CLASSIFICAZIONE EDSS

Deep Learning

Angelo Nazzaro

Università degli Studi di Salerno

Anno Accademico 2024-2025

Indice

1 Introduzione

La sclerosi multipla (SM) è una malattia cronica, spesso invalidante, che colpisce il sistema nervoso centrale. I sintomi della sclerosi multipla possono essere lievi, come per esempio l'intorpidimento a livello degli arti, oppure severi, come la perdita della vista; il progresso e la gravità della sintomatologia di questa malattia non sono prevedibili e variano da individuo ad individuo. Questa è caratterizzata dallo sviluppo di tessuto cicatriziale (sclerosi), al posto della normale componente tissutale che sostituisce la normale componente tissutale del sistema nervoso, interferendo con la trasmissione degli impulsi nervosi [Gri23].

La Scala di Invalidità Espansa (EDSS) è una scala che ha lo scopo di valutare il livelli di disabilità delle persone con SM; va da 0, corrispondente a un esame neurologico normale, a 10. All'interno contempla livelli intermedi e sempre maggiori di invalidità. Il punteggio si ottiene sommando i punteggi parziali dei diversi sistemi funzionali legati all'attività del sistema nervoso (piramidale, cerebellare, sfinterica eccetera). Essa consente una più agevole valutazione dell'evoluzione della malattina e permette di verificare l'efficacia della terapia in atto [Mul25].

2 Obiettivi e Research Questions

Nel seguito di questo lavoro, analizzeremo la classificazione del livello di disabilità dei pazienti in due contesti:

- **Classificazione binaria**: i pazienti sono raggruppati in due classi in base al loro punteggio sulla EDSS, ovvero la classe *positiva* che include tutti i pazienti con un valore EDSS inferiore o uguale a 2,0 e la classe *negativa* con tutti i pazienti con EDSS superiore a 2,0. Un paziente appartenente alla classe positiva è un paziente che non presenta lesioni significative e ha una compromissione funzionale minima o nulla, mentre la classe negativa comprende pazienti con lesioni neurologiche evidenti e sintomi clinici.

- **Classificazione multiclasse**: i punteggi EDSS sono stati mappati su tre categorie: *normal*, *mild* e *severe*. Nello specifico, i punteggi da 0 a 2,0 sono stati etichettati come normal, indicando una disabilità da lieve a assente; i punteggi compresi tra 2,5 e 4,0 sono stati etichettati come mild, indicando pazienti con compromissione moderata ma deambulazione preservata, mentre i punteggi superiori a 4,0 sono stati etichettati come severe, corrispondenti a individui con disfunzioni motorie o sistemiche significative.

Per affrontare questo compito, svilupperemo e confronteremo due tipologie di modelli: uno basato su Convolutional Neural Networks (CNN) e uno su Vision Transformer (ViT) [DBK+21]. Poiché disponiamo di un dataset limitato e la destinazione d'uso è clinica, ci concentreremo su modelli "leggeri", al di sotto degli 80M di parametri, che possano essere distribuiti in ambienti con risorse computazionali limitate. Inoltre, l'impiego di modelli complessi potrebbe essere non adatto in un contesto applicativo come quello medico, dove i dati scarseggiano a causa di alti costi di acquisizione e annotazione e continui problemi di privacy dei pazienti e politiche amministrative che influiscono sulla condivisione di dati.
In sintesi, le principali *research questions* che guideranno questo lavoro sono:

RQ 1. Come l'utilizzo di funzioni di perdita pesate o specifiche, come la Focal Loss, congiunto a tecniche di bilanciamento dei dati, influisce nella gestione dello sbilanciamento di classe durante l'addestramento dei modelli?

Obiettivo: Valutare se queste tecniche migliorano la sensibilità e la precisione verso le classi minoritarie.

RQ 2. Quali sono le prestazioni dei modelli in contesti intra-domain (addestramento e test sulla stessa sequenza MRI) rispetto a contesti cross-domain (addestramento su una sequenza, test su un'altra)?

Obiettivo: Valutare le capacità di generalizzazione e di trasferibilità dei modelli tra le diverse modalità di acquisizione.

RQ 3. In che misura modelli leggeri basati su CNN e Vision Transformer riescono a classificare accuratamente i livelli di disabilità dei pazienti da singole sequenze MRI, e quali vantaggi offrono in termini di applicabilità clinica?

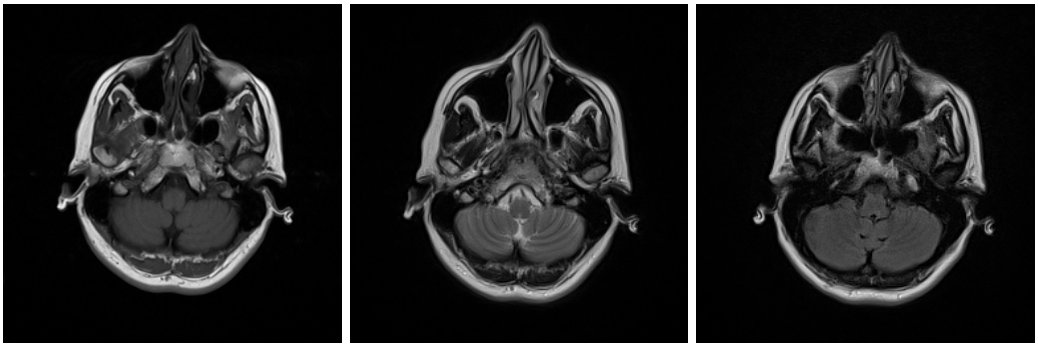
Obiettivo: Confrontare l'efficacia e la praticità di architetture diverse in un contesto medico.

3 Metodologia

3.1 Analisi dei Datasets

I dati a disposizione comprendono sequenze MRI cerebrali scomposte in immagini 2D in scala di grigio in formato `.png`. Le sequenze sono state acquisite in tre modalità diverse: T1, T2 e FLAIR, dando luogo a tre diversi datasets in quanto ogni modalità di acquisizione evidenzia in maniera differente diversi tipi di tessuti e potrebbe essere più o meno adatta alla classificazione. Ad esempio, le sequenze T1 forniscono una rappersentazione più dettagliata della struttura anatomica e delle alterazioni patologiche, le sequenze T2 sono più propense per il rilievo di emoraggie e lesioni nel contenuto fluido, mentre la modalità FLAIR è particolarmente efficace nell'indivudazione di lesioni. In Fig. 1 si evidenziano le differenze tra le diverse modalità.

Si riportano di seguito il quantativo di sequenze disponibili per ogni modalità: T1 - 952 sequenze, T2 - 964 , FLAIR - 1006 sequenze. In totale, si hanno a disposizione 2923 sequenze MRI.



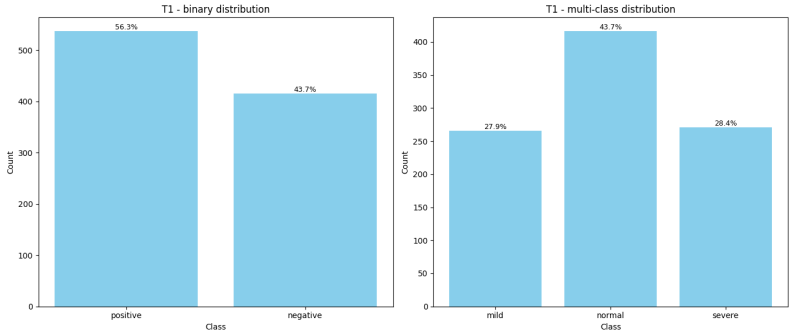
(a) Esempio MRI T1. (b) Esempio MRI T2. (c) Esempio MRI FLAIR.

Figura 1: Esempi di MRI dai datasets.

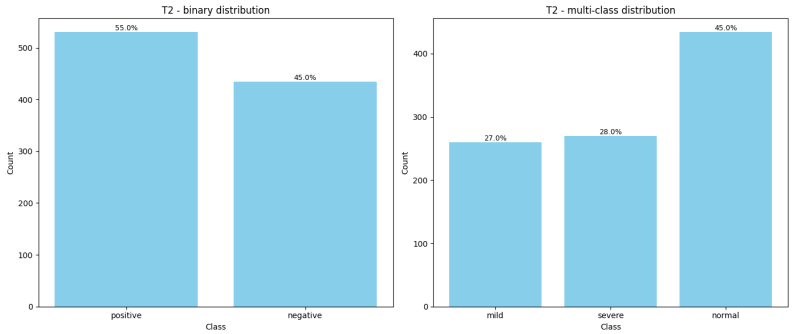
Distribuzione delle classi Tutti e tre i dataset/modalità (T1, T2 e FLAIR) presentano un profilo di distribuzione delle classi molto simile.

Nel contesto binario, la classe positiva è leggermente prevalente, con una media di circa il 55% delle istanze, mentre la classe negativa si attesta attorno al 45%. Tale squilibrio è trascurabile e non è stato pertanto oggetto di tecniche di bilanciamento. Nel contesto multiclasse, invece, lo squilibrio è più pronunciato: la classe *normal* costituisce circa il 45% del dataset, mentre le classi *mild* e *moderate* compaiono in proporzioni simili, suddividendosi tra il restante 55%.

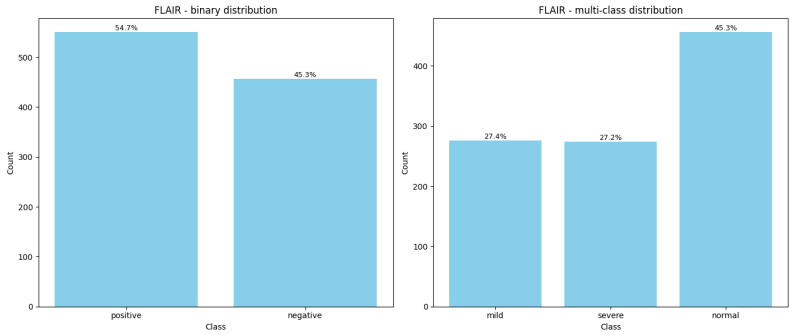
Le distribuzioni per ciascuna modalità sono mostrate in Fig. 2a, Fig. 2b e Fig. 2c.



(a) T1.



(b) T2.



(c) FLAIR.

Figura 2: Distribuzione delle classi per le tre modalità sia nel contesto binario (sinistra) che multiclasse (destra).

3.2 Data Augmentation

Dato l'evidente squilibrio nella distribuzione delle classi nel contesto multiclasse (Sez. 3.1), si è scelto di applicare un approccio di oversampling, incrementando il numero di istanze delle classi minoritarie tramite tecniche di data augmentation. L'oversampling è stato preferito rispetto all'undersampling vista la scarsa quantità di dati. In particolare, sono state applicate piccole rotazioni in senso orario e antiorario $(-20°, 20°)$ e l'aggiunta di rumore gaussiano. In Fig. 3 è mostrato un esempio delle transformazioni.

Queste tecniche sono state applicate con l'intento di simulare i movimenti che i pazienti fanno durante l'acquisizione e il rumore che può essere introdotto dallo scanner, cercando anche di rendere i modelli più robusti ad artefatti reali. Inoltre, data l'assenza di una competenza medica adeguata, l'utilizzo di tecniche più invasive, come color jitter, zoom e flip,

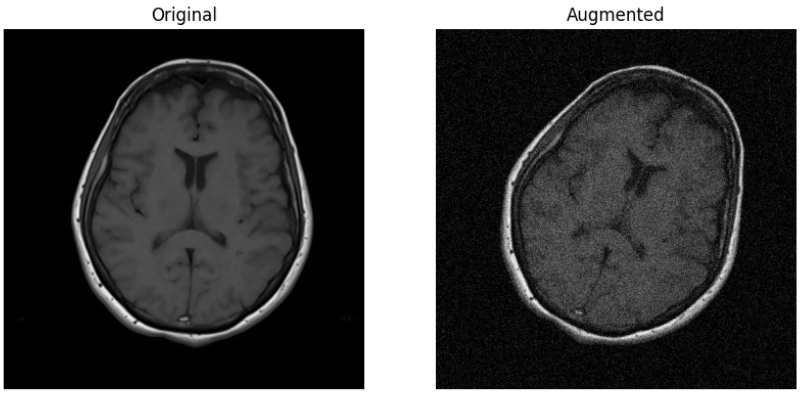
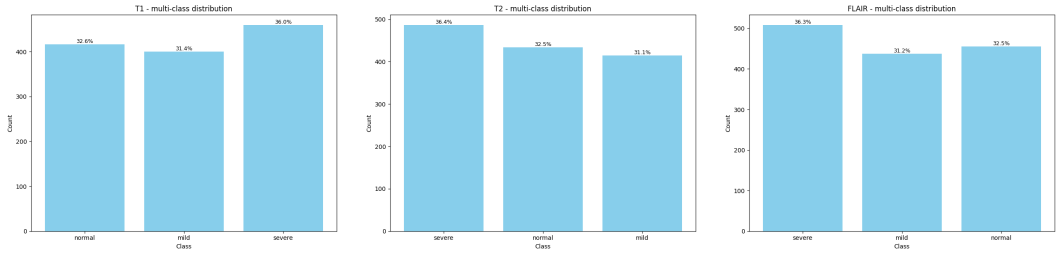


Figura 3: Esempio di applicazione di data augmentation.

è stata scartata poiché avrebbe potuto compromettere la validità anatomica delle immagini. È stata considerata anche la possibilità di generare dati sintetici, ma questa opzione è stata scartata per la stessa ragione: l'assenza di un'adeguata competenza medica avrebbe reso difficile verificare l'eventuale introduzione di artefatti o alterazioni non realistiche nelle immagini.

In Fig. 4 sono riportate le distribuzioni delle classi post data augmentation.



(a) T1. (b) T2. (c) FLAIR.

Figura 4: Distribuzione delle classi per le tre modalità nel contesto multiclasse post data augmentation.

3.3 Preprocessing

Le immagini sono state ridimensionate a 256×256 pixel e scalate tra $[0, 1]$ tramite min-max scaling. Inoltre, trattandosi di MRI cerebrali, è stata valutata anche l'applicazione di una fase di *skull stripping*, ovvero la rimozione del tessuto non cerebrale dalle immagini, mediante l'uso di strumenti specialistici come ANTs [TCH+21] o BET [Smi02]. Tuttavia, non è stato possibile adottare questa procedura poiché tali strumenti richiedono dati volumetrici e metadati di acquisizione (ad esempio volume e profondità), assenti nei nostri dati, che consistono unicamente in immagini 2D in formato `.png`.

3.4 Modelli

CNN Il modello basato su CNN è costituito da due blocchi fondamentali: un *ConvBlock* e un *DenseBlock*. Il ConvBlock è layer sequenziale composto da una convoluzione 2D, seguita da un'attivazione ReLU, batch normalization e max pooling. Il DenseBlock, invence, è composto da un layer denso seguito da un'attivazione ReLU. Tra le due tipologie di blocchi è posto un layer di *flattening* per il passaggio delle informazioni tra i due. Entrambi i tipi di blocco sono ripetuti N volte, dove N è un iperparametro (si veda Sez. 4.1) per maggiori informazioni). In Fig. 5 è riportata l'architettura completa della rete. Nel caso di classificazione multiclasse, la funzione sigmoide è sostituita dalla softmax.

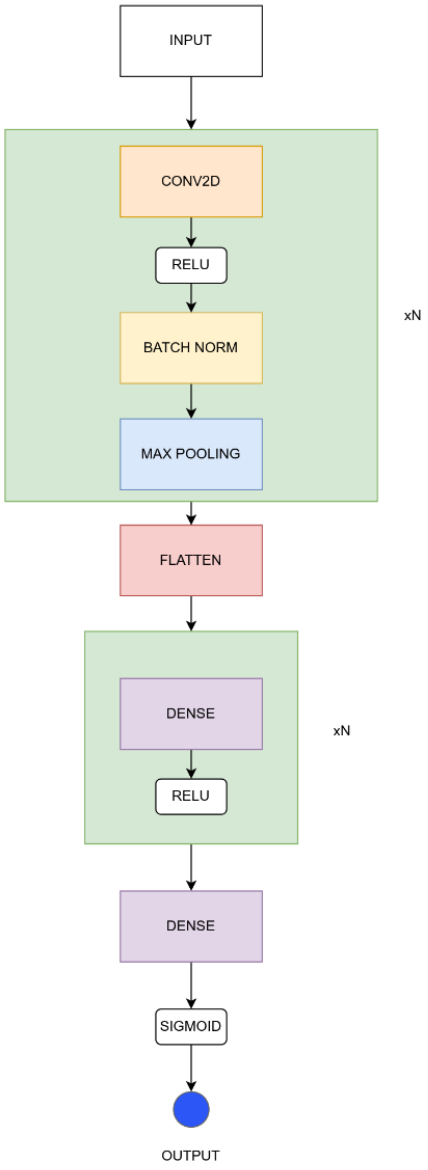


Figura 5: Architettura del modello basato su CNN.

ViT Il modello basato su Vision Transformer (ViT) è stato riprodotto in base a quanto riportato in [DBK+21]. Tuttavia, gli embeddings posizionali sono stati modellati come parametri apprendibili.

3.5 Funzioni di Perdita

CrossEntropy La CrossEntropy (CE) [ZS18] quantifica la differenza tra la distribuzione di probabilità prevista da un modello e la distribuzione reale della variabile target. È definita come segue:

$$\mathrm{CE}(y, \hat{y}) = -\sum_{i=1}^{N} \sum_{k \in \Omega} y_{ik} \log \hat{y}_{ik} \tag{1}$$

dove $y_{ik} \in \{0, 1\}$ indica se l'elemento i è di classe k, mentre $\hat{y}_{ik}$ è la probabilità prevista della classe k per l'elemento i, N è il numero di elementi e Ω è il numero di classi.

La CrossEntropy è una funzione di perdita molto diffusa utilizzata in una moltitudine di attività di classificazione, tuttavia soffre di squilibrio di classe e tende a dar maggior peso alla classe maggioritaria. Per tal motivo, è stata utilizzata solo nel contesto di classificazione binaria, dove lo squilibrio tra classi è trascurabile. Nel contesto di classificazione multiclasse, invece, si è esplorato l'utilizzo della FocalLoss sviluppata appositamente per gestire lo squilibrio tra classi.

FocalLoss La Focal Loss (FL) [LGG+18] è una variante della CrossEntropy progettata per affrontare problemi di squilibrio tra classi, riducendo l'influenza dei campioni classificati correttamente e concentrando l'attenzione del modello su quelli più difficili da classificare. La sua formulazione per il caso binario è la seguente:

$$\mathrm{FL}(p) = \begin{cases} -\alpha(1-p)^{\gamma} \log(p) & \text{se } y = 1, \\ -(1-\alpha)p^{\gamma} \log(1-p) & \text{se } y = 0, \end{cases} \tag{2}$$

dove $p \in [0, 1]$ rappresenta la probabilità prevista per la classe positiva, $\alpha \in [0, 1]$ è un fattore di bilanciamento tra le classi e $\gamma \geq 0$ è il parametro di focalizzazione che controlla quanto la perdita penalizzi maggiormente gli esempi difficili rispetto a quelli facili.

Nel caso multiclasse, la Focal Loss si estende sostituendo p con la probabilità $\hat{y}_{ik}$ assegnata alla classe corretta k per ciascun campione i, ottenendo:

$$\mathrm{FL}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k \in \Omega} \alpha_k (1 - \hat{y}_{ik})^{\gamma} y_{ik} \log(\hat{y}_{ik}), \tag{3}$$

dove α_k è il fattore di bilanciamento specifico per la classe k. L'effetto combinato di α e γ permette di mitigare lo squilibrio tra classi, evitando che le classi maggioritarie dominino il processo di apprendimento.

4 Risultati e Discussioni

In questa sezione vengono presentati, analizzati e discussi i risultati ottenuti sui vari task. Per garantire maggiore concisione e leggibilità nelle tabelle e nelle figure, verrà adottata la seguente nomenclatura per i modelli: *MODEL_TASK_MODALITY*. Ad esempio, un modello CNN addestrato su classificazione binaria sul dataset di sequenza MRI T1 assumerà il nome di **CNN_B_T1**, mentre un modello CNN addestrato su classificazione multiclasse sul dataset T1 assumerà il nome di **CNN_MC_T1**.

4.1 Setup Sperimentale

Configurazione e ambiente Tutti i modelli sono stati addestrati su un Apple Silicon M1 Pro con 16GB di RAM tramite Tensorflow [AAB+15] 2.19.0. Gli esperimenti sono stati tracciati tramite l'ausilio di Weights & Biases [Bie20].

Tutti i modelli sono stati addestrati per 2000 epoche con una batch size pari a 64 con ottimizzatore AdamW [LH19]. Per prevenire lo spreco di risorse quando le prestazioni del modello raggiungono un punto di plateau, è stato applicato l'early stopping con pazienza pari a 500 epoche.

Iperparametri Il tuning degli iperparametri è stato eseguito tramite ottimizzazione bayesiana [SLA12], generando in totale 10 configurazioni per ciascun modello e per ciascun task. Gli iperparametri considerati, insieme ai relativi intervalli o insiemi di valori di ricerca per ogni modello, sono riportati in Tab. 1.

La ricerca è stata condotta esclusivamente sul dataset contenente le sequenze T1, per due motivi principali:

- **Limitazioni computazionali**: Le risorse disponibili non avrebbero permesso di estendere il tuning a tutti i dataset senza tempi di ricerca eccessivamente lunghi.
- **Similarità tra dataset**: I tre dataset (T1, T2, FLAIR) presentano una natura simile dei dati. In linea con quanto riportato in [FKE+15, WSST16], si è scelto di effettuare l'ottimizzazione solo su T1, riutilizzando gli iperparametri ottenuti come punto di partenza per gli altri dataset, senza considerarli ottimali a priori.

In Tab. 2 sono riportate le combinazioni migliori degli iperparametri per ciascun modello e task.

Datasets Ogni dataset (T1, T2, FLAIR) è stato suddiviso, in maniera stratificata, in un insieme di addestramento, validazione e testing seguendo uno split $80\% - 10\% - 10\%$. Si è cercato di allocare quanti più dati possibili all'insieme di addestramento vista la scarsa disponibilità di dati, cercando di mantenere insieme di validazione e testing quanto più rappresentativi possibili. Di seguito si riportano le suddivisioni per ciascuna modalità:

- T1: Addestramento - 762, Validazione - 95, Testing - 96;
- T2: Addestramento - 771, Validazione - 96, Testing - 97;
- FLAIR: Addestramento - 804, Validazione - 101, Testing - 101;

Tabella 1: Intervalli degli iper-parametri considerati per CNN e ViT.

Iperparametro	**Intervallo**	**Modello**
Learning Rate	$[1e-5, 1e-2]$	CNN, ViT
Weight Decay	$[1e-6, 1e-3]$	CNN, ViT
Dropout	$[0.0, 0.5]$	CNN, ViT
N. Layer Convoluzionali	$\{2, 4, 8, 16\}$	CNN
N. Layer Densi	$\{1, 2, 4\}$	CNN
Unità dell'MLP	$\{64, 128, 256\}$	CNN
D. Model	$\{256, 512, 768\}$	ViT
Dim. Patch	$\{8, 16, 32\}$	ViT
N. Layer	$\{4, 6, 8, 12\}$	ViT
N. Teste	$\{4, 8, 12\}$	ViT
Unità dell'MLP	$\{512, 1024, 2048, 3072\}$	ViT

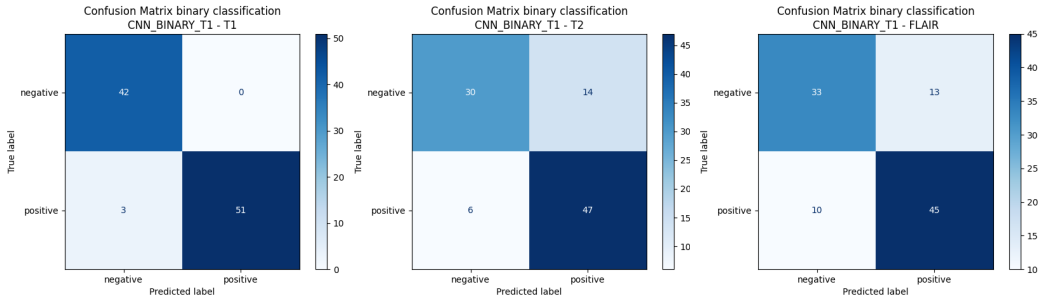
Tabella 2: Migliori configurazioni per modello e task. B = Binario, MC = Multiclasse.

	Task - Modello			
Iperparametro	B - CNN	B - ViT	MC - CNN	MC - ViT
Learning Rate	0.0005	0.002	0.003	0.0002
Weight Decay	0.00015	0.00005	0.0006	0.0009
Dropout	0.16639	0.074899	0.35309	0.01567
Unità MLP	128	3072	128	512
N. Layer Convoluzionali	8	-	8	-
N. Layer Densi	1	-	1	-
D. Model	-	768	-	768
Dim. Patch	-	32	-	16
N. Layer	-	12	-	8
N. Teste	-	12	-	8

4.2 Classificazione Binaria

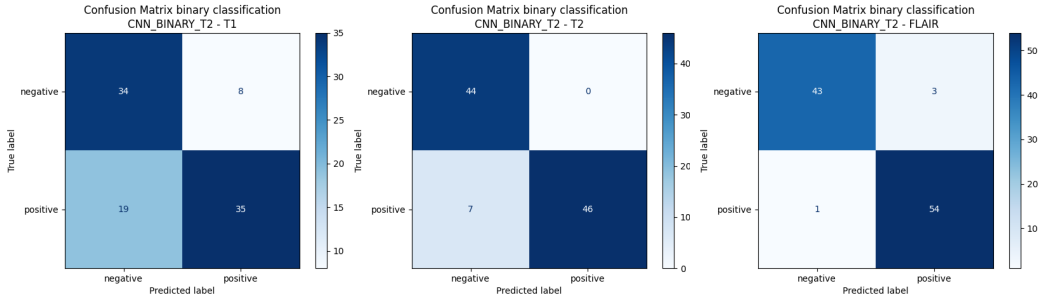
4.2.1 CNN

Nel contesto della classificazione binaria, tutti i modelli CNN hanno raggiunto prestazioni notevoli, superando in media la soglia del 90% su tutte le metriche in scenari intra-domain e mantenendosi tra il 70% e l' 80% nella valutazione cross-domain. In particolare, il modello `CNN_B_T1` ha dimostrato ottime capacità di discriminazione intra-domain, riuscendo a classificare quasi perfettamente tutte le istanze, con lievi confusioni di istanze positive classificate come negative. In ambito cross-domain, ha invece evidenziato difficoltà soprattutto nel riconoscere correttamente le istanze negative su FLAIR e T2 (Fig. 6). Un comportamento simile si osserva per `CNN_B_T2`, che mostra maggiore robustezza nel contesto cross-domain su FLAIR, ma prestazioni inferiori su T1, con frequenti errori di classificazione della classe positiva come negativa (Fig. 7).



(a) T1 (b) T2 (c) FLAIR

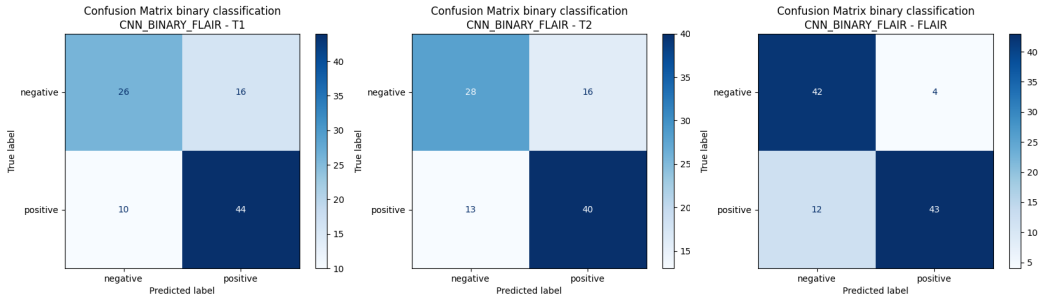
Figura 6: Matrici di confusione per il modello `CNN_B_T1` su diverse modalità.



(a) T1 (b) T2 (c) FLAIR

Figura 7: Matrici di confusione per il modello `CNN_B_T2` su diverse modalità.

Il modello `CNN_B_FLAIR` risulta il meno robusto tra i tre, con difficoltà marcate nella classificazione sia di istanze positive che negative per T1 e T2. Questo comportamento potrebbe essere legato alla natura informativa della modalità FLAIR, che enfatizza caratteristiche diverse rispetto a T1 e T2. Mentre queste ultime condividono una maggiore somiglianza nella rappresentazione anatomica e patologica complessiva, FLAIR tende a mettere in risalto specifici pattern legati alle lesioni, fornendo un contenuto informativo meno direttamente sovrapponibile alle altre modalità (Fig. 8).



(a) T1 (b) T2 (c) FLAIR

Figura 8: Matrici di confusione per il modello `CNN_B_FLAIR` su diverse modalità.

In generale, i modelli CNN hanno mostrato un'elevata efficacia, specialmente considerando la limitata disponibilità di dati visto che il numero di sequenze si aggira intorno

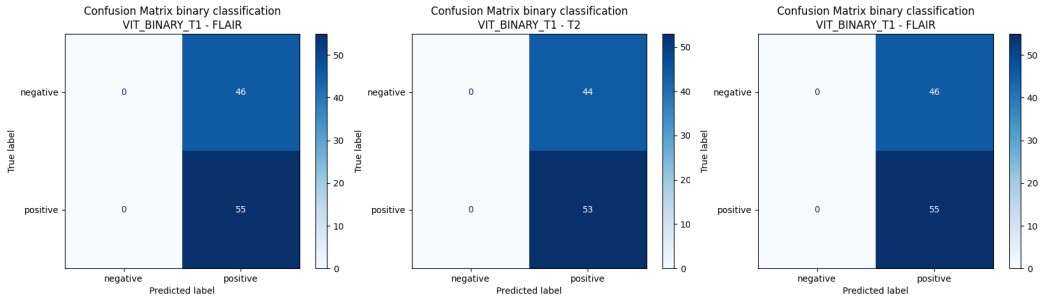
alle ≈ 1000 immagini per modalità. Questi risultati suggeriscono che le CNN, pur essendo architetture relativamente semplici rispetto alle controparti basate su Vision Transformer, risultano essere più *data-efficient*, riuscendo a raggiungere prestazioni elevate con soli ≈ 1.6M di parametri. Questa caratteristica è essenziale in ambiti come quello medico, dove i dati scarsessagiano a causa di costi elevati di acquisizione, annotazione e problemi relativi alla privacy dei pazienti e alla condivisione di dati tra istituzioni. In Tab. 3 sono riportate le metriche raggiunte dai singoli modelli in ogni contesto analizzato.

Tabella 3: Confronto tra gli approcci CNN binari sulle diverse modalità. A: Accuracy, F_1: F_1-score, P: Precision, R: Recall. I risultati migliori sono evidenziati in grassetto.

Modello	**Modalità**	A	P	R	F_1
CNN_B_T1	T1	**0.969**	**0.971**	**0.969**	**0.969**
CNN_B_T1	T2	0.794	0.799	0.794	0.791
CNN_B_T1	FLAIR	0.772	0.772	0.772	0.771
CNN_B_T2	T1	0.719	0.739	0.719	0.719
CNN_B_T2	T2	**0.928**	**0.938**	**0.928**	**0.928**
CNN_B_FLAIR	FLAIR	0.842	0.852	0.842	0.842
CNN_B_FLAIR	T1	0.729	0.728	0.729	0.726
CNN_B_FLAIR	T2	0.701	0.700	0.701	0.700
CNN_B_FLAIR	FLAIR	**0.960**	**0.961**	**0.960**	**0.960**

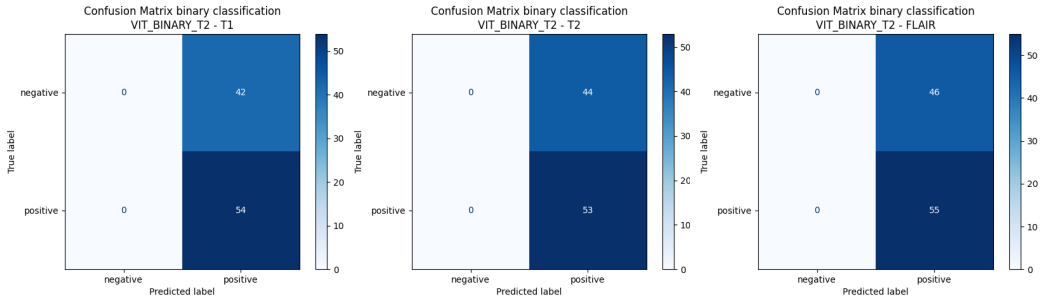
4.2.2 ViT

Per quanto concerne i modelli basati su ViT, questi hanno raggiunto prestazioni paragonabili a un classificatore casuale sfiorando la soglia del 50% in termini di accuratezza sia sulla valutazione intra-domain sia sulla valutazione cross-domain (Si veda Tab. 4). Analizzando le matrici di confusione dei vari modelli, in Fig. 10, 9, 11, si osserva che tutti i modelli predicono esclusivamente la classe positiva.



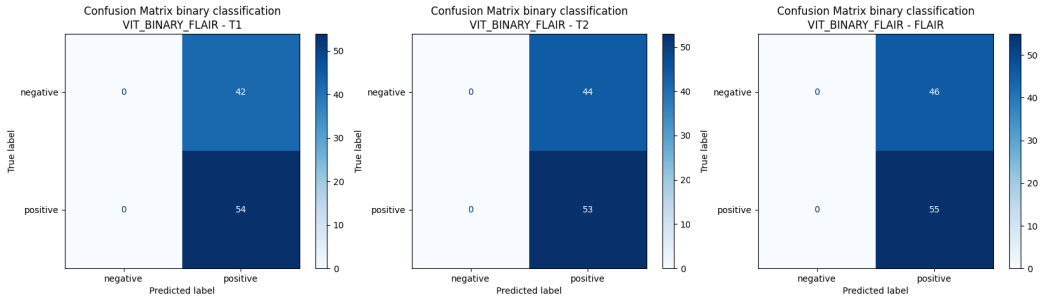
(a) T1 (b) T2 (c) FLAIR

Figura 9: Matrici di confusione per il modello `VIT_B_T1` su diverse modalità.



(a) T1 (b) T2 (c) FLAIR

Figura 10: Matrici di confusione per il modello `VIT_B_T2` su diverse modalità.

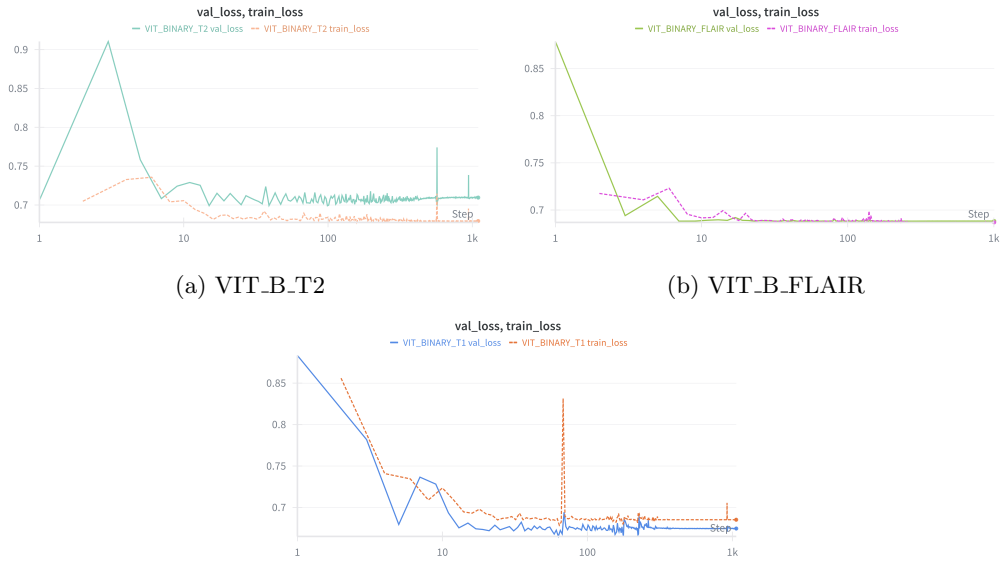


(a) T1 (b) T2 (c) FLAIR

Figura 11: Matrici di confusione per il modello `VIT_B_FLAIR` su diverse modalità.

Osservando le curve di apprendimento dei modelli in Fig. 12, emergono due comportamenti distinti. Per `VIT_B_T2` (Fig.12a) si nota un lieve overfitting nella fase finale di addestramento, suggerendo che tale modello possa aver parzialmente memorizzato i dati di addestramento, riducendo la propria capacità di generalizzazione. Al contrario, `VIT_B_FLAIR` (Fig.12b) e `VIT_B_T1` (Fig.12c) mostrano curve stabili, con perdite di addestramento e validazione simili e senza una divergenza evidente. Tuttavia, anche questi modelli presentano le stesse prestazioni e il medesimo bias verso la classe positiva, segno di un overfitting "strutturale" più che "classico".

Tale fenomeno può essere spiegato considerando la complessità dei modelli ViT, che contano circa 68M di parametri contro gli 1.6M dei corrispettivi basati su CNN. Tale complessità richiede un numero di dati maggiore per apprendere feature discriminative robuste. Con un dataset limitato, i modelli ViT possono ottenere buone prestazioni in una validazione interna, ma fallire quando si verifica un distribution shift (dato che il set di validazione è stato generato a partire da quello di addestramento), rivelando così la scarsa capacità di generalizzazione reale.



(a) VIT_B_T2

(b) VIT_B_FLAIR

(c) VIT_B_T1

Figura 12: Curve di apprendimento dei vari modelli binari basati su ViT.

Tabella 4: Confronto tra gli approcci ViT binari sulle diverse modalità. A: Accuracy, F_1: F_1-score, P: Precision, R: Recall. I risultati migliori sono evidenziati in grassetto.

Modello	**Modalità**	A	P	R	F_1
VIT_B_T1	T1	**0.563**	**0.316**	**0.563**	**0.405**
VIT_B_T1	T2	0.546	0.299	0.546	0.386
VIT_B_T1	FLAIR	0.545	0.297	0.545	0.384
VIT_B_T2	T1	**0.563**	**0.316**	**0.563**	**0.405**
VIT_B_T2	T2	0.546	0.299	0.546	0.386
VIT_B_T2	FLAIR	0.545	0.297	0.545	0.384
VIT_B_FLAIR	T1	**0.563**	**0.316**	**0.563**	**0.405**
VIT_B_FLAIR	T2	0.546	0.299	0.546	0.386
VIT_B_FLAIR	FLAIR	0.545	0.297	0.545	0.384

4.3 Classificazione Multiclasse

Nella seguente sezione, introduciamo una modifica alla nomeclatura dei modelli. In aggiunta a quanto introdotto all'inizio della Sez. 4, oltre al task e alla modalità, nel nome dei modelli comparirà `AUG` per indicare che il modello è stato addestrato su dati aumentati e Focal-Loss (Eq. 3); i modelli senza questa dicitura invece sono stati addestrati sulla distribuzione normale dei dati (Sez. 3.1) usando solo la FocalLoss per gestire lo sbilanciamento dei dati.

Si precisa, inoltre, che nella valutazione multiclasse è stata impiegata una *media micro* per il calcolo delle metriche.

4.3.1 CNN

Nel contesto multiclasse, i modelli CNN addestrati senza data augmentation mostrano buone prestazioni in ambito *intra-domain*, raggiungendo la soglia del 90% su tutte le metriche

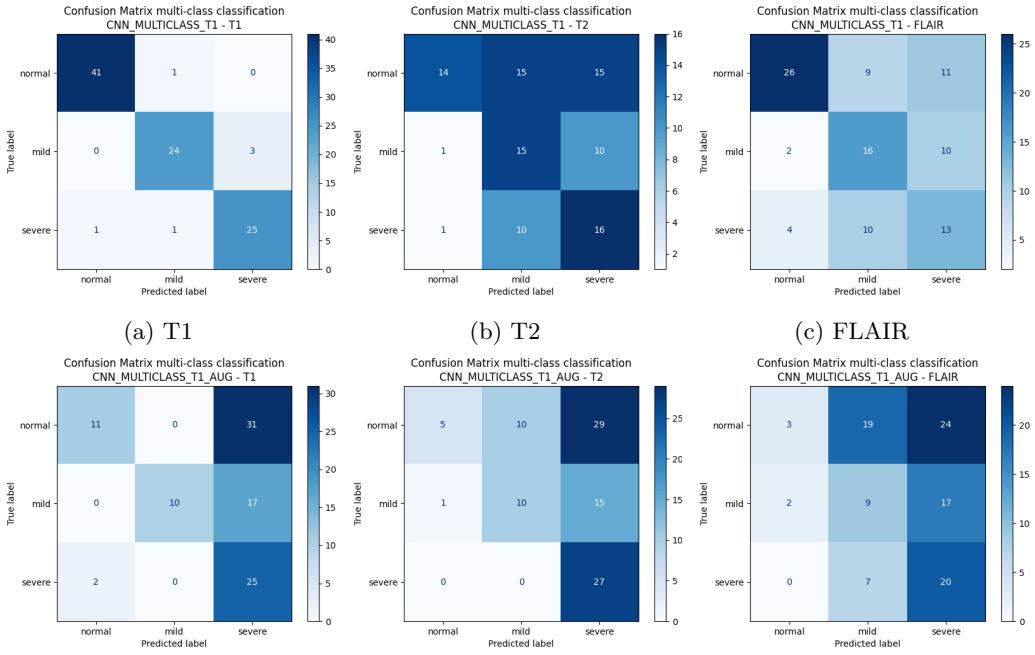
considerate, ad eccezione del modello `CNN_MC_T2`, che si ferma all'84%. La situazione cambia in ambito *cross-domain*, dove l'accuracy media scende al 55.6%, a fronte di un'accuracy media *cross-domain* del 75.95% osservata nel contesto binario. Questo calo va a confermare che il task multiclasse sia intrinsecamente più complesso e che i modelli, dovendo distinguere sfumature di disabilità più sottili, tendano ad imparare *feature* discriminanti specifiche della modalità di addestramento che sono meno trasferibili da una modalità all'altra.

Tabella 5: Confronto tra gli approcci CNN multiclasse basati sulla sola FocalLoss (NO_AUG) e l'approccio basato su data augmentation + FocalLoss (AUG). A: Accuracy, F_1: Micro-F_1, P: Micro-Precision, R: Micro-Recall. I risultati migliori sono evidenziati in grassetto.

		NO_AUG				**AUG**			
Modello	**Modalità**	A	P	R	F_1	A	P	R	F_1
CNN_MC_T1	T1	**0.938**	**0.938**	**0.938**	**0.938**	0.479	0.479	0.479	0.479
CNN_MC_T2	T1	0.594	0.594	0.594	0.594	**0.604**	**0.604**	**0.604**	**0.604**
CNN_MC_FLAIR	T1	0.542	0.542	0.542	0.542	0.552	0.552	0.552	0.552
CNN_MC_T1	T2	0.464	0.464	0.464	0.464	0.433	0.433	0.433	0.433
CNN_MC_T2	T2	**0.845**	**0.845**	**0.845**	**0.845**	**0.804**	**0.804**	**0.804**	**0.804**
CNN_MC_FLAIR	T2	0.577	0.577	0.577	0.577	0.443	0.443	0.443	0.443
CNN_MC_T1	FLAIR	0.545	0.545	0.545	0.545	0.317	0.317	0.317	0.317
CNN_MC_T2	FLAIR	0.614	0.614	0.614	0.614	0.634	0.634	0.634	0.634
CNN_MC_FLAIR	FLAIR	**0.911**	**0.911**	**0.911**	**0.911**	**0.653**	**0.653**	**0.653**	**0.653**

I risultati per i modelli addestrati con dati aumentati artificialmente sono sorprendenti visto che mostrano prestazioni ridotte sia in *intra-domain* che in *cross-domain*, spesso comparabili a quelle di un classificatore casuale. Un'eccezione parziale è il modello `CNN_MC_T2_AUG`, che supera il corrispettivo modello senza data augmentation in cross-domain su T1 e si comporta meglio di `CNN_MC_T1_AUG`. L'analisi delle matrici di confusione (Fig. 13, 14, 15) rivela che questi modelli, in genere, classificano correttamente la classe *normal*, ma faticano a distinguere tra *mild* e *severe*. Ciò suggerisce che le trasformazioni applicate non siano state sufficienti a indurre un apprendimento bilanciato delle classi, anzi sembrano aver avuto l'effetto opposto introducendo informazioni ridondanti o rumore, riducendo la capacità del modello di apprendere rappresentazioni realmente utili per la discriminazione fine tra le classi. A sostengo di ciò, in Fig. 16 si riportano le curve di apprendimento dei tre modelli che mostrano chiaramente un apprendimento instabile con picchi alti e frequenti nella perdita di validazione e una discostamento generale dalla perdita di addestramento, evidenziando chiari segni di difficoltà nel generalizzare.

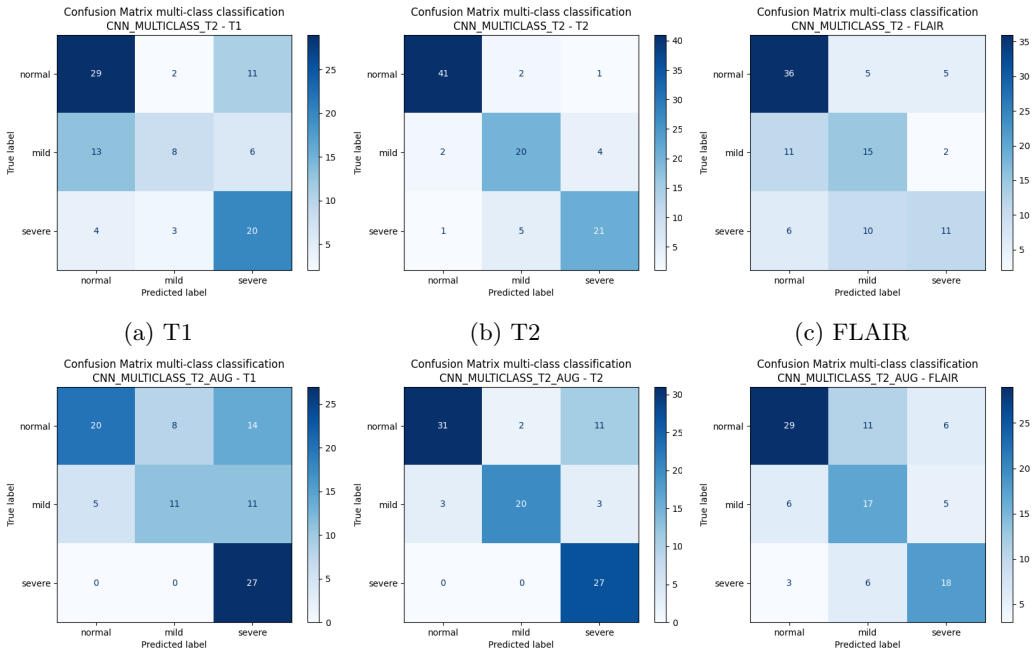
In Tab. 5 è riportato il confronto tra i modelli CNN con e senza data augmentation.



(a) T1 (b) T2 (c) FLAIR

(d) T1_AUG (e) T2_AUG (f) FLAIR_AUG

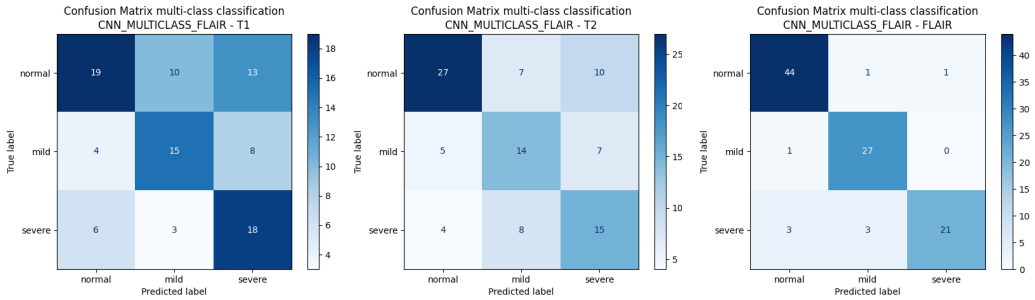
Figura 13: Matrici di confusione per i modelli `CNN_MC_T1` e `CNN_MC_T1_AUG` su diverse modalità.



(a) T1 (b) T2 (c) FLAIR

(d) T1_AUG (e) T2_AUG (f) FLAIR_AUG

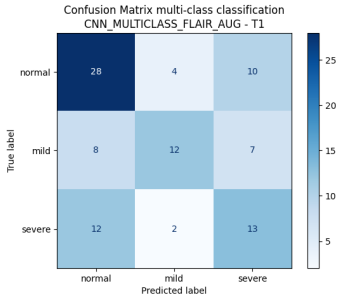
Figura 14: Matrici di confusione per i modelli `CNN_MC_T2` e `CNN_MC_T2_AUG` su diverse modalità.



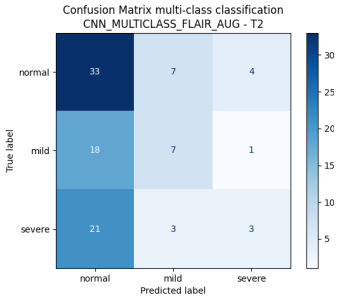
(a) T1

(b) T2

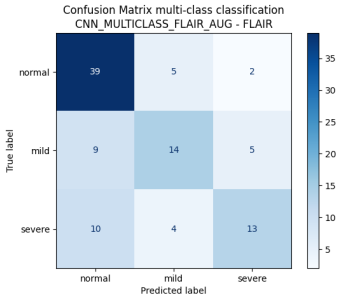
(c) FLAIR



(d) T1_AUG

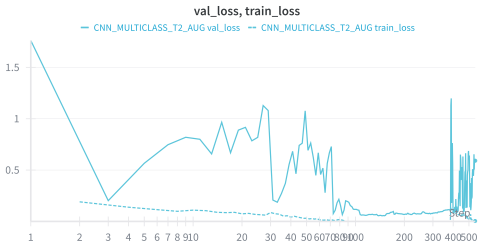


(e) T2_AUG

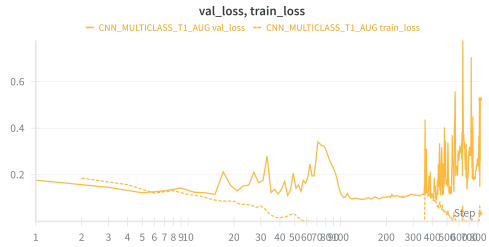


(f) FLAIR_AUG

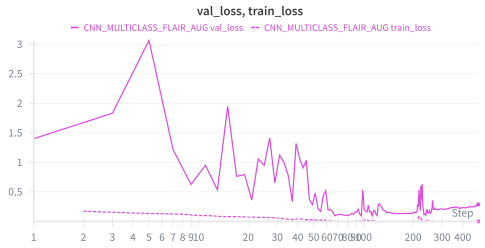
Figura 15: Matrici di confusione per i modelli `CNN_MC_FLAIR` e `CNN_MC_FLAIR_AUG` su diverse modalità.



(a) CNN_MC_T2_AUG



(b) CNN_MC_T1_AUG



(c) CNN_MC_FLAIR_AUG

Figura 16: Curve di apprendimento dei vari modelli multiclasse basati su CNN con data augmentation.

4.3.2 ViT

Nel contesto multiclasse, i modelli ViT addestrati senza data augmentation hanno mostrato un netto miglioramento in ambito *intra-domain*, raggiungendo in media la soglia del 75.2% su tutte le metriche considerate, rispetto alla controparte binaria che evidenzia prestazioni prossime al casuale. La situazione in ambito *cross-domain* rimane invece critica, con prestazioni che, come nel caso dei modelli CNN, si avvicinano a decisioni casuali. Ciò rafforza quanto osservato in precedenza sulla maggiore difficoltà del task multiclasse e sulla limitata trasferibilità delle feature apprese tra una modalità e l'altra.

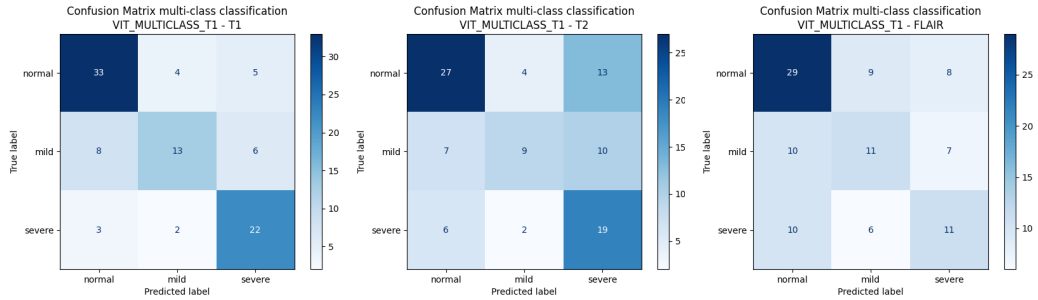
Il miglioramento dei ViT potrebbe essere parzialmente attribuito alla minore complessità dei modelli multiclasse ($\approx$ 26M di parametri) rispetto a quelli binari ($\approx$ 68M), confermando quanto già evidenziato per i modelli binari: una maggiore dimensionalità richiede più dati per una generalizzazione efficace, mentre i modelli multiclasse, essendo più compatti, riescono a generalizzare meglio con il numero limitato di dati disponibili.

Tabella 6: Confronto tra gli approcci ViT multiclasse basati sulla sola FocalLoss (NO_AUG) e l'approccio basato su data augmentation + FocalLoss (AUG). A: Accuracy, F_1: Micro-F_1, P: Micro-Precision, R: Micro-Recall. I risultati migliori sono evidenziati in grassetto.

		NO_AUG				**AUG**			
Modello	**Modalità**	A	P	R	F_1	A	P	R	F_1
ViT_MC_T1	T1	**0.708**	**0.708**	**0.708**	**0.708**	**0.552**	**0.552**	**0.552**	**0.552**
ViT_MC_T2	T1	0.573	0.573	0.573	0.573	0.479	0.479	0.479	0.479
ViT_MC_FLAIR	T1	0.469	0.469	0.469	0.469	0.448	0.448	0.448	0.448
ViT_MC_T1	T2	0.567	0.567	0.567	0.567	0.536	0.536	0.536	0.536
ViT_MC_T2	T2	**0.835**	**0.835**	**0.835**	**0.835**	**0.505**	**0.505**	**0.505**	**0.505**
ViT_MC_FLAIR	T2	0.515	0.515	0.515	0.515	0.485	0.485	0.485	0.485
ViT_MC_T1	FLAIR	0.505	0.505	0.505	0.505	0.475	0.475	0.475	0.475
ViT_MC_T2	FLAIR	0.644	0.644	0.644	0.644	**0.554**	**0.554**	**0.554**	**0.554**
ViT_MC_FLAIR	FLAIR	**0.713**	**0.713**	**0.713**	**0.713**	0.495	0.495	0.495	0.495

Per quanto riguarda l'approccio basato su data augmentation, i risultati evidenziano una tendenza analoga, se non più marcata, a quella osservata nei modelli CNN: le tecniche adottate non si sono rivelate sufficienti a favorire un apprendimento bilanciato tra le classi. Al contrario, sembrano aver prodotto l'effetto opposto, introducendo informazioni ridondanti o rumore, e spingendo i modelli a concentrarsi prevalentemente sulla classe *normal* nonostante l'oversampling mirato delle classi *mild* e *severe*.

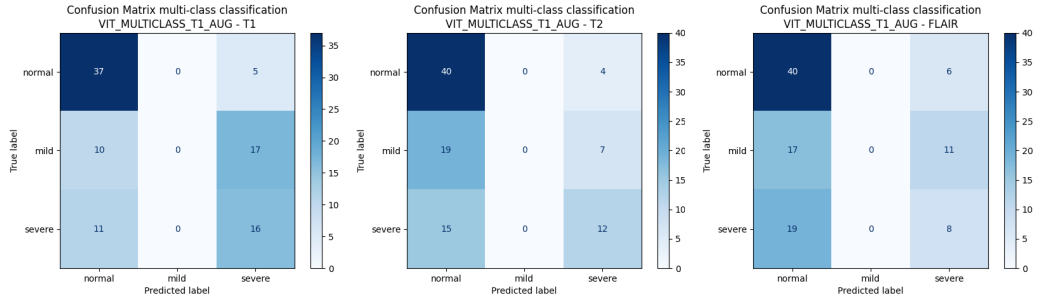
In Tab. 6 è riportato il confronto tra i due approcci considerati per i modelli ViT.



(a) T1

(b) T2

(c) FLAIR

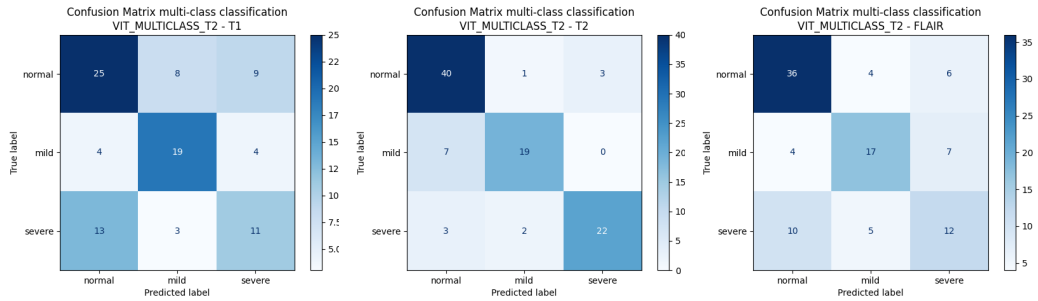


(d) T1_AUG

(e) T2_AUG

(f) FLAIR_AUG

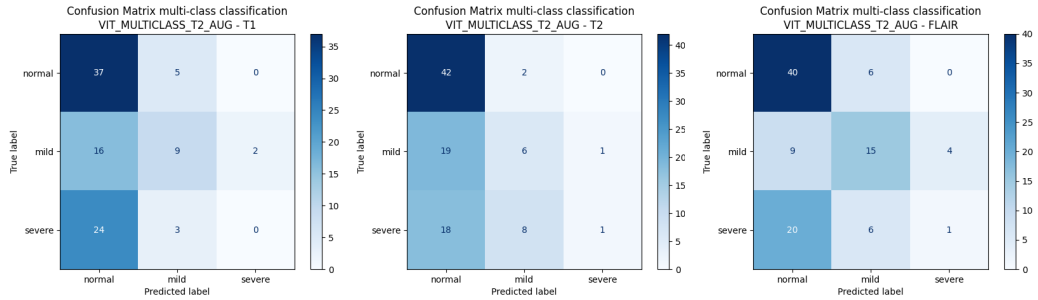
Figura 17: Matrici di confusione per i modelli `VIT_MC_T1` e `VIT_MC_T1_AUG` su diverse modalità.



(a) T1

(b) T2

(c) FLAIR

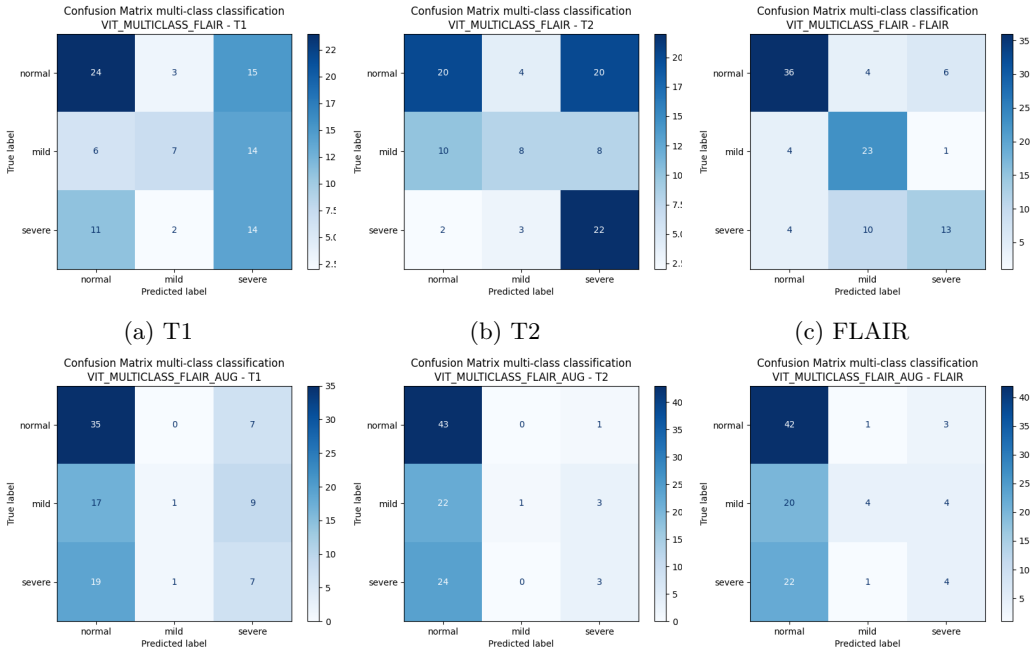


(d) T1_AUG

(e) T2_AUG

(f) FLAIR_AUG

Figura 18: Matrici di confusione per i modelli `VIT_MC_T2` e `VIT_MC_T2_AUG` su diverse modalità.



(a) T1 (b) T2 (c) FLAIR

(d) T1_AUG (e) T2_AUG (f) FLAIR_AUG

Figura 19: Matrici di confusione per i modelli `VIT_MC_FLAIR` e `VIT_MC_FLAIR_AUG` su diverse modalità.

5 Conclusioni

In questo lavoro è stata affrontata la classificazione del grado di disabilità in pazienti affetti da sclerosi multipla a partire da sequenze MRI, sia in un contesto binario che multiclasse, confrontando architetture leggere basate su CNN e Vision Transformer (ViT). Le analisi hanno evidenziato che:

- Le CNN leggere ($\approx$ 1.6M di parametri) offrono ottime prestazioni in scenari con quantità limitate di dati, raggiungendo $> 90\%$ di accuratezza in ambito *intra-domain* e mantenendo, nel contesto binario, una discreta robustezza *cross-domain*.
- I ViT soffrono nel regime di dati ridotti, con prestazioni prossime al casuale nel binario, ma mostrano miglioramenti nel multiclasse grazie a una riduzione del numero di parametri ($\approx$ 26M).
- L'utilizzo della sola FocalLoss si è dimostrato efficace nel gestire lo sbilanciamento delle classi nel contesto multiclasse. Al contrario, l'approccio combinato data augmentation + Focal Loss non ha portato benefici, degradando spesso le prestazioni, probabilmente a causa dell'introduzione di variabilità non informativa o rumore.
- La classificazione *cross-domain* rimane complessa in entrambi i paradigmi, suggerendo una forte dipendenza delle feature apprese dalla specifica modalità di acquisizione.

In sintesi, le CNN leggere si confermano una scelta efficace e più data-efficient in contesti clinici reali caratterizzati da dataset ridotti e vincoli computazionali, mentre i ViT richiedono ulteriori strategie di pretraining o fine-tuning per esprimere appieno il loro potenziale.

Come possibili sviluppi futuri si propongono:

- Utilizzo di modelli ViT pre-addestrati e fine-tuning sul dataset in esame.
- Approccio multi-modale per migliorare le capacità *cross-domain* dei modelli, combinando durante l'addestramento le feature estratte da ogni modalità.
- Esplorazione di tecniche di data augmentation più specifiche e supportate da evidenze mediche.
- Analisi qualitativa tramite mappe di attenzione (Grad-CAM o simili) per interpretare le decisioni del modello e valutarne la coerenza anatomica.

Riferimenti bibliografici

[AAB+15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[Bie20] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[DBK+21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[FKE+15] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[Gri23] Antonio Griguolo. Sclerosi multipla: cause, sintomi e come si cura, 2023.

[LGG+18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[LH19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[Mul25] Associazione Italiana Sclerosi Multipla. Edss - scala, 2025.

[SLA12] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms, 2012.

[Smi02] S. M. Smith. Fast robust automated brain extraction. *Human Brain Mapping*, 17(3):143–155, November 2002.

[TCH+21] Nicholas J. Tustison, Philip A. Cook, Andrew J. Holbrook, Hans J. Johnson, John Muschelli, Gabriel A. Devenyi, Jeffrey T. Duda, Sandhitsu R. Das, Nicholas C. Cullen, Daniel L. Gillen, Michael A. Yassa, James R. Stone, James C. Gee, and Brian B. Avants. The ANTsX ecosystem for quantitative biological and medical imaging. *Scientific Reports*, 11(1):9068, April 2021.

[WSST16] Martin Wistuba, Nicolas Schilling, and Lars Schmidt-Thieme. Two-stage transfer surrogate model for automatic hyperparameter optimization. In *ECML/PKDD (1)*, pages 199–214, 2016.

[ZS18] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.