



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

# Valutazione della degenerazione del morbo di Parkinson mediante modello di rete neurale convoluzionale

RELATRICI

Prof. Rita Francese

Dott.ssa Maria Frasca

Università degli studi di Salerno

CANDIDATO

Elpidio Mazza

Matricola: 0512106271

Anno Accademico 2021-2022

*Ai miei genitori, alle mie sorelle, per l'aiuto e i consigli preziosi.*

## Sommario

L'allungamento della durata media della vita, deve molto alle innumerevoli scoperte scientifiche e tecnologiche, le quali hanno permesso l'adozione di strumenti innovativi e sempre più affidabili. Internet è ormai il principale mezzo utilizzato per effettuare ricerche su temi di salute e sempre più persone dichiarano di utilizzare i canali digitali per comunicare con il medico di base, da ciò si può evincere che le tecnologie digitali sono sempre più utilizzate in ambito medico.

Tra queste tecnologie, quella che più si adatta a evitare libere interpretazioni, che potrebbero portare a diagnosi errate è l'apprendimento automatico, o Machine Learning. Queste tecniche possono essere applicate per risolvere un'ampia varietà di compiti. L'uso del ML in medicina si è diffuso negli ultimi anni, uno dei motivi è il miglioramento apportato dalla digitalizzazione del sistema di cartelle cliniche elettroniche (EHR) e dei vari dati, comprese le immagini mediche.

La malattia a cui è stata applicata tale tecnica, è il morbo di Parkinson, una malattia neuro-degenerativa, ad evoluzione lenta ma progressiva, che coinvolge principalmente, le funzionalità che controllano i movimenti e l'equilibrio.

In questo studio di tesi analizzeremo la costruzione di una rete neurale per riconosce in maniera automatica l'avanzamento del Parkinson, in quattro stadi, attraverso i modelli di ML. Il dataset a disposizione per questo lavoro di tesi, si compone di 32 pazienti dei quali 22 uomini e 10 donne, l'età media è di 59 anni e come età campionaria abbiamo un minimo di 39 anni un massimo di 72.

Il modello realizzato sarà in grado di apprendere direttamente dai dati le caratteristiche per effettuare la diagnosi, indicando lo stadio in cui si trova il paziente. Con la creazione del dataset, e l'addestramento del modello vedremo la disparità creatasi nei vari stadi della malattia, dove alcune classi avranno un minor numero di esempi di altre.

Il modello verrà poi migliorato sulla base di osservazioni fatte durante l'addestramento e l'analisi delle metriche, questo permetterà di impostare in maniera ottimale gli iperparametri, riuscendo a ottenere l'accuratezza finale del 94%.

Infine una piccola applicazione web sarà realizzata per poter utilizzare in maniera più agevole, a persone non esperte del settore informatico, il modello. La suddetta applicazione permetterà di ottenere la predizione di una risonanza caricandola come un'immagine.

<b>Indice</b>	<b>ii</b>
<b>Elenco delle figure</b>	<b>iv</b>
<b>Elenco delle tabelle</b>	<b>vi</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Struttura della tesi . . . . .	3
<b>2 Related work</b>	<b>4</b>
2.1 Parkinson . . . . .	4
2.1.1 Parkinson's Progression Markers Initiative . . . . .	4
2.1.2 Valutazione del morbo . . . . .	5
2.2 Modelli di deep learning usati . . . . .	6
2.2.1 Deep Learning . . . . .	6
2.2.2 Convolutional Neural Network . . . . .	7
2.2.3 Convolution Layer . . . . .	8
2.2.4 Max Pooling Layer . . . . .	8
2.2.5 Layer Normalization . . . . .	9
2.2.6 Flatten Layer . . . . .	10
2.2.7 Activation Functions . . . . .	10
2.2.8 Fully Connected Layer/Dense Layer . . . . .	11
2.2.9 Training and Test set . . . . .	11

2.2.10 Accuracy and Loss . . . . .	12
2.2.11 Confusion Matrix . . . . .	12
2.2.12 Data Augmentation . . . . .	13
2.2.13 Multiclass and Features . . . . .	13
2.3 Lavori correlati . . . . .	14
<b>3 Lavoro svolto</b>	<b>21</b>
3.1 Metodologia . . . . .	21
3.1.1 Preparazione dei dati . . . . .	21
3.1.2 Scelta dei range degli stadi . . . . .	22
3.1.3 Dataset . . . . .	23
3.1.4 Image pre-processing . . . . .	25
3.1.5 Scelta del Batch Size . . . . .	25
3.1.6 Scelta del numero di epochs . . . . .	26
3.1.7 Architettura del modello . . . . .	27
3.1.8 Metriche . . . . .	29
3.1.9 Applicazione . . . . .	30
3.2 Risultati . . . . .	31
<b>4 Conclusioni</b>	<b>32</b>
<b>Ringraziamenti</b>	<b>36</b>

---

## Elenco delle figure

---

1.1	Dati dei pazienti nel dataset . . . . .	2
1.2	Esempio di stratificazione di una risonanza magnetica del cervello . . . . .	2
1.3	Tabella contenente gli score relativi ad ogni visita, per ogni paziente . . . . .	3
2.1	Processo di convoluzione. . . . .	9
2.2	Processo di max pooling. . . . .	9
2.3	Processo di appiattimento. . . . .	10
2.4	Esempio di una rete con layer completamente connessi. . . . .	11
2.5	Struttura della confusion matrix. . . . .	13
2.6	Diverse feature map. . . . .	14
2.7	Results of grey matter SPM8 analysis. [1] . . . . .	15
2.8	Feature extraction based on the VBM analysis.. [2] . . . . .	15
2.9	Accuracy, Specificity and Sensitivity rates of SVM versus Number of PCA components in LOO validation [3] . . . . .	16
2.10	General framework of the proposed FSBi-LSTM method for AD diagnosis from MRI and PET neuroimages [4] . . . . .	17
2.11	The end-to-end network architecture of our proposed encoder-ST-ConvLSTM-decoder for tumor growth prediction. [5] . . . . .	18
2.12	Overall workflow of the proposed approach [6] . . . . .	19
2.13	Images generated during Augmentation. The images at the top are the original images of PD and HC, and the at the bottom is the synthetic images after (a, a') 50th iteration (b, b') 100th iteration (c, c') 150th iteration (d, d') 190th iteration [6]	20

3.1	Nuova organizzazione dl dataset . . . . .	22
3.2	Script di conversione da .DCM a .PNG . . . . .	23
3.3	Risultati funzione describe() della libreria pandas sulla colonna V15. . . . .	24
3.4	Snippet del dataset split. . . . .	24
3.5	Dataset rappresentante il numero di esempi presenti nelle varie classi. . . . .	25
3.6	Risultato delle immagini dopo le pre-elaborazioni . . . . .	26
3.7	Test con differenti batch size. . . . .	27
3.8	Analisi del training sul numero delle epoche. . . . .	28
3.9	Summary del modello. . . . .	29
3.10	Risultati della Confusion Matrix. . . . .	30
3.11	Viene mostrata una delle schermate dell'applicazione, in particolare quella che mostra il risultato della predizione. . . . .	30

---

**Elenco delle tabelle**

---

# CAPITOLO 1

---

## Introduzione

---

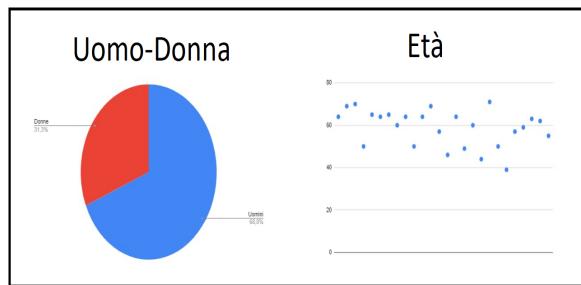
Le malattie neuro-degenerative sono patologie caratterizzate dalla perdita lenta e progressiva di una o più funzioni del sistema nervoso. La perdita delle cellule neuronali in determinate aree del cervello è graduale e irreversibile, ne conseguono disturbi del movimento o del funzionamento mentale. Una di queste malattie è il morbo di Parkinson che si caratterizza per il degrado dei neuroni localizzati nella substantia nigra (sostanza nera) del cervello. La diagnosi di questa patologia si basa sul riconoscimento dei sintomi e sulla valutazione degli esami strumentali (risonanze magnetiche o esami ematochimici). Le diagnosi precoci e l'assistenza preventiva possono aiutare i pazienti a migliorare la qualità della loro vita, sebbene il Parkinson non abbia cura.

Le reti neurali hanno il potenziale per estrarre le caratteristiche principali dai dati clinici e delle risonanze magnetiche. Questo permette di fornire un ulteriore supporto al personale medico che effettua una diagnosi.

L'obiettivo di questa tesi è quello di creare un modello di rete neurale, che possa stimare la progressione della malattia del Parkinson; partendo da una risonanza magnetica, il modello indicherà in quale fascia di progressione si trova il paziente.

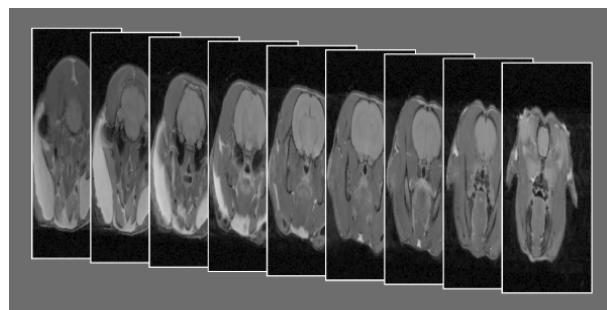
Il dataset messo a disposizione per questo lavoro di testi proviene da PPMI[7], si compone di 32 pazienti dei quali 22 uomini e 10 donne, l'età media è di 59 anni e come età campionaria abbiamo un minimo di 39 anni a un massimo di 72.

In particolare per ogni paziente c'è una lista di risonanze magnetiche, la risonanza magnetica è una tecnica diagnostica che sfrutta i campi magnetici per creare immagini



**Figura 1.1:** Dati dei pazienti nel dataset

tridimensionali dell'anatomia interna di una determinata area del corpo umano. Questi dati clinici sono in formato .DICOM uno standard che definisce i criteri per la comunicazione, la visualizzazione, l'archiviazione e la stampa di informazioni di tipo biomedico. Ogni risonanza si compone di 48 strati, ognuno dei quali rappresenta una determinata fetta del cervello.



**Figura 1.2:** Esempio di stratificazione di una risonanza magnetica del cervello

Oltre alle risonanze, viene fornito anche un altro file in formato .xlsx dove sono presenti gli score per ogni visita effettuata dal paziente e altri dati di tipo clinico:

- SC: Screening Visit - precedente alla visita di baseline dura circa otto ore.
- BL: Baseline Visit.
- V01-V15: sequenza di test programmati per tutti i pazienti.

SC	BL	V01	V02	V03	V04	V05	V06	V07	V08	V09	V10	V11	V12	V13	V14	V15
3102	30	36	42	36	33		66	70	76	85	85	85				
3105	46	45	50			66	75	80	85	111	122	132				
3107	31	31	31			32	32	32	33	33	33	33	33	33	47	47
3108	31	33	39			39	39	39	39	39	39	43	43	43	44	44
3111	36	36	36	36	42	45	56	60		60	63		63	71	71	71
3113	28	28	28			46	46	46	53	71	83	113				
3115	72	72	72			72	79	79	79	81	84	84	114	149	175	
3118	54	34	34	34	47	47	47	47	47	47	47	47	74	74	74	74
3119	11	40	42	43	43	55			55	55	57	62	62	62	62	62
3120	45	45	52	52	52	52	52	52	52	52	52	52	58	58		
3122	26	26	28		30	31	31	43	43	43	45	45	43	43	45	47
3123	15	17	21	22	22		36	37	38	38	38	38	38	38	41	41
3124	30	30	58			58	58	58	58	58	58	58	64	78	78	
3125	29	30	31	36		36	36	37	37	37	37	37	37	37	54	54
3126	20	39	43	43	55		55	57	43	50	56	64	64	71	71	
3127	19	19			20		31		31	31	31	31	31			
3128	28	28	28	40	40		40	40	40	40	62	62	63	63		
3130	13	14			14	14	14	20	20	24	24	26	32	44	46	46

**Figura 1.3:** Tabella contenente gli score relativi ad ogni visita, per ogni paziente

La metodologia utilizzata in questo lavoro di tesi, è suddivisa in varie fasi. Si è partiti identificando la funzione deep learning appropriata per gli obiettivi proposti, da questa analisi si è stati in grado di selezionare il framework che offrisse le funzionalità richieste, successivamente si è eseguita una fase di preparazione dei dati; infine l’addestramento e la validazione del modello che hanno portato alla registrazione dei risultati raggiunti.

I risultati previsti sono la realizzazione di un modello CNN, in grado di classificare un’immagine che rappresenta una risonanza magnetica del cervello, per stimare la progressione del morbo di parkinson, nel caso in cui fosse presente. Se la foto dovesse rappresentare altro, per la natura intrinseca dei modelli CNN, verrà comunque prodotto un risultato, non pertinente, appartenente alle classi di progressione.

## 1.1 Struttura della tesi

Nel capitolo 2 verrà descritta la "Parkinson’s Progression Markers Initiative", ovvero il morbo di Parkinson e la sua valutazione, l’architettura della rete neurale utilizzata e i lavori correlati.

Nel Capitolo 3 verranno descritte in dettaglio le scelte tecniche ed implementative con i relativi risultati ottenuti.

# CAPITOLO 2

---

## Related work

---

### 2.1 Parkinson

Il Parkinson è una malattia neurodegenerativa, ad evoluzione lenta ma progressiva, che coinvolge, principalmente, alcune funzioni quali il controllo dei movimenti e dell'equilibrio. La malattia fa parte di un gruppo di patologie definite "Disordini del Movimento" e tra queste è la più frequente. I sintomi del Parkinson sono forse noti da migliaia di anni: una prima descrizione sarebbe stata trovata in uno scritto di medicina indiana che faceva riferimento ad un periodo intorno al 5.000 A.C. ed un'altra in un documento cinese risalente a 2.500 anni fa. Il nome è legato però a James Parkinson, un farmacista chirurgo londinese del XIX secolo, che per primo descrisse gran parte dei sintomi della malattia in un famoso libretto, il "Trattato sulla paralisi agitante".

La malattia è presente in tutto il mondo ed in tutti i gruppi etnici. Si riscontra in entrambi i sessi, con una lieve prevalenza, forse, in quello maschile. L'età media di esordio è intorno ai 58-60 anni, ma circa il 5 % dei pazienti può presentare un esordio giovanile tra i 21 ed i 40 anni. Prima dei 20 anni è estremamente rara. Sopra i 60 anni colpisce 1-2% della popolazione, mentre la percentuale sale al 3-5% quando l'età è superiore agli 85.

#### 2.1.1 Parkinson's Progression Markers Initiative

L'iniziativa "Parkinson's Progression Markers Initiative" (PPMI) [7] è uno studio clinico basato sulla raccolta ed osservazione di determinati dati medici, per valutare in modo

completo le coorti di interesse significative utilizzando l'imaging avanzato, il campionamento biologico e le valutazioni cliniche e comportamentali per identificare i biomarcatori della progressione della malattia di Parkinson che potrebbero offrire ai ricercatori uno strumento essenziale per la ricerca di terapie in grado di rallentare o arrestare la progressione del morbo, o tentare di anticipare la diagnosi della malattia.

L'obiettivo principale delle ricerche terapeutiche riguardanti il morbo di Parkinson, è di individuare possibili biomarcatori, validabili dalla comunità scientifica, in quanto permetterebbero di velocizzare le ricerche mediche tese ad individuare criteri certi di diagnosi e cura. Attualmente i biomarcatori individuati non sono né ottimali, né è stata confermata la loro validità certa.

La PPMI si svolge presso i siti clinici negli Stati Uniti, in Europa, Israele ed Australia ed è resa possibile dagli sforzi di una collaborazione di ricercatori, finanziatori, governi, centri medici e volontari ma, in particolar modo, è sponsorizzata da "The Michael J.Fox Foundation for Parkinson's Research", una fondazione creata nel 2000 dall'omonimo attore Michael J.Fox, affetto da tale patologia da 28 anni.

L'attività condotta dalla PPMI è uno studio "open source", i dati e i campioni raccolti e acquisiti dai partecipanti volontari, affetti e non dalla malattia, permetteranno lo sviluppo di un database e di una biorepository completa, che è attualmente disponibile online e aggiornata ogni otto mesi. Trattandosi di dati raccolti da pazienti sparsi per il globo uno dei compiti principali della PPMI è quello di coordinare la gestione dei vari dati, definendo un protocollo per la raccolta e codifica dei dati. Il repository elaborato può essere scaricato mediante l'accesso al portale del sito del PPMI per permettere alla comunità scientifica di condurre ricerche complete ed esaustive.

### 2.1.2 Valutazione del morbo

La diagnosi della malattia di Parkinson resta tuttora una diagnosi clinica poiché non esiste un test obiettivo o dei marcatori biochimici e neuroradiologici specifici.

Nell'ultimo decennio, però, uno degli obiettivi della ricerca è stato migliorare la specificità dei criteri diagnostici classici: infatti la "United Kingdom Parkinson's disease Society Brain Bank" ha proposto criteri clinici che tuttora sono ampiamente utilizzati nella pratica clinica e nei protocolli di ricerca. Tali criteri diagnostici stabiliscono che il segno necessario per porre diagnosi della malattia del Parkinson sia la bradicinesia o acinesia, associata ad almeno uno degli altri segni cosiddetti maggiori, accennati precedentemente, ovvero la rigidità muscolare, tremore a riposo e l'instabilità posturale.

Tali criteri diagnostici sono stati recentemente rivisti da Gelb, Oliver e Gilman nel libro “Diagnosi Criteria for Parkinson’s Disease” dove sottolineano come la diagnosi clinica sia basata sulla combinazione di alcuni segni motori “cardinali” e sull’esclusione di sintomi ritenuti “atipici”.

In conclusione, i sintomi del morbo di Parkinson si manifestano in modo diverso nei diversi pazienti, i quali possono sperimentare alcuni sintomi e non altri e anche il ritmo con cui la malattia progredisce varia da individuo a individuo. Per questo, il tasso di diagnosi errata può essere relativamente alto.

## 2.2 Modelli di deep learning usati

### 2.2.1 Deep Learning

L’intelligenza artificiale può essere un potente strumento al servizio della medicina moderna, in alcuni campi già fa parte delle pratiche in uso. Applicazioni del genere, infatti, sono già impiegate in vari ambiti, dalla diagnostica alla chirurgia, dallo sviluppo dei farmaci alla riabilitazione.

Molto importante, poi, è l’applicazione dell’AI nella diagnostica predittiva, attraverso l’utilizzo e l’interpretazione dei dati, si possono cogliere i primi segnali di alcune malattie per aiutare i medici a fare diagnosi più accurate, con l’obiettivo di ridurre gli errori e sviluppare metodi per il trattamento medico individualizzato.

Il deep learning è quel campo di ricerca dell’apprendimento automatico che si basa su diversi livelli di rappresentazione che corrispondono a determinate caratteristiche, dove i concetti di alto livello (ovvero quelli che noi conosciamo) vengono definiti sulla base dell’unione di tutte le caratteristiche che lo formano.

Complicati algoritmi imparano a riconoscere determinate caratteristiche attraverso un’architettura di rete neurale, organizzate in diversi strati, dove ogni strato calcola i valori per quello successivo affinché l’informazione venga elaborata in maniera sempre più completa.

Questo ramo dell’apprendimento automatico consiste, in modelli ispirati al funzionamento del cervello umano.

Con una quantità sufficiente di dati, il sistema è in grado di apprendere la rappresentazione corretta e di risolvere problemi di apprendimento automatico senza la necessità di un pre-processamento dei dati, come invece avviene per le tecniche tradizionali di Machine Learning.

In altre parole, il Deep Learning è una tecnica di apprendimento in cui alle reti neurali artificiali vengono comunicati una vaste quantità di dati, in modo che possano imparare a svolgere vari compiti.

### 2.2.2 Convolutional Neural Network

Una rete neurale convoluzionale CNN o ConvNet è una classe di rete neurale artificiale, più comunemente utilizzata per analizzare le immagini visive. Le CNN sono anche conosciute come SIANN (Shift Invariant o Space Invariant Artificial Neural Networks), basate sull'architettura a peso condiviso dei kernel o filtri, che scorrono lungo l'input e forniscono mappe delle caratteristiche.

La maggior parte delle reti neurali convoluzionali sono soggette alla traslazione, a causa dell'operazione di downsampling (operazioni che riducono il numero di parametri come il layer max pooling o averaging layer) che applicano all'input.

Le CNN sono versioni regolarizzate di perceptron multistrato. I perceptron multistrato, reti completamente connesse, ovvero ogni neurone in uno strato è connesso a tutti i neuroni nello strato successivo. La "connettività completa" di queste reti le rende soggette a overfitting. Tipici metodi di regolarizzazione, o prevenzione dell'overfitting, includono: penalizzare i parametri durante l'allenamento (come il calo del peso) o tagliare la connettività (connessioni saltate, dropout, ecc.).

Le reti convoluzionali sono state ispirate da processi biologici in quanto il modello di connettività tra i neuroni ricorda l'organizzazione della corteccia visiva animale. I singoli neuroni corticali rispondono agli stimoli solo in una regione ristretta del campo visivo nota come campo recettivo. I campi ricettivi di diversi neuroni si sovrappongono parzialmente in modo tale da coprire l'intero campo visivo.

Le CNN utilizzano una pre-elaborazione relativamente ridotta rispetto ad altri algoritmi di classificazione delle immagini. Ciò significa che la rete impara a ottimizzare i filtri attraverso l'apprendimento automatico.

Una rete neurale convoluzionale è costituita da uno strato di input, strati nascosti e uno strato di output. In qualsiasi rete neurale feed-forward, tutti i livelli intermedi sono chiamati nascosti perché i loro input e output sono mascherati dalla funzione di attivazione e dalla convoluzione finale. I livelli nascosti includono livelli che eseguono convoluzioni, in genere questo include un livello che esegue un prodotto scalare del kernel di convoluzione con la matrice di input del livello. Questo prodotto è solitamente l'interno di Frobenius e la sua funzione di attivazione è comunemente ReLU. Quando il kernel di convoluzione scorre

lungo la matrice di input del suo livello, l'operazione di convoluzione genera una mappa delle caratteristiche, che a sua volta contribuisce all'input del livello successivo. Questo è seguito da altri livelli come livelli di pooling, livelli completamente connessi e livelli di normalizzazione.

### 2.2.3 Convolution Layer

Il layer CONV è il blocco principale di queste reti, utilizzando dei filtri che altro non sono che delle matrici di valori, chiamati anche pesi, estraggono delle mappe di caratteristiche dai dati in input.

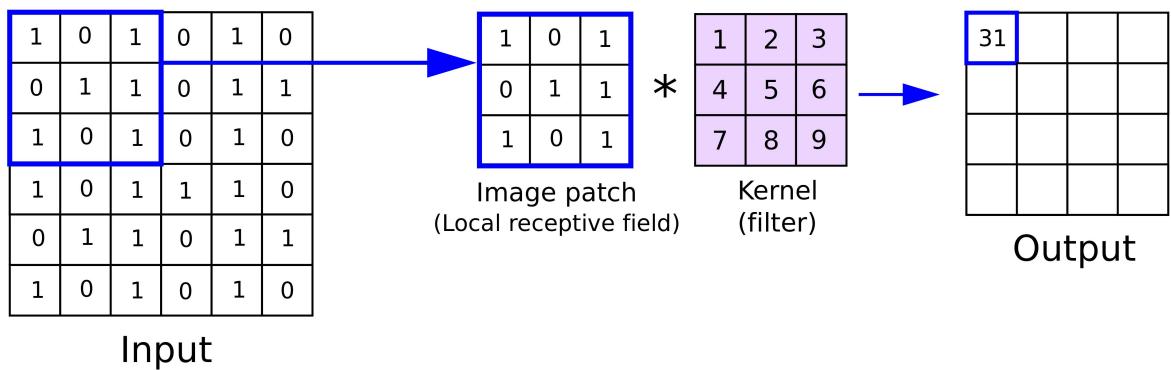
Uno strato convoluzionale contiene una serie di filtri i cui parametri devono essere appresi. L'altezza e il peso dei filtri sono inferiori a quelli del volume di ingresso. Ogni filtro è convogliato con il volume di input per calcolare una mappa di attivazione fatta di neuroni.

In altre parole, il filtro viene fatto scorrere lungo la larghezza e l'altezza dell'input e i prodotti scalari tra l'input e il filtro vengono calcolati in ogni posizione spaziale. Il volume di uscita dello strato convoluzionale si ottiene impilando le mappe di attivazione di tutti i filtri lungo la dimensione di profondità. Poiché la larghezza e l'altezza di ciascun filtro sono progettate per essere inferiori all'input, ogni neurone nella mappa di attivazione è connesso solo a una piccola regione locale del volume di input.

La dimensione del campo ricettivo di ciascun neurone è piccola ed è uguale alla dimensione del filtro. La connettività locale è motivata dall'architettura della corteccia visiva animale, dove i campi ricettivi delle cellule sono piccoli. La connettività locale dello strato convoluzionale consente alla rete di apprendere filtri che rispondono al massimo a una regione locale dell'input, sfruttando così la correlazione spaziale locale dell'input (ovvero per un'immagine di input, un pixel è più correlato ai pixel vicini che a i pixel distanti). Inoltre, poiché la mappa di attivazione si ottiene eseguendo la convoluzione tra il filtro e l'ingresso, i parametri del filtro sono condivisi per tutte le posizioni locali. La condivisione del peso riduce il numero di parametri per l'efficienza dell'espressione, l'efficienza dell'apprendimento e la buona generalizzazione.

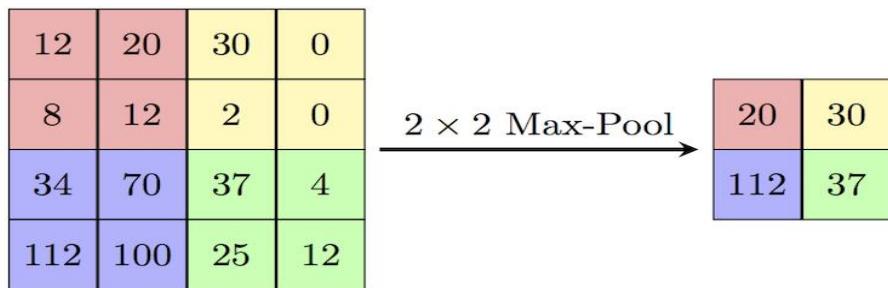
### 2.2.4 Max Pooling Layer

Uno strato di pooling è solitamente incorporato tra due strati convoluzionali. La sua funzione è di ridurre progressivamente la dimensione spaziale della rappresentazione, per ridurre la quantità di iper parametri e di conseguenza controllare l'overfitting. Il Pooling

**Figura 2.1:** Processo di convoluzione.

Layer opera in modo indipendente su ogni fetta di profondità dell'input e la ridimensiona spazialmente, utilizzando l'operazione MAX o AVERAGE.

Il livello di pooling riassume le caratteristiche presenti in una regione della mappa generata da un livello di convoluzione, questo meccanismo permette al modello di diventare più robusto alle variazioni nella posizione delle caratteristiche nell'immagine in input. L'operazione di pooling massimo, seleziona l'elemento massimo della regione della mappa coperta dal filtro. Pertanto, l'output dopo il livello max pooling sarebbe una mappa delle caratteristiche più importanti rispetto alla mappa precedente. Il pooling massimo è comunemente usato in quanto funziona meglio.

**Figura 2.2:** Processo di max pooling.

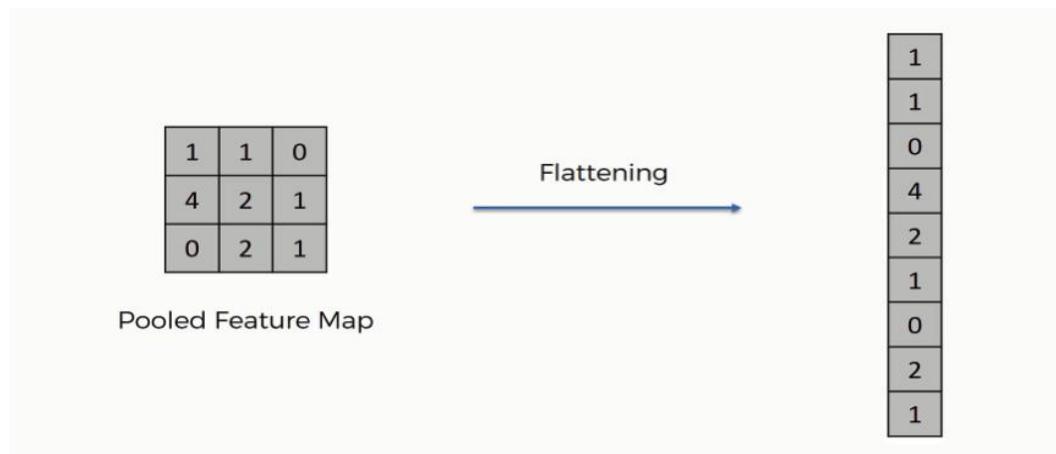
## 2.2.5 Layer Normalization

Questo strato è usato in ingresso per il ridimensionamento delle caratteristiche, e nella batch normalization agli strati nascosti, scala l'input in modo che l'output abbia una media prossima allo zero e una deviazione standard unitaria, per un addestramento efficiente. Applica una funzione di attivazione elementare, come la soglia max ( $0, x$ ) a 0. Tutti i pesi unici

dei neuroni sono moltiplicati per l'input e il loro totale è poi simulato attraverso la funzione di attivazione per controllare se è più o meno della soglia. Se è superiore alla soglia, allora restituisce un 1, altrimenti 0.

### 2.2.6 Flatten Layer

Flatten è lo strato subito dopo la serie di strati di convoluzione e pooling consecutivi. Gli strati di convoluzione producono mappe di attivazione tridimensionali in modo tale che solo l'output è necessario per trovare se un'immagine appartiene ad una particolare classe. Questo strato converte le mappe di caratteristiche 3D in vettori di caratteristiche 1D.



**Figura 2.3:** Processo di appiattimento.

### 2.2.7 Activation Functions

Una funzione di attivazione in una rete neurale definisce come la somma ponderata dell'input viene trasformata in un output da uno o più nodi in uno strato della rete.

La scelta della funzione di attivazione ha un grande impatto sulla capacità e sulle prestazioni della rete neurale e diverse funzioni di attivazione possono essere utilizzate in diverse parti del modello. Tecnicamente, la funzione di attivazione viene utilizzata all'interno o dopo l'elaborazione interna di ciascun nodo della rete, sebbene le reti siano progettate per utilizzare la stessa funzione di attivazione per tutti i nodi in un livello.

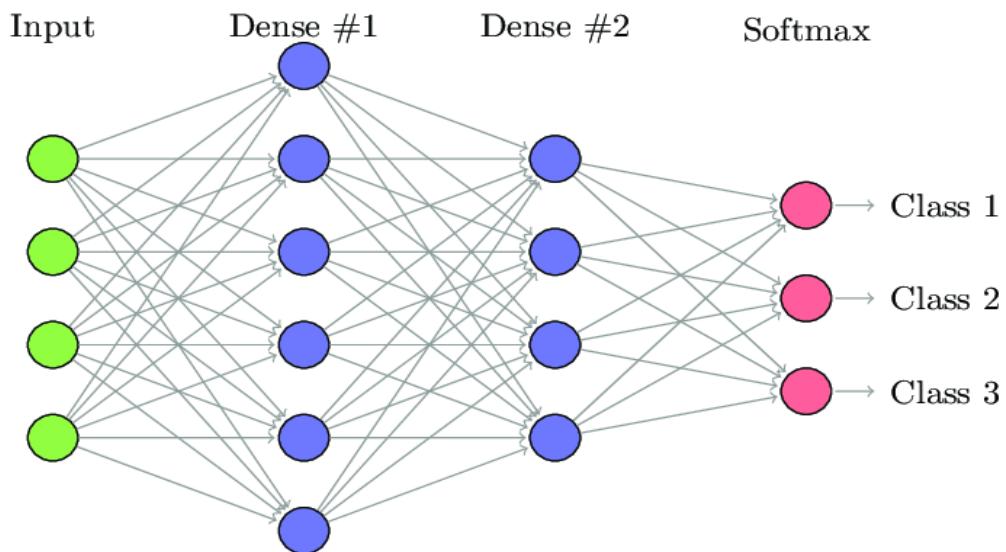
Una rete può avere tre tipi di livelli: livelli di input che prendono l'input grezzo dal dominio, livelli nascosti che prendono l'input da un altro livello e passano l'output a un altro livello e livelli di output che fanno una previsione. Tutti i livelli nascosti utilizzano in genere la stessa funzione di attivazione. Il livello di output utilizzerà in genere una funzione di attivazione diversa dai livelli nascosti e dipende dal tipo di previsione richiesta dal modello.

La funzione di attivazione ReLU (Rectified Linear Unit) è una funzione di attivazione semplicemente definita come  $f(\max) = \max(0, z)$ , la funzione è 0 per i valori negativi e crescente linearmente per valori positivi. La funzione è molto semplice, genera buoni risultati ed è ampiamente utilizzata per migliorare l’addestramento di reti profonde.

La funzione di attivazione sigmoid è anche chiamata funzione logistica. La funzione prende qualsiasi valore reale come input ed emette valori compresi nell’intervallo da 0 a 1. Più grande è l’input (più positivo), più il valore di output sarà vicino a **1,0**, mentre più piccolo è l’input (più negativo), più vicino sarà l’output sarà **0,0**.

### 2.2.8 Fully Connected Layer/Dense Layer

Il fully connected layer chiamato anche strato denso (o strato completamente connesso), è lo strato in cui i neuroni sono tutti collegati ai neuroni dello stato precedente. Questo permette di integrare le informazioni provenienti dagli strati precedenti rendendoli disponibili a tutti gli strati successivi. Questo porta il vantaggio che se l’input verrebbe rilevato in una sezione diversa, la rete si renderebbe conto degli spostamenti, della scala e della rotazione.



**Figura 2.4:** Esempio di una rete con layer completamente connessi.

### 2.2.9 Training and Test set

Un dataset di training è un agglomerato di esempi di dati usati per il processo di apprendimento, viene usato per adattare i parametri di un classificatore. Per la classificazione, un algoritmo di apprendimento supervisionato esplora i dati del dataset per determinare le

combinazioni ottimali di variabili che generano un buon modello predittivo. L'obiettivo è quello di produrre un modello che generalizzi i dati sconosciuti.

Un dataset di testing viene utilizzato per valutare il modello dopo la sua fase di addestramento. Il dataset di test contiene dati che variano sulle varie classi, che il modello dovrebbe affrontare, questi dati sono sconosciuti al modello in quanto non sono appartenenti al dataset di training.

### 2.2.10 Accuracy and Loss

L'accuratezza di un algoritmo di classificazione è un modo per misurare le prestazioni di un modello di classificazione. Di solito espressa in percentuale, ed è data dal numero di risposte esatte sul totale di tutti i dati predetti.

La funzione di loss, nota anche come la funzione di costo, tiene conto delle probabilità o dell'incertezza di una previsione in base a quanto la previsione varia dal valore reale.

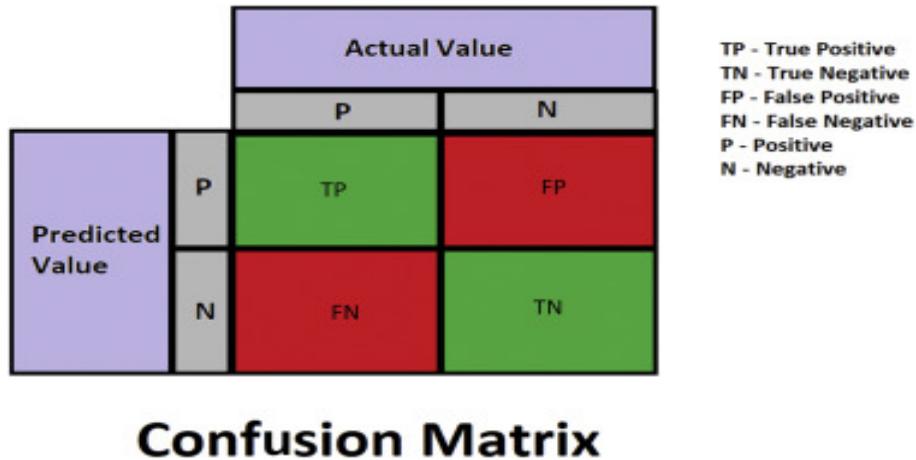
A differenza dell'accuratezza, la funzione di loss (o perdita) non è una percentuale ma una somma degli errori commessi per ciascun campione nelle fasi di addestramento e convalida. La perdita viene spesso utilizzata nel processo di addestramento, per individuare i valori "migliori" dei parametri e per aggiustare i pesi del modello.

Durante il processo di addestramento l'obiettivo è ridurre al minimo questo valore. A differenza dell'accuratezza, la perdita può essere utilizzata sia nei problemi di classificazione che di regressione.

La maggior parte delle volte vedremo che la precisione aumenta con la diminuzione della perdita, ma non sempre è così. Infatti precisione e perdita hanno definizioni diverse e misurano cose diverse ma non esiste una relazione matematica tra queste due metriche.

### 2.2.11 Confusion Matrix

La matrice di confusione, nota anche come la matrice di errore, è un layout di tabella che consente di visualizzare le prestazioni di un algoritmo. Ogni riga della matrice rappresenta le istanze di una classe effettiva, mentre ogni colonna rappresenta le istanze di una classe prevista, o viceversa. Il nome deriva dal fatto che rende facile vedere se il sistema sta confondendo due classi (cioè etichettando erroneamente una come un'altra).



**Figura 2.5:** Struttura della confusion matrix.

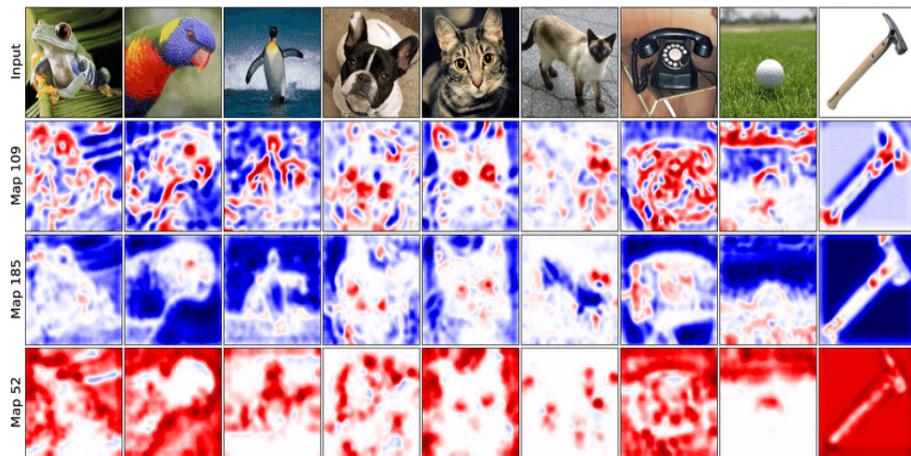
### 2.2.12 Data Augmentation

Per ottenere una buona accuratezza delle previsioni, i modelli di deep learning hanno bisogno di una grande quantità di dati e della diversità degli stessi, per ottenere buoni risultati dalla fase di training. Un modo per affrontare il problema dei dati limitati è quello di applicare diverse trasformazioni sui dati disponibili, per sintetizzare di nuovi. Questo metodo è applicabile anche nel caso in cui c'è uno squilibrio dei dati, ovvero le varie classi differiscono di un numero sostanziale di dati.

### 2.2.13 Multiclass and Features

Nell'apprendimento automatico, la classificazione multiclasse o multinomiale è il problema di classificare le istanze di una o più classi. Il classificatore preso un input ipotizzerà a quale classe appartiene.

Una feature è una parte d'informazione contenuta in un dato. Tipicamente nel caso delle immagini corrispondono a certe regioni, che possono essere delle strutture specifiche come dei punti, bordi o sagome di oggetti. Il concetto di feature è molto generale poiché la scelta di quest'ultime dipende dal problema che si sta affrontando.



**Figura 2.6:** Diverse feature map.

## 2.3 Lavori correlati

Questo sezione illustrerà lo stato dell'arte e i lavori presenti in letteratura sugli aspetti di ricerca trattati nel nostro studio.

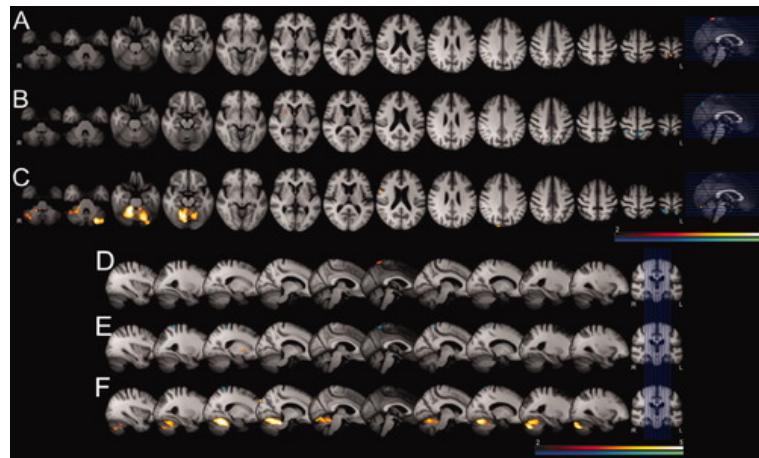
Sono stati presi in esame lavori sul Parkinson e altri studi di malattie neuro-degenerative come l'Alzheimer. Tutti gli studi sono incentrati nel predire la progressione di una malattia neuro-degenerativa attraverso l'utilizzo di una rete neurale.

Le varie implementazioni di queste reti neurali servono a focalizzare l'attenzione, di un determinato dato analizzato. Le varie tipologie di reti neurali utilizzate in questi studi spaziano da 2D, 3D e 4D, quest'ultima prende anche in considerazione il fattore temporale.

Tutti i dataset dei lavori presi in considerazione presentano pochi dati per un adeguato training del modello, per rimediare a questo problema alcuni dei lavori attuano tecniche di "data augmentation", un metodo che consiste nell'aumentare la dimensione del dataset a disposizione, senza effettivamente raccogliere nuovi elementi.

Un esempio, di "data augmentation" che viene applicata ai dati già presenti nel dataset è la rotazione dell'immagine.

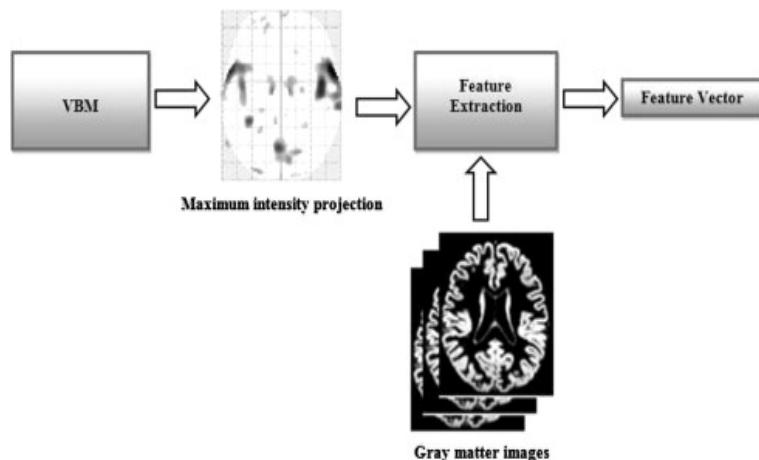
Fock et al. [1] hanno sviluppato un sistema di classificazione che utilizza immagini RM per diagnosticare il morbo di Parkinson. Gli autori di questo articolo hanno utilizzato varie regioni del cervello, come la materia bianca e la materia grigia; il processo si basava sull'utilizzo di un vettore per la diagnosi del Parkinson Disease (PD) contenente dati di natura clinica dell'individuo. Per la pre-elaborazione dei dati è stata utilizzata la morfometria basata su voxel (VBM). Tuttavia, la precisione della classificazione ottenuta oscilla tra 39,53% e 41,86% (questo lavoro è stato svolto su un dataset contenente i dati di 64 pazienti).



**Figura 2.7:** Results of grey matter SPM8 analysis. [1]

Babu et al.[2] hanno proposto un sistema diagnostico assistito da computer, per effettuare la diagnosi del morbo di Parkinson. Il loro approccio prevede due fasi: estrazione delle caratteristiche e classificazione. Per la prima fase, viene utilizzato il VBM per creare dati funzionali per la materia grigia (GM). Mediante l'utilizzo del modello RFE si è andato a eliminare le caratteristiche ricorsive, permettendo di selezionare quelle più discriminatorie. La classificazione viene eseguita nella seconda e ultima fase del sistema proposto, utilizzando l'apprendimento basato sulla proiezione e sulla funzione metacognitiva radiale, con l'accuratezza dell'81,21%.

Lo svantaggio di questo lavoro è dovuto alla sensibilità della VBM che è causata da disallineamento delle strutture cerebrali, da differenze nei modelli di piegatura delle RM e dallo spessore corticale. [8] Tutto ciò può confondere l'analisi statistica e diminuire la sensibilità dei veri effetti volumetrici o aumentare la possibilità di falsi positivi.



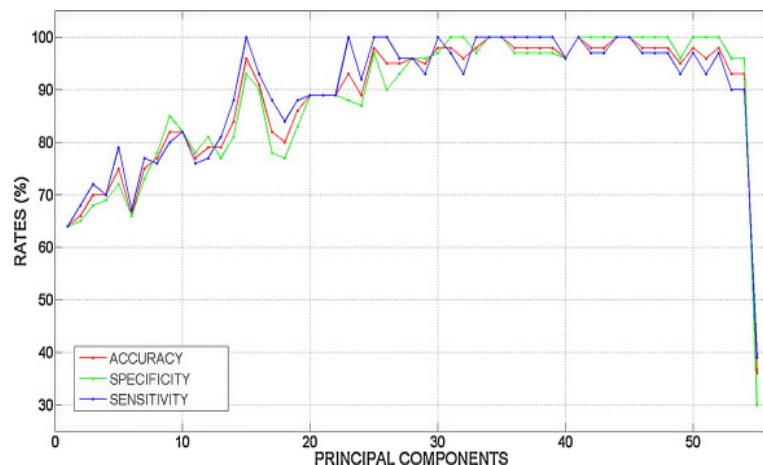
**Figura 2.8:** Feature extraction based on the VBM analysis.. [2]

Salvatore et al. [3] hanno valutato la fattibilità di un algoritmo di apprendimento automatico supervisionato, per la diagnosi assistita di pazienti con malattia di Parkinson (PD) e "paralisi sopra-nucleare progressiva" (PSP) clinicamente diagnosticata.

Il dataset di lavoro si basava su immagini morfologiche di risonanze magnetiche (MRI) di pazienti con PD (28), pazienti con PSP (28) e altri 28 soggetti sani.

Il modello utilizzato estraeva le caratteristiche dai componenti analizzati dalla PCA, utilizzando come ulteriore layer, una SVM (Support Vector Machines). L'algoritmo è stato in grado di ottenere biomarcatori morfologici basati su voxel di PD e PSP dalle immagini MR dopo un processo di normalizzazione.

I risultati mostrano una accuratezza della diagnosi, nel determinare PD e PSP, con una sensibilità maggiore del 90%. In conclusione questo lavoro svolto fornisce una classificazione dei pazienti con PD dai pazienti con PSP.



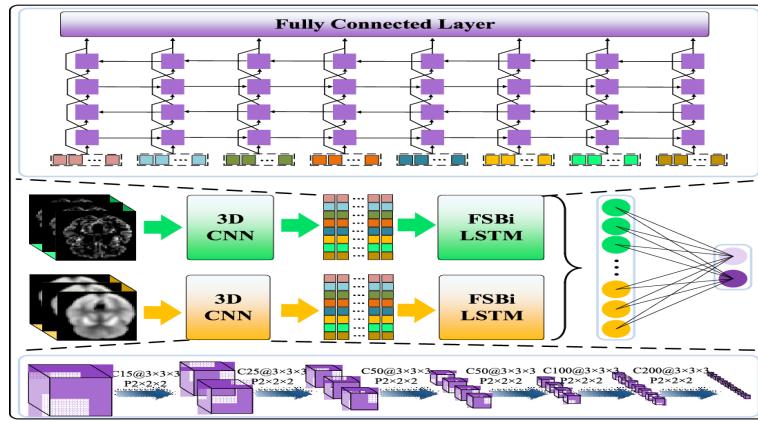
**Figura 2.9:** Accuracy, Specificity and Sensitivity rates of SVM versus Number of PCA components in LOO validation [3]

Chiyu Feng, Ahmed Elazab et al [4], descrivono come lo sviluppo del Deep Learning e delle reti neurali convoluzionali (CNN), hanno fatto grandi progressi nel riconoscimento delle immagini utilizzando la risonanza magnetica (MRI).

Tuttavia, a causa della limitata disponibilità di questi dati, è ancora impegnativo utilizzare efficacemente le CNN per la diagnosi di Alzheimer. A tal fine, viene progettato un nuovo modello di apprendimento basato sull'analisi di fattori temporali e immagini 3D.

In particolare, il layer 3D-CNN sfrutta il protrarsi della memoria a breve termine, con comunicazione bidirezionale singolarmente impilata (FSBi-LSTM).

Grazie ad un'architettura 3D-CNN viene ricavata una rappresentazione profonda delle caratteristiche delle risonanze (MRI).



**FIGURE 1.** General framework of the proposed FSBi-LSTM method for AD diagnosis from MRI and PET neuroimages. C is convolutional layer, the P is mean pooling layer, @ is the number of filters such as  $15@3 \times 3 \times 3$  is 15 filters which size are  $3 \times 3 \times 3$  and  $P2 \times 2 \times 2$  is pooling layers, which size are  $2 \times 0 \times 2$ .

**Figura 2.10:** General framework of the proposed FSBi-LSTM method for AD diagnosis from MRI and PET neuroimages [4]

Si utilizza una rete 3D-CNN per estrarre le caratteristiche primarie di entrambe le tipologie di dati clinici, MRI e PET, con l’ausilio del modello FSBi-LSTM per estrarre informazioni semantiche e spaziali di alto livello dell’output della 3D-CNN. Si è in grado di definire le caratteristiche che verranno passate al classificatore SoftMax per la diagnosi della malattia. Infine vengono confrontate le prestazioni con altri modelli di deep learning correlati.

Il dataset utilizzato proviene da Alzheimer’s Disease Neuroimaging Initiative (ADNI) [4] e contiene i dati clinici di pazienti affetti da Alzheimer.

Liu et al.[5] nel loro studio hanno utilizzato una rete 2D-CNN, sono riusciti ad estrarre le caratteristiche dalle RM dividendo le immagine in fette. Queste caratteristiche sono state poi date in input a un layer SBi-GRU che ricostruendo l’immagine da una classificazione finale.

Il metodo proposto supera questo approccio poiché la 3D-CNN può conservare più informazioni spaziali rispetto alle fette 2D senza perdita di informazioni.

Liu et al.[5] nel loro studio hanno utilizzato una rete 2D-CNN, sono riusciti ad estrarre le caratteristiche dalle RM dividendo le immagine in fette. Queste caratteristiche sono state poi date in input a un layer SBi-GRU che ricostruendo l’immagine da una classificazione finale.

Il metodo proposto supera l’approccio di un modello che fa uso di un 3D-CNN, quest’ultimo può conservare più informazioni spaziali rispetto alle fette di un 2D ma con una possibile perdita di informazioni.

Per Liu et al.[4] la FSBi-LSTM con scansioni progressive si è rivelata più efficace della convoluzione diretta, usando un kernel di convoluzione 2D per identificare le caratteristiche informative.

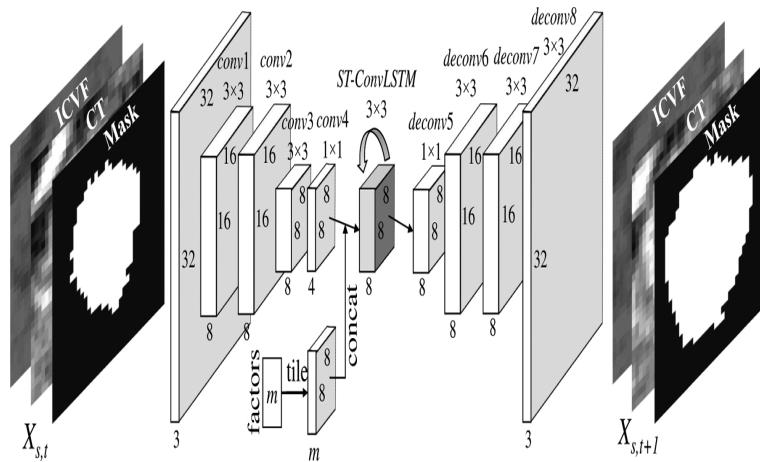
Dei recenti progressi sulle reti convoluzionali (ConvNets), si sono occupati Ling Zhang, Le

Lu et al [4], sono riusciti a dimostrare che è possibile ottenere una maggiore accuratezza con i ConvNets rispetto ai modelli matematici tradizionali nel predire i futuri volumi tumorali.

Gli approcci di modellazione basati su patch di immagini 2D, non possono fare pieno uso del contesto di imaging spazio-temporale, non fruttando al massimo i dati longitudinali 4D (3D + tempo) del paziente.

Nell'articolo, si fa utilizzo del modello convoluzione Long Short Term Memory (ConvLSTM), dove viene formulato il processo di crescita del tumore; la rete estraе le apparenze statiche del tumore e simultaneamente cattura i suoi cambiamenti dinamici temporali.

Il dataset è composto da trentatré pazienti (tredici maschi e venti femmine) raccolti dal von Hippel-Lindau presso i National Institutes of Health.



**Figura 2.11:** The end-to-end network architecture of our proposed encoder-ST-ConvLSTM-decoder for tumor growth prediction. [5]

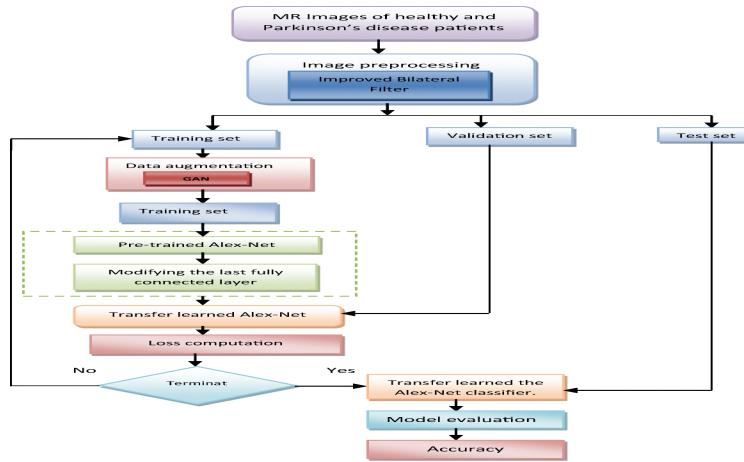
Sukhpal Kaur, Himanshu Aggarwal Rinkle Rani [6] nel loro studio della diagnosi del morbo di Parkinson, utilizzano una CNN con transfer learning per la classificazione della malattia del Parkinson.

La figura successiva mostra il flusso di lavoro complessivo dell'approccio proposto. Il dataset si compone di immagini RM di persone sane e pazienti con malattia di Parkinson raccolte da PPMI [7].

Inizialmente viene eseguita una fase di pre-elaborazione delle immagini RM, per adattarle alla varietà dei possibili diversi formati; in questo modo si riducono alcuni disturbi.

Successivamente si esegue una fase di data augmentation, per affrontare il problema della dimensione limitata del dataset e per migliorare le prestazioni dell'approccio proposto.

Il modello è un implementazione del transfer learning, su layer Alex-Net in precedenza pre-addestrato, dove gli ultimi strati vengono utilizzati per categorizzare nuove tipologie di



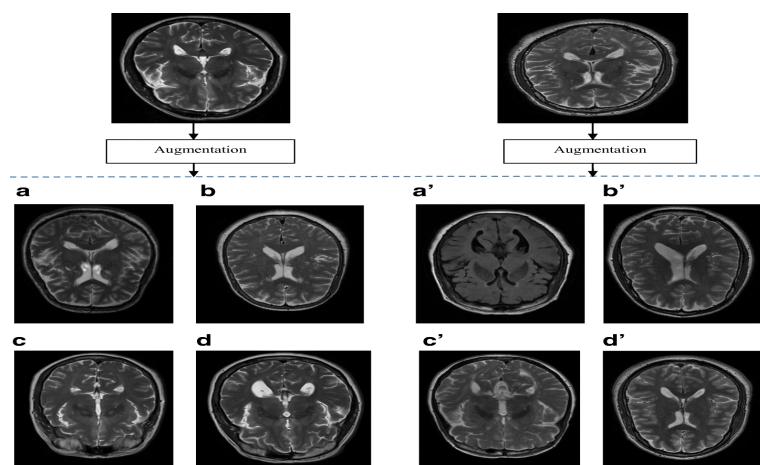
**Figura 2.12:** Overall workflow of the proposed approach [6]

immagini. Infine, le prestazioni del modello proposto vengono valutate su immagini di test RM di pazienti con PD e senza.

Il passaggio più critico è nel processo di pre-elaborazione dove si ha come obiettivo l'estrazione delle caratteristiche per un buon livello di classificazione finale. Le immagini MR contengono numerose variabilità, e per ridurre al minimo tale effetto che influisce negativamente sul processo Deep CNN, viene eseguita la normalizzazione prendendo la differenza tra i valori di intensità dell'immagine MR considerando L1 come valore di intensità minima e L2 come valore di intensità risultante come  $L = L2 - L1$ .

Con l'architettura Alex-Net nel livello di input la dimensione dell'immagine deve essere  $277 \times 277 \times 3$ ; di conseguenza alle immagini viene applicata un ridimensionamento per soddisfare le richieste del layer Alex-Net.

Per la mancanza di un gran numero di immagini etichettate, viene utilizzata la tecnica di Generative Adversarial Networks (GANs) [9] un sistema generativo composto da due reti opposte: una rete di generatori e una rete di discriminatori. Il generatore genera dati sintetici per ingannare il discriminatore, mentre il discriminatore separa i dati reali da quelli sintetici.



**Figura 2.13:** Images generated during Augmentation. The images at the top are the original images of PD and HC, and the at the bottom is the synthetic images after (a, a') 50th iteration (b, b') 100th iteration (c, c') 150th iteration (d, d') 190th iteration [6]

# CAPITOLO 3

---

## Lavoro svolto

---

### 3.1 Metodologia

In questo lavoro di tesi, viene proposta una possibile soluzione alla complessa classificazione del Morbo di Parkinson. Dopo gli studi effettuati e una prima analisi dei dati, l'architettura scelta è stata quella di un modello convoluzionale. Questo ha permesso che la rete apprendesse direttamente dai dati, che in questo caso si presentano come immagini, eliminando la necessità di estrarre manualmente le feature dalle complesse risonanze magnetiche; la struttura della rete neurale sarà illustrata in seguito.

La scelta del framework da utilizzare per la creazione del modello è stato quello di TensorFlow, uno dei framework più popolari in quest'ambito, sviluppato da tam Google Brain, noto soprattutto per la sua documentazione e il supporto della community, il framework supporta il linguaggio Python, un linguaggio orientato agli oggetti conosciuto per sua chiarezza, potenza e flessibilità.

#### 3.1.1 Preparazione dei dati

Nella prima fase sono state eseguite alcune azioni di preparazione dei dati. Il dataset per come si presentava inizialmente, ha subito una fase di ristrutturazione a causa di cartelle e dati ridondanti, che rendevano difficile e poco efficiente l'analisi strutturale e componentistica. Dopo la riorganizzazione, ogni paziente presenta una sua cartella principale, avente le risonanze magnetiche e un file '.xlsx', all'interno di questo file sono elencate gli ID delle

cartelle delle RM e i dati del paziente come il sesso, l'età in cui ha effettuato la visita e la data di acquisizione. Applicando questa metodologia all'organizzazione del dataset ho potuto semplificare il caricamento del dataset mediante script automatizzato.



**Figura 3.1:** Nuova organizzazione dl dataset

Nelle cartelle delle risonanze non erano presenti semplici immagini, infatti, nella computer vision spesso si lavora con immagini mediche nel formato DICOM. L'immagine Dicom è più di un'immagine, contiene le informazioni sui pixel, informazioni sul paziente e altri metadata. Si è avuto quindi la necessità di convertire questo tipo di immagine in un formato più portabile, per l'utilizzo successivo con la rete neurale.

Gli effetti di eventuali conversioni tra i formati, e del processo di ridimensionamento [10], non vanno a influire significativamente sui risultati finali e quindi si è deciso di trasformare i file DCM in PNG.

Per convertire le immagini, si è utilizzata la libreria pydicom, che mette a disposizione funzioni per gestire le immagini, visualizzarle ed estrarre l'array di pixel.

Ai pixel estratti, sono state effettuate operazioni di ridimensionamento, per portare i valori tra 0 e 255, ricostruendo poi l'immagine usando la libreria Pillow. Questo processo è stato automatizzato per tutti i file .DCM presenti nel dataset.

A questo punto ogni risonanza magnetica è stata trasformata nelle sue immagini corrispettive, che ne rappresentano i vari strati, in totale ogni RM si compone in 46 immagini che riportano vari strati e angolazioni del cervello.

### 3.1.2 Scelta dei range degli stadi

Per poter dividere i pazienti nei vari stati della malattia, si è preso in considerazione il file "Score.xlsx", contenente gli score delle risonanze magnetiche. Lo score è un intero che

```

1  import os
2  import pydicom
3  import numpy as np
4  from PIL import Image
5  from datasetLoader import extractDataForPaziente
6
7
8  def get_names(path_dir):
9      names_file = []
10     for root, dirnames, filenames in os.walk(path_dir):
11         for filename in filenames:
12             _, ext = os.path.splitext(filename)
13             if ext in ['.dcm']:
14                 names_file.append(filename)
15
16     return names_file
17
18 def convert_dcm_jpg(name):
19     im = pydicom.dcmread('Database/' + name)
20     im = im.pixel_array.astype(float)
21     rescaled_image = (np.maximum(im, 0) / im.max()) * 255 # float pixels
22     final_image = np.uint8(rescaled_image) # integers pixels
23     final_image = Image.fromarray(final_image)
24
25     return final_image
26

```

**Figura 3.2:** Script di conversione da .DCM a .PNG

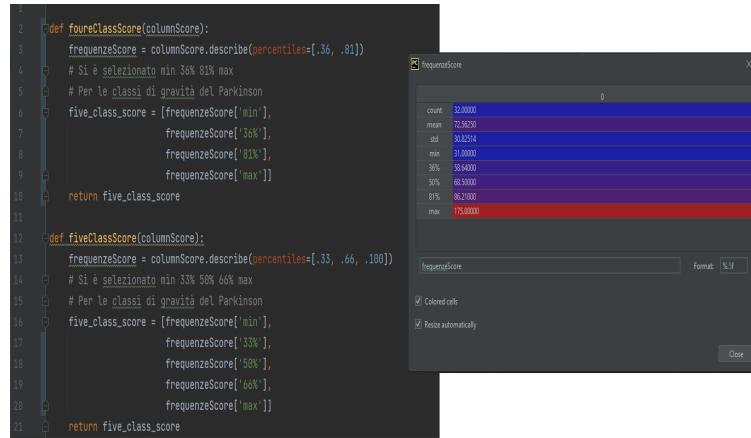
varia nel range [20,128], inizialmente il file non conteneva tutti i dati, si è quindi deciso di applicare una tecnica di data augmentation. Nel caso in cui un certo dato mancava, il dato veniva ricopiato dalla colonna precedente.

Per estrarre i range di frequenze e creare gli stadi della malattia è stata utilizzata la statistica. La libreria pandas, scritta per il linguaggio Python viene utilizzata per la manipolazione e l'analisi dei dati. In particolare offre strutture dati e operazioni per la manipolazione di tabelle numeriche e serie temporali. Tra queste operazioni troviamo la funzione describe, che restituisce la descrizione dei dati in DataFrame. La descrizione restituita fornisce dettagli statistici di base come percentuale, media, minimo, massimo, std (derivazione standard) e count (numero di valori non nulli). Questa funzione accetta anche dei parametri in percentuale, utilizzati per ottenere i range degli stadi; sono stati effettuati dei test per selezionare il numero di classi possibili da estrarre, in una prima fase sono state scelte 5 classi, ma causavano la divisione del dataset in parti disomogenee, e alcune classi avevano un numero di campione superiore ad altre. Si è scelto quindi di utilizzare 4 classi: min, 36%, 81% e max. I dati presi in esame sono l'ultima colonna ("V15") del file "Score".

### 3.1.3 Dataset

A questo punto, avendo a disposizione i range degli stadi della malattia è stato possibile costruire la struttura del nostro dataset. In particolare vengono create due cartelle, una per il test e una per il training, queste a loro volta composte da quattro sottocartelle raffiguranti gli stadi.

Per la percentuale di divisione del dataset, si è utilizzato il Principio di Pareto, nonché anche il più comune rate di split nel mondo del data science. L'80% dei dati del dataset viene



**Figura 3.3:** Risultati funzione describe() della libreria pandas sulla colonna V15.

dedicato per il training e il restante 20% per il testing e la validazione del modello.

```

#Percentuale di composizione del train
percentage_train = sizeTrain/100
#Percentuale di composizione del test
percentage_test = ((100 - sizeTrain)/100)
#Genera nomi casuali
rName = randomword
listPath, listScore = finalDataFrame['Directory ID'], finalDataFrame['ScoreVisit']

for (path, score) in zip(listPath, listScore):
    path_destination = divisionClass.getClassByScore(score)

    # Prendi tutte le immagini
    files = os.listdir(path)
    for fname in files:
        # Metti nella cartella del train
        val = random.random()
        if val < percentage_train:
            shutil.copy2(os.path.join(path, fname), dirTrain + "/" + path_destination + "/" + rName() + ".png")
        # Metti nella cartella del test
        else:
            shutil.copy2(os.path.join(path, fname), dirTest + "/" + path_destination + "/" + rName() + ".png")

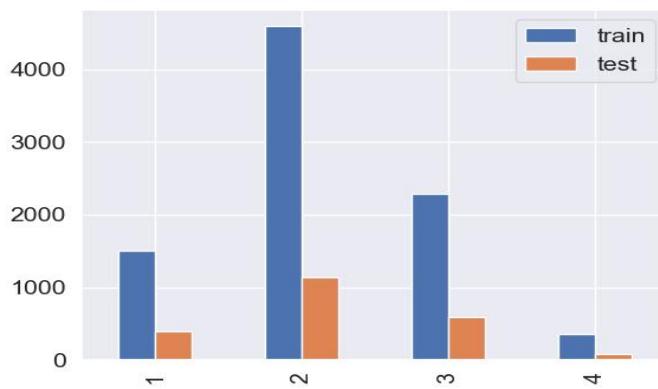
```

**Figura 3.4:** Snippet del dataset split.

Attraverso il file "Score" il punteggio delle risonanze dei relativi pazienti, sono messi in relazione mediante l'id paziente e l'identificativo delle risonanze. In questo modo si determina rapidamente l'effettivo score dato a una risonanza.

Lo script utilizzerà questa informazione per determinare la cartella di destinazione, ovvero in quale stadio posizionare le immagini; poiché è stato scelto di suddividere il dataset in questo modo, mediante un valore probabilistico lo script determina la macro cartella di destinazione, questo processo è molto importante per effettuare un buon split per i nostri dati.

I nomi delle immagini, che inizialmente erano tutti come part\_ seguito da un numero che ne indicava il frammento, sono stati sostituiti generandone casuali per evitare che i file venissero soprascritti al momento del salvataggio.



**Figura 3.5:** Dataset rappresentante il numero di esempi presenti nelle varie classi.

### 3.1.4 Image pre-processing

La pre-elaborazione dell’immagine, è il nome con cui si indicano delle operazioni sulle immagini a livello di astrazione più basso; il cui scopo è un miglioramento dei dati dell’immagine che sopprimono le distorsioni indesiderate o migliorano alcune caratteristiche.

Importante sottolineare che tutto ciò non aumenta il contenuto delle informazioni sull’immagine, ma rende più facile per il modello astrarre determinate features.

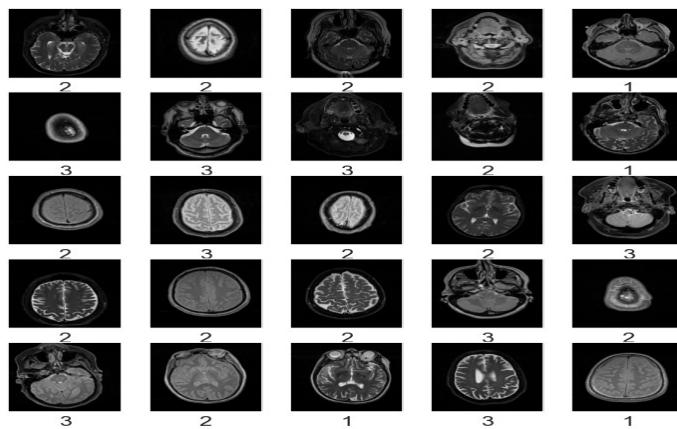
I modelli richiedono in input immagini quadrate, è stato necessario procedere con una fase di standardizzazione. Se si applicasse l’allungamento delle dimensioni, si provocherebbe uno "spazio morto" che verrebbe occupato da pixel con valori nulli. Per non creare confusione al modello, è stato scelto l’approccio downscaling, scegliendo come larghezza e un’altezza dimensione pari a 150x150.

Questa operazione di downscaling potrebbe aiuterà il modello a scansionare localmente parti dell’immagine concentrandosi di più sui dettagli.

Le immagini poi sono state normalizzazione, processo che modifica l’intervallo dei valori di intensità dei pixel, è stata applicata la divisione per "255.0" trasformando anche i dati in virgola mobile, poiché tale processo rende le reti neurali più efficienti.

### 3.1.5 Scelta del Batch Size

Quando il modello entra in fase di addestramento, il training set gli viene sottoposto, ma molto spesso potrebbe essere troppo grande per essere elaborato tutto in una volta. Quindi si può dividere il training set in sottogruppi uniformi, chiamati batch. Il numero di esempi contenuti in ogni bath è chiamato batch size.



**Figura 3.6:** Risultato delle immagini dopo le pre-elaborazioni

Il numero di epoch ed il batch size influisce sulla velocità di addestramento del modello, ma anche nel suo modo di performarsi.

Anche se nel mondo del machine learning non ci sono regole scolpite nella roccia, è sconsigliato usare sia batch size troppo grandi e sia quelli troppo piccoli. Se il batch size è troppo piccolo allora non si avrà un'ottimizzazione delle performance, oltretutto non sarà sfruttato a pieno le elaborazioni accelerate che avvengono durante il processo. Viceversa se il batch size è troppo grande si potrebbe avere un problema di esaurimento della memoria o una tendenza all'overfitting più accentuata. I batch size tipici sono 32, 64 o 128 (solitamente potenze di 2 per motivi di allocazione della memoria).

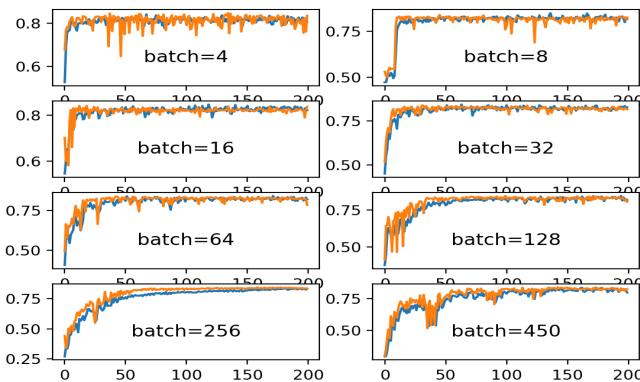
Dai test condotti nella sperimentazione, monitorando il processo di addestramento si è notato che aumentando il numero del batch size il modello tendesse a convergere verso un'accuratezza inferiore e incrementando notevolmente il tempo di addestramento.

Invece quando si utilizzava un numero di batch size inferiore, il calcolo dell'errore produceva più rumore rispetto a quando si utilizzava una dimensione batch più grande, e grazie al rumore l'algoritmo riesce a migliorarsi più efficacemente.

Pertanto si è riusciti a trovare un buon compromesso, che rapidamente, utilizzando una dimensione di batch più piccola ottimizzasse il modello con l'aiuto del rumore "indesiderato", riuscendo anche a ridurre il tempo di formazione del modello. Il batch size impostato per il modello è 16.

### 3.1.6 Scelta del numero di epochs

Un'epoca è quando un intero set di dati viene passato alla rete, e poiché questo set risulta essere troppo grande vengono divise in batch. Il numero di epoche è un iperparametro che



**Figura 3.7:** Test con differenti batch size.

definisce quante volte il modello riceverà il training set. Questo significa che ogni campione nel set di dati di addestramento ha avuto l'opportunità di aggiornare i parametri del modello interno. Il giusto numero di epoche dipende dalla perplessità (o complessità) intrinseca del set di dati. Una buona regola pratica è iniziare con un valore che è 3 volte il numero di colonne nei dati. Se si scopre che il modello alla fine di queste epoche ha ancora margine di migliorarsi, se ne aumentano il numero, altrimenti quello che si scopre è che il modello ha smesso di migliorarsi molto prima dell'epoca finale, e a quel punto si riprova con un valore più piccolo. Il numero di epoche influisce significativamente il tempo di addestramento del modello.

Inizialmente si è fissato il numero di epoche a 64; analizzando il suo comportamento in fase di training, il modello già dalla 20esima epoca riusciva a raggiungere una buona accuratezza e nelle successive si nota come l'aggiornamento dei pesi non riesca a portare nuovi benefici, nota che potrebbe farci intendere che il modello sia riuscito ad apprendere la struttura dei dati e non riesca più a migliorarsi aggiornando i pesi degli archi.

Dopo questa analisi il numero di epoche sono state impostate a 24.

### 3.1.7 Architettura del modello

Il modello scelto è di tipo sequenziale, ogni livello ha esattamente un input e un output. Questi livelli (o layers) sono impilati insieme formando l'intera rete.

Come primo strato troviamo un layer convoluzionale 2D, dove la dimensione dell'input combacia con la dimensione scelta per le nostre immagini ( $150 \times 150$ ). La finestra di scorrimento è stata impostata  $3 \times 3$  e il numero di output a 64, come funzione di attivazione è stata scelta quella lineare rettificata "Relu".

```

Epoch 18/64
437/437 [=====] - 189s 432ms/step - loss: 0.0117 - accuracy: 0.9959 - mae: 1.0069 - val_loss: 0.2735 - val_accuracy: 0.9411 - val_mae: 1.0140
Epoch 19/64
437/437 [=====] - 188s 431ms/step - loss: 0.0258 - accuracy: 0.9931 - mae: 1.0069 - val_loss: 0.3047 - val_accuracy: 0.9182 - val_mae: 1.0140
Epoch 20/64
437/437 [=====] - 185s 423ms/step - loss: 0.0253 - accuracy: 0.9923 - mae: 1.0069 - val_loss: 0.3472 - val_accuracy: 0.9108 - val_mae: 1.0140
Epoch 21/64
437/437 [=====] - 193s 441ms/step - loss: 0.0301 - accuracy: 0.9896 - mae: 1.0069 - val_loss: 0.3205 - val_accuracy: 0.9262 - val_mae: 1.0140
Epoch 22/64
437/437 [=====] - 190s 434ms/step - loss: 0.0142 - accuracy: 0.9959 - mae: 1.0069 - val_loss: 0.2936 - val_accuracy: 0.9279 - val_mae: 1.0140
Epoch 23/64
437/437 [=====] - 189s 433ms/step - loss: 0.0077 - accuracy: 0.9983 - mae: 1.0069 - val_loss: 0.3293 - val_accuracy: 0.9296 - val_mae: 1.0140

```

**Figura 3.8:** Analisi del training sul numero delle epoche.

Il secondo layer è di tipo Dropout, è stato inserito per prevenire l'overfitting dei dati, questo layer permette di selezionare dei neuroni in modo casuale e ignorarli durante l'allenamento. Il parametro rate è impostato a 0.2, e il layer è attivo solo nella fase di training.

Il terzo layer che viene utilizzato è un layer di MaxPooling2D, con una dimensione di finestra e di scorrimento pari a 2 X 2. Questo layer andrà a selezionare il massimo valore che trova nella sua finestra.

La rete si sussegue con un altro strato alternato di Conv2D, dropout, Conv2D e MaxPooling2D con output rispettivamente 32 e 16, per selezionare features di più alto livello.

È presente poi un layer di appiattimento, che trasforma l'input dei livelli precedenti (di tipo multidimensionale) in un output monodimensionale.

Nello strato finale troviamo due layer Dense, dove il primo ha 128 neuroni e come funzione di attivazione "relu" e il secondo ha come funzione di attivazione "softmax" e quattro neuroni indicanti i 4 stadi della progressione del parkinson.

Per il train del modello è stato impostato l'ottimizzatore Adam, lasciando i parametri di default, la funzione di perdita scelta è quella di crossentropia, ottima quando sono presenti due o più classi di etichette. La grandezza del batch size è stata di 16 con 24 epche, validation split impostato a 0.2 e si è attivata la modalità shuffle per mescolare i dati prima di ogni epoca.

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 64)	1792
dropout (Dropout)	(None, 148, 148, 64)	0
max_pooling2d (MaxPooling2D)	(None, 74, 74, 64)	0
conv2d_1 (Conv2D)	(None, 72, 72, 32)	18464
dropout_1 (Dropout)	(None, 72, 72, 32)	0
conv2d_2 (Conv2D)	(None, 70, 70, 16)	4624
max_pooling2d_1 (MaxPooling2D)	(None, 35, 35, 16)	0
flatten (Flatten)	(None, 19600)	0
dense (Dense)	(None, 128)	2508928
dense_1 (Dense)	(None, 4)	516

**Figura 3.9:** Summary del modello.

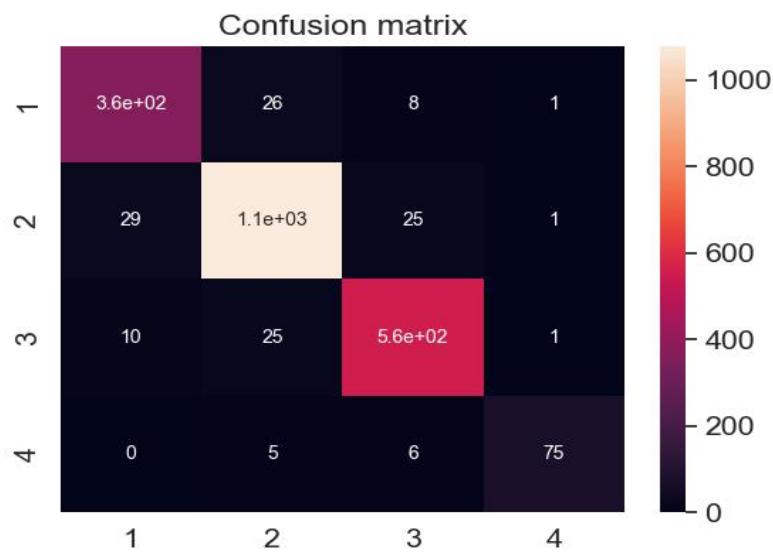
### 3.1.8 Metriche

Le metriche utilizzate per valutare il modello, sono state l'analisi dell'accuracy, il loss, il mean absolut error, e la matrice di confusione.

La prima metrica che andremo ad analizzare è la funzione di loss, il modello in fase di validazione ha ottenuto il valore 0.19 che è da considerarsi buono, in quanto è poco distante dallo 0, che potremmo definire come il modello non sbaglia mai, primo sintomo che potrebbe farci pensare a overfitting.

L'accuracy raggiunta è del 94%, il mean absolut error (la differenza media osservata nei valori previsti ed effettivi nell'intero set di test) è pari a 1.01 che in percentuale ci indica la differenza di valori tra quelli previsti ed effettivi variava di un 10%.

La matrice di confusione per un problema di regressione indica gli errori individuali commessi dal modello. Possiamo notare che la maggior parte della confusione avviene tra la prima/seconda classe e seconda/terza classe con valori che si aggirano tra 25-29, infine un ultimo valore che possiamo osservare è l'ultima classe in cui troviamo il valore 75, segno di come i pochi dati abbiano portato il valore del true positive relativamente basso rispetto ai valori che si trovano sulla diagonale principale ovvero i true positive delle rispettive classi.



**Figura 3.10:** Risultati della Confusion Matrix.

### 3.1.9 Applicazione

Come ultimo step si è realizzata una piccola applicazione, per poter utilizzare il modello con maggiore facilità. La parte back-end è stata realizzata con Flask, un micro-framework Web scritto in Python, mentre per la parte grafica, è stato adoperato il motore di template Jinja2. L'applicazione permette di caricare l'immagine di una risonanza magnetica in formato .jpg e ottenere la sua predizione.



**Figura 3.11:** Viene mostrata una delle schermate dell'applicazione, in particolare quella che mostra il risultato della predizione.

## 3.2 Risultati

I dati forniti per questo lavoro di tesi hanno fatto sì, che venissero prese alcune decisione per creare un dataset coerente e utilizzabile. Da qui la scelta del numero di classi. Le varie classi sono divise in base a dei valori che hanno portato a valutare come numero iniziale cinque. Successivamente però si sono rivelate inidonee in quanto hanno reso il dataset sbilanciato, da qui l'idea definitiva di porte le classi da cinque a quattro. Questa scelta ha ridotto la disomogeneità, ma non ha eliminato del tutto il problema.

Il modello di tipo sequenziale, creato utilizzando metodologie di deep learning, ha permesso alla rete neurale di apprendere in maniera autonoma dall'input, ovvero dalle immagini 2D delle risonanze.

Gli iperparametri del modello sono stati settati in base ad osservazioni fatte durante gli addestramenti e il test. Il numero di batch size più piccolo ha favorito un accuratezza maggiore, mentre il numero di epoche ha fatto sì che un tempo di addestramento inferiore, non andasse ad intaccare l'accuratezza.

La fase di addestramento del modello ha richiesto 3 ore, ottenendo un'accuratezza nel testing del 94%. Il report dalla confusion metrix, indica che la maggiore confusione del modello si trova tra la prima/seconda classe e la seconda/terza, il che sembra un buon risultato, in quanto sottolinea una sorta di progressione. Inoltre la confusion metrix evidenzia ancora una volta, il numero inferiore di esempi presenti nella quarta classe, prevedibile per le considerazioni fatte inizialmente.

## CAPITOLO 4

---

### Conclusioni

---

Le malattie neuro-degenerative sono patologie caratterizzate dalla perdita lenta e progressiva delle cellule neuronali in determinate aree del cervello, che portano al paziente disturbi del movimento e del funzionamento mentale. Tra queste malattie troviamo il morbo di Parkinson oggetto di studio in questa tesi. Questo elaborato si è focalizzato nell'applicare tecniche di apprendimento automatico al settore delle scienze biomediche, in particolare nel costruire un modello in grado di fornire supporto ai medici nell'individuazione della malattia del Parkinson. Questa malattia non ha cura, e solo attraverso una diagnosi precoce il paziente ne può alleviare i sintomi.

Si è partiti analizzando e preparando i dati, individuando quattro classi di progressione per la malattia, si è poi passati alla costruzione e addestramento del modello. Successivamente il modello è stato ottimizzato, riuscendo ad ottenere un'accuratezza del 94%. Le principali problematiche si sono focalizzate sui pochi dati a disposizione; quando si effettua un lavoro di questo tipo, più dati si hanno a disposizione e migliori saranno i risultati. Nel dataset a disposizione le varie classi contenevano una disparità dei dati anche maggiore del 50%. Ciò non ha impedito comunque di raggiungere un buon risultato, ma sicuramente migliorabile.

Nei futuri sviluppi sono convinto che rimuovendo dal dataset le immagini delle risonanze magnetiche poco significative possa ottimizzarne il risultato, visto che tra le varie immagini compaiono anche alcuni strati delle risonanze magnetiche che sono poco significative per analizzare la zona interessata del cervello. Altri sviluppi potrebbero anche concentrarsi nell'ottimizzare il modello che è stato costruito, aumentando la dimensione del dataset,

---

magari utilizzando tecniche di data augmentation sull'ultimo stadio della malattia, oppure focalizzarsi su una diversa struttura di rete, come potrebbe essere l'utilizzo di un modello 3DCNN.

I modelli di deep learning applicati in medicina permettono ai ricercatori e ai medici di avere un supporto moderno e tecnologico per la prevenzione, l'identificazione e la classificazione di patologie, offrendo una seconda opinione, al momento del processo decisionale.

---

## Bibliografia

---

- [1] H. Focke NK, "Individual voxel-base subtype prediction can differentiate progressive supranuclear palsy from idiopathic parkinson syndrome and healthy controls. hum brain mapp.," 2011. (Citato alle pagine iv, 14 e 15)
- [2] S. S. Babu GS, "A novel pbl-mcrbfn-rfe approach for identification of critical brain regions responsible for parkinson's disease. expert syst appl," 2014. (Citato alle pagine iv e 15)
- [3] C. A. Salvatore C, "Machine learning on brain mri data for differential diagnosis of parkinson's disease and progressive supranuclear palsy. j neurosci methods," 2014. (Citato alle pagine iv e 16)
- [4] p. y. t. w. f. z. h. h. x. x. chiyu feng, ahmed elazab and baiying lei, "Deep learning framework for alzheimer's disease diagnosis via 3d-cnn and fsbi-lstm," 2019. (Citato alle pagine iv, 16, 17 e 18)
- [5] X. W. R. M. Z. M. B. R. M. S. J. Y. Ling Zhang, Le Lu, "Spatio-temporal convolutional lstms for tumor growth prediction by learning 4d longitudinal patient data," 2019. (Citato alle pagine iv, 17 e 18)
- [6] H. A. . R. R. Sukhpal Kaur, "Diagnosis of parkinson's disease using deep cnn with transfer learning and data augmentation," 2020. (Citato alle pagine iv, 18, 19 e 20)
- [7] Parkinson's Progression Markers Initiative, "PPMI." <https://www.ppmi-info.org/>. (Citato alle pagine 1, 4 e 18)
- [8] Wikipedia contributors, "Voxel-based morphometry — Wikipedia, the free encyclopedia." [https://en.wikipedia.org/w/index.php?title=Voxel-based\\_](https://en.wikipedia.org/w/index.php?title=Voxel-based_)

- morphometry&oldid=1085241743, 2022. [Online; accessed 12-July-2022]. (Citato a pagina 15)
- [9] Y. W. H. Zhou, "Data augmentation computing model based on generative adversarial network," 2019. (Citato a pagina 19)
- [10] M. S. S. o. Muminov Bakhodir Boltaevich; Nasimov Rashid Hamid ogl, Gadoyboyeva Nigora Soibjon qizi, "Estimation affects of formats and resizing process to the accuracy of convolutional neural network," 2019. (Citato a pagina 22)

---

## Ringraziamenti

---

Un sentito grazie alla prof. ssa Rita Francese, relatrice di tesi, per il supporto costante e le dritte indispensabili.

Grazie anche al mia correlatrice dott. ssa Maria Frasca, per i suoi consigli e per avermi suggerito puntualmente le giuste modifiche da apportare alla mia tesi.

Alla mia famiglia, i miei amici, i miei compagni di università e a tutti quelli che hanno incrociato la loro vita con la mia lasciandomi qualcosa di buono. Grazie per essere stati miei complici, ognuno a suo modo, in questo percorso intenso ed entusiasmante, nel bene e nel male.

Ringrazio Raffy per tutto il tempo che mi hai dedicato. Grazie perché ci sei sempre stata.