bambini e nei giovani; per tale motivo occorre soppesare, per ciascuna casistica, questi possibili rischi con i benefici attesi dall'esame.

Le scansioni TC sono costituite, in media, da un centinaio di sezioni (o slices) allineate perpendicolarmente all'asse definito dalla colonna vertebrale del paziente. Le slice ottenute hanno generalmente una dimensione di 512 x 512 pixel ed il loro numero, per ciascuna scansione, varia in funzione della distanza tra le sezioni e dell'estensione dell'organo in esame. L'intero esame richiede, mediamente, un tempo che va da 5 a 15 minuti grazie alla capacità dei tomografi moderni di produrre una slice in una frazione di secondo [19].

Hounsfield Unit (HU)

Indipendentemente dalla modalità di scansione utilizzata, l'attenuazione subita dai raggi X varia in funzione della densità elettronica dei tessuti attraversati:

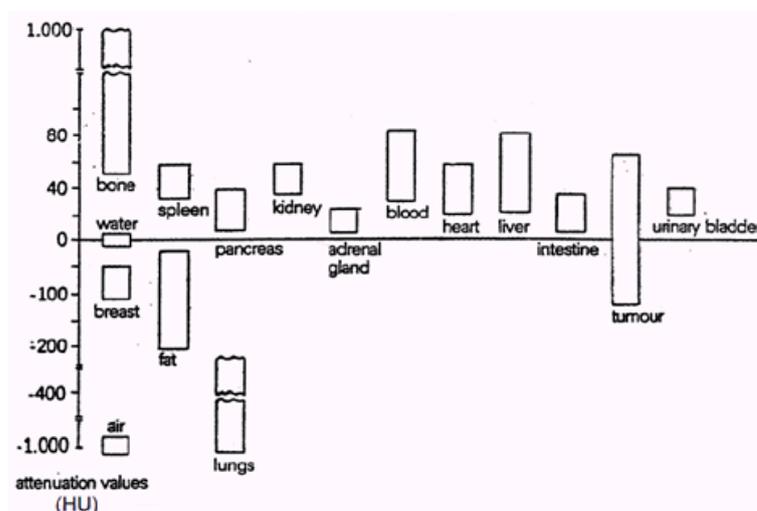


Figura 2.2: Scala Hounsfield relativa a diversi tessuti

un tessuto più denso, infatti, si oppone maggiormente al passaggio della radiazione. Di conseguenza, i tessuti ossei, avendo elevata densità, attenuano notevolmente la radiazione mentre gli organi ricchi d'aria come i polmoni la lasceranno passare più inalterata. Ogni pixel di ciascuna slice rappresenta le proprietà di assorbimento di un piccolo volume del corpo identificato dal pixel stesso. Tale assorbimento viene descritto con un valore espresso non in unità assolute ma su una scala correlata all'attenuazione dell'acqua in unità denominate Hounsfield Unit (HU). Pertanto, per calcolare il corrispondente valore in HU di un certo tessuto X, noto il relativo coefficiente di attenuazione μ_X ed indicando il coefficiente di attenuazione dell'acqua con μ_{acqua} , si può fare riferimento alla relazione seguente:

$$\mu(HU) = \frac{\mu_X - \mu_{\text{acqua}}}{\mu_{\text{acqua}}} \cdot 1000$$

Tutti i coefficienti di attenuazione lineare dei vari tessuti vengono rapportati all'attenuazione dell'acqua e moltiplicati per 1000 per definire il cosiddetto coefficiente di attenuazione lineare in unità Hounsfield; i valori di tale coefficiente, calcolato per i diversi tessuti, permettono di definire la scala Hounsfield (riportata in Figura 2.2). Segue dunque che i tessuti come l'osso compatto hanno valori alti positivi, mentre i tessuti molli, avendo un'attenuazione simile all'acqua, sono caratterizzati da valori in unità Hounsfield vicini allo zero mentre l'aria, la cui attenuazione è praticamente nulla, ha un coefficiente negativo vicino a -1000 in HU e rappresenta quindi il limite inferiore della scala Hounsfield [19].

Ogni gradazione o intensità del grigio di un pixel di una slice è correlata all'attenuazione del fascio di raggi X caratterizzante il tessuto rappresentato. Dunque,

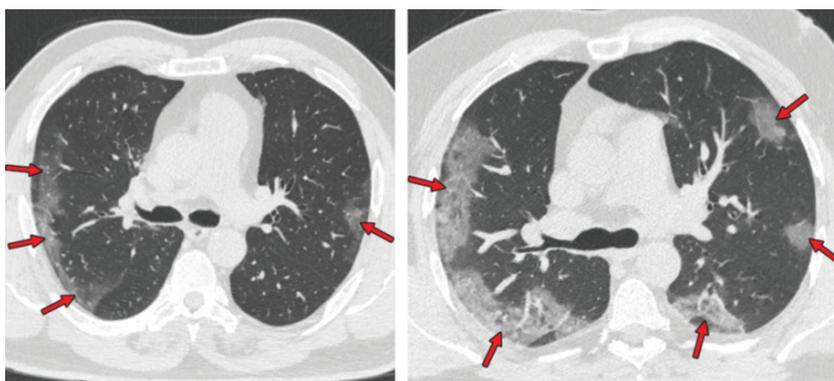


Figura 2.3: Tipiche TC di due pazienti con polmonite da COVID-19, che mostrano aree bilaterali di opacità a vetro smerigliato [12]

i valori della scala Hounsfield sono tutti potenzialmente traducibili in diverse tonalità di grigio che vanno dal nero, rappresentante i tessuti a bassa densità come i polmoni, al bianco utilizzato per rappresentare tessuti ad elevata densità come l'osso. Complessi algoritmi matematici e tecniche di ricostruzione delle immagini consentono di generare immagini delle sezioni in esame a partire dai parametri d'attenuazione rilevati. L'output del dispositivo per TC è un'immagine che associa i toni di grigio alle unità HU.

2.4 Descrizione delle principali lesioni associate al Covid-19 osservabili con TC

I primi studi hanno suggerito che le immagini TC presentano un elevato numero di potenziali indicatori di infezione associati al Covid-19.

La polmonite da Covid-19 mostra un quadro TC aspecifico e vario, simile ad altre infezioni polmonari. I reperti di più comune riscontro alle scansioni TC mostrano aree a “vetro smerigliato” o “ground glass” multifocali bilaterali, associate ad aree con consolidazioni con distribuzione a chiazze, prevalentemente periferiche, e con un maggior coinvolgimento delle regioni posteriori e dei lobi inferiori, come si può ben riscontrare delle scansioni TC riportate in Figura 2.3. Come evidenziato in vari studi, nella polmonite Covid-19 sono stati osservati il pattern “ground glass puro”, focale o multifocale, ed il pattern “crazy paving”, caratterizzato dalla presenza di aree di “ground glass” sovrapposto ad ispessimento liscio dell'interstizio interlobulare ed intralobulare. Meno comune è la presenza esclusiva di consolidamenti, di calcificazioni, del “reversed halo sign” (area focale di “ground glass” delimitata da anello periferico più o meno completo di consolidazione) o di versamenti pleurici.

Se nelle prime fasi della patologia sono presenti opacità “ground glass” periferiche bilaterali, con il progredire della malattia, aumentano la percentuale di parenchima interessato, comportando una sua maggiore estensione verso una zona più centrale; le aree caratterizzate già in precedenza da “ground glass” aumentano di densità, fino ad arrivare alla vera e propria consolidazione parenchimale.

La TC è utile non solo per la diagnosi precoce ma anche nella fase intermedia del processo morboso del paziente in quanto consente di definire quanto sia necessario essere veloci e terapeuticamente impattanti, influenzando pertanto il decorso clinico del paziente [9].

Capitolo 3

Apprendimento automatico

In questo capitolo viene presentata una breve panoramica sull'intelligenza artificiale e sulle reti neurali, focalizzando l'attenzione sulle reti neurali convoluzionali (CNN) e sulla loro architettura.

3.1 Intelligenza Artificiale (IA)

Come descritto in precedenza, il metodo standard di diagnosi dell'infezione da Covid-19 è la tecnica della RT-PCR ma questa presenta alcune limitazioni quali il tempo necessario per l'analisi e gli errori principalmente dovuti ad un quadro aspecifico della patologia, con sintomi simili alle più comuni influenze virali. Per tale motivo si è posta una grande attenzione all'utilizzo di reti neurali, che appartengono al campo dell'intelligenza artificiale, consentendo una diagnosi di positività più rapida ed accurata.

Dagli anni '50 la maggior parte dei ricercatori informatici ha lavorato sull'apprendimento automatico per eseguire diversi task, emulando la struttura e le funzionalità delle reti neurali biologiche. Nel contesto della pandemia da Covid-19, questa tecnologia ha ottenuto una grande attenzione in quanto si è rivelata estremamente utile nell'ambito di diagnosi, nell'identificazione dei pazienti più a rischio e nella prevenzione della diffusione dell'epidemia [20].

Per IA si intende un metodo di analisi dei dati ispirato al funzionamento del cervello umano basato sull'esperienza e che sviluppa vari algoritmi di calcolo noti come reti neurali, estremamente utili in molteplici ambiti quali: data analysis, selezione ed estrazione di features dai dati e classificazione ovvero assegnazione dei dati in input ad una specifica categoria di appartenenza. È molto importante sottolineare che l'IA fornisce risposte non certe in assoluto ma certe con un determinato livello di "confidenza", cioè con una certa probabilità [21].

3.2 Machine Learning e Deep Learning

Grazie all'aumento della ricerca in questo campo negli ultimi anni, i sistemi si sono evoluti sempre di più, dando origine dapprima al machine learning (ML) ed in seguito ad un nuovo sottoinsieme del ML chiamato Deep Learning (DL) che ha anche ricevuto molta attenzione durante la pandemia di Covid-19.

Il machine learning ed il deep learning sono sottoclassi dell'intelligenza artificiale che utilizzano algoritmi di apprendimento matematico-statistico, imparando automaticamente senza essere stati programmati per lo svolgimento dello specifico task.

Un flusso di lavoro di ML per un task di classificazione richiede che vengano esplicitamente indicate le features su cui lavorare, definite precedentemente con un processo di feature extraction svolto dall'operatore. Queste caratteristiche vengono quindi organizzate in un dataset ed utilizzate per costruire modelli che categorizzano gli oggetti presenti nelle immagini.

Il flusso di lavoro di DL, invece, offre notevoli vantaggi in quanto automatizza il processo di individuazione ed estrazione delle features significative e non richiede operazioni preliminari da parte dell'operatore. D'altra parte, i sistemi di deep learning necessitano di un grande database o di numerose informazioni fornite in input per un funzionamento ottimale. Inoltre, il DL esegue un apprendimento end-to-end, svolgendo autonomamente l'elaborazione dei dati grezzi di training e migliorando la sua accuratezza nel tempo durante l'allenamento. I primi layer si occupano dell'elaborazione dei dati di input o apprendono le features più semplici e l'output di tali livelli viene inviato ai layer successivi che eseguono l'apprendimento di features più complesse. Il deep learning è quindi maggiormente adatto all'elaborazione di dati di dimensione più grande e di elevata complessità [21][20].

3.3 Reti neurali

Lo sviluppo di reti neurali negli ultimi decenni sta presentando un trend sempre più in crescita, con continui progressi che consentono la costruzione e l'addestramento di reti sempre più complesse in grado di apprendere attraverso meccanismi in parte analoghi a quelli dell'intelligenza umana. In particolare, sono sempre più in grado di avvicinarsi (in alcuni casi anche superare) all'efficienza ed alle capacità del cervello umano nella risoluzione di alcuni task [22].

Le reti neurali, note anche come reti neurali artificiali (in inglese artificial neural network o ANN) rappresentano il componente principale degli algoritmi di deep learning. Il loro nome e la loro struttura rimandano al sistema nervoso umano ed alla sua organizzazione in neuroni e sinapsi, dei quali riprendono le modalità di comunicazione inter-neuronale tramite segnali biologici [23].

All'interno del cervello esiste una vera e propria rete costituita da centinaia di milioni di neuroni che si è tentato di riprodurre con un sistema che potesse funzionare altrettanto bene nell'elaborazione di informazioni [24]. Una ANN è un modello di calcolo con un'architettura costituita principalmente da tre livelli (layer):

- Input layer: livello che riceve in input le informazioni ed i dati del problema dall'operatore;
- Hidden layer: livello di collegamento che si occupa di apprendere le features dai dati in input;
- Output layer: livello finale che genera l'output della rete.

I livelli sono interconnessi tramite nodi (per analogia con le reti neurali biologiche possono essere chiamati neuroni) e si esegue un'elaborazione in cascata dei dati in ingresso, in cui ogni layer utilizza l'output del livello precedente come input da processare (paragonabile ad un impulso elettrico per il neurone biologico) [21].

Ciascun layer è costituito da numerosi neuroni, ciascuno con una propria funzione di attivazione e collegati tra loro mediante connessioni pesate (si parla di pesi sinaptici).

Nel progetto di tesi si lavora su un problema di classificazione ed il compito della rete consiste nel determinare la categoria di appartenenza dell'input. Per fare ciò, l'output di ciascun neurone dell'ultimo layer viene modificato generalmente mediante la funzione softmax (la più comune tra le funzioni applicate all'output layer) che trasforma i valori di output in una distribuzione di probabilità normalizzata, permettendo una migliore individuazione del neurone che definisce la classe di appartenenza del dato in input alla rete e al quale è associato una probabilità maggiore.

Le ANN, costituite da due o tre layer di neuroni interconnessi, sono definite reti neurali superficiali. Al contrario, le reti profonde, note come reti di Deep Learning (DL), possono essere costituite anche da centinaia di layer. Entrambe sono sistemi di intelligenza artificiale che apprendono direttamente dai dati in input.

Il DL è stato teorizzato per la prima volta negli anni '80, ma solo di recente ha avuto un grande sviluppo per due ragioni principali:

- la possibilità di ottenere una grande mole di dati precedentemente etichettati che consente un training più efficace della rete;
- l'utilizzo di Graphic Processing Unit (GPU) ad alte prestazioni che presentano un'architettura molto efficiente per lo sviluppo di reti di deep learning (per le quali è richiesta un'elevata potenza computazionale) e consentono anche la riduzione dei tempi di addestramento fino ad una durata di poche ore.

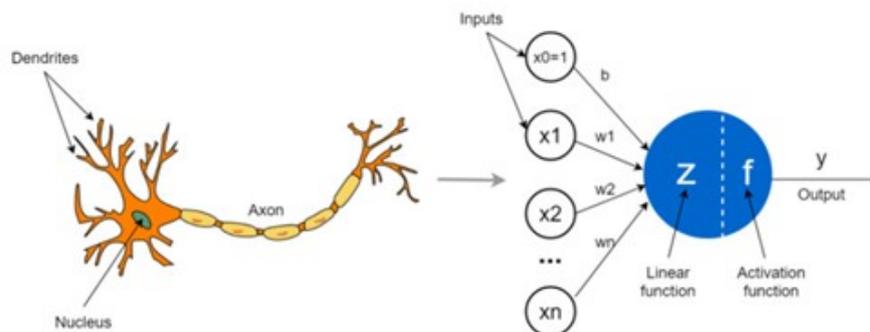


Figura 3.1: Rappresentazione neurone biologico e neurone artificiale [25]

Le due principali modalità di allenamenti delle reti sono:

- Addestramento della rete da zero (in inglese, ‘from scratch’) → per questo allenamento è necessario avere a disposizione un dataset ampio composto da dati etichettati.
- Addestramento con transfer learning → viene utilizzato nella maggior parte delle applicazioni di DL e consiste nella modifica e riaddestramento un modello preesistente e preaddestrato per riconoscere categorie di oggetti diversi [21].

3.4 Il neurone artificiale ed il suo modello matematico

L’elemento costitutivo di base di ogni rete neurale artificiale è il neurone artificiale, il cui design e funzionalità si basano sull’osservazione dei neuroni biologici che, a loro volta, rappresentano l’unità fondamentale delle reti neurali biologiche. Le somiglianze possono essere osservate in Figura 3.1 in cui si riporta a sinistra la rappresentazione di un neurone biologico con il soma, dendriti e assone mentre a destra si riporta la rappresentazione di un neurone artificiale con i suoi input, pesi, funzione di trasferimento, bias e output.

Un neurone biologico è composto da:

- Soma o corpo cellulare → sede del nucleo che regola le attività del neurone stesso. Da qui si dipartono i dendriti e l’assone;
- Dendriti → ricevono segnali sotto forma di impulsi da parte di altre cellule nervose;

- Assone → collegato al corpo cellulare ma presenta una struttura più robusta e spessa rispetto ai dendriti e permette la propagazione e l'invio dell'impulso verso i dendriti di altre cellule nervose tramite le sinapsi [24].

Il primo modello di neurone artificiale, proposto nel 1943 da due ricercatori statunitensi McCulloch e Pitts, prevedeva una struttura abbastanza simile al neurone biologico, nel quale l'informazione giunge al neurone attraverso i dendriti ed il soma si occupa dell'elaborazione dell'informazione e della sua trasmissione attraverso l'assone. Analogamente, nel neurone artificiale, l'informazione viene ricevuta a livello del corpo centrale del neurone artificiale tramite input ponderati e qui si procede alla loro somma, con l'aggiunta di un bias, ed all'elaborazione di tale somma con una funzione di trasferimento. Infine, un neurone artificiale invia al neurone successivo le informazioni elaborate [26]. Si riporta di seguito la descrizione matematica che caratterizza il funzionamento di ciascun modello di neurone artificiale:

$$y = f\left(b + \sum_{i=1}^n x_i \cdot w_i\right) \quad (3.1)$$

Dove i termini che compaiono nella relazione precedente corrispondono a:

- y è il valore di output dal neurone;
- f è una funzione di attivazione;
- b è il bias;
- n è il numero di elementi;
- x_i è il valore di ingresso al neurone;
- w_i è il valore del peso.

Come riportato dall'Equazione 3.1, la variabile incognita più importante del modello neurale è la funzione di attivazione, scelta sulla base del task che il neurone artificiale deve svolgere. In particolare, le principali funzioni di attivazione sono: funzione Step, funzione lineare e funzione non lineare (sigmoide, tangente iperbolica o ReLU).

La funzione *Step* è una funzione binaria che restituisce uno dei due possibili valori di output (ad esempio zero e uno) a seconda che venga superato o meno un valore di una soglia predefinito.

La funzione *lineare* definisce un output proporzionale all'input fornito.

La funzione *sigmoide* è una funzione non simmetrica rispetto allo zero che restituisce in output valori in un intervallo continuo tra 0 e 1, con un andamento a forma di S.

La funzione *tanh* (tangente iperbolica) è simile alla funzione sigmoidea ma è simmetrica rispetto all'origine, restituendo valori di output nel range tra -1 e 1 [20].

Infine, la *ReLU* (ovvero rectified liner unit) è una funzione di attivazione molto utilizzata che restituisce zero se la somma ponderata degli input è minore o uguale a zero, altrimenti restituisce il risultato stesso della somma.

È quindi possibile definire l'attivazione o la disattivazione di un neurone in base a queste funzioni di attivazione.

Nel 1957, lo psicologo visionario Rosenblatt, sviluppò il primo modello di rete neurale (chiamato *Perceptron*) all'interno della sua pubblicazione 'The Perceptron, a Perceiving and Recognizing Automaton'. Il Perceptron può essere pensato come un'unità che elabora gli input in modo analogo al neurone artificiale definito da McCulloch e Pitts, fornendo in output un solo valore. L'innovazione, rispetto al primo modello del neurone, è la possibilità di modificare i pesi e le soglie sulla base di un accurato addestramento automatico.

In particolare, l'apprendimento parte da valori dei pesi e delle soglie inizializzati in modo casuale per poi proseguire alla loro modifica in maniera proporzionale (costante di proporzionalità che viene definita learning rate) all'errore commesso tra il valore ottenuto ed il valore reale. Questo processo viene ripetuto, variando i pesi e le soglie per ridurre al minimo l'errore commesso. Al termine di tale operazione, i pesi vengono fissati e si può procedere alla fase di inference nella quale vengono posti in input alla rete nuovi dati per essere processati sulla base del task che la rete deve svolgere [27].

3.5 Rete Neurale Convolutionale (CNN)

Una Rete Neurale Convolutionale (in inglese *Convolutional Neural Network*, indicata anche con CNN o ConvNet) è il tipo di ANN più comunemente utilizzata nell'analisi delle immagini e la connessione tra i neuroni che la costituiscono trae ispirazione dall'organizzazione della corteccia visiva nel cervello. Sono chiamate così in quanto al loro interno si utilizza un'operazione matematica detta convoluzione che consente l'estrazione delle features dall'immagine.

Le CNN, come le altre ANN, non necessitano dell'estrazione manuale delle features da immagini, video, testi o suoni forniti in input per il processo di apprendimento e sono progettate per apprendere automaticamente le relazioni delle features che estrae [21]. Tuttavia, le CNN sono molto costose a livello computazionale in quanto necessitano di una grande mole di dati per stimare i suoi milioni di parametri apprendibili, richiedendo, dunque, un'unità di elaborazione grafica (GPU) per l'addestramento del modello [28].

Quando si analizza un'immagine, questa viene rappresentata come una matrice di pixel (o valori) ed ognuno di essi rappresenta un dato di input alla rete neurale. Le ANN solitamente non vengono utilizzate per l'analisi di immagini in quanto presentano una limitazione importante riguardante il numero di neuroni di input

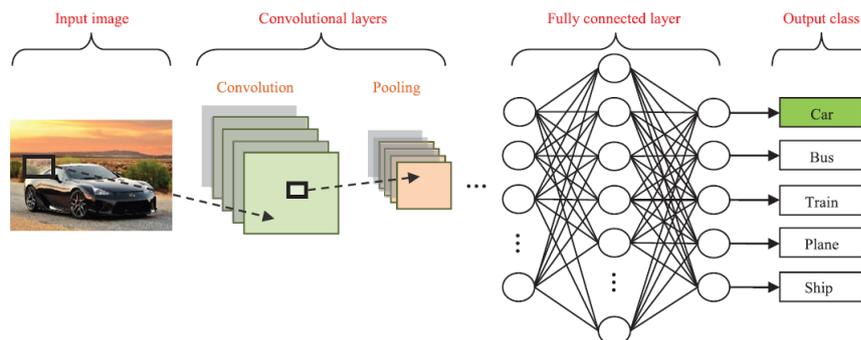


Figura 3.2: Tipica architettura CNN di classificazione [30]

nell'input layer della rete, che risulterebbe troppo elevato poiché associato al numero totale dei pixel dell'immagine. Ciò si traduce in un numero molto elevato di pesi e collegamenti tra l'input layer e l'hidden layer, aumentando notevolmente la complessità dell'algoritmo. Le reti neurali convoluzionali sono state sviluppate proprio per superare questi limiti.

Le CNN sono maggiormente in grado di lavorare sulle immagini in quanto, rispetto ad una ANN classica, i layer hanno i neuroni disposti in 3 dimensioni: larghezza, altezza e profondità (da non confondere con la profondità della rete neurale che fa riferimento al numero di layer che la compone) [29]. Esistono diverse varianti di architetture CNN ma tutte, in generale, includono diversi elementi costitutivi come i layer convoluzionali (in inglese, convolutional layer), livelli di pooling (in inglese, pooling layer) e livelli completamente connessi (in inglese, fully connected layer).

Un'architettura tipica è costituita da ripetizioni di un insieme di diversi livelli convoluzionali, un pooling layer, seguito da uno o più livelli completamente connessi. Le features estratte possono diventare progressivamente più complesse quando propagano all'interno della rete verso i layer più profondi [28]. Sebbene questa sia l'architettura più comune trovata in letteratura, negli anni sono state proposte diverse modifiche architetture con l'obiettivo di migliorare l'accuratezza della classificazione e ridurre i costi computazionali. La Figura 3.2 illustra la tipica architettura CNN per un'attività di classificazione delle immagini.

3.5.1 Architettura delle reti neurali convoluzionali

Input layer

Il primo strato della rete prende in input le immagini che costituiscono il dataset e che devono essere processate.

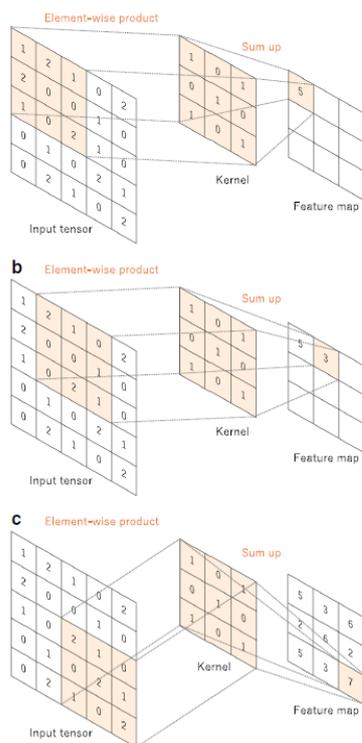


Figura 3.3: Operazioni di convoluzione [28]

Layer convoluzionale

Il livello convoluzionale è il componente principale dell'architettura di una CNN e combina operazioni lineari e non lineari, rispettivamente come l'operazione di convoluzione e la funzione di attivazione, per estrarre le principali features dall'immagine in input.

La convoluzione è un'operazione lineare in cui un piccolo array, chiamato kernel, viene applicato all'input, che è anch'esso un array di numeri, chiamato tensore. In particolare, come riportato in Figura 3.3, si calcola un prodotto elemento per elemento tra ciascun valore del kernel ed ogni posizione del tensore e si sommano i risultati per ottenere il valore di output nella posizione corrispondente del tensore di output, chiamato mappa delle features.

Il kernel scorre su tutta l'immagine, motivo per cui viene anche chiamato sliding window (finestra scorrevole). Questo processo viene ripetuto applicando più kernel con caratteristiche diverse in modo da generare un certo numero di mappe di features; pertanto, kernel diversi possono essere considerati come estrattori di features diverse, consentendo così alla rete di imparare a distinguere alcune

caratteristiche dell'input quali contorni, particolari insiemi di colori o forme.

Due importanti iperparametri che caratterizzano l'operazione di convoluzione sono la dimensione del kernel, tipicamente 3×3 , ma a volte 5×5 o 7×7 , ed il numero di kernel applicati all'input che generalmente è arbitrario e determina la profondità delle mappe delle features di output ad ogni layer. Per svolgere l'operazione di convoluzione quando il centro del kernel si trova sul bordo, sovrapponendosi all'elemento più esterno del tensore di input, si utilizza la tecnica del padding, solitamente zero padding, in cui vengono aggiunte righe e colonne su ciascun bordo del tensore di input, consentendo al centro del kernel di sovrapporsi anche agli elementi posti alle estremità del tensore; in questo modo si conserva la dimensione della mappa di features dopo l'operazione di convoluzione e, di conseguenza, costruire una rete con più layer. Infatti, senza padding, ogni feature map sarebbe più piccola dopo ogni operazione di convoluzione, riducendo di molto il numero di possibili layer della rete. La distanza tra due posizioni successive del kernel è chiamata passo (stride) ed il suo valore tipico è 1 anche se, a volte, si può utilizzare un passo più grande per ottenere il downsampling delle mappe delle features.

Dunque, la dimensione della mappa delle features dipende da alcuni fattori che vengono definiti prima dell'inizio del processo di addestramento della rete: numero di filtri applicati, dimensione del kernel, dimensione dell'input, stride e padding [28].

Funzione di attivazione non lineare

Gli output dell'operazione di convoluzione vengono quindi passati attraverso una funzione di attivazione non lineare. Sebbene in passato fossero abbastanza utilizzate la funzione sigmoide o la tangente iperbolica (tanh) in quanto rappresentative a livello matematico del comportamento di un neurone biologico, attualmente la funzione di attivazione più comunemente utilizzata è l'unità lineare rettificata (ReLU), che calcola semplicemente il $\max(0, x)$, consentendo un più efficace addestramento della rete [28].

Pooling layer

Questo layer semplifica l'output mediante un'operazione di downsampling utile a ridurre la dimensione, nuovamente tramite l'utilizzo di kernel, delle mappe delle features e del numero di parametri appresi dalla rete. Si ottengono, dunque, nuove mappe di features con le sole caratteristiche essenziali all'addestramento. Esistono diverse tipologie di pooling layer: Max pooling, Average pooling, Global Max pooling e Global Average Pooling. Le prime due sono le più comuni e permettono di estrarre rispettivamente il valore massimo ed il valore medio da sottoporzioni (anche definite patch) delle mappe delle features in input, scartando tutti gli altri valori. Il global Max pooling ed il global Average pooling effettuano un downsampling

estremo in cui la dimensione della mappa di features viene ridotta ad un array 1×1 prendendo rispettivamente il solo massimo o la sola media fra tutti i valori di ogni mappa, mentre, ancora una volta, la profondità rimane invariata. Questa funzione solitamente viene applicata solo una volta, prima dei fully connected layers [28].

Dropout layer

A causa delle grandi dimensioni dell'immagine, i neuroni consentono ai layer di memorizzare i dati ma questa è un'attività indesiderata che deve essere evitata in modo da eseguire una corretta previsione. Dunque, i dati indesiderati creati vengono eliminati casualmente in questo layer, evitando così il possibile overfitting della rete [31].

Fully connected layer (FC layer)

Un livello completamente connesso presenta neuroni con connessioni a tutte le attivazioni del layer precedente.

Le mappe delle features ottenute come output del pooling layer vengono generalmente appiattite in un array unidimensionale di numeri all'interno di un Flatten layer che viene poi collegato ad uno o più strati completamente connessi. In caso si svolga un task di classificazione, il fully connected layer in genere ha lo stesso numero di nodi di output del numero di classi e fornisce in output solitamente le probabilità di appartenenza dell'input a ciascuna classe [28].

Funzione di attivazione dell'ultimo strato

Ogni livello completamente connesso è seguito da una funzione non lineare che deve essere selezionata in modo appropriato sulla base del task da eseguire. Nel caso di classificazione multiclasse si utilizza la funzione softmax che normalizza l'output dell'ultimo livello completamente connesso per ottenere le probabilità di appartenenza a ciascuna classe [28].

3.5.2 Addestramento delle reti

L'addestramento consiste in un algoritmo di ottimizzazione dei parametri e delle performance della rete neurale ed il metodo maggiormente utilizzato segue l'algoritmo di backpropagation in cui la loss function e l'algoritmo di ottimizzazione del gradient descent giocano un ruolo fondamentale.

Questo algoritmo consente di definire le caratteristiche dei kernel nei layer convoluzionali e di modificare i pesi nei fully connected layer in modo da minimizzare le differenze tra le truth-label del dataset e le predizioni della rete [28].

Generalmente, i valori dei pesi w e dei bias b che caratterizzano ogni neurone, in un primo momento sono inizializzati in modo casuale e, in seguito, vengono modificati durante l'addestramento della rete. L'algoritmo di backpropagation è costituito da due fasi: la forward propagation e la backward propagation. Nella prima fase si procede dal layer di input al layer di output, calcolando le attivazioni dei neuroni in base ai valori correnti dei pesi.

Nell'ultimo layer, il valore di output della rete viene confrontato con il valore di output desiderato mediante il calcolo della funzione di costo (in inglese, cost function), indicata con J , data proprio dall'errore tra l'output atteso e l'output della rete:

$$J(w, b) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}_i, y_i) \quad (3.2)$$

dove:

- L rappresenta la loss function (ne esistono diverse tipologie);
- \hat{y}_i indica l'output generato dalla rete relativo all' i -esimo input;
- y_i indica il rispettivo output atteso (truth label).

La scelta della loss function rappresenta una fase fondamentale in quanto quest'ultima è uno degli iperparametri e deve essere definita in base al task da svolgere.

La seconda fase consiste nella backward propagation in cui i pesi vengono modificati e corretti a partire dal layer di output fino al layer di input.

L'algoritmo di ottimizzazione di discesa del gradiente (gradient descent) aggiorna iterativamente i parametri apprendibili della rete, come kernel e pesi, in modo da minimizzare la loss e, di conseguenza, la cost function. Il gradiente permette di definire la direzione in cui la loss function è più ripida ed ogni parametro apprendibile viene aggiornato nella direzione negativa del gradiente, con un passo determinato sulla base di un iperparametro chiamato learning rate (o tasso di apprendimento). Matematicamente, il gradiente è esprimibile tramite una derivata parziale della loss rispetto a ciascun parametro apprendibile. Dunque, l'aggiornamento di ciascun singolo parametro segue la seguente formulazione:

$$w = w - \alpha \cdot \frac{\partial L}{\partial w} \quad (3.3)$$

dove:

- w è il parametro da ottimizzare (possono essere, ad esempio, i pesi o i kernel);
- α indica il learning rate;
- L definisce la loss function [28].

Sono state proposte ed utilizzate in letteratura molte versioni dell'algoritmo di gradient descent come *SGD* (Stochastic gradient descent), *RMSprop* o *Adam*.

Il learning rate è uno degli iperparametri più importanti per l'addestramento della rete in quanto influisce sulla velocità di apprendimento ed un maggior focus su di esso è riportato al Paragrafo 5.3.2.

3.5.3 Transfer Learning

Nel deep learning, il transfer learning è una tecnica ampiamente utilizzata in quanto è in grado di costruire modelli ben performanti in breve tempo. Generalmente, non si avvia il processo di apprendimento da zero (anche per gli alti costi computazionali) ma da modelli pre-addestrati su dataset di grandi dimensioni (il più comune è il dataset ImageNet costituito da 14 milioni di immagini divise in 1000 diverse classi) per risolvere task simili a quello per cui è stato progettato. Pertanto, spesso si usa importare modelli dalla letteratura (ResNet, DenseNet, VGG, Inception, MobileNet) sfruttando proprio il meccanismo del transfer learning [32].

3.5.4 Overfitting

L'overfitting è un comportamento indesiderato in cui un modello apprende regolarità statistiche specifiche dal Training set, adattandosi eccessivamente a questo dataset e, pertanto, fornendo previsioni accurate per i dati di addestramento ma non per i nuovi dati [33]. Il modo migliore per rilevare l'overfitting della rete consiste nel monitorare la loss function e l'accuracy su Training set e Validation set in quanto, una performance nettamente migliore sul Training set rispetto al Validation set indicherebbe un probabile overfitting sui dati di Training del modello. L'overfitting si verifica per diversi motivi, ad esempio:

- La ridotta dimensione dei dataset comporta che non vi siano campioni sufficienti da essere rappresentativi di tutti i possibili valori che possono assumere i dati di input. Dunque, la soluzione migliore è ottenere un dataset più ampio, in modo tale che il modello sia in grado di generalizzare meglio.
- I dati di Training presentano informazioni irrilevanti, definiti anche dati rumorosi.
- L'elevata complessità del modello [33].

Per ridurre questo comportamento, oltre all'utilizzo di un dataset più ampio che non sempre è possibile nell'ambito dell'imaging medico, sono state proposte ulteriori soluzioni che comprendono: la regolarizzazione (L1 o L2), il data augmentation, il batch normalization e la riduzione della complessità dell'architettura.

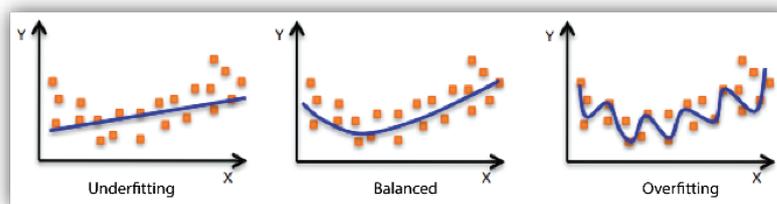


Figura 3.4: Esempi di underfitting o overfitting [35]

La regolarizzazione aggiunge alla loss function un fattore di penalizzazione, grazie al quale la loss aumenta quando i valori dei pesi sono molto alti. La riduzione dei valori dei pesi può limitare l'importanza dei dati di Training, ottenendo una maggiore generalizzazione su altri dataset. In questo modo si riduce la complessità del modello e soprattutto si riduce l'overfitting. Esistono due tipi di regolarizzazione: la *regolarizzazione L1* e la *regolarizzazione L2*.

La regolarizzazione L1 (o regolarizzazione Lasso) tende a ridurre i pesi a zero, con il rischio di eliminare alcuni input (poiché verranno moltiplicati con un valore nullo), mentre la regolarizzazione L2, nota anche come weight decay di cui si parlerà anche in seguito nel Paragrafo 5.3.2, penalizza i pesi del modello in modo che essi assumano solo valori piccoli [34]. Il dropout è un'ulteriore tecnica di regolarizzazione che imposta a zero, durante l'addestramento, le attivazioni selezionate in modo casuale, al fine di rendere il modello meno sensibile a pesi specifici nella rete.

Il data augmentation è un processo di modifica dei dati mediante trasformazioni come flipping, traslazione, cropping, rotazione e random delete, in modo che il modello non veda esattamente gli stessi input.

Il batch normalization normalizza in modo adattivo i valori di input del layer successivo, riducendo la dipendenza dall'inizializzazione e migliorando l'algoritmo di gradient descent consentendo di impostare valori più elevati di learning rate [28].

Un evento opposto all'overfitting consiste nell'underfitting che si verifica quando il modello non è in grado di ridurre l'errore su Training e di conseguenza non apprende i diversi pattern dei dati. In questo caso le possibili soluzioni prevedono addestramenti più lunghi, rimozione del rumore dai dati o la modifica dell'architettura della rete, con, ad esempio, un aumento del numero di layer [36].

Viene riportato in Figura 3.4 un esempio grafico dei fenomeni di underfitting ed overfitting dei processi di apprendimento delle reti.

Capitolo 4

Stato dell'arte

In questa sezione vengono brevemente descritti diversi studi o progetti di ricerca presenti in letteratura e condotti nel campo della diagnosi di polmonite da Covid-19 da immagini mediche come immagini RX o scansioni di tomografia computerizzata.

Le prestazioni finali sono fortemente influenzate dalle scelte riguardanti il metodo applicato per il pre-processing dei dati, le architetture utilizzate, i parametri e le modalità di allenamento ed anche l'aggregazione delle predizioni.

Il primo studio che viene presentato è stato condotto da Aswathy et al.[37] in cui si propone un'architettura a due stadi: il primo stadio si occupa di rilevare la positività al Covid-19 (Covid-19 Detection) mentre il secondo stadio si occupa di determinare la gravità della patologia (Covid-19 Severity Detection). Entrando nel dettaglio, è stato eseguito il data augmentation per migliorare la capacità di apprendimento.

Subito dopo il pre-processing delle immagini di input, è stata impiegata una rete ResNet-50 per il rilevamento di Covid-19 mentre per l'estrazione delle features atte a valutare la gravità dei pazienti sono state utilizzate due reti pre-addestrate, una ResNet-50 ed una DenseNet-201, ed una rete di backpropagation. Per ciascuna rete, il fully connected layer permette di estrarre 1000 features. Di conseguenza, si ottengono 2000 features per ogni immagine e per ridurre al minimo la dimensione viene utilizzato l'algoritmo di Principal Component Analysis (PCA). Il set di features ridotte viene posto come input alla rete feed-forward, la quale si occupa di classificare le immagini di input, utilizzando le features in input ottenute per ogni singola immagine, in categorie di gravità Alta, Media e Bassa. La struttura proposta ha fornito una sensibilità del 98.58% in fase di rilevazione dei pazienti positivi affetti da Covid-19 ed un'accuratezza complessiva del 98.5%. Per il rilevamento della gravità della condizione del paziente è stata ottenuta un'accuratezza media del 97.84%.

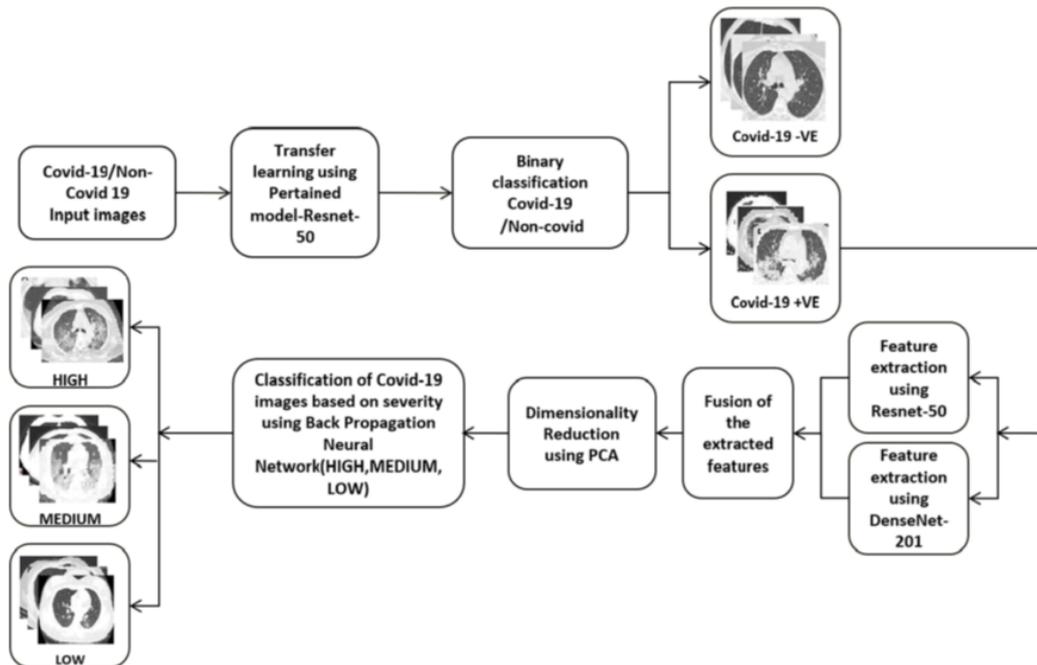


Figura 4.1: Architettura della soluzione proposta da Aswathy [37]

È stato seguito un altro approccio nello studio di Mishra et al.[38] in cui viene proposta una strategia basata sul metodo del decision fusion, che integra le previsioni di ciascuno dei diversi modelli di Deep CNN.

Innanzitutto, sono stati esaminati diversi modelli di base che comprendono: VGG-16, ResNet-50, InceptionV3, DenseNet-121 e DenseNet-201. Il principio fondamentale della tecnica del decision fusion è la possibilità di far fronte agli errori dei singoli modelli integrando le loro previsioni attraverso una strategia di majority voting, che permette di aumentare l'efficienza complessiva dei modelli di base.

È stato dimostrato che la strategia suggerita può produrre risultati piuttosto alti, con un punteggio superiore all'86% su ciascun parametro di performance preso in considerazione. Il modello di fusione decisionale supera le performance di ciascuno dei modelli di base, ottenendo i massimi valori medi di accuratezza, AUC e F1-score pari rispettivamente a 88.34%, 88.32% e 86.70%. Inoltre, considerando le performance delle diverse architetture di base, la rete DenseNet-121 supera singolarmente le prestazioni degli altri modelli. Al contempo, il metodo proposto ottiene alti valori di sensibilità e specificità medie, con un punteggio rispettivamente di 88.13% e 90.51%. Dunque, la tecnica basata sulla fusione delle decisioni si traduce in un tasso di falsi positivi inferiore rispetto a qualsiasi singolo modello ed un tasso di falsi positivi molto più basso, migliorando dunque la precisione media rispetto ai modelli separati.

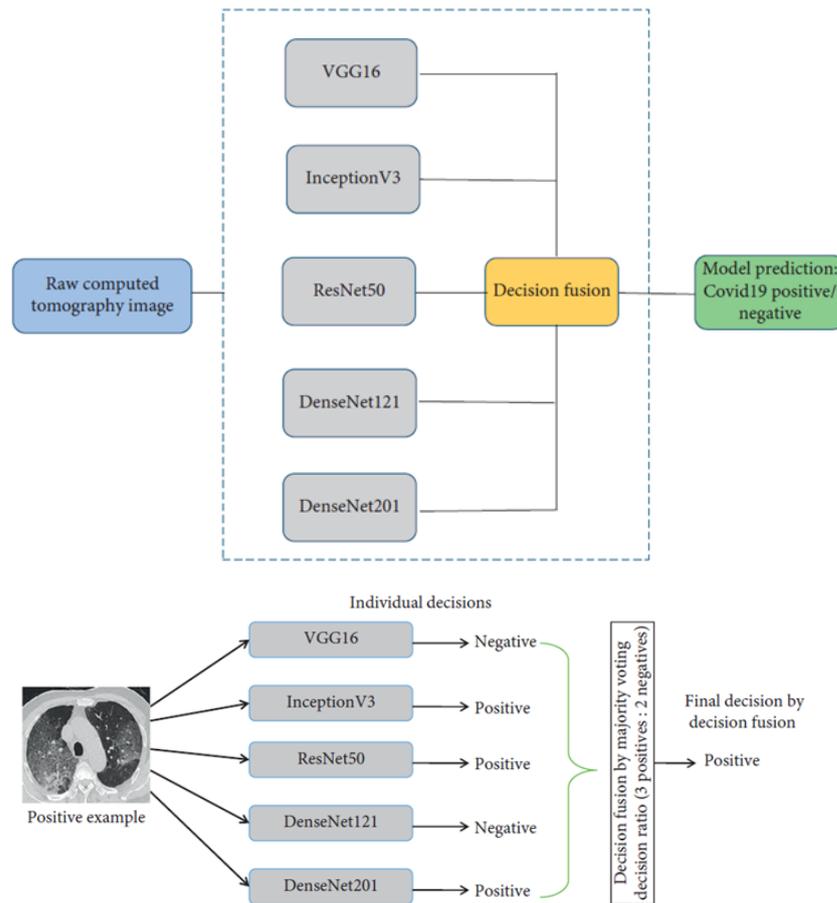


Figura 4.2: Struttura del meccanismo di predizione basato sul decision fusion e relativo esempio proposto nello studio di Mishra et al. [38]

In un ulteriore studio, per addestrare EfficientNet-B4 a distinguere tra casi di positività al Covid-19 e casi di pazienti sani dalle scansioni TC del torace, Anwar et al.[39] hanno utilizzato un learning rate ciclico, un learning rate costante e la riduzione del learning rate al plateau con l'obiettivo di definire il metodo che porta alle migliori performance.

Inizialmente è stato testato un learning rate fisso pari a 0.0001. Oltre ad un learning rate costante si sono esplorati altri metodi che comprendono il learning rate ciclico e il learning rate ridotto al plateau. Nel caso del learning rate ciclico, quest'ultimo oscilla tra un limite superiore e inferiore durante l'allenamento del modello. È stato utilizzato il metodo triangular2, in cui il learning rate diminuisce della metà di quello del triangolo precedente. Nella configurazione di riduzione al plateau, si verifica un decremento del learning rate quando le prestazioni del modello si stabilizzano e raggiungono un plateau. In questo modo è più facile

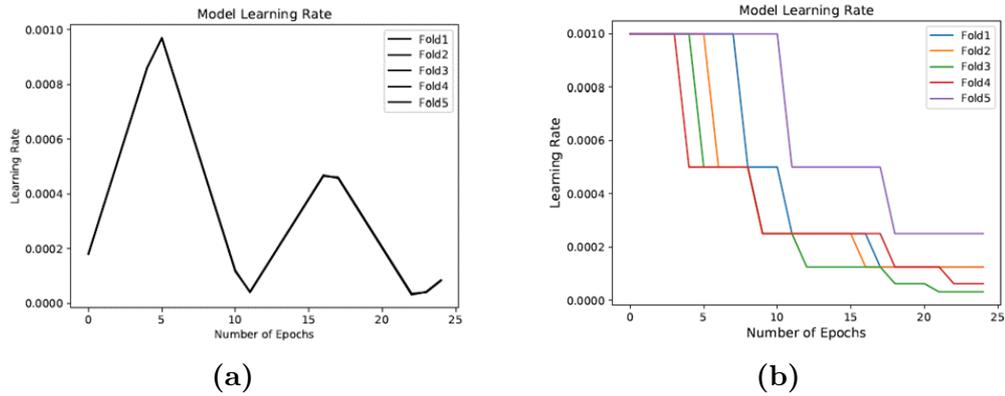


Figura 4.3: Andamenti del: a) learning rate ciclico b) learning rate ridotto sul plateau [39]

muoversi lentamente quando il modello si avvicina alla convergenza. Inoltre, per i tre metodi è stata utilizzata la focal loss, in grado di ridurre l'errore attribuendo alle istanze più difficili più peso e agli esempi più facilmente classificabili meno peso, aumentando la probabilità che gli esempi difficili producano previsioni accurate.

Il metodo del learning rate ciclico fornisce la massima precisione di 0.97 nella classe Covid, mentre la strategia di riduzione al plateau produce la massima precisione di 0.94 nella classe non Covid. Nella classe Covid, la recall più alta è 0.84, ottenuta mediante l'approccio di riduzione al plateau, mentre nella classe non Covid è 0.98, ottenuto utilizzando il learning rate ciclico. L'F1-score per il metodo 'Reduce on Plateau' era 0.9, mentre per Cyclic Learning Rate e Constant Learning Rate erano rispettivamente 0.86 e 0.82.

Nello studio di Nguyen et al.[40] sono stati raccolti tre importanti dataset pubblici: il China Consortium of Chest CT Image Investigation (CC-CCII), il COVID-CTset dall'Iran e il MosMedDat dalla Russia ed a questi è stato aggiunto un ulteriore dataset interno denominato UTSW. Tutti i dati sono stati separati in due classi: pazienti positivi al Covid-19 e pazienti negativi al Covid-19.

In questa indagine, la maggior parte delle scansioni avevano un numero di slices che variavano da paziente a paziente e per tale motivo è stata scelta $512 \times 512 \times 128$ come dimensione dell'input alla rete. Dunque, slices di zero (zero padding) sono state aggiunte alle scansioni TC nei casi in cui il numero totale di slices nelle scansioni del singolo pazienti era inferiore a 128. Per i pazienti con più di 128 slices è stato scelto un volume continuo e randomico di 128 strati.

Sono stati eseguiti processi di clipping e normalizzazione poiché la formattazione dei dati variava tra i diversi dataset a disposizione. Utilizzando le strategie di data augmentation, sono state applicate diverse tecniche di normalizzazione per la fase

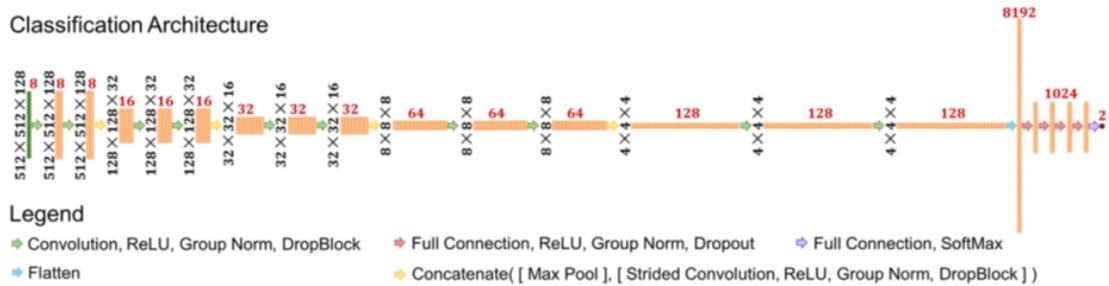


Figura 4.4: Architettura di deep learning proposta da Nguyen et al. [40]

di training della rete. I dati sono stati normalizzati in due modi: 1) da 0 a 1 o 2) da 0 a $\frac{\max(data)}{2^n}$, dove n è l'intero più basso che può essere mantenuto mantenendo 2^n maggiore del valore massimo nel volume TC. Ad ogni iterazione durante la fase di training, il metodo di normalizzazione viene eseguito randomicamente con una probabilità del 50%. L'operazione di data augmentation, se applicata, ha operato il random flip delle immagini (con una probabilità del 50%), la trasposizione ed ha ruotato o ridimensionato le immagini (anche in questo caso, con una probabilità del 50%). L'angolo di rotazione è stato calcolato moltiplicando un numero intero casuale (0, 1, 2 o 3) per 90° .

Utilizzando l'ottimizzatore Adam con un learning rate pari a $1 \cdot 10^{-5}$, ogni modello CNN è stato addestrato per un diverso numero di epoche per ogni dataset. Per evitare l'overfitting sui vari dataset durante l'allenamento, sono state utilizzate, oltre al data augmentation anche tecniche di regolarizzazione chiamate Dropout e DropBlock.

Complessivamente, ciascun modello ha ottenuto i migliori risultati sul dataset su cui è stato addestrato, mostrando un valore di accuracy pari al 97% quando la rete è stata addestrata e testata sui dati CC-CCII. Il modello con l'accuratezza più scarsa sul proprio dataset era COVID-CTset, che aveva un'accuratezza di 86%. Quando i modelli sono stati testati su un dataset sui quali non erano stati addestrati hanno avuto performance nettamente peggiori, indicando una scarsa capacità di generalizzare bene su nuove tipologie di dataset. Il modello basato sul dataset COVID-CTset testato sul dataset UTSW ha avuto le prestazioni peggiori, con una precisione del 38%. Sul dataset MosMedData, nessuno dei tre modelli ha funzionato bene.

Un ulteriore studio proposto da Serte ed al.[41] suggerisce l'utilizzo di una rete convoluzionale (CNN) alla quale vengono poste in input le singole slice di ogni scansione TC 3D. Per ciascuna delle scansioni TC, il modello 2D- CNN definisce le previsioni che vengono quindi combinate per classificare le immagini Covid-19. L'aspetto importante del documento è l'uso della fusione delle previsioni per

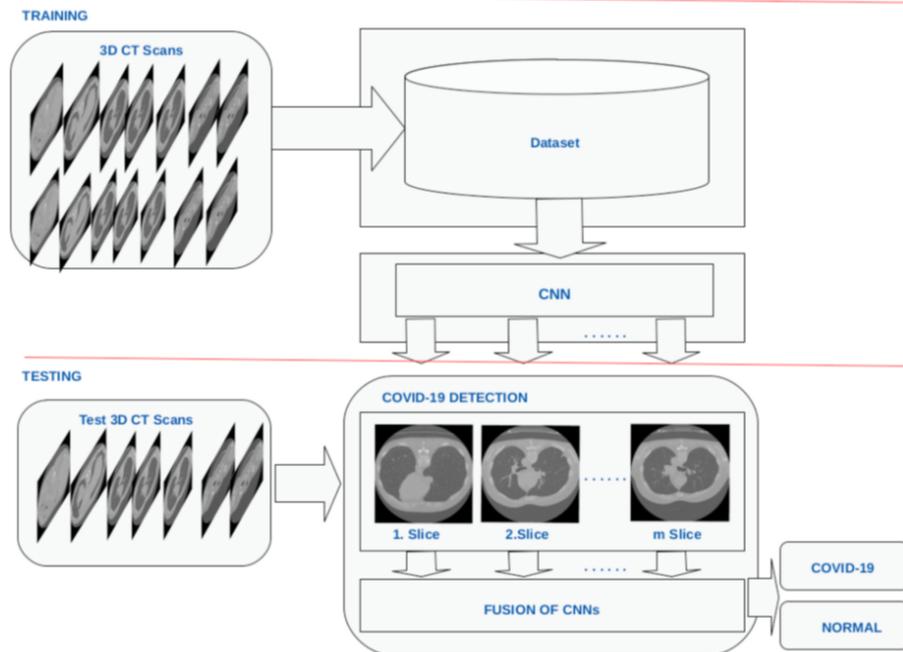


Figura 4.5: Architettura che definisce la presenza di Covid-19 sulla base di scansioni TC proposta da Serte et al. [41]

aggregare i singoli giudizi della 2D-CNN. I risultati mostrano che la combinazione delle previsioni di diverse CNN 2D funziona meglio rispetto all'utilizzo di una singola rete CNN 3D. Il sistema di intelligenza artificiale suggerito utilizza i modelli ResNet-50 e ResNet-18 e il majority voting per stimare il Covid-19 sulla scansione TC 3D di una persona. L'algoritmo ha dunque in output la classe 'Covid-19' e la classe 'sano' come previsioni dopo aver ricevuto una scansione TC 3D come input. Il metodo suggerito divide le slice assiali contenenti polmone in tre categorie: slices di polmone superiore, medio e inferiore. Le slices del polmone superiore e inferiore raffigurano rispettivamente le caratteristiche ossee e una piccola area della regione polmonare. Al contrario, il polmone di un paziente è ben visibile nelle slice centrali della scansione e per questo motivo il sistema utilizza queste sezioni del polmone medio per la fase di training.

Tutte le scansioni TC vengono ridimensionate a 256x256 immagini RGB, che vengono poi inserite nei modelli CNN. Viene successivamente effettuato il tuning dei parametri della rete, la quale è stata precedentemente addestrata sul dataset ImageNet. Le stesse operazioni vengono effettuate sia per l'architettura ResNet-50 sia per la ResNet-18. Ciascuna di queste fette assiali medie viene posta in input al modello di rete neurale convoluzionale. Le previsioni effettuate vengono quindi combinate mediante la tecnica del majority voting per classificare i pazienti come

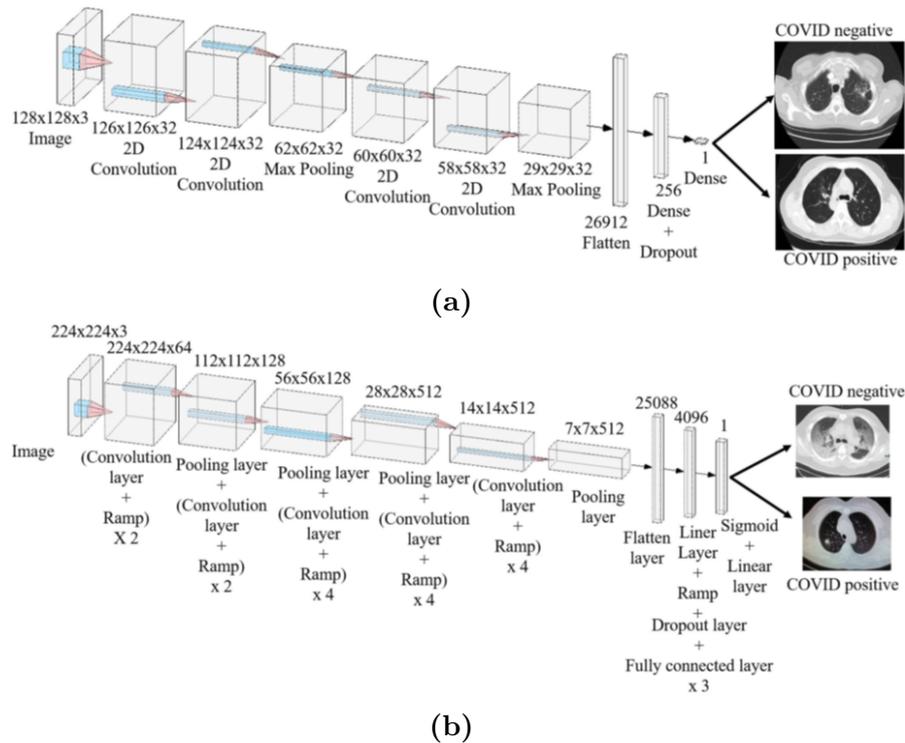


Figura 4.6: Architetture proposte da Shah et al.: a) modello CTnet-10; b) modello VGG-19. Entrambe vengono utilizzate per la classificazione e per effettuare un confronto di performance tra architetture diverse [42]

positivi al Covid-19 o come soggetti sani.

Le performance delle due reti addestrate (ResNet-18 e ResNet-50) sono state testate e confrontate utilizzando anche 2 metodi di decision fusion: SMP e MV che rappresentano rispettivamente il meccanismo decisionale basato sulla massima probabilità (o max voting) ed il meccanismo decisionale basato sul majority voting. Tra le tecniche suggerite, il modello ResNet-50 basato sul majority voting offre i migliori risultati. È stato dimostrato che il modello di deep learning suggerito ha un valore AUC del 96% per il rilevamento di Covid-19 sulle immagini TC.

Ulteriori prove sono state effettuate nello studio di Shah et al.[42] in cui è stato creato un modello con il nome CTnet-10 per la diagnostica Covid-19. DenseNet-169, VGG-16, ResNet-50, InceptionV3 e VGG-19 sono altri modelli esaminati. Con una precisione del 94.52%, il VGG-19 ha superato tutti gli altri modelli di deep learning. Il set di dati Covid-19 CT era costituito da scansioni TC di pazienti i cui risultati dei test Covid-19 erano stati verificati anche mediante RT-PCR. Un'immagine di input è stata inserita nella reti portando ad un risultato che è un

valore compreso tra 0 e 1: meno di 0.5 in questo caso denota positività al Covid-19, mentre maggiore o uguale a 0.5 denota negatività al Covid-19. I layer del modello sono stati sottoposti a operazioni di regolarizzazione utilizzando il dropout con una probabilità di 0.3. È stato definito l'ultimo layer con un singolo neurone e la classificazione binaria finale viene effettuata sulla base della funzione di attivazione sigmoidea. Tutti i modelli sono stati testati con parametri simili tra loro.

Il modello CTnet-10 ha un'accuratezza dell'82.1% che denota comunque una buona capacità da parte di tale architetture nello svolgimento del task. Le altre reti hanno però mostrato performance che risultano essere superiori. In particolare, la rete VGG-16 pre-addestrata ha dimostrato un'accuratezza dell'89% tra scansioni TC infette e non infette da Covid-19. Si potrebbe ottenere una precisione fino al 93.15% tramite il data augmentation ed il fine tuning. L'accuracy per il modello Inception V3 è stata del 53.4% mentre il modello ResNet-50 ha raggiunto un'accuratezza del 60%.

È stata testata anche una rete pre-addestrata VGG-19 con transfer learning per la dimensione ridotta del dataset e ha prodotto un'accuratezza del 91.78%. Un ulteriore incremento dell'accuracy al 94.52% viene ottenuto dopo alcune regolazioni, consentendo di ricavare la migliore performance tra tutti gli altri modelli testati. La seconda migliore accuratezza è stata osservata in una rete DenseNet-169 con un valore pari al 93.15%.

Si può riportare anche lo studio di Perumal et al.[43] nel quale gli autori hanno suggerito, per il rilevamento di Covid-19 da immagini CXR e TC, l'utilizzo del transfer learning e di un modello basato sui descrittori di Haralick che si concentrano esclusivamente sulla regione di interesse. In questo caso, la tecnica del transfer learning può offrire un grande aiuto con la diagnosi in quanto si è riscontrato che le abilità apprese da un modello addestrato a riconoscere la polmonite virale possono essere tradotte nella diagnosi di Covid-19.

Per iniziare, è stato utilizzato il transfer learning sul dataset NIH costituito da radiografie del torace. Al modello vengono fornite immagini TC di diverse malattie polmonari, tra cui Covid-19. Per ottenere immagini di alta qualità, viene prima eseguito il pre-processing dei dati con tecniche di miglioramento dell'immagine che prevedono l'utilizzo di filtri Weiner e l'equalizzazione dell'istogramma rispettivamente per migliorare il contrasto e ridurre il rumore nelle scansioni TC. Le immagini processate vengono utilizzate per estrarre le texture features, tramite i descrittori di Haralick, le quali sono poste in input ai diversi modelli CNN predefiniti. Le reti CNN come ResNet-50, VGG-16 e InceptionV3 vengono utilizzate per estrarre le caratteristiche dell'immagine. È stato anche utilizzato un dropout di 0.2 o 0.3 che accelera l'esecuzione rimuovendo i neuroni il cui contributo alla resa non è così grande. I pesi e i valori di bias vengono inizializzati in modo casuale ed aggiornati utilizzando l'ottimizzatore Adam. L'early stopping viene utilizzato per impedire

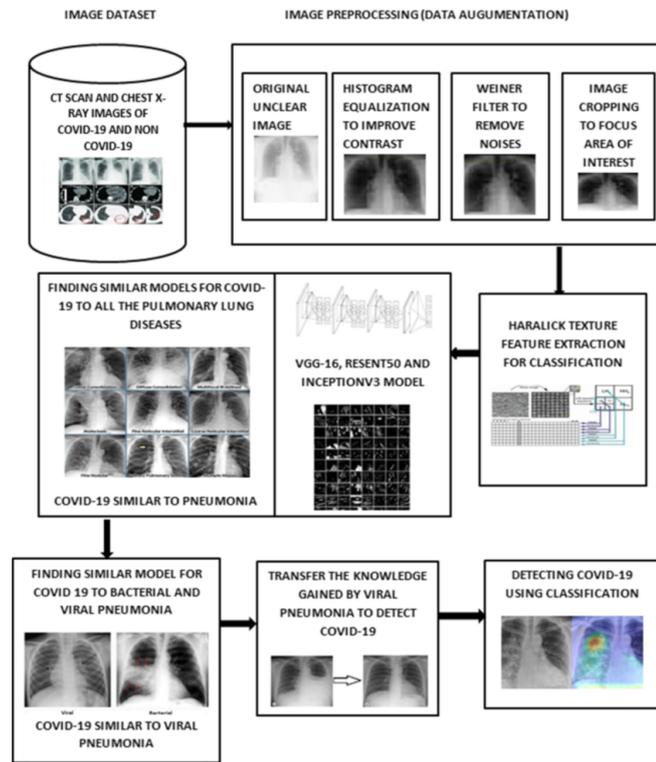


Figura 4.7: Architettura generale del modello CNN proposto da Perumal et al. [43]

l'overfitting dei dati. Poiché la creazione da zero di un nuovo modello richiede tempi di addestramento aggiuntivi, è stato implementato un sistema di transfer learning che utilizza l'architettura VGG-16, la quale presenta le performance migliori. Esistono un totale di quattordici metriche per le texture features, tutte derivate dalla matrice delle co-occorrenze che fornisce dettagli su come l'intensità in un certo pixel in un dato punto si relaziona ai pixel vicini. Le immagini N-dimensionali vengono create utilizzando la Gray-Level Co-occurrence Matrix (GLCM), dove N è il numero di livelli di intensità nell'immagine. In base ai valori dei descrittori è possibile discriminare tra immagini polmonari con presenza di polmonite virale o immagini polmonari infette da Covid-19. In particolare, questo studio porta alla conclusione che i descrittori F4 (varianza), F6 (media cumulativa) ed F7 (varianza cumulativa) dovrebbero rientrare solo in intervalli specifici a seconda della classe del paziente. Altri descrittori mostrano valori con variazioni minori.

Per tutti i modelli con training preliminare, inclusi VGG-16, ResNet-50 e InceptionV3, viene calcolato l'indice di misclassificazione. È stato riscontrato che, rispetto all'apprendimento tradizionale, il transfer learning ha raggiunto una maggiore accuratezza per la somiglianza tra i casi di polmonite ed i casi di Covid-19.

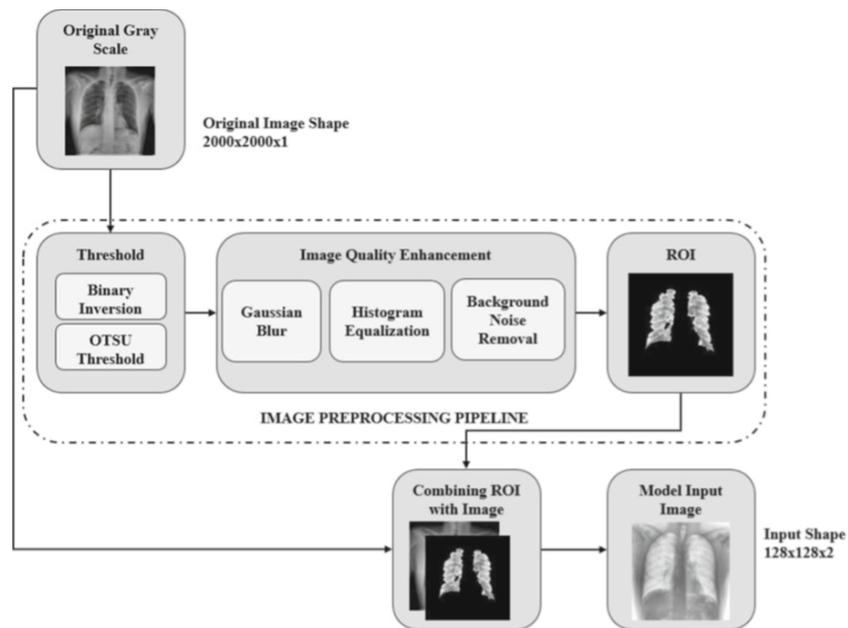


Figura 4.8: Operazioni di pre-processing delle immagini in Rajawat [44]

Con un indice di misclassificazione di 0.012, il modello VGG-16 classifica accuratamente i dati Covid-19. Il modello suggerito produce una precisione del 91%, una recall del 90% e un'accuratezza del 93% tramite VGG-16 utilizzando la tecnica del transfer learning.

Lo studio proposto da Rajawat et al.[44] definisce un modello di deep learning C-COVIDNet basato su una CNN per il rilevamento dell'infezione da Covid-19 ed è stato addestrato su un set di immagini radiografiche del torace suddivise in tre classi: Covid-19, polmonite e paziente sano. Affinché le caratteristiche necessarie siano presenti nell'input, l'area di interesse (ROI, in questo esempio, i polmoni) viene estratta e separata dallo sfondo non informativo, attraverso operazioni di pre-processing delle immagini.

Il modello suggerito è un'architettura CNN e l'immagine ROI viene sovrapposta all'input in scala di grigi originale come secondo canale in modo da essere posta in input al modello con una dimensione 128x128x2. Il volume di input viene normalizzato utilizzando la strategia del min-max scaling. L'intervallo dei valori dei pixel normalizzato è [- 0.5, 0.5]. Per l'eliminazione del rumore di fondo che risultano essere dati non informativi, le immagini vengono divise in regioni connesse e qualsiasi di queste abbia una dimensione pari o inferiore a 500 pixel viene eliminata impostando il valore su zero.

Ogni strato di convoluzione utilizza la dropout function (con una probabilità

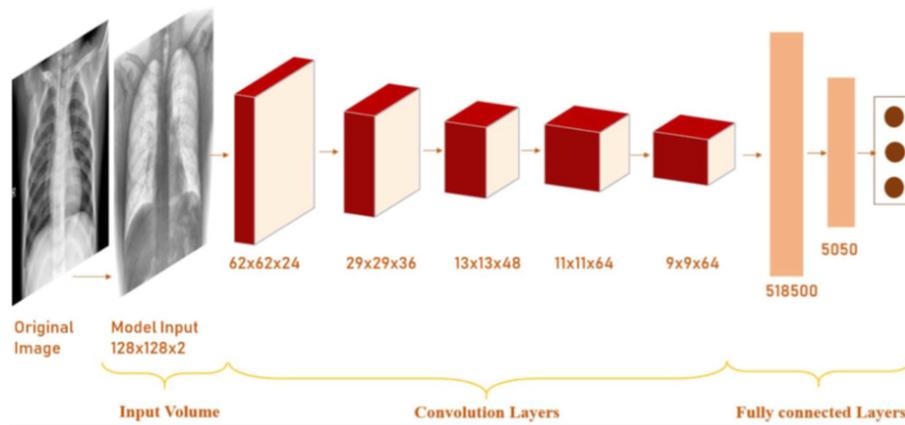


Figura 4.9: Architettura C-COVIDNet proposta da Rajawat et al. [44]

variabile tra 0.2 e 0.1 in base al livello delle convoluzioni) che riduce la possibilità di overfitting e migliora la generalizzazione. Massimizzare il valore del punteggio F1 per robustezza e accuratezza per l'efficacia senza overfitting è il criterio di prestazione del modello. Con la rete in esame viene raggiunta un'accuratezza in fase di training del 97.5% e un F1-score del 97.91%. L'accuratezza sul Test set e sul Validation set è rispettivamente dell'87% e dell'89%.

Capitolo 5

Materiali e metodi

Dopo aver presentato una panoramica delle reti neurali convoluzionali e della loro architettura ed analizzati i paper più influenti presenti in letteratura, vengono ora mostrati i materiali e metodi impiegati per la realizzazione dell'algoritmo. In primo luogo, viene presentato il dataset utilizzato per il training e testing della rete neurale. Segue la presentazione completa dell'architettura del metodo proposto per la classificazione di immagini TC nelle sue diverse fasi:

- Pre-processing
- Training della rete
- Testing e post-processing

Vengono presentate anche le architetture di alcune delle reti addestrate in unione alle metriche impiegate in fase di testing per ottenere i risultati della classificazione.

5.1 STOIC Dataset

Per lo svolgimento del progetto di tesi è stato utilizzato lo STOIC Dataset, il più grande set di dati di immagini di tomografia computerizzata di pazienti COVID-19, raccolto nell'ambito dello STOIC project, che aveva l'obiettivo di creare un dataset ampio e pubblico che potesse essere utilizzato per valutare il valore diagnostico e prognostico della TC nella polmonite da COVID-19.

Il progetto STOIC mirava a costruire un set di dati di almeno 10.000 scansioni TC di individui con sospetta polmonite da COVID-19, valutati durante la prima ondata della pandemia SARS-CoV-2 in Francia, da marzo ad aprile 2020. Il progetto ha coinvolto 20 ospedali universitari: 15 di Assistance Publique des Hôpitaux de Paris e cinque di altre città (Strasburgo, Lione, Rennes e Montpellier). Secondo le raccomandazioni dell'Autorità sanitaria francese, la TC è stata eseguita come

test di screening solo nei pazienti che si presentavano in ospedale con dispnea e/o desaturazione misurata mediante pulsossimetria. Le acquisizioni TC sono state eseguite senza somministrazione di mezzo di contrasto tranne quando si sospettava un'embolia polmonare alla presentazione.

Come standard di riferimento è stato utilizzato il risultato del tampone molecolare o test RT-PCR. Se il primo risultato della RT-PCR era negativo ma si è rivelato positivo nei 7 giorni successivi alla TC, il paziente è stato considerato positivo per l'infezione da SARS-CoV-2. Sulla base di questo dataset è stata indetta una Challenge, denominata STOIC Challenge, il cui fine consiste nello sviluppo di metodi di apprendimento automatico volti ad individuare i pazienti con Covid-19 in forma grave. I partecipanti alla Challenge hanno avuto accesso ai dati del progetto STOIC, recentemente pubblicato su Radiology e rilasciati con licenza CC BY-NC 4.0, per svolgere la fase di addestramento della rete.

Il dataset utilizzato è formato da un numero limitato di pazienti rispetto all'intero STOIC Dataset in quanto, nella fase di qualificazione, il campione di dati open access era costituito solo da 2000 scansioni, che corrisponde a circa il 20% dell'intero dataset. Le scansioni disponibili sono state selezionate casualmente dal database STOIC e sono comprensive delle label cliniche corrispondenti (positività al test RT-PCR e gravità).

Come detto in precedenza, l'obiettivo principale della Challenge STOIC2021 è prevedere dalla scansione TC chi svilupperà COVID-19 grave, definito come intubazione o morte entro un mese dall'acquisizione della TC, utilizzando l'AUC (Area Under the Curve) come metrica principale sulla base della quale si valutano le performance dei classificatori e viene stilata la classifica finale. L'obiettivo secondario è prevedere chi ha avuto una RT-PCR positiva.

Il materiale scaricato comprende due cartelle:

- la cartella *data* all'interno della quale sono presenti le scansioni TC in formato *mha*;
- la cartella *metadata* all'interno della quale vi è il file *reference.csv* in cui sono riportate le labels, per ogni paziente, relative a:
 - **ID paziente** che rappresenta il codice identificativo del paziente;
 - **probCovid** che, in formato binario, indica il risultato del RT-PCR: (1) se c'è positività, (0) se c'è negatività al Covid-19;
 - **probSevere** che, in formato binario, indica se entro un mese il paziente svilupperà (1) o non svilupperà (0) sintomi gravi dovuti al Covid-19.

Come detto, il dataset a disposizione consta di 2000 scansioni TC rilasciate in formato *mha*, estensione associata molto di frequente ai file medicali, delle quali:

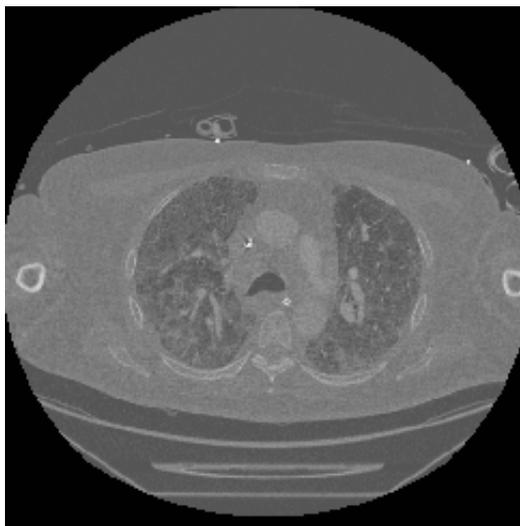


Figura 5.1: Slice di una scansione TC del Dataset STOIC

- 795 TC appartengono a soggetti positivi al Covid-19 al tampone RT-PCR e che non sviluppano sintomi gravi;
- 301 TC appartengono a soggetti positivi al Covid-19 al tampone RT-PCR e che sviluppano sintomi gravi;
- 904 TC appartengono a soggetti negativi al Covid-19 al tampone RT-PCR.

Le dimensioni delle immagini delle scansioni TC ad alta risoluzione sono $N \times 512 \times 512$, dove N indica il numero di slices che caratterizza la scansione e varia per ogni paziente. Inoltre, ogni immagine è codificata su 16 bit con segno. Un esempio di slice appartenente al Dataset STOIC è riportato in Figura 5.1

Inoltre, il dataset contiene le seguenti informazioni, disponibili come intestazioni nei file *mha*:

- Età → il dataset comprende pazienti di età compresa tra i 35 e 85 anni, con una media di circa 60 anni;
- Genere → Maschio o Femmina.

Per poter leggere le immagini e le relative informazioni contenute in ogni file *mha*, è stata utilizzata la libreria Simple ITK, frequentemente utilizzata in molte applicazioni per l'analisi di immagini mediche.

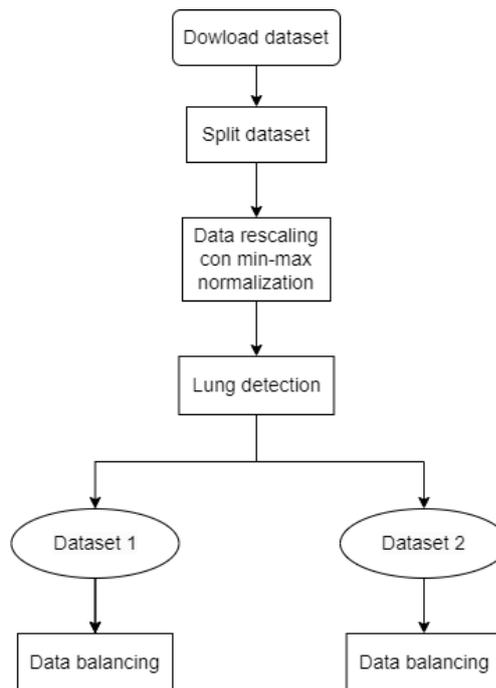


Figura 5.2: Flowchart pre-processing

5.2 Data pre-processing

Un riassunto delle operazioni di pre-processing è riportato nel flowchart in Figura 5.2.

Dopo il download del dataset, si procede alla fase di data pre-processing che consiste in operazioni preliminari effettuate sui dati a disposizione prima che siano dati in input alle reti da addestrare. Il processamento iniziale delle immagini è una parte essenziale della pipeline del progetto in quanto consente di ridurre il rumore indesiderato, enfatizzare specifiche features delle immagini ed ottenere in output immagini più fruibili al modello che si vuole utilizzare, migliorandone la capacità di apprendimento e le performance globali di classificazione. Dopo l’iniziale split del dataset, sono stati implementati altri quattro step di pre-processing. Il primo step consiste nel rescaling delle immagini mediante min-max scaling. Segue l’operazione di data selection per eliminare immagini non significative ai fini del task. Lo step successivo consiste nella creazione di un secondo dataset in cui viene segmentato il polmone. L’ultimo step consiste nel data balancing, utile ad evitare sbilanciamenti dovuti a classi più numerose.

5.2.1 Split dataset

Inizialmente si provvede alla separazione del dataset in tre subsets: Training set, Validation set e Test set. Questi verranno usati rispettivamente per l'addestramento del modello, per effettuare il tuning (cercare la migliore regolazione) degli iperparametri sulle differenti reti provate e l'ultimo per la conferma dei risultati e per l'analisi della performance. Si è scelto di suddividere il dataset in Training (70%), Validation (10%) e Test (20%) set, secondo le percentuali riportate tra parentesi. Queste percentuali sono circa in linea con le percentuali di split del dataset trovate in letteratura ed utilizzate in task simili.

Dunque, al termine di queste fase, si hanno:

- 1400 pazienti nel Training set;
- 200 pazienti nel Validation set;
- 400 pazienti nel Test set.

5.2.2 Data rescaling

Come detto in precedenza, l'output di un dispositivo per TC è un'immagine che associa i toni di grigio alle unità HU, che possono coprire una gamma molto ampia di valori. Tuttavia, quando si osserva un'immagine, il numero di toni di grigio distinguibili e visualizzabili anche sui monitor di alta qualità è molto limitato e quindi, anche avendo a disposizione 2000 o più valori diversi in HU, non sarebbe possibile convertirli in altrettanti toni di grigio. È per tale motivo che vi è la necessità di effettuare il rescaling delle immagini in modo da ridurre la dinamica di valori nell'immagine.

Dunque, il secondo step è rappresentato dal rescaling delle immagini che costituiscono le scansioni TC mediante normalizzazione basata sul min-max scaling. Questa operazione è inoltre fondamentale in quanto consente di lavorare su immagini più omogenee ed avere tutti gli input alla rete nello stesso range di valori, abbattendo al massimo la variabilità che può tradursi in un modello spesso instabile, con scarse performance durante l'addestramento e con una stretta dipendenza dai valori delle immagini in input.

È per tale motivo che il processo di rescaling implementato in fase di pre-processing porta a risultati sicuramente vantaggiosi. In questo progetto, il rescaling opera per avere tutte le slices delle scansioni TC nello stesso range di valori [0:255]. Questa operazione è volta a portare la dinamica su 256 valori per adattare le immagini di input alla dinamica richiesta dalle reti.

Per effettuare un rescaling più raffinato ed avendo un range di valori estremamente variabile tra le scansioni TC, sono state definite 4 soglie in HU. In particolare, oltre a definire gli estremi dell'intervallo di valori che rappresentano l'oggetto di

interesse, sono stati definiti altri limiti per i quali si procede alla saturazione dei pixel (operazione di clipping).

Le 4 soglie impostate sono state definite come segue:

- Due soglie definiscono un intervallo [-1050:2500] tale per cui tutti i valori al di fuori di tale range vengono saturati all'estremo più vicino dell'intervallo stesso. Il limite superiore, in particolare, è importante per limitare il problema di pixel con HU molti alti in pazienti con protesi ossee impiantate (i sistemi protesici possono richiedere valori molto alti di rappresentazione in HU). Questo, infatti, comprometterebbe l'operazione di rescaling dell'immagine che rischierebbe sicuramente di risultare alterata;
- Due soglie limitano il range di valori in HU dell'oggetto di interesse (polmone) all'intervallo [70:1800].

Oltre alle soglie definite in HU, è stato necessario definire le corrispondenti soglie nella dinamica [0:255], determinando il numero di bins da dedicare all'oggetto ed il numero di bins da destinare alle porzioni delle immagini che non sono di interesse.

Dopo l'operazione di clipping, tutti i valori dei pixel dell'immagine vengono portati in un range positivo di valori, sommando a ciascuno di essi il valore assoluto del minimo di ogni immagine. Successivamente si definiscono il minimo (che generalmente coincide con 0) ed il massimo dell'immagine, utili per le operazioni di normalizzazione. Si è scelto di utilizzare la normalizzazione secondo la metodologia del min-max scaling, adattandone l'espressione per i diversi range che si definiscono. Si riporta di seguito la formula matematica generale del min-max scaling, senza alcuna distinzione di range e dove X rappresenta l'immagine da normalizzare.

$$X_{scalato} = \frac{X - \min(X)}{\max(X) - \min(X)} \cdot n_{bit} \quad (5.1)$$

Si è scelto di non ricavare una colormap lineare in seguito alla normalizzazione ma si ottengono pendenze diverse in base a quanto si comprimono i livelli di HU in uint8. In particolare, è stato utilizzato un fattore di compressione che assegna 200 toni di grigio per il range in HU che definisce l'oggetto di interesse ed i rimanenti valori vengono assegnati a ciò che non è di interesse. Di conseguenza, i valori di $\min(x)$ e di $\max(x)$ variano in funzione del range di valori che si analizza. Inoltre, per avere una dinamica su 255 valori, si moltiplica per n_{bit} che definisce il numero di bit scelto per rappresentare ciascun range di valori in HU. Indicando con Min e Max rispettivamente il valore minimo e massimo di ciascuna immagine, sono state effettuate le seguenti operazioni di rescaling delle immagini:

- Nella dinamica [0:255], i bins tra 0 e 13 sono stati assegnati al corrispondente range di valori [Min:70] in HU;

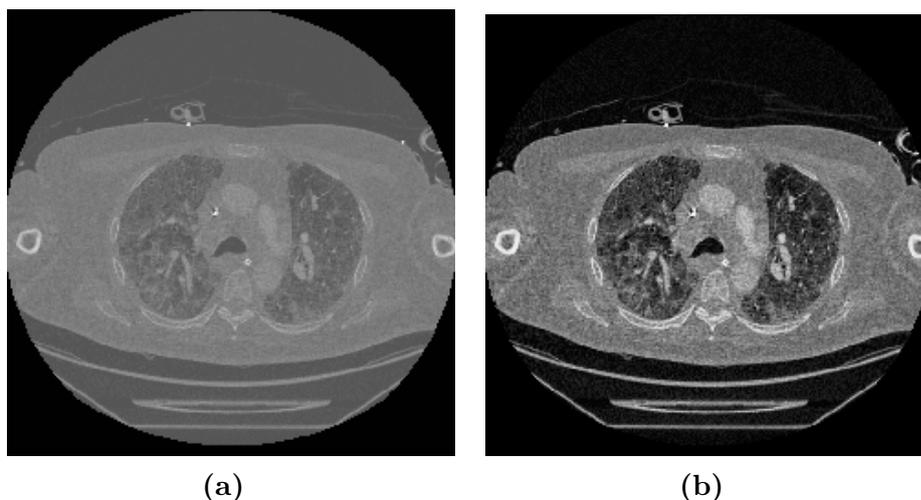


Figura 5.3: Effetto del rescaling sulle immagini della scansione TC

- Il range [14:214] in bins, utilizzato nella rappresentazione del polmone, è stato impiegato per rappresentare il corrispondente range di valori [71:1800] in HU;
- Il range [215:255] in bins che, per la maggiore, viene impiegato per la rappresentazione delle ossa della cassa toracica, è stato utilizzato per rappresentare il corrispondente range di valori [1801:Max] in HU.

L'intera operazione viene ripetuta per l'intero volume 3D della scansione TC.

Al termine dell'operazione di normalizzazione, le immagini risultano come in Figura 5.3b e saranno salvate come file *png*, suddividendole all'interno di cartelle appositamente denominate secondo l'ID del paziente a cui appartengono. Le immagini sono state salvate nel formato Portable Network Graphics (*.png*) per mantenere le caratteristiche omogenee dell'immagine ed avere una compressione del file senza perdite. L'effetto dell'operazione di rescaling sulle scansioni può essere visualizzato in Figura 5.4.

5.2.3 Data selection

Il processo di data selection è uno dei componenti principali della pipeline definita nella fase di progettazione e sviluppo di un modello predittivo in quanto consente di ridurre il numero di variabili di input alle reti di classificazione eliminando i dati ridondanti o irrilevanti, ed aumentando il potere predittivo degli algoritmi di classificazione grazie alla selezione dei soli dati informativi. Inoltre, le operazioni di data selection vengono utilizzate per ridurre il tempo di training della rete in quanto un minor quantitativo di dati comporta lo sviluppo di algoritmi di apprendimento più veloci. Nel progetto di tesi, per il data selection di immagini

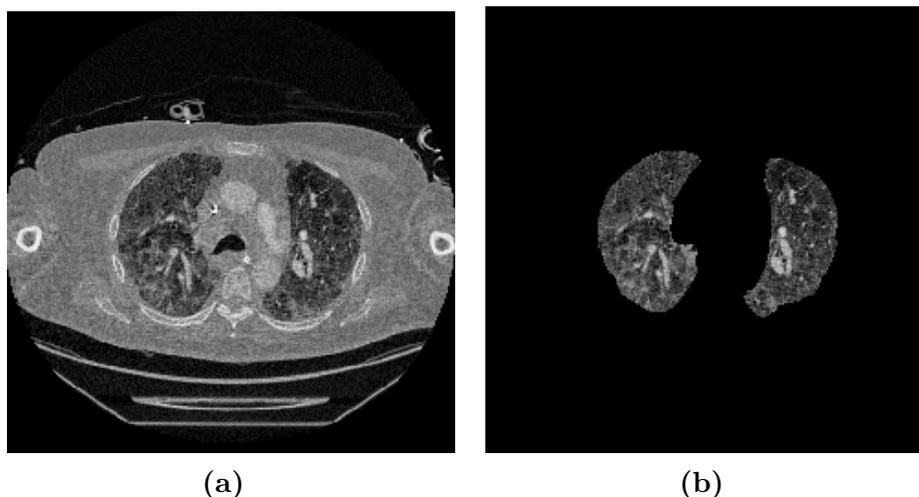


Figura 5.4: Esempi di immagini appartenenti ai due dataset costruiti

informative contenenti i polmoni, è stato implementato innanzitutto un algoritmo di lung detection, il cui flowchart è riportato in Figura 5.5. Tale processo si basa su una detection 3D del volume polmonare che risulta vantaggiosa in quanto consente di lavorare su regioni tridimensionali connesse, aumentando, in definitiva, la robustezza e stabilità della detection stessa. Al termine dell'operazione di lung detection si ottengono due dataset:

1. *Dataset 1* costituito da scansioni TC originali contenenti le immagini informative non segmentate, come riportato in Figura 5.4a;
2. *Dataset 2* costituito da scansioni TC con slices contenenti le segmentazioni dei polmoni. In questo caso tutto ciò che non è polmone (dunque non informativo) viene mascherato e posto a 0. Un esempio è riportato in Figura 5.4b.

La scelta di costruire due dataset nasce dalla volontà di provare due strade diverse, addestrando le reti su ciascuno dei due dataset in modo da determinare su quale fra i due le reti performano meglio. L'algoritmo di lung detection definisce gli indici della prima e dell'ultima slice informativa della scansione TC che devono essere salvate nel dispositivo. Tutte le slices comprese tra questi due indici, secondo l'algoritmo implementato, devono essere ugualmente salvate secondo le stesse modalità utilizzate in fase di salvataggio post-rescaling.

Dato lo spessore non troppo elevato delle slices, si è notato come le fette consecutive presentino caratteristiche molto simili tra loro. Di conseguenza, prima che l'algoritmo di data selection effettui il salvataggio, per ridurre ulteriormente il grande dataset a disposizione ed evitare ridondanza ed un possibile overfitting

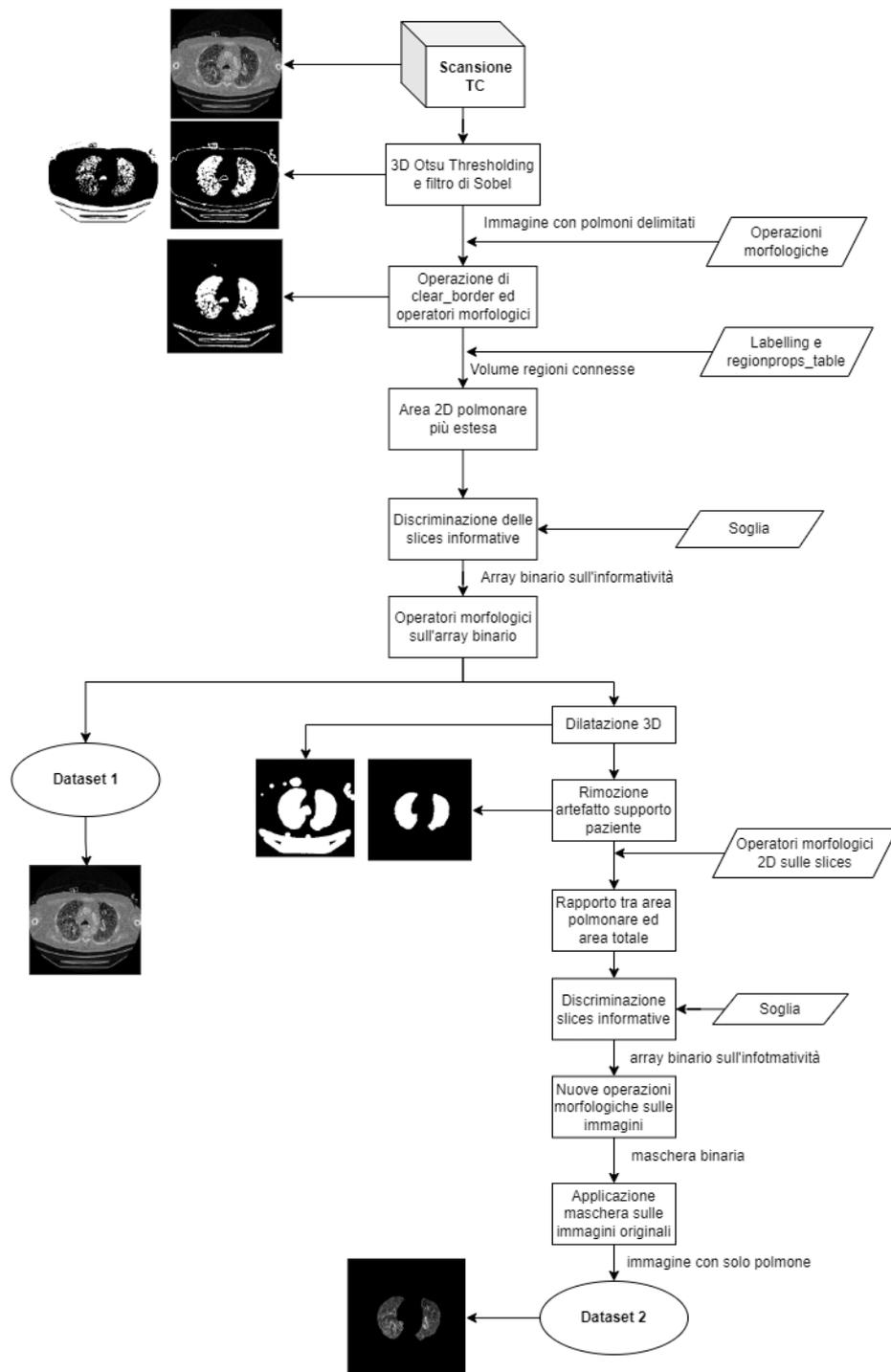


Figura 5.5: Flowchart data selection

della rete, si è scelto di selezionare e salvare solo una slice contenente polmone ogni tre slice informative che ricadono nell'intervallo sopracitato.

Analizzando nello specifico le operazioni compiute dall'algoritmo di data selection, il primo passo consiste nel definire una matrice 3D contenente tutte le slices dell'intera scansione TC del singolo paziente. A questo punto si effettua un'operazione di thresholding automatico, utilizzando la soglia definita mediante il metodo Otsu applicato su tutto il volume della matrice 3D.

La sogliatura di Otsu utilizza tecniche di analisi dell'istogramma delle luminosità ed è particolarmente adatto ad immagini bimodali ovvero ad immagini che presentano istogrammi con una netta separazione tra i due picchi principali di due distribuzioni Gaussiane. Il metodo consiste dunque nell'individuare una soglia automatica definita come il valore assunto dai pixel nell'istogramma delle luminosità tale per cui viene minimizzata la sovrapposizione tra le code delle due distribuzioni Gaussiane.

Al termine dell'operazione di sogliatura, si ottiene una matrice 3D di immagini binarie nelle quali le parti scure delle immagini originali (come i polmoni o la zona extra toracica) vengono rappresentate in bianco e le parti chiare delle immagini originali (come la gabbia toracica, gli organi o gli addensamenti polmonari) vengono rappresentate in nero, come si può vedere nella Figura 5.6a.

Per generare una netta separazione ed un più facile riconoscimento, da parte dell'algoritmo di lung detection, della zona polmonare rispetto alla zona di non interesse, sono stati definiti in grigio i limiti del polmone mediante l'utilizzo del filtro di Sobel (il risultato di tale operazione è mostrato in Figura 5.6b). Quest'ultimo è un operatore utilizzato nell'elaborazione delle immagini digitali per effettuare il riconoscimento dei contorni di oggetti presenti nell'immagine mediante operazioni di convoluzione.

Per mezzo del comando `'clear_border'` della libreria `skimage`, è possibile eliminare gli oggetti direttamente collegati ai bordi esterni del volume della scansione. Nel caso in esame, viene quindi rappresentato in nero lo spazio extra toracico, in modo da evidenziare e far risaltare in bianco solo ed esclusivamente i volumi polmonari, come mostrato in Figura 5.6c, con la possibilità della presenza dell'artefatto dovuto al supporto paziente, di cui si tratterà in seguito.

Successivamente, viene utilizzato il comando `'label'` della libreria `skimage.measure` per etichettare con numerazione progressiva le regioni connesse all'interno del volume, costituite da pixel adiacenti tra loro e con lo stesso valore. Per mezzo del comando `'regionprops_table'`, si calcola il volume delle regioni connesse: da questo si riscontra una grande variabilità nei valori ed anche la presenza di molte regioni costituite da un numero veramente esiguo di pixel.

A questo punto, si individuano le due regioni connesse più estese con i rispettivi due volumi (che generalmente dovrebbero corrispondere ai due volumi polmonari) e tutto ciò che è al di fuori di tale volume viene portato a nero nella matrice 3D in

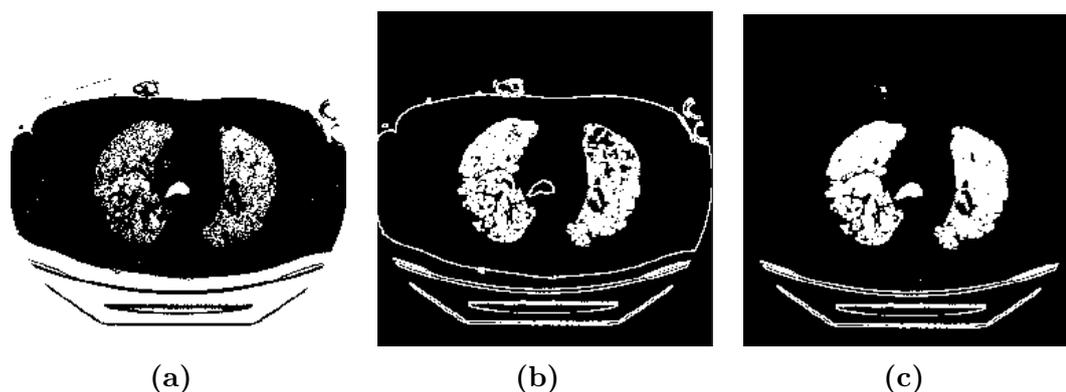


Figura 5.6: 5.6a è il risultato dell'operazione di Otsu thresholding; 5.6b è il risultato dell'utilizzo del filtro di Sobel; 5.6c è il risultato del comando `'clear_border'`

quanto considerato non informativo. Lavorando poi sul 2D, sono state analizzate tutte le singole slices della matrice in modo da determinare l'area polmonare 2D più estesa che sarà usata come area massima (o area-soglia) dall'algoritmo di data selection. Questi passaggi sono stati necessari per personalizzare l'algoritmo di data selection sul singolo paziente poiché definire un valore univoco di superficie polmonare massima sarebbe risultato complesso a causa delle diverse dimensioni delle casse toraciche.

Definita l'area polmonare massima, si analizzano nuovamente tutte le singole slices, calcolando l'area polmonare di ciascuna e confrontandola con l'area-soglia: la slice viene considerata informativa se il rapporto tra le due aree sopracitate supera una soglia prestabilita come iperparametro. Considerando che le slices di pazienti con Covid in forma grave possano avere slices con molti addensamenti che potrebbero limitare dunque l'area polmonare identificata dall'algoritmo, si è scelto di porre, per tale classe, una soglia meno stringente rispetto ai pazienti delle altre due classi. Nello specifico, se l'area polmonare della slice analizzata ha un'estensione maggiore del 30% (in pazienti sani o con Covid non grave) o del 40% (nel caso di pazienti con Covid severo) rispetto all'area polmonare massima, allora quella slice viene considerata informativa e contenente polmone; se ciò non si verifica, la slice viene definita come non informativa. Dopo aver salvato il risultato del confronto di ogni slice all'interno di un array, si otterrà un vettore binario relativo all'informatività di ciascuna slice che costituisce la scansione TC del paziente.

Per definire una sequenza consecutiva di immagini informative, eliminando singole valutazioni di immagini considerate informative o non informative, vengono applicate sull'array binario le operazioni morfologiche di erosione, dilatazione e binary fill holes. Si è scelto un kernel della dilatazione di dimensione leggermente maggiore di quello dell'erosione in modo da considerare un maggior numero di

immagini rispetto a quello previsto dall'algoritmo di data selection. A questo punto è necessario distinguere la fase di salvataggio delle immagini che appartengono al Dataset 1 e le fasi di segmentazione dei polmoni utili a definire le immagini che apparterranno al Dataset 2:

1. Per il salvataggio delle immagini non segmentate, innanzitutto si identificano gli indici della prima e dell'ultima immagine informativa direttamente dall'array binario e si salvano le immagini corrispondenti;
2. Per la segmentazione dei polmoni, prima di tutto viene effettuata una dilatazione tridimensionale sulla matrice 3D binaria delle scansioni con un kernel sferico di raggio 13, in modo da avere un volume polmonare con un'estensione maggiore. Per limitare la presenza dell'artefatto dovuto al supporto paziente dello scanner TC, vengono effettuate alcune prove:
 - Si verifica se vi sia un oggetto tridimensionale interamente contenuto nella metà inferiore del volume della scansione TC;
 - Si verifica se vi sia un oggetto contenuto nella metà inferiore delle immagini le cui variazioni di area sono abbastanza limitate tra una slice e la successiva (variazione entro massimo il 15%);
 - Si verifica se vi sia un oggetto nella slice che abbia un'eccentricità superiore al 95% ovvero sia un oggetto stretto e lungo.

Se tutte le verifiche precedenti hanno dato un esito positivo, il volume individuato consiste nel supporto paziente e viene eliminato da ciascuna immagine che lo presenta. Un esempio di operazione di rimozione dell'artefatto del supporto paziente è riportato in Figura 5.7

Si procede poi con un'analisi 2D delle slices per definire le immagini segmentate che devono essere salvate; vengono dunque ripetute le operazioni di erosione e dilatazione mediante due kernel circolari e si utilizza nuovamente il comando *'label'* sulle slice prive dell'artefatto del supporto paziente, al fine di etichettare nuovamente con numerazione progressiva le regioni connesse all'interno di ciascuna immagine. Tramite il comando *'regionprops_table'* applicato alle immagini, vengono calcolate le aree delle regioni connesse e si procede a identificare e sommare le due aree più estese ottenendo, infine, l'area polmonare. A questo punto si calcola il rapporto tra l'area polmonare della slice e l'area totale dell'immagine, calcolata come il numero totale di pixel che la costituiscono (definita come 512x512). Vengono dunque ripetuti i passaggi per tutte le slice che costituiscono la scansione, salvando in un array i risultati del rapporto tra le due aree.

Come effettuato in precedenza, si confrontano i valori dell'array con una soglia predefinita: se l'area polmonare della singola slice presenta un'estensione

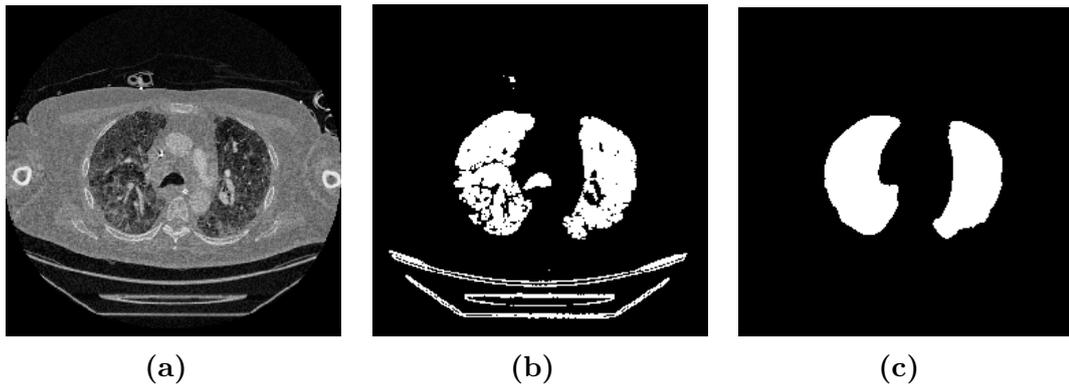


Figura 5.7: 5.7a è l'immagine originale con la presenza dell'artefatto del supporto paziente; 5.7b è la maschera binaria con la presenza dell'artefatto; 5.7c è il risultato della rimozione dell'artefatto del supporto paziente dalla maschera binaria

maggiore del 2.5% dell'area totale, allora quella slice con il polmone segmentato viene considerata informativa; se ciò non si verifica, la slice viene definita come non informativa.

I valori delle soglie utilizzate finora sono stati tutti determinati tramite numerose prove effettuate su diversi pazienti del dataset, in modo da poterle definire nel modo migliore.

Dall'analisi di ogni slice si salva il risultato del confronto all'interno di un array binario utile per il salvataggio delle immagini del Dataset 2. Anche in questo caso vengono effettuate sull'array operazioni morfologiche di erosione, dilatazione e binary fill holes in modo da definire una sequenza consecutiva di immagini informative. In uscita, quindi, si otterrà un array binario che indica l'informatività delle immagini segmentate. Si procede alla fase di salvataggio esclusivamente delle immagini che rientrano nei limiti imposti nel salvataggio delle immagini del Dataset 1 e solo nel caso in cui nell'array binario vi sia almeno una slice informativa nell'intervallo sopracitato.

Si utilizza l'immagine binaria su cui si è lavorato come maschera di segmentazione binaria che permette, applicata alla slice originale, di ottenere in output un'immagine con il solo polmone segmentato e si procede al salvataggio della stessa all'interno del Dataset 2. Qualora tutte le maschere fossero prive di polmone dopo le operazioni applicate, il paziente non fa parte del Dataset 2 al fine di evitare il salvataggio di immagini totalmente nere e prive di informatività.

Come detto in precedenza, per limitare la ridondanza e il possibile overfitting della rete, è stato deciso anche di non salvare tutte le immagini comprese tra questi

indici ma di estrarre e salvare esclusivamente un'immagine ogni tre, per ambo i dataset costruiti.

5.2.4 Data balancing

Per evitare problemi di sbilanciamento fra le tre classi ed i pazienti oggetto dell'analisi, l'ultima operazione di pre-processing consiste nel data balancing. Questa operazione viene eseguita al fine di uniformare il numero di immagini per paziente e per classe poiché, al termine dello script di data selection, il numero di immagini varia da paziente a paziente, creando un bias che può portare la rete ad apprendere meglio alcuni pazienti o alcune classi. Questa operazione deve essere eseguita per i pazienti del Training set e del Validation set di ambo i dataset (Dataset 1 e Dataset 2), definendo lo stesso numero di immagini per paziente per entrambi i subsets. Le operazioni di data balancing eseguite sui due dataset sono divise tra loro e completamente indipendenti, avendo per taluni soggetti un numero di immagini salvate diverso tra i due dataset.

Inizialmente, è stata calcolata la mediana del numero di slices dei pazienti per ciascuna delle 3 diverse classi che si analizzano e si definisce un valore soglia che coincide con la massima mediana tra quelle calcolate. A questo punto, per ogni singolo paziente, si possono verificare due casi:

- Numero di slices \geq soglia \rightarrow random delete di immagini in modo che il paziente abbia un numero di slice pari alla mediana;
- Numero di slices $<$ soglia \rightarrow data augmentation mediante flip orizzontale di alcune slices stabilite in modo casuale. Se per alcuni pazienti il numero delle immagini è minore della metà della mediana, si applica il flip orizzontale di tutte le immagini e non viene applicata alcun'altra trasformazione. Dunque, quel paziente avrà un numero di slices minore rispetto alla maggior parte degli altri pazienti.

Terminata l'operazione di data balancing sui pazienti, si è riscontrato uno sbilanciamento nel numero di immagini totali ottenute per ciascuna classe e, in particolare, un minor numero di immagini per la classe 'Covid severo'. Questo risulterebbe essere un problema in fase di training della rete in quanto quest'ultima tenderebbe ad addestrarsi meglio solo sulle classi più rappresentate e ad avere performance minori a livello complessivo. È per tale motivo che si è preferito procedere ad un bilanciamento, per quanto possibile, anche nel numero di immagini per ciascuna classe. È stato dunque definito il numero di immagini che contiene la classe più rappresentata; sulle immagini di tale classe non è stata applicata alcuna trasformazione. Nel caso delle classi meno rappresentate, è stato calcolato il numero di immagini da aggiungere per ciascun paziente per bilanciare le classi ed eseguito

il flip orizzontale di quelle immagini per le quali non era stata già effettuata questa operazione in precedenza dalla prima parte del data balancing.

In questo modo è stato ottenuto un dataset che risulta essere abbastanza bilanciato sia sul numero di slices per paziente sia sul numero di immagini per ogni classe.

5.3 Training delle reti CNN

Una parte fondamentale del progetto di tesi consiste nell'addestramento di reti neurali convoluzionali (CNN) di classificazione. Per fare ciò, il transfer learning ha dimostrato di essere il metodo più efficace per svolgere il task di classificazione. Il training delle tre reti ha lo scopo di analizzare le slice della scansione TC polmonare, elaborate in fase di pre-processing e fornite come input per classificarle in base alla positività al Covid-19 ed alla sua relativa severità.

In questo lavoro sono state addestrate e testate, separatamente su ambo i dataset, dieci diverse architetture di CNN, più o meno complesse, con l'obiettivo di identificare il modello che meglio performasse nel task di classificazione di pazienti sani e patologici: ResNet50, ResNet101, Wide-ResNet50, csp-ResNet50, csp-ResNeXt50, csp-DarkNet50, SEResNet50, HRNet, DenseNet121 e Res2Net50.

Le reti di classificazione sono state addestrate in Spyder di Anaconda, un ambiente di sviluppo integrato (IDE) open source per la programmazione scientifica in linguaggio Python.

Il dataset per la classificazione (costituito da Training set e Validation set) prevede un numero di slide che varia a seconda del dataset che si sceglie di porre in input alla rete. In particolare, come già specificato in precedenza, sono stati costruiti due dataset per l'addestramento della rete con slices di dimensione 512x512: il Dataset 1 comprende 128777 slice del Training set appartenenti a 1399 pazienti, 17630 immagini del Validation set appartenenti a 200 pazienti e 29333 immagini del Test set provenienti da 400 pazienti; il Dataset 2 comprende 128503 slice del Training set appartenenti a 1395 pazienti, 17341 immagini del Validation set provenienti da 198 pazienti e 29231 immagini del Test set appartenenti a 400 pazienti.

Come si può evincere, non solo cambiano le caratteristiche delle immagini ma varia anche il numero di slice ed il numero di pazienti per ogni set di dati, avendo eliminato, nel Dataset 2, le immagini completamente nere o col solo supporto paziente (operazione non effettuata per la costruzione del Dataset 1).

L'allenamento è supervisionato, ovvero le immagini con le rispettive label vengono fornite in input alla rete con la seguente codifica:

- classe 0 : paziente positivo al Covid-19 in forma grave;
- classe 1 : paziente positivo al Covid-19 in forma leggera;

- classe 2: paziente sano.

Per valutare la capacità di apprendimento della rete è stata riportata, ad ogni epoca, l'accuracy calcolata sulle immagini del Validation set e la loss calcolata sul Training set.

Si è deciso di sfruttare le architetture disponibili della libreria *MMClassification*, in seguito all'elevata tendenza all'overfitting delle reti disponibili della libreria torchvision.

MMClassification è un framework open source per classificazione delle immagini basato su PyTorch ed offre una vasta gamma di architetture di reti neurali. Tra le caratteristiche principali che hanno fatto propendere verso l'utilizzo di questa libreria, vi sono: disponibilità di vari backbone e modelli pre-addestrati, configurazioni di allenamenti su larga scala, l'elevata efficienza ed estensibilità e la disponibilità di toolkit potenti [45].

5.3.1 Architettura delle reti

Come scritto in precedenza, in questo progetto sono state addestrate e testate, su ciascuno dei due dataset costruiti, dieci diverse architetture di reti neurali convoluzionali tra le tante disponibili nel framework *MMClassification*. Per rendere maggiormente fluida l'analisi, saranno presentate, di seguito, esclusivamente le reti con le performance migliori o le reti che sono utili a definire il sistema di classificazione più adatto al task da svolgere.

ResNet

La ResNet, abbreviazione di Residual Network, è una tipologia di rete neurale convoluzionale ampiamente utilizzata nell'ambito della classificazione di immagini e pubblicata nel 2015 all'interno del documento "Deep Residual Learning for Image Recognition" di He Kaiming et al.[46].

Per migliorare l'addestramento delle reti di deep learning riducendo l'overfitting e diminuire il tempo richiesto per l'addestramento, è stato introdotto dal team di esperti precedentemente citati, un sistema di apprendimento residuo [47]. La ResNet è stata sviluppata per tentare di risolvere il problema del vanishing gradient che si verifica in reti con un numero elevato di layer. Con la retropropagazione del gradiente ai precedenti layer, le ripetute moltiplicazioni possono dar luogo a gradienti infinitamente piccoli che diminuiscono l'apprendimento delle features da parte dei layer iniziali. Questo fenomeno porta ad una saturazione o ad un rapido peggioramento delle performance quanto più la rete va in profondità [48].

Con l'architettura ResNet si introduce anche il concetto di skip connections sul quale esse si basano e che consente l'aggiunta di più layer convoluzionali ad una CNN senza incorrere nel problema del vanishing gradient [46]. L'insieme dei

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figura 5.8: Architettura delle reti ResNet [47]

layer che contengono skip connections è anche definito blocco residuo [38]. Questa soluzione prevede l'aggiunta di più connessioni tra blocchi distanti che permettono di propagare l'input saltando alcuni layer.

L'approccio presentato consente l'addestramento di reti con elevata profondità senza comprometterne le prestazioni. Per tale motivo, questa rete è diventata una delle architetture più utilizzate in letteratura negli ultimi anni. Ci sono diverse architetture ResNet che si differenziano per il numero di strati e per le skip connections. Il numero di layer varia da 18 a 152.

In questo progetto sono state implementate la ResNet50 e la ResNet101. Come specificato dal nome, la prima contiene 50 layer (48 livelli convoluzionali, un livello MaxPool e un average pooling layer seguito da un fully connected layer che permette di classificare l'immagine in ingresso in una delle classi attraverso l'applicazione della funzione softmax) mentre la seconda contiene 101 layer (100 strati convoluzionali ed un fully connected layer per generare, come nella ResNet50, la classificazione finale dell'immagine) [46].

In questo progetto vengono modificati i fully connected layer, generalmente costituiti da 1000 neuroni in seguito all'addestramento sul dataset ImageNet, in modo da ottenere un nuovo layer di classificazione costituito da un numero di neuroni pari al numero di classi del task da svolgere (in questo caso pari a tre). La dimensione tipica dell'input di queste reti è di $224 \times 224 \times 3$. L'architettura di queste reti è stata illustrata in figura [49].

Wide ResNet

Le Wide Residual Network o W-ResNet sono una variante ed un miglioramento

group name	output size	block type = $B(3, 3)$
conv1	32×32	$[3 \times 3, 16]$
conv2	32×32	$\begin{bmatrix} 3 \times 3, 16 \times k \\ 3 \times 3, 16 \times k \end{bmatrix} \times N$
conv3	16×16	$\begin{bmatrix} 3 \times 3, 32 \times k \\ 3 \times 3, 32 \times k \end{bmatrix} \times N$
conv4	8×8	$\begin{bmatrix} 3 \times 3, 64 \times k \\ 3 \times 3, 64 \times k \end{bmatrix} \times N$
avg-pool	1×1	$[8 \times 8]$

Figura 5.9: Architettura delle reti Wide ResNet[47]

delle ResNet standard. Queste architetture sono state introdotte nel documento ‘Wide Residual Networks’, pubblicato nel 2016, ed aggiornate nel 2017 da Nikos Komodakis e Sergej Zagoruyko.

Sono state sviluppate con l’intento di migliorare le prestazioni e la rapidità del processo di addestramento tramite una minore profondità dell’architettura ed un aumento dell’ampiezza della rete. In questo modo, i modelli che presentano tale architettura convergono verso la soluzione ottimale molto più rapidamente delle ResNet standard o di altre reti più profonde [47].

HRNet

Le High Resolution Network (HRNet) sono CNN tipicamente utilizzate per task di segmentazione, di detection di oggetti o per task di classificazione delle immagini. Come illustrato nella Figura 5.10, la rete presenta un primo flusso di convoluzioni ad alta risoluzione (branch 1) e si aggiungono gradualmente, uno per volta, altre sottoreti (o flussi) convoluzionali ad alta o bassa risoluzione al fine di formare più branch. In questo modo, tali reti sono costituite dal collegamento in parallelo di più sottoreti a risoluzione multipla [50]. Queste reti hanno la caratteristica di sostituire il tradizionale metodo seriale con il metodo parallelo di sottoreti con diverse risoluzioni con l’obiettivo di migliorare le performance. Inoltre, tali architetture sfruttano la tecnica della fusione multiscala ripetuta per garantire lo scambio di

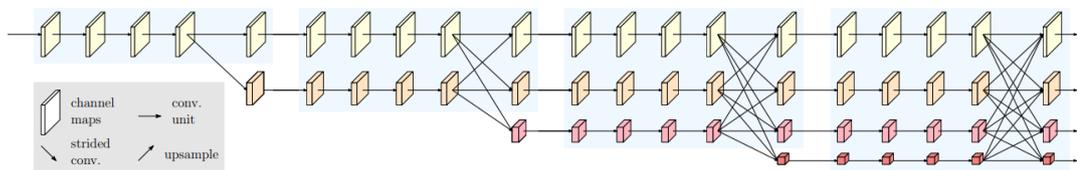


Figura 5.10: Architettura delle reti HRNet [50]

informazioni tra le sottoreti e migliorare così l'accuratezza delle previsioni [51].

5.3.2 Parametri ed iperparametri delle reti

Lo sviluppo e l'addestramento delle reti neurali sono ancora processi imprevedibili e impegnativi, soprattutto per l'utilizzo di diversi iperparametri che influenzano direttamente i risultati dell'addestramento e delle performance finali dei modelli di deep learning.

I parametri sono i coefficienti del modello e vengono definiti direttamente dal modello stesso che, durante l'apprendimento, li ottimizza al fine di minimizzare l'errore (secondo un metodo di ottimizzazione). La sola attività che si deve svolgere con i parametri consiste nella loro inizializzazione. Un esempio di parametri interni del modello sono i pesi dei neuroni.

A differenza dei parametri, gli iperparametri sono variabili che il modello non aggiornerà seguendo la strategia di ottimizzazione ma la loro modifica richiederà l'intervento dell'operatore. Dunque sono valori da impostare prima del processo di apprendimento e definiscono la struttura della rete e le modalità di addestramento della rete [34].

Nel machine learning e deep learning, l'ottimizzazione degli iperparametri consiste nella selezione di un insieme ottimale di iperparametri per l'algoritmo di apprendimento. I parametri ed iperparametri utilizzati per il training delle architetture proposte in questo progetto sono descritti in Tabella 5.1:

Learning rate	5e-4
Batch size	32
Numero di epoche	70
Funzione di perdita	Cross entropy loss function
Metodo di ottimizzazione	Adam
Weight_decay	0.0001
Data augmentation	random resize crop, random horizontal flip, CutMix e MixUp
Metrica di valutazione	Accuracy

Tabella 5.1: Parametri ed iperparametri di training delle reti

Di seguito si esaminano alcuni degli iperparametri (ad es. il learning rate, il batch size o il numero di epoche) fornendo una loro breve descrizione e definendo i valori imposti per il processo di addestramento. Inoltre vengono presentati alcuni metodi che permettono di rendere l'algoritmo più performante come, ad esempio,

la loss function, l'optimization function, il data augmentation o la normalizzazione dei dati.

Learning Rate

Nello sviluppo di reti neurali di classificazione, il learning rate o tasso di apprendimento è uno degli iperparametri più importanti da ottimizzare per il miglioramento delle performance. Come già riportato in precedenza, esso assume un ruolo fondamentale nella fase di backpropagation ed in particolare nell'aggiornamento dei parametri secondo una funzione di ottimizzazione. Se la direzione verso il minimo della funzione di costo può essere definita calcolando il gradiente della loss function, il learning rate definisce la dimensione del passo, ad ogni iterazione, in quella direzione, indicando l'impatto che il gradiente ha sulla definizione del peso e controllando anche la velocità di apprendimento del modello [52].

Un valore basso del tasso di apprendimento consente un lento apprendimento del modello, con il rischio, tuttavia, di far convergere anche troppo lentamente il modello incorrendo nel cosiddetto "Vanishing Gradient"; al contrario, un valore troppo elevato consente un apprendimento più veloce, ma, se il peso viene aggiornato troppo rapidamente, potrebbe superare e mancare il punto minimo facendo divergere il modello ed incorrendo nel problema del cosiddetto "Exploding Gradient" [34].

Generalmente, il tasso di apprendimento assume valori compresi tra 0 e 1, in funzione dell'architettura della rete neurale e del dataset utilizzato per l'addestramento. Sebbene sia possibile una riduzione del learning rate durante il processo di training, è stato scelto un andamento (espresso dal parametro *policy learning rate*) fisso e costante durante tutte le epoche dell'addestramento pari al valore iniziale, definito initial learning rate, di 0.0005.

Batch Size

Il batch size rappresenta il numero di campioni elaborati prima che il modello ed i suoi parametri vengano aggiornati [53]. Potrebbe risultare inefficiente e controproducente per l'addestramento alimentare la rete neurale con tutte le immagini del dataset ed è per tale motivo che occorre suddividere i dati in campioni più piccoli chiamati batch, consentendo un minor consumo di memoria ed un addestramento più rapido. Se si scegliesse un batch size troppo grande vi sarebbe il rischio che il modello si generalizzi troppo e, perciò, si adatti male ai nuovi dati. Un valore piccolo del batch comporta, invece, una stima del gradiente meno accurata [34]. Nelle architetture proposte si è scelto di suddividere il dataset in batch di dimensioni pari a 32.

Numero di epoche

Il numero di epoche risulta essere un altro iperparametro dalla grande importanza in quanto definisce il numero di volte in cui l'algoritmo elabora l'intero dataset utilizzato per l'addestramento. Il numero di epoche dipende dalla tipologia

dei dati e dal task da svolgere [34]. Generalmente, gli algoritmi di machine learning e deep learning migliorano le proprie performance con l'aumento del numero di epoche. Tuttavia, in caso di overfitting, un addestramento con un numero di epoche molto grande non è efficiente in quanto, dopo un certo punto, la rete tende a migliorare le performance solo sul Training set e non più sul Validation set.

In questo progetto si addestrano le reti per 70 epoche e, ad ogni epoca, vengono calcolate la loss e l'accuracy e salvate nel log file, in unione alla configurazione dei pesi della rete associati a quell'epoca.

Loss function

La loss function, o funzione di perdita, rappresenta un altro elemento estremamente importante per l'addestramento delle reti neurali, poiché, insieme alle funzioni di ottimizzazione, svolge un ruolo diretto nell'adattamento ai dati in input da parte del modello. Durante l'addestramento si tenta di ridurre al minimo la loss, ovvero la differenza tra gli output previsti e quelli target, per generare un modello migliore. Le funzioni di perdita maggiormente utilizzate in task di classificazione sono la binary cross entropy loss function e category cross entropy loss function [54]. Entrambe si basano sul concetto di entropia, una variabile correlata alla casualità delle probabilità di output durante il processo di apprendimento [55].

La cross entropy loss function viene generalmente utilizzata per la regolazione dei pesi dei neuroni del modello durante l'allenamento. Tale funzione calcola la differenza tra la distribuzione di probabilità prevista dal modello di classificazione e la distribuzione effettiva della variabile analizzata. La binary cross entropy loss function viene utilizzata in task con solo due possibili classi, mentre la categorical cross entropy loss function viene generalmente utilizzata in task con tre o più classi. Sebbene esistano diverse tipologie di loss function da poter implementare, in questo progetto è stata utilizzata la cross-entropy loss function, definita matematicamente dalla seguente espressione:

$$L_{CE} = - \sum_{i=1}^n t_i \cdot \log(p_i) \quad (5.2)$$

dove:

- t_i è la truth label;
- p_i è la probabilità di output predetta dalla rete che l'input appartenga alla i -esima classe;
- n rappresenta il numero di classi (nel caso in esame, è pari a 3)

Algoritmo di ottimizzazione

Gli ottimizzatori sono algoritmi utilizzati per minimizzare la funzione di costo o per massimizzare le prestazioni del modello. In particolare, si tratta di funzioni matematiche che si basano sui parametri apprendibili del modello come ad esempio pesi e bias, al fine di ridurre le perdite [56]. Per tale scopo, la scelta dell'ottimizzatore più giusto è uno degli elementi chiave nella progettazione delle reti neurali.

Oltre alla tecnica del Gradient Descent già descritta in precedenza nel Paragrafo 3.5.2, esistono molte varianti di ottimizzatori ma si è impiegato l'ottimizzatore maggiormente utilizzato in letteratura, l'Adaptive Moment Estimation, noto anche come ottimizzatore di Adam, un derivato dell'algoritmo di discesa del gradiente.

Questo ottimizzatore si è dimostrato più efficiente rispetto ad altri ottimizzatori come lo Stochastic Gradient descent o RMSProp, soprattutto quando si lavora con una grossa mole di dati e con un'architettura di rete neurale densa [57]. È un algoritmo che si basa sulla normalizzazione del gradiente rispetto alla radice della media quadratica del gradiente stesso. Inoltre, memorizza sia la media decrescente dei gradienti passati, come effettuato dal momentum, sia la media quadratica decrescente dei gradienti passati, come gli algoritmi RMSProp e Adadelta. Pertanto, combina i vantaggi di questi metodi.

L'Adam Optimizer consente anche di impostare vari iperparametri e personalizzare perciò la rete neurale. Inoltre, presenta altri vantaggi quali la facilità di implementazione, l'efficienza computazionale ed una limitata richiesta di memoria [56]. Per l'implementazione dell'ottimizzatore Adam sono previsti quattro parametri di cui effettuare il tuning per ottenere i valori ottimizzati, tra i quali vi sono il learning rate, la variabile beta, la variabile epsilon ed il decadimento dei pesi (weight decay).

Si descrivono ciascuno di essi (ad eccezione del learning rate, di cui si è già scritto in precedenza), definendo anche i valori che sono stati attribuiti loro:

- *Beta*: è costituito da due valori β_1 e β_2 . Nello specifico, β_1 , o stima del primo momento, è il tasso di decadimento esponenziale per il termine del momentum. Il suo valore predefinito in PyTorch è 0.9. β_2 , chiamato anche stima del secondo momento, è il tasso di decadimento esponenziale per il termine di velocità. Il valore predefinito è 0,999 nell'implementazione PyTorch;
- *Epsilon*: sebbene non sia un iperparametro da ottimizzare, è un numero molto piccolo per evitare errori di divisione per zero nell'implementazione. Il valore predefinito è 1e-08 [57].
- *Weight decay* a cui, di seguito, viene dedicata una descrizione più accurata.

Decadimento dei pesi (weight decay)

Il decadimento del peso rientra nei metodi di regolarizzazione ed è per tale motivo che il suo valore deve essere impostato accuratamente. Generalmente, questo parametro permette una migliore generalizzazione da parte del modello, una minore complessità del modello ed una riduzione dell'overfitting. Il decadimento del peso è definito da un fattore λ costante, compreso tra 0 e 1, che viene moltiplicato, ad ogni epoca, per ciascun peso durante l'algoritmo di Gradient descent. Dataset ed architetture di minor dimensione normalmente richiedono valori maggiori di λ mentre dataset più grandi ed architetture più profonde richiedono valori minori [58]. Per questo lavoro si è scelto di utilizzare un valore di weight decay pari a 0.0001.

Data augmentation e train pipeline

Il data augmentation, come già descritto in precedenza, è una strategia per limitare l'overfitting e aumentare la capacità di generalizzazione della rete. Consiste in un insieme di trasformazioni delle immagini in input (tra cui rotazione, flip, variazione del contrasto o della luminosità, distorsioni, zoom, etc.) che consentono di aumentare la dimensione del dataset durante l'addestramento della rete. Già nella fase di pre-processing è stato svolto il data augmentation; in particolare, un flip orizzontale è stato applicato su alcune slices per garantire un bilanciamento delle immagini appartenenti a ciascuna delle tre classi.

Prima di ogni iterazione, durante l'addestramento del modello, ogni campione di immagini viene ulteriormente trasformato, seguendo le operazioni definite in modo sequenziale dalla train pipeline, caratteristica delle reti addestrate con il framework *MMClassification*. Si è scelto dunque di seguire lo stesso iter di operazioni, con gli stessi parametri e le stesse trasformazioni, ad eccezione della normalizzazione che si è deciso di personalizzare. Segue una lista ordinata delle operazioni svolte nella pipeline di training:

1. **Random Resize Crop:** utilizzato per ritagliare una regione casuale dell'immagine e ridimensionare l'immagine ad una dimensione specificata (in questo caso, 224, misura necessaria per l'allenamento delle reti disponibili di *MMClassification*);
2. **Random Horizontal Flip:** trasformazione che capovolge orizzontalmente l'immagine con una data probabilità (nel caso in esame è stato scelto un valore di probabilità pari a 0.5);
3. **Normalizzazione e standardizzazione:** le immagini sono state normalizzate in modo che fossero convertite tutte nella stessa scala, all'interno dell'intervallo [0:1]. Inoltre, si è scelto di standardizzare i dati in modo che abbiano una

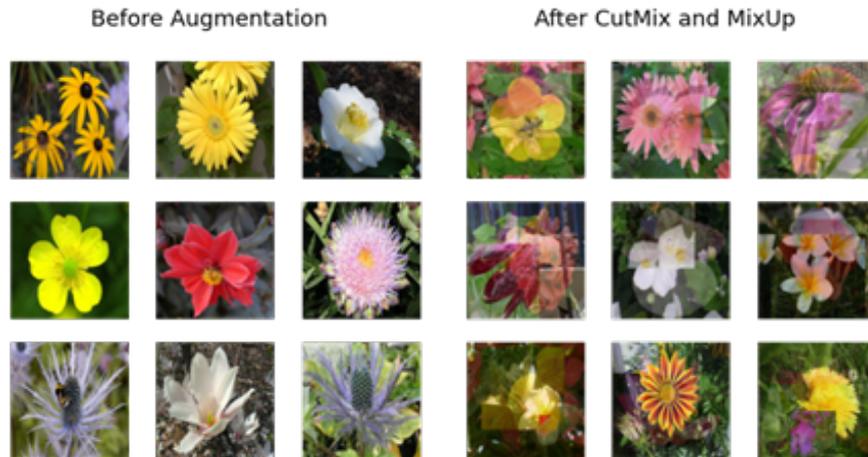


Figura 5.11: Effetto delle operazioni di CutMix e MixUp su immagini generiche [59]

distribuzione normale con media uguale a 0 e deviazione standard uguale a 1, su ognuno dei 3 canali RGB [34];

4. **Image To Tensor:** trasformazione utilizzata per porre in input alla rete dei tensori.

CutMix e MixUp

Le operazioni di CutMix e MixUp rientrano nelle strategie di data augmentation poiché vengono utilizzate per generare esempi diversi tra le classi. Il CutMix ritaglia in modo casuale parti di un'immagine e le incolla su altre immagini di training, impedendo ai modelli di adattarsi eccessivamente a specifiche caratteristiche delle immagini per eseguire la classificazione. Il MixUp interpola i valori dei pixel tra due immagini ovvero genera una nuova immagine mediante interpolazione lineare ponderata di due immagini esistenti [59]. Entrambe le operazioni permettono di migliorare la robustezza del modello e le sue proprietà di generalizzazione ad esempi fuori dalla normale distribuzione [60]. Si riportano in Figura 5.11 esempi di applicazione delle operazioni di CutMix e MixUp su delle immagini.

5.3.3 Salvataggio del modello addestrato

Come già descritto in precedenza, nel progetto di tesi le reti sono state addestrate per 70 epoche e, per ciascuna di esse, vengono calcolate la loss e l'accuracy dell'addestramento eseguito e salvate nel log file, in unione ai corrispondenti pesi della

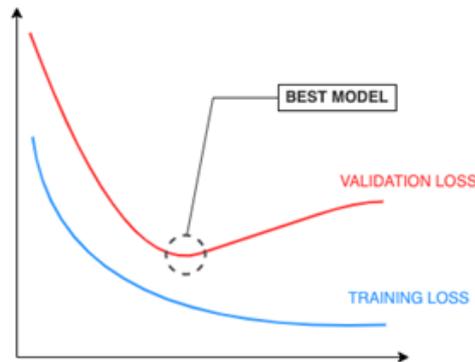


Figura 5.12: Andamento loss function sul Training set e sul Validation set durante l'allenamento della rete [61]

rete relativi a tale epoca. Ad ogni epoca si genera un nuovo modello caratterizzato da una certa performance e da specifiche configurazioni degli iperparametri e parametri.

Normalmente la loss, durante l'addestramento e la convalida, si prevede che diminuisca. Tuttavia, la versione con la 'best accuracy', ovvero con le migliori performance (indicata come miglior modello) raramente coincide con la versione ottenuta all'ultima epoca ovvero al termine del processo di training della rete.

Infatti, in un tipico caso di overfitting come quello riportato in Figura 5.12, inizialmente la Training loss e la Validation loss diminuiscono durante l'addestramento ma poi, da un certo punto in avanti, la Validation loss inizia ad aumentare nonostante la Training loss continui a diminuire. Dopo ciò, è meno probabile che le versioni successive siano in grado di generalizzare bene i dati a causa dell'overfitting sui dati del Training set. In questo caso, il modello finale che fornisce le performance migliori coincide con la versione ottenuta nel punto in cui la Validation loss ha iniziato a divergere. Quindi, vengono salvati i pesi ed i valori di accuracy nel log file in modo tale che, al termine del processo di apprendimento, si proceda all'identificazione, direttamente dal log file, dell'epoca con l'accuratezza migliore. A questo punto, i pesi del modello associati all'epoca in corrispondenza della quale sono state ottenute le prestazioni migliori saranno caricati nel modello finale che verrà salvato su disco, pronto per essere utilizzato nella successiva fase di testing [61].

5.4 Fase di testing e post-processing

In fase di testing vengono effettuate nuovamente le operazioni di rescaling e di data selection delle immagini per verificare la coerenza delle immagini che vengono

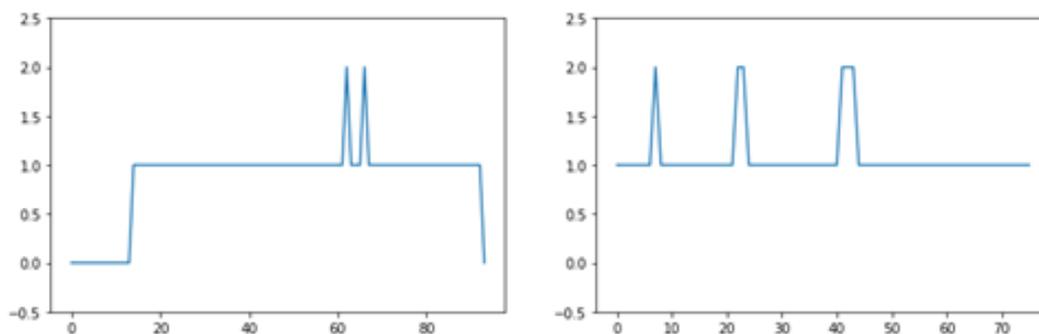


Figura 5.13: Andamenti delle predizioni di un classificatore

fornite in input al classificatore addestrato. Queste operazioni vengono svolte non solo sul Training set e sul Validation set, come avvenuto in fase di pre-processing, ma anche sul Test set. In questo modo, anche per il Test set vengono ricavate le immagini contenenti il solo polmone che verranno fornite in input, in fase di testing, al classificatore addestrato sul Dataset 2. Dunque, dopo aver caricato il modello addestrato, si procede alla fase di inference sui 3 subsets, dalla quale si ottiene in output, per ogni paziente, un array di predizioni che indicano la classificazione assegnata sulle singole slices da parte del modello. Vengono mostrate in Figura 5.13 due esempi dell’andamento predittivo sulle slices di due pazienti, che riportano in ascissa le slices ed in ordinata le classi predette dal classificatore per ciascuna slice. Ottenute le predizioni su ogni slice della scansione TC del singolo paziente, si procede alla fase di post-processing, utile ad assegnare la classe finale del paziente. Per fare ciò, sono state vagliate diverse strategie di aggregazione dei risultati, alcune delle quali presenti in letteratura.

1. **Strategia 1:** consiste nel majority voting (o hard voting), il quale assegna al paziente la classe più frequentemente prevista dalla rete di classificazione. Questa strategia è implementabile tramite il calcolo della moda dell’array di predizioni relativo al paziente. Questo approccio è il più diffuso in letteratura per aggregare le valutazioni effettuate da reti CNN 2D. È stato gestito anche il caso in cui ci fossero due classi con lo stesso numero di predizioni, attribuendo al paziente la classe con una gravità maggiore tra quelle predette in modo da porsi nella situazione peggiore;
2. **Approccio 2:** è stata assegnata al paziente la classe della sequenza consecutiva di slices più lunga con la stessa predizione;
3. **Approccio 3:** prevede di considerare sani esclusivamente i pazienti per i quali il 60% delle predizioni risultavano negative al Covid-19. In caso contrario, si utilizza l’approccio del majority voting tra le sole due classi rimaste (‘Covid

non grave' e 'Covid grave'), assegnando perciò al paziente la classe patologica più presente all'interno dell'array delle predizioni. È stato scelto di effettuare il controllo solo per le predizioni negative poiché si preferisce avere una maggior certezza che il paziente sia effettivamente sano: classificare negativo un paziente positivo al Covid ha un maggiore peso rispetto al caso opposto, sia in termini di diffusione dell'infezione che in termini di salute del paziente;

4. **Strategia 4:** questa strategia prevede dapprima una media delle softmax. In particolare, si utilizzano le percentuali di appartenenza a ciascuna classe contenute nel layer di output della rete. Queste vengono quindi concatenate al fine di creare un array di softmax di dimensione $N \times 3$, dove N rappresenta il numero di slices che costituiscono la scansione del paziente e 3 indica il numero di classi. L'array viene dunque mediato, ottenendo un vettore unidimensionale di lunghezza 3 che rappresenta la softmax media. A questo punto si procede ad assegnare al paziente la classe con lo score più alto;
5. **Approccio 5:** consiste nel max voting [62], una tecnica generalmente utilizzata per i problemi di classificazione che rappresenta uno dei modi più semplici per combinare le previsioni. Questa strategia utilizza la softmax, precedentemente presentata, di dimensione $N \times 3$ ed assegna al paziente la classe con lo score (o voto) più alto.

5.4.1 Ensemble di più reti e dei loro output

Definite ed addestrate le reti con le rispettive architetture, è stato testato un ensemble (ovvero una combinazione) di più reti e dei loro output per generare la previsione finale, tentando di ottenere performance superiori rispetto alle singole reti. Per la combinazione degli output dei classificatori si è scelto di testare un algoritmo, il cui schema viene riportato in Figura 5.14, che sfrutta le softmax ottenute per ogni paziente da ciascuna rete per definirne la classe di appartenenza.

In particolare, viene svolta inizialmente la fase di inference per ogni paziente, durante la quale si applica ciascun modello sui pazienti di Train, Validation e Test set e per ognuno di essi si genera una lista contenenti tutte le softmax finali. La dimensione finale di tale lista di softmax è $\text{num_slices} \times 3$, dove num_slices indica il numero di immagini per il singolo paziente e la seconda dimensione coincide col numero di classi delle quali si riporta la probabilità. La lista è poi stata salvata come file Pickle nel dispositivo in modo da poterla utilizzare per la successiva fase di fusione degli output.

Per la combinazione degli output, sono stati tentati approcci a 2 reti, 3 reti e 4 reti per analizzare quale configurazione desse i migliori risultati. Essendo l'AUC la metrica utilizzata dalla Challenge per stilare la classifica finale, il confronto dei risultati è stato effettuato facendo riferimento proprio a tale metrica.

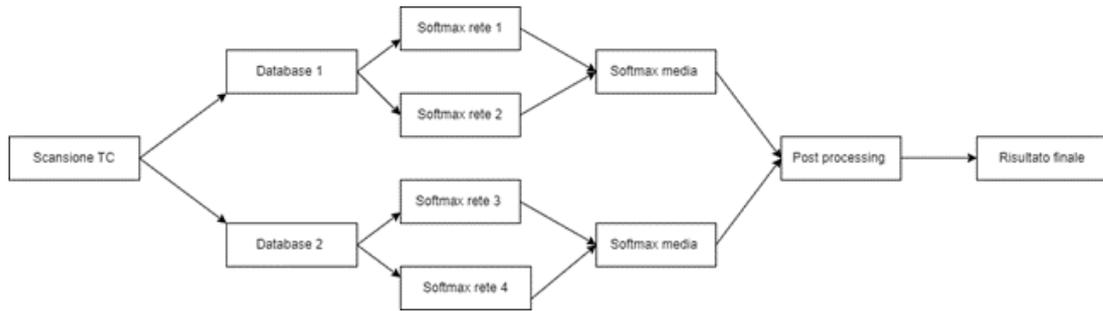


Figura 5.14: Schema generale dell’algoritmo di fusione degli output dei classificatori

Il metodo utilizza la combinazione a due reti su ambo i dataset e prevede di calcolare, per ciascun dataset, la media delle softmax di due reti (approccio noto come Model Average Ensemble), ottenendo una softmax media pari alla dimensione delle softmax mediate. Sono state concatenate dunque le softmax medie risultanti in un unico array e, per ogni riga dell’array, che coincide con le probabilità medie di appartenenza della singola slice a ciascuna classe, viene identificata la classe con la probabilità maggiore e si crea un vettore contenente le predizioni per ogni slice. Questo approccio sui due dataset consente di: ottenere le performance migliori, combinare i risultati sui due dataset e sfruttare il metodo di post-processing migliore.

5.5 Metodi di valutazione

La valutazione del modello di classificazione è una parte fondamentale del progetto di tesi. In particolare, i risultati del metodo esposto nel progetto vengono confrontati con le truth label, ovvero con i risultati dei test RT-PCR. Nei seguenti paragrafi verranno, dunque, esaminate diverse metriche di validazione utilizzate generalmente per valutare quantitativamente le performance e la capacità di generalizzazione dell’algoritmo di classificazione proposto.

5.5.1 Confusion Matrix

Nello sviluppo delle reti di deep learning per i task di classificazione, la confusion matrix è un layout di tabella che può essere utilizzato per visualizzare le prestazioni del modello addestrato. È una tabella di contingenza con due dimensioni (“True label” e “Predicted label”) e label di classi identiche in ambo le dimensioni. Ogni sua colonna rappresenta il numero di istanze predette in una classe mentre ogni riga rappresenta il numero di istanze reali in una classe [63].

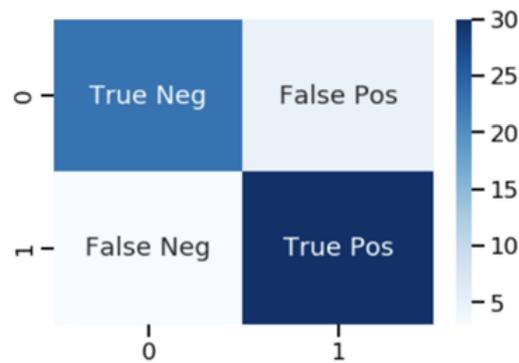


Figura 5.15: Confusion Matrix per classificazione binaria [64]

La matrice ha dimensioni $N \times N$ dove N rappresenta il numero di classi del dataset che in questo progetto è pari a tre: paziente Covid grave, paziente Covid non grave e paziente sano. I corretti classificati sono disposti lungo la diagonale principale della confusion matrix, mentre i pazienti classificati in modo errato dall'algoritmo (anche denominati misclassificati) sono posti nelle altre celle.

La classificazione binaria (Positivo/Negativo) è il caso più comune in letteratura e per la quale si richiede una confusion matrix di dimensione 2×2 , come mostrato in Figura 5.15.

Per la classificazione binaria vi sono quattro potenziali risultati:

- *True Positive (TP)* indica che il modello ha previsto un risultato positivo e l'istanza reale era positiva;
- *False Positive (FP)* indica che il modello ha previsto un risultato positivo, ma il risultato reale era negativo;
- *False Negative (FN)* indica che il modello ha previsto un risultato negativo, ma l'istanza reale era positiva;
- *True Negative (TN)* indica che il modello ha previsto un risultato negativo e l'istanza reale era negativa.

5.5.2 Metriche di validazione

Dalle confusion matrix è possibile ricavare alcune metriche di valutazione delle prestazioni dei modelli di classificazione addestrati. Tra le metriche disponibili, le più comuni sono:

- *Accuratezza:* è la metrica più semplice ed è calcolata come rapporto tra campioni previsti correttamente e campioni totali [44]. Spesso si utilizza

l'accuratezza per misurare la capacità di classificazione del modello, ma non è sufficiente per una valutazione complessiva del modello. Pertanto, anche altre metriche dovrebbero essere esaminate per valutarne l'efficacia. Si calcola come:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

- *Precisione*: è data dal rapporto tra i veri positivi ed il numero totale di positivi previsti dal modello di classificazione [65]. Matematicamente, può essere scritta come:

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

- *Recall* (o *Sensibilità* o *True Positive Rate*): quantifica il numero di predizioni positive corrette sul numero di istanze positive reali [65]. La formula matematica può essere espressa come:

$$Recall/Sensibilità/TPR = \frac{TP}{TP+FN}$$

- *F1 score*: è la media armonica tra precisione e recall e si attesta nel range di valori tra 0 e 1. Questa metrica è importante in quanto consente di capire quanto il classificatore sia preciso (quante istanze classifica correttamente) e quanto sia robusto [66]. Matematicamente, può essere espresso come:

$$F1score = \frac{2 \cdot (precisione \cdot recall)}{precisione + recall}$$

- *Specificità* (o *True negative rate*): corrisponde al rapporto tra le predizioni negative correttamente classificate sul totale delle istanze negative reali. Si può esprimere matematicamente come:

$$Specificità/TNR = \frac{TN}{TN+FP}$$

- *False Positive Rate*: è il rapporto tra le predizioni positive classificate in modo errato sul numero di istanze negative reali. Si esprime come:

$$FPR = \frac{FP}{TN+FP}$$

Nella classificazione multiclasse, come in questo lavoro, tutte le metriche vengono calcolate come in binario con la differenza che devono essere calcolate per ogni classe. Per rendere più semplice la successiva trattazione delle definizioni e delle metriche, si riporta in Figura 5.16 un esempio di confusion matrix a 3 classi, in cui P sta per paziente, G indica Covid in forma grave, S indica paziente sano, C indica covid non severo. Le due lettere che seguono la P indicano, in ordine, la classe predetta e la classe reale.

A partire da queste confusion matrix, le definizioni di vero positivo, vero negativo, falso positivo e falso negativo e le relative metriche sono le stesse della classificazione binaria ma vengono definite singolarmente per ciascuna classe:

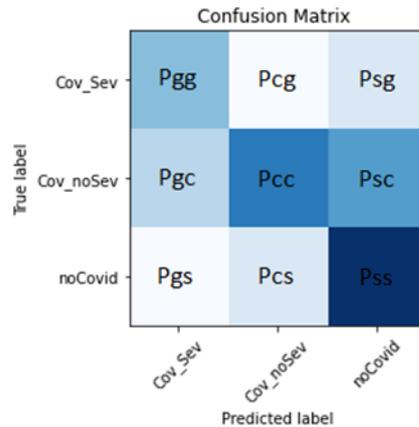


Figura 5.16: Confusion Matrix a 3 classi

- I *True Positive* relativi ad una classe indicano il numero di previsioni in cui i dati appartenenti ad una determinata classe sono correttamente classificati ed assegnati a tale classe [67]. Ad esempio, per la classe dei pazienti sani, i True Positive rappresentano il numero di previsioni in cui un paziente sano è correttamente classificato come ‘noCovid’. Quindi:

$$VP(\text{noCovid})=Pss$$

- I *True Negative* per una particolare classe vengono calcolati sommando i valori in ogni riga e colonna eccetto la riga e la colonna della classe per la quale si vogliono definire il numero di Veri Negativi [67]. Ad esempio, per la classe dei pazienti sani si omettono le righe e colonne della classe ‘noCovid’ e si sommano tutti gli altri valori. Pertanto, in questo caso specifico, i veri negativi corrispondono alla somma:

$$TN(\text{noCovid})=Pcc+Pgg+Pcg+Pgc$$

- I *False Positive* per una classe possono essere calcolati sommando tutti i valori nella colonna corrispondente alla specifica classe, escluso il valore dei True Positive [67]. Nel caso si voglia ricavare il numero di False Positive per la classe ‘noCovid’, si calcola la seguente somma:

$$FP(\text{noCovid})=Psg+Psc$$

- I *False Negative* per una particolare classe vengono definiti come la somma di tutti i valori nella riga corrispondente alla specifica classe, escluso il valore dei True Positive [67]. Pertanto, i Falsi Negativi per la classe ‘noCovid’ sono calcolati con la seguente somma:

$$FN(\text{noCovid})= Pgs+Pcs$$

Se nei problemi di classificazione binaria si otteneva un singolo valore, in task di classificazione multi-classe le metriche vengono calcolate per le singole classi, ottenendo quindi, per il progetto in esame, tre valori per ogni metrica. Se si volesse effettuare una valutazione complessiva dell'intero algoritmo di classificazione con una metrica globale, esistono tre tecniche di averaging implementabili in casi di classificazione multi-classe che vengono utilizzate anche in questa tesi per una valutazione completa delle performance del classificatore [68]:

- *Micro-average*: consiste nel rapporto tra la somma degli elementi posti lungo la diagonale principale della confusion matrix e la somma di tutte le celle della matrice. È quindi l'analogo dell'accuratezza e per tale motivo raramente utilizzata;
- *Macro-average*: è una semplice media aritmetica della metrica calcolata sulle tre classi. Questa tecnica assegna pesi uguali a tutte le classi, rendendola una buona opzione per attività di classificazione bilanciate;
- *Weighted-average*: tiene conto dello squilibrio di classe calcolando la media delle metriche binarie ponderate per il numero di campioni di ciascuna classe reale.

5.5.3 ROC Curve e AUC

Una receiver operating characteristic curve, o curva ROC, è un grafico che rappresenta le prestazioni diagnostiche di un sistema di classificazione binario al variare della soglia di discriminazione tra le classi. Il metodo nasce a partire dal 1941 per l'applicazione nel campo dei ricevitori radar militari (da cui il nome) e da quel momento lo si è utilizzato in campo medico, radiologico, biometrico, nella previsione dei disastri naturali ed anche nella valutazione delle performance dei modelli di apprendimento automatico.

L'analisi ROC fornisce strumenti per individuare i modelli con un alta capacità di classificazione e per scartare le reti non ottimali. Essa viene costruita tracciando il true positive rate (TPR), anche noto come sensibilità o recall, rispetto al false positive rate (FPR), calcolabile anche come (1-specificità), a varie impostazioni di soglia. Ogni risultato di previsione o istanza della confusion matrix rappresenta un punto nello spazio ROC.

In Figura 5.17, vengono riportati quattro esempi di curve ROC. La curva ROC che coincide con la diagonale, nota come linea di non discriminazione, descrive il comportamento di un modello che classifica gli elementi in modo casuale. Questa linea definisce il limite di performance accettabili in quanto i punti al di sopra della diagonale rappresentano dei buoni risultati in termini di classificazione (come nei casi riportati della seconda e terza curva ROC) mentre al di sotto si hanno risultati peggiori della totale casualità (come riportato nell'ultimo esempio di curva).

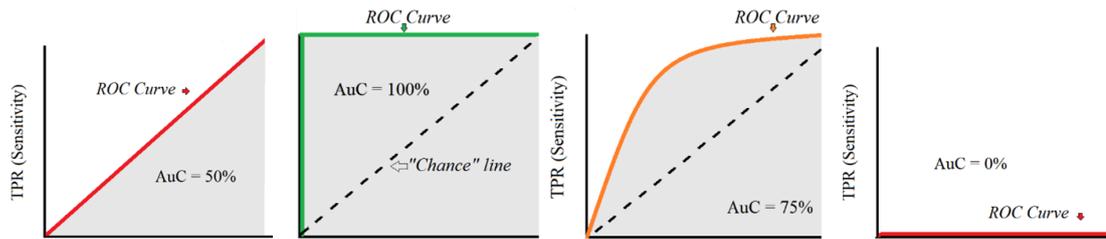


Figura 5.17: Esempi di curve ROC [70]

Un classificatore ottimale che svolge una classificazione perfetta (rappresentato dalla seconda curva ROC riportata) genera risultati posti nell'angolo in alto a sinistra in corrispondenza della coordinata (0,1) dello spazio ROC, che rappresenta il 100% di sensibilità (quindi nessun falso negativo) ed il 100% di specificità (quindi nessun falso positivo) [69].

Dunque, quanto più la curva ROC è vicina all'angolo in alto a sinistra del piano e lontana dalla linea di non discriminazione, tanto migliore sarà la capacità di classificazione della rete neurale.

Dalla curva ROC è possibile calcolare un ulteriore parametro di valutazione denominato AUC (Area Under Curve).

L' AUC è una delle metriche più utilizzate per la valutazione delle performance dei modelli e viene impiegata nei task di classificazione binaria (compreso nella Challenge) con varie impostazioni di soglia. Assume valori nel range tra 0 ed 1, con i valori più alti che indicano migliori prestazioni di classificazione e distinzione tra pazienti positivi, positivi gravi e sani [66]. Dunque, l'AUC misura il grado di separabilità ovvero quanto il modello è in grado di discriminare tra le classi.

Di seguito si riportano i range di valori che l'AUC può assumere insieme ad una valutazione del classificatore che li determina:

- $AUC=0.5$ → il classificatore non è informativo e classifica in modo casuale;
- $0.5 < AUC < 0.7$ → il classificatore è poco accurato;
- $0.7 < AUC < 0.9$ → il classificatore è moderatamente accurato;
- $0.9 < AUC < 1$ → il classificatore è molto accurato;
- $AUC=1$ → il classificatore è perfetto.

Pertanto, si considera sufficiente un classificatore con un valore di AUC superiore all'80% [71].

Capitolo 6

Risultati

Nel presente capitolo sono illustrati i risultati dei metodi proposti nel paragrafo 5.5. Le architetture CNN (ResNet 50, ResNet 101, W-ResNet 50, HRNet) vengono addestrate sulle slice pre-processate e sono caratterizzate tutte dagli stessi parametri di allenamento. Si riportano dapprima i risultati delle metriche per ciascuno dei singoli classificatori proposti e, in ultimo, le metriche ottenute dall'ensemble dei modelli, ossia dalla combinazione delle reti.

I risultati delle classificazioni vengono riportati e confrontati al fine di indagare quale sistema porti al conseguimento delle migliori performance sul dataset a disposizione. L'obiettivo della sfida consiste nella previsione di Covid-19 grave, definito come intubazione o morte entro un mese dall'acquisizione della TC. Per valutare i risultati dei partecipanti alla STOIC Challenge e stilare la classifica della sfida, è stata utilizzata la metrica AUC calcolata per la severità al test Covid-19 RT-PCR (AUC calcolata solo con pazienti positivi al Covid-19), che costituisce lo standard di riferimento (truth label). La AUC per la positività al test sarà valutata come metrica secondaria nella classifica.

Dunque, l'identificazione della miglior rete o della miglior combinazione di modelli si basa esclusivamente sui valori di AUC ottenuti, sia in termini di positività che in termini di severità. Tuttavia, per avere una visione globale delle performance diagnostiche delle reti, sono state riportate anche altre metriche, utilizzando anche in questo caso i risultati della RT-PCR come standard di riferimento.

Le performances dei classificatori vengono valutate sia sulle immagini di Training che su quelle di Test, non mostrando le performance sul Validation set per non appesantire troppo la trattazione dei risultati. Inoltre, per rendere i risultati più facilmente interpretabili, verranno riportare solamente, oltre all'AUC, le confusion matrix su Training e Test set, in unione alle seguenti due metriche:

- accuratezza: importante per una valutazione generale della qualità del classificatore;

- Macro F1 score e Weighted F1 score: trattandosi di un problema di classificazione multi-classe, si utilizzano le due tecniche di averaging ‘macro’ e ‘weighted’. La metrica F1 score è importante poiché consente di capire quanto il classificatore sia preciso e robusto. Dunque, questa metrica consente, in modo indiretto, di avere anche un’idea delle performance in termini di precisione e recall.

6.1 Risultati sul Training set

In questo paragrafo sono riportati i risultati ottenuti per ogni rete addestrata sui pazienti del Training set, in funzione delle metriche sopracitate. Questi risultati sono stati conseguiti utilizzando l’approccio 3 (descritto al paragrafo 5.4), il quale permette di ottenere performance superiore ad altri approcci di aggregazione delle previsioni, come si evince, in seguito, nel paragrafo 6.3.

6.1.1 Risultati ResNet50

Di seguito si riportano i risultati del testing della ResNet50 sulle immagini del Training set. In Figura 6.1 vengono mostrate due confusion matrix, una per ogni dataset costruito.

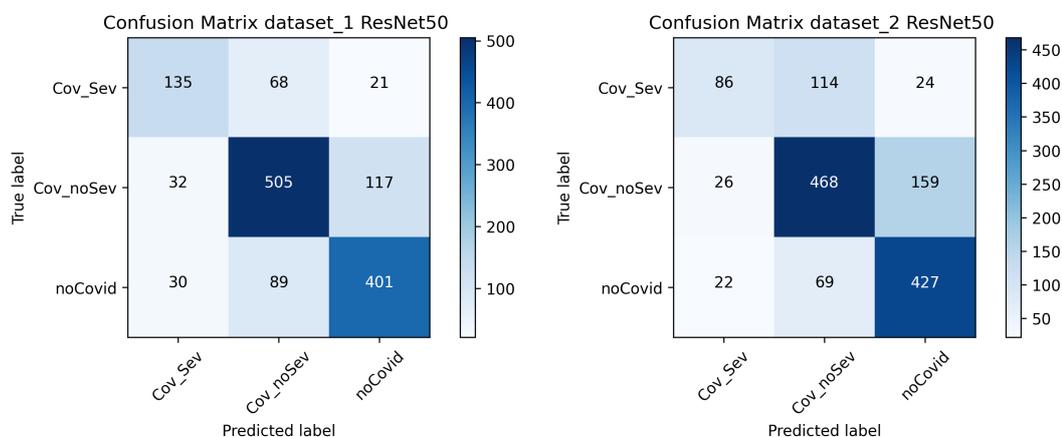


Figura 6.1: Confusion matrix sul Training set con rete ResNet50 su ambo i dataset

A partire dalle confusion matrix riportate, vengono calcolate le metriche di valutazione specificate nel paragrafo 6, al fine di quantificare ed analizzare le prestazioni del modello di classificazione sul Training set (Tabella 6.1).

DATASET	Accuracy	AUC severity	AUC positivity	Macro F1 Score	Weighted F1 Score
Dataset 1	0.7446	0.7749	0.8070	0.7220	0.7435
Dataset 2	0.7032	0.6715	0.8078	0.6518	0.6943

Tabella 6.1: Tabella metriche di valutazione ResNet50 sui Training set dei due dataset

Analizzando la tabella riportata con le metriche, è possibile riscontrare buone performance del classificatore sulle immagini di Training. Questa architettura di rete permette di raggiungere sulla metrica principale dell'AUC severity un valore pari a 0.7749 sul Dataset 1 ed un valore pari a 0.6715 sul Dataset 2, dimostrandosi un classificatore abbastanza accurato sulle immagini di Training, soprattutto sul Dataset 1.

6.1.2 Risultati ResNet101

Di seguito si riportano i risultati del testing della ResNet101 sulle immagini del Training set. In Figura 6.2 vengono mostrate due confusion matrix, una per ogni dataset costruito.

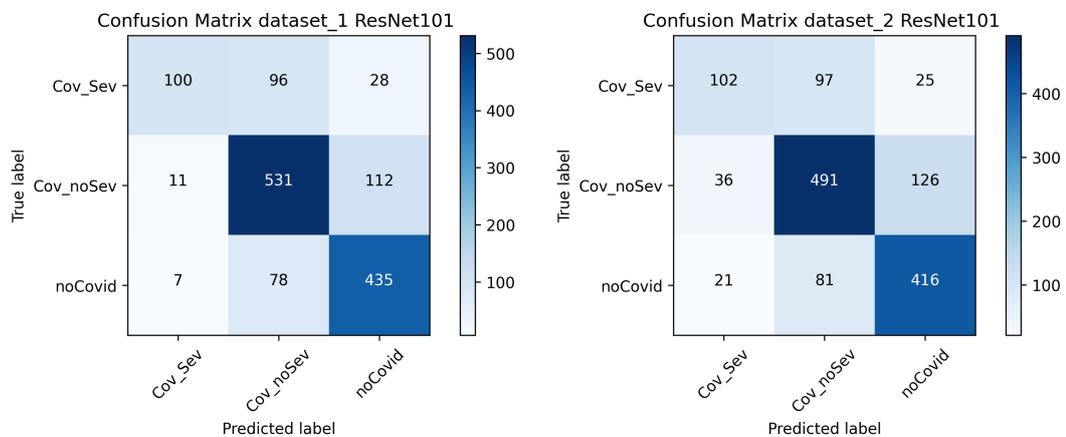


Figura 6.2: Confusion matrix sul Training set con rete ResNet101 su ambo i dataset

Come per la ResNet50, in Tabella 6.2 vengono riportate le metriche di valutazione per analizzare le performance della rete di classificazione sul Training set.

DATASET	Accuracy	AUC severity	AUC positivity	Macro F1 Score	Weighted F1 Score
Dataset 1	0.7625	0.7155	0.8385	0.7203	0.7548
Dataset 2	0.7233	0.7033	0.8155	0.6808	0.7180

Tabella 6.2: Tabella metriche di valutazione ResNet101 sui Training set dei due dataset

Analizzando la tabella, è possibile apprezzare buone performance del classificatore sulle immagini di Training. Questa architettura permette di raggiungere, in termini di AUC severity, un valore inferiore alla rete precedente (di circa il 6%) sul Dataset 1, mentre ottiene una performance superiore di circa il 3% sul Dataset 2. In termini di AUC positivity, metrica secondaria della Challenge, questa architettura consente invece di ottenere performance superiori in ambo i dataset. Facendo poi un confronto anche sulle altre metriche, si riscontrano delle performance che risultano essere superiori per tale rete rispetto alle altre architetture proposte.

6.1.3 Risultati W-ResNet50

Vengono ora riportati i risultati del testing della W-ResNet50 sulle immagini del Training set. In Figura 6.3 sono mostrate due confusion matrix, una per ogni dataset costruito.

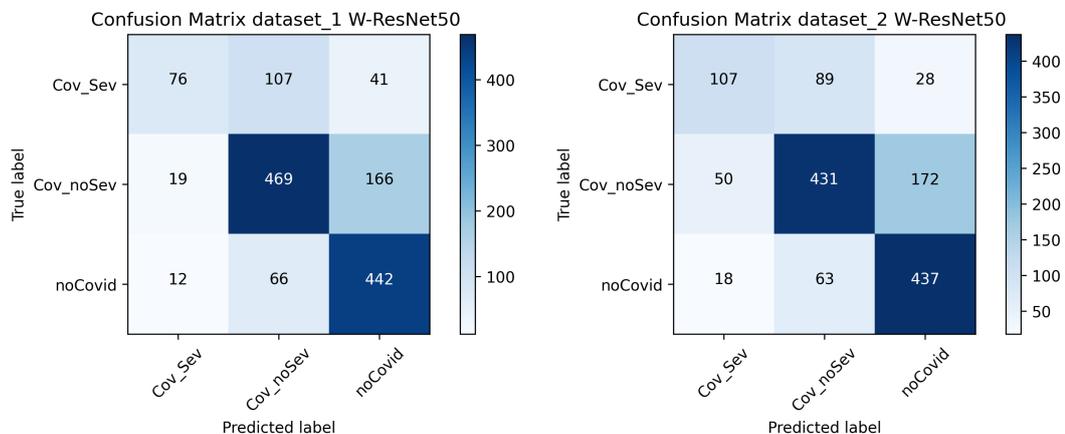


Figura 6.3: Confusion matrix sul Training set con rete W-ResNet50 su ambo i dataset

Si riportano in Tabella 6.3 le metriche di valutazione sul Training set.

DATASET	Accuracy	AUC severity	AUC positivity	Macro F1 Score	Weighted F1 Score
Dataset 1	0.7060	0.6564	0.8071	0.6464	0.6934
Dataset 2	0.6989	0.7098	0.8078	0.6635	0.6936

Tabella 6.3: Tabella metriche di valutazione W-ResNet50 sui Training set dei due dataset

Dalla tabella è possibile notare un valore di AUC severity sul Dataset 1 nettamente inferiore rispetto alle due reti precedenti, mentre sul Dataset 2 si ottengono performance in linea con le prestazioni ottenute dalla ResNet101. In termini di AUC positivity, si ottengono prestazioni pari a quelle ottenute con l'architettura ResNet50, su ambo i dataset.

6.1.4 Risultati HRNet

Di seguito si riportano i risultati del testing della HRNet sulle immagini del Training set. In Figura 6.4 vengono mostrate due confusion matrix, una per ogni dataset costruito.

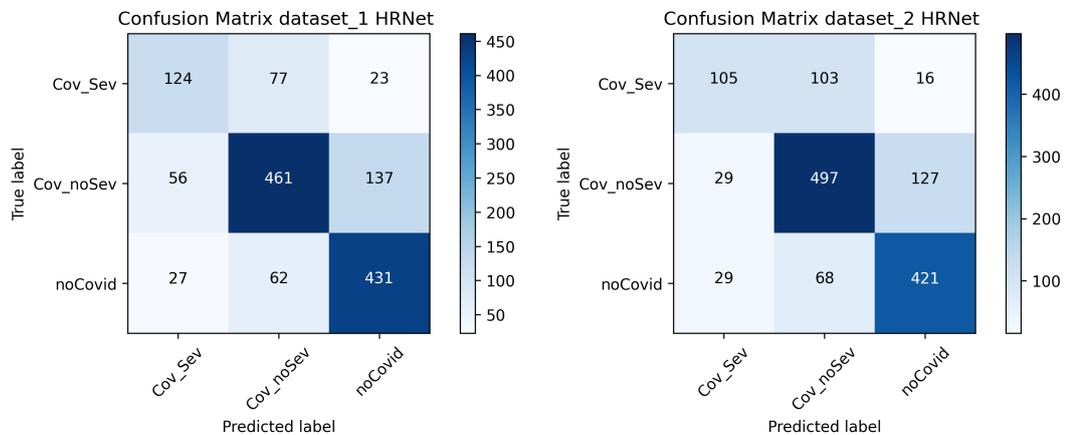


Figura 6.4: Confusion matrix sul Training set con rete HRNet su ambo i dataset

A partire dalle confusion matrix riportate, vengono calcolate le metriche di valutazione per effettuare un confronto con le prestazioni ottenute dalle reti precedenti sul Training set (Tabella 6.4).

DATASET	Accuracy	AUC severity	AUC positivity	Macro F1 Score	Weighted F1 Score
Dataset 1	0.7268	0.7414	0.8233	0.6955	0.7247
Dataset 2	0.7333	0.7096	0.8248	0.6911	0.7283

Tabella 6.4: Tabella metriche di valutazione HRNet sui Training set dei due dataset

Analizzando la tabella riportata, è possibile apprezzare buone performance del classificatore sulle immagini di Training su ambo i dataset e sulle due AUC calcolate. In particolare, questa architettura permette di raggiungere, sulla metrica principale dell’AUC severity, un valore pari a 0.7414 sul Dataset 1 ed un valore pari a 0.7096 sul Dataset 2. In definitiva, si nota una performance sul Dataset 2 in linea con le due reti precedenti, mentre sul Dataset 1 le performance sono superiori alle reti ResNet101 e W-ResNet50, rispettivamente del 3% e del 9%. Al contrario, rispetto alla ResNet50 si ottengono prestazioni inferiori del 3%.

6.2 Risultati sul Test set

In questo paragrafo sono riportati i risultati ottenuti per ogni rete addestrata sui pazienti del Test set, in funzione delle metriche sopracitate. Questi risultati sono stati conseguiti utilizzando l’approccio 3, il quale permette di ottenere performance superiore ad altri approcci di aggregazione delle previsioni, come si evince, in seguito, nel paragrafo 6.3.

6.2.1 Risultati ResNet50

Si riportano i risultati del testing della ResNet50 sulle immagini del Test set. In Figura 6.5 vengono mostrate due confusion matrix, una per ogni dataset costruito.

A partire dalle due confusion matrix si calcolano le metriche di valutazione specificate nel paragrafo 6, al fine di quantificare ed analizzare le prestazioni del modello di classificazione sul Test set (Tabella 6.5).

DATASET	Accuracy	AUC severity	AUC positivity	Macro F1 Score	Weighted F1 Score
Dataset 1	0.6150	0.5908	0.7154	0.5300	0.6111
Dataset 2	0.6450	0.5681	0.7388	0.5334	0.6299

Tabella 6.5: Tabella metriche di valutazione ResNet50 sui Test set dei due dataset

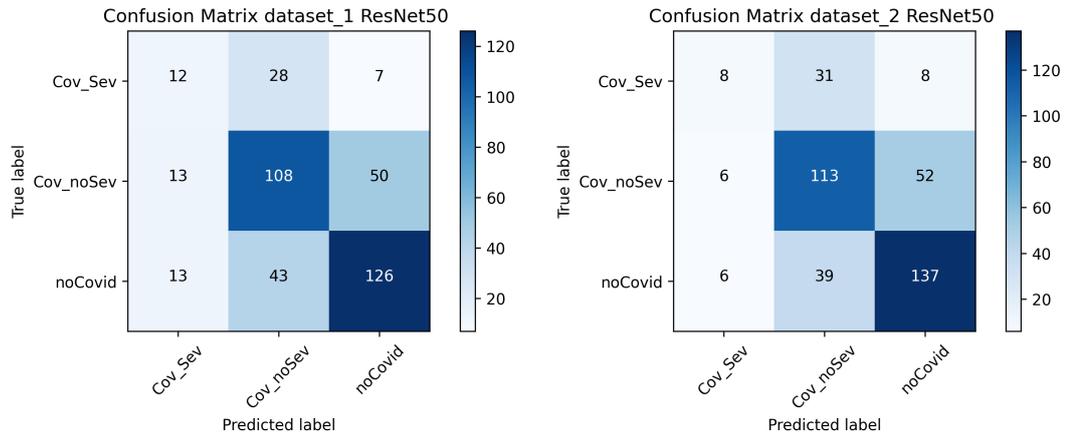


Figura 6.5: Confusion matrix sul Test set con rete ResNet50 su ambo i dataset

Analizzando la tabella riportata con le metriche, è possibile notare performance medie del classificatore sulle immagini di Test. Questa architettura di rete permette di raggiungere sulla metrica principale dell’AUC severity, un valore pari a 0.5908 sul Dataset 1 ed un valore pari a 0.56.81 sul Dataset 2, mostrando performance modeste su tale metrica. Al contrario, i risultati sono migliori in termini di AUC positivity, con una performance su ambo i dataset leggermente superiore al 70%.

6.2.2 Risultati ResNet101

Di seguito si riportano i risultati del testing della ResNet101 sulle immagini del Test set, mostrando in Figura 6.6 le due confusion matrix, una per ogni dataset costruito.

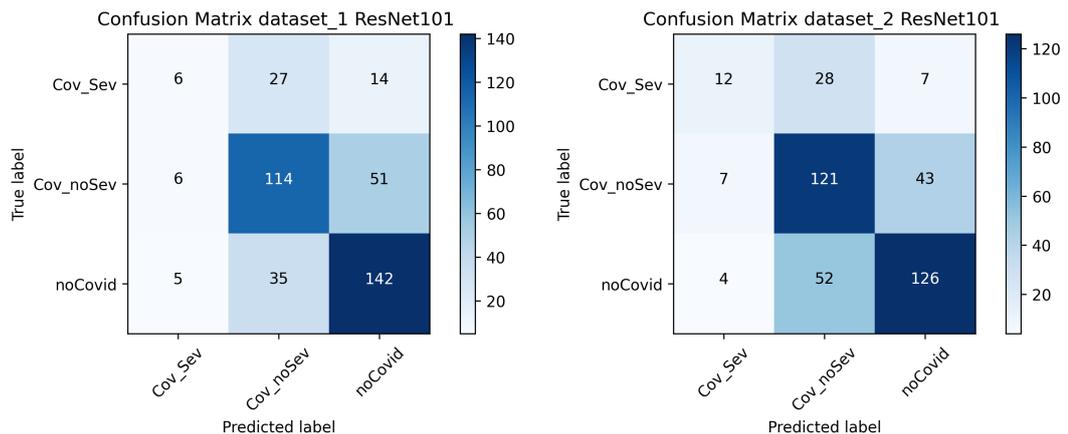


Figura 6.6: Confusion matrix sul Test set con rete ResNet101 su ambo i dataset

A partire dalle confusion matrix, vengono riportate in Tabella 6.6 le metriche di valutazione del classificatore sui pazienti del Test set.

DATASET	Accuracy	AUC severity	AUC positivity	Macro F1 Score	Weighted F1 Score
Dataset 1	0.6550	0.5482	0.7410	0.5249	0.6351
Dataset 2	0.6475	0.6121	0.7315	0.5658	0.6387

Tabella 6.6: Tabella metriche di valutazione ResNet101 sui Test set dei due dataset

Analizzando la tabella riportata, si nota un valore di AUC severity sul Dataset 1 inferiore rispetto alla rete precedente, mentre sul Dataset 2 si ottiene un miglioramento che porta ad una performance superiore al 60% sul riconoscimento del Covid in forma grave. Inoltre, sul Dataset 2 l'AUC positivity è circa pari alla stessa metrica calcolata per la ResNet50 sul Test set, mentre si ottiene un miglioramento del 3% su questa metrica secondaria per l'architettura in analisi.

6.2.3 Risultati W-ResNet50

Di seguito si riportano i risultati del testing della W-ResNet50 sulle immagini del Test set. In Figura 6.7 vengono mostrate due confusion matrix, una per ogni dataset.

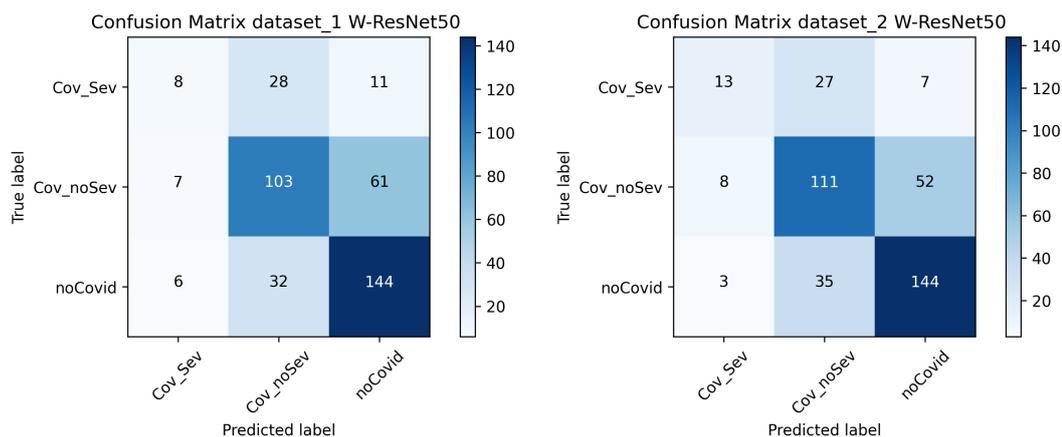


Figura 6.7: Confusion matrix sul Test set con rete W-ResNet50 su ambo i dataset

A partire dalle confusion matrix, vengono riportate in Tabella 6.7 le metriche di valutazione.

DATASET	Accuracy	AUC severity	AUC positivity	Macro F1 Score	Weighted F1 Score
Dataset 1	0.6375	0.5667	0.7305	0.5252	0.6206
Dataset 2	0.6700	0.6227	0.7603	0.5865	0.6593

Tabella 6.7: Tabella metriche di valutazione W-ResNet50 sui Test set dei due dataset

Analizzando la tabella, si riscontra sul Dataset 1 un valore di AUC severity ancora inferiore alla ResNet50, mentre sul Dataset 2 si ottiene un ulteriore miglioramento che attesta le performance a circa 0.62 in termini di AUC severity sul Test set del Dataset 2. In termini di AUC positivity, sul Dataset 2 si ottiene circa 0.76, superando dunque le performance delle altre architetture; sul Dataset 1 l'AUC positivity è inferiore dell'1% rispetto alla ResNet101.

6.2.4 Risultati HRNet

Si riportano infine i risultati del testing della HRNet sulle immagini del Test, comprensive delle due confusion matrix, una per ogni dataset costruito, rappresentate in Figura 6.8.

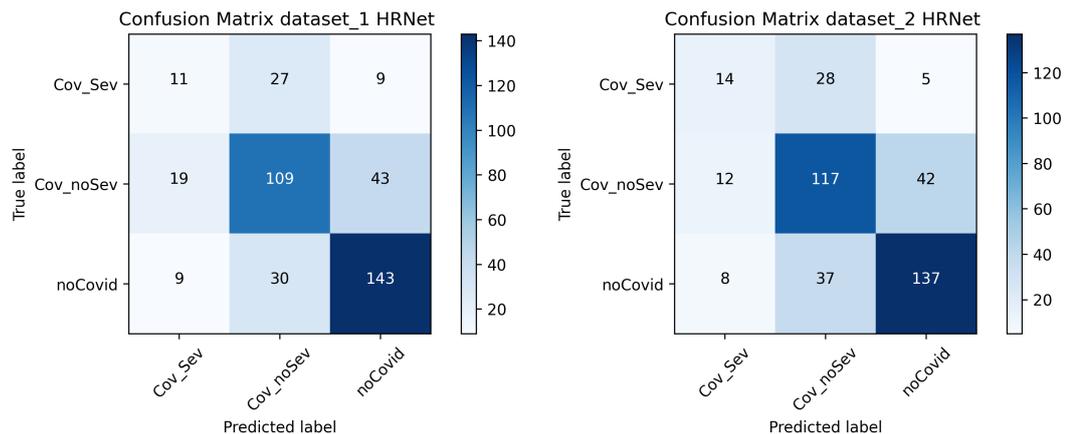


Figura 6.8: Confusion matrix sul Test set con rete HRNet su ambo i dataset

A partire dalle confusion matrix, si possono calcolare le metriche di valutazione per analizzare le prestazioni del classificatore sul Test set (Tabella 6.8).

DATASET	Accuracy	AUC severity	AUC positivity	Macro F1 Score	Weighted F1 Score
Dataset 1	0.6575	0.5774	0.7736	0.5538	0.6518
Dataset 2	0.6700	0.6206	0.7686	0.5857	0.6646

Tabella 6.8: Tabella metriche di valutazione HRNet sui Test set dei due dataset

Analizzando la tabella riportata, si riscontrano performance abbastanza simili alle prestazioni ottenute dalla W-ResNet50, sia in termini di AUC severity che in termini di AUC positivity, con un aumento del 3% dell'AUC positivity sul Test set del Dataset 1, rispetto alla rete precedente. Facendo poi un confronto anche sulle altre metriche, si riscontrano delle performance che risultano essere superiori per tale rete rispetto alle altre architetture.

6.3 Risultati ensemble di più reti sui due dataset

In ultimo vengono presentati i risultati sui tre set utilizzando un ensemble delle predizioni per generare la previsione finale ed ottenere performance superiori rispetto alle singole reti. Questo algoritmo di fusione delle predizioni sfrutta le softmax ottenute per ogni paziente da ciascuna rete per definirne la classe di appartenenze. Per la combinazione degli output, sono stati tentati approcci di combinazioni a 2 reti, 3 reti e 4 reti, per analizzare quale configurazione consentisse di massimizzare l'AUC.

L'approccio che ha fornito migliori risultati (riportato anche in Figura 6.9) prevede la combinazione, per ciascun dataset, delle softmax di due reti. Per il

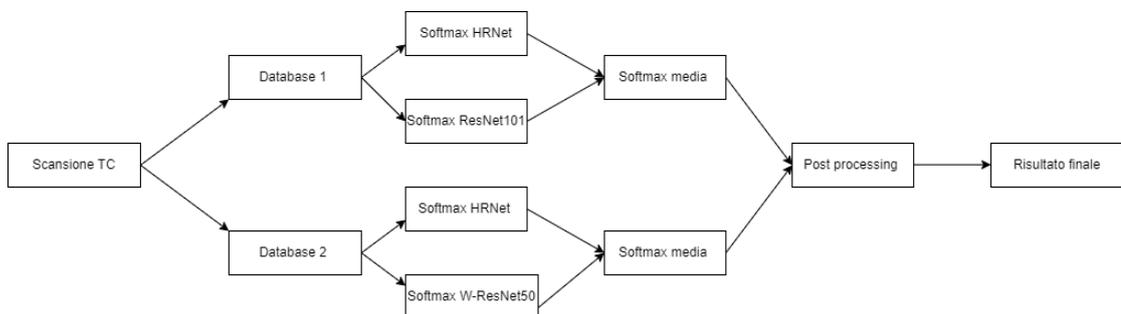


Figura 6.9: Schema fusione delle decisioni mediante utilizzo delle softmax

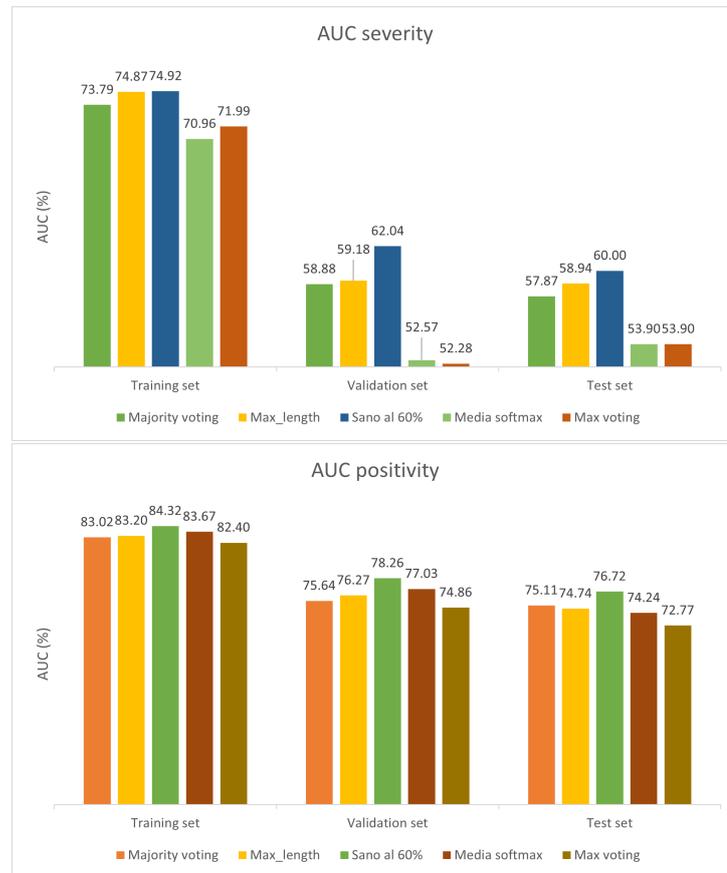


Figura 6.10: AUC severity ed AUC positivity per l'ensemble delle reti, calcolate per i cinque approcci di aggregazione delle predizioni.

Dataset 1, la combinazione di reti che fornisce le performance migliori è costituita dalla HRNet e dalla ResNet 101, mentre per il Dataset 2 è costituita dalla HRNet e dalla W-ResNet 50. Sono state mediate dunque le softmax, su ogni dataset, di queste coppie di reti e le due softmax medie risultanti (una per ogni dataset) sono state concatenate in un unico array. Per ogni riga dell'array, che coincide con le probabilità medie di appartenenza della singola slice a ciascuna classe, viene identificata la classe con lo score maggiore e si crea un vettore contenente le predizioni per ogni slice.

Sull'array di predizioni ottenuto per ciascun paziente, si utilizza l'approccio 3 come tecnica di post-processing (definito nel paragrafo 5.4.1), al fine di ottenere la classe finale del paziente.

Riprendendo i diversi approcci di aggregazione dei risultati, descritti in precedenza nel paragrafo 5.4, si riportano, per ciascuno di essi, i valori di AUC severity

ed AUC positivity calcolati su Training set, Validation set e Test set, per l'ensemble delle reti sopracitate. I grafici in Figura 6.10 mostrano le prestazioni di tale configurazione del sistema con le cinque diverse strategie di post-processing, sia in termini di AUC severity che di AUC positivity.

Dall'osservazione dei risultati riportati, è evidente che, per ambo le metriche riportate, la strategia 3 fornisce complessivamente le performance finali migliori su tutti i set (Training, Validation e Test set), rispetto alle altre tecniche di post-processing, con un incremento delle prestazioni mediamente compreso tra 1% ed il 3%.

In Tabella 6.9 si riportano i valori delle metriche calcolate con la soluzione dell'ensemble delle reti e con l'Approccio 3 come metodo di combinazione delle predizioni, comprensive del Validation set per avere una panoramica completa delle performance del miglior sistema di classificazione sviluppato, mentre in Figura 6.11 vengono riportate le rispettive confusion matrix ottenute sul Training e sul Test set, così come erano state riportate nelle singole reti.

Reti	Set	Accuracy	AUC severity	AUC positivity	Macro F1 Score	Weighted F1 Score
HRNet-ResNet101/W-ResNet50-HRNet	Training	0.7661	0.7492	0.8432	0.7362	0.7615
	Validation	0.6900	0.6204	0.7826	0.6067	0.6762
	Test	0.6800	0.6000	0.7672	0.5802	0.6674

Tabella 6.9: Tabella confronti singole reti sui due Dataset

6.4 Confronto finale tra classificazioni

Analizzate le performance delle singole reti e delle combinazioni tra le reti, si è effettuato un confronto tra i risultati dei migliori classificatori per determinare quale approccio portasse alle migliori prestazioni. In particolare, si riportano in Tabella 6.10, per un confronto, i risultati che hanno generato le migliori performance, mediate sui tre set, delle singole reti applicate sui due dataset.

Dalla tabella si osserva che le performance ottenute su ambo i dataset sono molto simili tra le diverse reti (con una variazione di AUC severity pari all'1%). Su ambo i dataset, effettuando una media tra AUC severity ed AUC positivity sui tre set, si ottengono le migliori performance utilizzando la HRNet per la fase di testing.

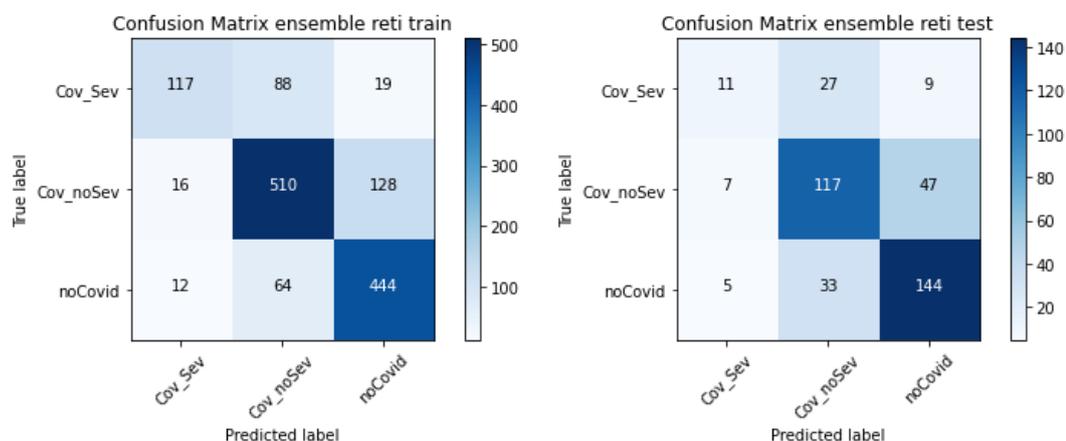


Figura 6.11: Confusion Matrix ottenute dall'ensemble di più classificatori

Medie 3 set	Rete	AUC severity	AUC positivity	Media AUC
Dataset 1	ResNet50	0.6505	0.7582	0.7043
	ResNet101	0.6003	0.7804	0.6904
	HRNet	0.6501	0.7846	0.7174
	W-ResNet50	0.5944	0.7714	0.6829
Dataset 2	ResNet50	0.6039	0.7693	0.6866
	ResNet101	0.6191	0.7638	0.6914
	HRNet	0.6478	0.7921	0.7199
	W-ResNet50	0.6401	0.7866	0.7134

Tabella 6.10: Tabella confronti singole reti sui due Dataset, mediate sui tre subsets

Vengono riportate nella Tabella 6.11 i valori di AUC severity, AUC positivity (mediate su Training, Validation e Test set) e la media delle due, calcolate per le singole reti sui due dataset e per l'ensemble di più reti, al fine di effettuare una analisi delle performance e definire la configurazione che determina le prestazioni migliori.

Sono riportate anche altre combinazioni di reti, in modo da fornire un confronto più completo.

Dall'analisi dei risultati, si riscontrano performance superiori in termini delle due metriche di AUC con la concatenazione delle due softmax medie, ricavate dalla combinazione delle reti HRNet e ResNet101 per il Dataset 1 e HRNet e W-ResNet50 per il Dataset 2.

In generale si riscontra che la combinazione delle reti permette di ottenere

Tipo di dataset	Rete/Reti	AUC severity	AUC positivity	Media AUC
Dataset 1	HRNet	0.6501	0.7846	0.7174
Dataset 2	HRNet	0.6478	0.7921	0.7199
Combinazione 2 Dataset	HRNet-ResNet101/ W-ResNet50- HRNet	0.6565	0.7977	0.7271
Combinazione 2 Dataset	HRNet-ResNet50/ W-ResNet50- HRNet	0.6536	0.7916	0.7226

Tabella 6.11: Tabella confronti performance

prestazioni superiori rispetto alle singole reti.

Capitolo 7

Conclusioni e sviluppi futuri

In questo progetto di tesi è stato proposto un algoritmo di classificazione completamente automatico basato su reti di deep learning che avevano come fine ultimo la diagnosi, direttamente da scansioni TC, di pazienti con forme gravi di Covid-19 e pazienti positivi all'infezione.

La parte su cui è stata posta maggior attenzione consiste nella fase di pre-processing, ossia nell'ottimizzazione del dataset posto in input alle reti di classificazione. Questa fase ha previsto il rescaling delle immagini al fine di portare la dinamica delle slice su 256 valori, il 3D lung detection per l'individuazione del polmone dalle immagini TC, il data selection che ha permesso di rimuovere immagini ridondanti o non informative in modo da concentrare l'apprendimento delle reti solo su immagini contenenti il tessuto polmonare, la creazione di un secondo dataset contenente le immagini con la sola segmentazione del polmone ed infine il data balancing che ha permesso di evitare sbilanciamenti nel numero di slices a disposizione per ciascuna classe e per ciascun paziente.

In questo contesto, nonostante i tentativi di eliminare l'artefatto del supporto paziente, è stata riscontrata l'indesiderata presenza del supporto paziente nelle immagini del Dataset 2 in circa 8% dei pazienti di Training set, il 4% dei pazienti del Validation set e in circa il 4% sui pazienti del Test set. Tale corruzione potrebbe essere eliminata applicando tecniche di segmentazione polmonare più raffinate.

Complessivamente le performance risultano non troppo elevate, con il raggiungimento di un 65% di AUC severity mediato sui 3 set, ed un valore di AUC positivity medio pari a circa 80% per la configurazione che porta alle migliori performance di classificazione.

In particolare, il sistema che porta alle migliori prestazioni delle due metriche di AUC consiste nella concatenazione delle due softmax medie, ricavate dalla combinazione delle reti HRNet e ResNet101 addestrate e testate sul Dataset 1 e HRNet e W-ResNet50 addestrate e testate sul Dataset 2. Si ottiene un aumento percentuale medio delle performance pari al 1-2% rispetto alle performance della

HRNet ossia della rete con le migliori performance singole, con un incremento principalmente su Test set e Validation set. La classifica della Challenge è basata sulle performance che la soluzione proposta ottiene sul Test set in termini di AUC severity; in questo progetto di tesi è stata ottenuta una AUC severity sul Test set pari a 0.60024, che colloca questo lavoro in 116° posizione. In termini di AUC positivity (metrica secondaria non presa in considerazione per stilare la classifica), si hanno performance superiori a molti altri progetti che hanno una miglior posizione in classifica.

Un approccio a 2 classi potrebbe essere testato per analizzare le performance in termini di singoli task: uno per la rilevazione della positività del paziente al Covid-19 ed uno per definire la severità della patologia. Questa possibilità è stata scartata, preferendo lo sviluppo di una rete a 3 classi per avere una maggiore e netta divisione tra le classi ed evitare una propagazione degli errori che avrebbe inficiato sul risultato finale. Tuttavia, una prova potrebbe essere condotta per aumentare le performance e la posizione in classifica in quanto, generalmente, i classificatori binari presentano migliori prestazioni rispetto a classificatori multi-classe.

Considerando poi che per le reti non sono state effettuate operazioni di tuning dei parametri ma che questi sono stati impostati in maniera casuale, si potrebbero aumentare ulteriormente le performance utilizzando delle reti ottimizzate in modo complessivo.

Un altro tentativo può essere effettuato per l'addestramento di altre reti come i transformer (vision transformer e swin transformer) o altre reti al di fuori delle dieci architetture testate in questo progetto.

I risultati ottenuti dal sistema sviluppato rappresentano comunque una buona base per una successiva fase di miglioramenti che possono essere apportati.

Bibliografia

- [1] *Apparato Respiratorio*. URL: <https://www.my-personaltrainer.it/salute-benessere/apparato-respiratorio.html> (cit. alle pp. 1, 3-5).
- [2] *Trachea*. URL: <https://www.medicinapertutti.it/argomento/trachea/#:~:text=La%5C%20trachea%5C%20%5C%C3%5C%A8%5C%20un%5C%20canale,16%5C%2D18%5C%20mm%5C%20di%5C%20diametro.> (cit. a p. 2).
- [3] *Polmoni*. URL: <https://www.nurse24.it/studenti/anatomia/polmone-organo-come-funziona-come-fatto.html> (cit. alle pp. 2, 4).
- [4] *Polmoni: anatomia e funzione*. URL: <https://www.my-personaltrainer.it/fisiologia/polmoni.html> (cit. a p. 2).
- [5] *Polmoni e apparato respiratorio*. URL: <https://www.my-personaltrainer.it/fisiologia/polmoni1.html> (cit. a p. 2).
- [6] *Polmoni*. URL: <https://www.humanitas.it/enciclopedia/anatomia/apparato-respiratorio/polmoni/> (cit. a p. 3).
- [7] *Polmone umano*. URL: https://it.wikipedia.org/wiki/Polmone_umano (cit. a p. 3).
- [8] *Pleura*. URL: <https://it.wikipedia.org/wiki/Pleura> (cit. a p. 3).
- [9] Antonello D'Andrea et al. «L'imaging integrato nel percorso del paziente con COVID-19: Dalla diagnosi, al monitoraggio clinico, alla prognosi». In: *G. Ital. Cardiol* 21 (2020), pp. 345-353 (cit. alle pp. 5-7, 9, 14).
- [10] A Sironi di Vittorio. «IL FAVOLOSO INNESTO: STORIA, ANTROPOLOGIA ED ETICA DELLA VACCINAZIONE». In: (2022) (cit. a p. 6).
- [11] Woan Ching Serena Low, Joon Huang Chuah, Clarence Augustine TH Tee, Shazia Anis, Muhammad Ali Shoaib, Amir Faisal, Azira Khalil e Khin Wee Lai. «An overview of deep learning techniques on chest X-ray and CT scan identification of COVID-19». In: *Computational and Mathematical Methods in Medicine* 2021 (2021), pp. 1-17 (cit. a p. 6).

- [12] Thomas C Kwee e Robert M Kwee. «Chest CT in COVID-19: what the radiologist needs to know». In: *Radiographics* 40.7 (2020), pp. 1848–1865 (cit. alle pp. 6, 7, 13).
- [13] *Long Covid, cosa succede ai nostri polmoni?* URL: <https://www.humanitas-care.it/news/long-covid-cosa-succede-ai-nostri-polmoni/> (cit. alle pp. 6, 7).
- [14] *Polmonite o broncopolmonite: senza febbre? Sintomi.* URL: <https://www.doveecomemicro.it/enciclopedia/malattie/polmonite> (cit. a p. 7).
- [15] *Long Covid: quali sono gli effetti a lungo termine sui polmoni?* URL: <https://www.humanitas-sanpiox.it/news/long-covid-quali-sono-gli-effetti-a-lungo-termine-sui-polmoni/> (cit. a p. 7).
- [16] *Tampone Molecolare Covid-19: Tutto quello che c'è da sapere.* URL: <https://www.multimed.it/blog/blog/salute/tampone-molecolare-covid-19-tutto-quello-che-c-e-da-sapere> (cit. a p. 8).
- [17] *Tampone Rapido - Test Rapido COVID: Cos'è e Come Funziona?* URL: <https://www.my-personaltrainer.it/salute/tampone-rapido-covid.html> (cit. a p. 9).
- [18] *TC (Tomografia computerizzata).* URL: [https://www.airc.it/cancro/affronta-la-malattia/guida-agli-esami/tc-tomografia-computerizzata#:~:text=La%5C%20tomografia%5C%20computerizzata%5C%20\(TC%5C%20oppure,studio%5C%20dei%5C%20tumori%5C%20e%5C%20di](https://www.airc.it/cancro/affronta-la-malattia/guida-agli-esami/tc-tomografia-computerizzata#:~:text=La%5C%20tomografia%5C%20computerizzata%5C%20(TC%5C%20oppure,studio%5C%20dei%5C%20tumori%5C%20e%5C%20di) (cit. a p. 10).
- [19] Giuseppe Coppini, Stefano Diciotti e Guido Valli. *Bioimmagini*. Paris: Pàtron Editore, 2012 (cit. alle pp. 11, 12).
- [20] Sagar Sharma, Simone Sharma e Anidhya Athaiya. «Activation functions in neural networks». In: *Towards Data Sci* 6.12 (2017), pp. 310–316 (cit. alle pp. 15, 16, 19).
- [21] Agostino Giorgio. «Intelligenza Artificiale e diagnosi precoce del COVID-19». In: *LA COMUNICAZIONE* (), p. 16 (cit. alle pp. 15–18, 20).
- [22] *RETI NEURALI*. URL: <https://www.intelligenzaartificiale.it/reti-neurali/> (cit. a p. 16).
- [23] *Reti neurali*. URL: <https://www.ibm.com/it-it/cloud/learn/neural-networks> (cit. a p. 16).
- [24] *Reti neurali artificiali: cosa sono, come funzionano e perché vengono costruite.* URL: <https://www.geopop.it/reti-neurali-artificiali-cosa-sono-come-funzionano-e-perche-vengono-costruite/> (cit. alle pp. 17, 19).

- [25] *Il futuro dell'elettronica è simulare il cervello umano: si studia come replicare il funzionamento dei neuroni biologici*. URL: https://www.hwupgrade.it/news/scienza-tecnologia/il-futuro-dell-elettronica-e-simulare-il-cervello-umano-si-studia-come-replicare-il-funzionamento-dei-neuroni-biologici_109501.html (cit. a p. 18).
- [26] Giorgio Buttazzo. «Reti Neurali in grado di apprendere». In: *Ithaca: Viaggio nella Scienza* 2020.16 (2020), pp. 195–208 (cit. a p. 19).
- [27] Chiara Macchiavello. «Introduzione alle reti neurali». In: *atti del seminario tenuto all'università di Pavia il 17 (1992)* (cit. a p. 20).
- [28] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do e Kaori Togashi. «Convolutional neural networks: an overview and application in radiology». In: *Insights into imaging* 9 (2018), pp. 611–629 (cit. alle pp. 20–25, 27).
- [29] *Convolutional Neural Networks*. URL: <https://cs231n.github.io/convolutional-networks/> (cit. a p. 21).
- [30] Waseem Rawat e Zenghui Wang. «Deep convolutional neural networks for image classification: A comprehensive review». In: *Neural computation* 29.9 (2017), pp. 2352–2449 (cit. a p. 21).
- [31] Shankar Shambhu, Deepika Koundal, Prasenjit Das e Chetan Sharma. «Binary classification of covid-19 ct images using cnn: Covid diagnosis using ct». In: *International Journal of E-Health and Medical Communications (IJEHMC)* 13.2 (2021), pp. 1–13 (cit. a p. 24).
- [32] *Transfer learning from pre-trained models*. URL: <https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751> (cit. a p. 26).
- [33] *Che cos'è l'overfitting?* URL: <https://aws.amazon.com/it/what-is/overfitting/> (cit. a p. 26).
- [34] *Neural Networks: parameters, hyperparameters and optimization strategies*. URL: <https://towardsdatascience.com/neural-networks-parameters-hyperparameters-and-optimization-strategies-3f0842fac0a5> (cit. alle pp. 27, 57, 59, 60, 63).
- [35] *Fitting del modello: Underfitting vs. overfitting*. URL: https://docs.aws.amazon.com/it_it/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html (cit. a p. 27).
- [36] *ML | Underfitting and Overfitting*. URL: <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/> (cit. a p. 27).

- [37] AL Aswathy, Anand Hareendran e Vinod Chandra SS. «COVID-19 diagnosis and severity detection from CT-images using transfer learning and back propagation neural network». In: *Journal of infection and public health* 14.10 (2021), pp. 1435–1445 (cit. alle pp. 28, 29).
- [38] Arnab Kumar Mishra, Sujit Kumar Das, Pinki Roy e Sivaji Bandyopadhyay. «Identifying COVID19 from chest CT images: a deep convolutional neural networks based approach». In: *Journal of Healthcare Engineering* 2020 (2020) (cit. alle pp. 29, 30, 55).
- [39] Talha Anwar e Seemab Zakir. «Deep learning based diagnosis of COVID-19 using chest CT-scan images». In: *2020 IEEE 23rd international multitopic conference (INMIC)*. IEEE. 2020, pp. 1–5 (cit. alle pp. 30, 31).
- [40] Dan Nguyen, Fernando Kay, Jun Tan, Yulong Yan, Yee Seng Ng, Puneeth Iyengar, Ron Peshock e Steve Jiang. «Deep learning-based COVID-19 pneumonia classification using chest CT images: model generalizability». In: *Frontiers in Artificial Intelligence* 4 (2021), p. 694875 (cit. alle pp. 31, 32).
- [41] Sertan Serte e Hasan Demirel. «Deep learning for diagnosis of COVID-19 using 3D CT scans». In: *Computers in biology and medicine* 132 (2021), p. 104306 (cit. alle pp. 32, 33).
- [42] Vruddhi Shah, Rinkal Keniya, Akanksha Shridharani, Manav Punjabi, Jainam Shah e Ninad Mehendale. «Diagnosis of COVID-19 using CT scan images and deep learning techniques». In: *Emergency radiology* 28 (2021), pp. 497–505 (cit. a p. 34).
- [43] Varalakshmi Perumal, Vasumathi Narayanan e Sakthi Jaya Sundar Rajasekar. «Detection of COVID-19 using CXR and CT images using Transfer Learning and Haralick features». In: *Applied Intelligence* 51 (2021), pp. 341–358 (cit. alle pp. 35, 36).
- [44] Neha Rajawat, Bharat Singh Hada, Mayank Meghawat, Soniya Lalwani e Rajesh Kumar. «C-COVIDNet: A CNN model for COVID-19 detection using image processing». In: *Arabian Journal for Science and Engineering* 47.8 (2022), pp. 10811–10822 (cit. alle pp. 37, 38, 68).
- [45] URL: <https://github.com/open-mmlab/mclassification> (cit. a p. 54).
- [46] *Guide: ResNet-50*. URL: <https://datagen.tech/guides/computer-vision/resnet-50/> (cit. alle pp. 54, 55).
- [47] *A Review of Popular Deep Learning Architectures: ResNet, InceptionV3, and SqueezeNet*. URL: <https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/> (cit. alle pp. 54–56).

-
- [48] *An Overview of ResNet Architecture and Its Variants*. URL: <https://builtin.com/artificial-intelligence/resnet-architecture> (cit. a p. 54).
- [49] Awais Khan, Muhammad Attique Khan, Muhammad Younus Javed, Majed Alhaisoni, Usman Tariq, Seifedine Kadry, Jung-In Choi e Yunyoung Nam. «Human gait recognition using deep learning and improved ant colony optimization». In: (2022) (cit. a p. 55).
- [50] Jingdong Wang et al. «Deep high-resolution representation learning for visual recognition». In: *IEEE transactions on pattern analysis and machine intelligence* 43.10 (2020), pp. 3349–3364 (cit. a p. 56).
- [51] Yuyang Li, Bolin Fu, Xidong Sun, Donglin Fan, Yeqiao Wang, Hongchang He, Ertao Gao, Wen He e Yuefeng Yao. «Comparison of Different Transfer Learning Methods for Classification of Mangrove Communities Using MCCUNet and UAV Multispectral Images». In: *Remote Sensing* 14.21 (2022), p. 5533 (cit. a p. 57).
- [52] *Neural Network: Introduction to Learning Rate*. URL: <https://studymachinelearning.com/neural-network-introduction-to-learning-rate/> (cit. a p. 59).
- [53] *Difference Between a Batch and an Epoch in a Neural Network*. URL: <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/> (cit. a p. 59).
- [54] *Loss Functions and Their Use In Neural Networks*. URL: <https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9> (cit. a p. 60).
- [55] *Entropy in Machine Learning*. URL: <https://www.javatpoint.com/entropy-in-machine-learning#:~:text=Entropy%5C%20is%5C%20defined%5C%20as%5C%20the,or%5C%20impurity%5C%20in%5C%20the%5C%20system.> (cit. a p. 60).
- [56] *Optimizers in Deep Learning*. URL: <https://medium.com/mllearning-ai/optimizers-in-deep-learning-7bf81fed78a0> (cit. a p. 61).
- [57] *Tuning Adam Optimizer Parameters in PyTorch*. URL: <https://www.kdnuggets.com/2022/12/tuning-adam-optimizer-parameters-pytorch.html> (cit. a p. 61).
- [58] *Hyper-parameter Tuning Techniques in Deep Learning*. URL: <https://towardsdatascience.com/hyper-parameter-tuning-techniques-in-deep-learning-4dad592c63c8> (cit. a p. 62).
- [59] *CutMix, MixUp, and RandAugment image augmentation with KerasCV*. URL: https://keras.io/guides/keras_cv/cut_mix_mix_up_and_rand_augment/ (cit. a p. 63).

-
- [60] *Cutmix.. vs Mixup.. vs Gridmask ..vs Cutout...* URL: <https://www.kaggle.com/code/saife245/cutmix-vs-mixup-vs-gridmask-vs-cutout> (cit. a p. 63).
- [61] *Good practices for neural network training: identify, save, and document best models.* URL: <https://nicjac.dev/posts/identify-best-model/> (cit. a p. 64).
- [62] *Max-voting.* URL: <https://www.oreilly.com/library/view/ensemble-machine-learning/9781789136609/6571afd0-0bac-4bb6-9698-c68504baebae.xhtml> (cit. a p. 66).
- [63] *Confusion matrix.* URL: https://en.wikipedia.org/wiki/Confusion_matrix (cit. a p. 67).
- [64] *Confusion Matrix Visualization.* URL: <https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea> (cit. a p. 68).
- [65] *Guide to Confusion Matrices Classification Performance Metrics.* URL: <https://towardsdatascience.com/guide-to-confusion-matrices-classification-performance-metrics-a0ebfc08408e> (cit. a p. 69).
- [66] *Metrics to Evaluate your Machine Learning Algorithm.* URL: <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234> (cit. alle pp. 69, 72).
- [67] *Understanding the Confusion Matrix (II).* URL: <https://dev.to/overrideveloper/understanding-the-confusion-matrix-264i> (cit. a p. 70).
- [68] *Multi-class Model Evaluation with Confusion Matrix and Classification Report.* URL: <https://towardsai.net/p/1/multi-class-model-evaluation-with-confusion-matrix-and-classification-report> (cit. a p. 71).
- [69] *Receiver operating characteristic.* URL: https://en.wikipedia.org/wiki/Receiver_operating_characteristic (cit. a p. 72).
- [70] *AUC (Area under the ROC Curve).* URL: <https://machine-learning.paperspace.com/wiki/auc-area-under-the-roc-curve> (cit. a p. 72).
- [71] Graziella D'Arrigo, Fabio Provenzano, Claudia Torino, Carmine Zoccali e Giovanni Tripepi. «I test diagnostici e l'analisi della curva ROC». In: *G Ital Nefrol* 28.6 (2011), pp. 642–647 (cit. a p. 72).

Ringraziamenti

Ringrazio innanzitutto il mio relatore Massimo Salvi per avermi concesso la possibilità di lavorare su questo progetto di tesi e per avermi accompagnato nel percorso di tesi, sia triennale che magistrale. È sempre stato estremamente disponibile, dispensandomi consigli ogni volta ve ne fosse bisogno. Mi ha trasmesso la sua passione per ciò su cui lavora e per ciò che insegna. Le auguro di poter raggiungere le Sue aspirazioni.

Ringrazio anche il Prof. Molinari per la passione che mi ha trasmesso durante le sue spiegazioni in questi anni.

Ringrazio i miei genitori per tutti gli sforzi che hanno fatto e per non avermi mai fatto mancare nulla. Grazie per avermi incoraggiato prima di ogni esame, di avermi trasmesso il vostro orgoglio nei bei momenti e di avermi dato ancora più supporto dopo ogni momento no.

Grazie mamma, grazie papà, per avermi fatto crescere con dei valori che porterò sempre con me, per avermi fatto diventare, grazie ai vostri insegnamenti, l'uomo che sono ora.

Senza di voi e senza la vostra fiducia nelle mie potenzialità non sarei arrivato a questo incredibile traguardo.

Scusate se nell'ultimo periodo sono stato un bel po' teso ed ho avuto qualche sclero di troppo. Purtroppo non sono il tipo che riesce a gestire bene la tensione.

Infine, voglio dirvi che questa tesi è dedicata anche a voi per tutto l'appoggio ed il sostegno dal punto di vista economico, morale ed affettivo e per avermi mostrato un'enorme vicinanza nonostante i 1600 km di distanza tra noi.

Grazie papà, grazie mamma. Vi amo.

Grazie Davide, o per me Daviduccio, per tutte le notti che mi hai ospitato a casa tua, per le risate che ci siamo fatti in questi due anni a Torino, per avermi fatto compagnia e per tutti gli aiuti preziosi che mi hai dato in questi 5 anni. In quest'ultimo periodo così stressante hai reso le mie giornate più leggere e spensierate e di questo ti ringrazio tanto.

Sono felice di esserci avvicinati molto in questi ultimi due anni e mi auguro tu sappia che puoi sempre contare su di me. Sono orgoglioso di te e della forza che metti nello studio e non vedo l'ora che anche tu possa raggiungere questo traguardo.

Grazie a Serena, che ormai da più di 8 anni sei entrata nella mia vita, sostenendomi in ogni momento nonostante i tanti chilometri di distanza. Mi hai supportato tantissimo, facendomi sempre sentire quanto tu fossi fiera di me.

Grazie dell'aiuto che mi hai dato in questi anni, grazie di avermi corretto l'italiano nelle e-mail che mandavo e grazie di avermi sopportato nei momenti di sclero; soprattutto in quest'ultimo periodo ammetto che non è stato facile e per questo scusami.

Non basterebbero le parole per ringraziarti di ciò che fai ed hai fatto per me ogni giorno e della forza che mi trasmetti, soprattutto nei periodi più duri e stressanti. Grazie delle parole di supporto, soprattutto quando, spesso, ti dicevo che non mi sentivo preparato abbastanza. Grazie per la festa che fai quando torno a casa dopo tanti mesi di distanza.

Grazie della tua presenza nella mia vita universitaria con le videochiamate fissate ogni giorno alle 19.30.

Cercherò anche io di risollevarvi sempre il morale in tutti i momenti no e di sostenervi nel tuo prossimo percorso verso la laurea, dandovi il massimo per farvi sentire supportata come tu hai fatto per me.

Grazie di tutto e Ti Amo.

Grazie a Seba, coinquilino, migliore amico e compagno di fantacalcio, per tutti i momenti che abbiamo passato insieme in questi anni di convivenza. Mi sembra ieri che al primo anno guardavamo ogni domenica le partite delle 15 su Sky go ed ora siamo entrambi laureati. Abbiamo patito il freddo in inverno ed il caldo in estate, ma nonostante questo non abbiamo mai smesso di morire dalle risate e cazzeggiare ogni giorno. Per questo sono stra-contento di tutta l'esperienza che abbiamo condiviso e la ricorderò per tutta la mia vita.

Sei un grande punto di riferimento; mi aiuti e mi hai aiutato e spronato tantissimo e sono veramente felice di averti conosciuto.

Grazie alla mia famiglia: zii, zie, cugini e cugine, per essermi stati tutti vicini ed avermi accompagnato in tutto il mio percorso. Anche con semplici parole mi avete dimostrato tanto affetto e fiducia in questi anni e di ciò vi sono immensamente grato.

Grazie alla nonna, che in questi 5 anni non mi ha mai fatto mancare il suo supporto ed il suo affetto, chiamandomi o informandosi dopo ogni esame che sostenevo. Le tue parole sono per me sempre tanto importanti.

Grazie alla mia seconda famiglia, Cinzia, Salvatore e Clelia per il supporto e l'in bocca al lupo che mi avete mandato prima di ogni esame, per i saluti di sfuggita che ci mandiamo quando sono in videochiamata con Serena e grazie di avermi accolto tra di voi come un figlio e fratello. La vostra vicinanza ed il vostro supporto non è mai mancato e vi ringrazio enormemente di questo.

Grazie ai miei amici che mi sono stati vicini e si sono sempre preoccupati di chiedermi come andasse la vita universitaria a Torino. Non so se voi ricordate, ma io ricordo ancora molto bene l'importanza che ha avuto per me di un video di auguri che tutti insieme mi avete mandato al mio primo compleanno lontano da casa. Grazie. Siete e sarete sempre parte della mia vita.

Ringrazio anche tutti i miei colleghi di Torino ed in particolare Matteo, che ha condiviso con me gli anni universitari. È stato davvero bello poter assistere insieme alle lezioni di Knafitz o fare tardi fino a dopo mezzanotte per i vari progetti che ci assegnavano. Insieme abbiamo sicuramente alleggerito le nostre esperienze universitarie e per me questo è stato fondamentale. Ti ringrazio.