

# Progetto Sistemi Multimediali - Classificazione di livelli di gravità dell'Alzheimer: Report

Romito Emanuele - e.romito3@studenti.uniba.it

Giugno 2024

## 1 Abstract

Questo progetto esplora lo stato dell'arte relativo all'utilizzo di Reti Neurali Convolutionali (CNN) per compiti di classificazione nel dominio dell'Healthcare, con un focus specifico sulle pratiche di preprocessing e scelte di carattere architettonico adottate. Le CNN, grazie alla loro capacità di estrarre caratteristiche rilevanti da qualsiasi tipo di immagine, sono ampiamente utilizzate per diagnosticare e classificare vari tipi di patologie. Questo lavoro esamina le metodologie più avanzate e le pratiche ottimali per l'implementazione delle CNN nel contesto della diagnosi dell'Alzheimer, fornendo una panoramica delle soluzioni più efficaci.

## 2 Stato dell'Arte

Le tecniche di Deep Learning (DL) sono attualmente considerate le più efficaci per eseguire operazioni di computer vision, come il rilevamento e la classificazione degli oggetti. Il termine "deep" in deep learning si riferisce al numero di strati attraverso cui i dati vengono trasferiti. Questi modelli possiedono una struttura gerarchica in cui i layer sono interconnessi, permettendo di trasformare le caratteristiche di basso livello in caratteristiche di alto livello. Grazie a queste peculiarità, i modelli di deep learning sono considerati più potenti rispetto ai modelli di machine learning tradizionali. Attualmente esistono numerosi modelli di deep learning, tra cui Autoencoder (AE), Deep Belief Network (DBN), Convolutional Neural Network (CNN) e Recurrent Neural Network (RNN). Le CNN rappresentano un tipo eccezionale di reti neurali artificiali feed-forward, particolarmente adatte al riconoscimento di schemi visivi bidimensionali (2D). Le CNN si basano su concetti chiave come campi ricettivi locali e sharing dei parametri. La struttura delle CNN comprende tre tipi principali di layer: layer convoluzionali, layer di subsampling e layer di output [1]. Questi layer, in una CNN, sono organizzati in una disposizione feed-forward, tramite un'architettura stratificata:

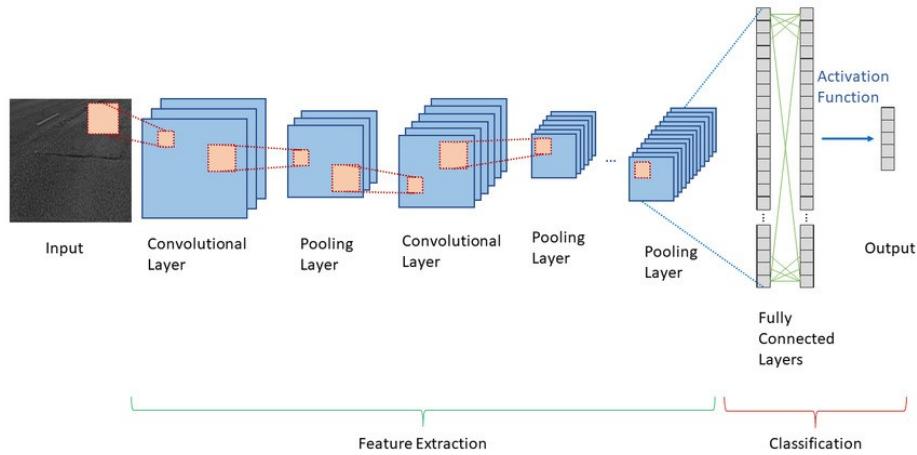


Figure 1: Architettura di una CNN.

- I layer convoluzionali si occupano di eseguire l'estrazione delle feature dalle immagini di input, applicando filtri convoluzionali per identificare pattern e strutture rilevanti a vari livelli di astrazione [1].
- I layer di subsampling, anche noti come strati di pooling, si occupano di ridurre la dimensionalità delle mappe di feature prodotte dai layer convoluzionali [1].
- Il layer di output è solitamente un layer denso (fully connected layer) seguito da una funzione di attivazione, come softmax per problemi di classificazione multiclasse, che trasforma le caratteristiche ad alto livello in probabilità di appartenenza alle diverse classi di output [1].

Le CNN vengono sviluppate utilizzando un budget computazionale prefissato, per poi essere scalate per ottenere maggior accuratezza: è possibile infatti fare scaling della risoluzione delle immagini di input, fare scaling dei layer che compongono la rete o fare scaling del numero dei parametri contenuti da un layer. Generalmente, si tende a fare scaling solamente di una delle dimensioni sopracitate per volta. Questa pratica però, può rendere inefficiente il modello, in quanto intuitivamente se si aumentasse la risoluzione delle immagini in input, allora la rete necessiterebbe di più layer per incrementare le capacità ricettive e così via. In questa applicazione dunque ci si è soffermati sullo studio e sull'implementazione di una CNN che effettua il cosiddetto *compound scaling*, ovvero EfficientNet: l'idea dietro lo studio effettuato dai ricercatori in [2] è stata quella di effettuare uno scaling uniforme delle dimensioni, ottenendo ottimi risultati in termini di accuratezza rispetto al numero dei parametri comparati ad altri foundation models.

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPs	Ratio-to-EfficientNet
<b>EfficientNet-B0</b>	<b>77.1%</b>	<b>93.3%</b>	<b>5.3M</b>	<b>1x</b>	<b>0.39B</b>	<b>1x</b>
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
<b>EfficientNet-B1</b>	<b>79.1%</b>	<b>94.4%</b>	<b>7.8M</b>	<b>1x</b>	<b>0.70B</b>	<b>1x</b>
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
<b>EfficientNet-B2</b>	<b>80.1%</b>	<b>94.9%</b>	<b>9.2M</b>	<b>1x</b>	<b>1.0B</b>	<b>1x</b>
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
<b>EfficientNet-B3</b>	<b>81.6%</b>	<b>95.7%</b>	<b>12M</b>	<b>1x</b>	<b>1.8B</b>	<b>1x</b>
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
<b>EfficientNet-B4</b>	<b>82.9%</b>	<b>96.4%</b>	<b>19M</b>	<b>1x</b>	<b>4.2B</b>	<b>1x</b>
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
<b>EfficientNet-B5</b>	<b>83.6%</b>	<b>96.7%</b>	<b>30M</b>	<b>1x</b>	<b>9.9B</b>	<b>1x</b>
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
<b>EfficientNet-B6</b>	<b>84.0%</b>	<b>96.8%</b>	<b>43M</b>	<b>1x</b>	<b>19B</b>	<b>1x</b>
<b>EfficientNet-B7</b>	<b>84.3%</b>	<b>97.0%</b>	<b>66M</b>	<b>1x</b>	<b>37B</b>	<b>1x</b>
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

Figure 2: Performance di EfficientNet rispetto ad altri modelli [2].

### 3 Lavori Correlati

In [3] si analizzano metodi per utilizzare al meglio CNN in compiti di Image Classification su dataset prettamente di dominio medico, in particolare ChestX-ray 14, un dataset di oltre centomila immagini. I ricercatori in [3] affermano come immagini di carattere medico sono molto difficili da reperire (il dataset sopracitato ad esempio seppur essendo uno dei più vasti nel dominio dell'healthcare conta milioni di immagini in meno rispetto a ImageNet), sottolineando come sia importante reperire nuove immagini (che rimane pur sempre la soluzione più efficace) e riuscire ad aumentare le performance dei modelli su dataset molto ridotti, una di queste tecniche è sicuramente quella della *Data Augmentation* (DA): i risultati ottenuti in [3] indicano come i processi di augmentation aumentano le performance del modello a prescindere dall'architettura scelta, nonostante processi di DA aggressivi risultano controproducenti, favorendo metodi di DA più semplici; inoltre in [3] viene sottolineato come modelli di dimensione ridotta non riescono ad imparare sufficientemente dai dati, ottenendo delle scarse performance, mentre utilizzando modelli che impegnano enormi risorse computazionali spesso porta all'overfitting: è dunque cruciale prevenire questi fenomeni con un corretto uso delle tecniche di DA e di *Dropout*.

## 4 Design della Pipeline Sperimentale

### 4.1 Dataset

Il dataset utilizzato per addestrare la CNN è l' "Alzheimer's Dataset", contenente immagini di risognanze magnetiche del cervello suddivise in 4 classi: Mild Demented, Moderate Demented, Non Demented, Very Mild Demented. Possiede 6400 immagini suddivise in immagini di training e immagini di test.

### 4.2 Hardware

Per eseguire l'addestramento della CNN è stata utilizzata una TPU Tesla 4 e 12 GB di RAM.

### 4.3 Software

Per organizzare il codice di ML ci si è serviti di un progetto colab, nel quale è stato utilizzato un Jupyter Notebook per eseguire le celle di codice Python per effettuare il preprocessing dei dati e l'addestramento della rete.

### 4.4 Preprocessing

Dopo una rapida analisi, è chiaro come la mancanza di un dataset di validazione è una grossa problematica riguardante questo dataset. Si è effettuato dunque un resampling del dataset (utilizzando uno split 80/20) per ottenere anche un dataset di validazione. Un'altra problematica principale è l'evidente sbilanciamento dei sample presenti nel dataset tra le classi. Utilizzando SMOTE, è stato effettuato un over-sampling delle immagini per uniformare la distribuzione tra le classi (ottenendo 2000 campioni per classe). Tramite la libreria *keras-cv*, è stata effettuata una operazione di RandAugment [4], ovvero delle operazioni di DA randomiche sul dataset di addestramento.

### 4.5 Architettura

Per configurare il modello si è partiti da un modello di base basato su EfficientNetB0, ovvero il modello appartenente alla famiglia EfficientNet con il minor numero di parametri, questo perchè su utilizzando un dataset di poco più di 10000 immagini il rischio di overfitting è molto alto. Il modello di base dunque si basa sulla cosiddetta "Backbone" di EfficientNetB0 (ovvero quella parte della rete che si occupa di fare estrazione delle feature) senza la sua parte di "Head" (ovvero quella parte che si occupa di fare classificazione). Tale modello viene dunque posto in una struttura sequenziale utilizzando l'API Python di Keras:

- Input Layer
- Backbone EfficientNetB0
- Layer di Pooling

- Layer di Dropout(0.5)
- Layer Denso(512, relu)
- Layer di Dropout(0.2)
- Layer Denso(256, relu)
- Layer di Dropout(0.2)
- Layer Denso(128, relu)
- Layer di Dropout(0.2)
- Layer Denso(4, softmax)

I layer densi compongono la parte dell'architettura che si occupa di fare classificazione, quindi l'Head. Tra i layer di pooling e i layer densi inoltre, sono stati posti dei layer di Dropout, i quali pongono a zero gli input dei layer successivi in con una frequenza definita dal rateo (in questo caso 0.5 e 0.2) durante il processo di training, ciò finalizzato al prevenire l'overfitting. Per prevenire tale fenomeno inoltre, per quanto riguarda l'architettura del modello, vengono utilizzate altre due pratiche:

1. Adam: un algoritmo di ottimizzazione che regola dinamicamente il learning rate per ciascun parametro, consentendo aggiornamenti più stabili ed efficienti e accelerando la convergenza.
2. Learning Rate Schedule: una tecnica che permette di adattare il learning rate durante il training del modello. Questo approccio riduce progressivamente il learning rate in funzione del tempo, permettendo al modello di effettuare aggiustamenti iniziali più grandi e poi raffinamenti più piccoli man mano che il training procede, migliorando la stabilità e l'efficienza del processo di apprendimento.

## 5 Risultati

Sono stati ottenuti risultati soddisfacenti in termini di accuratezza durante il processo di training, con una validation accuracy rasente al 0.98 ed una loss (è stata utilizzata una Sparse Categorical Cross Entropy) di circa 0.006. Per quanto riguarda la fase di test del modello si è riusciti ad ottenere una accuracy di circa 0.76 con una loss pari a 1.2.

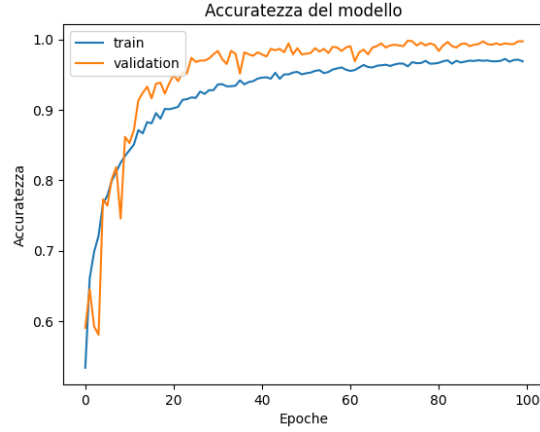


Figure 3: Accuratezza del modello.

Consultando il report si intuisce come il modello ha performato in maniera eccellente in termini di precision per le classi "MildDemented" e "ModerateDemented", mentre per le classi "NonDemented" e "VeryMildDemented" ha ottenuto sicuramente dei risultati peggiori.

	precision	recall	f1-score	support
MildDemented	0.861702	0.452514	0.593407	179.000000
ModerateDemented	1.000000	0.916667	0.956522	12.000000
NonDemented	0.820755	0.815625	0.818182	640.000000
VeryMildDemented	0.665428	0.799107	0.726166	448.000000
accuracy	0.759969	0.759969	0.759969	0.759969
macro avg	0.836971	0.745978	0.773569	1279.000000
weighted avg	0.773760	0.759969	0.755791	1279.000000

Figure 4: Report di Classificazione.

## 6 Bibliografia

- [1] S. Panigrahi, A. Nanda and T. Swarnkar, "Deep Learning Approach for Image Classification," 2018 2nd International Conference on Data Science and Business Analytics (ICDSBA), Changsha, China, 2018
- [2] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." International conference on machine learning. PMLR, 2019.
- [3] Yadav, Samir Jadhav, Shivajirao. (2019). Deep convolutional neural network based medical image classification for disease diagnosis. Journal of Big Data. 6. 10.1186/s40537-019-0276-2.
- [4] Cubuk, Ekin D., et al. "Randaugment: Practical automated data augmentation with a reduced search space." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020.