POLITECNICO DI TORINO

Master Degree course in Biomedical Engineering

Master Degree Thesis

Uso di imaging e dati clinici nella predizione dell'andamento della capacità polmonare in pazienti con fibrosi

Supervisors
Prof. Gabriella Balestra
Prof. Samanta Rosati
Dot.ssa Federica Amato
Dott. Marco Bologna

Candidato
Domenico Emanuele Alessandria

Academic Year 2023-2024

Abstract

This research aims to analyze the effectiveness of the combined use of clinical data and CT (Computerized Tomography) images in predicting the course of lung capacity in patients with Idiopathic Pulmonary Fibrosis (IPF), a chronic respiratory system disease. Despite efforts in medical research, the exact causes of IPF remain unknown. The investigation relies on the use of intelligent models to address diagnostic and prognostic challenges related to IPF, with a particular focus on the use of Vision Transformers (ViT), aiming to surpass the winning models of the "OSIC Pulmonary Fibrosis Progression" challenge on Kaggle.

The dataset used includes 176 chest CT scans and tabular features, including Forced Vital Capacity (FVC) values acquired during follow-up visits. Contrary to expectations, the analysis of trained models revealed the presence of systematic bias. Ablation tests and correlation were conducted to explore possible causes of this bias. The results indicate that models trained with ViT did not outperform existing reference models, and bias analysis emphasized a lack of significant learning from inputs, with model predictions based on statistically more likely values.

In conclusion, this study highlights that, using exclusively CT images and tabular data, it was not possible to achieve an optimal regression task in predicting the course of lung capacity in patients with IPF. Further research is needed to address the presented challenges, exploring alternative approaches or integrating additional information to improve prediction accuracy.

Sommario

Questa ricerca si propone di analizzare l'efficacia dell'utilizzo combinato di dati clinici e immagini TC (Tomografia Computerizzata) per la previsione dell'andamento della capacità polmonare in pazienti affetti da fibrosi polmonare idiopatica (IPF), una malattia cronica del sistema respiratorio. Nonostante gli sforzi nella ricerca medica, le cause esatte della IPF rimangono ancora sconosciute. L'indagine si basa sull'impiego di modelli intelligenti per affrontare le sfide diagnostiche e prognostiche legate all'IPF, con particolare attenzione all'utilizzo dei Vision Transformers (ViT), al fine di superare i modelli vincitori della challenge "OSIC Pulmonary Fibrosis Progression" su Kaggle. Il dataset utilizzato comprende 176 TC toraciche e feature tabulari, inclusi i valori di Forced Vital Capacity (FVC) acquisiti durante le visite di follow-up. Nonostante le aspettative, l'analisi dei modelli addestrati ha rivelato la presenza di un bias sistematico. Test di ablazione e correlazione sono stati condotti per esplorare le possibili cause di tale bias. I risultati indicano che i modelli addestrati con i ViT non hanno superato i modelli di riferimento esistenti, e l'analisi del bias ha sottolineato la mancanza di apprendimento significativo dagli input, con le previsioni dei modelli basate su valori statisticamente più probabili. In conclusione, questo studio evidenzia che, utilizzando esclusivamente immagini TC e dati tabulari, non è stato possibile raggiungere un ottimale task di regressione per predire l'andamento della capacità polmonare nei pazienti affetti da IPF. Ulteriori ricerche sono necessarie per affrontare le sfide presentate, esplorando approcci alternativi o integrando ulteriori informazioni per migliorare la precisione delle previsioni.

Elenco degli Acronimi

BPCO Broncopneumopatia Cronica Ostruttiva

CM Confusion Matrix

CNN Convolutional Neural Network

FVC Forced Vital Capacity

GPU Graphics Processing Unit

GT Ground-Truth

IA Intelligenza Artificiale

IPF Idiopathic Pulmonary Fibrosis

LLLm Laplace Log-Likelihood modificata

LR Learning Rate

ML Machine Learning

OSIC Open Source Imaging Consortium

RBF Radial Basis Functions

SVD Singular Value Decomposition

TC Tomografia Computerizzata

ViT Vision Trasformer

WB Weights & Biases

Indice

Elenco delle figure VII

Elenco delle tabelle IX

1 Introduzione e Stato Dell'Arte 1
- 1.1 Vantaggi dell'IA . . . 2
- 1.2 Analisi della Letteratura . . . 2

2 Materiali 7
- 2.1 Dataset . . . 7
- 2.2 Strumenti Software/Hardware Utilizzati . . . 8
  - 2.2.1 Tool per lo Sviluppo dell'Algoritmo . . . 9
  - 2.2.2 Modelli Impiegati . . . 11
  - 2.2.3 Stumenti Hardware . . . 13

3 Metodi e Risultati: Challenge Kaggle 15
- 3.1 Esperimento 1: Riproduzione del Modello Proposto dal Vincitore della Challenge . . . 16
- 3.2 Esperimento 2: Riproduzione del Modello Fibro-CoSANet . . . 16
- 3.3 Esperimento 3: Utilizzo dei ViT in Fibro-CoSANet . . . 18
  - 3.3.1 Workflow . . . 18
  - 3.3.2 Data Preparation . . . 19
  - 3.3.3 Model Training . . . 20
  - 3.3.4 Risultati . . . 21
- 3.4 Esperimento 4: Rimozione Volume Polmonare . . . 22
  - 3.4.1 Data Preparation . . . 22
  - 3.4.2 Model Training . . . 22
  - 3.4.3 Risultati . . . 22
- 3.5 Esperimento 5: ViT con Immagine Mediata . . . 23
  - 3.5.1 Data Preparation . . . 24
  - 3.5.2 Model Training . . . 24
  - 3.5.3 Risultati . . . 26
- 3.6 Esperimento 6: FVC come Valore Tabulare . . . 26
  - 3.6.1 Data Preparation . . . 27
  - 3.6.2 Model Training . . . 27
  - 3.6.3 Risultati . . . 28
- 3.7 Esperimento 7: Utilizzo di Scheduler . . . 28
  - 3.7.1 Data Preparation . . . 29
  - 3.7.2 Model Training . . . 29

3.7.3 Risultati . . . 30
3.8 Esperimento 8: Crop del Volume Originale . . . 31
3.8.1 Data Preparation . . . 31
3.8.2 Model Training . . . 32
3.8.3 Risultati . . . 33
3.9 Esperimento 9: Segmentazione dell'Immagine . . . 34
3.9.1 Data Preparation . . . 34
3.9.2 Model Training . . . 36
3.9.3 Risultati . . . 36
3.10 Esperimento 10: Modifica Normalizzazione Età ed Embedding dei Modelli . . . 37
3.10.1 Data Preparation . . . 37
3.10.2 Model Training . . . 37
3.10.3 Risultati . . . 38
3.11 Esperimento 11: Utilizzo delle Radial Basis Functions . . . 38
3.11.1 Data Preparation . . . 39
3.11.2 Model Training . . . 39
3.11.3 Risultati . . . 39

4 Metodi e Risultati: Confutazione del Task 41
4.1 Esperimento 1: Correlazione tra Feature Tabulari e Ground Truth . . . 42
4.1.1 Correlazione . . . 42
4.1.2 Risultati . . . 43
4.2 Esperimento 2: Training Senza Feature Tabulari . . . 43
4.2.1 Model Training . . . 43
4.2.2 Risultati . . . 43
4.3 Esperimeto 3: Training con Solo Immagini Casuali . . . 44
4.3.1 Model Training . . . 44
4.3.2 Risultati . . . 44
4.4 Esperimento 4: Variazione del Ground Truth . . . 45
4.4.1 Model Training . . . 45
4.4.2 Risultati . . . 45
4.5 Esperimento 5: Test di Ablazione Modificando il Ground Truth . . . 47
4.5.1 Analisi dei Diversi Ground Truth . . . 47
4.5.2 Training e Test di Ablazione con il Ground Truth 1 . . . 49
4.5.3 Training e Test di Ablazione con il Ground Truth 2 . . . 51
4.5.4 Test di Correlazione . . . 52
4.5.5 Confronto Ground Truth e Predizione . . . 53
4.6 Esperimento 6: Modifica del Modello . . . 56
4.6.1 Data Preparation . . . 56
4.6.2 Model Training . . . 56
4.6.3 Risultati . . . 56
4.7 Esperimento 7: Utilizzo delle Radial Basis Functions . . . 58
4.7.1 Metodologia . . . 58
4.7.2 Model Training . . . 58
4.7.3 Risultati e Analisi . . . 58

5 Metodi e Risultati: Task di classificazione 61
5.1 Esperimento 1:Modifica del Ground Truth e Classificazione 61
5.1.1 Modifica Ground Truth 61
5.1.2 Training e Risultati 61
5.2 Esperimento 2: Leave One Out 63
5.2.1 Training e Risultati 63
5.3 Esperimento 3: Leave One Out e Up-Weighting 65
5.3.1 Training e Risultati 65
5.3.2 Training e Risultati con Downsampling 66

6 Conclusione 69

Elenco delle figure

1.1 Architettura Fibrosis-Net [26] . . . 4
1.2 Architettura FVC-NET [28] . . . 4

2.1 Istogramma del numero di pazienti . . . 8
2.2 Architettura ViT modificato da [27] . . . 12
2.3 Struttura RBF [24] . . . 13

3.1 Diagramma a blocchi . . . 15
3.2 Architettura Fibro-CoSANet . . . 17
3.3 Doppio flusso Fibro-CoSANet . . . 17
3.4 Workflow . . . 18
3.5 Architettura Fully Connected . . . 20
3.6 Sovrapposizione maschera con immagine reale . . . 23
3.7 Slice mediata . . . 24
3.8 Score nei 5 Fold . . . 25
3.9 Loss di training e validation . . . 26
3.10 Scheduler Multistep per un solo fold . . . 29
3.11 Scheduler Linear per un solo fold . . . 30
3.12 Esempio Slice media 55% . . . 31
3.13 L1 loss Training e Validation dei 5 fold . . . 32
3.14 Score dei 5 fold . . . 33
3.15 Applicazione segmentazione . . . 35
3.16 Confroto Slice con e senza segmentazione . . . 36

4.1 Workflow confutazione del task. . . . 42
4.2 Confronto Best Model e Best Model Senza TAB . . . 44
4.3 Confronto Best Model e Modello con immagini casuali . . . 45
4.4 Confronto Best Model e Modello con immagini casuali e GT casuale . . . 46
4.5 Analisi GT per diversi pazienti . . . 46
4.6 confronto tra i vari GT . . . 48
4.7 Box plot distribuzione MAE per i 4 GT . . . 48
4.8 Confronto Score tra il Training con GT1 e Best Model . . . 49
4.9 Confronto tra 3 diversi training con GT1 . . . 50
4.10 Confronto tra training con GT2 e GT1 . . . 51
4.11 Correlation Heatmap . . . 52
4.12 Istogramma GT0 . . . 53
4.13 Scatterplot valori GT1 . . . 54
4.14 Confronto GT1 . . . 55

4.15 Istogrammi delle previsioni dei 5 fold usando GT0 56
4.16 Confronto tra i modelli che utilizzano ResNet18. 57

5.1 CM esperimento 1 62
5.2 Leave One Out 63
5.3 CM metodo Leave One Out 64
5.4 CM esperimento 3 65
5.5 CM esperimento 3 con downsampling 66

Elenco delle tabelle

2.1 Esempio di riga dei file Train.csv e Test.csv. 8

3.1 Differenti Batch Size e Leraning Rate con Score relativo 21
3.2 Submission esperimento 3 21
3.3 Submission esperimento 4 22
3.4 Differenti Batch Size e Leraning Rate con Score relativo Immagine media 25
3.5 Submission esperimento 5 26
3.6 Differenti Batch Size e Leraning Rate con Score relativo FVC iniziale 27
3.7 Submission esperimento 6 28
3.8 Submission con Multistep 30
3.9 Submission con Linear 30
3.10 Differenti Batch Size e Leraning Rate con Score relativo 55% del Volume 32
3.11 Submission esperimento 8 33
3.12 Differenti Batch Size e Leraning Rate con Score relativo Segmentazione 36
3.13 Differenti Batch Size e Leraning Rate con Score relativo esperimento 10 37
3.14 Submission esperimento 10 fold 4 38
3.15 Submission esperimento 10 merge dei 5 modelli 38
3.16 Submission esperimento 11 39
3.17 Training variazione di Sigma e Num Centers 40

4.1 Correlazioni e P-value tra Slope e Feature tabulari 43
4.2 Submission RBF e input corretti 58
4.3 Submission RBF e input casuali 58

5.1 Metriche esperimento 1 62
5.2 Metriche esperimento 2 64
5.3 Metriche esperimento 3 66
5.4 Metriche esperimento 3 con downsampling 67

Capitolo 1

Introduzione e Stato Dell'Arte

La Fibrosi Polmonare Idiopatica (IPF) è una patologia cronica del tessuto polmonare, caratterizzata dalla formazione progressiva di cicatrici che compromettono la funzione respiratoria. Questa condizione, prevalentemente riscontrata in individui sopra i 50 anni, presenta una maggiore incidenza negli uomini e rappresenta una sfida medica[11].

L'origine precisa della IPF rimane sconosciuta, ma è ritenuta risultato di una combinazione di fattori genetici e esposizione a sostanze inalate nel corso della vita[11]. Gli esiti di questa malattia includono affanno progressivo, tosse persistente e, in alcuni casi, deformità delle dita (ippocratismo digitale)[11].

La diagnosi si avvale di esami di imaging avanzati, come la TC (Tomografia Computerizzata) ad alta risoluzione, e in taluni casi richiede una biopsia polmonare. Il trattamento, focalizzato sulla gestione dei sintomi e sul rallentamento della progressione della malattia, comprende farmaci antifibrotici, corticosteroidi e immunosoppressori[11].

L'aspettativa di vita per i pazienti con IPF è migliorata grazie alle terapie disponibili, ma la prognosi varia significativamente tra gli individui. L'utilizzo dell'Intelligenza Artificiale (IA) può svolge un ruolo cruciale nell'ambito della IPF, contribuendo alla diagnosi precoce e all'analisi delle immagini polmonari. Inoltre, l'IA potrebbe supportare la previsione della progressione della malattia e valutare l'efficacia delle terapie adottate.

L'elaborato dettaglia il progetto di ricerca sviluppato presso l'azienda di sviluppo software e intelligenza artificiale synbrAIn. All'interno della quale è stato condotto un approfondito studio sull'applicazione dell'IA per predire l'evoluzione della IPF. L'obiettivo centrale è stato investigare come tool basati su IA, facenti uso di immagini TC e dati clinici tabulari, possano offrire predizioni accurate riguardo alla progressione di questa specifica malattia polmonare cronica, quantificata tramite spirometria.

Le immagini ottenute tramite TC derivano dall'impiego di una combinazione di raggi X. Questi raggi X attraversano il corpo umano e vengono successivamente rielaborati al computer per creare un'immagine tridimensionale. Questa tecnica consente di visualizzare in dettaglio gli organi interni, i vasi sanguigni e la struttura ossea [23].

La spirometria, invece, è un test che misura la quantità di aria che una persona può espirare in un respiro forzato. Durante il test, il paziente soffia in un dispositivo chiamato spirometro, che registra i volumi e i flussi respiratori. Questo permette di calcolare il dato noto come Capacità Vitale Forzata (FVC dall'inglese $ForcedVitalCapacity$), che è un indicatore importante della funzionalità polmonare[22].

Il processo metodologico ha coinvolto la progettazione e l'addestramento di modelli di deep learning che integrano informazioni provenienti da immagini mediche e dati clinici tabulari. Questa

integrazione ha mirato a migliorare la precisione delle previsioni e a fornire un quadro completo sull'evoluzione dell'IPF.

1.1 Vantaggi dell'IA

La metà dei soggetti affetti da IPF riceve una diagnosi errata, in quanto i sintomi vengono spesso confusi con altre malattie respiratorie o cardiache, come l'asma, la broncopneumopatia cronica ostruttiva (BPCO) o l'insufficienza cardiaca[10]. La diagnosi comporta l'esclusione di tutte le altre cause note di malattia polmonare fibrotica e può includere una o più delle seguenti procedure: [10]

- Anamnesi ed esame obiettivo.
- Test di funzionalità respiratoria.
- Tomografia computerizzata ad alta risoluzione (HRCT).
- Broncoscopia.
- Biopsia polmonare.

Nella metà dei soggetti affetti da IPF, occorre almeno un anno per giungere a una diagnosi corretta[10]. Tale periodo è particolarmente critico in quanto, senza trattamento, la IPF peggiora nel tempo e può progredire rapidamente [10].

L'introduzione dell'IA nella gestione della IPF apre nuove prospettive, soprattutto per la diagnosi, la previsione della progressione e l'ottimizzazione delle terapie.

Per la diagnosi precoce, l'IA può svolgere un ruolo cruciale nell'analisi di immagini radiologiche, come le TC. Grazie alle sue capacità di apprendimento automatico, può identificare segni precoci di fibrosi polmonare, facilitando una diagnosi più tempestiva.

Nella previsione della progressione della malattia, l'IA si distingue per la sua abilità nel valutare dati complessi e identificare pattern significativi. Attraverso l'analisi di dati clinici e radiologici, può contribuire a prevedere la direzione e l'entità della progressione dell'IPF, fornendo così informazioni preziose per una gestione più accurata e tempestiva.

Nel contesto terapeutico, l'IA offre un potenziale considerevole. La capacità di analizzare dettagliatamente dati clinici e profili individuali dei pazienti consente all'IA di suggerire terapie personalizzate. Questo approccio mirato potrebbe migliorare l'efficacia delle terapie adottate, adattandole alle specifiche esigenze di ciascun paziente.

L'integrazione dell'IA nella gestione dell'IPF rappresenta un avanzamento significativo, promettendo di migliorare la diagnosi, prevedere la progressione della malattia e ottimizzare le strategie terapeutiche, apportando benefici tangibili nella cura dei pazienti affetti da IPF.

1.2 Analisi della Letteratura

Obiettivo principale di questo lavoro è stato l'utilizzo dell'IA per prevedere la progressione del'IPF, la ricerca è stata condotta a partire dalla challenge Kaggle *Osic Pulmonary Fibrosis Progression*, creata dall'OSIC (Open Source Imaging Consortium) [6]. La competizione ha costituito un banco di prova cruciale per la comunità scientifica e gli esperti di IA impegnati nella previsione e comprensione di questa malattia polmonare progressiva ed è stata considerata pertanto un ottimo punto di inizio.

Questa competizione è valutata con la metrica LLLm (Laplace Log-Likelihood modificata). Nelle applicazioni mediche, è utile valutare la fiducia di un modello nelle sue decisioni, di conseguenza, la metrica è progettata per riflettere sia l'accuratezza che la certezza di ogni previsione[6]. Per ogni misurazione FVC reale, si prevede sia una FVC che una misura di confidenza (deviazione standard σ).
La metrica viene calcolata come:

$$\sigma_{\text{clipped}} = \max(\sigma, 70), \tag{1.1}$$

$$\Delta = \min(|\text{FVC}_{\text{true}} - \text{FVC}_{\text{predicted}}|, 1000), \tag{1.2}$$

$$\text{metric} = -\frac{\sqrt{2}\Delta}{\sigma_{\text{clipped}}} - ln(\sqrt{2}\sigma_{\text{clipped}}). \tag{1.3}$$

Si nota che la metrica riportata nell'espressione 1.3 assume valori negativi ed è in scala logaritmica, per cui più il valore si avvicina allo 0 migliore sarà il modello testato.

Partendo dal dataset fornito, che include immagini TC e dati tabulari dei pazienti affetti da IPF, numerose squadre e ricercatori hanno sviluppato modelli avanzati per predire l'evoluzione della malattia. L'eterogeneità dei dati e la complessità della IPF hanno presentato sfide significative, sollecitando l'implementazione di approcci innovativi e integrati.

In questa sezione viene riportata un'overview dei modelli proposti all'interno della challenge al fine di garantire un confronto adeguato delle performance. Si specifica che, un'ulteriore analisi della letteratura condotta al di fuori della competizione, non ha rilevato la presenza di studi che affrontino il problema della previsione della progressione della malattia utilizzando immagini TC e dati clinici.

Fibro-CoSANet

L'articolo[1] introduce un nuovo approccio avanzato basato su una rete neurale convoluzionale (CNN) con un modulo di auto-attenzione per predire la progressione della IPF. Il metodo proposto utilizza sia immagini TC del torace che informazioni demografiche dei pazienti, come sesso, età, stato di tabagismo e volume polmonare stimato. L'obiettivo principale è predire la pendenza della curva di declino della FVC, un indicatore cruciale della funzione polmonare.

Ciò che distingue questo approccio è la sua formulazione unica del problema, che si basa su un'ipotesi lineare e fa uso della decomposizione a valori singolari (SVD) per calcolare la pendenza iniziale della FVC, agendo come pseudo-etichetta durante l'addestramento della rete. Il modello risultante, denominato Fibro-CoSANet, supera i metodi esistenti sullo stesso dataset, ottenendo un punteggio di LLLm di -6.68, indicando un'elevata accuratezza e certezza delle predizioni.

Fibrosis-Net

Lo studio proposto da Wong et al. [26] presenta un nuovo approccio basato su una CNN chiamata Fibrosis-Net per la previsione della progressione della fibrosi polmonare da immagini TC toraciche (Figura 1.1. I risultati indicano che la Fibrosis-Net supera significativamente le soluzioni vincitrici della sfida (LLLm di -6.8188), evidenziando una maggiore efficacia nella previsione della progressione della fibrosi polmonare. La validazione dimostra che la Fibrosis-Net si basa su indicatori visivi clinicamente rilevanti (honeycombing) nelle immagini TC per le sue previsioni. Fibrosis-Net emerge come un modello promettente per la predizione della progressione della fibrosi polmonare, con potenziali implicazioni nella gestione di questa malattia considerata incurabile.

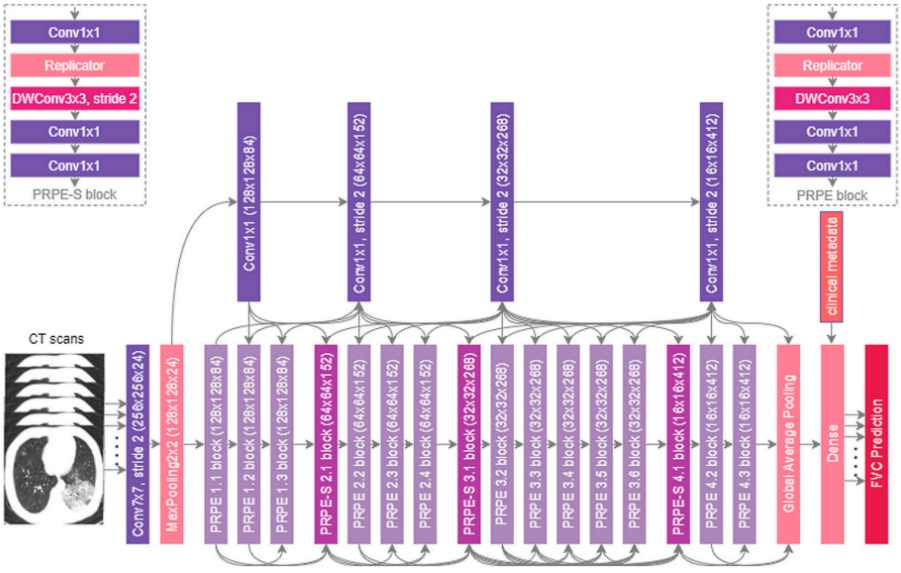

Figura 1.1. Architettura Fibrosis-Net [26]

FVC-NET

Lo studio [28] si propone di prevedere la progressione della fibrosi polmonare attraverso l'implementazione di una rete neurale profonda denominata FVC-Net. La FVC-Net integra il punteggio dell'immagine, focalizzato sul grado di "honeycombing" (un indicatore di fibrosi polmonare), con i metadati del paziente (Figura 1.2). I risultati indicano un miglioramento significativo rispetto ad altri modelli di apprendimento profondo per la previsione della progressione della fibrosi polmonare, valutato tramite la LLLm (-6.641). Lo studio suggerisce prospettive promettenti nell'utilizzo dell'apprendimento profondo per la diagnosi e la gestione della fibrosi polmonare, invitando a ulteriori approfondimenti in questo campo.

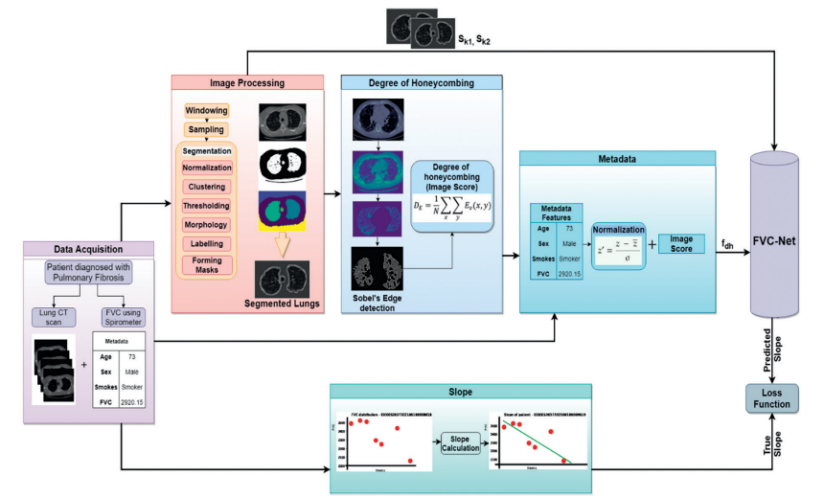

Figura 1.2. Architettura FVC-NET [28]

Vincitore Challenge

Il modello più performante emerso dalla competizione è risultato essere una combinazione di due approcci distinti: EfficientNet B5 e Quantile Regression Dense Neural Network [3]. Questi modelli sfruttano sia le informazioni contenute nelle immagini TC che nei dati tabulari, contribuendo così a migliorare l'accuratezza delle previsioni e raggiungendo un score di LLLm pari a -6.8305 .

La metrica di valutazione utilizzata per misurare le prestazioni dei modelli è stata il punteggio LLLm. Questa metrica tiene conto sia del valore predetto della FVC che della confidenza associata alla previsione.

Oltre all'approccio efficace, l'autore ha descritto altre tecniche che non hanno portato ai risultati sperati. Tra queste, il calcolo del volume polmonare, la data augmentation delle immagini e i modelli basati sulle caratteristiche dell'istogramma delle immagini; sono stati citati come approcci che non hanno ottenuto il successo atteso nel contesto della competizione.

Capitolo 2

Materiali

In questa capitolo, vengono presentati i materiali e gli strumenti utilizzati durante lo svolgimento della ricerca. L'uso appropriato di strumenti e risorse è cruciale per garantire la validità e l'affidabilità dei risultati ottenuti.

2.1 Dataset

Il dataset fornito da Kaggle per la challenge *OSIC Pulmonary Fibrosis Progression* [6] include una TC del torace e le informazioni cliniche associate per un gruppo di pazienti. Ogni paziente ha una scansione TC acquisita al tempo $Week\_0$ e numerose visite di follow-up nel corso di circa 1-2 anni, durante le quali è avvenuta la misurazione della FVC.
Nel set di training, è fornita una TC e l'intera cronologia delle misurazioni FVC. Nel set di test, viene fornita una TC e solo la misurazione iniziale della FVC. Il numero totali di casi per il set di training è di 176 scansioni TC, il complessivo numero di casi presenti nei set di test pubblici e privati ammonta approssimativamente a 200, distribuiti in proporzione di circa 15% nel set pubblico e 85% in quello privato. Poiché si tratta di dati medici autentici, l'intervallo temporale delle misurazioni varia notevolmente, costituendo una sfida significativa. Al fine di prevenire potenziali discrepanze nella tempistica delle visite di follow-up, si è reso necessario prevedere la misurazione di ciascun paziente per ogni settimana possibile. Le settimane non comprese nelle ultime tre visite sono state escluse dal calcolo del punteggio al fine di garantire una valutazione accurata.

Dati disponibili:

- Train.csv. Il set di addestramento contenente la cronologia completa delle informazioni cliniche.
- Test.csv. Il set di test contenente solo le misurazioni di base.
- cartella *train*. Contenente le TC di base dei pazienti del set di training in formato DICOM.
- cartella *test*. Contenente le TC di base dei pazienti del set di test nel formato DICOM.
- sample_submission.csv. Un esempio di formato per l'invio delle previsioni.

Colonne nei file Train.csv e Test.csv (Tabella 2.1):

- Patient. Un ID univoco per ogni paziente, corrispondente anche al nome della cartella DICOM del paziente.

- Weeks. Il numero relativo di settimane pre/post la TC di base (può essere negativo).
- FVC. La capacità polmonare registrata in ml.
- Percent. Approssima la FVC del paziente come percentuale della FVC tipica per una persona con caratteristiche simili.
- Age. Età del paziente.
- Sex. Sesso del paziente.
- SmokingStatus: Stato di tabagismo del paziente.

Patient	Weeks	FVC	Percent	Age	Sex	Smoking Status
ID00009637202177434476278	22	3578	83.37	69	Male	Ex-smoker

Tabella 2.1. Esempio di riga dei file Train.csv e Test.csv.

Si è condotta un analisi sul set di dati del train, e si è riscontrato che su 176 pazienti presenti, il 79% è di sesso maschile mentre il restante 21% è di sesso femminile. Inoltre il 67% è ex fumatore, il 28% non ha mai fumato e il restante 5% è fumatore. L'età media è di 67.3, come si evince dalla Figura 2.1, con la maggior parte dei soggetti inclusi aventi età ≥ 50 anni.

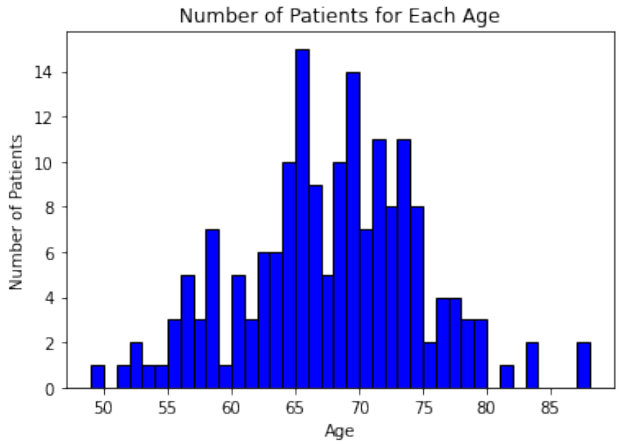

Figura 2.1. Istogramma del numero di pazienti per ogni età del taining set

2.2 Strumenti Software/Hardware Utilizzati

Questa sezione offre un approfondimento dettagliato sugli strumenti software e hardware che sono stati impiegati durante lo svolgimento del lavoro.

2.2.1 Tool per lo Sviluppo dell'Algoritmo

Kaggle

Kaggle è una piattaforma di competizione di data science ed una comunità online di data scientist e professionisti dell'apprendimento automatico appartenete a Google LLC. Kaggle consente agli utenti di trovare e pubblicare set di dati, esplorare e creare modelli in un ambiente di data science basato sul Web, collaborare con altri data scientist e ingegneri di machine learning e partecipare a concorsi per risolvere le sfide della data science[17]. Oltre a utilizzare Kaggle per il dataset, lo si è utilizzato anche in fase di inference creando un notebook per la challenge *OSIC Pulmonary Fibrosis Progression Predict lung function decline* e utilizzando le late submission per verificare la bontà dei modelli precedentemente allenati.

Weights & Biases

Weights & Biases (WB) rappresenta una piattaforma chiave nel monitoraggio e nella visualizzazione degli esperimenti di Machine Learning (ML). La sua utilità si manifesta nella registrazione di metriche, grafici, codici, modelli e altro ancora direttamente dagli script di Python [5]. Nel corso di questo lavoro, WB ha svolto un ruolo essenziale nel monitorare in tempo reale le funzioni di loss e gli score durante il training. Un ulteriore vantaggio offerto da questa piattaforma è la possibilità di monitorare le lunghe sessioni di training direttamente da dispositivi mobili.

Per usufruire di WB, è sufficiente registrarsi sul sito e successivamente creare un nuovo progetto. Una volta completati questi passaggi, è necessario accedere al codice di training, installare la libreria wandb e seguire le istruzioni fornite di seguito:

Listing 2.1. Esempio di inizializzazione di Weights and Biases (wandb)

```
import wandb # Import della libreria
wandb.login() # login su W&B

# Creazione del run con i config utilizzati
run = wandb.init(
    project=<Nome del progetto creato>,
    config={
        "learning_rate": 1e-4,
        "epochs": 40,
        "Loss": "L1",
        "batch_size": 8,
        "fold": 5
    }
)
# Esempi di creazione dei log, come chiave si ha il nome del grafico
# mentre come oggetto la variabile che si vuole monitorare
wandb.log({"loss_train": running_loss})
wandb.log({"scheduler": current_lr})
```

Successivamente andando su W&B e aprendo il progetto è possibile visualizzare i logs con i grafici creati durante l'esecuzione del codice.

Docker

Docker rappresenta una piattaforma open-source concepita per semplificare il ciclo di sviluppo, distribuzione e gestione delle applicazioni attraverso il principio della containerizzazione. La sua

funzionalità principale consiste nell'abilitare gli sviluppatori a confezionare le applicazioni insieme alle loro dipendenze in ambienti isolati dal sistema operativo (i contenitori o container) che risultano molto meno onerosi in termini di spazio rispetto alle macchine virtuali, che sono l'altra soluzione adottatta per isolare componenti software. Tale approccio garantisce coerenza nell'esecuzione delle applicazioni in diversi ambienti di sviluppo, test e produzione. L'impiego di Docker offre agli sviluppatori la capacità di semplificare in modo efficiente i processi di creazione, gestione e scalabilità dei microservizi. Nel container si è installato Jupyter Notebook che ha fornito un'interfaccia grafica del server attraverso un tunnel SSH collegato al PC locale. Poiché ogni contenitore rappresenta un ambiente isolato, non sono state necessarie preoccupazioni per dipendenze o librerie in conflitto [2].

Per la creazione di un container, è necessario prima di tutto definire un'immagine, ovvero l'ambiente di virtuale di base di cui i contenitori sono una istanza specifica. Le istruzioni per la definizione di un'immagine sono contenute all'interno di un Dockerfile, che definisce l'immagine di base, il codice sorgente dell'applicazione, le dipendenze e le configurazioni necessarie per l'esecuzione del servizio. Il Dockerfile utilizzato in questa tesi è denominato Dockerfile e ha la seguente struttura:

Listing 2.2. Dockerfile.txt

```
# Usa un'immagine base con Python 3.8.5
FROM python:3.8.5

# Imposta la directory di lavoro per il dataset e lo script
WORKDIR /script

# Copia i file nella directory del contenitore
COPY . /script

# Installa le dipendenze Python
RUN pip3 install -r requirements.txt
RUN pip install torch==1.8.0+cu111 torchvision==0.9.0+cu111 torchaudio=
    =0.8.0 -f https://download.pytorch.org/whl/torch_stable.html
RUN apt-get update && apt-get install -y libgl1-mesa-glx
RUN pip3 install jupyter

# Imposta una variabile d'ambiente per evitare interazioni interattive
# con DEBIAN_FRONTEND
ENV DEBIAN_FRONTEND=noninteractive

WORKDIR /data

# Espone la porta 8888 per Jupyter Notebook
EXPOSE 8888

# Avvia Jupyter Notebook
ENTRYPOINT ["sh", "-c", "jupyter␣notebook␣--allow-root
␣␣␣␣--no-browser␣--ip=0.0.0.0"]
```

Il Dockerfile presentato è progettato per configurare un'immagine Docker con Python, PyTorch, Jupyter Notebook e alcune librerie aggiuntive, creando un ambiente ottimizzato per l'esecuzione di applicazioni Python, con particolare attenzione all'applicazione di ML utilizzando PyTorch. La configurazione del contenitore prevede l'esposizione della porta 8888 per consentire l'accesso remoto a Jupyter Notebook.

Dopo aver caricato il Dockerfile sul server nella directory desiderata, è sufficiente accedere al server, navigare fino alla directory contenente il Dockerfile e eseguire i seguenti comandi:

Listing 2.3. Comandi esecuzione Docker

```
# Costruisce un'immagine Docker e le assegna il nome "nome_da_immagine"
# utilizzando il Dockerfile nella directory corrente.
docker build -t <nome da dare all'immagine>

# Esegue un container Docker con l'opzione "-ti" che permette
#  l'interazione in modalita pseudo-TTY.
# L'opzione "--gpus all" abilita l'accesso a tutte le GPU disponibili
# all'interno del container.
# L'opzione "-p <porta del server>:<porta dell'ambiente Docker,
# solito 8888>"
# mappa la porta del server all'interno del container alla porta
# specificata.
# L'opzione "-d" avvia il container in background (modalita detach).
# L'opzione "-v /home/sentic:/data" monta la directory /home/sentic del
# sistema  host dentro il percorso /data del container.

Docker run -ti --gpus all -p <porta del server>:<portda dell ambiente
docker di solito 8888> -d -v /home/sentic:/data <nome immagine>
```

Risulta rilevante sottolineare che, mediante l'ultima istruzione fornita, è possibile generare l'immagine del Dockerfile assegnando un token predefinito, attivando eventuali GPU presenti sul server e creando i volumi necessari per ottenere accesso ai dati del server. Una volta eseguiti questi comandi e creato il container, basta aprire un browser e accedere a http://localhost:8888/ per visualizzare Jupyter Notebook, ottenendo così un'interfaccia grafica del server.

2.2.2 Modelli Impiegati

Vision Transformers (ViT)

I modelli di Vision Trasformer (ViT) sono una categoria di reti neurali basate sull'architettura Transformer, originariamente progettata per il trattamento di dati sequenziali come testi. ViT estende questa architettura all'elaborazione di immagini, rappresentando un approccio innovativo alla visione artificiale.

A differenza delle tradizionali reti CNN, i modelli ViT non utilizzano layer convoluzionali. Invece, trattano l'immagine come una griglia di pixel, appiattendo l'input in un vettore unidimensionale e poi applicando l'architettura Transformer. Questo consente ai modelli ViT di catturare relazioni spaziali globali tra i pixel senza l'uso diretto di convoluzioni.

Il modello ViT divide l'immagine in *patch* (fette di immagine) e le tratta come sequenze di input. Ogni patch viene appiattita in un vettore e inserita all'inizio della sequenza, successivamente trattata attraverso i meccanismi di autoattenzione dei Transformer (Figura 2.2). Questo approccio consente ai modelli ViT di catturare informazioni globali e di gestire efficientemente anche immagini di grandi dimensioni.

Grazie alla loro capacità di catturare relazioni a lungo raggio nelle immagini, i modelli ViT hanno dimostrato prestazioni competitive in varie sfide e compiti di visione artificiale, superando spesso le tradizionali CNN su determinati compiti. La loro versatilità li rende adatti a diverse applicazioni, inclusa la classificazione di immagini, la segmentazione semantica e l'elaborazione di immagini mediche[9].
Il ViT utilizzato per questo lavoro è google/vit-base-patch16-224-in21k, un modello di encoder a

trasformatore, analogo a BERT [8], progettato per il riconoscimento delle immagini su larga scala e per eseguire Feature Extraction. Questo modello è stato pre-addestrato su un vasto dataset supervisionato chiamato ImageNet-21k[19], comprendente 14 milioni di immagini e 21.843 classi, a una risoluzione di 224x224 pixel. Nel processo di pre-addestramento, le immagini vengono trattate come sequenze di patch di dimensioni fisse (16x16) e vengono incorporate linearmente. Un token speciale [CLS] è aggiunto all'inizio di ogni sequenza per consentire al modello di eseguire attività di classificazione. Per catturare informazioni sulla posizione delle patch, vengono anche aggiunti incorporamenti di posizione assoluti prima di presentare la sequenza al trasformatore[27].

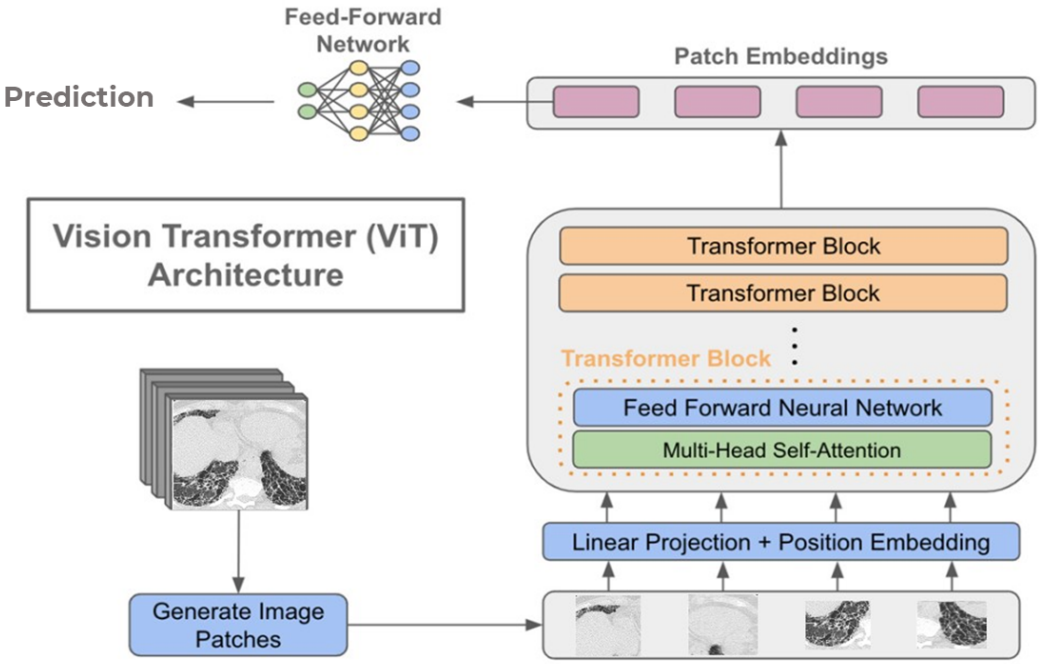

Figura 2.2. Architettura ViT modificato da [27]

Radial Basis Functions

L'articolo di Everton Gomede [13], esplora le caratteristiche, le applicazioni e i vantaggi delle reti neurali a funzioni radiali di base (RBF). Queste reti sono un tipo di rete neurale feedforward che utilizza le funzioni radiali di base come funzioni di attivazione. A differenza delle funzioni di attivazione tradizionali, come le funzioni sigmoidali o ReLU, le funzioni radiali di base introducono un nuovo livello di flessibilità abilitando la non linearità nella modellizzazione della rete. Le RBF eccellono nella modellizzazione di relazioni non lineari, rendendole preziose in vari domini, tra cui il riconoscimento di pattern, l'analisi di regressione e il clustering dei dati. L'architettura di una rete neurale RBF tipicamente consiste di tre layer: input layer, hidden layer e output layer (Figura 2.3). L'input layer riceve i dati di input, che vengono poi passati attraverso il layer nascosto, dove le funzioni radiali di base vengono applicate. Ogni neurone nel layer nascosto rappresenta una funzione radiale di base, con il suo centro corrispondente a un punto di dati specifico e la sua larghezza che determina l'estensione dell'influenza sull'output. L'output layer combina le risposte dei neuroni dello hidden layer per produrre l'output finale. L'allenamento di una rete neurale RBF comporta due passaggi principali: la selezione del centro e la determinazione del peso. La selezione del centro mira a identificare i punti di dati rappresentativi che fungono da centri per le funzioni radiali di base. Diverse tecniche, tra cui algoritmi di clustering come k-means, possono essere impiegate a questo scopo. La determinazione del peso comporta l'ottimizzazione dei pesi associati ad ogni funzione radiale di base per minimizzare la differenza tra l'output previsto della rete e l'output desiderato. Questo processo di ottimizzazione può essere realizzato attraverso tecniche come la regressione dei minimi quadrati o la discesa del gradiente. Le RBF trovano ampio uso nell'analisi di regressione, consentendo la modellizzazione e la previsione accurate di variabili continue basate sui dati di input.

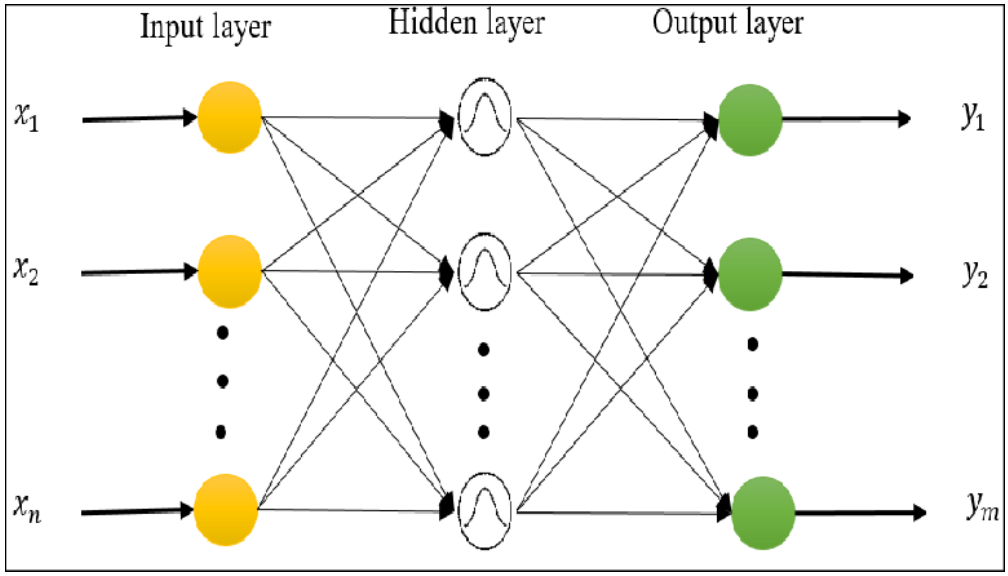

Figura 2.3. Struttura RBF [24]

2.2.3 Stumenti Hardware

Nelle diverse sessioni di addestramento condotte, sono state utilizzate tre diverse unità di elaborazione grafica (GPU) ospitate su un server fornito da SynbrAIn. Le GPU, appartenenti alla serie

NVIDIA GeForce RTX 3090, presentano notevoli differenze nella capacità di allocazione di memoria. In particolare, le GPU denominate GPU0 e GPU1 manifestano una capacità di allocazione di 11 GB ciascuna, mentre la GPU2 dispone di una capacità di 24 GB.

Questa diversità nelle capacità di allocazione di memoria delle GPU può influenzare significativamente la gestione delle risorse durante le fasi di addestramento dei modelli. È consuetudine assegnare compiti più impegnativi in termini di memoria alle GPU con capacità superiori, ad esempio durante l'addestramento di modelli di dimensioni maggiori o l'elaborazione di dataset più estesi. Tuttavia, è essenziale considerare attentamente tali differenze nella capacità di memoria durante la pianificazione e l'esecuzione delle sessioni di addestramento per massimizzare l'utilizzo efficiente delle risorse disponibili.

Per verificare le risorse disponibili è sufficiente eseguire il comando da linea di comando del server:

Listing 2.4. Comando per verificare la disponibilità delle GPU sul server

```
# Comando per verificare la disponibilita delle GPU sul server
nvidia-smi
```

Capitolo 3

Metodi e Risultati: Challenge Kaggle

In questo capitolo, saranno dettagliati tutti gli approcci impiegati al fine di conseguire il modello ottimale mediante l'utilizzo del framework ViT descritto in 2.2.2, con l'obiettivo di migliorare in maniera efficace i risultati della competizione di Kaggle descritto in 2.2.1.

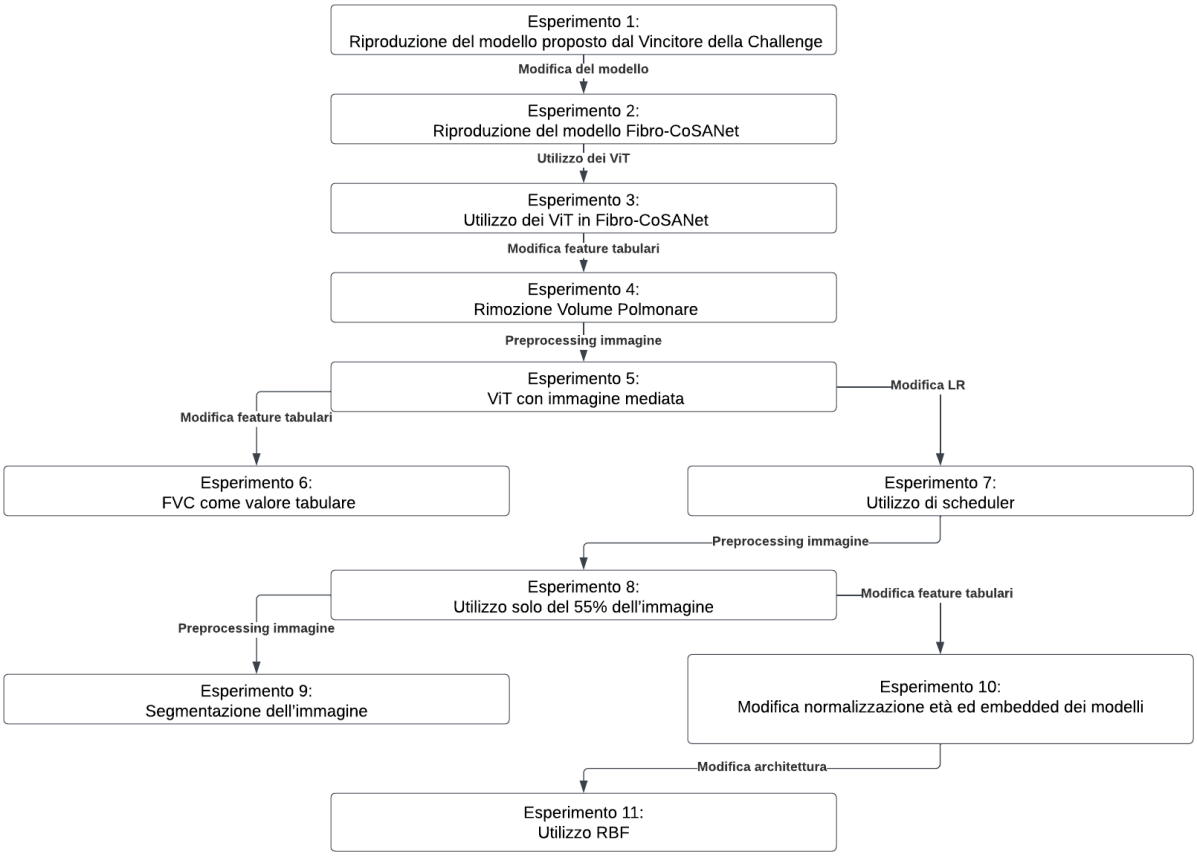

Figura 3.1. Diagramma a blocchi esperimenti Challenge Kaggle

Il diagramma a blocchi, rappresentato nella Figura 3.1, fornisce una panoramica degli esperimenti condotti in questo capitolo. Esso illustra chiaramente le varie modifiche apportate tra

un esperimento e l'altro. Inoltre, consente di visualizzare immediatamente se un esperimento ha ottenuto risultati insoddisfacenti, poiché il flusso si interrompe in quel punto.

Questo approccio visuale è utile per comprendere rapidamente la progressione degli esperimenti e per identificare eventuali trend o correlazioni tra le modifiche apportate e le prestazioni complessive del modello. La chiara rappresentazione grafica facilita l'analisi del processo sperimentale e contribuisce a una migliore comprensione delle dinamiche che hanno portato ai risultati ottenuti.

3.1 Esperimento 1: Riproduzione del Modello Proposto dal Vincitore della Challenge

Inizialmente, sono stati replicati i risultati ottenuti dal vincitore della challenge menzionata[3]. L'autore ha adottato una soluzione basata su due modelli principali: il modello di regressione quantilica e l'EfficientNet B5. L'addestramento di entrambi i modelli è stato inizialmente eseguito partendo da zero. Nel dettaglio, l'EfficientNet è stato addestrato per 30 epoche, mentre la regressione quantilica per 600 epoche. Successivamente, l'autore ha apportato sostanziali modifiche all'architettura della regressione quantilica, basandosi sui risultati ottenuti durante la fase di validazione, dove tale architettura ha dimostrato prestazioni superiori. Dopo l'eliminazione di tutte le caratteristiche relative ai "Percent" per entrambi i modelli, l'autore ha ottenuto un notevole miglioramento nei risultati della private leaderboard. Seguendo fedelmente questi passaggi e utilizzando i pesi per l'ensemble dei modelli definiti dall'autore (55% per la regressione quantilica e 45% per EfficientNet), la replicazione dei risultati ottenuti dal vincitore è risultata relativamente semplice, ottenendo uno score dalla submission di -6.8305.

3.2 Esperimento 2: Riproduzione del Modello Fibro-CoSANet

Dopo aver replicato i risultati ottenuti dal vincitore della challenge, sono state condotte ricerche sullo stato dell'arte di questa competizione, rivelando diversi approcci che hanno ottenuto risultati superiori al vincitore stesso. Tra le varie proposte trovate, si è scelto di prendere ispirazione dal lavoro di Al Nazi et al. [1], il quale si basa sull'utilizzo di Fibro-CoSANet. Questo modello sfrutta immagini TC e informazioni demografiche all'interno di un framework di reti neurali convoluzionali, arricchito da uno strato di attenzione impilato. Rilevanti esperimenti condotti sul dataset di OSIC Pulmonary Fibrosis Progression hanno chiaramente evidenziato la superiorità di Fibro-CoSANet rispetto ad altri approcci.

Durante il processo di preparazione del dataset (Figura 3.2), viene selezionata casualmente una slice TC per l'analisi. Successivamente, questa slice viene elaborata attraverso Fibro-CoSANet. Le caratteristiche demografiche vengono normalizzate, e il volume polmonare viene stimato, combinandolo con le caratteristiche convoluzionali derivate dalla TC. La pendenza della FVC viene calcolata mediante il metodo dei minimi quadrati della SVD. Questa pendenza di FVC calcolata funge da misurazione Ground-Truth (GT) per valutare l'accuratezza della pendenza di FVC prevista.

Fibro-CoSANet invia l'immagine TC casualmente selezionata a reti neurali convoluzionali preaddestrate (Fig. 3.3), f_{enc}, come ad esempio ResNet18, generando una mappa di caratteristiche TC, f_c. Successivamente, è presente uno strato di autoattenzione impilato sopra l'ultimo strato convoluzionale della CNN, consentendo alla rete di concentrarsi su regioni specifiche. La mappa di caratteristiche risultante, f_a, dal modulo di autoattenzione impilato, viene quindi elaborata attraverso il global average pooling per produrre la rappresentazione finale, f_d, dell'immagine TC. In parallelo, la rete fonde le caratteristiche demografiche e il volume polmonare stimato, f_s, con f_d, i quali passano attraverso uno strato lineare per prevedere la pendenza della FVC. Successivamente,

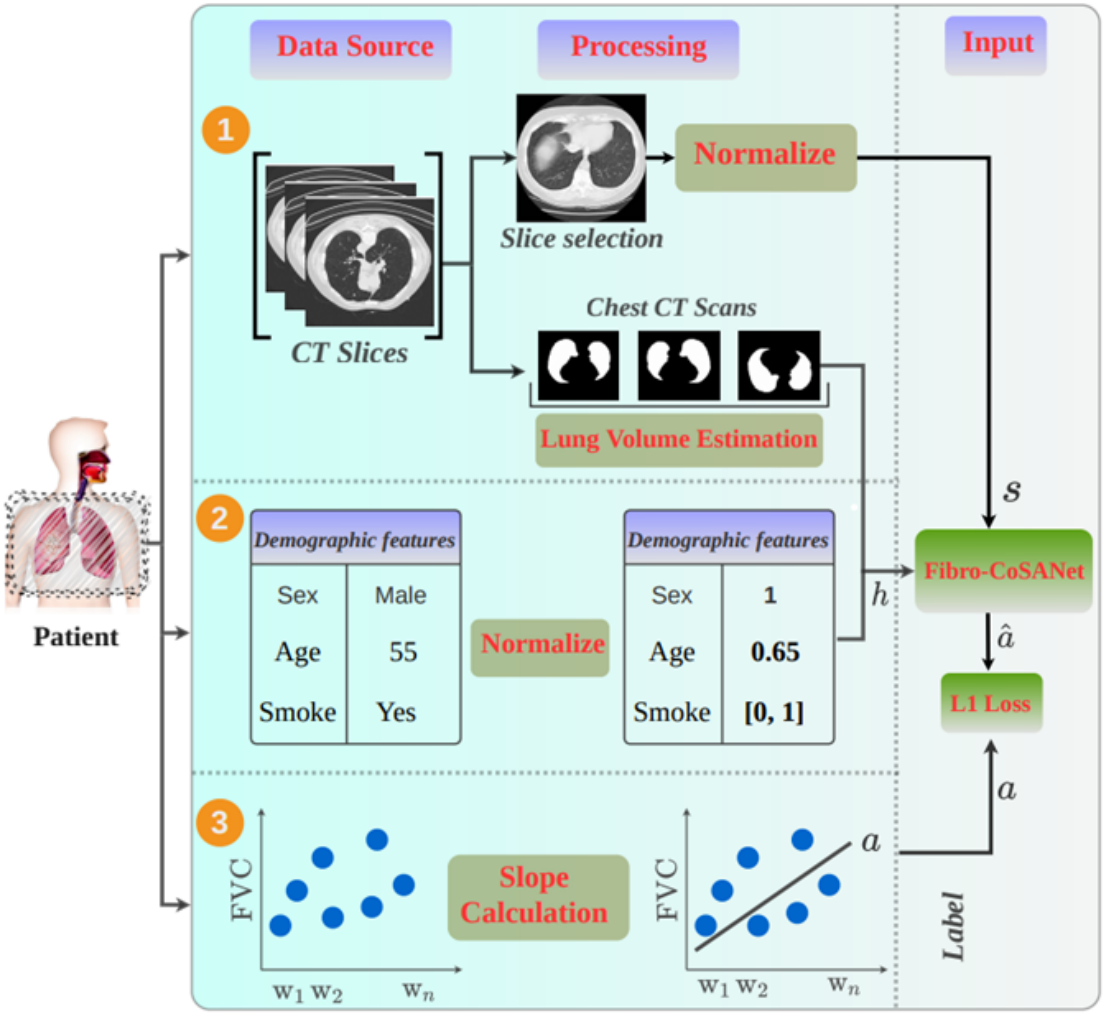

Figura 3.2. Illustrazione del processo di preparazione del dataset (1, 2) e generazione del ground–truth (3) per l'addestramento di Fibro-CoSANet [1].

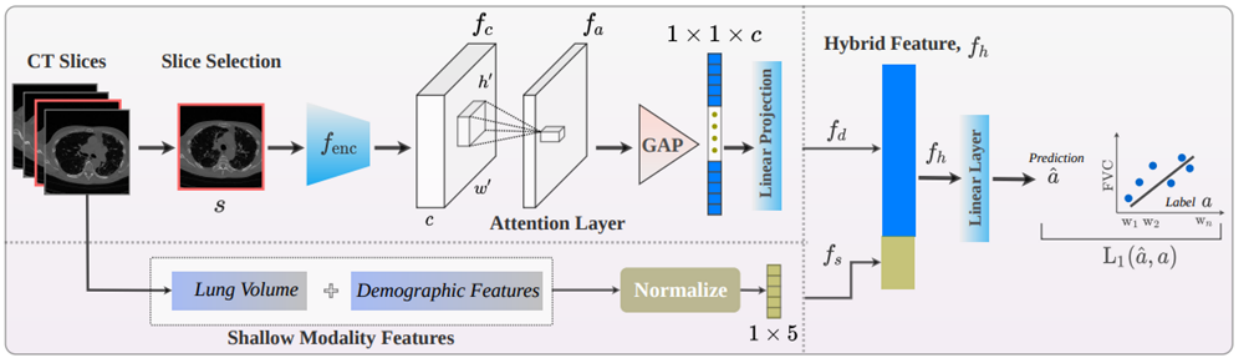

Figura 3.3. Illustrazione del doppio flusso (dual-stream) della pipeline di Fibro-CoSANet [1].

si calcola la L1 loss tra la pendenza prevista, â, e la pseudo pendenza GT, a. Ripercorrendo gli step di Fibro-CoSANet e si è ottenuto uno score di -6.687.

3.3 Esperimento 3: Utilizzo dei ViT in Fibro-CoSANet

Prendendo ispirazione dall'architettura di Fibro-CoSANet, i primi modelli sono stati sviluppati sostituendo l'utilizzo delle CNN con il ViT. In particolare, è stato adottato il modello ViT di Google denominato $vit - base - patch16 - 224 - in21k$, reperibile su Hugging Face [27]. La scelta di questo specifico modello su Hugging Face, $vit - base - patch16 - 224 - in21k$, indica l'utilizzo di un modello basato su patch di dimensioni 16x16 pixel, con un'input size di 224x224 e preaddestrato su un dataset contenente 21.000 classi. Questo modello è stato impiegato come estrattore di caratteristiche per ottenere rappresentazioni dell'immagine TC, rimpiazzando il ruolo precedentemente svolto dalle CNN.

La transizione da CNN a ViT rappresenta un cambiamento significativo nell'architettura, capitalizzando sulle potenzialità dei transformer nella rappresentazione delle caratteristiche. Ciò offre contemporaneamente maggiore flessibilità e interpretabilità delle informazioni estratte. Questo nuovo approccio è stato introdotto con l'obiettivo di influenzare positivamente le prestazioni del modello, soprattutto in termini di adattabilità a differenti contesti e tipologie di dati.

3.3.1 Workflow

Dopo aver introdotto questa sostanziale modifica al modello, sono stati condotti diversi esperimenti mantenendo, in generale, un workflow coerente per ogni iterazione di test (Figura 3.4).

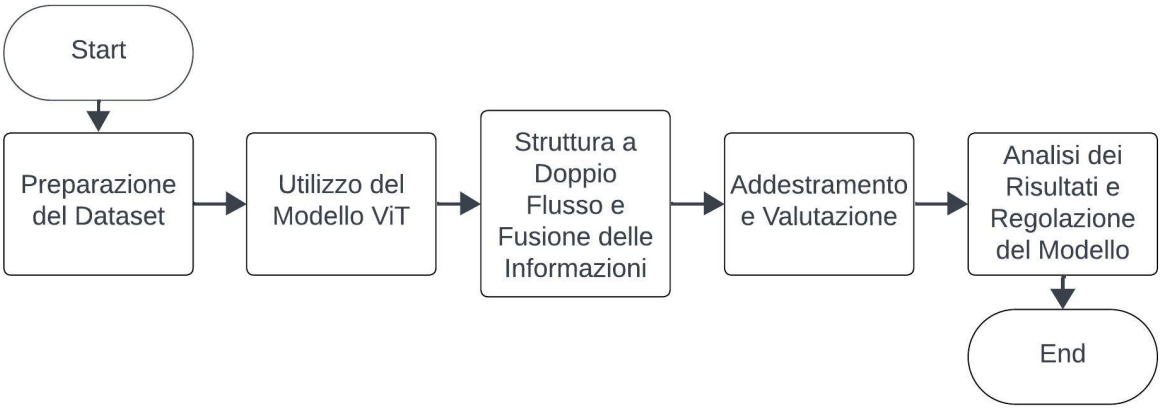

Figura 3.4. Workflow

Il workflow utilizzato in ogni esperimento ha mantenuto le seguenti fasi principali:

1. Preparazione del Dataset. La preparazione del dataset ha previsto la selezione di un'immagine della TC, eventualmente sottoposta a preprocessing. Inoltre, sono state selezionate e normalizzate caratteristiche demografiche, alle quali sono state aggiunte ulteriori feature tabulari.

2. Utilizzo del Modello ViT. Il modello ViT, in particolare "vit-base-patch16-224-in21k" di Hugging Face, è stato impiegato per l'estrazione delle caratteristiche dall'immagine TC. Questo modello, basato su transformer, ha sostituito il ruolo precedentemente svolto dalle CNN.

3. Struttura a Doppio Flusso e Fusione delle Informazioni. Il flusso di elaborazione è stato mantenuto a doppio canale, con uno specifico per le informazioni estratte dal ViT e l'altro per le caratteristiche demografiche. La fusione di queste informazioni è stata eseguita in modo da permettere al modello di integrare efficacemente le rappresentazioni ottenute dal ViT con le caratteristiche demografiche.

4. Addestramento e Valutazione. Il modello è stato addestrato utilizzando un approccio supervisionato, con l'obiettivo di prevedere la pendenza della FVC. L'accuratezza del modello è stata valutata mediante confronto della predizione con la pendenza FVC calcolata durante il processo di preparazione del dataset, che è stata utilizzata come GT.

5. Analisi dei Risultati e Regolazione del Modello. Dopo ciascun esperimento, sono stati analizzati i risultati ottenuti, valutando le performance del modello nella previsione della variazione della capacità polmonare. Eventuali modifiche o regolazioni alla struttura del modello sono state apportate sulla base delle osservazioni ricavate dall'analisi dei risultati.

Mantenendo una coerenza nel workflow durante gli esperimenti, è stato possibile valutare gli effetti specifici dell'integrazione del modello ViT rispetto alla versione originale di Fibro-CoSANet.

3.3.2 Data Preparation

Dopo aver modificato l'architettura del modello per adattarla all'utilizzo di ViT, è stato avviato il processo di addestramento. Nella fase di input, il modello riceve una slice casuale dal centro del volume, da cui ViT estrae 32 feature caratteristiche dell'immagine. Successivamente, queste feature estratte vengono linearizzate e concatenate alle feature tabulari.

Per quanto riguarda le feature tabulari, sono state apportate alcune modifiche significative:

- Feature di Sesso: Le feature relative al sesso sono state trasformate utilizzando il metodo di one-hot encoding[20]. In questo contesto, l'encoding rappresenta le categorie in cui un individuo può appartenere. Ad esempio, la presenza del valore "1" nella feature corrisponde al genere maschile, mentre la presenza di "0" indica il genere femminile.

- Feature dello Stato Tabagico: La feature relativa allo stato tabagico è stata gestita attraverso un'encoding più complesso. Adottando il metodo di one-hot encoding, sono state create quattro categorie distintive:

  – "11" rappresenta un individuo fumatore.

  – "01" indica un individuo ex-fumatore.

  – "00" corrisponde a un individuo che non ha mai fumato.

  – "10" denota uno stato di fumo non definito.

Questo approccio all'encoding delle feature tabulari consente al modello di comprendere e utilizzare in modo efficace informazioni riguardanti il soggetto durante il processo di addestramento. Le nuove rappresentazioni delle feature tabulari sono state quindi integrate con le feature estratte dal ViT, creando un input completo e informativo per il modello.

3.3.3 Model Training

Dopo che le feature estratte dal ViT e le feature tabulari sono state concatenate, il flusso di dati prosegue attraverso un layer fully-connected. Questo layer presenta un hidden layer di dimensione 32 e un output layer di dimensione 1. L'obiettivo di questa configurazione è prevedere il valore della pendenza (slope) della FVC.

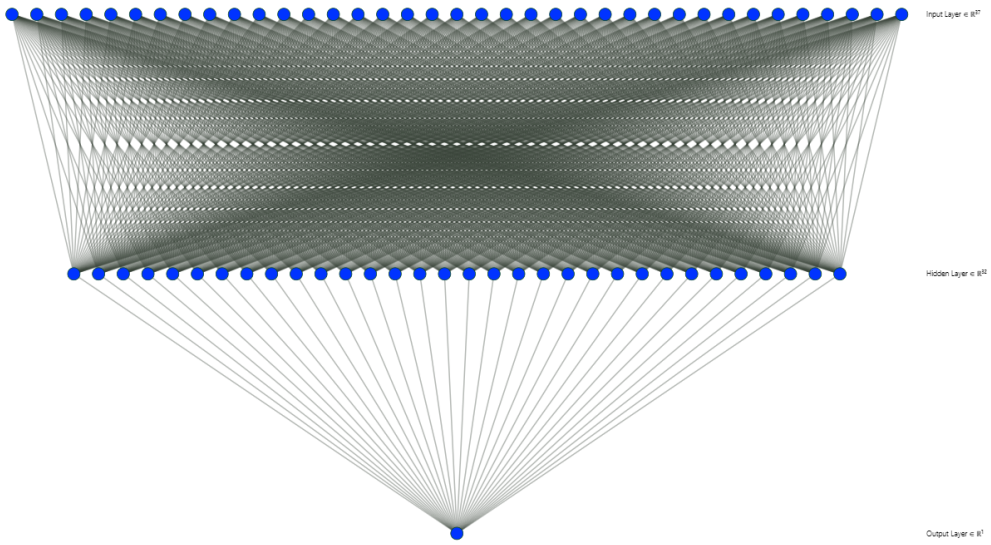

Figura 3.5. Architettura Fully Connected

Questo valore predetto sarà confrontato con la pendenza reale calcolata durante la preparazione del dataset, servendo come metrica di valutazione per l'efficacia del modello. In sintesi, il modello apprende relazioni complesse tra le feature estratte dalle immagini TC, le informazioni demografiche e il volume polmonare stimato, per poi prevedere la pendenza della Capacità Vitale Forzata. In particolare, tale capacità è fornita dall'architettura fully-connected (Figura 3.5), che con un hidden layer di 32 unità offre al modello la possibilità di catturare pattern complessi e svolgere una previsione precisa.

È positivo notare come siano stati eseguiti diversi training con l'architettura proposta, testando l'efficacia di vari valori di learning rate (LR) e batch size per facilitare il fine-tuning del modello. Inoltre, l'applicazione della k-fold cross-validation [15] (con k=5) , ha consentito di ottenere modelli con maggiore capacità di generalizzazione, addestrati su diverse porzioni del dataset.

Di seguito viene riportata una panoramica degli approcci adottati durante il training al fine di massimizzare le performance ottenibili:

- Hyperparameters Tuning:
  - Variabilità del LR: Modificando il valore del LR durante il fine-tuning, è possibile influenzare la convergenza del modello. È comune eseguire diverse iterazioni di training con LR diversi per trovare il valore ottimale che massimizza la performance del modello.
  - Adattamento del Batch Size: La dimensione del batch è un altro iperparametro critico durante il fine-tuning. Esperimenti con diverse dimensioni di batch possono influenzare la stabilità del modello, la velocità di convergenza e la sua capacità di generalizzazione.
- Tecniche per garantire la robustezza del modello:

– K-Fold Cross-Validation: La suddivisione del dataset in 5 fold è una pratica solida per valutare la capacità del modello di generalizzare su diverse porzioni dei dati. Addestrare il modello su differenti combinazioni di training e validation sets aiuta a mitigare il rischio di overfitting e ad ottenere una stima più affidabile delle performance del modello.
– Ottenere 5 Modelli Diversi: L'utilizzo di k-fold cross-validation consente di ottenere 5 modelli diversi, ognuno addestrato su un'80% diverso del dataset. Questi modelli possono essere successivamente valutati in modo da ottenere una comprensione più completa delle performance del modello su dati non visti.

In generale, l'impiego delle tecniche descritte in associazione all'utilizzo di diverse configurazioni di hyperparameter contribuisce a sviluppare modelli più robusti e generalizzabili per la previsione della pendenza della FVC nella IPF.

3.3.4 Risultati

La prima sottomissione effettuata su Kaggle è stata eseguita con un modello addestrato mediante una configurazione specifica. Nel dettaglio, è stato adottato un batch size di 32 unità e un LR di 1×10^{-4}. Tale scelta è stata motivata da prestazioni incoraggianti riscontrate nel quarto fold della cross-validation, in cui il modello ha conseguito il miglior punteggio con una valutazione di -6.654 (Tabella 3.1). L'adozione di questa configurazione è stata frutto di una ponderata selezione degli

Batch Size	Learning Rate	Fold	Score
32	1×10^{-3}	4	-6.7449
32	1×10^{-4}	4	-6.654
32	1×10^{-5}	4	-6.764
16	1×10^{-3}	4	-6.867
16	1×10^{-4}	4	-6.88
16	1×10^{-5}	4	-6.893
8	1×10^{-3}	4	-6.982
8	1×10^{-4}	4	-6.864
8	1×10^{-5}	4	-7.231

Tabella 3.1. Differenti Batch Size e Leraning Rate con il miglior Score ottenuto dal Validation Set

iperparametri al fine di massimizzare le prospettive di successo nella competizione. La decisione di basarsi sui risultati del quarto fold per la sottomissione su Kaggle riflette una strategia empirica, avente l'obiettivo di trasferire il successo osservato durante la fase di validazione al set di test effettivo.

Kaggle Submission	Private Score	Public Score
Succeeded	-6.9151	-6.9131

Tabella 3.2. Submission esperimento 3

Dai risultati della sottomissione, emerge che il valore ottenuto durante la fase di validazione è in linea con il valore ottenuto nel Private Score (Eq.1.3), e si avvicina considerevolmente al risultato del vincitore della challenge, che è di -6.8305 (Tabella 3.2). Ciò suggerisce una buona capacità del modello di generalizzare su dati non visti e di performare in modo competitivo rispetto agli altri

partecipanti alla challenge. La coerenza tra i risultati ottenuti nella fase di validazione e quelli nel Private Score indica una robusta capacità predittiva del modello, consentendo di ottenere risultati paragonabili ai migliori nella competizione. Questo allinea i risultati ottenuti con gli obiettivi della sfida e rafforza la validità delle scelte fatte nel processo di sviluppo e addestramento del modello.

3.4 Esperimento 4: Rimozione Volume Polmonare

Per valutare l'efficacia del parametro tabulare relativo al volume polmonare, è stato eseguito un nuovo training utilizzando lo stesso workflow dell'esperimento 1 (sezione 3.3.1).

3.4.1 Data Preparation

Si è proceduto alla rimozione del parametro del volume polmonare dalle feature tabulari, comportando una modifica specifica nella composizione delle feature utilizzate durante il training del modello. Tale decisione è stata guidata da un'analisi dei risultati ottenuti e dall'obiettivo di valutare l'impatto di questa modifica sulle prestazioni complessive del modello. La scelta di eliminare il parametro del volume polmonare può rappresentare una strategia per semplificare o migliorare l'efficacia della rappresentazione delle feature tabulari, contribuendo così a una migliore generalizzazione del modello. Le feature relative al sesso e allo stato tabagico, precedentemente elaborate mediante il metodo one-hot encoding, sono state mantenute, mentre l'età è stata normalizzata attraverso un processo dedicato.

3.4.2 Model Training

Sono state estratte le feature da una slice casuale utilizzando il ViT. Queste feature sono state poi accorpate alle feature tabulari e, successivamente, sono state sottoposte a uno strato lineare con un solo output, al fine di ottenere la previsione della pendenza dell'FVC. Gli iperparametri utilizzati sono gli stessi che hanno prodotto i migliori risultati nella sezione precedente, ossia una dimensione di batch pari a 32 e LR di 1×10^{-4}, come confermato dalla Tabella 3.1. Successivamente, è stata effettuata una nuova sottomissione su Kaggle per valutare l'impatto di questa modifica sulle prestazioni globali del modello.

3.4.3 Risultati

Kaggle Submission	Private Score	Public Score
Succeeded	-6.905	-6.955

Tabella 3.3. Submission esperimento 4

Dopo aver osservato limitate differenze nelle prestazioni del modello con (-6.9151 come riportato nella Tabella 3.2) o senza (-6.905 come riportato nella Tabella 3.3) la feature tabulare legata al volume polmonare, si è deciso di transire verso una rappresentazione grafica di questa caratteristica. Per ottenere tale parametro, è stato inizialmente impiegato il metodo di watershed per generare le maschere del polmone, e successivamente è stata estratta la feature sommando i voxel della maschera binaria ottenuta [12].

La scelta di adottare un approccio grafico è stata guidata dalla volontà di ottenere una comprensione più chiara e visuale del ruolo di questa feature nei processi decisionali del modello. L'utilizzo

di maschere binarie per definire il volume polmonare offre una rappresentazione visiva immediata delle regioni coinvolte nell'analisi, agevolando una valutazione qualitativa più approfondita.

Tuttavia, l'analisi della Figura 3.6 evidenzia che il metodo di watershed utilizzato non segmenta correttamente il polmone, generando invece una segmentazione casuale e, in alcuni casi, producendo segmentazioni completamente nere. Questa inefficacia potrebbe derivare da varie ragioni, tra cui la complessità delle immagini mediche, la presenza di artefatti o la necessità di ulteriori ottimizzazioni dei parametri del metodo stesso.

Dato che sia il modello con il volume polmonare come feature che quello senza hanno prodotto risultati simili, e considerando l'inefficienza del metodo di watershed, si è presa la decisione di rimuovere definitivamente il volume polmonare dalle feature tabulari. Tale scelta è stata motivata dalla constatazione che il contributo informativo del volume polmonare sembra non incidere significativamente sulle prestazioni del modello, rendendo questa feature meno rilevante ai fini della previsione della pendenza della FVC.

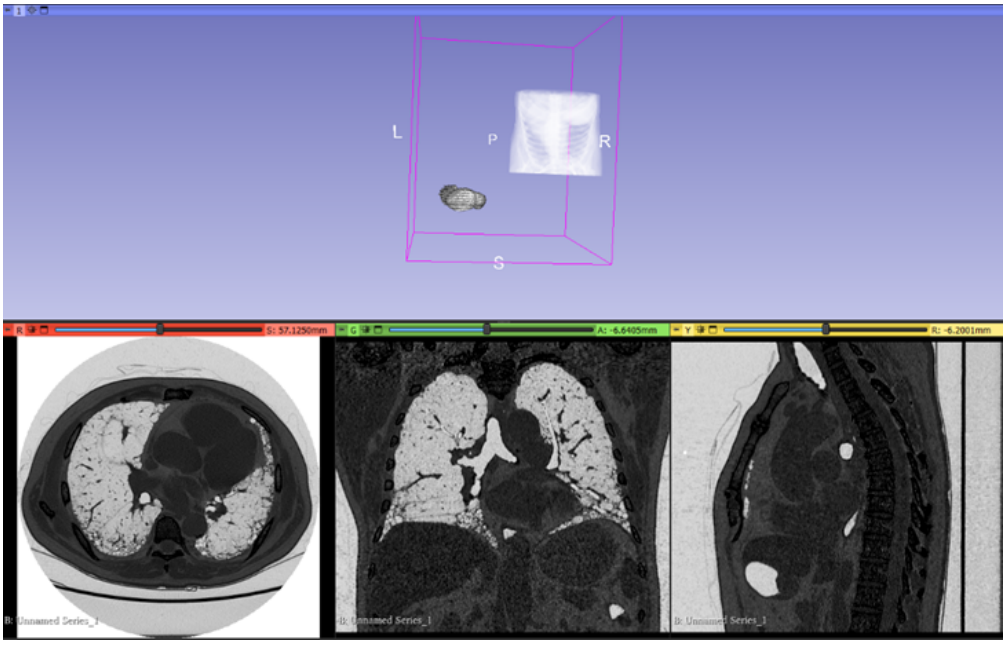

Figura 3.6. Sovrapposizione maschera con immagine reale

3.5 Esperimento 5: ViT con Immagine Mediata

Gli esperti aziendali hanno consigliato di evitare l'estrazione di una slice casuale dal centro del volume, ritenendo che questo approccio potesse risultare poco significativo per il compito in questione, specialmente considerando la variabilità tra i diversi pazienti. In alternativa, è stato suggerito di utilizzare un'immagine ottenuta dalla media lungo l'asse z del volume (Figura 3.7). Questa strategia è stata ritenuta più efficace, poiché ha dimostrato di offrire informazioni più rappresentative sull'intero volume polmonare. Tale approccio si è rivelato particolarmente utile in contesti simili, ad esempio, affrontando la sfida di prevedere la presenza del COVID-19 da immagini TC toraciche. Utilizzando un'immagine ottenuta dalla media lungo l'asse z, si è garantita l'incorporazione di informazioni relative a tutto il volume polmonare anziché focalizzarsi su una singola slice, migliorando così la capacità predittiva del modello.

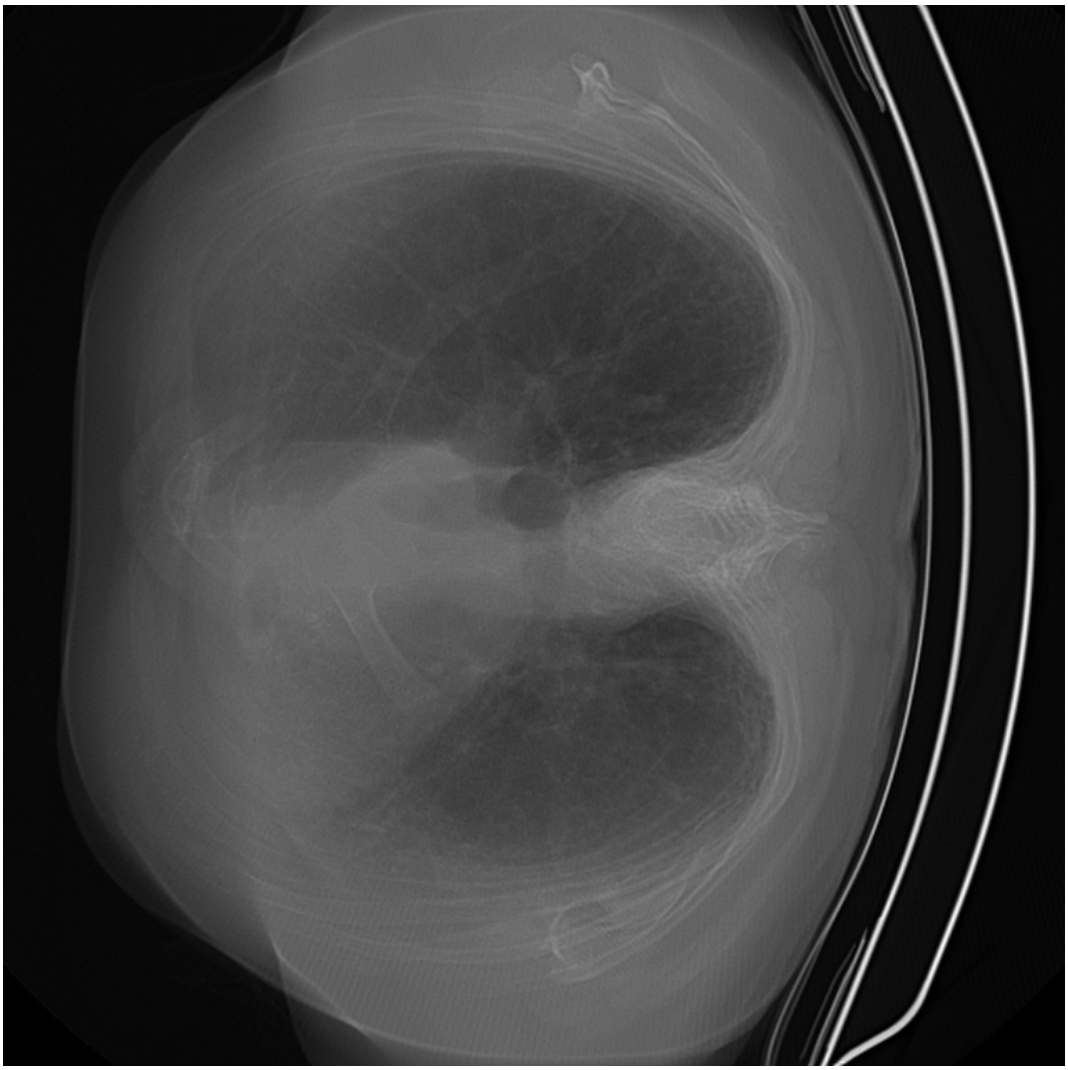

Figura 3.7. Slice mediata paziente ID00009637202177434476278

3.5.1 Data Preparation

Sfruttando queste nuove slice come input, è stato eseguito il processo di estrazione delle feature da parte dei ViT. Le caratteristiche così ottenute sono state successivamente integrate con i dati tabulari normalizzati, e in alcuni casi, è stato applicato il metodo di one-hot encoding per gestire specifiche variabili categoriche.

3.5.2 Model Training

Sono stati eseguiti numerosi allenamenti, regolando gli iperparametri per trovare la configurazione ottimale che massimizzasse lo score in ciascun fold. Questa pratica è comune e mira a individuare la combinazione migliore per ottenere le prestazioni più elevate su diverse porzioni del dataset durante la cross-validation. L'obiettivo è trovare parametri che massimizzino la capacità del modello di generalizzare su dati non visti, garantendo risultati competitivi in ogni configurazione specifica. La loss utilizzata durante questi allenamenti è stata la L1 loss, che è stata ottimizzata per ridurre la

differenza assoluta tra le predizioni del modello e i valori effettivi della pendenza della FVC.

Batch Size	Learning Rate	Fold	Score
32	1×10^{-3}	2	-6.449
32	1×10^{-4}	2	-6.479
32	1×10^{-5}	2	-6.494
16	1×10^{-3}	4	-6.567
16	1×10^{-4}	4	-6.663
16	1×10^{-5}	4	-6.532
8	1×10^{-3}	4	-6.652
8	1×10^{-4}	4	-6.454
8	1×10^{-5}	4	-6.541

Tabella 3.4. Differenti Batch Size e Leraning Rate con il miglior Score ottenuto dal Validation Set usando immagine media

Come evidenziato nella Tabella 3.4, il risultato più elevato in termini di score è stato raggiunto nel fold 2, adoperando un Batch Size di 32 e un LR di 1×10^{-3}. Questa configurazione specifica ha dimostrato di essere particolarmente efficace in quel particolare sottoinsieme del dataset durante la fase di validazione, evidenziando l'importanza della scelta degli iperparametri nel determinare le performance del modello (Figura 3.8 Figura 3.9).

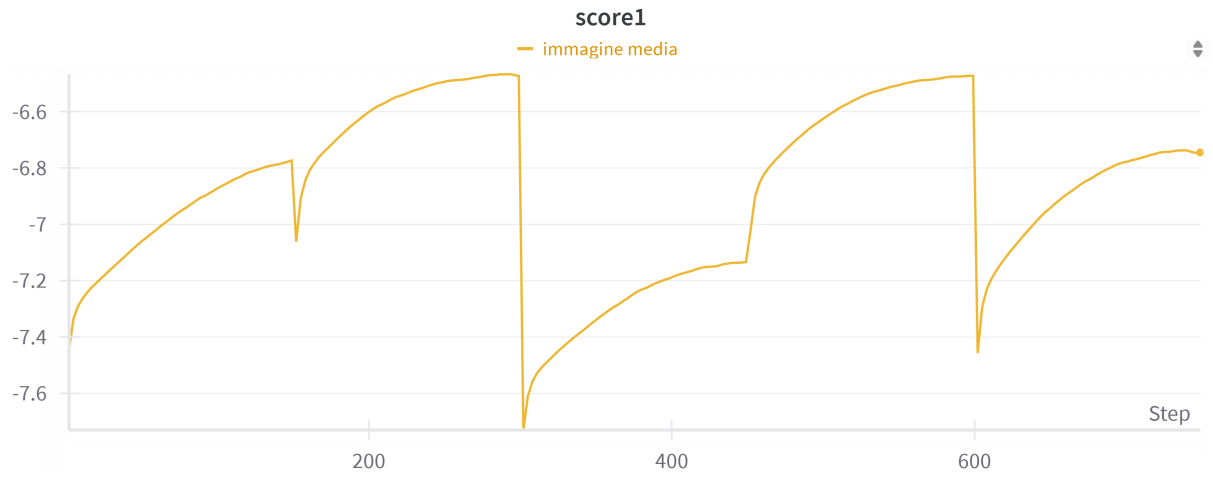

Figura 3.8. Score nei 5 Fold

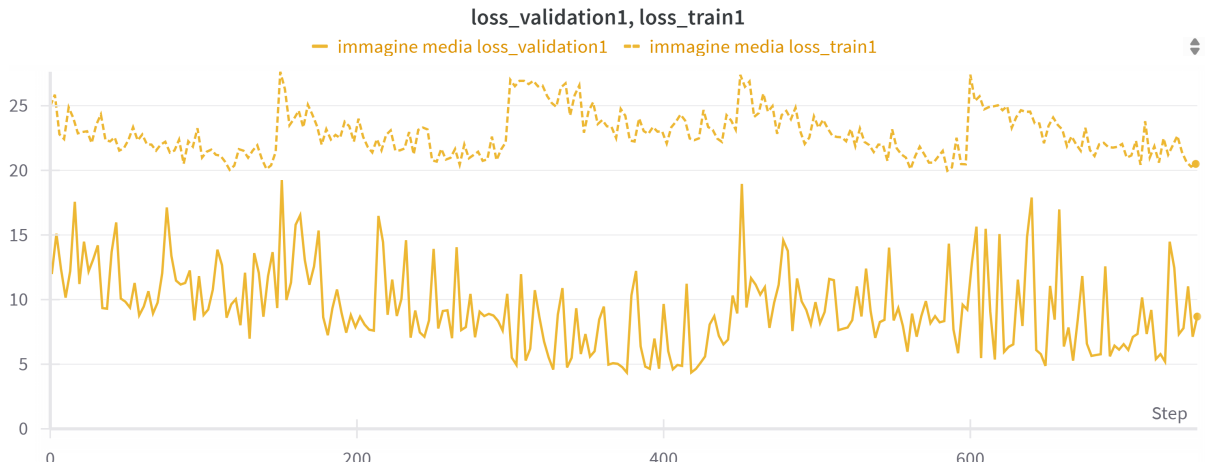

Figura 3.9. Loss di training e validation (L1 loss)

3.5.3 Risultati

Sfruttando il modello del secondo fold, addestrato con un Batch Size di 32 e un LR di 1×10^{-3}, è stata eseguita un'operazione di inferenza su Kaggle per valutare la coerenza tra i risultati ottenuti durante la fase di validazione e quelli ottenuti nella competizione. Questa fase di verifica è fondamentale per assicurare che le buone performance osservate durante l'allenamento e la validazione siano traslate in risultati altrettanto positivi nell'ambito della sfida Kaggle, garantendo così la capacità del modello di generalizzare efficacemente su nuovi dati.

Kaggle Submission	Private Score	Public Score
Succeeded	-6.8878	-6.9177

Tabella 3.5. Submission esperimento 5

Nonostante l'esistenza di un notevole divario tra lo score ottenuto nella fase di validazione (-6.449) e quello ottenuto nella fase di test (-6.8878 Tabella 3.5), si è riscontrato un'ulteriore miglioramento rispetto al modello precedente descritto nella Sezione 3.4.

Questo risultato sottolinea che l'approccio di utilizzare immagini mediate lungo l'asse z, che forniscono informazioni globali sull'intero volume, utilizzando una sola slice, è efficace. Inoltre, si evidenzia che questa metodologia non sovraccarica il costo computazionale del training, dimostrando la praticità di questa strategia nel contesto della sfida Kaggle. La capacità del modello di migliorare nonostante il cambiamento nella rappresentazione delle immagini conferma l'efficacia dell'approccio adottato.

3.6 Esperimento 6: FVC come Valore Tabulare

L'approccio di aggiungere il primo valore di FVC disponibile dopo la settimana iniziale ($week\_0$, settimana in cui viene effettuata la TC) è stato esplorato per valutare se potesse apportare contributi positivi al modello. Questa strategia potrebbe essere rilevante poiché il primo valore di FVC può fornire informazioni preziose sulla capacità polmonare iniziale del paziente, contribuendo così a migliorare la previsione della sua futura progressione.

L'inclusione di questa variabile aggiuntiva è stata valutata durante il training del modello per determinare se avesse un impatto significativo sulle performance complessive. Tale analisi consente di comprendere se l'aggiunta del primo valore di FVC come feature influenzi positivamente la capacità del modello di effettuare previsioni accurate sulla progressione della FVC.

3.6.1 Data Preparation

Le immagini utilizzate per l'estrazione delle feature sono state elaborate mediante il calcolo della media lungo l'asse z, come illustrato nella sezione precedente. Per quanto riguarda le feature tabulari, il genere e lo stato tabagico sono stati trasformati utilizzando il metodo di "one-hot encoding". Inoltre, l'età è stata normalizzata insieme a una nuova feature che incorpora il primo valore di FVC registrato dopo l'esecuzione della TC.

3.6.2 Model Training

Per esplorare gli effetti dell'aggiunta del primo valore di FVC, sono stati condotti diversi training con variazioni nei parametri chiave del modello, come il LR e il batch size, come descritto dettagliatamente nella Tabella 3.6. Questa procedura mirava a individuare la combinazione ottimale di iperparametri in grado di massimizzare le prestazioni del modello, tenendo conto della nuova feature introdotta.

Durante ciascun allenamento, le immagini mediate lungo l'asse z sono state sottoposte all'estrazione di 32 feature tramite i ViT. Queste caratteristiche sono state successivamente integrate con le feature tabulari, comprensive del primo valore di FVC. Tale approccio integrato ha consentito al modello di sfruttare simultaneamente le informazioni spaziali derivanti dalle immagini e i dati tabulari, con l'obiettivo di massimizzare l'utilità delle diverse fonti informative disponibili.

Batch Size	Learning Rate	Fold	Score
32	1×10^{-3}	4	-6.774
32	1×10^{-4}	4	-6.852
32	1×10^{-5}	4	-7.01
16	1×10^{-3}	4	-6.798
16	1×10^{-4}	4	-6.828
16	1×10^{-5}	4	-6.921
8	1×10^{-3}	4	-6.965
8	1×10^{-4}	4	-7.04
8	1×10^{-5}	4	-7.13

Tabella 3.6. Differenti Batch Size e Leraning Rate con il miglior Score ottenuto dal Validation Set usando immagine media con l'aggiunta dell'FVC iniziale come valore tabulare

3.6.3 Risultati

Dopo aver scaricato il modello ottimale ottenuto con un batch size di 32 e un LR di 1×10^{-3}, è stata effettuata una submission su Kaggle. L'obiettivo era valutare se l'introduzione della nuova feature tabulare, rappresentata dal primo valore di FVC, apportasse miglioramenti significativi, come ci si attendeva in base alle analisi e ai training precedenti. Questa fase di sperimentazione e valutazione su Kaggle fornisce un'importante verifica della validità dell'approccio e della rilevanza della nuova feature nel migliorare le prestazioni predittive del modello nel contesto della sfida proposta. Sfortunatamente, dai risultati della submission, è emerso un valore di -7.0272 (Tabella

Kaggle Submission	Private Score	Public Score
Succeeded	-7.0272	-7.1557

Tabella 3.7. Submission esperimento 6

3.7) che non ha migliorato i risultati precedenti ottenuti -6.8878. Questo indica che l'introduzione della nuova feature tabulare, rappresentata dal primo valore di FVC, non ha portato a miglioramenti significativi nelle prestazioni predittive del modello su Kaggle.

Questa valutazione è cruciale per comprendere l'efficacia della nuova feature e conferma che, nonostante le prove durante il training e la validazione, l'aggiunta del primo valore di FVC potrebbe non essere rilevante o addirittura dannosa quando si tratta di generalizzare su nuovi dati. Questi risultati forniscono importanti indicazioni per il processo decisionale, contribuendo a raffinare l'approccio e concentrarsi su quelle caratteristiche che realmente contribuiscono al miglioramento delle prestazioni del modello.

3.7 Esperimento 7: Utilizzo di Scheduler

Per cercare di migliorare le prestazioni del modello descritto nella sezione 3.5, sono stati implementati due diversi tipi di scheduler per regolare il LR durante il processo di training. Entrambi sono stati implementati utilizzando la libreria PyTorch [18] e sono denominati "MultiStepLR" e "LinearLR".
MultiStepLR:

- Questo scheduler varia il LR in modo specifico a determinati passi predefiniti durante l'allenamento. Questo approccio consente un aggiornamento discretizzato del LR, permettendo un controllo più preciso sulla sua evoluzione nel corso delle iterazioni di training. In sostanza, il LR viene modificato in maniera distinta in corrispondenza di milestone predefinite, offrendo la flessibilità di adattare il LR in risposta alle caratteristiche specifiche dell'andamento della funzione obiettivo nel corso dell'addestramento.

LinearLR:

- Questo scheduler modifica il LR in modo lineare durante il processo di training. L'andamento lineare implica una variazione costante nel LR, con un decremento o incremento costante lungo le iterazioni.

È importante sottolineare che come ottimizzatore è stato utilizzato Adam. Adam [16] è un algoritmo di ottimizzazione stocastico ampiamente utilizzato nelle reti neurali. La sua denominazione deriva dall'acronimo Adaptive moment estimation (stima adattiva del momento). Adam combina concetti

provenienti dagli ottimizzatori di momento e dal metodo di AdaGrad. Utilizza momenti di primo e secondo ordine per adattare il LR per ciascun parametro del modello. I momenti di primo ordine seguono la direzione della pendenza attuale, mentre i momenti di secondo ordine adattano il LR per ciascun parametro in base alla magnitudine storica delle pendenze. Questa adattabilità aiuta Adam ad affrontare problemi di convergenza e ad adattarsi a diverse scale di LR per diversi parametri. L'utilizzo di Adam come ottimizzatore in combinazione con gli scheduler di LR mira a ottimizzare il processo di addestramento del modello, cercando di garantire una convergenza più rapida e una migliore capacità di adattamento ai dati specifici

3.7.1 Data Preparation

Per valutare l'efficacia dei due LR, è stata utilizzata per entrambi la stessa preparazione del dataset descritta nella sottosezione 3.5.1. Questa scelta è stata basata sul fatto che tale composizione ha precedentemente portato a risultati migliori. Utilizzando una metodologia uniforme per la preparazione del dataset, si è cercato di isolare l'effetto specifico della variazione del LR sulle prestazioni del modello. Questo approccio consente una comparazione più diretta tra le due strategie di adattamento del LR, garantendo che eventuali differenze osservate siano attribuibili principalmente alle variazioni nel LR piuttosto che alle differenze nella preparazione del dataset.

3.7.2 Model Training

Per valutare quale dei due scheduler, MultiStepLR o LinearLR, portasse a risultati migliori, sono stati allenati due modelli con Adam come ottimizzatore, un batch size di 32 e variazione del LR da 1×10^{-3} a 1×10^{-5} (Figura 3.10 Figura 3.11). Successivamente, i modelli con i risultati migliori sono stati salvati e testati su Kaggle per determinare quale fosse più adatto. L'obiettivo era anche verificare se l'adozione di uno di questi scheduler avesse effettivamente migliorato le performance complessive del modello.

Il processo di sperimentazione coinvolgeva l'addestramento di due modelli con diversi valori di LR, seguendo la strategia di ciascuno degli scheduler. Salvando i modelli con i migliori risultati ottenuti durante il training, è stato possibile confrontare le performance di entrambi i modelli su Kaggle, valutando quale configurazione di scheduler e LR portasse a una maggiore accuratezza nella predizione della sfida Kaggle.

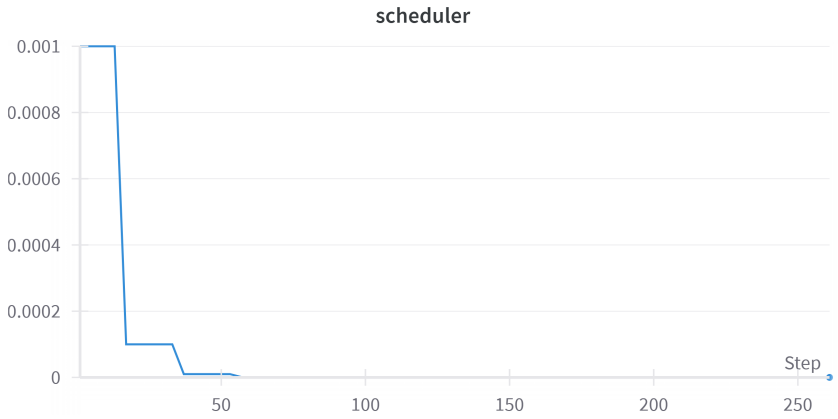

Figura 3.10. Scheduler Multistep per un solo fold

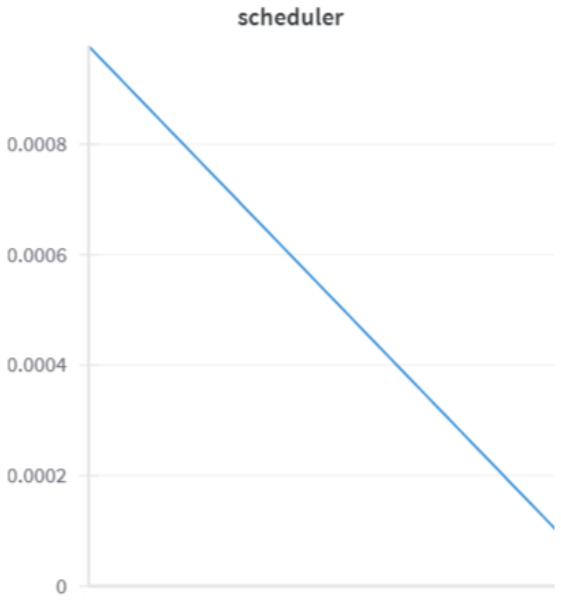

Figura 3.11. Scheduler Linear per un solo fold

3.7.3 Risultati

Kaggle Submission	Private Score	Public Score
Succeeded	-6.8772	-6.9175

Tabella 3.8. Submission con Multistep

Kaggle Submission	Private Score	Public Score
Succeeded	-6.875	-6.9835

Tabella 3.9. Submission con Linear

Come si può evincere dalle tabelle 3.9 e 3.8, entrambi i modelli allenati con gli scheduler LinearLR e MultiStepLR mostrano miglioramenti rispetto al modello precedente (-6.8878 Tabella 3.5). Inoltre, si può notare che l'utilizzo dello scheduler lineare sembra essere associato ad una lievemente migliorata capacità di generalizzazione rispetto a MultiStepLR.

Questi risultati suggeriscono che la strategia di adattamento del LR attraverso un andamento lineare ha contribuito positivamente alle performance del modello. Tuttavia, è importante considerare che la scelta tra LinearLR e MultiStepLR potrebbe dipendere dalla specifica natura del dataset e della sfida Kaggle, e potrebbe essere soggetta a ulteriori esperimenti per determinare la configurazione ottimale per il modello in questione. La scelta dello scheduler può essere influenzata da fattori come la complessità del problema, la dimensione del dataset, e la dinamica delle pendenze durante il training.

3.8 Esperimento 8: Crop del Volume Originale

L'approccio proposto da Alexander Wong et al. [26] presenta il modello denominato Fibrosis-net, basato sulle CNN. Nel contesto di questo modello, le caratteristiche estratte dall'immagine sono ottenute solamente dal "55% inferiore del subset della TC, dove tipicamente si manifesta la fibrosi polmonare" [26]. Per replicare questa strategia nell'esperimento in oggetto, si è scelto di adottare la medesima prassi, utilizzando esclusivamente il 55% inferiore del subset delle TC, prima di procedere con la media dell'immagine lungo l'asse z.

3.8.1 Data Preparation

Anche in questa specifica sperimentazione, sono stati mantenuti gli stessi attributi tabulari, ovvero sesso e stato tabagico (entrambi elaborati mediante la tecnica di one-hot encoding), mentre l'età è stata mediata. L'immagine fornita in input al ViT è stata sottoposta a una pre-elaborazione specifica, consistente nell'ottenere una singola slice che rappresenta la media lungo l'asse z del 55% inferiore del subset della TC (Figura3.12). Tale procedura è stata adottata con l'obiettivo di ottenere una rappresentazione volumetrica sintetizzata in una sola slice, aumentando la probabilità di isolare la regione informativa del polmone pertinente alla IPF.

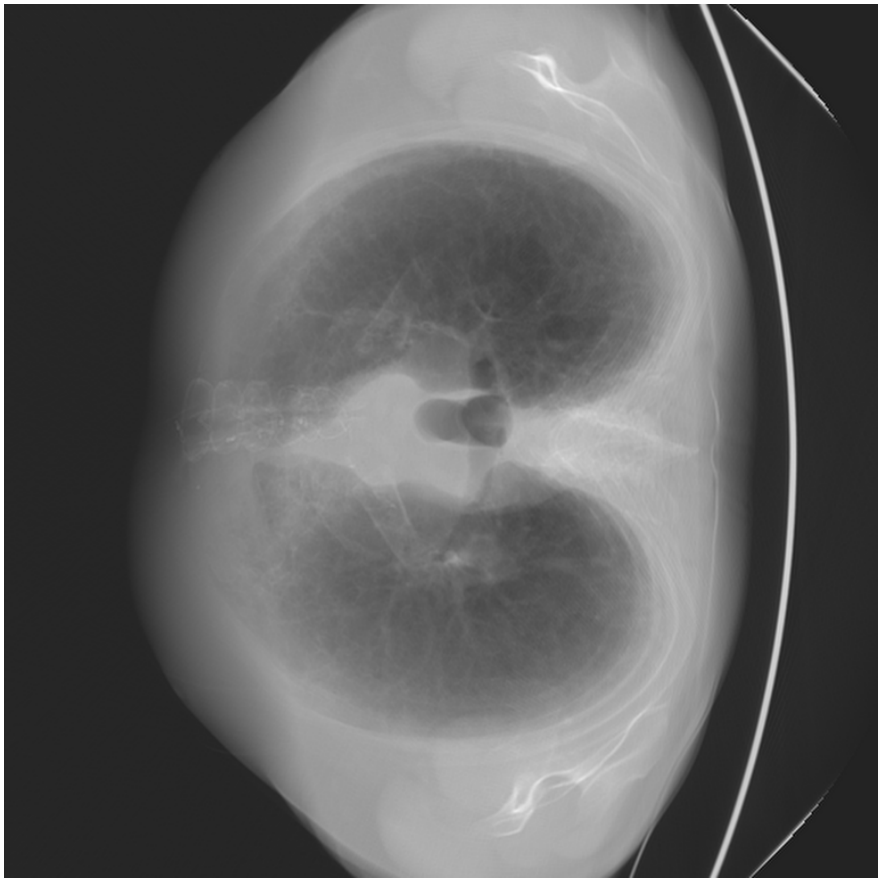

Figura 3.12. Slice della media del 55% del volume del paziente ID00026637202179561894768

3.8.2 Model Training

In questo esperimento, è stata mantenuta l'architettura precedentemente utilizzata. Dall'input, il ViT estrae 32 feature, che vengono concatenate alle feature tabulari. Una volta unite, passano attraverso uno strato lineare (come illustrato in Figura 3.5), con un solo neurone in output per prevedere la pendenza dell'andamento della FVC.

Come ottimizzatore è stato utilizzato Adam, con un LR che varia da 1×10^{-3} a 1×10^{-5} attraverso uno scheduler lineare con pendenza negativa di 45 gradi. Per garantire la robustezza del modello, sono stati utilizzati i fold con precisione 5, assegnando l'80% del dataset al training e il restante 20% alla validazione. Complessivamente, sono stati addestrati 3 modelli, ciascuno con diverse dimensioni di batch, come riportato nella Tabella 3.10. Dai risultati è emerso che il modello più performante è stato ottenuto nel fold 4 con un batch di dimensione pari a 16 (Figura 3.13 Figura 3.14).

Batch Size	Fold	Score
32	4	-6.543
16	4	-6.422
8	4	-6.545

Tabella 3.10. Differenti Batch Size con il miglior Score ottenuto dal Validation Set usando Slice mediata dal 55% inferiore del volume della TC

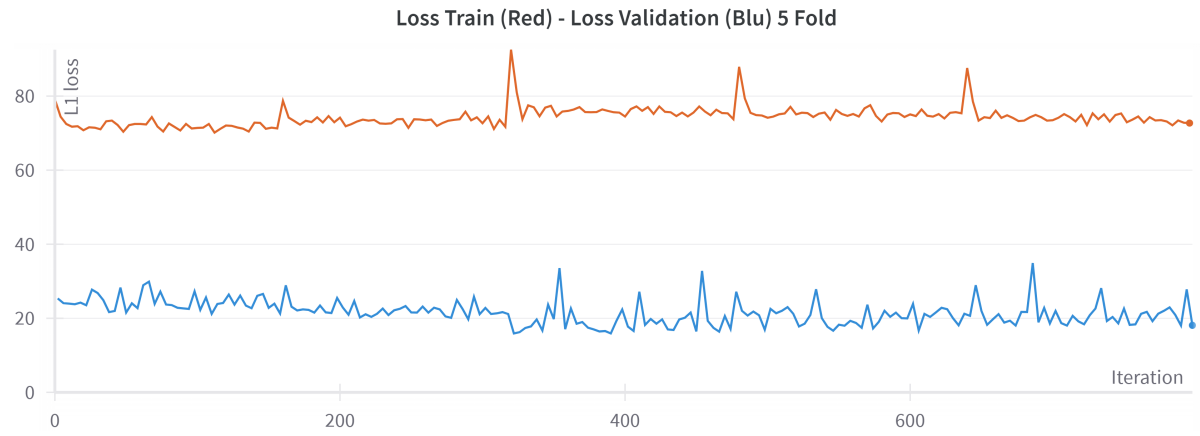

Figura 3.13. L1 loss Training e Validation dei 5 fold

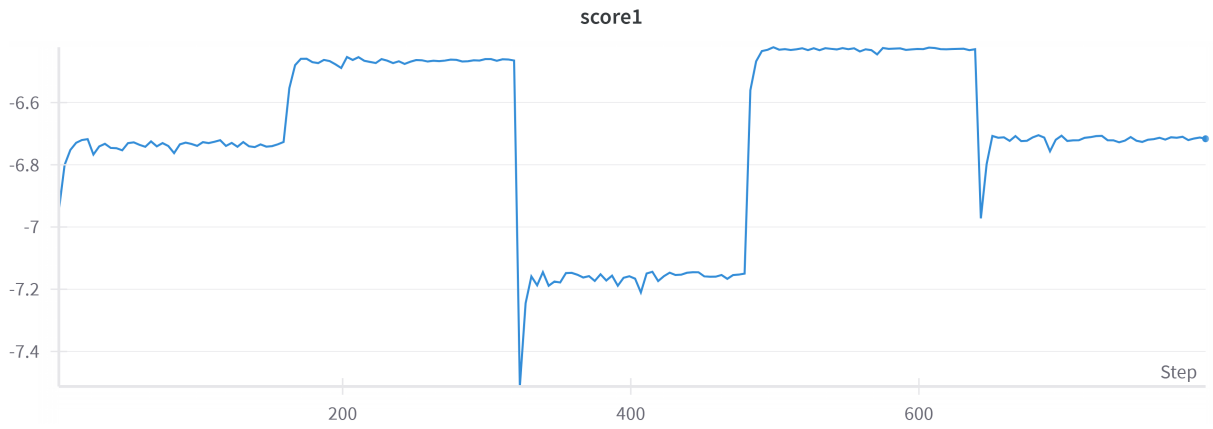

Figura 3.14. Score dei 5 fold

3.8.3 Risultati

Il modello generato dal fold 4 è stato scaricato e utilizzato durante la fase di inference per sottomettere i risultati sulla piattaforma Kaggle, come indicato nella Tabella 3.11. Questa strategia ha prodotto risultati notevolmente superiori rispetto a quelli ottenuti in precedenza (-6.875, come evidenziato nella Tabella 3.9). Tale miglioramento indica che concentrarsi sul 55% inferiore del subset della TC può effettivamente rappresentare la regione più informativa per estrarre feature significative relative alla IPF. Con uno score di -6.868, il modello ha conquistato la medaglia di bronzo nella leaderboard della challenge.

Kaggle Submission	Private Score	Public Score
Succeeded	-6.868	-6.9209

Tabella 3.11. Submission esperimento 8

3.9 Esperimento 9: Segmentazione dell'Immagine

Osservando i risultati ottenuti utilizzando il 55% del volume, si è presa la decisione di segmentare le slice del polmone prima di procedere alla media lungo l'asse z. Questa nuova strategia mira a isolare in modo più efficace la zona di interesse, escludendo le regioni che non sono rilevanti per il polmone. Per implementare questa segmentazione delle slice, si è seguita la strategia proposta da Bhat et al. [4], i quali hanno partecipato alla challenge raggiungendo il quarto posto nella leaderboard.

3.9.1 Data Preparation

Il metodo di segmentazione polmonare inizia con la normalizzazione delle intensità dei pixel nell'immagine TC, riducendo le variazioni di luminosità attraverso la sottrazione della media e la divisione per la deviazione standard. Successivamente, si seleziona un'area centrale dell'immagine e si applica un algoritmo di KMeans con due cluster per separare i tessuti molli/densi (foreground) dall'aria/-polmoni (background). Il risultato della segmentazione viene sottoposto a operazioni morfologiche, come erosione e dilatazione [21], per rimuovere dettagli indesiderati e garantire che la maschera risultante corrisponda accuratamente alla forma dei polmoni.Gli step successivi sono:

- Colorazione delle segmentazioni: Dopo la segmentazione polmonare, si procede al labeling delle diverse regioni dell'immagine utilizzando numeri o label. Ciascuna label rappresenta un cluster o una specifica porzione dell'immagine risultante dalla segmentazione.
- Mappatura delle label a Colori [7]: Ogni label viene associata a un colore specifico attraverso un'appropriata mappatura. Ad esempio, potrebbe essere definito un dizionario che associa ogni numero di label a un colore RGB unico. Questa mappatura conferisce a ciascuna regione dell'immagine un colore distinto.
- Applicazione della Mappatura all'Immagine Etichettata: L'immagine etichettata viene quindi elaborata applicando la mappatura dei colori. Ogni pixel con una label specifica viene colorato con il colore corrispondente assegnato a quella label.
- Visualizzazione dell'Immagine Colorata: L'immagine risultante, ora colorata in base alle etichette, viene visualizzata (Figura 3.15). Questo permette di distinguere chiaramente le diverse regioni dell'immagine in base ai colori assegnati.
- Identificazione della label Polmonare: Osservando l'immagine colorata, la label associata alla regione dei polmoni viene identificata. Questa label rappresenta la porzione di interesse in cui sono presenti i polmoni.
- Creazione della Maschera Polmonare: Una maschera binaria viene quindi creata, assegnando il valore 1 ai pixel corrispondenti alla regione polmonare e 0 agli altri pixel.
- Applicazione della Maschera Polmonare: La maschera polmonare viene applicata all'immagine originale o all'immagine di segmentazione polmonare colorata, mantenendo solo i pixel associati alla regione polmonare.
- Visualizzazione dei Polmoni Selezionati: L'immagine risultante mostra selettivamente la regione polmonare rispetto alle altre, evidenziandola visivamente e consentendo un'analisi mirata (Figura 3.15).

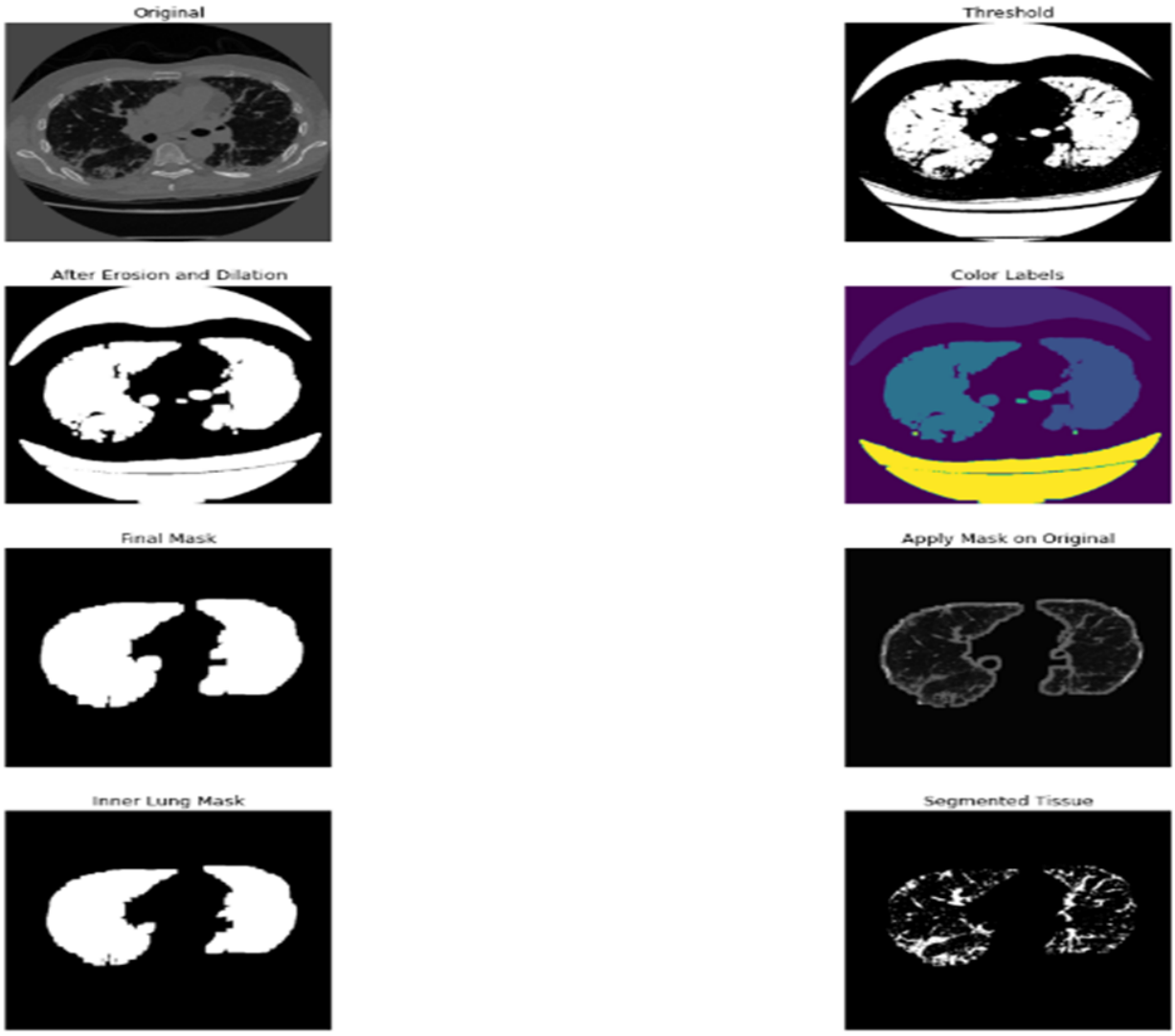

Figura 3.15. Applicazione segmentazione [4] Patient ID- ID000076372021774119564 30 Slice -15.dcm

Dopo aver applicato il metodo di segmentazione a ciascuna slice delle immagini nel dataset della challenge, la maschera risultante è stata moltiplicata per l'immagine originale per ottenere una regione di interesse circoscritta per ogni slice. Quest'operazione è stata eseguita individualmente per ciascuna slice del volume, creando così un volume segmentato. Successivamente, è stata calcolata la media lungo l'asse z, limitandola al 55% inferiore del subset della TC. In questo modo, si è ottenuta una rappresentazione focalizzata e segmentata del polmone, preservando solo le regioni di interesse identificate dalla maschera in ciascuna slice (Figura 3.16).

Utilizzando questa immagine risultante come input per il ViT al fine di estrarre le caratteristiche dell'immagine, non sono state apportate modifiche alle feature tabulari. Le caratteristiche estratte dall'immagine e le feature tabulari sono aggregate successivamente, seguendo la stessa procedura utilizzata nell'esperimento precedente (Esperimento 8, vedi sezione 3.8.1).

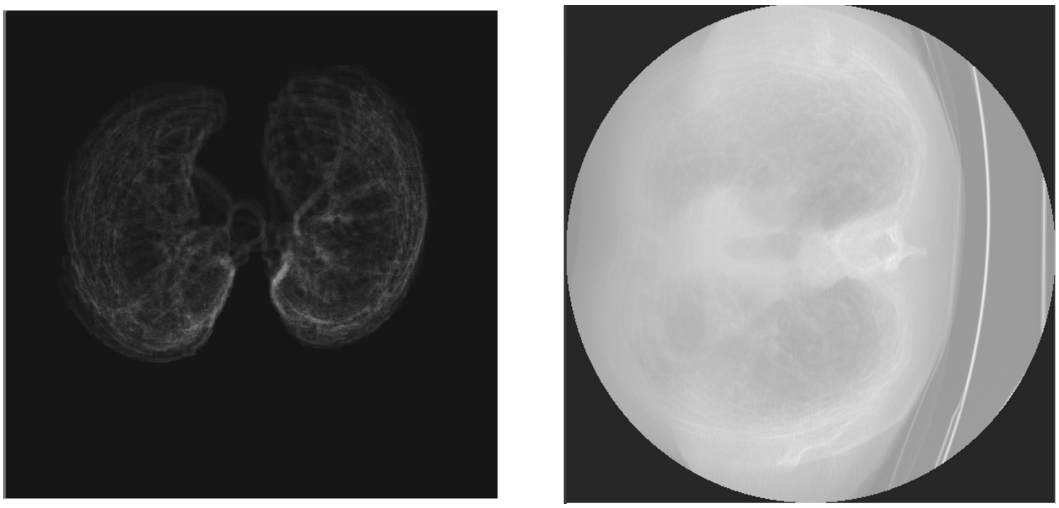

Figura 3.16. Confronto tra le immagini delle slice ottenute mediante la media del 55% inferiore del subset della TC per il paziente con ID00183637202241995351650. Sulla sinistra, è stata applicata la segmentazione, mentre sulla destra, no

3.9.2 Model Training

Per valutare l'utilità della segmentazione, sono stati eseguiti diversi training utilizzando uno scheduler lineare con una pendenza negativa di 45° per variare il LR ad ogni iterazione, partendo da 1×10^{-3} e diminuendo fino a 1×10^{-5}. Sono stati condotti tre training, variando il Batch Size (Tabella 3.12). Come ottimizzatore è stato utilizzato Adam, e la loss è stata calcolata utilizzando la L1 loss sia nel training che nella fase di validazione.

Batch Size	Fold	Score
32	4	-6.449
16	4	-6.432
8	4	-6.495

Tabella 3.12. Differenti Batch Size con il miglior Score ottenuto dal Validation Set usando Slice mediata dal 55% inferiore del volume segmentato della TC

3.9.3 Risultati

Il processo sperimentale ha comportato il download del miglior modello ottenuto, seguito da una submission su Kaggle per valutare l'efficacia dell'esperimento. Tuttavia, le submission sono risultate fallite a causa del requisito della challenge che richiede un notebook senza accesso a Internet. Questa limitazione ha causato problemi con il modulo "morphology" di scikit-image [25], essenziale per ottenere le segmentazioni delle slice.

Poiché non è stato possibile testare e confrontare i risultati sul test set, si è optato per valutarli sul validation set. Come evidenziato nella Tabella 3.12, il miglior risultato raggiunto da questo esperimento in termini di score è di -6.432, mentre nel miglior modello ottenuto nell'esperimento 8 in fase di validazione si è ottenuto un valore di -6.422 (Tabella 3.11). Da questa prima analisi sembra che, nonostante la segmentazione focalizzi maggiormente l'attenzione sulle zone di interesse, non si siano ottenute migliorie rispetto alle aspettative. È possibile che ulteriori valutazioni e analisi

siano necessarie per comprendere appieno l'impatto della segmentazione e determinare se ci siano scenari specifici in cui può portare vantaggi. Per questo lavoro si è deciso di non utilizzarle in quanto aumentavano il costo computazionale senza migliorare le prestazioni del modello.

3.10 Esperimento 10: Modifica Normalizzazione Età ed Embedding dei Modelli

Dato che le modifiche apportate all'immagine non hanno portato miglioramenti significativi, ad eccezione di quelle applicate nell'esperimento 8, si è deciso di procedere con la modifica delle feature tabulari utilizzate nell'esperimento 8. Questo approccio mira a esplorare come le variazioni nella rappresentazione delle feature tabulari possano influenzare le prestazioni del modello, cercando di individuare combinazioni ottimali di caratteristiche che portino a una migliore previsione della progressione della IPF.

3.10.1 Data Preparation

L'unica modifica apportata alla preparazione del dataset, rispetto a quella utilizzata nell'esperimento 8 (Sezione 3.8.1), è stata una differente normalizzazione della feature tabulare relativa all'età. In particolare, si è determinato il massimo valore di età tra i vari pazienti e il minimo valore di età. Successivamente, è stata applicata la seguente formula di normalizzazione [14]:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Questa normalizzazione consente di trasformare la feature dell'età in un intervallo compreso tra 0 e 1, migliorando la stabilità del modello e facilitando il processo di apprendimento durante il training.

3.10.2 Model Training

Anche per questo esperimento sono stati condotti 3 training utilizzando l'ottimizzatore Adam, uno scheduler lineare con una pendenza negativa di 45° e 3 diverse dimensioni di batch: 32, 16, 8. Come funzione loss si è utilizzata la L1 loss.

Questo approccio consente di esplorare come variazioni nella dimensione del batch possano influire sul processo di addestramento e sulle prestazioni generali del modello. L'uso di uno scheduler lineare con una pendenza negativa di 45° può contribuire a un fine-tuning più efficace dei pesi del modello durante le iterazioni di training. La scelta di Adam come ottimizzatore e della L1 loss come funzione di perdita è coerente con l'approccio adottato negli esperimenti precedenti. Come

Batch Size	Fold	Score
32	4	-6.476
16	4	-6.399
8	4	-6.445

Tabella 3.13. Differenti Batch Size con il miglior Score ottenuto dal Validation Set esperimento 10

evidenziato nella Tabella 3.13, il miglior modello è stato ottenuto nel fold 4 utilizzando un batch di dimensione 16. Per questo esperimento, sono stati scaricati tutti i modelli ottenuti dai 5 fold, ciascuno allenato con un batch di dimensione 16. Questa scelta di utilizzare modelli provenienti

da diversi fold consente di valutare la robustezza del modello rispetto alle variazioni nei dati di training. Inoltre, l'embedding dei 5 modelli consente di ottenere una rappresentazione aggregata delle informazioni apprese da ciascun fold, potenzialmente migliorando la generalizzazione e le prestazioni complessive del modello.

3.10.3 Risultati

Inizialmente, è stata effettuata una submission su Kaggle utilizzando solo il modello scaricato dal fold 4 (Tabella 3.14). Da questa submission è emerso che la modifica apportata alla normalizzazione dell'età ha avuto un impatto positivo in termini di score, rispetto al miglior modello ottenuto nell'esperimento 8. Questo risultato indica che la variazione nella normalizzazione dell'età ha contribuito a migliorare le prestazioni del modello nella previsione della progressione della fibrosi polmonare, confermando l'efficacia di tale modifica.

Kaggle Submission	Private Score	Public Score
Succeeded	-6.8676	-6.9209

Tabella 3.14. Submission esperimento 10 fold 4

Successivamente è stato effettuato un embedding dei 5 modelli ottenuti dai 5 fold del k-fold. Ciascun modello è stato applicato singolarmente su ogni paziente del set di test, e successivamente è stata calcolata la media delle 5 slope ottenute. Questo approccio mira a ottenere una migliore generalizzazione aggregando le informazioni apprese dai diversi fold durante il training. La media delle slope ottenute dai modelli può contribuire a ridurre l'impatto di variazioni casuali nei dati di training di ciascun fold, migliorando così la robustezza del modello complessivo.

Kaggle Submission	Private Score	Public Score
Succeeded	-7.0272	-7.1557

Tabella 3.15. Submission esperimento 10 merge dei 5 modelli

Dalla Tabella 3.15 emerge che il modello ottenuto tramite il merge dei 5 modelli ha ottenuto uno score di -7.0272, risultato decisamente peggiore rispetto alle aspettative. Questo è in netto contrasto con lo score migliore ottenuto utilizzando solo il modello del fold 4 (-6.8676). Tale risultato suggerisce che, in questo caso specifico, la media delle slope ottenute dai modelli dei diversi fold non ha portato a una migliore generalizzazione e performance complessive.

3.11 Esperimento 11: Utilizzo delle Radial Basis Functions

Durante questo esperimento, è stato esplorato l'impatto della sostituzione del fully connected layer finale con un RBF sulle performance del modello. La scelta di utilizzare una RBF al posto del fully connected laye finale potrebbe introdurre nuove dinamiche di apprendimento nel modello, influenzando la sua capacità di apprendere relazioni complesse nei dati. Questa modifica rappresenta un'alternativa nella struttura del modello, e l'obiettivo è valutare se tale cambio influisca positivamente sulle prestazioni rispetto alla configurazione tradizionale con il fully connected layer.

3.11.1 Data Preparation

Per la preparazione dei dati, non sono state apportate delle modifiche rispetto alla procedura utilizzata nell'esperimento 10, come descritto nella sezione dedicata Data Preparation(3.10.1).

3.11.2 Model Training

Per determinare la combinazione ottimale degli iperparametri per l'implementazione della funzione RBF, è stato condotto un esperimento in cui il numero di centri e i valori di sigma sono stati variati incrementalmente. Durante ogni fase di addestramento, è stato calcolato l'errore assoluto medio (MAE) tra i valori reali (GT) e le predizioni del modello. Il modello migliore è stato selezionato in base al MAE più basso. Da notare che tali modelli ricevono in input le caratteristiche estratte dal ViT, le quali sono aggregate alle feature tabulari. Questo approccio consente al modello di integrare informazioni provenienti da entrambi gli aspetti, visuale e tabulare, per ottimizzare le prestazioni complessive del sistema.

Dai risultati dell'esperimento (Tabella 3.17), emerge che la combinazione ottimale dei parametri per l'implementazione della funzione RBF è caratterizzata da un Minimo MAE Medio di 5.420037, con i seguenti valori associati:

- Sigma: 0.850
- Numero di Centri: 132

Questi parametri rappresentano la configurazione che ha prodotto le prestazioni migliori in termini di errore medio assoluto durante il processo di addestramento. La scelta di questi valori ottimali può contribuire a massimizzare l'efficacia della RBF nell'ambito specifico della predizione della progressione della fibrosi polmonare.

3.11.3 Risultati

Dopo l'individuazione della combinazione ottimale dei parametri per il modello RBF, è stata eseguita una prova del modello tramite una submission su Kaggle. Durante questa fase, è stato utilizzato il modello configurato con i parametri ottimizzati per valutare le prestazioni.

Kaggle Submission	Private Score	Public Score
Succeeded	-8.4809	-8.6692

Tabella 3.16. Submission esperimento 11

Dai risultati della submission, come evidenziato nella Tabella 3.16, si osserva un notevole peggioramento delle prestazioni del modello con l'utilizzo del RBF. Lo score ottenuto di -8.4809 è significativamente più basso rispetto allo score ottenuto dal miglior modello senza l'uso del RBF, che era di -6.8676. È importante sottolineare che la metrica utilizzata (1.3) è in scala logaritmica.

Sigma	Num Centers	MAE Mean
0.100	2	5.523270
0.100	7	5.523270
0.100	12	5.523270
0.100	17	5.523270
0.100	22	5.523270
0.100	27	5.523270
0.100	32	5.523270
0.100	37	5.523270
0.100	42	5.523270
0.100	47	5.523270
⋮	⋮	⋮
0.950	122	5.472397
0.950	127	5.420220
0.950	132	5.503790
0.950	137	5.471154
1.000	2	5.522473
1.000	7	5.522765
1.000	12	5.488241
1.000	17	5.484763
1.000	22	5.493216
1.000	27	5.589926
1.000	32	5.522643
1.000	37	5.527410
1.000	42	5.499669
1.000	47	5.478541
1.000	52	5.503628
1.000	57	5.565581
1.000	62	5.551499
1.000	67	5.489645
1.000	72	5.484334
1.000	77	5.520588
1.000	82	5.547077
1.000	87	5.498299
1.000	92	5.440788
1.000	97	5.461324
1.000	102	5.480016
1.000	107	5.423836
1.000	112	5.533311
1.000	117	5.473283
1.000	122	5.474999
1.000	127	5.483226
1.000	132	5.474325
1.000	137	5.470300

Tabella 3.17. Vari training con variazione di Sigma e di Num Centers e il valore de MAE medio

Capitolo 4

Metodi e Risultati: Confutazione del Task

In questo capitolo, viene condotta un'analisi approfondita al fine di valutare la validità e l'adeguatezza del dataset fornito dalla challenge per il task di regressione prescritto. L'obiettivo principale è mettere in discussione la fattibilità della previsione della variazione della FVC utilizzando le informazioni contenute nel dataset. La necessità di indagare deriva dal fatto che nonostante diverse modifiche volte a migliorare lo score, non si è riscontrata alcuna miglioria nei risultati, evidenziando la possibilità di un bias tra i vari esperimenti condotti nel capitolo precedente. A tal fine, saranno descritti diversi esperimenti e test condotti per valutare la coerenza e la robustezza del task di regressione.
Overview dei test condotti:

- Test di Ablazione: Sono stati eseguiti test di ablazione in cui sono state escluse o modificate determinate feature o componenti del dataset al fine di valutare l'impatto diretto sul FVC. L'obiettivo è comprendere quali informazioni siano fondamentali per la previsione accurata della FVC e identificare eventuali componenti ridondanti o insignificanti.
- Test di Correlazione: Saranno esaminati test di correlazione per valutare la relazione tra le diverse feature del dataset e l'output desiderato. Ciò aiuterà a identificare la presenza di correlazioni significative o, al contrario, la mancanza di connessioni dirette tra alcune feature e la FVC.

Il workflow seguito per ciascun esperimento è dettagliato nella Figura 4.1. Dopo la selezione del test da condurre, è stata eseguita l'implementazione su Jupyter Notebook garantendo la ripetibilità dei risultati ottenuti in ciascun test.

I risultati degli esperimenti saranno attentamente esaminati, con particolare enfasi sulle loro implicazioni per la validità del task di regressione nel contesto del dataset fornito. Attraverso un'analisi approfondita, il capitolo dedicato alla "Confutazione del Task" si propone di mettere in luce la robustezza e la coerenza del dataset, offrendo una prospettiva critica sull'adeguatezza del task di regressione proposto.

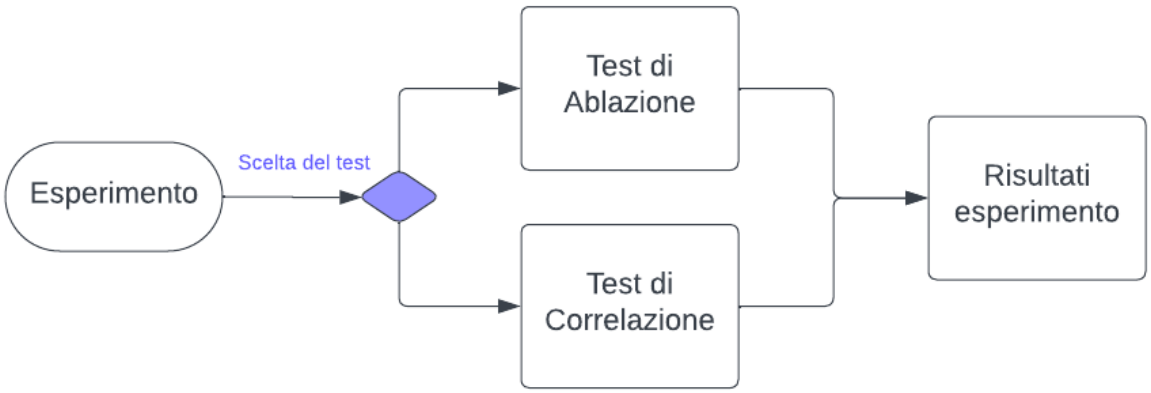

Figura 4.1. Workflow confutazione del task.

4.1 Esperimento 1: Correlazione tra Feature Tabulari e Ground Truth

In questa sezione, sarà delineato l'esperimento condotto per esplorare la correlazione tra le feature tabulari e il GT, al fine di determinare la rilevanza di specifiche caratteristiche rispetto ad altre. Oltre alle informazioni relative al sesso, all'età e allo stato tabagico, è stata introdotta una nuova feature tabulare che amalgama le informazioni riguardanti sesso e stato tabagico. Per realizzare ciò, sono stati assegnati i seguenti valori:

- 0 : per i soggetti di sesso maschile fumatori
- 1 : per i maschi ex-fumatori
- 2 : per i maschi non fumatori
- 3 : per i maschi con stato di fumo non definito
- 4 : per le femmine fumatrici
- 5 : per le femmine ex-fumatrici
- 6 : per le femmine non fumatrici
- 7 : per le femmine con stato di fumo non definito.

Per ciascun paziente, è stato applicato il metodo dei minimi quadrati della SVD per ottenere la retta di regressione. La pendenza di questa retta è stata considerata come il GT per il successivo confronto e l'analisi delle relazioni tra le variabili coinvolte.

4.1.1 Correlazione

Dopo aver completato la fase di organizzazione dei dati, sono state calcolate le correlazioni di Spearman tra le feature tabulari (sesso, età, smoking status, sesso_smoking status) e la slope (il GT), insieme ai relativi valori di p-value. Questa analisi mira a valutare la forza e la direzione delle

relazioni tra le variabili tabulari e la pendenza, nonché a determinare l'eventuale significatività statistica di tali correlazioni.

4.1.2 Risultati

Dalla tabella 4.1, emerge che il massimo valore di correlazione, in valore assoluto, è di 0.133. Questo valore relativamente basso indica una debole associazione tra le feature tabulari considerate (sesso, età, smoking status, sex_smoking status) e la slope (GT). La mancanza di correlazioni significative, evidenziata dai valori di p-value superiori a 0.05, suggerisce che non vi sia supporto statistico per affermare l'esistenza di un legame significativo tra le variabili tabulari e la pendenza.

	Età	Sesso	Smoking Status	Sex_Smoking status
Slope	Corr: 0.106 p: 0.158	Corr: -0.0533 p: 0.48	Corr: -0.133 p: 0.077	Corr: -0.0556 p: 0.4632

Tabella 4.1. Correlazioni e P-value tra Slope e Feature tabulari

4.2 Esperimento 2: Training Senza Feature Tabulari

I risultati ottenuti nella sezione precedente hanno indicato una mancanza di correlazione tra le feature tabulari e la slope. Di conseguenza, si è proceduto ad adottare un approccio diverso, prendendo il miglior modello identificato nel Capitolo 3 e riallenandolo escludendo le feature tabulari. In questo nuovo contesto, il modello è stato addestrato utilizzando esclusivamente le informazioni provenienti dalle immagini mediate, allo scopo di valutare l'efficacia di tale approccio in assenza di contributi dalle variabili tabulari.

4.2.1 Model Training

Per questo esperimento, sono stati mantenuti gli stessi parametri utilizzati per ottenere il miglior modello nel Capitolo 2. Di conseguenza, sono stati impiegati l'ottimizzatore Adam, la L1 loss, un batch size di 16 e uno scheduler lineare. Questa coerenza nella configurazione dei parametri consente una comparazione diretta tra il modello originale, che includeva le feature tabulari, e la nuova implementazione che esclude tali variabili, fornendo così una valutazione chiara dell'impatto della rimozione delle feature tabulari sull'efficacia del modello.

4.2.2 Risultati

Per valutare l'importanza delle feature tabulari, è stata condotta un'analisi dell'andamento dello score durante il training, confrontando tale andamento con quello ottenuto durante il training del Best Model, come illustrato nella Figura 4.2. Questo confronto mira a evidenziare eventuali variazioni nella performance del modello quando le feature tabulari sono escluse, consentendo così una valutazione chiara dell'impatto di tale rimozione sulla capacità predittiva complessiva del modello. Come evidenziato dall'andamento dei due score (Figura 4.2), non emerge una differenza significativa tra i due trend. Sembrerebbe che la rimozione delle feature tabulari non abbia avuto un impatto significativo, né positivo né negativo, sull'andamento complessivo dello score durante il training. Questo suggerisce che il modello, anche senza l'inclusione di tali feature, sia in grado di mantenere una performance paragonabile a quella ottenuta dal Best Model nel Capitolo 3. La

Figura 4.2. Confronto Best Model e Best Model Senza TAB

mancanza di variazioni sostanziali potrebbe indicare che, per il compito specifico, le informazioni tabulari potrebbero non essere determinanti o potrebbero essere compensate efficacemente dalla rappresentazione delle sole immagini mediate nel modello.

4.3 Esperimeto 3: Training con Solo Immagini Casuali

Dopo aver confermato che le feature tabulari sono superflue per la previsione della slope, è stata eseguita un'analisi per valutare quanto le immagini contribuiscano al processo predittivo. A tale scopo, è stato condotto un training utilizzando solo immagini con un rumore bianco completamente casuale. Questa procedura mira a valutare la robustezza del modello alle informazioni contenute nelle sole immagini, escludendo qualsiasi segnale utile e introducendo un elemento di casualità nel processo di apprendimento.

4.3.1 Model Training

Anche durante questo test di ablazione, sono stati mantenuti invariati gli stessi parametri utilizzati per ottenere il miglior modello (Capitolo 3). Questa decisione è stata presa per mantenere la coerenza nella configurazione del modello e per esaminare specificamente l'influenza della presenza di rumore bianco sulle immagini, isolandola dagli altri fattori del modello.

4.3.2 Risultati

Come nel test di ablazione precedente, è stato eseguito un confronto tra il valore di score ottenuto durante il training di questo esperimento e lo score del training del miglior modello precedentemente identificato. Dall'analisi rappresentata nella Figura 4.3, emerge che in termini di score non ci sia alcuna differenza sostanziale tra i due training. Questo risultato appare sorprendente, poiché suggerisce che una rete addestrata solo su immagini con rumore bianco casuale ha prodotto risultati paragonabili a una rete addestrata con dati reali. Tale osservazione potrebbe sollevare interrogativi sulla necessità o sulla rilevanza delle informazioni contenute nelle immagini originali rispetto al compito specifico della previsione della slope.

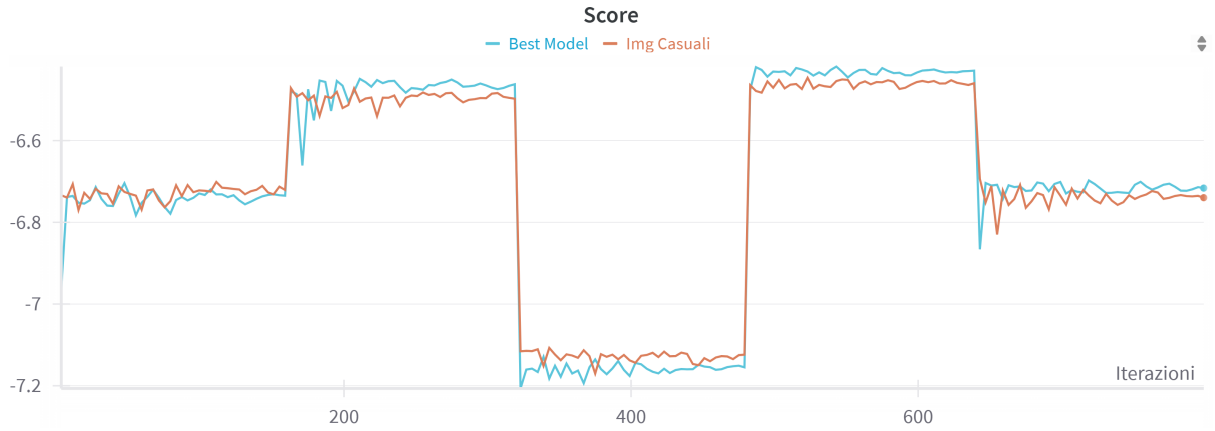

Figura 4.3. Confronto Best Model e Modello con immagini casuali

4.4 Esperimento 4: Variazione del Ground Truth

Poiché né le feature tabulari né le immagini sembravano influenzare significativamente l'output del modello, si è proceduto a modificare casualmente il GT. L'obiettivo è stato quello di verificare se la presenza di eventuali bias persiste anche quando il GT è variato in modo casuale. Questo approccio consente di esaminare se il modello è sensibile a variazioni casuali nel GT e se la sua capacità predittiva viene influenzata da tali perturbazioni, contribuendo a fornire ulteriori informazioni sulla robustezza del modello nei confronti di possibili distorsioni nei dati di addestramento.

4.4.1 Model Training

Anche per questo esperimento, non sono stati variati i parametri del training, mantenendo costanti l'ottimizzatore Adam, la L1 loss, un batch size di 16 e uno scheduler lineare. Questa decisione mira a garantire una coerenza nella configurazione del modello e permette di analizzare specificamente l'effetto della variazione casule del GT.

4.4.2 Risultati

Nel confronto tra il Best Model e il Modello con immagini casuali e GT casuale (Figura 4.4), emerge che il modello addestrato con GT casuale genera un output completamente casuale, con valori significativamente diversi dal Best Model. Tale osservazione suggerisce che, nel contesto specifico, l'assenza del bias precedentemente riscontrato può essere attribuita al GT utilizzato. Si nota che la rete sembra apprendere a restituire il GT fornito, trascurando in gran parte l'influenza dell'input, fenomeno che potrebbe spiegare la mancanza di coerenza nei modelli precedenti.

Dai risultati ottenuti, è emersa la necessità di condurre un'analisi approfondita sui GT di ciascun paziente. Si è notato che, sebbene in molti casi la retta di regressione approssimasse correttamente la distribuzione dei punti dell'FVC, in altri casi non vi era alcuna corrispondenza. Questa disparità sembrava essere attribuibile principalmente al fatto che, durante il calcolo della retta di regressione, non veniva considerata l'intercetta, e il valore di FVC al tempo t=0 veniva forzato (Figura 4.5). Queste condizioni potrebbero contribuire alla variabilità e alla mancanza di accuratezza nella rappresentazione della relazione tra le variabili coinvolte.

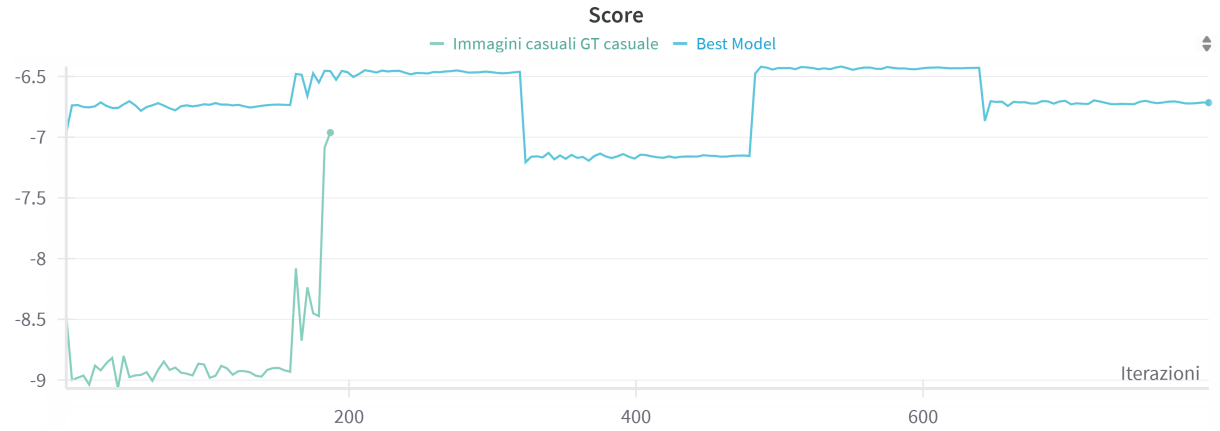

Figura 4.4. Confronto Best Model e Modello con immagini casuali e GT casuale

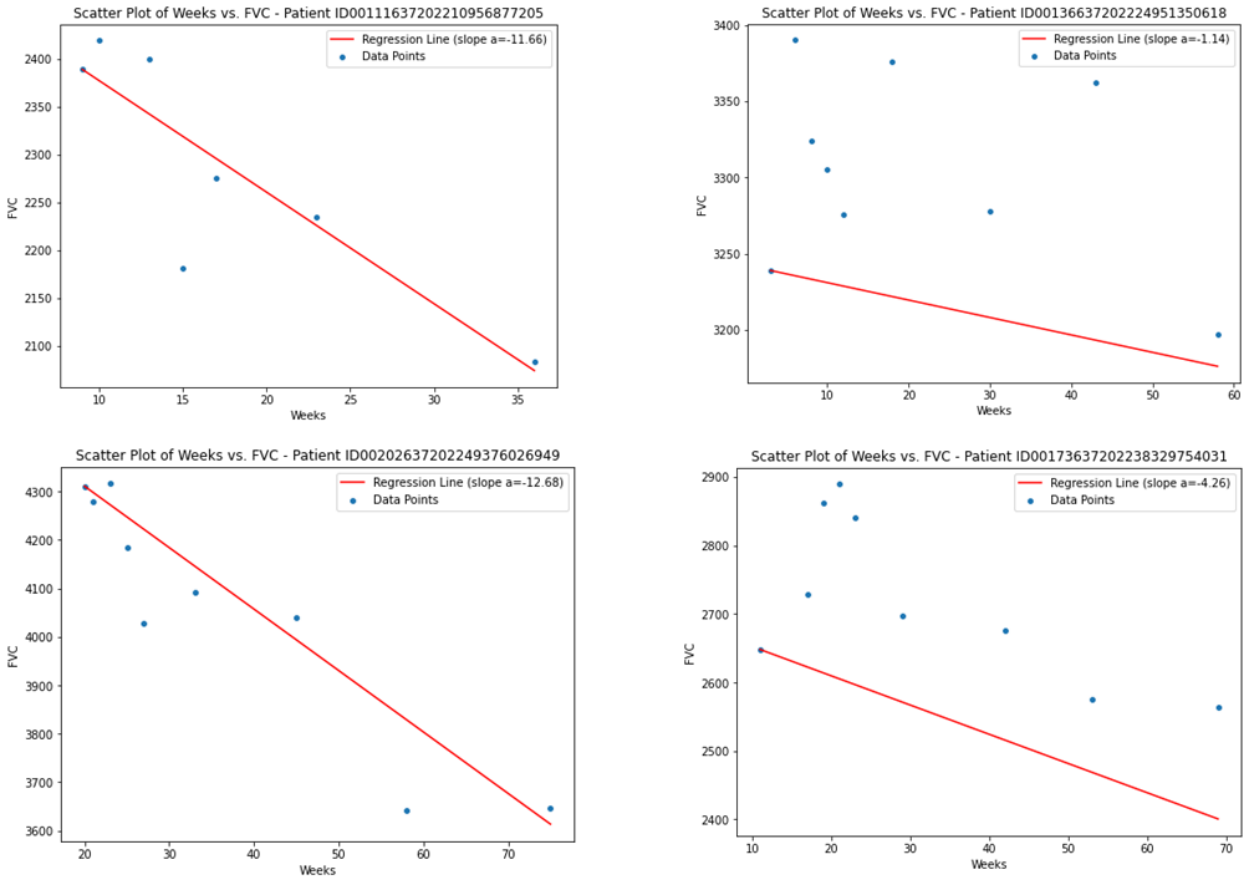

Figura 4.5. Analisi GT per diversi pazienti

4.5 Esperimento 5: Test di Ablazione Modificando il Ground Truth

In questa sezione, è stato condotto un test di ablazione mirato a valutare gli effetti della modifica del GT sui risultati del task di regressione. I risultati precedenti avevano evidenziato discrepanze tra i valori di FVC forniti e quelli approssimati. Per esplorare alternative, sono state considerate variazioni significative al GT:

1. Calcolo dell'Intercept:
    - Obiettivo: Abbandonare l'imposizione del punto di inizio e calcolare l'intercetta.
    - Risultati: Verifica dell'impatto sulle prestazioni del modello e sulla coerenza della regressione.
2. Funzione di Secondo Ordine:
    - Obiettivo: Approssimare il GT con una curva di secondo ordine.
    - Risultati: Esplorazione delle variazioni nella previsione della pendenza della FVC.
3. Funzione di Terzo Ordine:
    - Obiettivo: Utilizzare una funzione polinomiale di terzo ordine come GT.
    - Risultati: Analisi delle implicazioni di una curva più complessa nel modello di regressione.

Queste variazioni sono state implementate per valutare l'effetto della complessità della funzione di GT sulla capacità predittiva del modello. L'analisi dei risultati ottenuti da queste modifiche ha fornito una comprensione più approfondita delle dinamiche del task di regressione e ha contribuito a plasmare le decisioni successive nell'approccio metodologico.

4.5.1 Analisi dei Diversi Ground Truth

Sono stati calcolati quattro tipi distinti di GT (Figura 4.6):

1. Ground Truth Lineare con Punto Iniziale Forzato (GT0): Un ground truth lineare in cui il punto iniziale è imposto.
2. Ground Truth Lineare con Interpolazione Calcolata (GT1): Un ground truth lineare in cui il punto iniziale non è forzato, e l'intercetta è calcolata.
3. Ground Truth Quadratico (GT2): Calcolo di un ground truth quadratico.
4. Ground Truth Cubico (GT3): Calcolo di un ground truth cubico.

Per valutare l'accuratezza di ciascun tipo di ground truth, è stato calcolato il MAE per ognuno di essi. La Figura 4.7 presenta un box plot che evidenzia la distribuzione del MAE tra i quattro tipi di ground truth. L'analisi rivela che i ground truth 1, 2 e 3 manifestano errori inferiori rispetto al ground truth 0, confermando una migliore capacità di rappresentare coerentemente l'andamento dell'FVC nel tempo.

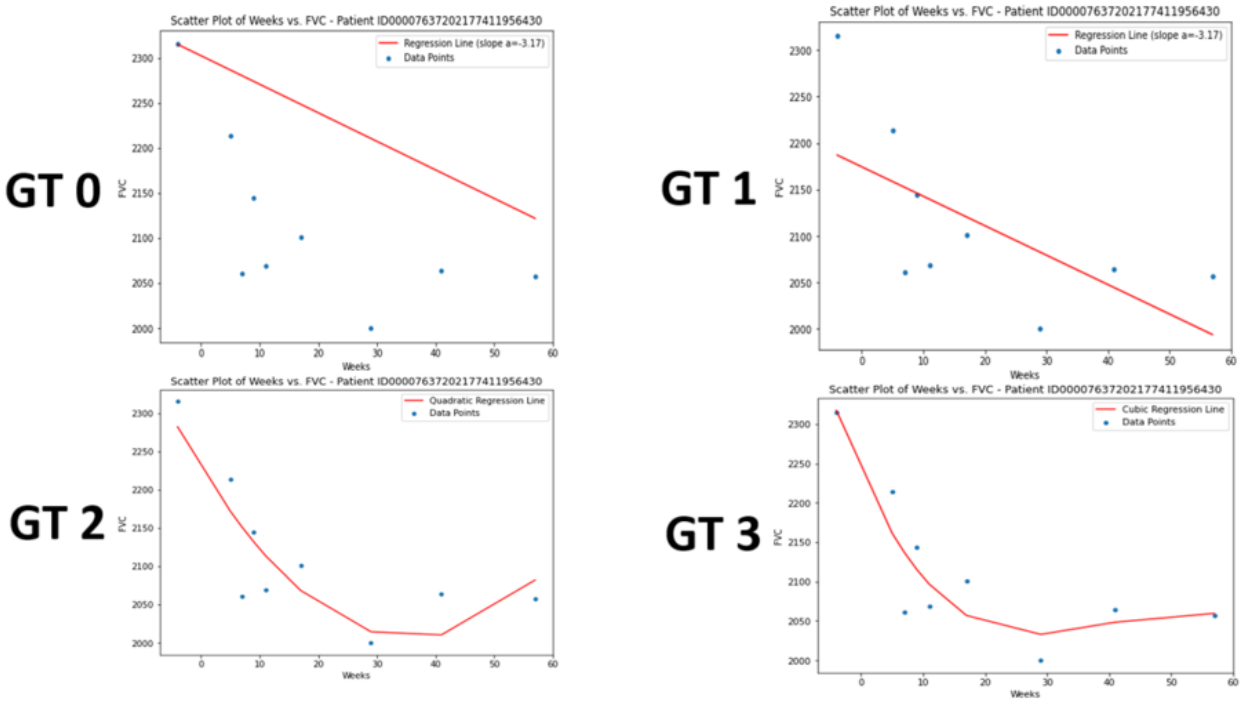

Figura 4.6. confronto tra i vari GT

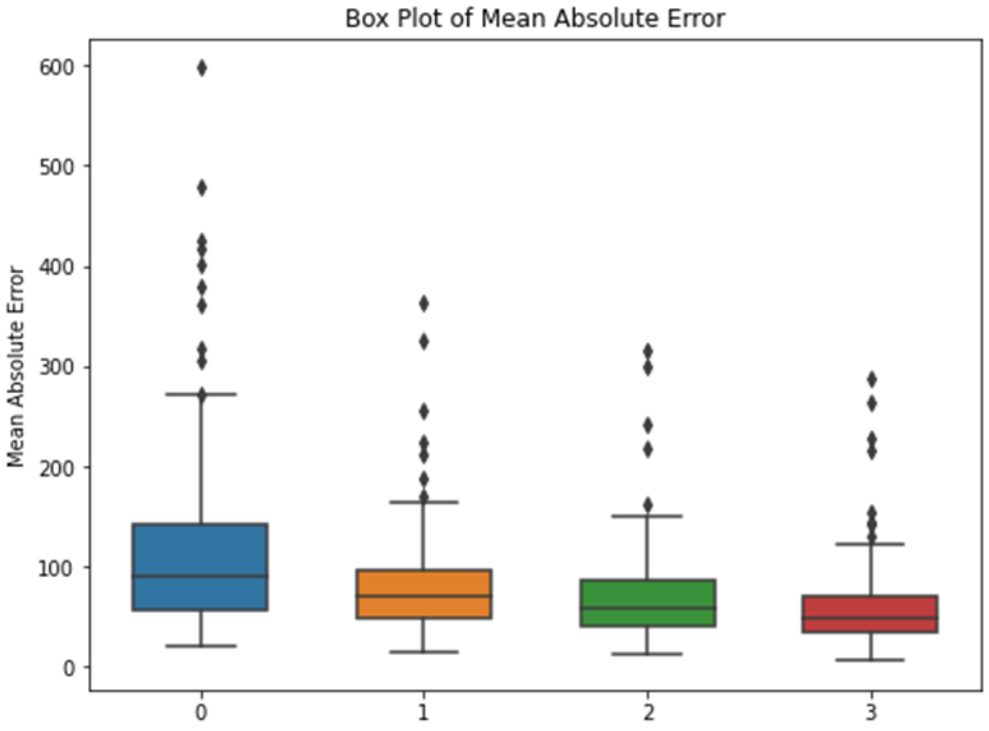

Figura 4.7. Box plot distribuzione MAE per i 4 GT

4.5.2 Training e Test di Ablazione con il Ground Truth 1

Dopo aver verificato che il GT1 avesse una approssimazione migliore dell'andamento dell'FVC rispetto al GT0 si è deciso di effettuare un training con questo nuovo GT.

Model Training

L'implementazione del nuovo ground truth ha richiesto una riconfigurazione del modello migliore precedentemente definito. La modifica principale è stata apportata al livello del fully connected layer (vedi Figura 3.5), originariamente progettato con un singolo neurone in output per la previsione della pendenza dell'FVC. Nella nuova configurazione, il livello Fully-connected è stato adattato per ospitare due neuroni in output: uno dedicato alla stima della pendenza e l'altro per la stima dell'intercetta. Questa modifica è stata essenziale per allineare il modello alle nuove caratteristiche introdotte nel ground truth, garantendo così una previsione più accurata e flessibile della dinamica dell'FVC nel tempo.

Successivamente, il modello è stato sottoposto a un processo di allenamento utilizzando gli stessi iperparametri impiegati per ottenere il miglior modello nel Capitolo 3. In particolare, sono stati utilizzati l'ottimizzatore Adam, la L1 loss come funzione di perdita, un batch size di 16, e uno scheduler lineare. Questo approccio ha consentito di mantenere una coerenza tra gli esperimenti e valutare l'impatto della modifica del ground truth sulla performance del modello. Durante il processo di allenamento del modello adattato al nuovo ground truth, sono stati forniti due tipi di input: l'immagine mediata e le relative feature tabulari. Questa configurazione è stata progettata per garantire che il modello ricevesse informazioni complete e complementari provenienti da entrambe le fonti. L'immagine mediata rappresenta la rappresentazione grafica del 55% inferiore del subset della TC, mentre le feature tabulari contengono informazioni demografiche e caratteristiche specifiche di ciascun paziente

Risultati

Per valutare l'impatto della variazione nel GT sull'allenamento del modello, è stato confrontato il training di questo esperimento con quello del Best Model. Questo confronto mira a identificare eventuali miglioramenti o cambiamenti nelle prestazioni del modello quando si utilizza il nuovo GT, consentendo una valutazione comparativa delle due configurazioni. Analizzando attentamente il processo di training di entrambi i modelli, è possibile ottenere informazioni dettagliate sul contributo del nuovo ground truth alle prestazioni complessive del modello.

Figura 4.8. Confronto Score tra il Training con GT1 e Best Model

I risultati ottenuti dalla Figura 4.8 indicano un deterioramento significativo delle prestazioni del nuovo modello rispetto al vecchio modello. Questo risultato risulta sorprendente, considerando che il Best Model era stato addestrato utilizzando il GT0, il quale approssimava in modo errato la retta dell'andamento dell'FVC (Figura 4.5), come evidenziato in precedenza nell'analisi mediante i box plot (Figura 4.7). La discrepanza tra le aspettative basate sulla qualità del ground truth e le prestazioni osservate solleva interrogativi significativi sulla ragione di tale degrado delle prestazioni e richiede ulteriori indagini per comprendere appieno i fattori coinvolti. Una possibile spiegazione potrebbe essere la mancanza di aumento di complessità del modello interno.

Test di Ablazione

È stato eseguito un confronto tra i risultati ottenuti da tre diversi training: uno utilizzando solo l'immagine mediata come input, un altro utilizzando solo immagini randomiche come input, e infine un terzo utilizzando sia l'immagine mediata che le feature tabulari (ovvero il training precedentemente descritto). I risultati di questi test di ablazione sono stati analizzati per valutare l'impatto del nuovo GT1 e delle diverse modalità di input sulle prestazioni del modello. L'analisi

Figura 4.9. Confronto tra 3 diversi training con GT1

dei risultati mostra che, nonostante l'aumento della complessità del modello con la transizione al nuovo GT1 e l'aggiunta di input casuali come le immagini randomiche, permane un problema di bias nella rete neurale (Figura 4.9). La rete sembra apprendere a restituire i valori del ground truth indipendentemente dall'input fornito.L'osservazione di questo fenomeno solleva dubbi sulla capacità del modello di apprendere in modo significativo dai dati di input forniti, suggerendo che con questa configurazione di input, potrebbe essere problematico ottenere previsioni accurate per l'FVC.

4.5.3 Training e Test di Ablazione con il Ground Truth 2

La verifica dell'impiego del GT2 è stata condotta con l'obiettivo di esaminare se, nonostante le aspettative di possibili miglioramenti, il training con questo nuovo ground truth presenta ancora il problema del bias.

Model Training

L'introduzione del GT2 ha richiesto un ulteriore adeguamento dell'architettura del Best Model. In particolare, la modifica è stata apportata al livello Fully-connected, che originariamente aveva un solo neurone in output per la previsione della pendenza dell'FVC. Nella nuova configurazione, il livello Fully-connected è stato adattato per avere tre neuroni in output: uno dedicato alla stima della pendenza, uno per l'intercetta e un terzo per il termine quadratico del modello. Questa modifica è stata necessaria per adeguare il modello alle variazioni introdotte nel ground truth, consentendo una previsione più precisa e flessibile della dinamica dell'FVC nel tempo. Successivamente, è stato effettuato il training del modello con gli stessi iperparametri precedentemente utilizzati per ottenere il miglior modello nel Capitolo 3, ovvero l'ottimizzatore Adam, la L1 loss, un batch size di 16 e uno scheduler lineare.

Risultati

Dalla Figura 4.10 emerge che, nonostante la complessità aggiunta con l'introduzione del GT2, le prestazioni del modello non migliorano significativamente rispetto all'utilizzo del GT1. Questo fenomeno pone ulteriori dubbi sulla validità delle informazioni contenute nei dati di input per predire in modo accurato la dinamica dell'FVC. La mancanza di miglioramenti sostanziali con l'aumentare della complessità del modello e del ground truth suggerisce che il problema potrebbe derivare da limitazioni intrinseche dei dati forniti dalla challenge, che potrebbero non contenere informazioni sufficienti per una previsione accurata dell'FVC. Data la mancanza di miglioramenti

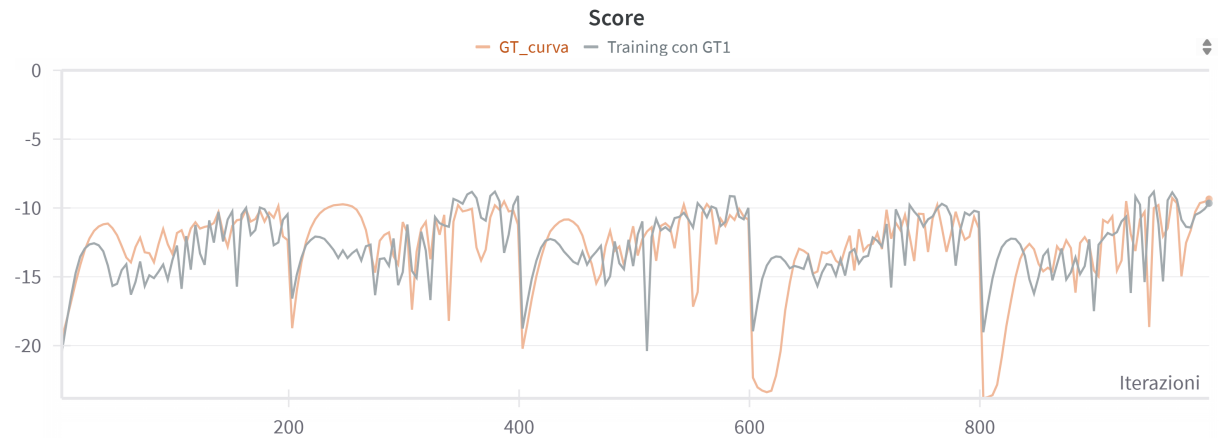

Figura 4.10. Confronto tra training con GT2 e GT1

significativi con l'utilizzo del GT2 rispetto al GT1, la valutazione di ulteriori variazioni del ground truth, come GT3, è stata ritenuta non prioritaria. I risultati ottenuti con GT1 e GT2 suggeriscono che le limitazioni nei dati di input possono essere la causa principale delle sfide incontrate nella previsione dell'FVC.

4.5.4 Test di Correlazione

Si è creata una heatmap per esaminare la correlazione tra tutti i parametri dei diversi GT e le feature tabulari (Figura 4.11). L'analisi della heatmap ha rivelato che non esiste una correlazione evidente o significativa tra i GT e le feature tabulari. Questo ulteriormente suggerisce che le feature tabulari potrebbero non contenere informazioni rilevanti per la previsione dell'andamento dell'FVC. La mancanza di correlazioni significative sottolinea la necessità di esplorare alternative strategie di acquisizione o di integrazione di dati che possano arricchire l'informazione disponibile per il modello.

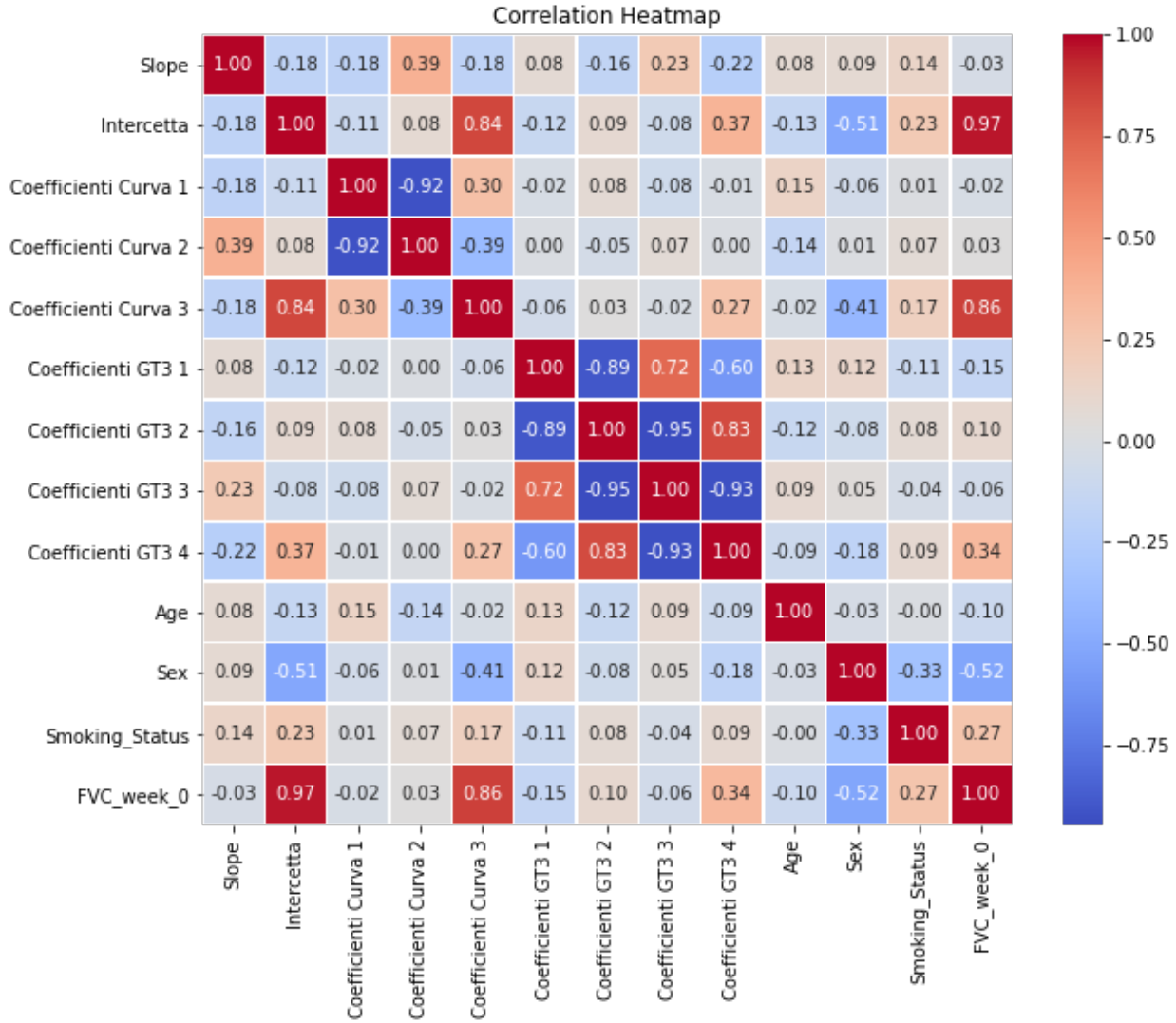

Figura 4.11. Correlation Heatmap

4.5.5 Confronto Ground Truth e Predizione

Nella presente sotto-sezione è stata condotta un'analisi approfondita e un confronto tra i valori del ground truth GT0 e GT1, accostando tali valori alle rispettive previsioni nei test di validazione. L'obiettivo è verificare se persiste un fenomeno di bias anche in questa configurazione.

Rappresentazione grafica dei Ground Truth

Per comprendere meglio le caratteristiche dei ground truth (GT0 e GT1), sono state eseguite rappresentazioni grafiche. Nel caso di GT0, è stato generato un istogramma in cui sull'asse delle ascisse sono riportati i valori di slope e sull'asse delle ordinate è indicato il numero di pazienti associato a ciascun valore di slope. Questo approccio ha permesso di ottenere una visione dettagliata della distribuzione dei valori di slope presenti nel GT0 (Figura 4.12).

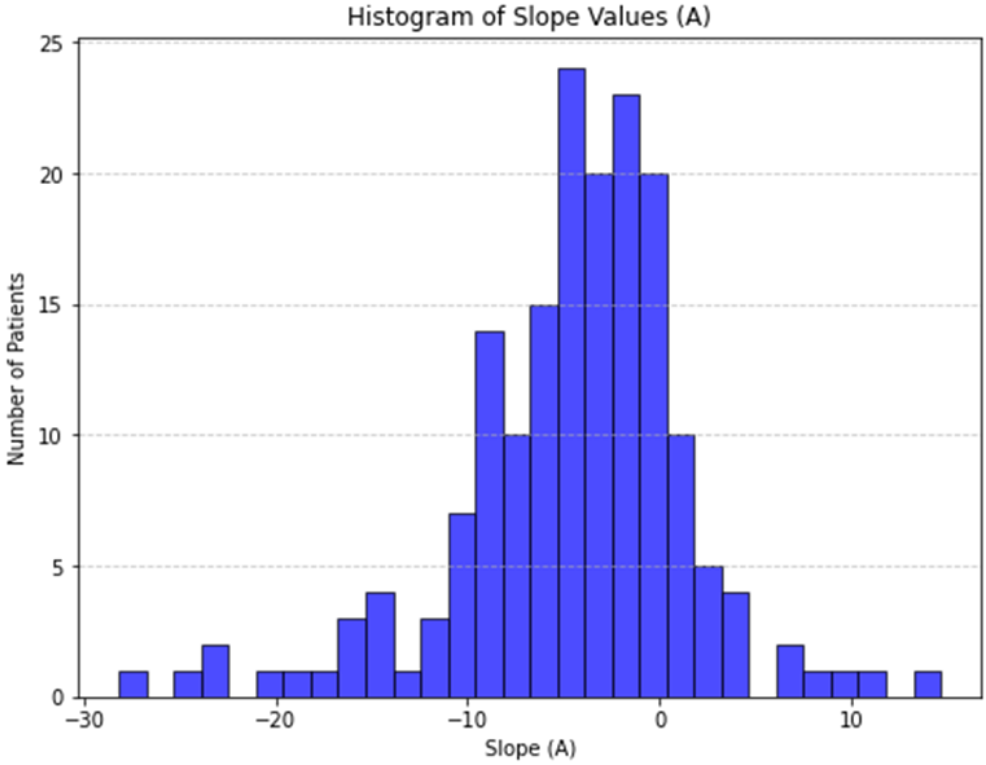

Figura 4.12. Istogramma GT0

Per quanto riguarda il GT1, è stato creato uno scatter plot, dove gli esempi sono disposti su un piano bidimensionale con l'asse delle ascisse che rappresenta i valori di slope e l'asse delle ordinate che indica i valori di intercetta. Questa visualizzazione offre un'analisi visuale della relazione tra slope e intercetta nel GT1, evidenziando eventuali pattern o cluster presenti nei dati (Figura 4.13).

Dalle figure presentate emerge un'omogeneità nei valori di slope nel GT0 e di intercetta nel GT1, con la maggior parte dei pazienti che condividono gli stessi valori. Questo fenomeno suggerisce una possibile limitazione o uniformità nei dati del ground truth, il che potrebbe avere impatti rilevanti sulle prestazioni del modello durante le fasi di addestramento e previsione.

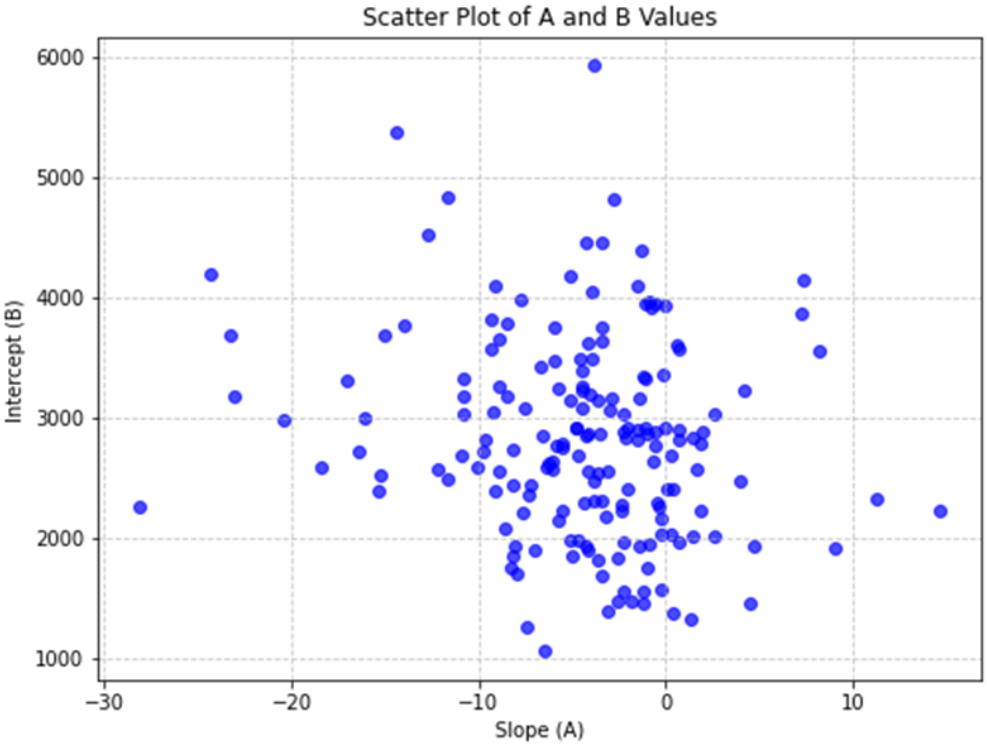

Figura 4.13. Scatterplot valori GT1

Confroto Ground Truth e Predizione

La presenza di eventuali discrepanze o bias tra i GT è stata ulteriormente analizzata mediante una rappresentazione grafica dei valori predetti (all'interno del validation set) rispetto ai valori del GT1. Questo approccio fornisce un'ulteriore prospettiva sulla relazione tra le predizioni del modello e i valori del GT1, permettendo di identificare eventuali tendenze o differenze sistematiche. La Figura 4.14 evidenzia una limitata variazione nei valori di slope predetti dal modello, concentrati principalmente in un range compreso tra -5 e -1.5. Questa osservazione suggerisce una significativa discrepanza tra i valori del GT1 e le previsioni del modello, indicando la possibile presenza di un fenomeno di bias o inefficacia nel processo di apprendimento del modello rispetto ai dati di input. Per approfondire questa interpretazione, è stata eseguita una rappresentazione grafica di tutti i valori predetti per i pazienti utilizzando il GT0.

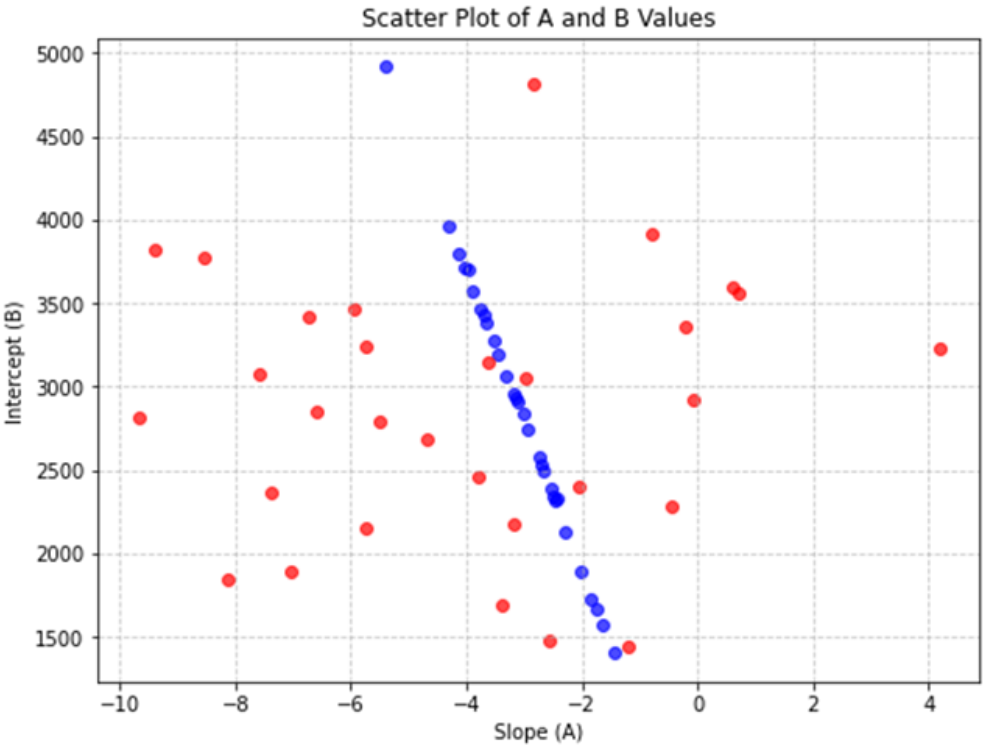

Figura 4.14. Confronto GT1 (in Rosso) e predizione (in blu).

Analisi Previsioni Ground Truth 0

L'analisi delle previsioni basate su GT0 è stata condotta mediante la rappresentazione di istogrammi per ciascuno dei 5 fold nel validation set. Questo approccio mira a fornire una visione completa delle previsioni effettuate dal modello sull'intero dataset di training.

Dall'analisi degli istogrammi presentati nella Figura 4.15, emerge chiaramente che la rete, basandosi sul GT0, genera previsioni di valori di slope ristretti a un intervallo che varia da -4.20 a -3.3. Questa previsione risulta notevolmente discorde rispetto ai valori del GT0 evidenziati nell'istogramma della Figura 4.12. Tale discordanza sottolinea ulteriormente la tendenza della rete a restituire valori concentrati in una gamma specifica, indipendentemente dalla varietà di input. L'osservazione della mancanza di slope positive nei risultati predetti è un elemento di notevole importanza, specialmente nell'ambito medico. Questa carenza potrebbe indicare che il modello, basandosi sul GT0, non riesce ad apprendere in modo efficace le variazioni positive di FVC, il che potrebbe avere implicazioni significative nella valutazione dell'efficacia di una terapia. La capacità di rilevare miglioramenti o cambiamenti positivi nella funzionalità polmonare è cruciale per monitorare l'efficacia dei trattamenti e per adattare le strategie terapeutiche in base alla risposta del paziente. Pertanto, la mancanza di slope positive nelle previsioni del modello solleva interrogativi sulla sua affidabilità nell'affrontare scenari clinici reali. Con i risultati ottenuti è stato dimostrato che i dati di input presentano numerosi bias che non consentono la realizzazione di un modello robusto capace di predirre in modo efficace e corretto la previsione dell'andamento dell'FVC.

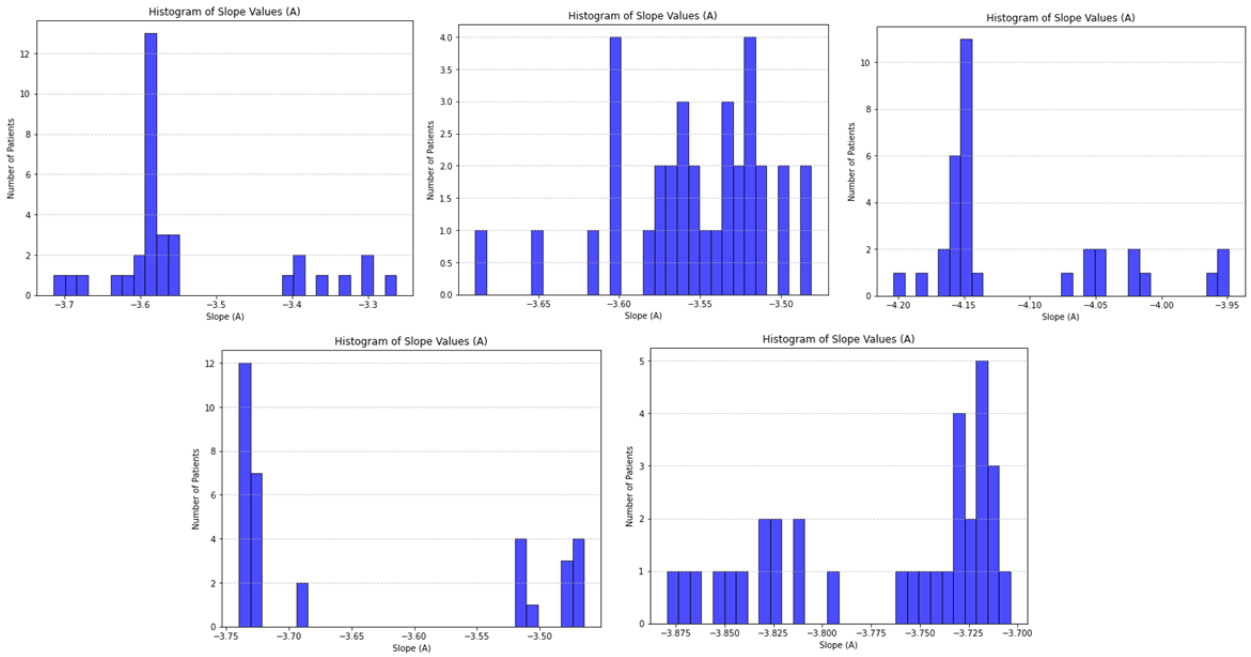

Figura 4.15. Istogrammi delle previsioni dei 5 fold usando GT0

4.6 Esperimento 6: Modifica del Modello

In questa sezione, saranno presentate le modifiche apportate all'architettura del modello, con particolare attenzione all'introduzione di un estrattore di caratteristiche ResNet18 in sostituzione del precedente ViT. L'obiettivo di questa modifica è esplorare come un diverso estrattore di feature, noto per le sue prestazioni in ambito di visione artificiale, possa influenzare le prestazioni generali del modello. Sono stati addestrati due modelli distinti, uno con l'input corretto e l'altro con un input randomico, al fine di valutare se la presenza di bias sia intrinseca al ViT.

4.6.1 Data Preparation

Per entrambi i modelli, simile al Best Model, sono state utilizzate le immagini mediate dal 55% inferiore del subset della TC e le feature tabulari come input. Tuttavia, per il secondo modello, sono state utilizzate solo immagini randomiche per l'estrazione delle feature, senza l'aggiunta di feature tabulari.

4.6.2 Model Training

Per addestrare entrambi i modelli, sono stati utilizzati l'ottimizzatore Adam, la loss L1, un batch size di 16 e uno scheduler lineare.

4.6.3 Risultati

Al fine di investigare se la presenza del bias fosse correlata all'uso del ViT, sono stati generati grafici per visualizzare l'andamento degli score durante i due training. Questi grafici sono stati creati per fornire una comprensione più dettagliata delle dinamiche di apprendimento dei modelli con l'architettura ViT e ResNet18, consentendo una valutazione più approfondita del possibile

impatto del tipo di estrattore di caratteristiche sull'output del modello. Dall'analisi comparativa, come evidenziato nella Figura 4.16, emerge che i due modelli presentano andamenti simili negli score, indicando che il bias osservato non sia direttamente causato dalla scelta dell'estrattore di feature utilizzato. Tuttavia, questa somiglianza suggerisce la presenza di sfide più profonde nell'addestramento del modello, che potrebbe avere difficoltà nell'effettuare previsioni accurate dei valori di FVC indipendentemente dall'approccio di estrazione delle caratteristiche impiegato.

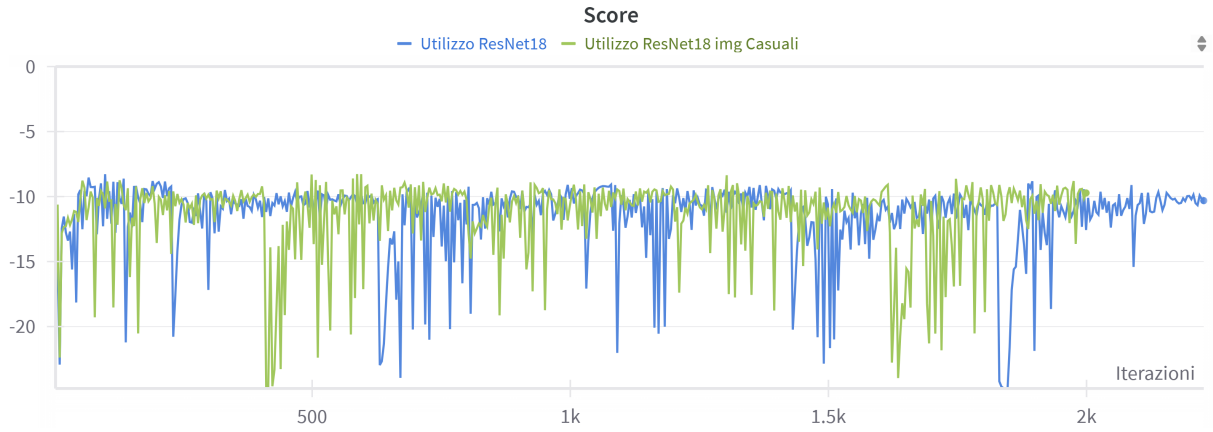

Figura 4.16. Confronto tra i modelli che utilizzano ResNet18.

Analisi Esperto

Per esplorare ulteriormente le potenziali cause del bias osservato nel modello, un esperto aziendale ha condotto ulteriori esperimenti utilizzando un approccio personalizzato. In particolare, è stato implementato un dataloader personalizzato per l'addestramento di una HybridResNet50, che ha integrato dati tabulari e immagini. Nel corso degli esperimenti, sono state eseguite le seguenti prove:

1. Test di overfitting su due casi: L'addestramento del modello su un sottoinsieme dei dati è stato eseguito con successo, indicando che il modello è in grado di apprendere in modo adeguato da un numero limitato di campioni.

2. Addestramento su tutto il dataset senza downsample (task di regressione): Il modello non è riuscito ad apprendere in modo efficace dai dati, suggerendo che l'addestramento su tutto il dataset non ha prodotto previsioni accurate per il task di regressione.

3. Addestramento su dataset con downsample (task di regressione): Ancora una volta, il modello non è riuscito ad apprendere in modo efficace, suggerendo che l'applicazione di downsample non ha portato a miglioramenti significativi nelle prestazioni.

4. Addestramento su dataset con downsample (task di classificazione, slope positivo/negativo): Anche in questo caso, il modello ha mostrato difficoltà nel processo di apprendimento, indicando che la classificazione basata sulla polarità del slope non ha portato a un miglioramento delle prestazioni.

L'insuccesso del modello nell'apprendimento durante queste prove potrebbe indicare sfide più profonde o complessità nei dati, che rendono difficile per il modello generalizzare e produrre previsioni accurate sia nel caso di regressione che di classificazione. Questi risultati contribuiscono alla comprensione delle difficoltà intrinseche nell'affrontare il task specifico di previsione dell'FVC su questo dataset.

4.7 Esperimento 7: Utilizzo delle Radial Basis Functions

Dopo aver identificato la presenza di bias nei modelli precedenti e aver esplorato diverse architetture per affrontare questa problematica, si è condotto un nuovo esperimento utilizzando una rappresentazione basata su RBF. L'obiettivo principale era valutare se questa metodologia avrebbe potuto migliorare la flessibilità del modello e mitigare il bias osservato nelle predizioni dell'FVC.

4.7.1 Metodologia

L'implementazione dell'approccio RBF ha coinvolto l'utilizzo di funzioni radiali come strato di trasformazione. Questa nuova architettura è stata progettata per consentire al modello di apprendere rappresentazioni più complesse dei dati, con l'intento di migliorare la capacità predittiva dell'andamento dell'FVC. Durante il training, è stato utilizzato un dataloader personalizzato, con input costituiti da immagini mediate e feature tabulari. Inoltre, è stato effettuato un secondo training utilizzando solo feature estratte da immagini randomiche.

4.7.2 Model Training

Per ciascuno dei modelli RBF allenati, sono stati utilizzati i seguenti parametri:

- Sigma: 0.850
- Numero di Centri: 132

Questi valori sono stati identificati come ottimali in precedenza nella sezione 3.11 del Capitolo 3.

4.7.3 Risultati e Analisi

Al fine di valutare la presenza di bias nei modelli proposti, sono state effettuate submission su Kaggle. L'utilizzo del valore di score ottenuto sulla piattaforma come discriminante ha permesso di confrontare le performance dei modelli in un contesto esterno e oggettivo.

Kaggle Submission	Private Score	Public Score
Succeeded	-8.4809	-8.6692

Tabella 4.2. Submission RBF e input corretti

Kaggle Submission	Private Score	Public Score
Succeeded	-6.9056	-6.9833

Tabella 4.3. Submission RBF e input casuali

L'analisi dei risultati delle due submission (Tabella 4.2 e Tabella 4.3) suggerisce che il modello allenato con input casuali ha ottenuto valori di score decisamente migliori rispetto al modello allenato con gli input corretti. Questa discrepanza nei risultati solleva interrogativi sulla presenza di eventuali bias o inefficienze nel modello addestrato con gli input corretti, sottolineando l'importanza di esaminare attentamente le prestazioni dei modelli in contesti reali e confrontarle con le aspettative generate durante il training. Tale analisi contribuirà a fornire indicazioni cruciali sulla robustezza delle metodologie proposte e sulla loro capacità di generalizzazione su dati di test non precedentemente osservati.

Capitolo 5

Metodi e Risultati: Task di classificazione

In questo capitolo, sono presentati gli esperimenti condotti modificando il task originale di regressione, visti gli esiti non soddisfacenti presentati nei capitoli precedenti. Dopo aver constatato l'impossibilità di creare un modello in grado di prevedere la capacità polmonare basandosi esclusivamente sui dati disponibili, si è optato per semplificare l'approccio. Tale esperimento ha puntato all'ottenimento di un modello di classificazione che, rinunciando all'obiettivo più ambizioso di predire con precisione il valore di FVC, fosse limitato alla stima dell'andamento della capacità polmonare (positiva o negativa). Per tale scopo sono state impiegate le immagini TC, a partire dalle quali sono state estratte le feature tramite modello ViT, e le feature tabulari presenti all'interno del dataset fornito dalla competizione.

5.1 Esperimento 1:Modifica del Ground Truth e Classificazione

Visti gli obiettivi di questa sezione, si è dapprima proceduto con la modifica del GT.

5.1.1 Modifica Ground Truth

Al fine di ottenere la modifica desiderata del GT, rendendo il set di dati adatto alla task in questione, si è proceduto come segue:

- Calcolo del GT mediante SVD
- Assegnazione di Classi: Dopo il calcolo della slope, si è proceduto all'assegnazione di classi distintive. Se la slope risultava positiva, veniva associato il valore di classe 0; in caso di slope negativa, si assegnava il valore di classe 1.

5.1.2 Training e Risultati

In seguito alla definizione del nuovo GT, è stato avviato il processo di training del modello. La fase di training è stata eseguita sull'80% del set di addestramento, mentre il 20% rimanente è stato dedicato alla validazione. Al completamento del training, il modello è stato testato utilizzando il Validation set, e l'analisi della classificazione è stata valutata attraverso una matrice di confusione (CM, dall'inglese *confusion matrix*), la cui rappresentazione grafica è mostrata nella Figura 5.1. Inoltre, sono state calcolate le metriche di Accuratezza, Precisione, Recall e F1 Score, come riportato nella

Tabella 5.1. Come evidenziato dalla CM, si osserva che la rete tende a riconoscere esclusivamente la classe 1, senza mai identificare correttamente soggetti nella classe 0. Tale andamento suggerisce una possibile difficoltà del modello nel cogliere in maniera efficace le caratteristiche distintive associate a questa specifica classe.

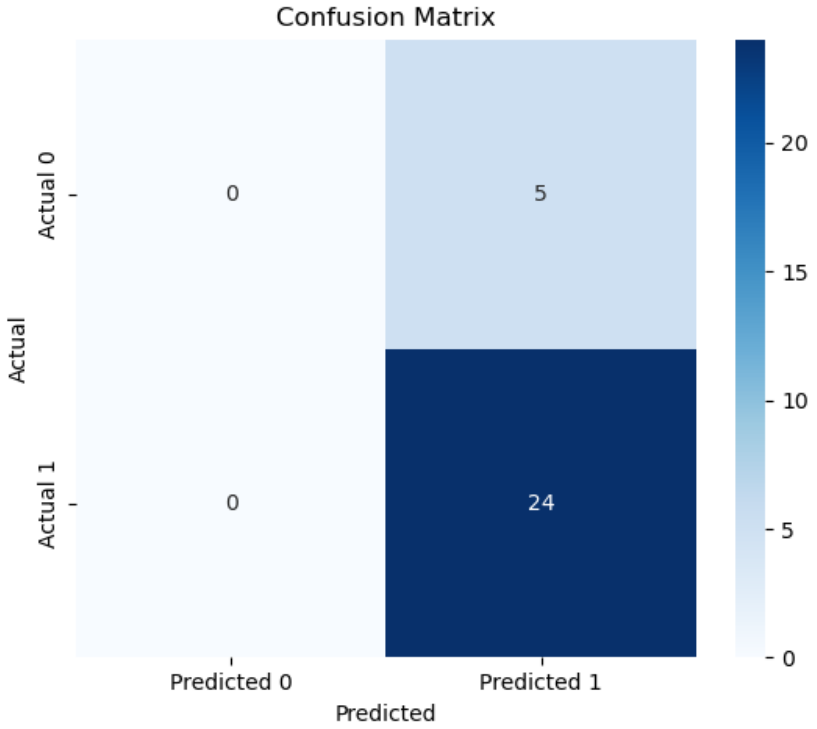

Figura 5.1. CM esperimento 1

Accuracy	0.827
Precision	0.827
Recall	1.0
F1 Score	0.9056603773584906

Tabella 5.1. Metriche esperimento 1

Data la limitata generalizzazione osservata nei risultati del modello, è stata eseguita un'analisi approfondita del dataset per valutare la distribuzione dei soggetti nelle diverse classi. Dall'analisi emerge uno sbilanciamento significativo, con soli 26 casi appartenenti alla classe 0 su un totale di 176 casi. Questo sbilanciamento giustifica il motivo per cui la rete predice esclusivamente soggetti appartenenti alla classe 1, come evidenziato nella CM (Figura 5.1).

5.2 Esperimento 2: Leave One Out

L'attenzione di questo esperimento è stata focalizzata sulla mitigazione dello sbilanciamento presente nel dataset che, vista la limitita numerosità del dataset impiegato, si è preferito non risolvere tramite semplice downsampling. Per affrontare questa problematica, è stata implementata la strategia del Leave One Out [29]. Tale approccio prevede la creazione di sottoinsiemi di addestramento in cui viene escluso un singolo elemento del dataset alla volta, consentendo al modello di apprendere dai dati rimanenti (Figura 5.2). Questa operazione viene ripetuta iterativamente per ogni campione nel dataset, garantendo che ogni istanza venga omessa almeno una volta durante il processo di addestramento. L'implementazione del Leave One Out mira a favorire una maggiore robustezza del modello, mitigando gli effetti dello sbilanciamento delle classi e migliorando la capacità predittiva complessiva.

Figura 5.2. Esempio suddivisione Dataset con il medoto Leave One Out[29]

5.2.1 Training e Risultati

Dopo aver completato l'addestramento delle 176 reti utilizzando il metodo Leave One Out, è stata condotta una fase di valutazione aggiuntiva per verificare l'efficacia di questo approccio nella

risoluzione della problematica dello sbilanciamento nel dataset. In questa fase, ciascun modello addestrato genera una previsione per l'unico elemento escluso durante il training. Questa procedura di validazione è stata ripetuta per tutti i 176 modelli, e i risultati sono stati analizzati tramite una CM (Figura 5.3) per valutare le prestazioni complessive del metodo in termini di classificazione delle diverse categorie. Inoltre, come nel precedente esperimento, sono state calcolate le metriche riportate nella Tabella 5.2 per valutare le performance del modello dopo l'addestramento mediante il metodo Leave One Out. Dai risultati presentati nella CM e nella Tabella 5.2, emerge che no-

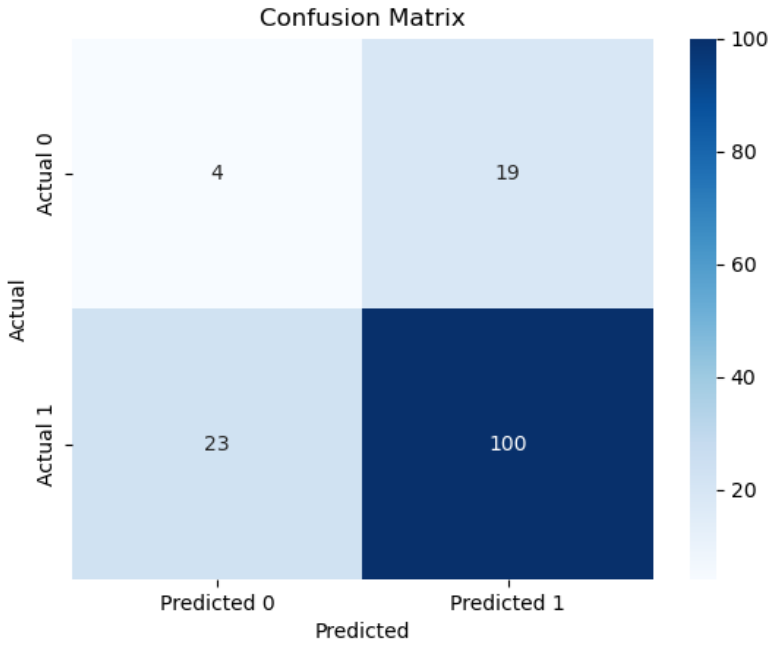

Figura 5.3. CM metodo Leave One Out

Accuracy	0.712
Precision	0.840
Recall	0.813
F1 Score	0.8264462809917354

Tabella 5.2. Metriche esperimento 2

nostante l'impiego del metodo Leave One Out, i modelli sembrano avere difficoltà nell'apprendere correttamente le classi, con una marcata predilezione per la classe 1.

5.3 Esperimento 3: Leave One Out e Up-Weighting

Nel contesto dell'Esperimento 3, è stato implementato un approccio combinato utilizzando il metodo Leave One Out insieme alla tecnica di Up-Weighting. Quest'ultima consiste nell'assegnare pesi differenziati alle classi durante il processo di addestramento, attribuendo maggior peso alle istanze della classe sottorappresentata (classe 0) e minor peso a quelle sovrarappresentate (classe 1). Questa strategia mira a compensare lo sbilanciamento delle classi, fornendo al modello un incentivo aggiuntivo per apprendere efficacemente dai casi meno frequenti.

5.3.1 Training e Risultati

Nel corso del training della rete, è stato adottato il metodo Leave One Out insieme all'Up-Weighting per gestire la disparità di campioni tra le classi. Tale approccio combinato è stato progettato con l'obiettivo di migliorare le prestazioni del modello, focalizzandosi principalmente sulla corretta classificazione della classe 0. Durante il training, l'intero dataset è stato utilizzato come input per il modello. Una volta concluso il processo di training della rete è stata eseguita la fase di testing del modello. Questa fase ha coinvolto la valutazione delle prestazioni mediante la generazione di una CM (Figura 5.4) e il calcolo delle metriche di riferimento, le quali sono dettagliatamente riportate nella tabella 5.3.

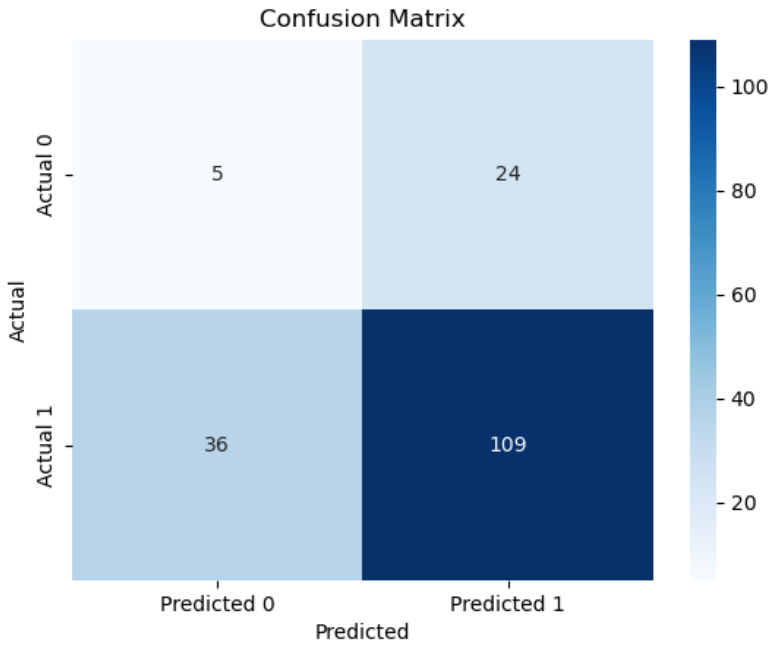

Figura 5.4. CM esperimento 3

Nonostante l'adozione combinata del metodo Leave One Out e dell'Up-Weighting per mitigare la disparità di campioni tra le classi, non emergono segnali di miglioramento significativo in termini di performance di classificazione del modello.

Accuracy	0.655
Precision	0.819
Recall	0.751
F1 Score	0.784

Tabella 5.3. Metriche esperimento 3

5.3.2 Training e Risultati con Downsampling

Poiché la combinazione delle due strategie non ha portato a miglioramenti significativi nel modello, si è infine optato per un approccio di training con un dataset bilanciato, utilizzando la strategia di downsampling. Durante la fase di addestramento, il downsampling è stato implementato per affrontare la disparità di campioni tra le classi, riducendo il numero di campioni della classe sovrarappresentata (classe 1) al fine di ottenere un equilibrio con la classe sottorappresentata (classe 0). I risultati di questa modifica sono dettagliati di seguito tramite CM (Figura 5.5) e tabella 5.4 con le metriche relative.

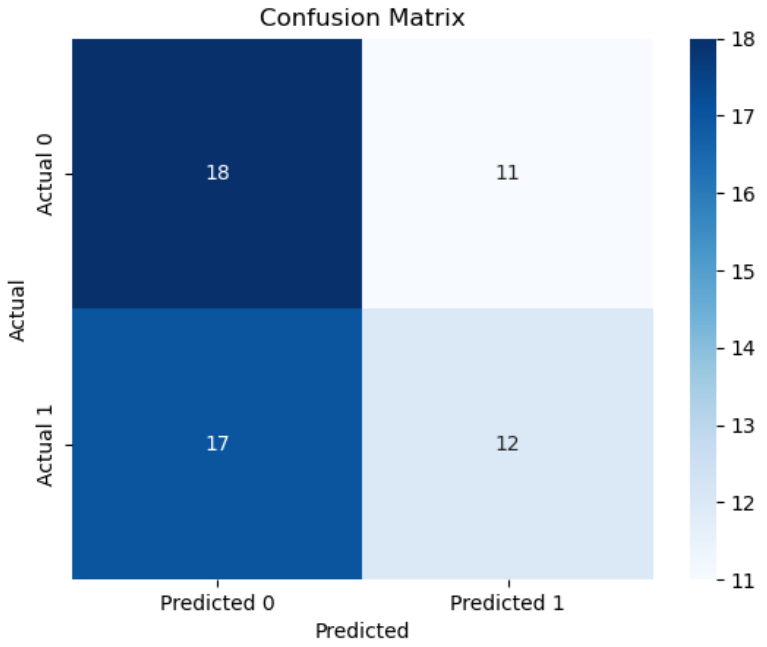

Figura 5.5. CM esperimento 3 con downsampling

Accuracy	0.517
Precision	0.5217
Recall	0.413
F1 Score	0.4615

Tabella 5.4. Metriche esperimento 3 con downsampling

Nonostante l'implementazione del downsampling per bilanciare il dataset, emerge una limitata capacità del modello nel compiere corrette classificazioni. Ciò è evidenziato da valori relativamente bassi sia in termini di Accuracy che di Recall. Questi risultati suggeriscono che, con i dati attualmente disponibili, il modello potrebbe avere difficoltà nel individuare informazioni rilevanti per effettuare una corretta classificazione.

Capitolo 6

Conclusione

Nela prima parte di questa tesi, sono stati dettagliati e analizzati gli esperimenti condotti per sviluppare un modello predittivo avanzato per l'andamento della IPF. L'approccio adottato ha coinvolto l'impiego di ViT come estrattore di feature, sottolineando la rilevanza della progettazione e dell'ottimizzazione del modello per ottenere risultati competitivi.

Uno dei risultati più significativi è emerso dall'esperimento descritto nella sezione 3.4, in cui si è constatato che la feature tabulare relativa al Volume Polmonare fosse inefficace. Nonostante la sua inaccurata segmentazione, il miglior modello trovato in letteratura (sezione 3.2) lo utilizza come dato di input, sollevando interrogativi data la evidente discrepanza tra la qualità del dato di input e l'eccellenza del risultato ottenuto dal modello in termini di LLLm.

L'esperimento, descritto nella sezione 3.5 , ha introdotto l'utilizzo dell'immagine mediata e dimostrato l' impatto positivo di questo preprocessing sullo score del modello. Questa metodologia ha dimostrato l'efficacia della slice mediata nel fornire informazioni globali sul volume polmonare, mantenendo nel contempo un basso costo computazionale. In particolare, la media del volume nella zona di maggiore interesse ha generato un risultato migliore (esperimento trattato nella sezione 3.8).

Ulteriori esperimenti, come l'utilizzo di solo il 55% del volume delle TC, di uno scheduler lineare (esperimento delineato nella sezione 3.7), la normalizzazione delle feature tabulari tramite il metodo min-max (esperimento esposto nella sezione 3.10), e l'adozione dei ViT, hanno contribuito all'ottenimento di risultati notevoli, benchè inferiori rispetto ai migliori modelli disponibili.

Nonostante l'obiettivo iniziale fosse orientato verso l'ottenimento del miglior modello mediante l'utilizzo dei ViT in congiunzione con il dataset disponibile, gli esperimenti trattati nelle sezioni 3.9 e 3.11 hanno sollevato interrogativi critici riguardo alla praticabilità del compito considerando il dataset in questione. Tale riflessione è emersa soprattutto nell'esperimento presente nella sezione 3.9, dove l'impiego di immagini segmentate non solo non ha portato a miglioramenti nelle prestazioni del modello, bensì ne ha determinato un peggioramento. Questo risultato è notevole, poiché l'utilizzo di immagini contenenti solo i polmoni dovrebbe, teoricamente, fornire informazioni più rilevanti rispetto alle immagini non segmentate delle TC. L'apparente discrepanza tra le aspettative di miglioramento e il degrado delle prestazioni ha sottolineato la possibile presenza di limitazioni intrinseche nel dataset o nella metodologia, richiedendo ulteriori approfondimenti per una comprensione completa delle cause.

Le prime considerazioni hanno delineato la necessità di un'analisi più approfondita del dataset fornito dalla challenge. Non avendo riscontri nella letteratura che avvalorassero l'ipotesi di inadeguatezza del dataset per il task in esame, si è proceduto con l'esecuzione di svariati esperimenti, come dettagliato nel Capitolo 4, mirati a valutare la praticabilità del task a partire dai dati disponibili. Gli esperimenti condotti hanno messo in evidenza che le feature tabulari non mostravano

alcuna correlazione significativa con l'output assegnato a ciascuna istanza. Questa osservazione ha portato all'esecuzione di test di ablazione, al fine di valutare come il modello si comportasse in assenza di tali feature. Dall'addestramento è emerso che, con o senza le feature tabulari, il modello manifestava lo stesso andamento in termini di score. Per ulteriori indagini su questo aspetto, è stato condotto un addestramento utilizzando solo immagini casuali, ma i risultati hanno sorprendentemente mostrato che il modello manteneva invariato il proprio andamento.

La variazione del GT è stata l'unica modifica che ha influenzato l'andamento dello score. Ciò ha indicato che i modelli attuali apprendessero un modello generico basato su dati di decadimento medio per la popolazione, il che spiegava la loro coerenza tra differenti input. Questa evidenza è stata confermata quando è stato modificato il GT utilizzato dai migliori modelli della competizione, il quale non approssimava correttamente i valori di FVC. In questi casi, i modelli hanno mostrato prestazioni inferiori con i nuovi GT, nonostante l'approssimazione corretta. La conferma di tale evidenza è emersa in modo coerente da tutti gli esperimenti di ablazione condotti. Inoltre, data la natura inesplorata di questo scenario, si è consultato anche un esperto per analizzare il problema. Le conclusioni dell'esperto sono risultate concordi con quelle emerse da questa ricerca, sottolineando che i modelli non riescono ad apprendere in modo corretto l'output atteso a partire dal dataset a disposizione. Nella fase conclusiva del lavoro, si è tentato di trasformare il task in uno di classificazione. Tuttavia, anche in questo contesto, i risultati ottenuti nel Capitolo 5 evidenziano che a causa di un forte sbilanciamento del dataset, i vari modelli non riescono ad apprendere in modo accurato quanto desiderao e restituivano valori casuali. Anche in questo caso, al fine di ottenere ulteriori conferme, è stato consultato un esperto il quale, attraverso propri esperimenti, ha ratificato questa conclusione.

In sintesi, il presente lavoro descrive una serie di esperimenti che convergono verso la conclusione che il dataset fornito dalla challenge non risulta idoneo per il task di regressione richiesto. Questa valutazione è di cruciale importanza, considerando che modelli finalizzati a riconoscere la progressione della IPF potrebbero, attraverso una previsione erronea, influire sulla capacità di un medico nel riconoscere la gravità della condizione di un paziente o nel determinare l'efficacia di un determinato trattamento. Per gli sviluppi futuri, risulta imperativo acquisire un dataset più adatto al task di regressione, tenendo conto delle specifiche esigenze e delle caratteristiche cruciali della IPF. Un dataset più rappresentativo e bilanciato potrebbe migliorare notevolmente la capacità dei modelli di apprendere pattern più accurati e generalizzabili.

Inoltre, è consigliabile esplorare l'utilizzo di metriche più facilmente interpretabili e che rispecchino in modo più fedele la rilevanza clinica del task. Metriche che tengano conto delle sottigliezze mediche specifiche della progressione della IPF potrebbero fornire una valutazione più accurata e informativa, evitando fraintendimenti anche da parte di esperti del settore.

Questi sviluppi futuri dovrebbero essere orientati a garantire una maggiore affidabilità e interpretabilità nelle previsioni dei modelli, contribuendo così a una migliore assistenza medica e decisionale nella gestione della Fibrosi Polmonare Idiopatica.

Bibliografia

[1] Zabir Al Nazi et al. "Fibro-CoSANet: pulmonary fibrosis prognosis prediction using a convolutional self attention network". In: *Physics in Medicine amp; Biology* 66.22 (nov. 2021), p. 225013. ISSN: 1361-6560. DOI: 10.1088/1361-6560/ac36a2. URL: http://dx.doi.org/10.1088/1361-6560/ac36a2.

[2] AppMaster. "Utilizzo di Docker per l'architettura a microservizi". In: (2023). URL: https://appmaster.io/it/blog/architettura-a-microservizi-in-docker.

[3] Artkulak. *OSIC Pulmonary Fibrosis Progression 1st place solution*. https://github.com/artkulak/osic-pulmonary-fibrosis-progression. 2020.

[4] Abhishek Bhat. *Tissue Segmentation used in 4th place solution*. https://www.kaggle.com/code/abhishekgbhat/tissue-segmentation-used-in-4th-place-solution/notebook. 2021.

[5] Lukas Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020. URL: https://www.wandb.com/.

[6] Ahmed Shahin Carmela Wegworth David Elizabeth Estes Julia Elliott Justin Zita Simon-Walsh Slepetys Will Cukierski. *OSIC Pulmonary Fibrosis Progression*. 2020. URL: https://kaggle.com/competitions/osic-pulmonary-fibrosis-progression.

[7] Yining Deng e B.S. Manjunath. "Unsupervised segmentation of color-texture regions in images and video". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.8 (gen. 2001), pp. 800–810. DOI: 10.1109/34.946985. URL: https://doi.org/10.1109/34.946985.

[8] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[9] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV].

[10] *Fibrosi polmonare idiopatica - Roche*. URL: https://www.roche.it/cosa-facciamo/le-nostre-aree/malattie-rare/fibrosi-polmonare-idiopatica.

[11] European Lung Foundation. *IPF – Fibrosi polmonare idiopatica - European Lung Foundation*. Apr. 2023. URL: https://europeanlung.org/it/information-hub/lung-conditions/ipf-fibrosi-polmonare-idiopatica/.

[12] Furcifer. *Q-Regression with CT Tabular Features (PyTorch)*. Accessed: 2024-01-29. 2020. URL: https://www.kaggle.com/code/furcifer/q-regression-with-ct-tabular-features-pytorch/notebook.

[13] Everton Gomede. *Radial Basis Functions Neural Networks: Unlocking the Power of Non-linearity*. https://medium.com/@evertongomede/radial-basis-functions-neural-networks-unlocking-the-power-of-nonlinearity-c67f6240a5bb. Accessed: 2022-08-24. 2021.

[14] Yogendra Kumar Jain e Santosh Kumar Bhandare. "Min Max Normalization based Data perturbation Method for privacy protection". In: *International journal of computer and communication technology* (ott. 2013), pp. 233–238. DOI: 10.47893/ijcct.2013.1201. URL: https://doi.org/10.47893/ijcct.2013.1201.

[15] Ping Jiang e Jie-Jie Chen. "Displacement prediction of landslide based on generalized regression neural networks with K-fold cross-validation". In: *Neurocomputing* 198 (lug. 2016), pp. 40–47. DOI: 10.1016/j.neucom.2015.08.118. URL: https://doi.org/10.1016/j.neucom.2015.08.118.

[16] Diederik P. Kingma e Jimmy Lei Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1412.6980* (2014). URL: https://arxiv.org/abs/1412.6980.

[17] MUO. *A Beginner's Guide to Kaggle for Data Science*. Retrieved June 10, 2023. Apr. 2023. URL: https://www.makeuseof.com/beginners-guide-to-kaggle/.

[18] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. https://pytorch.org. 2019.

[19] Tal Ridnik et al. *ImageNet-21K Pretraining for the Masses*. 2021. arXiv: 2104.10972 [cs.CV].

[20] Paul Rodríguez et al. "Beyond one-hot encoding: Lower dimensional target embedding". In: *Image and Vision Computing* 75 (lug. 2018), pp. 21–31. DOI: 10.1016/j.imavis.2018.04.004. URL: https://doi.org/10.1016/j.imavis.2018.04.004.

[21] Khairul Anuar Mat Said e Asral Bahari Jambek. "Analysis of image processing using morphological erosion and dilation". In: *Journal of Physics: Conference Series* 2071.1 (ott. 2021), p. 012033. DOI: 10.1088/1742-6596/2071/1/012033. URL: https://doi.org/10.1088/1742-6596/2071/1/012033.

[22] *Spirometria*. URL: https://www.my-personaltrainer.it/spirometria.html.

[23] *TAC - Tomografia assiale computerizzata*. URL: https://www.my-personaltrainer.it/salute/tac.html.

[24] Jahangir Tahmasi, Abbas Ahmadi e Behzad Mosalla Nezhad. "Uncovering the hidden relation between air pollution and ischemic heart disease: Experience from Tehran". In: (giu. 2021).

[25] Stefan Van der Walt et al. *scikit-image: Image processing in Python*. https://scikit-image.org/docs/stable/api/skimage.morphology.html. 2014.

[26] Alexander Wong et al. *Fibrosis-Net: A Tailored Deep Convolutional Neural Network Design for Prediction of Pulmonary Fibrosis Progression from Chest CT Images*. 2021. arXiv: 2103.04008 [cs.CV].

[27] Bichen Wu et al. *Visual Transformers: Token-based Image Representation and Processing for Computer Vision*. 2020. arXiv: 2006.03677 [cs.CV].

[28] Anju Yadav et al. "FVC-NET: An Automated Diagnosis of Pulmonary Fibrosis Progression Prediction Using Honeycombing and Deep Learning". In: *Computational Intelligence and Neuroscience* 2022 (2022). URL: https://api.semanticscholar.org/CorpusID:246411081.

[29] Zach. *A Quick Intro to Leave-One-Out Cross-Validation (LOOCV)*. 1. Accessed: 2024-02-27. 2020.