

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Matematica

Riconoscimento di Immagini tramite Distanza Tangente

Tesi di Laurea in Matematica Computazionale

Relatore:
Chiar.ma Prof.ssa
VALERIA SIMONCINI

Presentata da:
GIUSEPPE AGUZZONI

Anno Accademico 2022-2023

Introduzione

L'analisi e il riconoscimento di caratteri alfanumerici scritti a mano rivestono un ruolo fondamentale in diverse applicazioni, dalla classificazione automatica di codici postali all'elaborazione di manoscritti. Questa tesi si propone di studiare l'affidabilità ed efficienza del riconoscimento di cifre scritte a mano, focalizzandosi sull'utilizzo della "distanza tangente" come metrica per il confronto delle immagini. Tale distanza consente di quantificare le variazioni locali tra le immagini causate da deformazioni, come traslazioni e rotazioni.

Durante la tesi, verrà esaminata in dettaglio la teoria alla base della distanza tangente e le sue applicazioni. Saranno esplorati metodi efficienti e affidabili per il calcolo della distanza, rendendola adattabile a diversi database, con particolare attenzione al database MNIST, ma includendo anche risultati su altri come USPS e NIST. Questi metodi, non richiedendo informazioni a priori sul database, potranno essere utilizzati su database completamente diversi, ampliando le applicazioni a settori come il riconoscimento facciale e delle espressioni.

Nonostante grandi deformazioni possano rappresentare una sfida per la corretta classificazione, l'uso delle strategie descritte in questa tesi permette già di trattare in modo automatico un significativo numero di caratteri alfanumerici soggetti a grandi modifiche di forma.

Indice

Introduzione	i
1 Introduzione alla Classificazione di Immagini	1
1.1 Decomposizione ai Valori Singolari SVD	1
1.2 Classificazione tramite SVD	5
2 Il Metodo della Distanza Tangente	9
2.1 Creazione di una metrica	10
2.2 Spazio tangente di un'immagine	11
2.3 Distanza tangente bilaterale	13
2.4 Costanti molla	14
3 Algoritmi della Distanza Tangente nel riconoscimento di Ci-	
fre	17
3.1 Metodo delle Secanti	18
3.2 Strategie per l'identificazione dei vettori tangenti dei digit . .	21
3.3 Algoritmo per il calcolo della Distanza Tangente	26
3.4 Analisi dei Risultati	29
Conclusioni	35
Bibliografia	37

Elenco delle figure

1.1	SVD Ridotta	3
1.2	Valori Singolari	5
1.3	Immagini del database MNIST	6
2.1	Idea Geometrica della Distanza Tangente	13
2.2	Costante Molla	16
3.1	Metodo di approssimazione tramite linee secanti	19
3.2	Esempio traslazione orizzontale tramite secanti	20
3.3	Esempio rotazione tramite secanti	21
3.4	Centroidi di MNIST	28
3.5	Funzionamento del K-Nearest Neighbors	30
3.6	5-nearest neighbors mediante distanza euclidea	30
3.7	4-nearest neighbors mediante distanza tangente bilaterale . . .	31
3.8	Dendrogramma utilizzando la distanza euclidea	32
3.9	Dendrogramma utilizzando la distanza tangente unilaterale . .	33
3.10	Dendrogramma utilizzando la distanza tangente bilaterale . .	34

Capitolo 1

Introduzione alla Classificazione di Immagini

Per una corretta classificazione gli algoritmi considerati hanno bisogno di un insieme abbastanza grande di immagini già classificate che permette di avere oggetti con cui confrontare le immagini da classificare. Questo insieme è chiamato *training set* mentre l'insieme delle immagini da classificare è chiamato *test set*. Considereremo le immagini in tre modi differenti:

- come immagini in scala di grigi di dimensione 28 x 28 pixel
- come vettori dello spazio \mathbb{R}^{784}
- come funzioni di due variabili $f(x, y)$

1.1 Decomposizione ai Valori Singolari SVD

Definizione 1.1. Una matrice $U \in \mathbb{C}^{n \times n}$ con $n \in \mathbb{N}$ si dice unitaria se soddisfa: $UU^* = I$ con I matrice identità e U^* corrisponde alla trasposta coniugata di U .

Teorema 1.2. (Esistenza della decomposizione ai valori singolari SVD)
Sia $A \in \mathbb{C}^{m \times n}$, $q = \min\{m, n\} \implies$ esiste $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_q, 0, \dots, 0) \in$

$\mathbb{R}^{m \times n}$ con $\sigma_1 \geq \dots \geq \sigma_q \geq 0$ e due matrici unitarie $U \in \mathbb{C}^{m \times m}$, $V \in \mathbb{C}^{n \times n}$ tali che $A = U\Sigma V^*$.

Dimostrazione. Sia A $m \times n$ con $m \geq n$, ciò non è restrittivo. Allora: sia $x = \operatorname{argmax}_{\|x\|_2=1} \|Ax\|_2$ e sia $Ax = \sigma_1 y$ con $\|y\|_2 = 1$ e $\sigma_1 = \|A\|$. Definiamo $X_1 = [x, \tilde{X}_2] \in \mathbb{R}^{n \times n}$, $Y_1 = [y, \tilde{Y}_2] \in \mathbb{R}^{m \times m}$ tali che X_1, Y_1 siano unitarie. Definiamo ora:

$$A_1 = Y_1^T A X_1 = \begin{bmatrix} \sigma_1 & d^T \\ 0 & B \end{bmatrix},$$

poiché $y^T Ax = \sigma_1$ e $\tilde{Y}_2^T Ax = \sigma_1 \tilde{Y}_2^T y = 0$. Inoltre $d = 0$, poichè ponendo $z = [\sigma_1; d]$ si ha:

$$\begin{aligned} \sigma_1^2 &\geq \frac{\|A_1 z\|_2^2}{\|z\|_2^2} = \frac{1}{\sigma_1^2 + d^T d} \|A_1[\sigma_1; d]\|_2^2 = \frac{1}{\sigma_1^2 + d^T d} \|[\sigma_1^2 + d^T d; Bd]\|_2^2 \\ &\geq \sigma_1^2 + d^T d \end{aligned}$$

il che è assurdo a meno che $d = 0$. Otteniamo quindi la decomposizione iterando il procedimento appena compiuto sulla matrice B e risulteranno $U = Y_1 \cdots Y_{n-1}$, $V = X_1 \cdots X_{m-1}$. \square

Definizione 1.3. Avendo la SVD di una matrice $A \in \mathbb{C}^{m \times n}$ definiamo:

- σ_i : i valori singolari della matrice, $\sigma_i = \sqrt{\lambda_i(AA^*)}$;
- u_i, v_i : (le colonne di U e V) rispettivamente i vettori singolari sinistri e destri.

Otteniamo U e V matrici quadrate e Σ avente le stesse dimensioni di A . Se $m \geq n$ allora la matrice U $m \times m$ risulta essere grande e una parte delle colonne di U sono moltiplicate per 0 in Σ . In questo caso si può ricorrere alla *SVD ridotta* che permette di risparmiare tempo e spazio di archiviazione creando una U' di dimensione $m \times n$, avente come colonne le prime m colonne di U e Σ di dimensione $n \times n$. Lo stesso vale per $n \geq m$ in cui si otterrà la matrice V' di dimensione $n \times m$ rispetto a $n \times n$, avente per colonne le prime

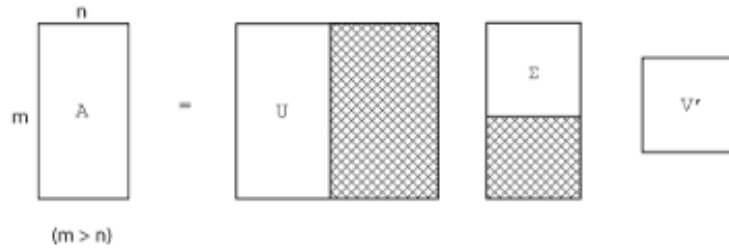


Figura 1.1: SVD ridotta: la parte oscurata della matrice U è quella che verrebbe moltiplicata per 0 e quindi non utile al fine della scomposizione mentre la matrice dei valori singolari si scompone in Σ di dimensione $n \times n$ e 0 (parte oscurata).

n colonne di V .

Si nota che il numero di valori singolari non nulli coincide con il rango di A : supponiamo per esempio $m \geq n$ e $\Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_{n-1}, 0)$ allora

$$\begin{aligned}
 A &= [U_1, U_2] \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^* \\
 &= [u_1, \dots, u_{n-1}, u_n, U_2] \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^* \\
 &= [u_1, \dots, u_{n-1}] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{n-1} \end{bmatrix} \begin{bmatrix} v_1^* \\ \vdots \\ v_{n-1}^* \end{bmatrix}
 \end{aligned}$$

otteniamo $\text{Range}(A) = \text{Range}([u_1, \dots, u_{n-1}])$.

Esiste inoltre un'ulteriore forma per questa decomposizione chiamata *SVD troncata* che fornisce un'approssimazione di rango basso della matrice: se per $k \in \{1, \dots, n-1\}$, $\sigma_{k+1} \ll \sigma_k$ allora

$$A = \sum_{i=1}^n u_i \sigma_i v_i^T = \sum_{i=1}^k u_i \sigma_i v_i^T + \sum_{i=k+1}^n u_i \sigma_i v_i^T \approx \sum_{i=1}^k u_i \sigma_i v_i^T =: A_k, \quad (1.1)$$

poiché l'apporto che ci dà la sommatoria da $k+1$ a n è trascurabile.

Teorema 1.4. Sia $A \in \mathbb{R}^{m \times n}$ con $m \geq n$ (analogo per $n \geq m$), e sia $\text{rank}(A) = r \leq n$. Fissato $k < r$ si ha che:

$$\min_{\text{rank}(B)=k} \|A - B\|_2$$

è risolto dalla SVD troncata al k -esimo termine dello sviluppo.

Dimostrazione. Esprimendo A tramite lo sviluppo $A = \sum_{j=1}^{\text{rank}(A)} \sigma_j u_j v_j^T$ e ponendo $A_k = \sum_{j=1}^k \sigma_j u_j v_j^T$ otteniamo:

$$\|A - A_k\|_2 = \left\| \sum_{j=1}^{\text{rank}(A)} \sigma_j u_j v_j^T - \sum_{j=1}^k \sigma_j u_j v_j^T \right\|_2 = \left\| \sum_{j=k+1}^{\text{rank}(A)} \sigma_j u_j v_j^T \right\|_2 = \sigma_{k+1}.$$

Resta da dimostrare che per una generica matrice $B \in \mathbb{R}^{m \times n}$, con $\text{rank}(B) = k$ tale residuo è maggiore di σ_{k+1} .

Prendendo una tale B otteniamo $\dim(\ker(B)) = n - k$ e definiamo $\{x_1, \dots, x_{n-k}\}$ base per $\ker(B)$.

Da Grassmann sappiamo $\dim(\ker(B) \cap \text{span}\{v_1, \dots, v_{k+1}\}) \geq 1$, scegliendo $z \in \ker(B) \cap \text{span}\{v_1, \dots, v_{k+1}\}$ t.c. $\|z\|_2 = 1$ si ha:

$$\begin{aligned} \|A - B\|_2^2 &\geq \|Az - Bz\|_2^2 = \|Az\|_2^2 = \left\| \sum_{j=1}^{k+1} \sigma_j u_j v_j^T z \right\|_2^2 = \sum_{j=1}^{k+1} \sigma_j^2 (v_j^T z)^2 \\ &\geq \sum_{j=1}^{k+1} \sigma_{k+1}^2 (v_j^T z)^2 = \sigma_{k+1}^2. \end{aligned}$$

□

Si può notare dalla figura 1.2 una struttura a "gomito" dei grafici, questo suggerisce che il troncamento della SVD passato il "gomito" ci darà una perdita di informazioni trascurabile poichè i valori singolari che andiamo a perdere sono significativamente più piccoli rispetto a quelli mantenuti.

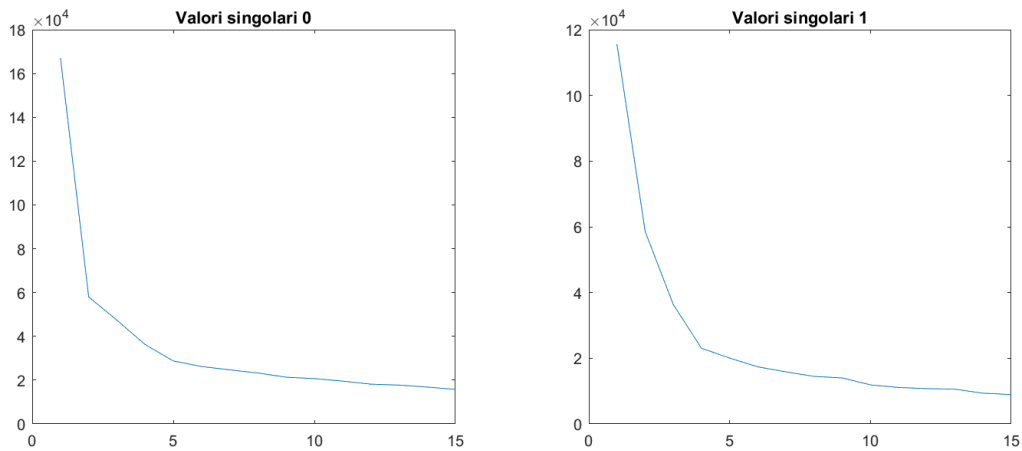


Figura 1.2: Grafici relativi ai primi 15 valori singolari del training set 0 e 1.

1.2 Classificazione tramite SVD

Per il database MNIST, il training set contiene tra 5000 e 6500 immagini per ogni cifra, mentre il test set comprende tra 800 e 1000 immagini per ciascuna cifra. Selezionando un sottoinsieme sia dal training set che dal test set, le immagini del test set verranno classificate confrontandole con tutte le immagini del training set per valutare l'accuratezza dell'algoritmo.

Il numero di immagini nel training e test sets è riportato nella seguente tabella:

cifra	0	1	2	3	4	5	6	7	8	9
train	5923	6724	5958	6131	5842	5421	5918	6265	5851	5949
test	980	1135	1032	1010	982	892	958	1028	974	1009

La classificazione delle immagini tramite SVD avviene nel modo seguente:

Passo 1: per ogni cifra i del training set costruiamo una matrice A^i avente per colonne le immagini sotto forma di vettore $\in \mathbb{R}^{784}$, ottenendo 10 matrici di dimensioni $784 \times m$, dove m è il numero di immagini nel training set per la cifra considerata. Calcoliamo la sua SVD troncata a rango k delle matrici



Figura 1.3: Esempio di immagini per ogni cifra del database MNIST

A^i , ottenendo: $A_{(k)}^i = U_{(k)}^i \Sigma_{(k)}^i (V_{(k)}^i)^T$.

Le colonne della matrice $U_{(k)}^i$ sono le direzioni principali. Ovvero le direzioni, tra loro ortogonali, privilegiate dai vettori del training set. Otteniamo quindi che le prime k colonne delle matrici U^i ($U_{(k)}^i$) forniranno una base per i nostri sottospazi H_i .

Passo 2: possiamo quindi calcolare la distanza tra un punto dello spazio z (la cifra che dobbiamo classificare) e tali sottospazi. Per farlo basta calcolare la componente ortogonale di z rispetto ai sottospazi ottenuti, ciò risulta semplice dato che disponiamo di basi ortogonali per ogni sottospazio. Allora $\forall i \in \{0, 1, \dots, 9\}$ troviamo:

$$d_i = \min_{x \in H_i} \|z - x\|_2 = \|z - U_{(k)}^i (U_{(k)}^i)^T z\|_2. \quad (1.2)$$

Passo 3: l'immagine z verrà classificata nel i t.c. $i = \operatorname{argmin}_i(d_i)$ ovvero verrà classificata con la cifra i corrispondente al sottospazio H_i a lei più vicino.

L'utilizzo della SVD troncata è utile per due principali motivi:

- elevato risparmio di costo computazionale: limitiamo notevolmente le dimensioni del nostro problema diminuendo così il tempo di calcolo; lavoriamo inizialmente con matrici $784 \times m$, con m numero di immagini

presenti nel training set della cifra analizzata (più di 5000 immagini per ogni cifra).

- Sfortunatamente, quando si lavora con la SVD completa, si rischia di ottenere valori singolari molto piccoli. Se non vengono trascurati, questi valori potrebbero generare errori nella classificazione, dati dall'intersezione che questi valori causano tra gli spazi H_i . Queste intersezioni potrebbero portare a più classificazioni possibili per un'immagine, complicando così il processo di classificazione e rendendo difficile ottenere una classificazione precisa. Utilizzando la SVD troncata ai primi k autovalori, si evitano tali intersezioni, garantendo una classificazione univoca per ciascuna immagine e migliorando l'accuratezza complessiva del sistema di classificazione.

Risultati

Utilizzando un set di addestramento composto da 5000 immagini per ciascuna cifra e un set di test con 500 immagini per cifra, l'applicazione del metodo SVD ha prodotto i seguenti risultati in termini di accuratezza nella classificazione:

cifra	0	1	2	3	4	5	6	7	8	9
SVD	98.6	99.0	90.8	94.2	93.2	89.4	93.8	89.6	87.4	90.2

È evidente come questo metodo dimostri un'eccellente accuratezza per alcune cifre, come lo zero e l'uno.

Capitolo 2

Il Metodo della Distanza Tangente

Un efficace algoritmo di classificazione per cifre scritte a mano dovrebbe essere in grado di riconoscere correttamente numeri scritti chiaramente, anche quando le differenze nei dettagli dell'immagine causano una notevole variazione nella posizione dei pixel rispetto a un modello di riferimento. In altre parole, l'algoritmo dovrebbe essere robusto abbastanza da gestire variazioni significative nelle rappresentazioni delle cifre mentre le classifica accuratamente. Per questo motivo la distanza euclidea non può essere considerata una corretta metrica per questo problema poichè se immaginiamo di deformare leggermente un'immagine (mediante rotazioni, traslazioni, ecc.) essa risulta facilmente riconoscibile da parte di una persona mentre nello spazio dei pixel, per la distanza euclidea, sarà cambiata notevolmente. Questo solitamente porta ad una incorretta classificazione dell'immagine da parte del classificatore definito nel capitolo precedente. In questo capitolo descriviamo una diversa distanza che sia resistente alle piccole deviazioni pixel per pixel. Cerchiamo una metrica che tenga conto delle differenze tra le cifre senza essere fortemente influenzata dalle variazioni locali nella loro rappresentazione pixel.

Per questo capitolo abbiamo preso come riferimento l'articolo [1] e [2].

2.1 Creazione di una metrica

Definizione 2.1. *Uno spazio metrico è una struttura matematica (X, d) composta da un'insieme X e una distanza d detta metrica che associa a due punti dell'insieme un numero reale non negativo di modo che valgano le seguenti proprietà: $\forall x, y, z \in X$*

1. $d(x, y) = 0 \iff x = y$
2. $d(x, y) = d(y, x)$
3. $d(x, y) \leq d(x, z) + d(z, y)$.

Stiamo lavorando per ideare una misura nello spazio dei pixel che sia in grado di determinare con precisione se due immagini che raffigurano la stessa cifra sono simili, anche se possono differire in alcuni dettagli. Nel contempo, ci proponiamo di garantire che questa misura sia abile a distinguere inequivocabilmente tra immagini appartenenti a modelli distinti, indicandole come differenti. In altre parole, cerchiamo un approccio che possa individuare le caratteristiche specifiche di una cifra, mantenendo allo stesso tempo la capacità di evidenziare le differenze rilevanti tra cifre appartenenti a categorie diverse.

Immaginiamo di deformare un'immagine E per adattarla al meglio ad un'altra immagine P e definiamo:

- $\delta(E, P)$ misura della differenza delle immagini nel miglior adattamento
- $\epsilon(E, P)$ misura dell'entità con cui E è stata deformata per adattarsi al meglio a P

Date queste misure possiamo definire una metrica sullo spazio dei pixel come

$$D(E, P) = \delta(E, P) + \epsilon(E, P). \quad (2.1)$$

È rilevante notare che il valore di $D(E, P)$ può risultare significativo anche quando $\delta(E, P)$ è di entità ridotta. Ciò accade poichè è sufficiente che l'immagine E sia stata notevolmente deformata per ottenere un buon adattamento

rispetto a P . La stessa logica si applica a $\epsilon(E, P)$; anche se l'immagine E non ha subito deformazioni sostanziali, il miglior adattamento potrebbe comunque non essere ottimale.

Questa configurazione della metrica riflette come $D(E, P)$ tiene in considerazione sia le deformazioni significative che la qualità complessiva dell'adattamento tra le due immagini. In altre parole, la metrica è progettata per essere sensibile non solo alla presenza di deformazioni visibili, ma anche alla bontà dell'allineamento tra le caratteristiche delle immagini.

2.2 Spazio tangente di un'immagine

Una volta identificate le caratteristiche desiderate nella nostra metrica, procediamo a definire lo spazio tangente di un'immagine. Questo spazio tangente sarà essenziale per il calcolo della distanza tangente, che stiamo cercando di sviluppare. La distanza tangente è concepita in modo tale da non essere sensibile alle piccole variazioni o modifiche nelle immagini. In pratica, stiamo costruendo uno spazio in cui possiamo valutare le differenze tra le immagini in modo più robusto, concentrandoci sulle caratteristiche rilevanti e minimizzando l'impatto di piccole variazioni.

Definizione 2.2. *Definiamo $S_{x^p}(\alpha) : \mathbb{R}^m \rightarrow \mathbb{R}^m$ come una mappatura che deforma l'immagine x^p mediante $\alpha \in \mathbb{R}^s$ vettore di s parametri di deformazione.*

Definizione 2.3. *La Superficie di Corruzione dell'immagine viene definita come:*

$$S_{x^p} = \{y \in \mathbb{R}^m \mid \exists \alpha \in \mathbb{R}^s \text{ t.c. } y = S_{x^p}(\alpha)\}. \quad (2.2)$$

Definizione 2.4. *Lo Spazio Tangente al sottospazio in x^p è generato dalle colonne della matrice jacobiana*

$$J = \left. \frac{\delta S_{x^p}(\alpha)}{\delta \alpha} \right|_{\alpha=0}. \quad (2.3)$$

Possiamo notare che in un intorno di $\alpha = 0$ si ottiene che S_{x^p} soddisfa:

$$S_{x^p}(\alpha) = S(P, 0) + \alpha \frac{\partial S(P, \alpha)}{\partial \alpha} + O(\alpha^2) \approx x^p + J\alpha \quad (2.4)$$

Il nostro approccio consiste nel calcolare la minima distanza tra le superfici lineari che meglio approssimano i sottospazi S_P e S_E delle immagini P e E .

Questo risolve tre problemi principali:

1. I sottospazi lineari hanno semplici espressioni analitiche che possono essere facilmente calcolate e memorizzate;
2. Trovare la minima distanza tra i sottospazi lineari è un semplice problema dei minimi quadrati;
3. Questa distanza è inoltre localmente invariante; purtroppo non sempre lo è globalmente.

Nella Figura 2.1 viene mostrata l'idea di come potrebbe essere raffigurata la distanza tangente; le linee verdi rappresentano le curve di deformazione dell'immagine rispetto ai vari α di deformazione.

Esistono due varianti di distanza tangente: unilaterale e bilaterale. La prima è meno efficace della seconda in quanto tiene conto solo dello spazio creato da un'immagine, mentre la seconda considera gli spazi di entrambe le immagini. L'approccio bilaterale si rivela più vantaggioso, consentendoci di ottenere risultati migliori in quanto tiene in considerazione le caratteristiche di entrambe le immagini coinvolte nella valutazione, migliorando così la completezza e l'accuratezza complessiva della misura di distanza, nonostante un costo computazionale più elevato.

La distanza tangente unilaterale viene definita nel modo seguente: sia X un'immagine di cui sappiamo l'etichetta, ovvero la cifra raffigurata nell'immagine, e Y una nuova immagine da classificare, otteniamo:

$$D(X, Y) = \|Y - JJ^T Y\|_2, \quad (2.5)$$

dove J è lo spazio tangente dell'immagine X .

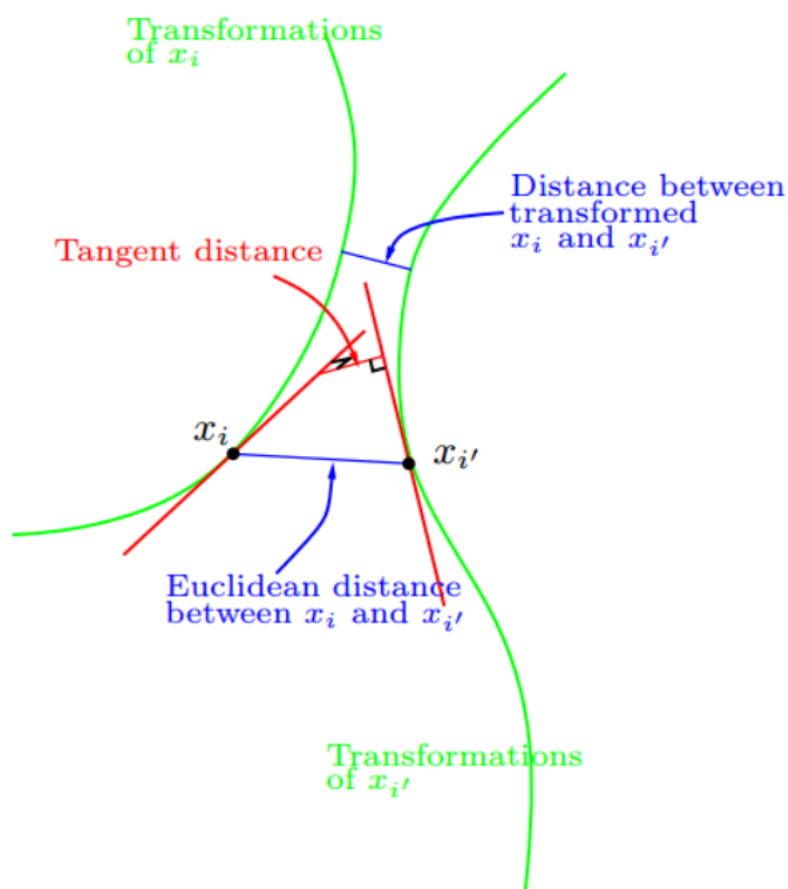


Figura 2.1: Idea geometrica della distanza tangente

2.3 Distanza tangente bilaterale

La distanza tangente che stiamo adottando, ritenuta la più esaustiva, è la distanza tangente bilaterale. Questa scelta riflette la nostra volontà di utilizzare un approccio che sia completo e consideri entrambi i lati della similarità tra immagini, contribuendo a garantire una valutazione accurata e robusta delle relazioni tra le caratteristiche delle cifre rappresentate. La definiamo come: prese due immagini E e P

$$TD(E, P) = \min_{x \in T_E, y \in T_P} \|x - y\|_2^2, \quad (2.6)$$

dove T_E è il piano tangente di S_E in E e T_P è il piano tangente di S_P in P .

Definiamo ora le matrici L_E e L_P come le jacobiane de E, P :

$$L_X = \frac{\partial S(X, \alpha)}{\partial \alpha} \Big|_{\alpha=0} = \left[\frac{\partial S(X, \alpha)}{\partial \alpha_1}, \dots, \frac{\partial S(X, \alpha)}{\partial \alpha_s} \right] \Big|_{\alpha=0}. \quad (2.7)$$

Cerchiamo quindi di trovare la distanza minore tra i due sottospazi:

$$TD(E, P) = \min_{\alpha_E, \alpha_P} \|(E + L_E \alpha_E) - (P + L_P \alpha_P)\|_2^2, \quad (2.8)$$

e sviluppandola troviamo:

$$TD(E, P) = \min_{\alpha_E, \alpha_P} \|(E - P) - [L_P, -L_E] \begin{bmatrix} \alpha_E \\ \alpha_P \end{bmatrix}\|_2^2, \quad (2.9)$$

dove α_E e α_P sono soluzioni del sistema lineare:

$$\begin{cases} (L_{PE} L_{EE}^{-1} L_E^T - L_P^T)(E - P) = (L_{PE} L_{EE}^{-1} L_{EP} - L_{PP}) \alpha_P \\ (L_{EP} L_{PP}^{-1} L_P^T - L_E^T)(E - P) = (L_{EE} - L_{EP} L_{PP}^{-1} L_{PE}) \alpha_E \end{cases}, \quad (2.10)$$

considerando $L_{EE} = L_E^T L_E$ e $L_{EP} = L_E^T L_P$, stessa cosa per L_{PE} e L_{PP} .

Una volta trovati α_E e α_P la distanza tangente bilaterale è facilmente calcolabile dato che le matrici definite hanno rango pieno.

Osservazione 2.5. La distanza tangente unilaterale è chiaramente meno costosa da calcolare ma la miglior classificazione data dalla distanza tangente bilaterale rende questo algoritmo, anche se più costoso, molto più interessante; inoltre la distanza tangente bilaterale ha la possibilità di essere ulteriormente migliorata tramite l'utilizzo di costanti molla.

2.4 Costanti molla

Per controllare la deformazione e incrementare le performance del classificatore introduciamo delle costanti molla su α_E e α_P e ciò attribuisce un peso alla distanza tra la proiezione dell'immagine sullo spazio tangente e l'immagine originale.

Incorporando quindi una costante molla k nell'equazione (2.8) della distanza tangente bilaterale otteniamo:

$$TD(E, P) = \min_{\alpha_E, \alpha_P} \|(E + L_E \alpha_E) - (P + L_P \alpha_P)\|_2^2 + k \|L_E \alpha_E\|_2^2 + k \|L_P \alpha_P\|_2^2, \quad (2.11)$$

dove ora α_E e α_P sono soluzioni del seguente sistema:

$$\begin{cases} (L_{PE} L_{EE}^{-1} L_E^T - (1+k)L_P^T)(E - P) = (L_{PE} L_{EE}^{-1} L_{EP} - (1+k)^2 L_{PP}) \alpha_P \\ (L_{EP} L_{PP}^{-1} L_P^T - (1+k)L_E^T)(E - P) = ((1+k)^2 L_{EE} - L_{EP} L_{PP}^{-1} L_{PE}) \alpha_E \end{cases} \quad (2.12)$$

Il sistema ottenuto presenta la stessa complessità della distanza tangente bilaterale, ma con alcune distinzioni notevoli. Una caratteristica distintiva è che questo sistema ha sempre una soluzione per ogni $k > 0$, e inoltre, è numericamente più stabile.

È interessante notare che impostando $k = 0$, si ottiene di fatto la distanza tangente originale, mentre con $k = +\infty$ si ritorna alla distanza euclidea. Questa variazione parametrica permette di regolare la sensibilità del sistema alle deformazioni tra le immagini. In questa prospettiva, la distanza tangente approssima quanto una nuova immagine debba deformarsi per adattarsi a quella originale, e la variabile k funge da parametro per quantificare la tensione della "molla" che rappresenta la resistenza alle deformazioni. Ad esempio, quando k è molto grande (vicino a $+\infty$), la molla è rigida e la metrica si avvicina alla distanza euclidea. Quando k è più piccolo, la molla è più elastica, riflettendo una maggiore tolleranza alle variazioni. Questa interpretazione ci riconduce alla metrica iniziale che abbiamo definito per lo spazio dei pixel, sottolineando come il parametro k influisca sulla flessibilità del sistema nel valutare e quantificare le differenze tra le immagini (equazione (2.1)).

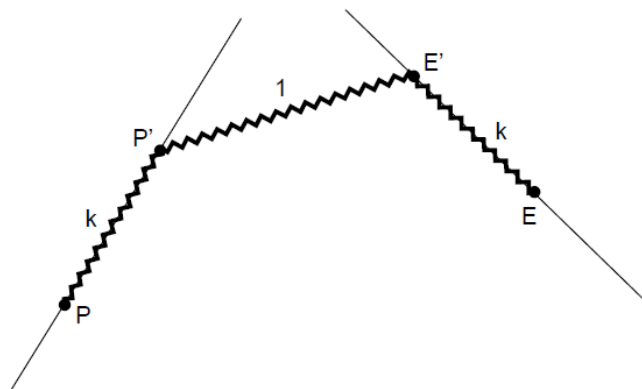


Figura 2.2: La distanza tangente tra E e P può essere considerata come l'energia elastica immagazzinata nelle tre molle che connettono P , P' , E' e E . In questo contesto si può considerare che P' e E' possano muoversi liberamente "senza attrito" sui piani tangenti, sottolineando la natura elastica del sistema e la sua capacità di adattarsi alle variazioni tra le immagini E e P .

Capitolo 3

Algoritmi della Distanza

Tangente nel riconoscimento di Cifre

In questo capitolo, illustreremo il processo di sviluppo di un sistema di riconoscimento, affrontando alcune difficoltà specifiche legate alla tipologia dei dati che stiamo utilizzando. È importante sottolineare che i dati presentano differenze rispetto alla teoria definita in precedenza, e pertanto, sarà necessario adattare l'approccio teorico per affrontare tali discrepanze. Esploreremo le modalità di programmazione del riconoscimento, considerando gli aspetti pratici e le specificità dei dati considerati.

Esploreremo due approcci distinti per il riconoscimento, ciascuno caratterizzato da una metodologia specifica. Il primo, noto come "metodo delle secanti", rappresenta un'implementazione iniziale e semplificata. Il secondo metodo, più avanzato e in sintonia con la teoria esposta nel secondo capitolo è volto ad agevolare i calcoli e ottenere una rappresentazione più completa dei dati.

Il metodo delle secanti, sebbene più elementare, costituisce una base di partenza, mentre il secondo metodo, più sofisticato, si avvicina maggiormente alla teoria concettuale esposta in precedenza.

3.1 Metodo delle Secanti

Il metodo delle secanti rappresenta una strategia per ottenere una soluzione approssimata di un'equazione nella forma $f(x) = 0$. In questo contesto specifico, applicheremo questo metodo per la costruzione dello spazio tangente, avvalendoci di linee secanti.

La procedura coinvolge l'uso di rette secanti alla curva descritta dalla funzione $f(x)$, cercando di trovare i punti in cui queste secanti intersecano l'asse x (ovvero dove $f(x) = 0$) [2]. Questa iterazione continua delle secanti converge gradualmente verso una soluzione dell'equazione, fornendo una via approssimativa per determinare i punti in cui la funzione si annulla. Nel nostro caso specifico, applicheremo questo metodo per costruire lo spazio tangente, impiegando le linee secanti per catturare le variazioni locali tra le immagini. Questo approccio, sebbene approssimato, fornisce una base per sviluppare uno spazio tangente che rappresenta le caratteristiche di adattamento e deformazione tra le immagini considerate.

Non procederemo con l'implementazione del metodo delle secanti, in quanto presenta un'utilità limitata rispetto all'approccio che andremo ad approfondire nella sezione successiva. Il motivo di questa scelta risiede nel fatto che questo metodo richiede una conoscenza a priori delle deformazioni che saranno applicate agli elementi dell'insieme in uso. Al contrario, il prossimo metodo non richiede alcuna conoscenza preliminare, come dimostreremo nelle sezioni successive.

Questo approccio risulta particolarmente vantaggioso nell'ambito dell'addestramento di intelligenze artificiali. Nell'apprendimento automatico, dove un modello necessita di un ampio e dettagliato set di addestramento, è comune che tale insieme contenga etichette associate ad ogni elemento, identificandone le caratteristiche chiave per un addestramento ottimale. In questo contesto, tali etichette possono includere informazioni sulle deformazioni utilizzate per generare ciascuna immagine.

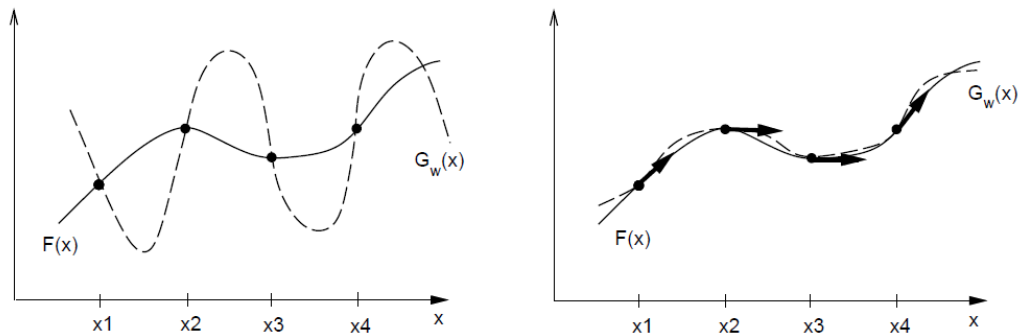


Figura 3.1: A sinistra, l'approssimazione di F con solo informazioni su F nei punti x_1, \dots, x_4 . A destra, dando anche informazioni sulla derivata di F nei punti si ottiene una migliore approssimazione.

Come evidenziato nella figura 3.1, l'aumento delle informazioni, che in questo caso rappresentano i vincoli che la funzione di approssimazione deve rispettare, conduce a un notevole miglioramento nella qualità dell'approssimazione [1]. Ciò si traduce in una migliore capacità di classificazione da parte dell'algoritmo, poichè un set di addestramento più ricco e dettagliato contribuisce a modellare in modo più accurato le relazioni tra le deformazioni e le caratteristiche delle immagini.

Esamineremo ora alcuni risultati derivati dall'applicazione di questo metodo di approssimazione. Inizieremo selezionando un'immagine che raffigura il numero 2 dal database USPS, caratterizzata dalle dimensioni 32×32 . Successivamente, applicheremo deformazioni come traslazione orizzontale e rotazione a tale immagine. L'obiettivo è comprendere come il metodo influenzi la rappresentazione dell'immagine originale nel contesto di queste deformazioni specifiche.

Questo processo consentirà di valutare l'efficacia dell'approssimazione nello spazio tangente nel catturare e modellare le variazioni indotte da traslazioni e rotazioni, contribuendo così a migliorare la comprensione della dinamica di adattamento tra le immagini.

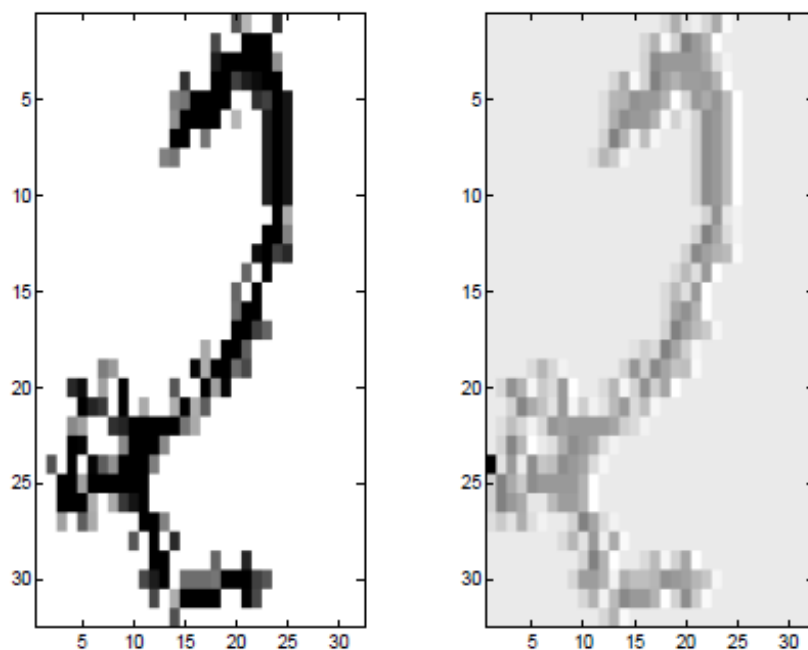


Figura 3.2: A sinistra, osserviamo un'autentica traslazione dell'immagine originale, mentre a destra vediamo l'immagine che rappresenta lo spazio tangente della traslazione, calcolato mediante il metodo delle secanti.

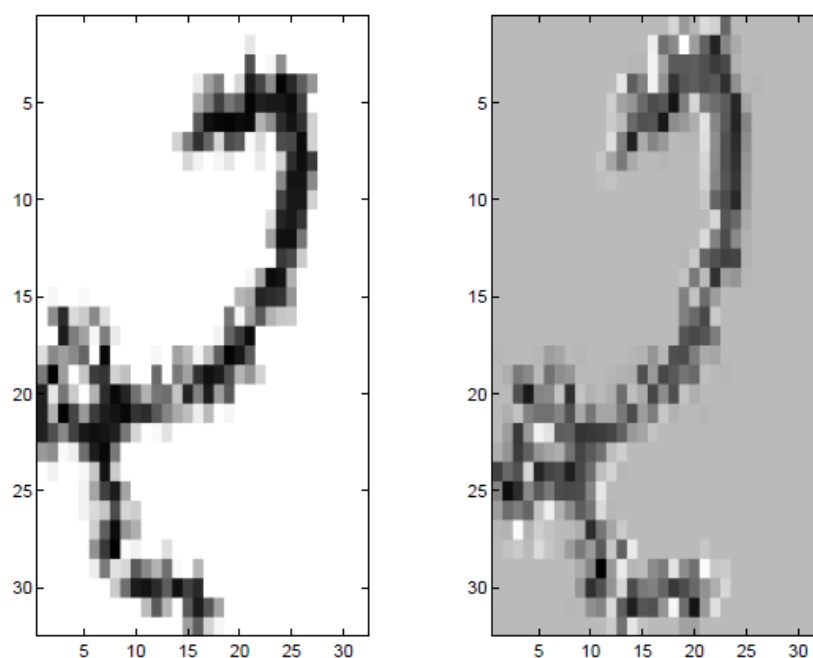


Figura 3.3: A sinistra, osserviamo un'autentica rotazione dell'immagine originale, mentre a destra vediamo l'immagine che rappresenta lo spazio tangente della rotazione, calcolato mediante il metodo delle secanti.

3.2 Strategie per l'identificazione dei vettori tangenti dei digit

Per realizzare il metodo più completo, occorre prima comprendere come trovare i vettori tangenti delle immagini che si vogliono analizzare, le quali sono di dimensione $n_1 \times n_2$. Per fare ciò, consideriamo i digits come funzioni continue: $f_p : \mathbb{R}^2 \rightarrow \mathbb{R}$ tale che $f_p \in C^1(\mathbb{R}^2, \mathbb{R})$ e con $f_p(i, j) = p_{ij}$, $\forall i \in \{1, \dots, n_1\}$, $\forall j \in \{1, \dots, n_2\}$. In questo modo, la rappresentazione continua coincide con le rappresentazioni discrete che già possediamo. Il grafico di questa rappresentazione "disegna" l'immagine del numero, simile alla matrice associata, ma in modo continuo senza transizioni da un pixel all'altro.

Prendiamo quindi una mappatura

$$\sigma : \mathbb{R}^{n_1 * n_2} \rightarrow C^1(\mathbb{R}^2, \mathbb{R}) \text{ t.c. } p \mapsto f_p.$$

Chiediamo inoltre che questa mappatura abbia una certa regolarità: che $\forall(x, y) \in \mathbb{R}^2$ sia differenziabile con continuità. Questa proprietà dice che una curva regolare verrà parametrizzata in un cammino di funzioni su cui è possibile fare il limite del rapporto incrementale.

Consideriamo un digit p e applichiamo una deformazione di un parametro α rappresentante una traslazione orizzontale. Otterremo così p^1 , nella nostra rappresentazione continua quindi avremo $f_{p^1}(x, y) = f_p(x + \alpha, y)$. Questa funzione può essere generalizzata per tutte le deformazioni che intendiamo applicare ai nostri digits.

Affinchè possiamo calcolare i vettori tramite lo sviluppo di Taylor troncato al primo ordine, come illustrato nella formula (2.4), dovremmo derivare le formule pertinenti per ciascuna tipologia di trasformazione che vogliamo applicare. Questo ci consentirà di ottenere una comprensione dettagliata del comportamento delle immagini sotto varie deformazioni, fondamentale per il nostro approccio di analisi e riconoscimento.

Data una funzione $f \in C^1(\mathbb{R}^2, \mathbb{R})$ otteniamo:

- **Traslazione Orizzontale:** deformazione sull'asse x

$$O_\alpha : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x + \alpha \\ y \end{pmatrix}$$

$$\frac{\partial}{\partial \alpha}(f(O_\alpha(x, y)))|_{\alpha=0} = \frac{\partial f}{\partial x}(x, y)$$

associamo alla traslazione orizzontale l'operatore differenziale $L_O = \frac{\partial}{\partial x}$.

- **Traslazione Verticale:** deformazione sull'asse y

$$V_\alpha : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ y + \alpha \end{pmatrix}$$

$$\frac{\partial}{\partial \alpha}(f(V_\alpha(x, y)))|_{\alpha=0} = \frac{\partial f}{\partial y}(x, y)$$

associamo alla traslazione verticale l'operatore differenziale $L_V = \frac{\partial}{\partial y}$.

- **Rotazione:** rotazione in senso orario di α gradi

$$R_\alpha : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \cos(\alpha) - y \sin(\alpha) \\ x \sin(\alpha) + y \cos(\alpha) \end{pmatrix}$$

$$\frac{\partial}{\partial \alpha} (f(R_\alpha(x, y)))|_{\alpha=0} = y \frac{\partial f}{\partial x}(x, y) - x \frac{\partial f}{\partial y}(x, y)$$

associamo alla rotazione in senso orario l'operatore differenziale

$$L_R = y \frac{\partial}{\partial x} - x \frac{\partial}{\partial y}.$$

- **Omotetie:** dilatazione o contrazione (Scaling)

$$S_\alpha : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x + \alpha x \\ y + \alpha y \end{pmatrix}$$

$$\frac{\partial}{\partial \alpha} (f(S_\alpha(x, y)))|_{\alpha=0} = x \frac{\partial f}{\partial x}(x, y) + y \frac{\partial f}{\partial y}(x, y)$$

associamo allo scaling l'operatore differenziale $L_S = x \frac{\partial}{\partial x} + y \frac{\partial}{\partial y}$.

- **Deformazioni Iperboliche Parallele:** dilatazione di un asse e contrazione dell'altro

$$PH_\alpha : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x + \alpha x \\ y - \alpha y \end{pmatrix}$$

$$\frac{\partial}{\partial \alpha} (f(PH_\alpha(x, y)))|_{\alpha=0} = x \frac{\partial f}{\partial x}(x, y) - y \frac{\partial f}{\partial y}(x, y)$$

associamo alla deformazione iperbolica parallela l'operatore differenziale

$$L_{PH} = x \frac{\partial}{\partial x} - y \frac{\partial}{\partial y}.$$

- **Deformazioni Iperboliche Diagonali:** dilatazione di una diagonale dei quadranti del piano e contrazione dell'altra

$$DH_\alpha : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x + \alpha y \\ y + \alpha x \end{pmatrix}$$

$$\frac{\partial}{\partial \alpha} (f(DH_\alpha(x, y)))|_{\alpha=0} = y \frac{\partial f}{\partial x}(x, y) + x \frac{\partial f}{\partial y}(x, y)$$

associamo alla deformazione iperbolica diagonale l'operatore differenziale

$$L_{DH} = y \frac{\partial}{\partial x} + x \frac{\partial}{\partial y}.$$

Una volta ottenuti gli operatori differenziali possiamo calcolare i vettori tangenti nel seguente modo:

- Si considera un digit di matrice P e vettore p e rappresentazione continua f_p ;
- Una volta scelta la deformazione da applicare, si prende l'operatore associato e lo si compone con f_p dalla destra, ottenendo così una funzione;
- Per calcolare la funzione ottenuta al punto precedente nei punti (i, j) , $\forall i \in \{1, \dots, n_1\}$, $j \in \{1, \dots, n_2\}$, creiamo una matrice $n_1 \times n_2$ con tali valori nelle rispettive posizioni. Questa matrice sarà associata al vettore tangente relativo alla deformazione utilizzata. Infine, si trasforma questa matrice in un vettore mettendo una colonna sotto l'altra.

Data la natura delle immagini, conformemente a quanto precedentemente spiegato, incontreremo alcune problematiche nell'implementazione di queste operazioni, in particolare:

1. $p = p(i, j)$ con le variabili i e j intere e non continue
2. p assume valori interi nell'intervallo da 0 a 255, rappresentando la colorazione di ogni pixel, e di conseguenza non ha valori continui.

Di conseguenza, $\frac{\partial}{\partial x}p(x, y)$ non può essere calcolato come la classica derivata. La gestione delle derivate e delle operazioni continue su queste rappresentazioni richiede un'attenzione particolare e l'adozione di approcci specifici per affrontare le peculiarità della discrezione nell'ambito delle immagini. Per superare queste difficoltà, possono essere sviluppate alcune strategie specifiche:

1. **Somma di impulsi:** è un'approssimazione utilizzata per rendere continua una funzione discreta. In questa procedura, la funzione viene

considerata come una somma di impulsi localizzati, rappresentati dalle delta di Dirac. Formalmente ciò si esprime come:

$$\tilde{p}(x, y) = \sum_{i,j=1}^{n_1, n_2} p(i, j) * \delta(x - i, y - j)$$

Dove $\delta(x-i, y-j)$ rappresenta la delta di Dirac centrata nei punti (i, j) . Questo approccio è particolarmente utile quando si tratta di funzioni definite su griglie discrete, come nel caso delle immagini pixelate.

2. **Convoluzione:** $\hat{p} = \tilde{p}(x, y) * g_\sigma$ dove g_σ è funzione gaussiana

Purtroppo, queste strategie non sono ottimali, in quanto possono rischiare di modificare eccessivamente l'immagine originale. La somma di impulsi e la convoluzione sono operazioni che, se non gestite con attenzione, possono alterare significativamente i valori dei pixel e quindi l'aspetto dell'immagine. Inoltre, sorgono problemi significativi anche dal punto di vista del costo computazionale di tali operazioni. Lavorando in una dimensione di 784, dovremmo calcolare 784 derivate per ogni vettore tangente, con altrettante operazioni di somme e prodotti. Per risolvere queste problematiche, in particolare per mitigare il costo computazionale, possiamo adottare una strategia che prevede l'uso di derivate finite centrate anzichè derivate reali per il calcolo dei vettori tangenti. Questa soluzione, pur approssimando ulteriormente i nostri calcoli, permette una gestione più efficiente delle operazioni necessarie per il calcolo dei vettori tangenti.

Il concetto alla base delle differenze finite è quello di usare il rapporto incrementale per approssimare l'operazione di derivata. Pertanto, calcoleremo le differenze finite centrate nel seguente modo: con h piccolo;

$$\frac{\partial f}{\partial x}(x, y) \approx \frac{f(x + h, y) - f(x - h, y)}{2h}$$

$$\frac{\partial f}{\partial y}(x, y) \approx \frac{f(x, y + h) - f(x, y - h)}{2h}.$$

Con h sufficientemente piccolo, otteniamo un'approssimazione accurata delle derivate parziali della funzione nei punti desiderati. Questo approccio consente di ottenere una stima del tasso di variazione della funzione rispetto alle

singole variabili, pur semplificando il processo computazionale rispetto alle derivate esatte.

Per ottimizzare ulteriormente il costo computazionale, possiamo calcolare direttamente le derivate utilizzando i valori della matrice. Questo approccio consente di evitare la necessità di creare una funzione per l'interpolazione del digit, poichè i valori richiesti sono già contenuti nella matrice o nel vettore che lo compone.

Prendendo $h = 1$, calcoliamo le derivate parziali nel seguente modo:

$$p_x(i, j) \approx \frac{p(i+1, j) - p(i-1, j)}{2}$$
$$p_y(i, j) \approx \frac{p(i, j+1) - p(i, j-1)}{2}.$$

In questo modo, calcoliamo i vettori tangenti utilizzando un numero estremamente ridotto di operazioni di somme e prodotti, che sono computazionalmente efficienti per un calcolatore.

3.3 Algoritmo per il calcolo della Distanza Tangente

Avendo compreso come calcolare in modo ottimale i vettori tangenti per ogni trasformazione, per il tipo specifico di oggetti che stiamo trattando, dobbiamo ora definire l'algoritmo per il calcolo della distanza tangente. Trovare questa distanza corrisponde a individuare la distanza minima tra i rispettivi spazi tangenti nei punti associati ai digit originali. Tale calcolo confronta direttamente due digit tra loro, e pertanto, per la nostra classificazione basata sul set di addestramento, considereremo il centroide come rappresentante di ciascuna cifra da 0 a 9.

Passo 1: calcolare i vettori tangenti di un digit

Avendo la nostra immagine di dimensione 28×28 otteniamo:

- $v_x(i, j) = \frac{v(i+1, j) - v(i-1, j)}{2}$

- $v_y(i, j) = \frac{v(i, j+1) - v(i, j-1)}{2}$

Questi sono i vettori tangenti associati alle traslazioni orizzontale e verticale. Le altre deformazioni, ovvero la rotazione v_r , lo scaling v_s , la deformazione iperbolica parallela v_{ph} e quella diagonale v_{dh} , possono essere ottenute dagli operatori differenziali definiti nella sezione precedente:

- $v_r(i, j) = j \cdot v_x(i, j) - i \cdot v_y(i, j)$
- $v_s(i, j) = i \cdot v_x(i, j) + j \cdot v_y(i, j)$
- $v_{ph}(i, j) = i \cdot v_x(i, j) - j \cdot v_y(i, j)$
- $v_{dh}(i, j) = j \cdot v_x(i, j) + i \cdot v_y(i, j)$

Per evitare che gli indici superino i limiti sia dalla parte iniziale che finale, dobbiamo prendere alcune precauzioni nel calcolo. Ad esempio, è possibile impostare un valore nullo nel caso in cui l'indice ecceda i limiti. Infine, vettorizziamo queste matrici ottenendo così 6 vettori, ognuno di dimensione 784.

Passo 2: confronto con i centroidi del training set

Il centroide di un training set di immagini per ogni cifra è definito come il vettore medio delle immagini contenute in tale insieme. Per calcolare il centroide di ogni cifra, sommiamo i valori corrispondenti dei pixel di tutte le immagini nel training set e successivamente dividiamo ciascuna somma per il numero totale di immagini. Considerando ogni training set come $\{x_1^{(i)}, \dots, x_{n_i}^{(i)}\}$, dove $x_k^{(i)}$ rappresenta il vettore della k -esima immagine relativa alla cifra i , il calcolo del centroide è espresso come:

$$C^{(i)} = \frac{1}{n_i} \sum_{k=1}^{n_i} x_k^{(i)},$$

dove $C^{(i)}$ rappresenta il centroide per la cifra i .

Una volta calcolati i vettori tangenti di tutti i centroidi del training set e del digit da classificare, possiamo procedere con la classificazione. Indicando con p il vettore del digit che vogliamo classificare e con q_i l' i -esimo centroide vettorizzato otteniamo:

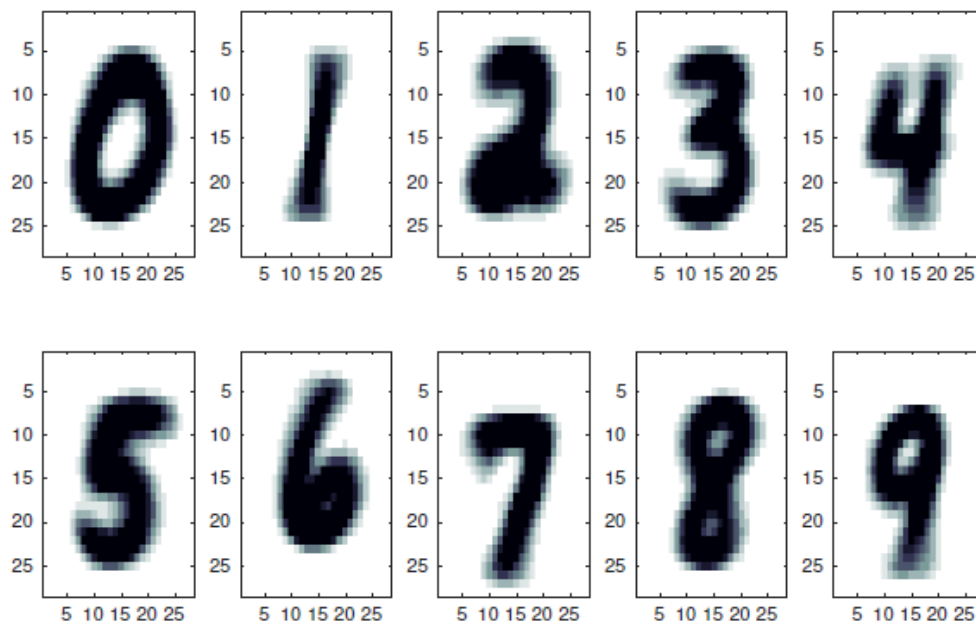


Figura 3.4: Rappresentazione grafica dei centroidi del database MNIST.

- $L_p = [v_x(p) \quad v_y(p) \quad v_r(p) \quad v_s(p) \quad v_{ph}(p) \quad v_{dh}(p)]$
- $L_{q_i} = [v_x(q_i) \quad v_y(q_i) \quad v_r(q_i) \quad v_s(q_i) \quad v_{ph}(q_i) \quad v_{dh}(q_i)]$

Calcoliamo quindi $\forall i \in \{0, \dots, 9\}$ la distanza tangente tra le due immagini mediante minimi quadrati come nella formula (2.9):

$$d_i = \min_{\alpha \in \mathbb{R}^{12}} \|(p - q_i) - [-L_p \quad L_{q_i}] \cdot \alpha\|.$$

Trovando $\tilde{i} = \arg \min_i d_i$ si avrà la classificazione del digit nella cifra a lui più vicina.

Il numero di vettori tangenti utilizzati è direttamente proporzionale al numero di deformazioni che vogliamo controllare. Un maggior numero di trasformazioni utilizzate porta a un miglior riconoscimento, poichè avremo più istanze della stessa immagine, ciascuna leggermente modificata. Questo approccio consente al modello di apprendere e generalizzare meglio, migliorando così le prestazioni di riconoscimento per variazioni nelle deformazioni delle cifre.

3.4 Analisi dei Risultati

A questo punto, possiamo implementare un algoritmo che integri la classificazione mediante SVD, ma in caso di un alto residuo, si basi sulla distanza tangente per valutare come ciò possa migliorare la classificazione. Per questo algoritmo, utilizzeremo sempre 5000 immagini come set di addestramento e 500 immagini come set di test per ogni cifra, come nel caso dell'algoritmo basato solo su SVD, in modo da poter confrontare le percentuali di accuratezza.

cifra	0	1	2	3	4	5	6	7	8	9
SVD	98.6	99.0	90.8	94.2	93.2	89.4	93.8	89.6	87.4	90.2
D.T.	98.8	99.2	92.4	94.8	95.4	93.6	94.8	92.2	89.6	90.4

Abbiamo osservato un miglioramento nel riconoscimento per tutte le cifre. In particolare, si nota un lieve miglioramento per l'uno e lo zero, che già avevano un'accuratezza oltre il 98%, ma una differenza significativa per quanto riguarda il cinque e il sette, rispettivamente il 4.2% e 2.6%.

Avendo riscontrato un miglioramento nella classificazione rispetto a quella precedentemente eseguita, proviamo ora a confrontare la distanza tangente con altre misure di distanza su un dataset diverso per valutare i risultati ottenuti.

Esamineremo alcuni risultati utilizzando un set di 50 immagini, approssimativamente 5 per ogni cifra, dal database USPS. Confronteremo tre tipi di distanze: distanza euclidea, distanza tangente unilaterale e distanza tangente bilaterale. Queste distanze saranno impiegate per individuare i k -nearest neighbors di un'immagine. L'algoritmo k -Nearest Neighbors (k -NN) è utilizzato per individuare le k immagini più simili (vicine) a partire da un'immagine di riferimento.

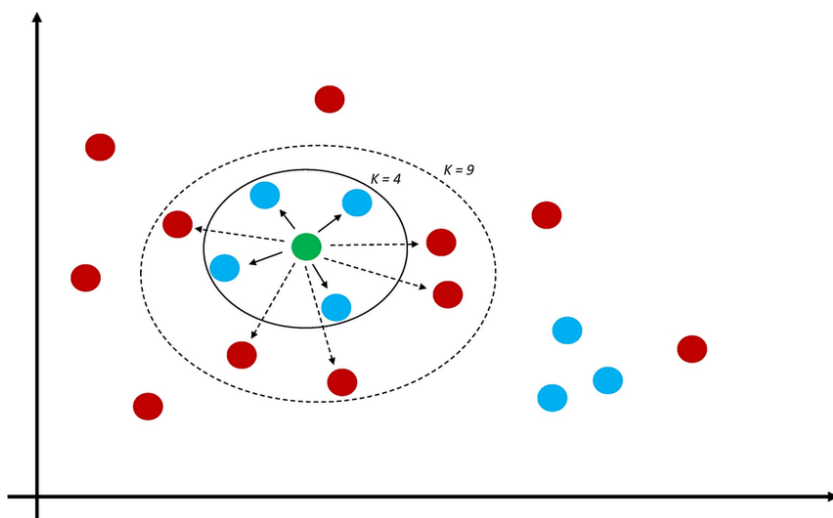


Figura 3.5: Considerando le immagini come punti nello spazio, l'algoritmo k -Nearest Neighbors cerca i k punti più vicini ad un punto specifico, rappresentato nella figura dal punto verde.



Figura 3.6: L'immagine mostra i 4-nearest neighbors mediante distanza euclidea. Nella prima colonna troviamo l'immagine che abbiamo dato come riferimento.



Figura 3.7: L'immagine mostra i 4-nearest neighbors mediante distanza tangente bilaterale. Nella prima colonna troviamo l'immagine che abbiamo dato come riferimento.

Dalle Figure 3.6 e 3.7, osserviamo che nonostante la dimensione ridotta del nostro training set, la distanza tangente bilaterale riesce a classificare più immagini appartenenti alla stessa cifra come vicine rispetto a quella euclidea. In particolare, questo è evidente per le cifre 2, 5 e 9.

Possiamo confrontare anche queste distanze nella creazione di dendrogrammi utilizzando l'approccio single-linkage. I dendrogrammi single linkage sono diagrammi gerarchici che rappresentano le relazioni di similarità tra i dati. In questo contesto, l'algoritmo single linkage collega i cluster (insiemi di immagini) in base alla distanza minima tra i punti più vicini di due cluster. In altre parole, due cluster sono collegati da un link il cui valore rappresenta la distanza minima tra i loro punti più vicini. Questo processo viene ripetuto fino a quando tutti i punti sono collegati nel dendrogramma, fornendo una visualizzazione gerarchica della struttura di similarità tra i dati.

Utilizzeremo le distanze euclidea, tangente unilaterale, definita nella Equazione (2.5), e tangente bilaterale, definita nell'Equazione (2.9), per generare

i dendrogrammi.

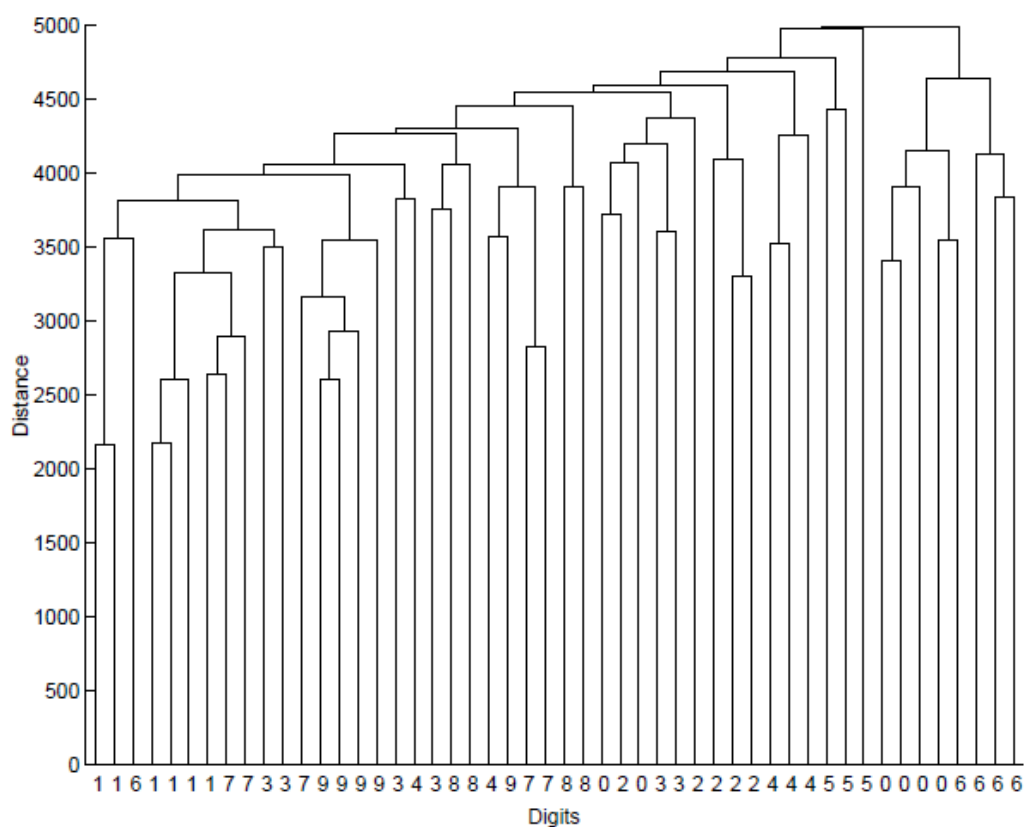


Figura 3.8: Il dendrogramma creato utilizzando la distanza euclidea mostra che nessuna cifra è stata unita, ma sono tutte leggermente separate.

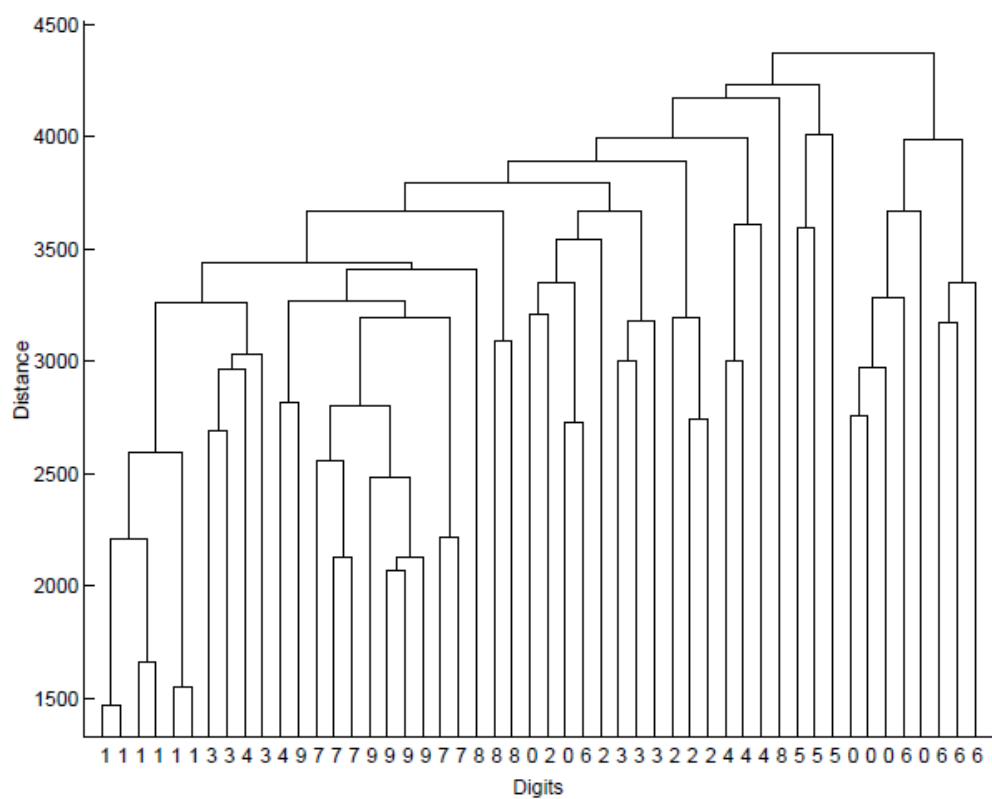


Figura 3.9: Il dendrogramma creato utilizzando la distanza tangente unilaterale è in grado di formare un cluster per le immagini che rappresentano il numero uno e mostra un miglioramento nei cluster delle altre cifre, sebbene non riesca ad unirle tutte.

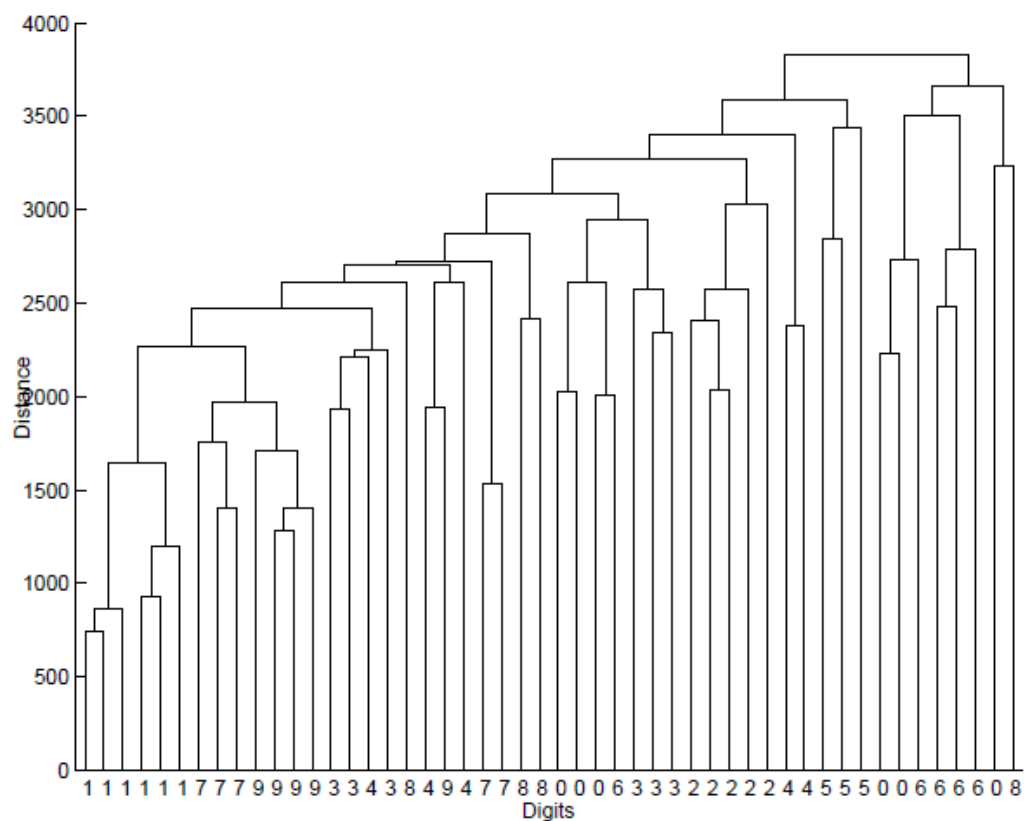


Figura 3.10: Il dendrogramma creato utilizzando la distanza tangente bilaterale è il migliore tra i tre. È in grado di formare cluster distinti per le immagini che rappresentano i numeri uno e due, e mostra un notevole miglioramento nei cluster delle altre cifre, come ad esempio i sei e i nove.

Conclusioni

Nel corso di questa analisi, abbiamo esaminato dettagliatamente l'applicazione della distanza tangente nel contesto del riconoscimento di cifre scritte a mano, confrontandola con altre metriche di distanza e il più semplice approccio SVD. Partendo dai risultati ottenuti utilizzando un set di addestramento di 5000 immagini e un set di test di 500 immagini per ciascuna cifra del database MNIST, è emerso che l'algoritmo basato sulla distanza tangente ha portato a miglioramenti significativi nella classificazione rispetto alle metriche di distanza convenzionali, in particolare rispetto alla distanza euclidea. La forza principale della distanza tangente risiede nella sua capacità di catturare variazioni locali nelle immagini, rendendola particolarmente efficace nel riconoscimento di cifre che possono subire deformazioni specifiche. Questo è in contrasto con l'approccio SVD, che, sebbene sia un metodo potente, può rivelarsi meno robusto in presenza di deformazioni locali.

Nonostante i risultati positivi ottenuti attraverso l'utilizzo della distanza tangente nel riconoscimento di cifre scritte a mano, è importante considerare alcuni aspetti negativi associati a questo approccio: l'algoritmo basato sulla distanza tangente richiede il calcolo delle derivate rispetto ai parametri della trasformazione, comportando un aumento della complessità computazionale, specialmente quando si tratta di dataset più ampi. Inoltre, per ottenere risultati ottimali, la distanza tangente richiede un set di addestramento ampio e rappresentativo che copra variazioni nelle immagini. In contesti in cui la raccolta di dati risulta complicata, potrebbe essere difficile ottenere prestazioni ottimali.

Bibliografia

- [1] Grégoire Montavon, Geneviève B. Orr, Klaus-Robert Müller. *Neural Networks: Tricks of the Trade, Second Edition*, pp. 239-274, Springer 2012.
- [2] Jen-Mei Chang, Michael Kirby, Lucas Krakow, Josh Ladd, Ethan Murphy. *Classification of Images with Tangent Distance*, tech. rep., Colorado State University, 2004.
- [3] Brunton SL, Kutz JN. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, pp.3-46, Cambridge: Cambridge University Press, 2019.

Ringraziamenti

Desidero esprimere la mia profonda gratitudine a tutte le persone che hanno contribuito al completamento di questa tesi. Il mio sincero ringraziamento va alla Prof.ssa Valeria Simoncini, per la guida fondamentale per il successo di questo lavoro.

Un sentito ringraziamento va anche ai miei genitori e ai membri della mia famiglia, che mi hanno sostenuto in ogni fase del mio percorso accademico. La loro costante presenza e il loro sostegno morale sono stati indispensabili per il mio successo.

Un ringraziamento speciale va ai miei amici più stretti, con cui ho passato dei momenti che non dimenticherò. La vostra amicizia, il confronto e la condivisione di idee hanno reso questa avventura un'esperienza speciale.