

Report Progetto Data Mining

Matteo Mariani 748241

1 Obiettivi del progetto

Il progetto trattato si pone come obiettivo principale l'applicazione di reti neurali per la classificazione automatica delle categorie di battiti cardiaci, utilizzando i segnali ECG forniti dal database PhysioNet MIT-BIH Arrhythmia.

Per affrontare questo problema sono state adottate e sperimentate diverse architetture di reti neurali profonde, tra cui ResNet-18, ResNet-34, ResNet-50 e Vision Transformers (ViTs), al fine di valutare l'efficacia e le prestazioni di ciascun modello nella classificazione dei battiti.

2 Il database PhysioNet MIT-BIH Arrhythmia nella letteratura

Il database MIT-BIH Arrhythmia, reso disponibile attraverso la piattaforma PhysioNet, rappresenta uno dei dataset più utilizzati e studiati nella comunità scientifica per lo sviluppo e la validazione di algoritmi di classificazione e rilevamento delle aritmie cardiache. Creato originariamente dal Laboratorio di Ingegneria Biomedica del Massachusetts Institute of Technology in collaborazione con il Beth Israel Hospital di Boston, questo dataset è stato pubblicato nel 1980 con l'obiettivo di fornire una base di riferimento standardizzata per la ricerca nel campo dell'analisi dei segnali elettrocardiografici (ECG).

Nel corso degli anni, il MIT-BIH Arrhythmia Database è diventato un punto di riferimento imprescindibile per le ricerche mirate al riconoscimento automatico di anomalie del ritmo cardiaco. Esso contiene registrazioni ECG a due canali provenienti da 48 soggetti, con una durata di 30 minuti ciascuna, annotati beat-by-beat da esperti cardiologi. I dati sono campionati a 360 Hz e comprendono una varietà di aritmie, rendendo il database particolarmente utile per addestrare e testare modelli di apprendimento automatico e reti neurali profonde.

Nella letteratura scientifica, numerosi studi hanno utilizzato questo dataset per sviluppare sistemi di classificazione automatica dei battiti cardiaci. Le prime ricerche si sono concentrate sull'estrazione manuale di feature, come le durate degli intervalli PR, QRS e QT o le caratteristiche spettrali dei segnali, successivamente classificate mediante algoritmi tradizionali come Support Vector Machines (SVM), k-Nearest Neighbors (k-NN) e alberi decisionali [2, 1, 3, 4]. Tali approcci, sebbene inizialmente efficaci, hanno mostrato limiti in termini di generalizzazione e capacità di gestire la complessità intrinseca dei segnali ECG.

Con l'avvento del deep learning, l'utilizzo di architetture basate su Convolutional Neural Networks (CNN) e Recurrent Neural Networks (RNN) ha permesso un significativo miglioramento delle prestazioni nella classificazione dei battiti cardiaci. Le CNN, in particolare, si sono dimostrate efficaci nell'apprendere automaticamente rappresentazioni significative dai segnali grezzi senza necessità di estrazione manuale delle feature. Alcuni lavori, come quelli di Kiranyaz et al.[6] e Acharya et al. [5], hanno introdotto modelli basati su CNN 1D e 2D applicati direttamente ai segnali ECG, ottenendo elevati livelli di accuratezza.

Più recentemente, sono stati sviluppati approcci che sfruttano le RNN, in particolare architetture basate su LSTM (Long Short-Term Memory), capaci di modellare efficacemente le dipendenze temporali presenti nei segnali ECG. Ad esempio, Mousavi et al. [7, 9] hanno proposto un approccio di deep learning sequence-to-sequence per la classificazione inter- e intra-paziente dei battiti cardiaci, ottenendo prestazioni competitive grazie alla capacità di gestire variazioni temporali e paziente-specifiche nei dati.

Un ulteriore passo avanti è stato compiuto con l'introduzione di tecniche di trasferimento dell'apprendimento (transfer learning) per la classificazione dei battiti cardiaci. Kachuee et al. [8] hanno realizzato un modello che utilizza una rappresentazione profonda trasferibile, addestrata su ampi dataset e successivamente adattata per la classificazione specifica del battito cardiaco, migliorando così la generalizzazione anche su dati non visti in precedenza.

Nonostante il valore indiscusso del MIT-BIH Arrhythmia Database come benchmark standard nel settore, la letteratura sottolinea anche alcune limitazioni, quali la dimensione ridotta del dataset e la scarsa rappresentatività di alcune classi di aritmie rare. Per ovviare a queste problematiche, è frequente l'uso di tecniche di bilanciamento dati, come il data augmentation o il re-sampling, e lo sviluppo di modelli robusti a classi sbilanciate.

3 Il Dataset

Il set di dati MIT-BIH include i segnali ECG di 48 soggetti diversi, registrati alla frequenza di campionamento di 360 Hz. Ogni registrazione contiene due derivazioni ECG. E si possono avere diverse combinazioni derivazioni. In particolare, sono disponibili registrazioni con le seguenti coppie di derivazioni: MLII-V1, MLII-V2, MLII-V4, MLII-V5 e V5-V2.

Per questo lavoro, sono stati utilizzati esclusivamente i record contenenti le derivazioni MLII e V1, in quanto nella letteratura è comune utilizzare la derivazione II (MLII) per il rilevamento dei battiti cardiaci e la V1 in quanto consente di osservare dettagli che la derivazione MLII (Lead II) potrebbe non evidenziare chiaramente.

Pertanto sono stati scartati i record che non presentano la coppia MLII-V1, nello specifico: { 117, 103, 124, 100, 114, 123, 104, 102 } .

Questa selezione ha permesso di garantire omogeneità nelle derivazioni utilizzate e di sfruttare la derivazione MLII, riconosciuta come standard per l'identificazione dei battiti cardiaci nella maggior parte degli studi clinici e di ricerca.

3.1 Pre-processing

Le operazioni di pre-elaborazione adottate per l'estrazione dei battiti dal segnale ECG sono volutamente semplici e non prevedono alcuna forma di filtraggio o tecniche di rimozione del rumore, con l'obiettivo di mantenere il più possibile l'integrità del segnale originale.

Per l'estrazione dei battiti dal segnale ECG sono stati seguiti i seguenti passaggi:

1. **Normalizzazione:** il segnale ECG grezzo è stato normalizzato utilizzando la tecnica *min-max scaling*, in modo da riportare tutti i valori nell'intervallo compreso tra 0 e 1.
2. **Identificazione dei picchi:** utilizzando le annotazioni fornite nei file del database MIT-BIH Arrhythmia, sono stati individuati i picchi R di ogni battito cardiaco.
3. **Suddivisione e campionamento dei battiti:** il segnale continuo è stato suddiviso in finestre da 280 campioni centrate su ciascun picco R individuato. Ad ogni finestra è stata assegnata l'etichetta corrispondente al tipo di battito, come indicato nel file di annotazione.

Ottenendo così un totale di 94.773 campioni di battiti cardiaci.

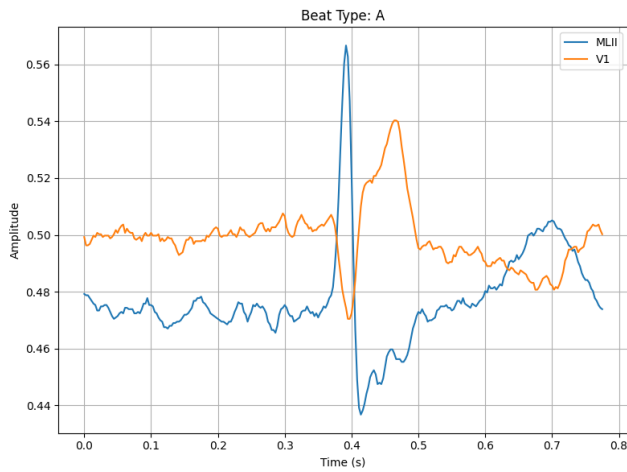


Figure 1: Esemplio di un battito di tipo A

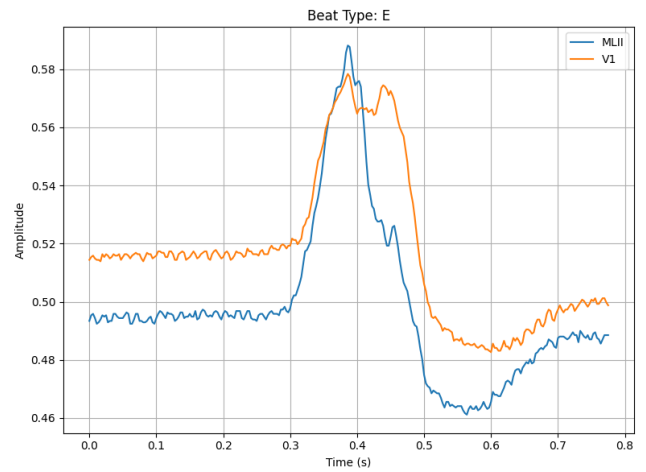


Figure 2: Esemplio di un battito di tipo E

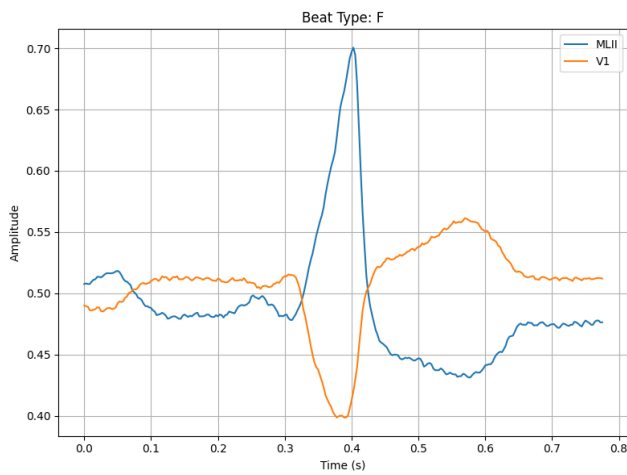


Figure 3: Esemplio di un battito di tipo F

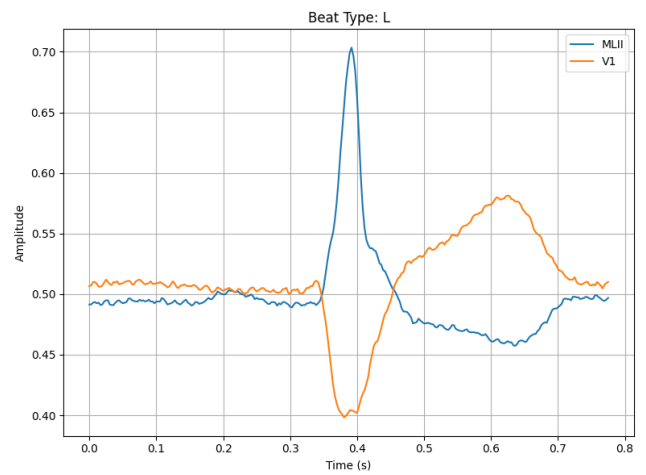


Figure 4: Esemplio di un battito di tipo L

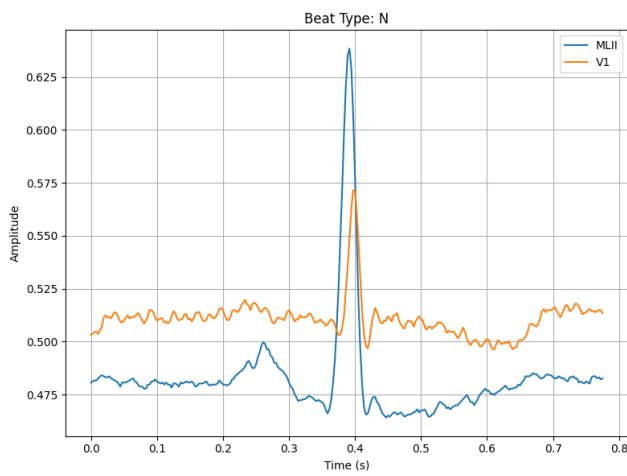


Figure 5: Esemplio di un battito di tipo N

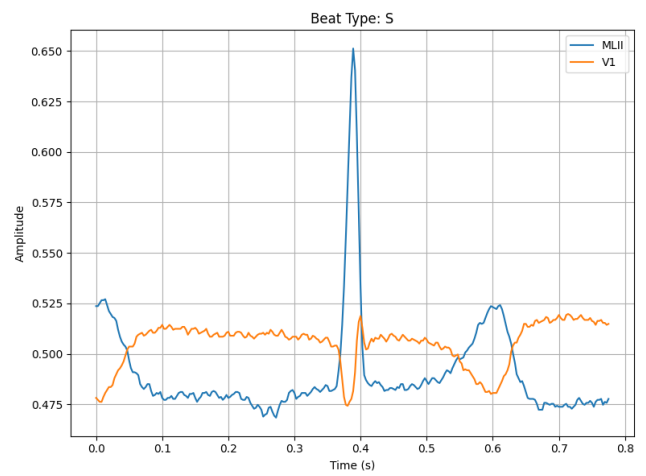


Figure 6: Esemplio di un battito di tipo S

3.2 Organizzazione dei dati

Al termine della fase di pre-processing, le finestre estratte sono state suddivise casualmente in due sottoinsiemi: il 75% è stato assegnato al *training set*, mentre il restante 25% è stato utilizzato per la valutazione del modello.

Per attenuare il problema dello sbilanciamento delle classi, particolarmente rilevante in questo tipo di dati, è stata applicata la tecnica SMOTE (*Synthetic Minority Over-sampling Technique*) al dataset di training, operando sulle singole tipologie di battiti cardiaci.

3.3 Operazione di SMOTE

La tabella seguente mostra il numero di battiti per ciascuna classe prima e dopo l'applicazione della tecnica SMOTE:

Classe	N		A	V	/	F	L	a
Prima	49200	98	1873	5272	2715	596	6056	113
Dopo	49200	19600	7492	21088	10860	4768	24224	1582

Classe	R	j	J	E	S	f	e
Prima	4296	168	39	80	1	195	12
Dopo	17184	2352	7800	16000	1	2730	2400

3.4 Raggruppamento in categorie

Come avviene in diversi studi in letteratura, i battiti cardiaci sono stati successivamente raggruppati in cinque categorie principali, stabiliti dall'Association for the Advancement of Medical Instrumentation (AAMI):

Category	Class
N	<ul style="list-style-type: none"> • Normal beat (N) • Left and right bundle branch block beats (L,R) • Atrial escape beat (e) • Nodal (junctional) escape beat (j)
S	<ul style="list-style-type: none"> • Atrial premature beat (A) • Aberrated atrial premature beat (a) • Nodal (junctional) premature beat (J) • Supraventricular premature beat (S)
V	<ul style="list-style-type: none"> • Premature ventricular contraction (V) • Ventricular escape beat (E)
F	<ul style="list-style-type: none"> • Fusion of ventricular and normal beat (F)
Q	<ul style="list-style-type: none"> • Paced beat (/) • Fusion of paced and normal beat (f) • Unclassifiable beat (U)
De	

Figure 7: Tabella delle categorie AAMI

4 Modelli utilizzati

Per questo progetto sono stati utilizzati e messi a confronto cinque modelli di reti neurali: ResNet18, ResNet34, ResNet50 e due versioni di un classico ViT (*ViT_V1* e *ViT_V2*).

Le due versioni di ViT utilizzano l'architettura Vision Transformer adattata per segnali monodimensionali (1D) con un classico positional encoding. La prima versione, *ViT_V1*, implementa una struttura standard composta dai seguenti parametri: `kernel_size = 10`, `kernel_stride = 10`, `kernel_in_channels = 2`, `embed_dim = 128`, `num_layers = 4`, `num_heads = 4`, `mlp_dim = 256`, `dropout = 0.1`. In questa configurazione, il segnale in ingresso viene suddiviso in patch di dimensione 10, ciascuna trasformata in uno spazio latente di dimensione 128, per poi essere processata da una serie di trasformatori multi-head self-attention.

La seconda versione, *ViT_V2*, si differenzia rispetto alla prima per la modifica introdotta nel primo layer di estrazione delle feature: è stato sostituito il layer convolutivo iniziale con una struttura più profonda e complessa, ispirata al lavoro di Mousavi et al. [9]. Questo blocco convolutivo, definito nella classe `CNNFeatureExtractor`, utilizza una serie di tre convoluzioni sequenziali con kernel di dimensioni diverse, normalizzazioni batch e funzioni di attivazione ReLU, accompagnate da operazioni di max pooling.

```

1 class CNNFeatureExtractor(nn.Module):
2     def __init__(self, input_length=280, num_channels=2, embed_dim=128):
3         super(CNNFeatureExtractor, self).__init__()
4         assert input_length % 10 == 0, "Input length must be divisible by 10"
5         assert embed_dim % 4 == 0, "Embed dimension must be divisible by 4"
6
7         self.embed_dim = embed_dim
8         self.embed_dim_2 = embed_dim // 2
9         self.embed_dim_4 = embed_dim // 4
10        self.input_length = input_length
11        self.num_channels = num_channels
12        self.patchSize = 10
13
14        self.input_reshape_channels = int(self.input_length/self.patchSize) *
self.num_channels
15
16        self.convLayers = nn.Sequential(
17
18            nn.Conv1d(in_channels= self.input_reshape_channels, out_channels=
self.embed_dim_4, kernel_size=3, stride=1, padding=1),
19            nn.BatchNorm1d(self.embed_dim_4),
20            nn.ReLU(),
21            nn.MaxPool1d(kernel_size=2, stride=2), # Output length: 10 -> 5
22
23            nn.Conv1d(in_channels=self.embed_dim_4, out_channels=self.
embed_dim_2, kernel_size=3, stride=1, padding=1),
24            nn.BatchNorm1d(self.embed_dim_2),
25            nn.ReLU(),
26            nn.MaxPool1d(kernel_size=3, stride=1),# Output length: 5 -> 3
27
28            nn.Conv1d(in_channels=self.embed_dim_2, out_channels=self.embed_dim
, kernel_size=1, stride=1, padding=0),
29            nn.BatchNorm1d(self.embed_dim),
30            nn.ReLU(),
31        )
32
33    def forward(self, x):
34        batch_size, _, _ = x.shape
35
36        # Reshape to (batch_size, 10*num_channels, 28)
37        x = x.view(batch_size, self.input_reshape_channels, self.patchSize)
38        return self.convLayers(x)
39

```

Tale modifica consente di estrarre rappresentazioni più ricche e gerarchiche dalle patch di input, migliorando la capacità del modello di cogliere pattern complessi nel segnale ECG. In particolare, il segnale viene prima ridimensionato e trasformato in una rappresentazione con un numero aumentato di canali, poi elaborato da convoluzioni 1D con kernel di dimensione 3 e 1, e infine proiettato nello spazio latente di dimensione 128.

Addestramento

L'addestramento di ciascun modello è stato effettuato per un totale di 60 epoche, utilizzando un learning rate iniziale pari a $2 \cdot 10^{-4}$ e un batch size di 256 elementi per i modelli *ResNet* e di 32 elementi per i modelli *ViT*. Come funzione di loss è stata adottata la *cross-entropy*, mentre per l'ottimizzazione dei pesi è stato impiegato l'algoritmo *Adagrad*, con un valore di weight decay pari a $1 \cdot 10^{-3}$. Inoltre, è stato utilizzato uno scheduler *ReduceLROnPlateau* con un valore minimo di learning rate impostato a $\eta_{\min} = 5 \cdot 10^{-6}$.

Di seguito sono riportati i risultati ottenuti durante il processo di training di ciascun modello.

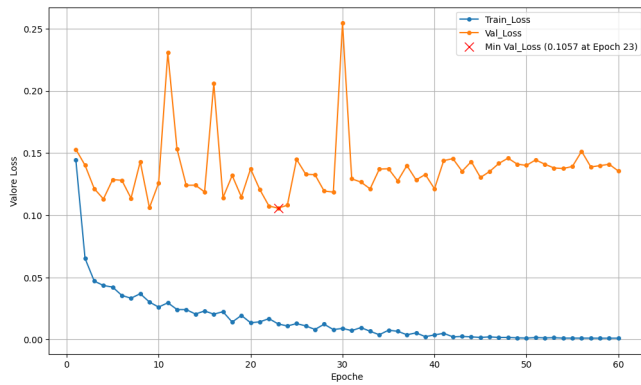


Figure 8: Loss del modello ResNet18 in funzione delle epoche durante il processo di addestramento

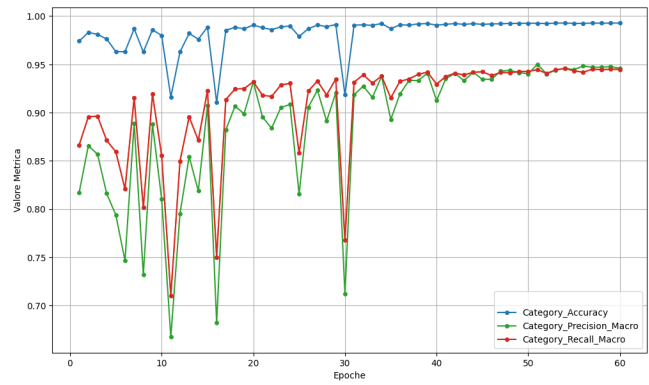


Figure 9: Metriche del modello ResNet18 in funzione delle epoche durante il processo di addestramento

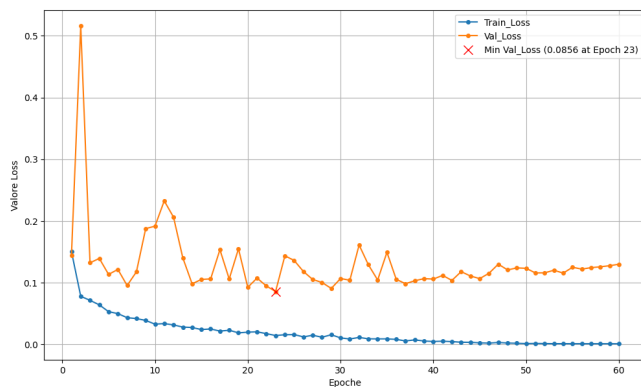


Figure 10: Loss del modello ResNet34 in funzione delle epoche durante il processo di addestramento

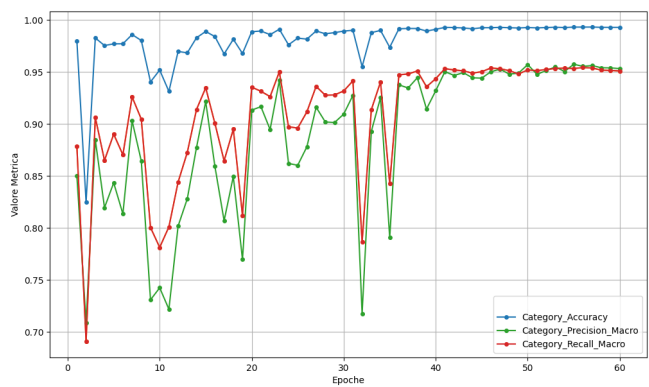


Figure 11: Metriche del modello ResNet34 in funzione delle epoche durante il processo di addestramento

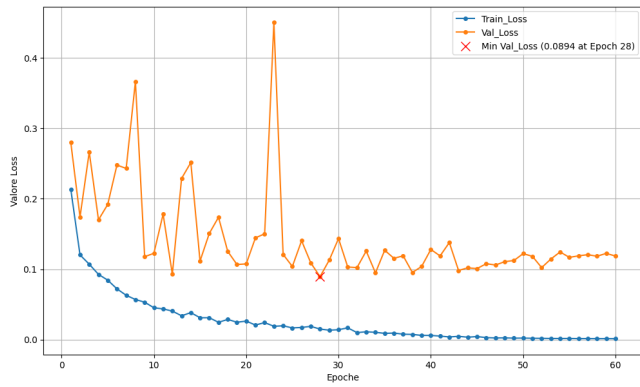


Figure 12: Loss del modello ResNet50 in funzione delle epoche durante il processo di addestramento

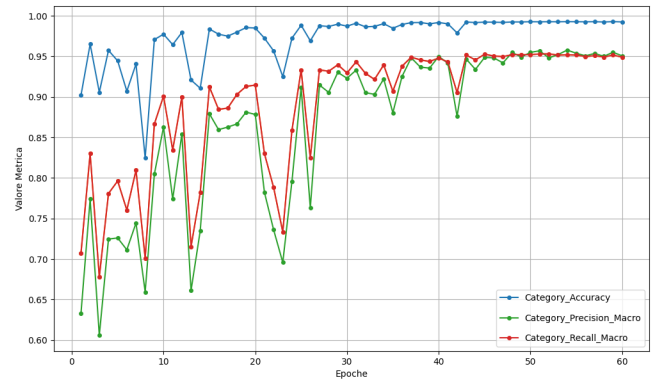


Figure 13: Metriche del modello ResNet50 in funzione delle epoche durante il processo di addestramento

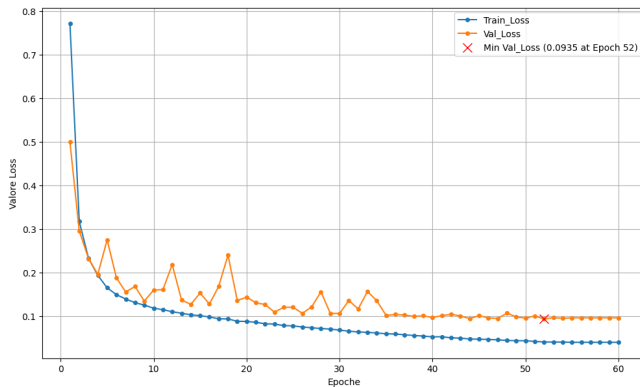


Figure 14: Loss del modello ViT_V1 in funzione delle epoche durante il processo di addestramento

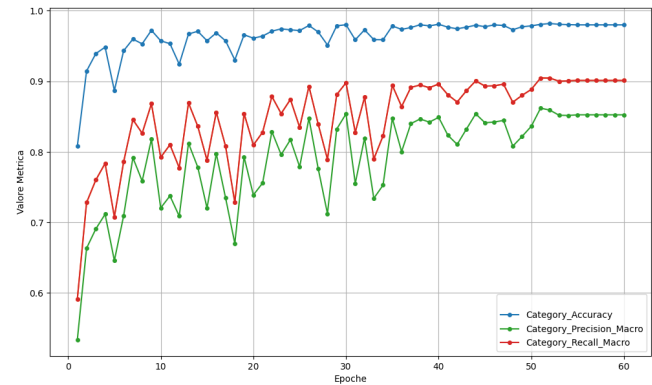


Figure 15: Metriche del modello ViT_V1 in funzione delle epoche durante il processo di addestramento

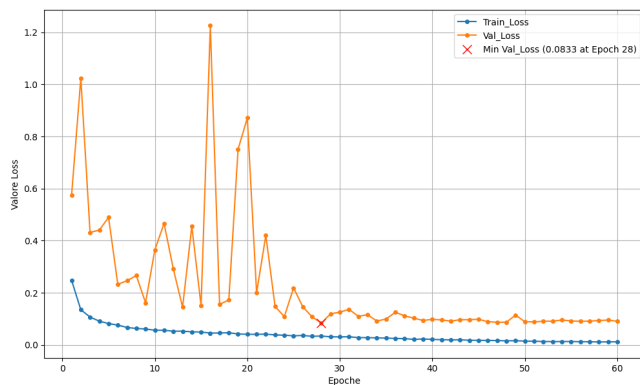


Figure 16: Loss del modello ViT_V2 in funzione delle epoche durante il processo di addestramento

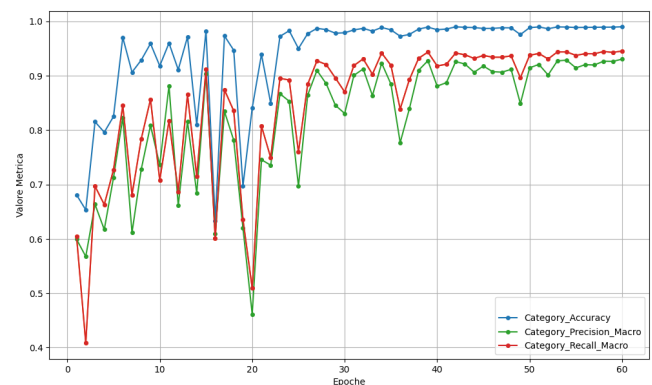


Figure 17: Metriche del modello ViT_V2 in funzione delle epoche durante il processo di addestramento

5 Risultati ottenuti

Per la valutazione di ciascun modello è stata utilizzata la versione corrispondente all'epoca in cui è stato registrato il valore minimo della *loss*. La categoria Q non è stata considerata in fase di valutazione, in quanto contenente valori non classificabili.

Categoria	Istanze	Precision	Recall	F1
Normal	19899	0.9974	0.9954	0.9964
SVEB	676	0.9346	0.9512	0.9428
VEB	1782	0.9822	0.9882	0.9852
Fusion	198	0.8271	0.8939	0.8592
Weighted Avg		0.9928	0.9926	0.9927
Overall Accuracy	0.9926			
Overall Kappa	0.9659			

Table 1: Risultati di Valutazione per ResNet18

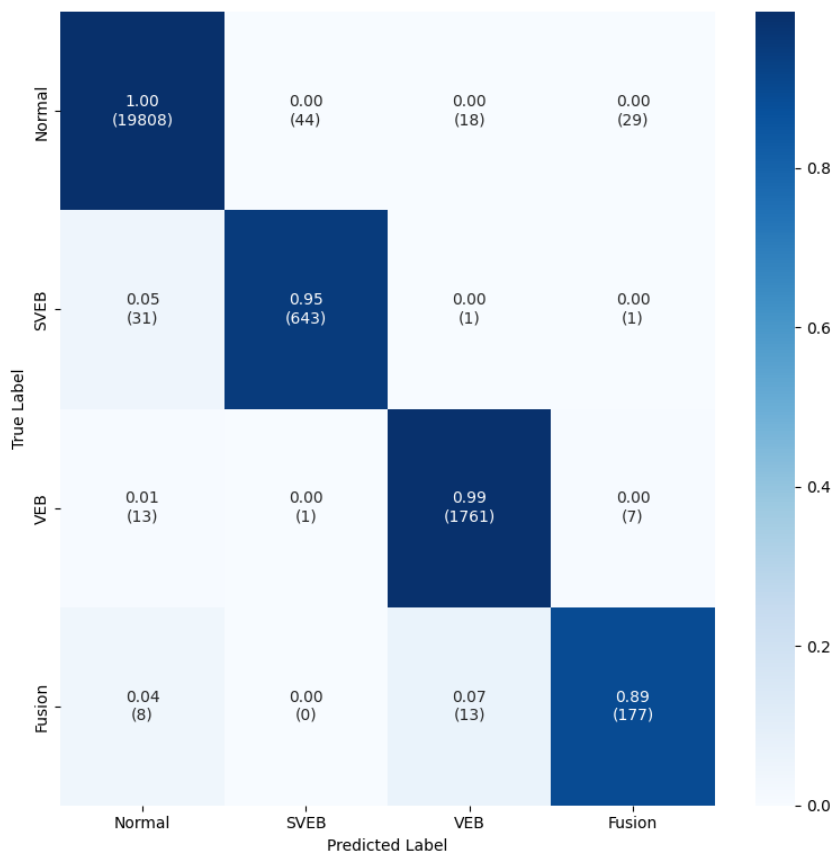


Figure 18: Matrice di confusione del modello ResNet18

Categoria	Istanze	Precision	Recall	F1
Normal	19899	0.9973	0.9937	0.9955
SVEB	676	0.9192	0.9423	0.9306
VEB	1782	0.9613	0.9888	0.9748
Fusion	198	0.8911	0.9091	0.9000
Weighted Avg		0.9912	0.9910	0.9911
Overall Accuracy	0.9910			
Overall Kappa	0.9588			

Table 2: Risultati di Valutazione per ResNet34

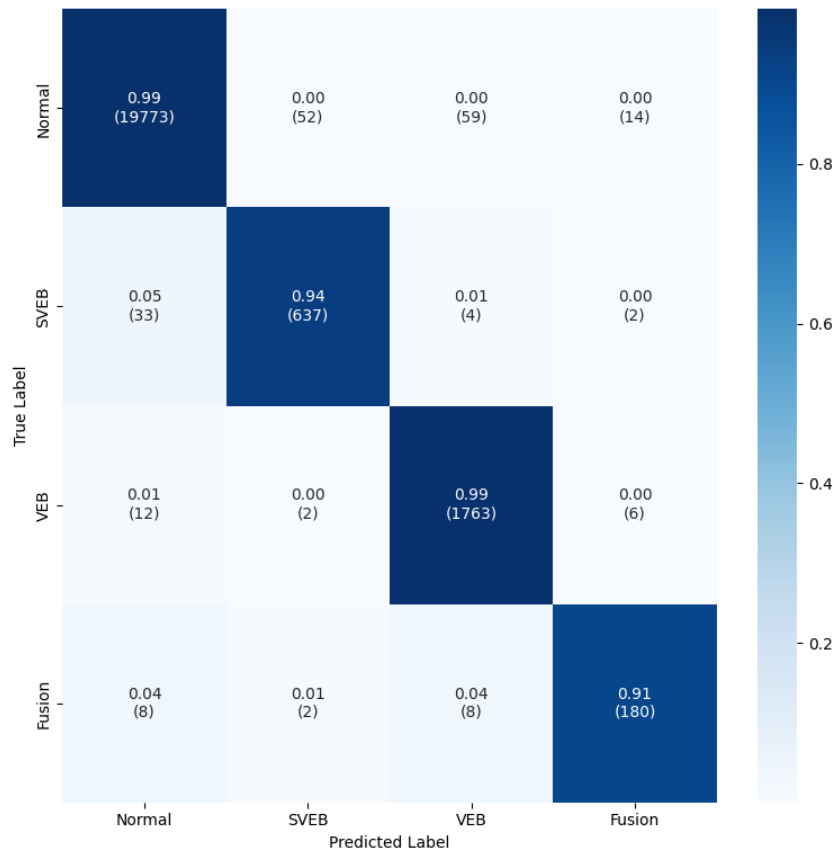


Figure 19: Matrice di confusione del modello ResNet34

Categoria	Istanze	Precision	Recall	F1
Normal	19899	0.9969	0.9895	0.9932
SVEB	676	0.8499	0.9393	0.8923
VEB	1782	0.9525	0.9787	0.9654
Fusion	198	0.8222	0.9343	0.8747
Weighted Avg		0.9874	0.9867	0.9869
Overall Accuracy	0.9867			
Overall Kappa	0.9393			

Table 3: Risultati di Valutazione per ResNet50

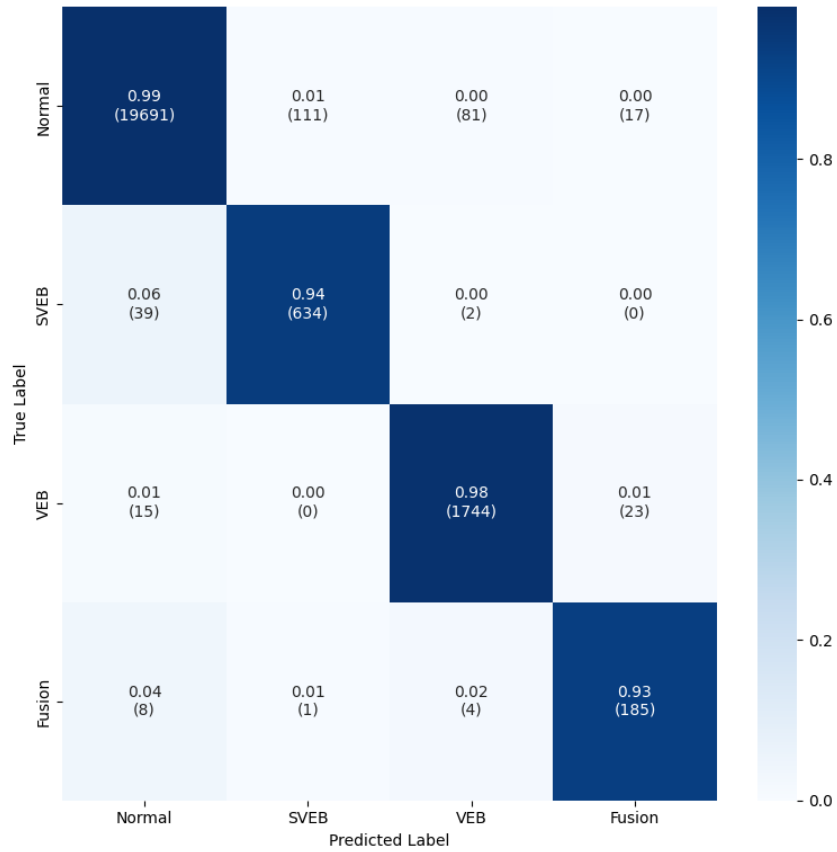


Figure 20: Matrice di confusione del modello ResNet50

Categoria	Istanze	Precision	Recall	F1
Normal	19899	0.9973	0.9844	0.9908
SVEB	676	0.7806	0.9423	0.8539
VEB	1782	0.9434	0.9719	0.9574
Fusion	198	0.7176	0.9495	0.8174
Weighted Avg		0.9841	0.9818	0.9825
Overall Accuracy	0.9818			
Overall Kappa	0.9189			

Table 4: Risultati di Valutazione per ViT_V1

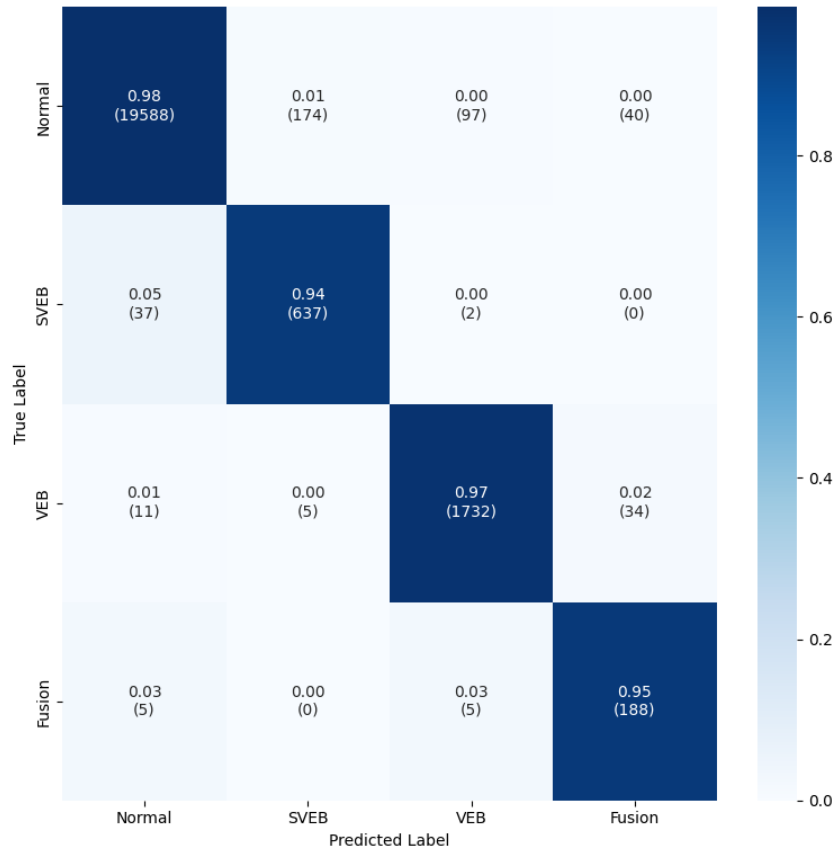


Figure 21: Matrice di confusione del modello ViT_V1

Categoria	Istanze	Precision	Recall	F1
Normal	19899	0.9969	0.9877	0.9923
SVEB	676	0.8035	0.9497	0.8705
VEB	1782	0.9584	0.9686	0.9634
Fusion	198	0.7824	0.9444	0.8558
Weighted Avg		0.9862	0.9847	0.9852
Overall Accuracy	0.9847			
Overall Kappa	0.9309			

Table 5: Risultati di Valutazione per ViT_V2

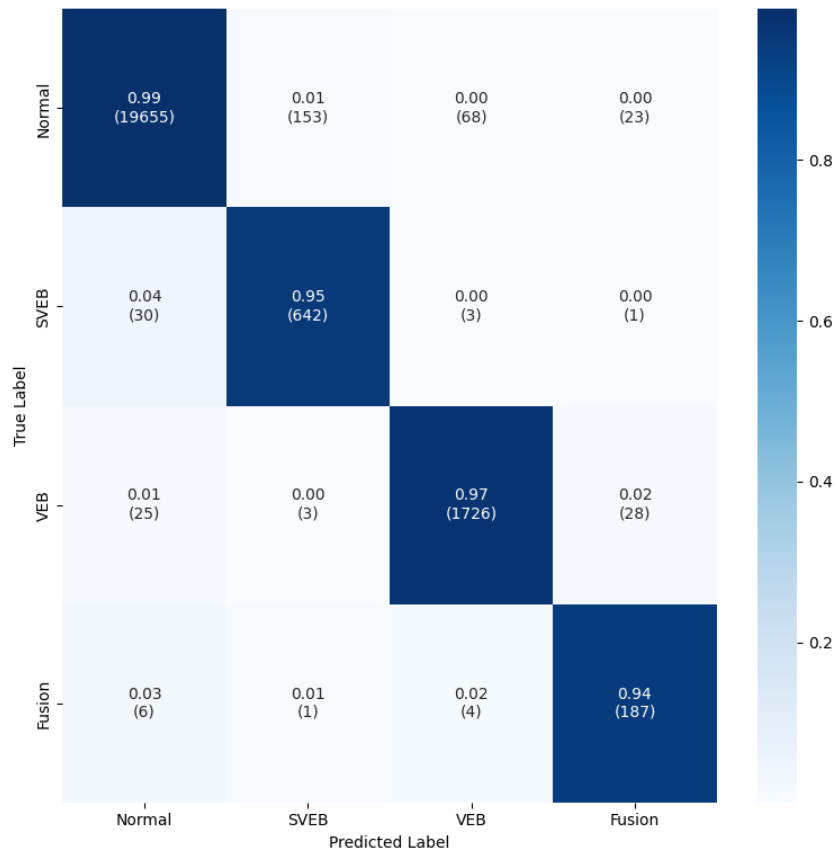


Figure 22: Matrice di confusione del modello ViT_V2

Modello	Overall Accuracy	Overall Kappa
ResNet18	0.9926	0.9659
ResNet34	0.9910	0.9588
ResNet50	0.9867	0.9393
ViT_V1	0.9818	0.9189
ViT_V2	0.9847	0.9309

Table 6: Riepilogo Overall Accuracy e Overall Kappa per Modello

6 Analisi dei Risultati

Questa sezione presenta un’analisi dei risultati di valutazione ottenuti dai diversi modelli di classificazione, ovvero **ResNet18**, **ResNet34**, **ResNet50**, **ViT_V1**, e **ViT_V2**.

6.1 Performance del Modello ResNet18

Il modello **ResNet18** dimostra performance complessive eccellenti, come evidenziato dalla sua **Overall Accuracy** del 0.9926 e un **Overall Kappa** di 0.9659. Questi valori indicano una notevole capacità del modello di classificare correttamente le istanze e un accordo molto elevato tra le classificazioni del modello e le etichette reali.

Analizzando le metriche per categoria:

- **Normal:** Con 19899 istanze, questa categoria presenta valori di **Precision** (0.9974), **Recall** (0.9954) e **F1-Score** (0.9964) estremamente elevati. Ciò suggerisce che il modello identifica quasi perfettamente le istanze "Normal", con pochissimi falsi positivi o falsi negativi.
- **SVEB:** Per le 676 istanze di questa categoria, il modello raggiunge un buon equilibrio con una **Precision** di 0.9346, una **Recall** di 0.9512 e un **F1-Score** di 0.9428. Questo indica una solida capacità di rilevare correttamente le istanze SVEB, pur mantenendo un tasso di falsi positivi contenuto.
- **VEB:** Con 1782 istanze, la categoria VEB mostra risultati impressionanti: **Precision** 0.9822, **Recall** 0.9882 e **F1-Score** 0.9852. Il modello è altamente efficace nel classificare le istanze VEB, minimizzando sia i falsi positivi che i falsi negativi.
- **Fusion:** Sebbene questa categoria abbia solo 198 istanze, il modello ottiene una **Precision** di 0.8271, una **Recall** di 0.8939 e un **F1-Score** di 0.8592. La **Recall** relativamente alta suggerisce che il modello è bravo a identificare la maggior parte delle istanze "Fusion", anche se con una precisione leggermente inferiore rispetto ad altre categorie, indicando qualche falso positivo.

Il **Weighted Avg F1-Score** di 0.9927 conferma la robustezza complessiva del modello ResNet18 su tutte le categorie.

6.2 Performance del Modello ResNet34

Il modello **ResNet34** segue da vicino ResNet18 in termini di prestazioni generali, con un **Overall Accuracy** di 0.9910 e un **Overall Kappa** di 0.9588. Questi risultati continuano a indicare un modello altamente performante, sebbene leggermente inferiore a ResNet18.

Esaminando le metriche per categoria:

- **Normal:** Analogamente a ResNet18, la categoria "Normal" mantiene elevate prestazioni con **Precision** 0.9973, **Recall** 0.9937 e **F1-Score** 0.9955.
- **SVEB:** I valori per SVEB sono **Precision** 0.9192, **Recall** 0.9423 e **F1-Score** 0.9306. Si osserva una lieve diminuzione della precisione rispetto a ResNet18.

- **VEB:** Per VEB, le metriche sono **Precision** 0.9613, **Recall** 0.9888 e **F1-Score** 0.9748. La recall rimane eccellente, ma la precisione si riduce leggermente.
- **Fusion:** La categoria Fusion mostra un miglioramento in termini di precisione e F1-Score rispetto a ResNet18, con **Precision** 0.8911, **Recall** 0.9091 e **F1-Score** 0.9000. Questo suggerisce che ResNet34 è più accurato nel classificare le istanze "Fusion" che identifica.

Il **Weighted Avg F1-Score** di 0.9911 è molto solido, confermando la buona performance generale di ResNet34.

6.3 Performance del Modello ResNet50

Il modello **ResNet50** mostra un leggero calo nelle prestazioni complessive rispetto ai modelli ResNet precedenti, con un'**Overall Accuracy** di 0.9867 e un **Overall Kappa** di 0.9393. Sebbene ancora buoni, questi valori indicano una performance leggermente meno ottimale.

Analisi per categoria:

- **Normal:** Le metriche rimangono molto alte: **Precision** 0.9969, **Recall** 0.9895 e **F1-Score** 0.9932.
- **SVEB:** Per SVEB, la **Precision** scende a 0.8499, mentre la **Recall** rimane alta a 0.9393, portando a un **F1-Score** di 0.8923. Questo suggerisce un aumento dei falsi positivi per questa categoria.
- **VEB:** Le metriche sono **Precision** 0.9525, **Recall** 0.9787 e **F1-Score** 0.9654. Si nota una riduzione in precisione e F1-Score rispetto ai modelli ResNet più piccoli.
- **Fusion:** La categoria Fusion presenta una **Precision** di 0.8222, **Recall** di 0.9343 e **F1-Score** di 0.8747. Anche qui, una buona recall ma una precisione che potrebbe essere migliorata.

Il **Weighted Avg F1-Score** di 0.9869 riflette il leggero peggioramento delle prestazioni complessive.

6.4 Performance del Modello ViT_V1

Il modello **ViT_V1** presenta le prestazioni più basse tra i modelli analizzati, con un'**Overall Accuracy** di 0.9818 e un **Overall Kappa** di 0.9189. Questi valori indicano una performance accettabile ma inferiore rispetto alla famiglia ResNet.

Dettaglio per categoria:

- **Normal:** Le metriche sono **Precision** 0.9973, **Recall** 0.9844 e **F1-Score** 0.9908. La recall per la categoria "Normal" è la più bassa tra tutti i modelli, indicando più falsi negativi.
- **SVEB:** La **Precision** di SVEB è notevolmente più bassa a 0.7806, sebbene la **Recall** sia ancora alta a 0.9423, risultando in un **F1-Score** di 0.8539. Ciò indica un numero significativo di falsi positivi.
- **VEB:** Le metriche sono **Precision** 0.9434, **Recall** 0.9719 e **F1-Score** 0.9574. Anche qui, la precisione è inferiore rispetto ai modelli ResNet.
- **Fusion:** Questa categoria mostra la **Precision** più bassa in assoluto a 0.7176, con una **Recall** di 0.9495 e un **F1-Score** di 0.8174. Il modello ha difficoltà a distinguere le istanze "Fusion" dagli altri tipi, pur riuscendo a rilevarne la maggior parte.

Il **Weighted Avg F1-Score** di 0.9825 conferma le prestazioni meno ottimali di ViT_V1.

6.5 Performance del Modello ViT_V2

Il modello **ViT_V2** mostra un leggero miglioramento rispetto a ViT_V1, raggiungendo un'**Overall Accuracy** di 0.9847 e un **Overall Kappa** di 0.9309. Questo lo posiziona tra ResNet50 e ViT_V1 in termini di prestazioni complessive.

Analisi per categoria:

- **Normal:** Le metriche sono **Precision** 0.9969, **Recall** 0.9877 e **F1-Score** 0.9923. Un miglioramento nella recall rispetto a ViT_V1.
- **SVEB:** Si registra una **Precision** di 0.8035 e una **Recall** molto alta di 0.9497, con un **F1-Score** di 0.8705. Un miglioramento in precisione rispetto a ViT_V1 per questa categoria.
- **VEB:** Le metriche sono **Precision** 0.9584, **Recall** 0.9686 e **F1-Score** 0.9634. La precisione è leggermente migliore rispetto a ViT_V1.
- **Fusion:** Anche per Fusion si nota un miglioramento, con **Precision** 0.7824, **Recall** 0.9444 e **F1-Score** 0.8558. Sebbene la precisione sia ancora la più bassa tra le categorie, è migliorata rispetto a ViT_V1.

Il **Weighted Avg F1-Score** di 0.9852 indica una performance complessivamente migliore di ViT_V2 rispetto alla sua versione precedente.

7 Conclusioni

Dall'analisi dei risultati di valutazione per i diversi modelli, emerge un quadro chiaro delle loro capacità nel compito di classificazione. Il modello **ResNet18** si conferma il **miglior performer** tra tutti i modelli testati, raggiungendo la più alta **Overall Accuracy** (0.9926) e il miglior **Overall Kappa** (0.9659). Le sue prestazioni sono eccezionali su tutte le categorie, dimostrando un'elevata precisione e recall, in particolare per le classi "Normal", "SVEB" e "VEB". La sua robustezza e capacità di generalizzazione lo rendono la scelta più efficace per questo specifico set di dati e problema di classificazione. Il modello **ResNet34** si posiziona come il **secondo miglior performer**, con metriche molto vicine a quelle di ResNet18. La sua performance è complessivamente molto solida, con un'ottima capacità di classificazione per la maggior parte delle categorie. È interessante notare il suo leggero miglioramento nella Precision e F1-Score per la categoria "Fusion" rispetto a ResNet18, sebbene con un slight trade-off in altre categorie. I modelli **ResNet50**, **ViT_V1** e **ViT_V2** mostrano prestazioni complessivamente inferiori rispetto a ResNet18 e ResNet34. Tra questi, **ViT_V2** si comporta meglio di **ViT_V1**, indicando che le modifiche apportate alla sua architettura hanno portato a un miglioramento. Tuttavia, entrambi i modelli ViT presentano metriche di precisione inferiori per le categorie minori come "SVEB" e "Fusion", suggerendo una maggiore tendenza ai falsi positivi per queste classi. ResNet50, nonostante sia un modello più profondo, non ha superato i suoi predecessori più leggeri (ResNet18 e ResNet34) in questo contesto specifico, indicando che la maggiore complessità del modello potrebbe non tradursi sempre in prestazioni superiori, soprattutto se il set di dati o il task non beneficiano di una maggiore capacità del modello.

In sintesi, i modelli basati sull'architettura **ResNet** si sono dimostrati superiori rispetto ai modelli **Transformer** per questo compito di classificazione, con **ResNet18** come scelta ottimale. La loro capacità di gestire efficacemente le diverse categorie di istanze, anche quelle meno numerose, e di mantenere un elevato equilibrio tra precisione e recall, li rende particolarmente adatti per applicazioni dove l'accuratezza e la minimizzazione degli errori di classificazione sono cruciali. Le matrici di confusione associate (non visualizzate qui ma menzionate nel report) sarebbero fondamentali per una comprensione più approfondita degli errori specifici commessi da ciascun modello.

Per quanto riguarda i modelli **ViT**, sebbene le versioni attuali (ViT_V1 e ViT_V2) non abbiano raggiunto le performance dei modelli ResNet, c'è un potenziale di miglioramento significativo. Si potrebbe considerare di aumentare la complessità dei modelli ViT, ad esempio aggiungendo un maggior numero di layer o aumentando la dimensione dei patch, per esplorare se una maggiore capacità del modello possa portare a prestazioni superiori. Ulteriori ottimizzazioni degli iperparametri e tecniche di training specifiche per i ViT potrebbero anch'esse contribuire a colmare il divario con le architetture ResNet.

8 Articoli

- [1] Ye C, Coimbra MT, Vijaya Kumar BK. Arrhythmia detection and classification using morphological and dynamic features of ECG signals. *Annu Int Conf IEEE Eng Med Biol Soc.* 2010;2010:1918-21. doi: 10.1109/IEMBS.2010.5627645. PMID: 21097000.
- [2] Osowski, S., Hoai, L. T., Markiewicz, T. (2004). Support vector machine-based expert system for reliable heartbeat recognition. *IEEE Transactions on Biomedical Engineering*, 51(4), 582–589. <https://doi.org/10.1109/TBME.2003.821650>
- [3] Sharma, S., Pachori, R. B., Upadhyay, A. (2014). Automated diagnosis of ECG using hybrid discrete wavelet transform and random forest classifier. *Biomedical Signal Processing and Control*, 14, 1–11. <https://doi.org/10.1016/j.bspc.2014.02.002>
- [4] Martis, R. J., Acharya, U. R., Mandana, K. M., Ray, A. K. (2013). Application of higher order statistics for atrial arrhythmia classification. *Biomedical Signal Processing and Control*, 8(6), 888–900. <https://doi.org/10.1016/j.bspc.2013.05.005>
- [5] Acharya, U & Oh, Shu Lih & Hagiwara, Yuki & Tan, Jen Hong & Abdul Rahim, Muhammad Adam & Gertych, Arkadiusz & Tan, Ru San. (2017). A Deep Convolutional Neural Network Model to Classify Heartbeats. *Computers in Biology and Medicine*. 89. 10.1016/j.compbiomed.2017.08.022.
- [6] Kiranyaz S, Ince T, Hamila R, Gabbouj M. Convolutional Neural Networks for patient-specific ECG classification. *Annu Int Conf IEEE Eng Med Biol Soc.* 2015;2015:2608-11. doi: 10.1109/EMBC.2015.7318926. PMID: 26736826.
- [7] Mousavi, S., Afghah, F. (2020). INTER- AND INTRA-PATIENT ECG HEART-BEAT CLASSIFICATION FOR ARRHYTHMIA DETECTION: A SEQUENCE TO SEQUENCE DEEP LEARNING APPROACH. *Transactions on Signal Processing*. <https://doi.org/10.18280/ts.410640>
- [8] Kachuee, M., Fazeli, S., Sarrafzadeh, M. (2018). ECG Heartbeat Classification: A Deep Transferable Representation. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)* (pp. 443–444). IEEE. <https://doi.org/10.1109/ICHI.2018.00076>
- [9] Mousavi S, Afghah F. Inter- and intra-patient ECG heartbeat classification for arrhythmia detection: A sequence to sequence deep learning approach. *Proc IEEE Int Conf Acoust Speech Signal Process.* 2019 May;2019:1308-1312. doi: 10.1109/icassp.2019.8683140. Epub 2019 Apr 17. PMID: 33082716; PMCID: PMC7570975.