



**UNIVERSITÀ DEGLI STUDI DI CATANIA**

DIPARTIMENTO DI MATEMATICA E INFORMATICA

CORSO DI LAUREA TRIENNALE IN INFORMATICA

---

*Simone Battiato*

Algoritmo di Machine Learning per l'analisi Forense  
delle manoscritture per l'identificazione dell'autore

---

RELAZIONE PROGETTO FINALE

---

Relatore: Luca Guarnera  
Correlatore: Sebastiano Battiato

---

Anno Accademico 2023 - 2024

# Abstract

In questa tesi, presentiamo un algoritmo di machine learning progettato per l'analisi forense della scrittura a mano al fine di identificare l'autore. L'analisi della scrittura a mano, parte integrante della scienza forense, sfrutta le caratteristiche distintive dello stile di scrittura di un individuo per accertarne l'identità. Lo studio comprende metodologie forensi tradizionali e approcci moderni che utilizzano l'intelligenza artificiale (IA).

La nostra ricerca si concentra sullo sviluppo e l'implementazione di reti neurali convoluzionali (CNN) e reti siamesi per migliorare l'accuratezza e l'affidabilità dell'identificazione della scrittura a mano. Il dataset utilizzato per l'addestramento include campioni diversi provenienti da più individui, garantendo la robustezza e la generalizzazione dei modelli.

La fase di preprocessing prevede la conversione del testo scritto a mano in immagini digitali, seguita dalla riduzione del rumore e dalla segmentazione per isolare i singoli caratteri. L'estrazione delle caratteristiche viene eseguita utilizzando le CNN, che sono perfezionate su modelli pre-addestrati come ResNet, AlexNet e SqueezeNet. L'architettura della rete siamese, impiegando la perdita contrastiva, viene poi utilizzata per confrontare e classificare i campioni di scrittura a mano basati su embedding appresi.

I risultati sperimentali dimostrano l'efficacia del nostro approccio, con i migliori modelli che raggiungono un'elevata accuratezza sia nelle fasi di addestramento che di validazione. L'analisi include curve di apprendimento, matrici di confusione e metriche di performance come precisione, richiamo e F1 score. Le valutazioni comparative con modelli pre-addestrati evidenziano la superiorità delle nostre reti perfezionate.

La ricerca evidenzia il potenziale del machine learning nell'analisi forense della scrittura a mano, offrendo miglioramenti significativi rispetto ai metodi tradizionali. I lavori futuri esploreranno ulteriori ottimizzazioni degli iperparametri e l'espansione del dataset per affrontare le attuali limitazioni

e migliorare il potere discriminante del modello.

**Parole chiave:** Scienza Forense, Analisi della Scrittura a Mano, Machine Learning, Reti Neurali Convoluzionali, Reti Siamesi, Identificazione dell'Autore.

# Indice

<b>1</b>	<b>Introduzione alle scienze Forensi</b>	<b>6</b>
1.1	Definizione e Origini delle Scienze Forensi . . . . .	6
1.2	Fasi delle Scienze Forensi . . . . .	6
1.2.1	Raccolta delle Prove . . . . .	6
1.2.2	Analisi delle Prove . . . . .	7
1.2.3	Interpretazione dei Risultati . . . . .	7
1.2.4	Presentazione delle Prove . . . . .	7
1.3	Principali Discipline delle Scienze Forensi . . . . .	7
1.3.1	Analisi del DNA . . . . .	8
1.3.2	Tossicologia Forense . . . . .	8
1.3.3	Antropologia Forense . . . . .	8
1.3.4	Balistica Forense . . . . .	8
1.4	Digital Forensics . . . . .	8
1.4.1	Tipi di Prove Digitali . . . . .	8
1.4.2	Metodologie di Digital Forensics . . . . .	9
1.4.3	Importanza della Digital Forensics . . . . .	9
1.5	Digital Forgery . . . . .	9
1.5.1	Tipi di Falsificazione Digitale . . . . .	9
1.5.2	Tecniche di Rilevamento della Falsificazione . . . . .	10
1.6	Grafologia Forense . . . . .	10
1.6.1	Metodologie della Grafologia Forense . . . . .	11
1.6.2	Applicazioni della Grafologia Forense . . . . .	11
1.6.3	Limitazioni e Sfide . . . . .	11
1.7	Importanza delle Scienze Forensi nel Sistema Giudiziario . . . .	12
1.8	Metodologie e Tecnologie Avanzate . . . . .	12
1.8.1	Implicazioni Legali ed Etiche . . . . .	12
<b>2</b>	<b>Analisi Forense delle manoscritte</b>	<b>13</b>
2.1	Introduzione . . . . .	13
2.2	Descrizione del Problema . . . . .	13
2.3	Tratti Distintivi della Scrittura . . . . .	13

2.3.1	Forma delle Lettere . . . . .	14
2.3.2	Spaziatura . . . . .	14
2.3.3	Inclinazione . . . . .	14
2.3.4	Pressione . . . . .	14
2.4	Approcci di Classificazione . . . . .	15
2.4.1	Metodi Tradizionali . . . . .	15
2.4.2	Metodi Basati sull'Intelligenza Artificiale . . . . .	15
2.5	Dataset e Preprocessing . . . . .	15
<b>3</b>	<b>Lavoro svolto</b>	<b>16</b>
3.1	Introduzione . . . . .	16
3.2	Creazione del Dataset . . . . .	16
3.2.1	Creazione dell'interfaccia grafica per l'estrazione di caratteri da immagini di parole . . . . .	16
3.2.1.1	Procedimento . . . . .	16
3.2.2	Creazione del Dataset Finale . . . . .	18
3.3	Reti Neurali Convoluzionali (CNN) . . . . .	18
3.3.1	Implementazione dei Modelli . . . . .	19
3.3.2	Modelli Utilizzati . . . . .	20
3.3.3	Addestramento del Modello . . . . .	22
3.3.4	Valutazione del Modello . . . . .	22
3.3.5	Verso la Classificazione dei Soggetti . . . . .	22
3.4	Rete Siamese per la Classificazione dei Soggetti e Caratteri . . . . .	23
3.4.1	Architettura della Rete Siamese . . . . .	23
3.4.2	Funzione di Distanza e Perdita . . . . .	23
3.4.2.1	Contrastive Loss . . . . .	23
3.4.3	Vantaggi delle Reti Siamese per la Classificazione dei Caratteri . . . . .	24
3.4.4	Implementazione della Rete Siamese . . . . .	24
<b>4</b>	<b>Metriche di Valutazione</b>	<b>26</b>
4.1	Analisi delle Curve di Apprendimento . . . . .	27
4.2	Analisi della Matrice di Confusione . . . . .	28
<b>5</b>	<b>Risultati Sperimentali</b>	<b>30</b>
5.1	Experiment Details . . . . .	31
5.1.1	Experiment 1 . . . . .	31
5.1.2	Experiment 2 . . . . .	32
5.1.3	Experiment 3 . . . . .	32
5.1.4	Experiment 4 . . . . .	33
5.2	Comparazione con altri modelli pretrainati con ImageNet . . . . .	33

5.2.1	AlexNet . . . . .	34
5.2.1.1	Matrice di Confusione e Curva di Loss e Accuracy per AlexNet . . . . .	34
5.2.2	SqueezeNet . . . . .	35
5.2.2.1	Analisi della Matrice di Confusione e della Curva Loss e Accuracy per SqueezeNet . . . . .	35
5.2.3	ResNet . . . . .	36
5.2.3.1	Analisi della Matrice di Confusione e della Curva Loss e Accuracy per ResNet . . . . .	36
5.3	Riepilogo dei Risultati degli Esperimenti . . . . .	37
5.4	Riflessioni e Considerazioni sui Risultati Sperimentali . . . . .	37
5.5	Modello Siamese . . . . .	38
5.5.1	Curve di Loss train e Validation . . . . .	38
5.5.2	Analisi della Curva di Addestramento . . . . .	38
5.5.3	Analisi della Curva di Validazione . . . . .	38
5.5.4	Analisi Complessiva . . . . .	39
5.6	Differenza Di grafia Tra Soggetti . . . . .	41
5.6.1	Risultati . . . . .	41
<b>6</b>	<b>Conclusioni e supplementi futuri</b>	<b>44</b>
	<b>Bibliografia</b>	<b>46</b>

# Capitolo 1

## Introduzione alle scienze Forensi

### 1.1 Definizione e Origini delle Scienze Forensi

Le scienze forensi, spesso chiamate anche scienze criminalistiche, rappresentano l'applicazione delle varie discipline scientifiche alle questioni legali. Questo campo multidisciplinare si occupa della raccolta, dell'analisi e dell'interpretazione di prove fisiche, chimiche e biologiche per supportare le indagini criminali e i procedimenti giudiziari. Le origini delle scienze forensi possono essere fatte risalire all'antica Cina, dove i medici legali eseguivano autopsie per determinare la causa della morte. Tuttavia, il campo moderno delle scienze forensi è emerso nel XIX secolo, con lo sviluppo di nuove tecniche e tecnologie scientifiche.

### 1.2 Fasi delle Scienze Forensi

Le scienze forensi possono essere suddivise in diverse fasi, ciascuna delle quali è cruciale per garantire che le prove siano raccolte, analizzate e presentate in modo accurato, affidabile e ripetibile:

#### 1.2.1 Raccolta delle Prove

La fase iniziale consiste nella raccolta delle prove dalla scena del crimine. Gli investigatori forensi devono essere meticolosi per evitare la contaminazione

delle prove. Questo può includere la raccolta di campioni di DNA, impronte digitali, fibre, residui di polvere da sparo e altro ancora.

### 1.2.2 Analisi delle Prove

Una volta raccolte, le prove vengono analizzate nei laboratori forensi. Questa fase coinvolge l'uso di varie tecniche scientifiche e tecnologie avanzate per identificare e quantificare le evidenze fisiche e chimiche.

### 1.2.3 Interpretazione dei Risultati

Gli esperti forensi interpretano i risultati delle analisi per determinare il loro significato in relazione al caso investigato. Questo può includere la determinazione dell'identità di una vittima o di un sospettato, la ricostruzione della sequenza degli eventi e l'identificazione di sostanze chimiche.

### 1.2.4 Presentazione delle Prove

Infine, le prove vengono presentate in tribunale. Gli esperti forensi forniscono testimonianze come testimoni esperti, spiegando i risultati delle loro analisi e la loro rilevanza per il caso.

## 1.3 Principali Discipline delle Scienze Forensi

Le scienze forensi comprendono una vasta gamma di discipline, ognuna con il proprio insieme di metodologie e applicazioni specifiche. Tra le principali aree di studio e applicazione troviamo:



**Figura 1.1:** Discipline scienze forensi.



### **1.3.1 Analisi del DNA**

L'analisi del DNA è una delle tecniche più potenti e affidabili nelle scienze forensi. Viene utilizzata per identificare individui con elevata precisione attraverso il confronto di campioni di DNA trovati sulla scena del crimine con quelli di sospettati o database di DNA.

### **1.3.2 Tossicologia Forense**

Questa disciplina si occupa dello studio delle sostanze tossiche e della loro influenza sul corpo umano. La tossicologia forense è cruciale per determinare se droghe, alcol o altre sostanze chimiche hanno contribuito alla morte o al comportamento criminale.

### **1.3.3 Antropologia Forense**

Gli antropologi forensi analizzano i resti scheletrici per determinare l'identità delle vittime, nonché la causa e il modo della loro morte. Possono anche stimare l'età, il sesso, la statura e l'ascendenza etnica dei resti.

### **1.3.4 Balistica Forense**

La balistica forense si occupa dell'analisi delle armi da fuoco e dei proiettili. Gli esperti di balistica possono determinare la traiettoria di un proiettile, identificare l'arma utilizzata e collegare proiettili a scene del crimine specifiche.

## **1.4 Digital Forensics**

Un'area di crescente importanza nelle scienze forensi è la digital forensics, che riguarda l'identificazione, la raccolta, la conservazione e l'analisi delle prove digitali. Questo campo è diventato essenziale con l'aumento dei crimini informatici e l'uso diffuso della tecnologia nella vita quotidiana.

### **1.4.1 Tipi di Prove Digitali**

Le prove digitali possono includere e-mail, messaggi di testo, registri di chiamate, file di computer, dati di rete e informazioni memorizzate su dispositivi mobili. Gli esperti di digital forensics utilizzano software specializzati per recuperare e analizzare questi dati, anche quando sono stati cancellati o nascosti.



**Figura 1.2:** Digital Forensics.

### 1.4.2 Metodologie di Digital Forensics

Le metodologie utilizzate nella digital forensics includono l'acquisizione forense, dove viene fatta una copia esatta dei dati digitali per l'analisi, e l'analisi dei dati, che coinvolge l'esame dettagliato delle prove digitali per identificare informazioni rilevanti per l'indagine.

### 1.4.3 Importanza della Digital Forensics

La digital forensics è cruciale non solo per i casi di crimini informatici, come l'hacking e la frode informatica, ma anche per indagini su crimini tradizionali dove le prove digitali possono fornire indizi critici, come nei casi di omicidio, rapimento e terrorismo.

## 1.5 Digital Forgery

La falsificazione digitale è un'area di crescente preoccupazione nelle scienze forensi, poiché la tecnologia avanzata ha reso più facile la manipolazione delle informazioni digitali. La digital forgery comprende la creazione, l'alterazione o la manipolazione di documenti digitali, immagini, audio e video con l'intento di ingannare.

### 1.5.1 Tipi di Falsificazione Digitale

Le forme comuni di falsificazione digitale includono:

- **Documenti Falsi:** Creazione o modifica di documenti elettronici, come contratti, certificati e ricevute.
- **Manipolazione di Immagini:** Alterazione di fotografie digitali per creare prove false o ingannevoli.

- **Falsificazione di Audio e Video:** Creazione di audio e video falsi utilizzando tecniche come il deepfake, che utilizza l'intelligenza artificiale per generare contenuti realistici ma falsi.

### 1.5.2 Tecniche di Rilevamento della Falsificazione

Gli esperti forensi utilizzano varie tecniche per rilevare la falsificazione digitale:

- **Analisi dei Metadati:** Esame dei dati incorporati nei file digitali, come data di creazione, modifiche e software utilizzato.
- **Analisi Forense delle Immagini:** Utilizzo di algoritmi per identificare incongruenze nei pixel, compressione e caratteristiche di illuminazione.
- **Riconoscimento di Deepfake:** Tecniche di machine learning per rilevare video e audio falsi analizzando pattern non naturali e artefatti.
- **Confronto di Documenti:** Analisi comparativa di documenti digitali per identificare differenze sottili che potrebbero indicare una manipolazione.

## 1.6 Grafologia Forense



**Figura 1.3:** Analisi di un manoscritto.

La grafologia forense è una disciplina che si occupa dell'analisi della scrittura a mano per scopi legali. Sebbene la grafologia sia spesso associata allo studio della personalità, la grafologia forense si concentra sulla verifica dell'autenticità dei documenti e sulla identificazione degli autori di scritti anonimi.

### 1.6.1 Metodologie della Grafologia Forense

Gli esperti di grafologia forense utilizzano varie tecniche per analizzare le caratteristiche della scrittura a mano, tra cui:

- **Analisi delle Linee:** Osservazione della pressione, continuità e regolarità delle linee tracciate.
- **Forma delle Lettere:** Studio delle dimensioni, proporzioni e angolazioni delle lettere.
- **Spaziatura e Allineamento:** Valutazione della spaziatura tra le lettere, le parole e le righe, nonché l'allineamento orizzontale della scrittura.
- **Movimento e Velocità:** Analisi del movimento e della velocità con cui le lettere sono tracciate, che possono rivelare il ritmo naturale dello scrittore.

### 1.6.2 Applicazioni della Grafologia Forense

La grafologia forense è utilizzata in vari contesti legali, tra cui:

- **Verifica di Documenti:** Confermare l'autenticità di firme su contratti, testamenti e altri documenti legali.
- **Identificazione di Scrittori:** Determinare l'autore di lettere anonime o minatorie.
- **Risoluzione di Dispute:** Risolvere controversie legali che coinvolgono la validità di documenti scritti.

### 1.6.3 Limitazioni e Sfide

La grafologia forense, come molte altre tecniche forensi, presenta delle sfide. L'accuratezza delle analisi può essere influenzata dalla qualità dei campioni di scrittura, dalle condizioni fisiche dello scrittore e dalla possibilità di imitazioni intenzionali. Pertanto, è essenziale che le analisi siano condotte da esperti qualificati con una solida formazione e esperienza nel campo.

## 1.7 Importanza delle Scienze Forensi nel Sistema Giudiziario

Le scienze forensi svolgono un ruolo cruciale nel sistema giudiziario, fornendo prove scientifiche che possono corroborare o confutare le testimonianze e le dichiarazioni degli imputati. Le prove forensi sono spesso considerate oggettive e affidabili, il che le rende estremamente persuasive in tribunale. Tuttavia, la validità e l'affidabilità delle prove forensi dipendono dalla correttezza dei metodi utilizzati per la loro raccolta e analisi, nonché dalla competenza degli esperti forensi.

## 1.8 Metodologie e Tecnologie Avanzate

Negli ultimi decenni, il campo delle scienze forensi ha visto notevoli progressi tecnologici. Le tecniche avanzate di analisi del DNA, come la PCR (Polymerase Chain Reaction) e il sequenziamento del DNA di nuova generazione, hanno migliorato la precisione e la velocità delle identificazioni. La spettrometria di massa e la cromatografia sono diventate strumenti standard per l'analisi chimica, mentre le tecnologie di imaging avanzato, come la tomografia computerizzata (CT) e la risonanza magnetica (MRI), sono utilizzate per esaminare dettagliatamente i resti umani senza la necessità di dissezione. Oltre a queste tecniche negli ultimi anni con il progresso delle Intelligenze Artificiali (IA) nuove metodologie hanno rivoluzionato il campo delle scienze forensi. Reti neurali come le CNN, architetture GAN, tecniche di data mining e machine learning hanno aiutato enormemente la fase di raccolta dati fino a migliorarne anche l'accuratezza in alcuni casi. Purtroppo oltre a dare un aiuto hanno anche introdotto nuove sfide che potremmo riassumere in Digital Forgery.

### 1.8.1 Implicazioni Legali ed Etiche

La falsificazione digitale pone seri problemi legali ed etici. La difficoltà di distinguere tra contenuti autentici e falsi può compromettere l'integrità delle indagini legali e dei procedimenti giudiziari. È fondamentale che gli esperti forensi siano ben addestrati nelle tecniche di rilevamento e che esistano regolamenti rigidi per l'ammissione delle prove digitali in tribunale.

## Capitolo 2

# Analisi Forense delle manoscritture

### 2.1 Introduzione

La grafologia forense è una disciplina che analizza la scrittura a mano per identificare gli autori di manoscritti. Questo campo trova applicazione in ambito criminale, legale e psicologico, dove l'analisi dei tratti distintivi della scrittura può fornire prove determinanti. La scrittura a mano è un comportamento unico per ciascun individuo, il che la rende uno strumento prezioso per l'identificazione forense.

### 2.2 Descrizione del Problema

Il principale problema nella classificazione di individui sulla base della grafologia forense è l'identificazione e l'analisi accurata dei tratti distintivi della scrittura. Questi tratti possono variare ampiamente tra individui diversi e persino per lo stesso individuo in circostanze diverse. Tuttavia, alcuni tratti distintivi della scrittura possono essere utilizzati per creare un profilo unico dell'autore.

### 2.3 Tratti Distintivi della Scrittura

I tratti distintivi della scrittura possono essere suddivisi in diverse categorie, ciascuna delle quali fornisce informazioni uniche sull'autore del manoscritto. Le principali categorie includono:

### 2.3.1 Forma delle Lettere

La forma delle lettere è uno dei tratti più visibili e distintivi della scrittura. Questo include:

- **Dimensione delle Lettere:** la grandezza delle lettere può variare notevolmente tra scrittori.
- **Proporzioni:** il rapporto tra l'altezza e la larghezza delle lettere.
- **Curvatura:** il grado di curvatura delle lettere, ad esempio la differenza tra lettere arrotondate e angolari.
- **Decorazioni:** l'uso di tratti aggiuntivi o decorazioni sulle lettere.

### 2.3.2 Spaziatura

La spaziatura si riferisce alla distanza tra lettere, parole e righe, e include:

- **Spaziatura tra Lettere:** la distanza uniforme o variabile tra le lettere di una parola.
- **Spaziatura tra Parole:** la quantità di spazio lasciato tra le parole.
- **Interlinea:** la distanza tra le righe di testo.

### 2.3.3 Inclinazione

L'inclinazione della scrittura può fornire informazioni importanti e include:

- **Inclinazione delle Lettere:** l'angolo con cui le lettere sono inclinate rispetto alla verticale.
- **Allineamento delle Righe:** la tendenza delle righe di testo a salire, scendere o essere orizzontali.

### 2.3.4 Pressione

La pressione esercitata sulla carta durante la scrittura è un altro tratto distintivo, e include:

- **Intensità della Pressione:** variazione nella forza applicata sulla penna.
- **Variazione della Pressione:** cambiamenti nella pressione all'interno di una singola lettera o parola.

## 2.4 Approcci di Classificazione

La classificazione degli individui basata sui tratti distintivi della scrittura può essere effettuata utilizzando metodi tradizionali e moderni approcci basati sull'intelligenza artificiale.

### 2.4.1 Metodi Tradizionali

I metodi tradizionali coinvolgono l'analisi manuale dei tratti distintivi da parte di esperti grafologi. Questi metodi includono:

- Confronto visivo delle caratteristiche della scrittura.
- Analisi delle differenze e delle somiglianze tra campioni di scrittura.

### 2.4.2 Metodi Basati sull'Intelligenza Artificiale

I moderni approcci basati sull'intelligenza artificiale offrono strumenti avanzati per l'analisi automatica dei tratti distintivi della scrittura. Questi metodi includono:

- **Reti Neurali Convoluzionali (CNN):** utilizzate per l'analisi delle immagini della scrittura a mano.
- **Support Vector Machines (SVM):** impiegate per la classificazione basata sui tratti distintivi.
- **Algoritmi di Apprendimento Supervisionato:** come k-nearest neighbors (KNN) e foreste casuali (random forests).

## 2.5 Dataset e Preprocessing

La costruzione di un dataset accurato è cruciale per l'analisi dei tratti distintivi della scrittura. Il preprocessing dei dati include:

- **Digitalizzazione:** conversione dei manoscritti in immagini digitali ad alta risoluzione.
- **Pulizia:** rimozione di rumore e imperfezioni dalle immagini.
- **Segmentazione:** suddivisione delle immagini in singole lettere o parole per l'analisi dettagliata.
- **Estrazione delle Caratteristiche:** identificazione e misurazione dei tratti distintivi rilevanti.



# Capitolo 3

## Lavoro svolto

### 3.1 Introduzione

Per affrontare il problema della classificazione dei caratteri, inizialmente dovevo costruire un dataset di immagini di caratteri provenienti tutti da soggetti differenti. La mia creazione del dataset parte dall'utilizzo di un dataset di immagini di parole: CVL. CVL è un database pubblico utilizzato per il riconoscimento degli scrittori, l'identificazione degli scrittori e il word spotting. Il database è composto da 7 diversi testi manoscritti (1 in tedesco e 6 in inglese). In totale, 310 scrittori hanno partecipato al dataset. Dato che le immagini erano separate già in parole, ho dovuto creare uno script che mi permettesse di estrarre i caratteri da esse.

### 3.2 Creazione del Dataset

Il primo passo della creazione del dataset è stato proprio la creazione dello script per l'estrazione dei caratteri.

#### 3.2.1 Creazione dell'interfaccia grafica per l'estrazione di caratteri da immagini di parole

##### 3.2.1.1 Procedimento

**Selezione delle Directory** Iniziamo selezionando la directory di input contenente le immagini che vogliamo processare e la directory di output dove verranno salvati i risultati. Questo può essere fatto tramite un'interfaccia grafica semplice che ci permette di navigare tra le cartelle del nostro computer.



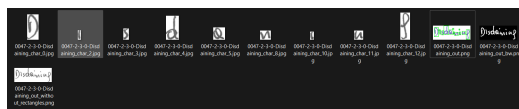
**Figura 3.1:** Esempio di Selezione Directory.

**Caricamento delle Immagini** Il programma supporta vari formati di immagine come TIFF, JPEG e PNG. Una volta selezionate le immagini, il programma le carica una alla volta per il processamento.

**Preprocessing** Prima di processare le immagini, vi è una fase di preprocessing fatta per adattare l'immagine all'estrazione. Questa fase si compone di diversi passi:

- **Conversione in scala di grigi:** utilizziamo la libreria `cv2` per semplificare l'immagine, riducendo le informazioni di colore e permettendoci di concentrarci solo sulle intensità luminose.
- **Soglia adattiva:** trasformiamo l'immagine in bianco e nero, distinguendo chiaramente i caratteri dallo sfondo. La soglia adattiva prende in considerazione solo piccole regioni dell'immagine, adattando la soglia a cambiamenti locali e di illuminazione.
- **Riduzione del rumore:** miglioriamo la qualità dell'immagine tramite un'operazione di denoising con la funzione `cv2.fastNlMeansDenoising`. Questo passaggio aiuta a eliminare piccoli artefatti che potrebbero interferire con il riconoscimento dei caratteri.
- **Operazioni morfologiche:** per separare i caratteri attaccati, effettuiamo operazioni di erosione e dilatazione, che aiutano a pulire e a migliorare la separazione tra i caratteri.

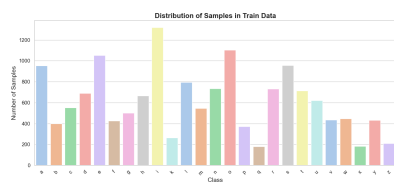
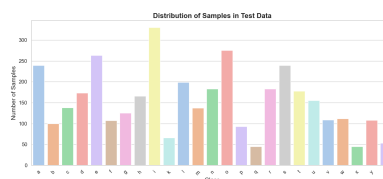
**Rilevazione dei Contorni** Il procedimento di rilevazione dei contorni dei caratteri utilizza una funzione di `cv2` che permette di rilevare i contorni da sinistra verso destra. Per ogni contorno rilevato viene calcolato un rettangolo di delimitazione. I caratteri vengono poi estratti e salvati come immagini separate. Infine, i rettangoli vengono disegnati sull'immagine originale per visualizzare i contorni rilevati.

**Figura 3.2:** Esempio output script

In alcuni casi, i caratteri, data la troppa vicinanza, vengono rilevati “attaccati”. Per ovviare a ciò, nello script è stato inserito un altro metodo di elaborazione che permette di estrarre il carattere a mano.

### 3.2.2 Creazione del Dataset Finale

Una volta che tutti i caratteri sono stati estratti, ho proceduto alla creazione del dataset. Per ogni soggetto da cui estraevo i caratteri, ho creato una cartella per ogni lettera dalla a alla z, etichettando così ogni immagine dei caratteri. Facendo ciò, ho creato tante classi quanti sono i caratteri per procedere successivamente alla creazione di un modello che riuscisse a classificare le immagini di caratteri correttamente.

**Figura 3.3:** Distribuzione Training set**Figura 3.4:** Distribuzione Test set**Figura 3.5:** Divisione Dataset

## 3.3 Reti Neurali Convolutionali (CNN)

Un altro approccio usato è stato quello delle reti neurali CNN. Dato che il dataset creato si basa su immagini di caratteri, l'utilizzo di una CNN può dare un ottimo aiuto nella classificazione di esse. In particolare, ho sfruttato diversi modelli pre-allenati con ImageNet conosciuti come ResNet, AlexNet, e SqueezeNet.

### 3.3.1 Implementazione dei Modelli

L'implementazione dei modelli è gestita dalla classe `GenericModel` che eredita dalla classe `LetterClassifier`. Questi modelli sono stati configurati specificamente per classificare immagini di caratteri adattandosi a un dataset con 25 classi, una per ogni lettera dell'alfabeto.

**Fine-Tuning** Il fine-tuning è una tecnica cruciale per adattare un modello pre-addestrato a un nuovo compito specifico. In questo contesto, utilizziamo modelli ResNet, AlexNet e SqueezeNet pre-addestrati su ImageNet e ne modifichiamo l'ultimo layer per adattarli alla nostra specifica classificazione dei caratteri.

**Trasformazioni** Le trasformazioni applicate ai dati sono fondamentali per migliorare la capacità del modello di generalizzare su dati non visti. Le trasformazioni includono tecniche di augmentation come il cambiamento di luminosità, la rotazione casuale e il ridimensionamento.

**Ottimizzatore SGD** L'ottimizzatore Stochastic Gradient Descent (SGD) è utilizzato per aggiornare i pesi della rete. L'SGD aggiorna i pesi in direzione opposta al gradiente della funzione di costo con l'obiettivo di minimizzare l'errore. È definito dai parametri di apprendimento (learning rate), momentum e weight decay:

- **Learning Rate:** determina la dimensione dei passi compiuti nell'aggiornamento dei pesi.
- **Momentum:** aiuta ad accelerare l'ottimizzazione e ridurre le oscillazioni.
- **Weight Decay:** aggiunge una penalizzazione sui pesi per prevenire overfitting.

**Funzione di Perdita CrossEntropyLoss** La funzione di perdita `CrossEntropyLoss` è utilizzata per misurare la differenza tra le predizioni del modello e le etichette vere. Questa funzione è particolarmente adatta per problemi di classificazione multi-classe.

**Early Stopping** L'early stopping è una tecnica di regolarizzazione che interrompe l'addestramento quando il modello smette di migliorare sulle performance di validazione, prevenendo così l'overfitting. Nella nostra imple-

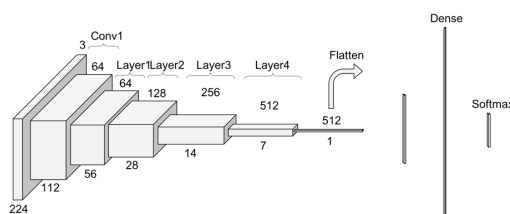
mentazione viene monitorata la perdita di validazione e, se non migliora per un certo numero di epoche (*patience*), l'addestramento viene interrotto.

### 3.3.2 Modelli Utilizzati

**ResNet** ResNet o Residual Network è una famiglia di reti neurali convoluzionali (CNN) sviluppata da Kaiming He e colleghi presso Microsoft Research presentata per la prima volta nel 2015. La ResNet ha introdotto una nuova architettura che ha permesso di costruire reti molto più profonde rispetto a quelle precedentemente possibili, risolvendo problemi di addestramento che affliggevano le reti profonde tradizionali.

La chiave del successo di ResNet è l'introduzione dei residual blocks. In una rete tradizionale, l'output di un layer viene passato come input al layer successivo. In ResNet, l'output di un layer viene sommato direttamente con l'input originale tramite una connessione di salto (skip connection). Questo permette di formare una "scorciatoia" per il flusso delle informazioni.

- $x$  è l'input del blocco.
- $F(xWi)$  è la funzione residua rappresentata dai layer convoluzionali.
- $y$  è l'output del blocco.



**Figura 3.6:** Architettura ResNet

## Vantaggi delle Residual Connections

**Facilità di Addestramento** Le connessioni di salto aiutano a mitigare il problema del vanishing gradient permettendo ai gradienti di fluire direttamente attraverso queste connessioni. Questo rende l’addestramento delle reti profonde più stabile e veloce.

**Prevenzione della Degradazione** Aggiungere più layer a una rete non sempre migliora le prestazioni; infatti, può peggiorarle se i nuovi layer non riescono a imparare le giuste rappresentazioni. Le residual connections aiutano a evitare che le prestazioni della rete si degradino man mano che la rete diventa più profonda.

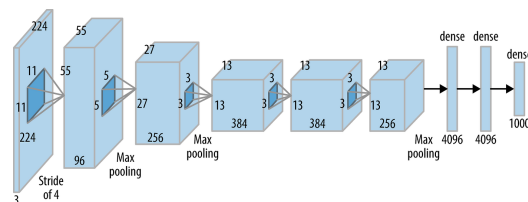
**Modularità** I residual blocks possono essere facilmente aggiunti a una rete esistente, rendendo ResNet molto flessibile e adattabile a diverse architetture.

**Architetture di ResNet** ResNet è disponibile in diverse varianti che differiscono per il numero di layer:

- ResNet-18: 18 layer
- ResNet-34: 34 layer
- ResNet-50: 50 layer
- ResNet-101: 101 layer
- ResNet-152: 152 layer

Le varianti con più layer offrono una maggiore capacità di rappresentazione ma richiedono anche più potenza computazionale e dati per l'addestramento. In particolare, la variante utilizzata da me è stata ResNet-18.

**AlexNet** AlexNet è una rete neurale convoluzionale (CNN) sviluppata da Alex Krizhevsky, Ilya Sutskever e Geoffrey Hinton. È stata introdotta nel 2012 e ha ottenuto un notevole successo nella competizione ImageNet. AlexNet ha introdotto innovazioni come l'uso di ReLU (Rectified Linear Unit) come funzione di attivazione e tecniche di dropout per prevenire l'overfitting.



**Figura 3.7:** Architettura AlexNet

**Caratteristiche di AlexNet** AlexNet è composta da 8 layer, di cui 5 sono layer convoluzionali e 3 sono layer completamente connessi. Utilizza max-pooling per ridurre la dimensionalità spaziale e ReLU per accelerare la convergenza durante l'addestramento.

**SqueezeNet** SqueezeNet è una rete neurale convoluzionale (CNN) sviluppata da Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally e Kurt Keutzer. È stata introdotta nel 2016 ed è nota per la sua efficienza in termini di parametri, riuscendo a mantenere un'accuratezza simile ad AlexNet ma con molti meno parametri.

**Caratteristiche di SqueezeNet** SqueezeNet introduce i Fire Modules, che combinano convoluzioni 1x1 e 3x3 per ridurre il numero di parametri mantenendo buone capacità di rappresentazione. Questo permette a SqueezeNet di essere molto più leggera rispetto ad altre reti di dimensioni simili.

### 3.3.3 Addestramento del Modello

L'addestramento del modello è gestito dalla funzione `train_model` che utilizza un ottimizzatore SGD e una funzione di perdita `CrossEntropyLoss`. Viene implementato anche un meccanismo di early stopping per evitare l'overfitting.

### 3.3.4 Valutazione del Modello

La valutazione delle performance del modello è effettuata tramite curve di loss e train e confusion matrix espletati in 4.

### 3.3.5 Verso la Classificazione dei Soggetti

Il prossimo passo verso la classificazione dei soggetti è l'individuazione di una distanza sulla base delle caratteristiche dei caratteri scritti dagli individui. Ogni individuo ha una grafia propria, pertanto è possibile sfruttare queste caratteristiche univoche per ogni soggetto per stilare questa distanza.

## 3.4 Rete Siamese per la Classificazione dei Soggetti e Caratteri

Una rete siamese è un'architettura di rete neurale che utilizza due sottoreti identiche con la stessa configurazione e pesi condivisi per confrontare coppie di input. È particolarmente efficace per il riconoscimento di somiglianze tra coppie di immagini [1].

### 3.4.1 Architettura della Rete Siamese

Una rete siamese è composta da:

- **Sottoreti Gemelle:** Due CNN identiche per estrarre caratteristiche dalle immagini di input.
- **Strato di Similarità:** Confronta i vettori di caratteristiche estratti utilizzando una funzione di distanza.
- **Funzione di Perdita:** Penalizza la rete se le immagini simili sono lontane e le immagini dissimili sono vicine.

Le immagini dei caratteri vengono organizzate in coppie etichettate come simili (stesso carattere, stesso autore) o dissimili (caratteri diversi o autori diversi).

### 3.4.2 Funzione di Distanza e Perdita

#### 3.4.2.1 Contrastive Loss

La **contrastive loss** è una funzione di perdita utilizzata per apprendere una rappresentazione delle distanze tra coppie di input. È particolarmente utile nei compiti di verifica, dove si vuole distinguere tra coppie di elementi simili e dissimili.

La formula della contrastive loss è la seguente:

$$L(y, D) = (1 - y) \cdot \frac{1}{2} D^2 + y \cdot \frac{1}{2} \max(0, m - D)^2$$

dove:

- $y$  è l'etichetta binaria che indica se la coppia è simile ( $y = 0$ ) o dissimile ( $y = 1$ ).
- $D$  è la distanza euclidea tra i vettori di caratteristiche delle immagini di input.



- $m$  è il margine che definisce la distanza alla quale le coppie dissimili non contribuiscono più alla perdita.

L'obiettivo della contrastive loss è minimizzare la distanza tra coppie simili e massimizzare la distanza tra coppie dissimili, fino al margine  $m$ .

### 3.4.3 Vantaggi delle Reti Siamese per la Classificazione dei Caratteri

Le reti siamese offrono diversi vantaggi per la classificazione dei caratteri:

- **Apprendimento di Somiglianze:** Le reti siamese sono progettate per apprendere direttamente le somiglianze tra le immagini, rendendole adatte per la classificazione dei caratteri scritti a mano.
- **Generalizzazione:** Grazie alla condivisione dei pesi, la rete può generalizzare meglio su dati non visti.
- **Robustezza alle Variazioni:** Le reti siamese possono gestire variazioni nell'orientamento, scala e stile di scrittura dei caratteri.

### 3.4.4 Implementazione della Rete Siamese

Per l'implementazione della rete siamese, ho seguito i seguenti passi:

**Creazione del Dataset** Ho creato una classe `PairDataset` che eredita da `Dataset` di PyTorch. Questa classe si occupa di gestire la generazione delle coppie di immagini e le etichette. La classe raccoglie tutti i percorsi delle immagini e le relative etichette, organizza le immagini per soggetto e carattere, e genera coppie di immagini etichettate come simili o dissimili.



**Figura 3.8:** Coppie simili e dissimili

**Rete di Embedding** Ho definito una classe `ResNetEmbedding` che utilizza una rete `ResNet18` pre-addestrata per estrarre vettori di caratteristiche dalle immagini. L'ultimo layer completamente connesso della `ResNet` è stato modificato per produrre un vettore di dimensione 128, che rappresenta l'embedding dell'immagine.

**Funzione di Perdita** La `ContrastiveLoss` è stata implementata come una sottoclasse di `torch.nn.Module`. Questa funzione di perdita calcola la distanza tra i vettori di embedding delle coppie di immagini e applica la formula della contrastive loss per penalizzare la rete se le coppie simili sono lontane o le coppie dissimili sono vicine.

**Task di Addestramento** La classe `SiameseNetworkTask` è stata definita per gestire l'intero processo di addestramento. Questa classe utilizza `PyTorch Lightning` per semplificare la gestione dell'addestramento e della validazione. La classe include la configurazione degli ottimizzatori, la definizione del passo di addestramento (`training_step`) e del passo di validazione (`validation_step`). Durante l'addestramento, gli embeddings delle immagini vengono calcolati utilizzando la rete di embedding e la perdita viene calcolata usando la `ContrastiveLoss`.

**Stabilimento delle distanze** una volta allenato il modello, il passo successivo è stato stabilire una distanza tra manoscritti di diversi soggetti. per fare ciò ho sfruttato le caratteristiche estratte dal mio modello e la contrastive loss spiegata precedentemente. in particolare ho estratto un vettore di feature da entrambi i soggetti, Questi vettori vengono mediati per ottenere un vettore medio per ogni classe. Infine sarà la contrastive loss a calcolare una distanza tra questi due vettori ed è definita in modo tale che più è vicina a 2 più i soggetti sono diversi più è vicina allo 0 più sono simili, viene infine calcolata una distanza media complessiva della loss per tutte le classi. Per l'estrazione delle lettere e la classificazione di esse sono stati sfruttati i modelli precedentemente descritti in 3.3 e lo script descritto in 3.2.1.1

# Capitolo 4

## Metriche di Valutazione

In questo capitolo, descriviamo le metriche e le metodologie utilizzate per valutare le prestazioni del classificatore di lettere. Le metriche principali impiegate includono l'accuratezza e la perdita durante l'addestramento e la validazione. Queste metriche sono essenziali per comprendere le prestazioni complessive del sistema e il comportamento di convergenza durante il processo di addestramento. Inoltre, metriche più dettagliate come la *precision*, *recall*, e *F1-score* saranno utilizzate per valutare l'accuratezza e la completezza della classificazione [2, 3, 4]

La metrica dell'accuratezza fornisce una misura diretta della proporzione di istanze correttamente classificate rispetto al totale delle istanze, offrendo una visione generale delle prestazioni del modello. La perdita fornisce informazioni sulla capacità del modello di minimizzare gli errori nel tempo ed è cruciale per diagnosticare la convergenza e identificare potenziali problemi di overfitting o underfitting [5] [6]. Ad esempio, se un modello mostra una perdita elevata sia nella fase di addestramento che in quella di test, indica un alto bias, un errore sistematico causato da un modello troppo semplice che tenta di adattarsi a schemi complessi (underfitting). Al contrario, se un modello mostra errori bassi nella fase di addestramento ma perdite elevate nella fase di test, suggerisce un'alta varianza, dove il modello si adatta troppo strettamente al rumore nei dati di addestramento, perdendo la capacità di generalizzare su nuovi dati non visti (overfitting).

Tuttavia, l'accuratezza da sola potrebbe non essere sufficiente, specialmente in scenari dove è presente uno squilibrio tra le classi. Per affrontare questo problema, incorporiamo le metriche di precisione e richiamo. La precisione, il rapporto tra le previsioni positive vere e il totale delle previsioni positive, indica l'accuratezza delle previsioni positive. Il richiamo, il rappor-

to tra le previsioni positive vere e il totale dei positivi effettivi, misura la capacità del classificatore di catturare tutte le istanze rilevanti. L’F1-score, la media armonica della precisione e del richiamo, fornisce una singola metrica che bilancia entrambe le preoccupazioni, offrendo una valutazione più completa delle prestazioni del modello.

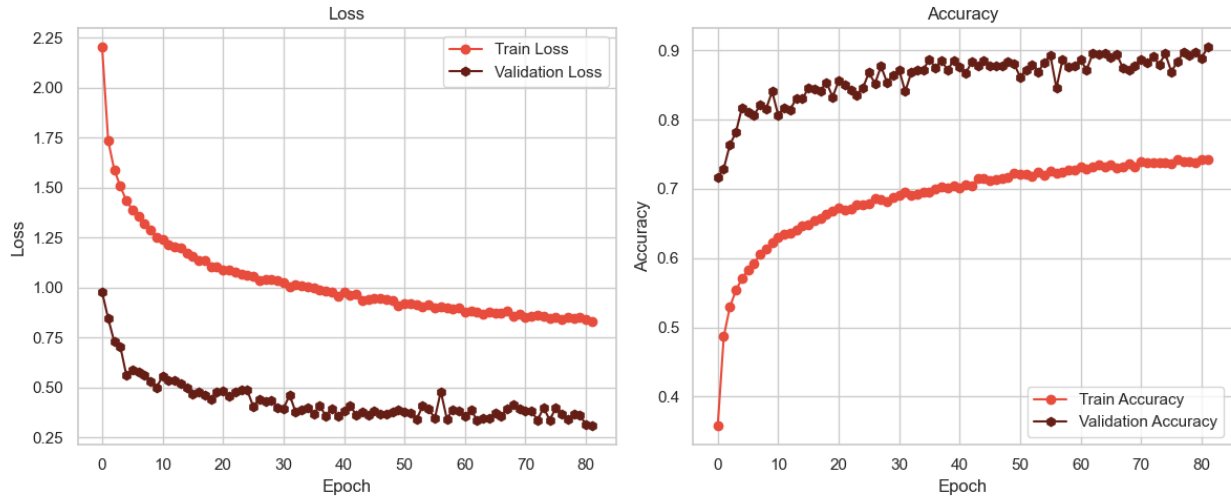
I valori di perdita e accuratezza si riferiscono al miglior modello ottenuto su circa 100 epoche. Un meccanismo di early stopping interrompe l’addestramento se non vi sono miglioramenti nel modello per 25 epoche consecutive. Per il miglior modello, le metriche di performance sono riportate nella sezione 4.1.

**Tabella 4.1:** riassunto delle performance Classificatore di caratteri, Best model

Metric	Value
Train Loss	0.831
Train Accuracy	0.743
Validation Loss	0.307
Validation Accuracy	0.905
Test Loss	0.298
Test Accuracy	0.913
Precision	0.901
Recall	0.915
F1 Score	0.906

## 4.1 Analisi delle Curve di Apprendimento

Le curve di apprendimento sono cruciali per diagnosticare il processo di addestramento del classificatore. Tracciando l’accuratezza e la perdita nel corso delle epoche sia per i dataset di addestramento che di validazione, possiamo osservare il comportamento di apprendimento del modello e identificare problemi come l’overfitting o l’underfitting. Una curva di apprendimento ben comportata mostrerà un miglioramento costante delle prestazioni sul set di addestramento, mentre le prestazioni sul set di validazione convergeranno a un livello simile, indicando che il modello generalizza bene sui dati non visti.



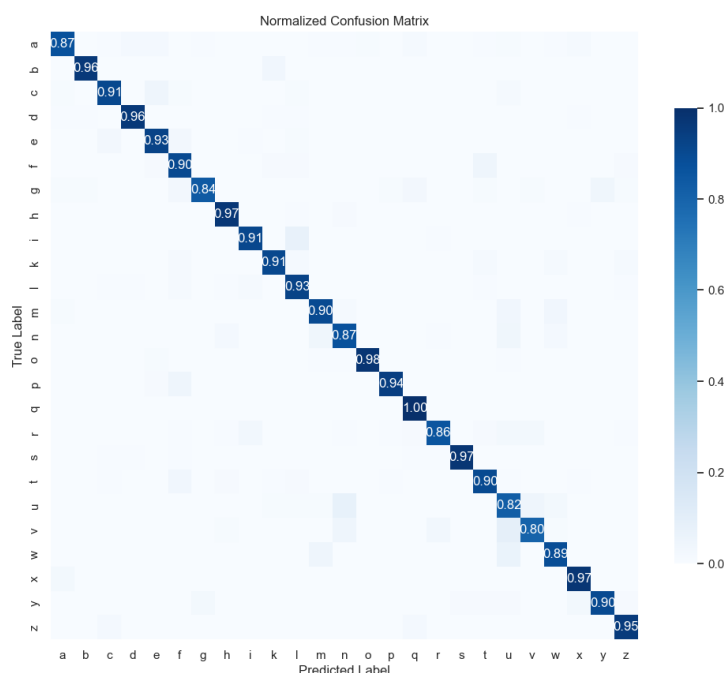
**Figura 4.1:** Curve di apprendimento per il Classificatore di caratteri, Best model.

È interessante notare che le prestazioni sul set di validazione superano quelle sui dati di addestramento. Questo fenomeno può essere attribuito alla maggiore complessità e varianza nei dati di addestramento rispetto ai dati di validazione, che sono più semplici e con minore varianza. Di conseguenza, il modello dimostra una forte capacità di generalizzazione anche su dati che non ha mai incontrato prima.

## 4.2 Analisi della Matrice di Confusione

Per ottenere una comprensione più approfondita della capacità del classificatore di distinguere tra le classi di lettere sono state analizzate le matrici di confusione. Una matrice di confusione fornisce un'analisi dettagliata delle prestazioni del classificatore mostrando il conteggio delle previsioni corrette (vere positive e vere negative) e degli errori (falsi positivi e falsi negativi) per ogni classe. Questa analisi aiuta a identificare aree specifiche in cui il modello può avere difficoltà, come confondere una classe con un'altra, e fornisce indicazioni per ulteriori miglioramenti del modello [7].

ecco un esempio della matrice di confusione effettivamente prodotta dal modello classificatore di Caratteri:



**Figura 4.2:** Matrice di Confusione per il Classificatore di caratteri, Best model.

Il classificatore dimostra prestazioni robuste nel prevedere correttamente ciascuna classe, come evidenziato dall'alta percentuale di veri positivi in tutte le categorie: circa il 90% in quasi tutte le classi con il valore minimo del 80%. Ciò indica che il modello è generalmente efficace nel distinguere tra le diverse lettere.

Tuttavia, esaminando più da vicino le previsioni errate, emergono alcuni pattern che meritano attenzione per ulteriori ottimizzazioni. Il classificatore mostra una tendenza a confondere le classi "g, q, v, u,". Questi errori suggeriscono che il modello potrebbe avere difficoltà a distinguere tra le lettere che coinvolgono forme o caratteristiche simili nella scrittura della lettera.

Per affrontare queste sfide, i futuri miglioramenti del classificatore potrebbero concentrarsi sul perfezionamento del processo di estrazione delle caratteristiche per catturare caratteristiche più distintive di ciascuna lettera, e magari ingrandimento del dataset con maggiori immagini per quest'ultime classi.

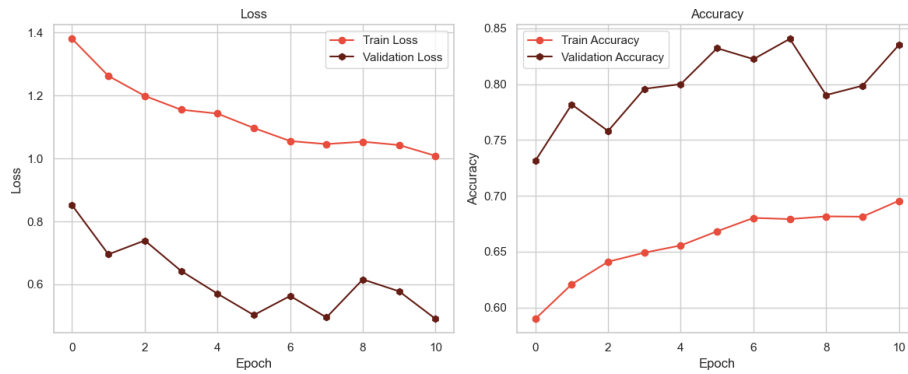
## Capitolo 5

# Risultati Sperimentali

Senza una valutazione completa del modello, non è possibile valutare accuratamente le sue prestazioni né prendere decisioni informate riguardo ai miglioramenti e alle implementazioni. Sono stati condotti diversi esperimenti per identificare i parametri ottimali, portando alla selezione di quelli che hanno dimostrato prestazioni superiori. Sono stati testati vari iperparametri e architetture di modelli per identificare la configurazione ottimale per questo compito. Gli esperimenti includono diverse impostazioni per il learning rate e il weight decay. Per ragioni di concisione e rilevanza, verranno discussi in dettaglio solo gli esperimenti più interessanti dal punto di vista delle prestazioni. Tuttavia, un riepilogo dettagliato di tutti gli esperimenti condotti, inclusi i loro risultati, è fornito nella Tabella 5.8.

## 5.1 Experiment Details

### 5.1.1 Experiment 1



**Figura 5.1:** Curve di Accuracy e Loss per l'esperimento 1.

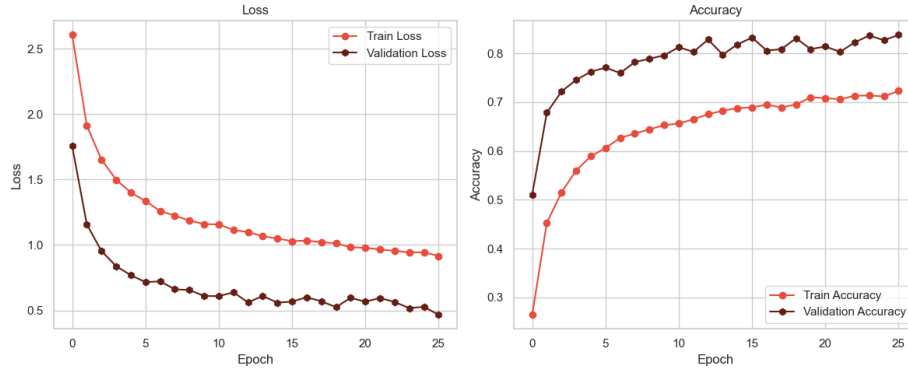
nell'esperimento 1, il learning rate era impostato a 0.01, con momentum del 0.9 e weight decay del 0.001. questa configurazione era impostata come la base iniziale del modello.

I risultati dimostrano un accuracy e loss in netto miglioramento ma il modello non è riuscito a raggiungere la convergenza.

Può essere già visto nel primo esperimento la capacità del modello di generalizzare data l'elevata accuracy nel validation del modello rispetto al training.



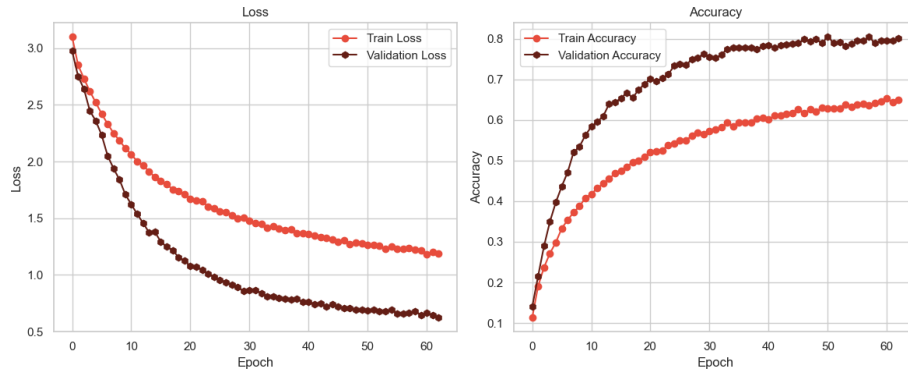
### 5.1.2 Experiment 2



**Figura 5.2:** Curve di Accuracy e Loss per l'esperimento 2.

Per il secondo esperimento ho abbassato il learning rate a 0.001 cercando di ottenere una migliore convergenza del modello. questo ha portato un leggero miglioramento nel training e nel validation che suggerisce che il modello si allena meglio con questo tipo di learning rate

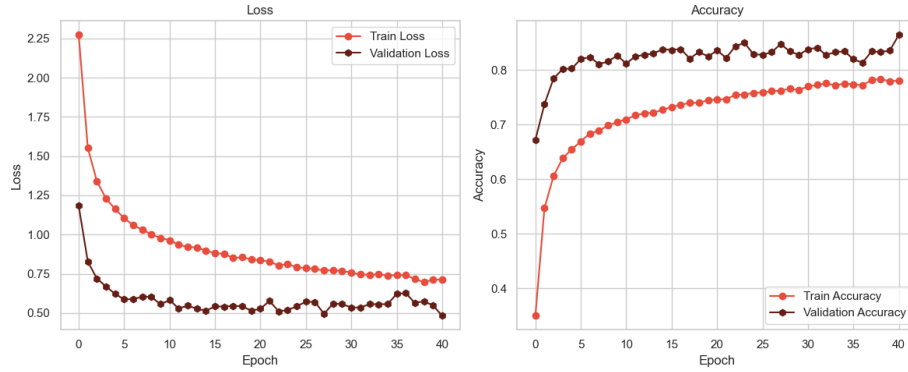
### 5.1.3 Experiment 3



**Figura 5.3:** Curve di Accuracy e Loss per l'esperimento 3.

nel terzo esperimento ho cercato di abbassare ulteriormente il learning rate a 0.0001 per vedere se magari riuscisse a migliorare ulteriormente nel validation ottenendo una convergenza più stabile. Il risultato però è stato un abbassamento delle performance dato dal rallentamento eccessivo nell'allenamento del modello, pertanto il learning rate migliore era proprio 0.001.

### 5.1.4 Experiment 4



**Figura 5.4:** Curve di Accuracy e Loss per l'esperimento 4.

il quarto e ultimo esperimento è stato quello definito, il learning rate lo ho riportato a 0.001 ciò che ho cambiato è stato il weight decay per cercare di ottenere un minimo di convergenza in più senza esagerare come nel caso del esperimento 3, ho notato un netto miglioramento sia per quanto riguarda il training che per il validation.

tutti gli esperimenti sono stati testati con il modello Resnet.

## 5.2 Comparazione con altri modelli pretrainati con ImageNet

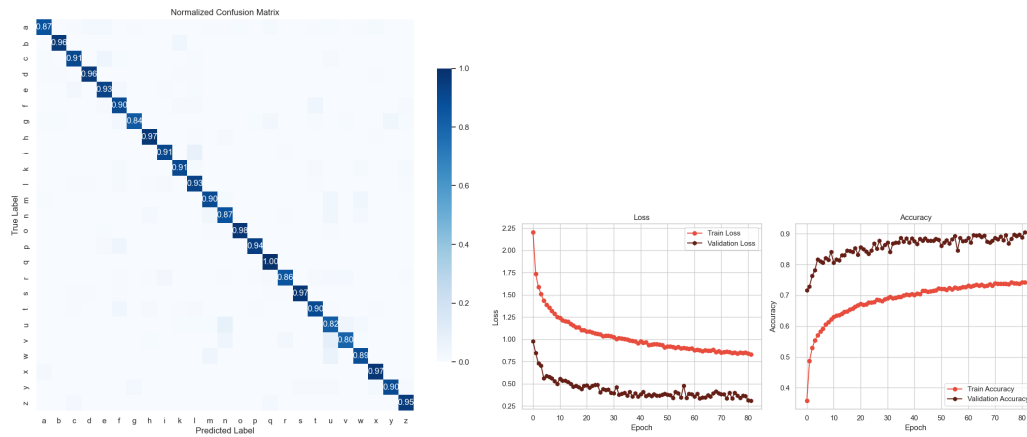
Per fornire una valutazione completa delle prestazioni del modello proposto, è stato condotto un confronto con diversi modelli affermati e noti nel campo della tecnologia all'avanguardia. I modelli presi in considerazione sono stati ResNet, AlexNet, e SqueezeNet. Ogni modello è stato valutato sia prima che dopo il fine-tuning, una procedura che permette al modello di adattarsi più efficacemente alle specificità del nuovo compito mantenendo i vantaggi dell'apprendimento trasferito dai pesi pre-addestrati [8].

Questa sezione presenta i risultati dei confronti menzionati, utilizzando matrici di confusione e curve di loss e accuracy. Viene analizzato il comportamento dei modelli e vengono fatte considerazioni utili.

### 5.2.1 AlexNet

Il modello AlexNet è stato valutato sullo stesso dataset degli altri modelli; ogni modello è stato addestrato per 100 epoche con gli stessi parametri dell'esperimento 4. Il modello ha 57.016.131 parametri addestrabili. Le prestazioni risultanti sono illustrate nelle figure seguenti.

#### 5.2.1.1 Matrice di Confusione e Curva di Loss e Accuracy per AlexNet



(a) Matrice di Confusione per AlexNet. (b) Curve di Loss e Accuracy Train e Valid test.

**Figura 5.5:** Analisi delle performance di AlexNet sul dataset dei caratteri.

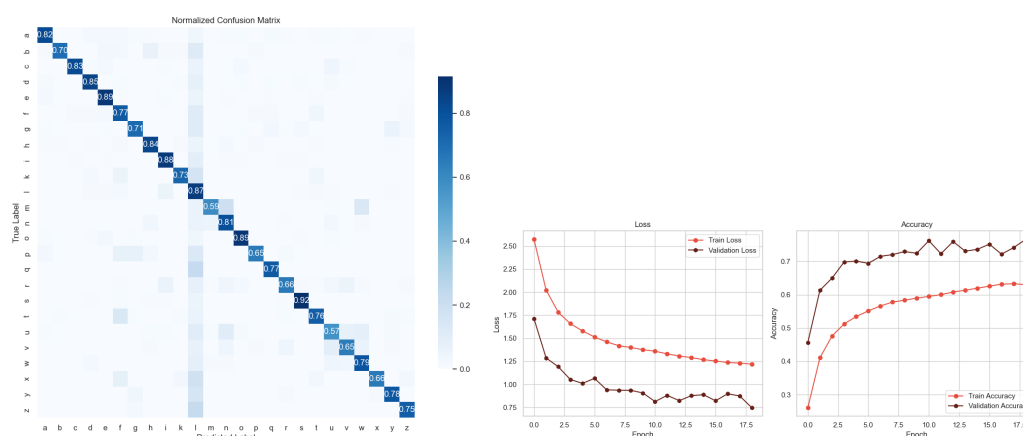
La matrice di confusione in Figura 5.5a rivela che AlexNet tende a classificare correttamente quasi tutte le classi, tanto che è stato preso come best model, le uniche classi in cui fa predizioni erranee sono le stesse in cui si parla nel capitolo 4

Per l’AlexNet non fine-tuned, la perdita di test era 3.335 e l’accuratezza di test era 0.036. Dopo il fine-tuning, il modello ha mostrato metriche migliorate: una precisione di 0.901, un recall di 0.915 e un F1 score di 0.906.”

## 5.2.2 SqueezeNet

Anche SqueezeNet, noto per la sua architettura leggera, è stato valutato. Il modello ha 736.963 parametri addestrabili.

### 5.2.2.1 Analisi della Matrice di Confusione e della Curva Loss e Accuracy per SqueezeNet



(a) Matrice di Confusione per SqueezeNet. (b) Curve di Loss e Accuracy Train e Valid test SqueezeNet.

**Figura 5.6:** Analisi delle performance di SqueezeNet sul dataset delle immagini di caratteri.

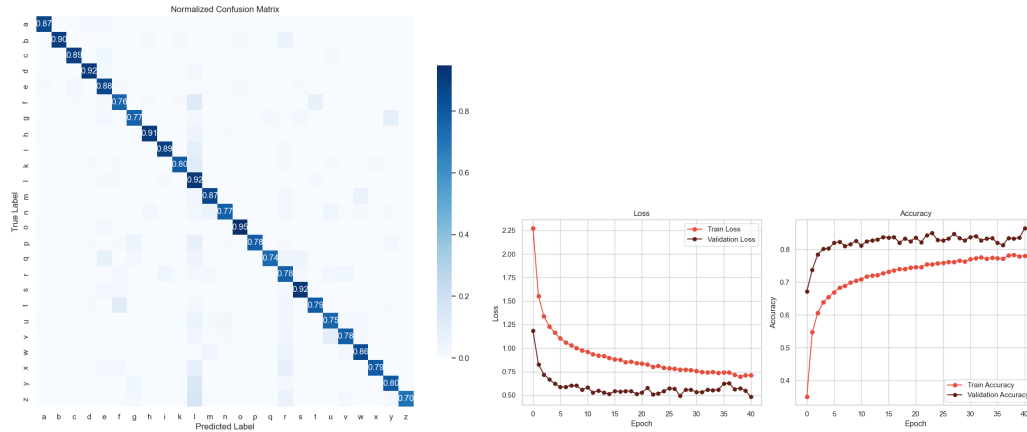
La matrice di confusione in Figura 5.6a mostra che SqueezeNet tende a classificare erroneamente molte classi nella classe "l". Questo suggerisce che SqueezeNet potrebbe non catturare le caratteristiche distintive di delle lettere in modo efficace, portando a una riduzione delle prestazioni per questa classe.

Per SqueezeNet non fine-tuned, la perdita di test era 3.929 e l'accuratezza di test era 0.048. Dopo il fine-tuning, le metriche di performance del modello sono migliorate significativamente, con una precisione di 0.826, un recall di 0.766 e un F1 score di 0.785.

### 5.2.3 ResNet

per ultimo il modello Resnet di cui si parla ampiamente nel paragrafo 3.3.2. Il modello ha 11.189.337 parametri addestrabili.

#### 5.2.3.1 Analisi della Matrice di Confusione e della Curva Loss e Accuracy per ResNet



(a) Matrice di Confusione per ResNet. (b) Curve di Loss e Accuracy Train e Valid test ResNet.

**Figura 5.7:** Analisi delle performance di ResNet sul dataset delle immagini di caratteri.

La matrice di confusione in Figura 5.6a mostra che ResNet tende a classificare in maniera più precisa rispetto a SqueezeNet ma ancora spanne sotto rispetto a AlexNet.

Per ResNet non fine-tuned, la perdita di test era 3.459, e l'accuratezza di test era 0.028. Dopo il fine-tuning, le metriche di performance del modello sono migliorate significativamente, con una precisione di 0.864, un recall di 0.832 e un F1 score di 0.843.

### 5.3 Riepilogo dei Risultati degli Esperimenti

I risultati dettagliati di tutti gli esperimenti condotti, inclusa l'analisi comparativa di AlexNet, ResNet e SqueezeNet, sono riassunti nella Tabella 5.8. Questo riepilogo mette in evidenza le metriche di prestazione, fornendo una panoramica completa dei punti di forza e delle debolezze di ciascuna configurazione del modello.

Experiment	Learning rate	Momentum	Weight decay	Train loss	Train accuracy	Validation loss	Validation accuracy	Test loss	Test accuracy
1	0.001	0.9	0.001	0.916	0.723	0.466	0.838	0.526	0.834
2	0.01	0.9	0.001	1.009	0.696	0.49	0.835	0.56	0.82
3	0.0001	0.9	0.001	0.985	0.706	0.529	0.831	0.528	0.828
ResNet	0.001	0.9	0.0001	0.714	0.78	0.483	0.864	0.481	0.852
AlexNet	0.001	0.9	0.0001	0.831	0.743	0.307	0.905	0.298	0.913
SqueezeNet	0.001	0.9	0.0001	1.22	0.63	0.746	0.766	0.794	0.77

**Figura 5.8:** Esperimenti su AlexNet, ResNet e SqueezeNet sono stati addestrati per 100 epoche.

### 5.4 Riflessioni e Considerazioni sui Risultati Sperimentali

L'obiettivo degli Esperimenti 1 a 3 su ResNet era identificare configurazioni ottimali di iperparametri e architetture di modelli per migliorare le prestazioni. L'Esperimento ResNet, in cui ho abbassato il weight decay a 0.0001, ha mostrato la più alta accuratezza di validazione di 0,864, indicando robuste capacità di generalizzazione senza sovradattamento significativo.

Tuttavia, ci sono diverse aree in cui si potrebbe apportare miglioramenti. Ad esempio, mentre gli esperimenti hanno testato diversi iperparametri, inclusi tassi di apprendimento e pesi di decadimento, ulteriori indagini su strategie di apprendimento adattive potrebbero portare a ulteriori miglioramenti delle prestazioni. Il problema dello squilibrio delle classi, particolarmente evidente nelle classificazioni errate delle classi "u,v,g,q" in tutti i modelli, rimane una sfida significativa che richiede ulteriori indagini e miglioramento del dataset.

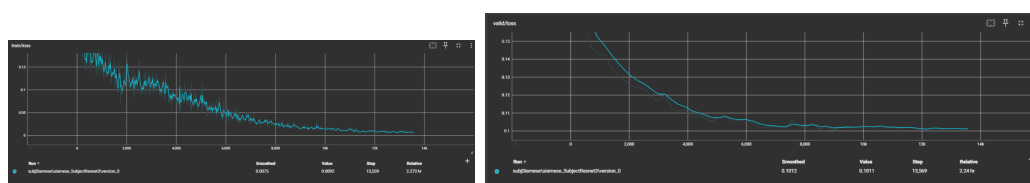
Ulteriori lavori potrebbero essere condotti sul numero di epoche di addestramento concesse ai modelli AlexNet, ResNet, SqueezeNet.

L'uso di supporti visivi, come le curve di accuratezza e perdita, le matrici di confusione e le curve di precisione-richiamo, è stato fondamentale per l'interpretazione dei risultati sperimentali. Queste visualizzazioni hanno facilitato la comprensione del comportamento dei modelli.

## 5.5 Modello Siamese

il Modello siamese è stato allenato con una ricostruzione del dataset iniziale dei caratteri, ottimizzato usando il modello di classificazione dei caratteri, inoltre per la costruzione delle coppie è stato usato il metodo spiegato in 3.4.4

### 5.5.1 Curve di Loss train e Validation



(a) Curva di Loss Train set.

(b) Curva di Loss Validation set.

**Figura 5.9:** Performance Modello Siamese.

### 5.5.2 Analisi della Curva di Addestramento

- **Andamento della curva:** La curva mostra una riduzione continua della perdita, il che indica che il modello sta imparando e migliorando durante l'addestramento.
- **Variabilità:** All'inizio, c'è una variabilità maggiore nella perdita, con fluttuazioni significative. Questo è comune nelle fasi iniziali dell'addestramento quando il modello sta ancora cercando di trovare un buon set di parametri.
- **Stabilizzazione:** Dopo circa 6000-8000 step, la perdita inizia a stabilizzarsi, con una riduzione più lenta e meno fluttuazioni.
- **Valore finale:** Il valore finale della perdita di addestramento è 0.0092, il che è abbastanza basso, indicando una buona capacità del modello di adattarsi ai dati di addestramento.

### 5.5.3 Analisi della Curva di Validazione

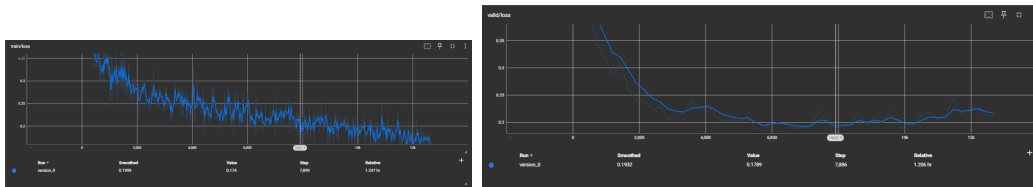
- **Andamento della curva:** Anche la curva della perdita di validazione mostra una riduzione costante, ma con un andamento leggermente diverso rispetto alla perdita di addestramento.

- **Stabilizzazione:** La perdita di validazione si stabilizza più rapidamente, attorno ai 6000 step, con una riduzione molto più graduale rispetto alla perdita di addestramento.
- **Valore finale:** Il valore finale della perdita di validazione è 0.1011, che è superiore al valore della perdita di addestramento. Questo gap tra la perdita di addestramento e quella di validazione indica la presenza di overfitting, anche se non è particolarmente marcato.

#### 5.5.4 Analisi Complessiva

- **Convergenza:** Entrambe le curve mostrano una buona convergenza, il che suggerisce che il modello sta imparando in modo efficace dai dati.
- **Overfitting:** C'è una leggera indicazione di overfitting poiché la perdita di validazione è superiore a quella di addestramento. Questo può essere mitigato utilizzando tecniche come la regolarizzazione, il dropout o l'early stopping.
- **Performance Generale:** Il modello sembra avere una buona capacità di generalizzazione, con una perdita di validazione che si stabilizza a un valore accettabile.

un altro esperimento è stato fatto aggiungendo un dropout rate del 0.2 questi sono i risultati:



(a) Curva di Loss Train set.

(b) Curva di Loss Validation set.

**Figura 5.10:** Performance Modello Siamese.

si può notare come aggiungendo il fattore di dropout rate la capacità di generalizzare ovvero l'accuracy del validation sia migliore rispetto al train set cosa che prima non avveniva per tanto l'aggiunta di questo parametro apporta sicuramente un miglioramento nelle capacità di generalizzare del modello. Ma nonostante ciò la loss nel validation rimane un po' più alta rispetto al modello senza dropout rate perciò bisognerebbe fare maggiori



esperimenti per trovare un equilibrio tra i parametri e magari aumentare anche il numero di batch da 64 a 254-512 per questo task specifico, cosa che con le basse performance del dispositivo con cui lavoro non ho potuto effettuare.

## 5.6 Differenza Di grafia Tra Soggetti

in questa sezione farò vedere una serie di distanze su uno stesso testo usando la contrastive loss spiegata in 3.4.2.1, ma scritto da diversi autori per far vedere la capacità del modello siamese di cogliere le differenti caratteristiche della grafia di ogni soggetto.

---

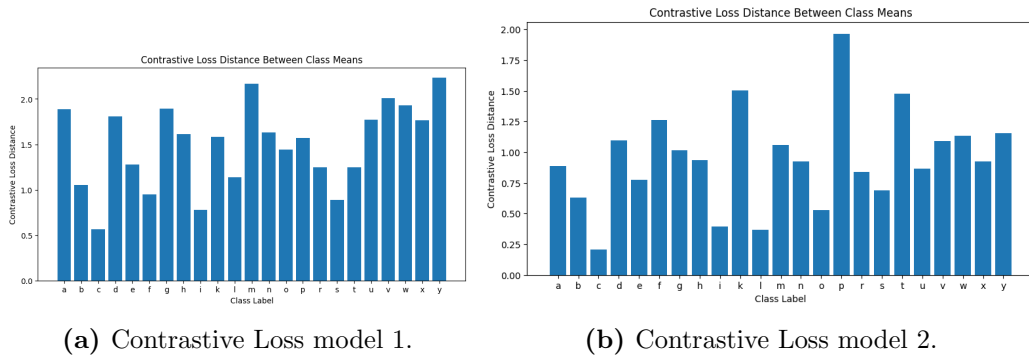
Imagine a vast sheet of paper on which straight Lines, Triangles, Squares, Pentagons, Hexagons, and other figures, instead of remaining fixed in their places, move freely about, on or in the surface, but without the power of rising above or sinking below it, very much like shadows - only hard and with luminous edges - and you will then have a pretty correct notion of my country and countrymen. Alas, a few years ago, I should have said "my universe": but now my mind has been opened to higher views of things.

---

**Figura 5.11:** Testo di riferimento.

i caratteri vengono estratti come nel metodo spiegato in 3.2.1.1 e sono stati raggruppati utilizzando i modelli espletati in 3.3 in particolare è stato usato il best model tra di essi.

### 5.6.1 Risultati



**Figura 5.12:** Contrastive loss distance.

queste due immagini mostrano la contrastive loss tra due soggetti che hanno scritto lo stesso testo di riferimento 5.11. Essa più è vicina al 2 più è alta la differenza nella scrittura del testo, vale a dire che i soggetti sono differenti e viceversa.

la contrastive loss trovata usando il primo modello 5.12a come possiamo vedere è abbastanza marcata mostrandoci che il modello ha saputo cogliere

egregiamente le differenze di grafia tra i due, la distanza totale è di 1.50 quindi sicuramente i soggetti sono diversi.

per quanto riguarda invece il secondo modello 5.12b invece ottimizzato con il dropout rate, la contrastive loss è un po' più bassa mostrandoci come, nonostante generalizzi meglio, la differenza di loss nel validation si fa sentire, la distanza totale in questo caso è del 0.95 che anche in questo caso ci fa capire che molto probabilmente i soggetti sono differenti.

infine questi sono i due testi scritti a mano dai due soggetti:

Imagine a vast sheet of paper on which straight Lines,  
Triangles, Squares, Pentagons, Hexagons, and other figures, instead  
of remaining fixed in their places, move freely about it, on or  
in the surface, but without the power of rising above or sinking  
below it, very much like shadows - only hard and with luminous  
edges - and you will then have a pretty correct notion of my  
country and countrymen. Alas, a few years ago, I should have  
said "my universe": but now my mind has been opened to  
higher views of things.

(a) Testo soggetto 1.

Imagine a vast sheet of paper on which straight Lines, Triangles,  
Squares, Pentagons, Hexagons, and other figures, instead of  
remaining fixed in their places, move freely about, on or in the  
surface, but without the power of rising above or sinking below it, very  
much like shadows - only hard and without luminous edges - and you will  
then have a pretty correct notion of my country and countrymen.  
Alas, a few years ago, I should have said "my universe": but now my mind  
has been opened to higher views of things.

(b) Testo soggetto 2.

**Figura 5.13:** Testi scritti a mano.

prendendo come riferimento il best model adesso comparerò lo stesso testo con 5 soggetti plottando le distanze in uno spazio 3dimensionale:

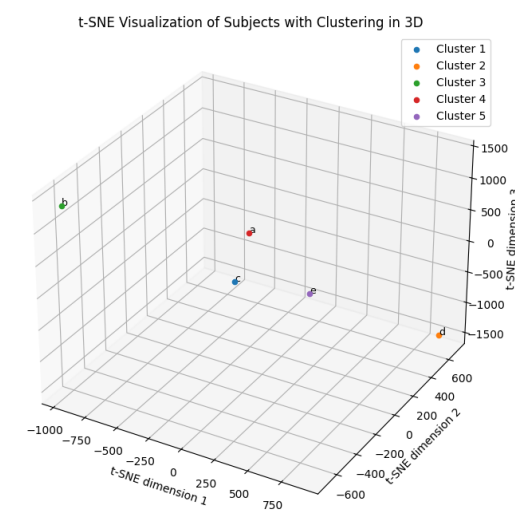


Figura 5.14: Distanze plottate.

come mostrato in figura i soggetti a c ed e sono quelli più "vicini", per tanto è possibile effettuare ulteriori studi, prendendo come riferimento quest'ultimi facendo un ulteriore comparazione.

infine ecco la matrice per le distanze calcolate con la contrastive loss:

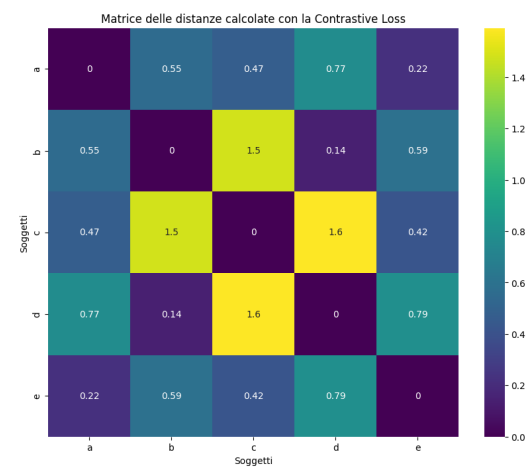


Figura 5.15: Matrice delle distanze.

## Capitolo 6

# Conclusioni e supplementi futuri

In conclusione, questa tesi ha dimostrato l'efficacia dell'uso di algoritmi di machine learning, in particolare le reti neurali convoluzionali (CNN) e le reti siamesi, per l'analisi forense della scrittura a mano. Il nostro approccio ha mostrato un'elevata accuratezza nell'identificazione degli autori basata sulle caratteristiche distintive della scrittura a mano, superando i metodi tradizionali in termini di precisione e affidabilità.

L'uso delle CNN per l'estrazione delle caratteristiche e delle reti siamesi per il confronto delle immagini ha permesso di sviluppare un sistema robusto e generalizzabile, in grado di affrontare le variabilità intrinseche della scrittura a mano. I risultati sperimentali hanno confermato la validità del nostro approccio, con metriche di performance elevate sia nelle fasi di addestramento che di validazione.

Tuttavia, ci sono ancora aree che possono essere migliorate. La gestione dello squilibrio delle classi e l'ottimizzazione degli iperparametri rimangono sfide aperte. Futuri lavori potrebbero concentrarsi sull'espansione del dataset e sull'implementazione di tecniche avanzate di regolarizzazione per migliorare ulteriormente le prestazioni del modello. Altri futuri lavori potrebbero riguardare anche il miglioramento dello script che estrae i caratteri dall'immagine, utilizzando anche il modello classificatore di caratteri nell'operazione.

Inoltre, l'integrazione di nuove tecniche di intelligenza artificiale e l'applicazione di questo approccio ad altri tipi di analisi forense potrebbero offrire ulteriori opportunità di miglioramento e applicazione. La continua evoluzione delle tecnologie di machine learning promette di portare significativi avan-

zamenti nel campo delle scienze forensi, rendendo le indagini più accurate e affidabili.

Questa ricerca rappresenta un passo avanti significativo nell'uso del machine learning per l'analisi forense della scrittura a mano, aprendo la strada a futuri sviluppi e applicazioni in questo campo vitale.

# Bibliografia

- [1] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 37:1–8, 2015.
- [2] David M Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2021.
- [3] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.
- [4] David M Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2021.
- [5] San Zhang. *Loss Functions for Deep Learning: A Comprehensive Review*. Springer, 2021.
- [6] Michael Martin. *Evaluating the Accuracy of Machine Learning Models*. O'Reilly Media, 2022.
- [7] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.
- [8] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.
- [9] Iplab, 2015. <http://iplab.dmi.unict.it/>.
- [10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Convolutional networks for images, speech, and time series. *Handbook of Brain Theory and Neural Networks*, 3361:1995, 1995.

- [11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [12] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.
- [13] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [15] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.
- [16] Awni Hannun, Chuan Guo, and Laurens van der Maaten. Measuring data leakage in machine-learning models with fisher information. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 760–770. PMLR, 27–30 Jul 2021.
- [17] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.
- [18] Alex Lenail. Nn-svg: Interactive web-based tool for designing neural networks, 2024. Accessed: 2024-07-08.
- [19] Alexander LeNail. Nn-svg: Publication-ready neural network architecture schematics. *The Journal of Open Source Software*, 4(33):747, 2019.
- [20] Manuel Martinez and Rainer Stiefelhagen. Taming the cross entropy loss. *arXiv preprint arXiv:1810.05075*, 2018.
- [21] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.



- [22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on Machine Learning*, 37:448–456, 2015.
- [24] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814, 2010.
- [25] G. Biau, G. Celeux, P. Craiuci, A. Glorieux, and O. Mestre. A survey of categorical encoding techniques for feature engineering in machine learning. *Statistics Surveys*, 11(1):1–43, 2018.
- [26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Softmax function: Definition, formula, applications, and examples. pages 227–232, 2016.
- [27] Robert Tibshirani. Lasso: The least absolute shrinkage and selection operator. *Journal of the Royal Statistical Society. Series B*, 58(1):267–286, 1996.
- [28] A. E. Hoerl and R. W. Kennard. Ridge regression: An introduction. *Biometrika*, 56(1):1–14, 1970.
- [29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [30] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Proceedings of the 14th International Conference on Learning Representations (ICML)*, 2011.
- [31] Python Software Foundation. Tkinter: Python interface to tcl/tk. <https://docs.python.org/3/library/tkinter.html>, 2021.
- [32] Project Jupyter Contributors. Project jupyter. <https://jupyter.org/>, 2021.
- [33] torch.nn.module — pytorch 1.10.0 documentation. Accessed: insert-date-here.

- [34] Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu. Understanding and improving early stopping for learning with noisy labels. *arXiv preprint arXiv:2106.15853*, 2021.
- [35] Lutz Prechelt. *Early Stopping — But When?* Springer, 2012.
- [36] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607, 2020.

# Ringraziamenti