ALMA MATER STUDIORUM - UNIVERSITA' DI BOLOGNA
SEDE DI CESENA

FACOLTA' DI SCIENZE MATEMATICHE, FISICHE E NATURALI
CORSO DI LAUREA MAGISTRALE IN SCIENZE E TECNOLOGIE INFORMATICHE

MODEL-BASED COMPRESSED SENSING PER UN NUOVO ALGORITMO PER IL RICONOSCIMENTO DEI VOLTI

Tesi di laurea in

METODI AVANZATI DI ELABORAZIONE DI IMMAGINI

Relatore

Prof.ssa Laura Montefusco

Co-relatore

Dott.ssa Damiana Lazzaro

Presentata da

Andrea Pierantoni

Sessione III
Anno Accademico 2011/2012

A Manuela, Pietro e Giulia:
i tre raggi che hanno reso la mia selva meno oscura.

Indice

Introduzione **7**

1 Riconoscimento automatico dei volti **11**

0.1 Perché usare il volto per il riconoscimento 13

0.2 Possibili applicazioni 14

1 Definizione del problema 16

1.1 Principali problematiche 18

2 Estrazione delle feature 20

2.1 Metodi lineari basati sull'aspetto globale del volto . . . 24

2.2 Metodi basati su kernel non lineari 32

2.3 Metodi basati sull'aspetto locale delle componenti del volto 32

3 Principlali tecniche di classificazione 34

3.1 Nearest Neighbour (NN) 34

3.2 Nearest Subspace (NS) 35

3.3 Support Vector Machine (SVM) 36

2 Riconoscimento di volti basato sulla rappresentazione sparsa **39**

1 Introduzione al compressed sensing 40

1.1 Formulazione del compressed sensing 42

2 Classificazione basata sulla rappresentazione sparsa (SRC) . . 44

2.1 Modellazione matematica 45

2.2 Robustezza in presenza di rumore 48

2.3 Algoritmo di classificazione 49

2.4 Algoritmo SRC ed estrazione delle feature 50

2.5 Robustezza alle occlusioni 54

2.6 Meccanismo di validazione 57

3 Risoluzione di approcci non convessi al problema del compressed sensing 63

1 Funzioni sparsificanti non convesse 64

2 Metodi iterativi pesati per problemi di minimizzazione non convessi 66

3 Una nuova strategia di minimizzazione 68

3.1 Strategia di penalizzazione basata sullo splitting 69

3.2 Algoritmo WNFCS 70

3.3 Regolazione dei parametri della funzione sparsificante . 73

4 Model-based compressed sensing 74

4.1 Model-based WNFCS 75

4 Sperimentazione e analisi dei risultati 77

1 Parametrizzazione dell'algoritmo WNFCS 78

2 Test eseguiti senza validazione 79

2.1 Extended Yale B database 80

2.2 AR database 86

2.3 Robustezza nei confronti delle occlusioni 91

3 Test eseguiti attivando il meccanismo di validazione 95

3.1 False reiezioni in assenza di impostori 96

3.2 Test di validazione con impostori 98

Conclusioni e sviluppi futuri 101

Bibliografia 103

Introduzione

Lo scopo di questa tesi è quello di presentare una nuova infrastruttura algoritmica per la risoluzione di approcci non convessi al problema del compressed sensing e di valutarne le prestazioni nell'ambito della face recognition, integrandola all'interno dello schema di un recente classificatore basato sulla rappresentazione sparsa.

La teoria del compressed sensing, sfruttata inizialmente per la realizzazione di nuovi protocolli di acquisizione e compressione dei segnali, ha riscosso nel tempo sempre più successo anche in numerosi altri campi applicativi, fra cui appunto il riconoscimento di volti. In particolare, negli ultimi anni è stato proposto un classificatore basato sulla rappresentazione sparsa (SRC), il cui obiettivo è quello di identificare il volto fornito in input descrivendolo come combinazione lineare di tutte le immagini contenute in un dizionario più che completo, coincidente con l'intero training set. In questo scenario, la soluzione al problema corrispondente risulterà essere tipicamente sparsa, presentando pochi coefficienti diversi da zero in corrispondenza delle immagini di training relative allo stesso individuo rappresentato nell'immagine di test. Questo approccio si differenzia notevolmente dai metodi di classificazione classici e permette inoltre l'introduzione di opportuni meccanismi per gestire al meglio la presenza di occlusioni e per rafforzare il processo di validazione.

Il classificatore SRC cerca di ricavare i coefficienti sparsi della soluzione attraverso l'approccio classico di minimizzazione della norma l_1. Tuttavia, recenti ricerche nell'ambito del compressed sensing hanno dimostrato che alcune funzioni sparsificanti non convesse riescono a produrre soluzioni più sparse - e quindi di maggiore qualità - anche a partire da un numero di misurazio-

ni lineari inferiore. Purtroppo, però, l'introduzione di queste nuove funzioni aumenta notevolmente la difficoltà del problema del compressed sensing, rendendo necessario, per la sua risoluzione, l'impiego di tecniche di rilassamento e di metodi iterativi pesati (come ad esempio l'IRLS e l'IRl_1). L'inconveniente di questi algoritmi è, d'altra parte, quello di risultare particolarmente onerosi dal punto di vista computazionale. È dunque in questo contesto che s'inserisce il nuovo framework algoritmico analizzato, che prende il nome di WNFCS - da *Weighted Non-linear Filters for Compressed Sensing* - e la cui innovazione consiste nella fusione della tecnica iterativa pesata, già citata, con un metodo di penalizzazione basato sullo splitting. Questo nuovo approccio riesce dunque a ridurre il numero di iterazioni necessarie per ottenere la convergenza; inoltre, la sua struttura estremamente generale permette di sfruttare la funzione non convessa che si ritiene più adeguata e di introdurre anche opportuni modelli di sparsità.

Al fine dunque di contestualizzare il lavoro svolto, si fornirà nel primo capitolo una panoramica generale del problema del riconoscimento dei volti, delineandone i numerosi domini applicativi ed evidenziandone le principali problematiche. Verrà, inoltre, mostrata la struttura di un tipico sistema di face recognition, passando poi in rassegna i metodi classici di estrazione delle feature e di classificazione.
Nel secondo capitolo, saranno inizialmente presentate le basi matematiche del compressed sensing, per poi descrivere nel dettaglio le caratteristiche e le potenzialità del nuovo classificatore basato sulla rappresentazione sparsa.
Nel terzo capitolo, si proseguirà la trattazione esponendo le fondamenta matematiche e algoritmiche su cui poggia il framework WNFCS, fornendo la definizione di modello di sparsità strutturata e dimostrando come questo nuovo approccio possa essere introdotto nello schema dell'algoritmo.
Il quarto capitolo riguarderà invece la fase di sperimentazione e analisi dei risultati e dimostrerà come il nuovo framework analizzato permetta al classificatore SRC di ottenere quasi sempre risultati migliori rispetto a quelli raggiunti con l'uso della classica minimizzazione l_1.
Al termine verranno riassunte le considerazioni complessive sulle prestazioni del sopra citato classificatore e verranno proposti alcuni suoi possibili sviluppi

futuri.

Capitolo 1

Riconoscimento automatico dei volti

Imago animi vultus, indices oculi.
- Cicerone, De oratore -

Il problema del riconoscimento automatico del volto o *face recognition* rappresenta una delle sfide più interessanti nell'ambito dell'analisi di immagini digitali, oltreché in quello di visione artificiale, e, in quanto tale, negli ultimi anni è stato oggetto di grande attenzione da parte dei ricercatori, anche a causa della molteplicità di domini applicativi cui si rivolge. Una prova evidente della crescente partecipazione della comunità scientifica a questo settore di ricerca è riscontrabile nel gran numero di conferenze e pubblicazioni internazionali ad esso dedicate; si ricorda, a titolo di esempio, la IEEE International Conference on Automatic Face and Gesture Recognition (IEEE FG), che dal 1995 ad oggi costituisce il principale punto d'incontro per tutti gli studiosi impegnati in questo ambito e che raggiungerà, proprio nell'Aprile 2013, la sua decima edizione.

Questo grande interesse nei confronti del problema di face recognition non è rimasto circoscritto soltanto all'ambiente scientifico accademico, ma, visti i suoi promettenti sviluppi, si è diffuso anche in ambito commerciale, dove grandi aziende di Information Technology come IBM, Samsung, Google, Apple e Microsoft - solo per citarne alcune fra le più famose -, in tempi recenti, hanno investito importanti somme nello sviluppo di sistemi di riconoscimento automatico di volti sempre più corretti, robusti ed affidabili. I possibili

impieghi di questi sistemi nel campo della sicurezza hanno suscitato, inoltre, l'interesse dei governi, al punto che, il National Institute of Standard and Technology (NIST) - agenzia del dipartimento di commercio degli Stati Uniti - ha istituito a partire dal 2000 il cosiddetto Face Vendor Recognition Test (FRVT), allo scopo, da una parte, di fornire una valutazione delle tecnologie di face recognition impiegate nelle applicazioni commerciali e nei prototipi all'avanguardia e, dall'altra, di suggerire delle linee guida per gli sviluppi futuri.
Proprio dalla necessità di valutare e confrontare le performance dei diversi algoritmi proposti, nel corso degli anni sono stati creati numerosi database pubblici di volti, tra i quali si ricordano quelli più utilizzati: FERET, Yale B, Extended Yale B, AR, ORL, PIE e Multi-PIE.

Il problema della face recognition, inoltre, non costituisce soltanto un ramo di ricerca della *computer science*, ma rappresenta un più ampio settore interdisciplinare della conoscenza, che coinvolge, fra le altre discipline, anche le neuroscienze e la psicologia. La ragione di questo interesse condiviso risiede nel fatto che il riconoscimento dei volti, per quanto sia comunemente considerata un'operazione quotidiana e scontata del nostro cervello, risulta invece essere un processo cognitivo complesso, il cui funzionamento non è stato ancora del tutto compreso. Ciò che sappiamo in merito è che il cervello deve identificare l'oggetto che vediamo come un volto, indipendentemente dalla dimensione e dall'angolo di visualizzazione; riconoscerne l'espressione, decodificando una particolare disposizione di occhi e bocca; ed, infine, accedere alla memoria per verificare se il viso è famigliare.
Il riconoscimento facciale, per di più, è una componente estremamente importante dell'interazione sociale umana e sembra che il nostro cervello abbia sviluppato un suo centro di elaborazione speciale per portare a termine questo compito così complesso e cruciale. Studi molto recenti nel campo del *brain-imaging* hanno infatti dimostrato che, quando una persona guarda in direzione di un volto, attiva particolari aree del cervello, che restano invece inermi qualora l'oggetto in esame sia di qualche altra natura - ad esempio, una casa o un'automobile. Le scoperte in questo senso sono state inoltre confermate da prove sperimentali del fatto che, se una di queste zone subisce

un trauma che ne compromette la funzionalità, il soggetto colpito accuserà il disturbo cognitivo noto come *prosopagnosia*, ovvero non sarà più in grado di distinguere i visi dei suoi conoscenti da quelli degli sconosciuti [6].
Nonostante le diverse prospettive attraverso cui le varie scienze coinvolte prendono in esame il problema del riconoscimento di volti e nonostante la consapevolezza di non poter riprodurre fedelmente, attraverso l'uso della tecnologia, la straordinaria capacità del cervello umano nel realizzare questo compito, il sentimento comune porta comunque a sperare che i progressi ottenuti in un settore siano d'aiuto anche per gli altri, in un continuo scambio di conoscenze, mirato ad una definitiva comprensione dei meccanismi sottesi al processo di face recognition.

0.1 Perché usare il volto per il riconoscimento

Negli ultimi anni, i sistemi biometrici si sono rivelati l'opzione più promettente per il riconoscimento degli individui. Ciò si deve al fatto che, anziché autenticare le persone e garantire loro l'accesso a domini, fisici o virtuali, attraverso l'uso di password, PIN, smart card, tessere identificative, chiavi e così via, questi metodi esaminano le caratteristiche fisiologiche e/o comportamentali di un soggetto per determinarne e/o accertarne l'identità.
Inoltre, password e PIN sono difficili da ricordare e possono essere rubati o indovinati; chiavi e simili possono essere perse, dimenticate, trafugate o duplicate; mentre le tessere magnetiche possono rovinarsi e diventare illeggibili. D'altra parte, i tratti biologici di un individuo non possono essere né persi, né dimenticati, né rubati, né facilmente ricostruiti.
Entrando più nel dettaglio, le tecnologie di riconoscimento biometriche includono l'identificazione basata su caratteristiche fisiologiche - come il volto, le impronte digitali, la geometria delle dita e delle mani, l'iride, la retina e la voce - e tratti caratteriali - come l'andatura, la firma e le dinamiche di battitura.
Fra tutte queste possibilità, offerte dai sistemi biometrici, il riconoscimento del volto risulta essere particolarmente vantaggioso per i seguenti motivi:

- **Non richiede necessariamente la collaborazione dell'utente:** per acquisire l'immagine del volto di un individuo, è sufficiente che egli si trovi nel raggio d'azione di una macchina fotografica o di una videocamera, mentre, ad esempio, per la scansione delle impronte digitali o dell'iride, l'utente deve posizionare la sua mano su un poggiamano o rimanere immobile in piedi di fronte a uno scanner oculare.
  Questa caratteristica risulta particolarmente vantaggiosa per le applicazioni nell'ambito della sicurezza e della videosorveglianza.

- **È economico:** le immagini del volto possono essere catturate utilizzando delle comuni macchine fotografiche, mentre, ad esempio, per il riconoscimento dell'iride e della retina è necessario dotarsi di strumentazioni piuttosto costose.

- **Non è intrusivo:** molte tecniche biometriche richiedono un uso condiviso della strumentazione necessaria per catturare le caratteristiche biologiche degli individui, favorendo in questo modo la trasmissione di germi e impurità fra i diversi utenti; le tecnologie di face recognition sono, invece, totalmente non intrusive e non espongono in alcun modo le persone coinvolte a rischi che possano compromettere la loro stessa salute [7].

0.2 Possibili applicazioni

Il riconoscimento dei volti è utilizzato principalmente per ottenere i seguenti obiettivi:

1. **Verifica** *(corrispondenza uno-a-uno)*: lo scopo, in questo caso, è quello di appurare che un individuo, di cui non è certa l'identità, sia veramente chi afferma di essere.

2. **Identificazione** *(corrispondenza uno-a-molti)*: data l'immagine di un individuo sconosciuto, s'intende determinare la sua identità, confron-

tando un'immagine del suo volto con un database di immagini che rappresentano soggetti noti.

Esistono numerosi campi applicativi in cui le tecniche di face recognition possono essere utilizzate per il raggiungimento degli scopi sopraelencati; i più comuni dei quali sono sintetizzati in Tabella 1.1 [8].

Area	**Applicazione specifica**
Intrattenimento	Videogiochi, realtà virtuale, programmi di allenamento
	Interazione uomo-macchina
Smart card	Patenti di guida, programmi assistenziali
	Controllo dell'immigrazione, documenti d'identità nazionali, passaporti
	Individuazione di frodi previdenziali
Sicurezza dell'informazione	Parental control per la TV, accesso personale ad un dispositivo
	Sicurezza di applicazioni e database, codifica di file
	Sicurezza di rete, accesso a Internet, informazioni mediche
	Sistemi di trading
Applicazione della legge e sorveglianza	Videosorveglianza avanzata, controllo di sistemi a circuito chiuso
	Controllo degli accessi agli edifici, analisi post-evento criminoso
	Sistemi anti-taccheggio, supporto all'investigazione

Tabella 1.1: Applicazioni tipiche del riconoscimento di volti

In aggiunta a queste applicazioni, le tecniche attuali sfruttate nelle tecnologie di face recognition sono state rielaborate e utilizzate in altri contesti pratici connessi al riconoscimento di volti come, ad esempio, la classificazione basata sul genere, il riconoscimento delle espressioni, il riconoscimento e il tracciamento delle caratteristiche del volto. Ognuna di queste tecniche ha una propria utilità peculiare, che la rende vantaggiosa in diversi domini applicativi: per esempio, il riconoscimento delle espressioni può essere utilizzato in campo medico per il monitoraggio dello stato di un paziente in terapia intensiva, mentre il riconoscimento di componenti particolari del volto può essere sfruttato per seguire gli occhi di un automobilista e determinarne stanchezza e stress. Infine, si ricorda che spesso le tecniche di face recognition vengono utilizzate congiuntamente ad altri metodi di riconoscimento biometrici, al fine di migliorarne le performance.

1 Definizione del problema

Il problema del riconoscimento automatico dei volti può essere formulato in questo modo: data un'immagine - o una sequenza video - che immortala il viso di un individuo e dato un database di volti di soggetti noti, come possiamo verificare o determinare l'identità della persona, ripresa nell'immagine di input?

Dalla definizione sopra riportata, è facile intuire come il problema di face recognition rappresenti, in realtà, un'applicazione specifica del più generale problema di *pattern recognition*, che consiste nell'analisi e identificazione di un oggetto di interesse - il *pattern* appunto - all'interno di dati grezzi, al fine di eseguirne la classificazione.
Come è noto, per problemi di questo genere è indispensabile disporre di un database, composto da un certo numero di esempi per ogni classe - detto *training set* - e di un insieme di esempi non ancora classificati - che costituiranno il cosiddetto *test set*. Nel caso particolare del riconoscimento di volti il database di training racchiude generalmente al suo interno più immagini del volto di uno stesso soggetto ed ogni individuo corrisponde a una classe. D'altra parte, nel test set saranno collezionate le immagini dei volti di quegli individui di cui si vuole accertare l'identità e che, proprio per questo motivo, potrebbero anche non essere raffigurati all'interno del training set.
Ci si può dunque trovare di fronte a due diversi scenari:

- Nel training set sono presenti alcune immagini del volto della persona da identificare: allora il risultato atteso consiste nella corretta classificazione dell'individuo.

- Nel training set non è contenuta alcuna immagine del volto della persona in esame: in questo caso, il risultato atteso è il rifiuto dell'immagine; ovvero il sistema ammette di non poter procedere con la classificazione, poiché l'individuo non appartiene a nessuna delle classi conosciute.

Il primo passo da compiere per facilitare il processo di classificazione è quello di aumentare il più possibile la separazione fra le classi. Le metodologie di

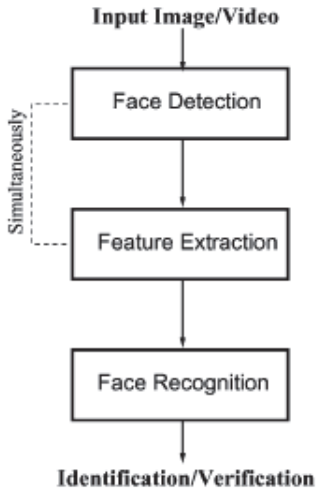


Figura 1.1: Configurazione di un generico sistema di face recognition

feature extraction servono proprio a questo e raggiungono il loro obiettivo estraendo, dalle immagini dei volti che compongono il training set, alcune caratteristiche particolari (*feature*), scartando invece tutto il resto. Queste tecniche cercano, inoltre, di minimizzare quanto più possibile il numero di feature necessarie, in modo da rendere più veloce la successiva fase di confronto e identificazione.

La risoluzione del problema di face recognition è strutturata generalmente in tre fasi principali (Figura 1.1): la prima fase, detta di *face detection*, consiste nel rilevamento del volto all'interno dell'immagine e nella sua segmentazione, ovvero la separazione del volto dallo sfondo; in seguito è prevista la fase di *feature extraction*, di cui si è già parlato sopra; infine, avviene la fase di *face recognition* vera e propria, che consiste nella classificazione del volto.

L'input del sistema è dato da una generica immagine in cui possono essere rappresentati uno o più volti, o anche nessuno. È dunque in questo frangente che si rivela indispensabile l'utilizzo di algoritmi di face detection. Nel corso del tempo, sono stati proposti diversi approcci per portare a termine questa fase, ma, il metodo di face recognition considerato in questa tesi assume che le immagini in input siano già state segmentate e allineate, ragion per cui si passerà ad esaminare in maggior dettaglio le due fasi successive.

Lo stadio di feature extraction, come già accennato, prevede l'estrazione di alcune caratteristiche peculiari, presenti nell'immagine del volto, che permet-

tano di facilitare l'individuazione della classe corretta. Le feature ricavate devono essere il più possibile indipendenti tra di loro, oltreché estremamente discriminative; inoltre, generalmente si tenta di minimizzare il loro numero, in modo da sveltire la successiva fase di confronto e classificazione.
La quantità d'informazione contenuta nell'immagine originale di un volto è solitamente molto elevata, il che significa che lo spazio di rappresentazione ad essa relativo ha una dimensione tale da rendere troppo onerosa l'elaborazione, necessaria per la classificazione. Fortunatamente, però, questa informazione è spesso anche estremamente ridondante ed è quindi possibile rimuovere da essa tutto ciò che non implica una conoscenza significativa e utile per l'identificazione del volto.
La fase di feature extraction consiste dunque nel trovare ed applicare l'opportuna proiezione che permetta di cambiare la rappresentazione dei volti portandoli dallo spazio delle immagini originali (*image space*) in un sottospazio di dimensione inferiore, che prende il nome di *feature space*. La selezione dello spazio di feature più consono allo scopo può rivelarsi, a sua volta, un problema piuttosto complesso e dal risultato di questa fase dipende in maniera critica il successo dell'intera operazione di face recognition.
In letteratura, esistono numerose proposte per l'estrazione delle feature, alcune delle quali saranno esaminate più dettagliatamente nel prossimo paragrafo.
La fase finale di face recognition ha lo scopo di produrre una classificazione del volto fornito in input, che renda possibile verificarne o determinarne l'identità e costituisce, quindi, il focus di questa tesi; infatti, il framework implementato cerca di essere il più possibile indipendente dalla precedente fase di feature extraction.
Anche per la realizzazione di questa componente del sistema, si sono succedute nel tempo diverse metodologie, che verranno analizzate nel seguito della dissertazione, in quanto costituiranno il termine di paragone per valutare le prestazioni del nuovo algoritmo proposto.

1.1 Principali problematiche

Come già ricordato, il problema del riconoscimento di volti rappresenta un caso specifico e arduo del piú generale problema di pattern recognition. La

difficoltà principale, che lo contraddistingue, deriva dal fatto che i visi, nella loro forma più comune (cioè, quella frontale), appaiono tutti circa allo stesso modo, dal momento che le reciproche differenze risultano piuttosto sottili. Di conseguenza, le immagini di volti frontali formano dei cluster molto densi nello spazio delle immagini e ciò fa in modo che risulti praticamente impossibile, per le tecniche tradizionali di pattern recognition, riuscire ad eseguire una classificazione che raggiunga un alto grado di successo. Tuttavia, il volto umano non è affatto un oggetto unico e rigido; infatti, esistono numerosi fattori che possono incidere sull'aspetto di un viso. Queste cause di variazione si possono classificare in due gruppi:

A) Fattori intrinseci: sono dovuti esclusivamente alla natura fisica del volto e sono indipendenti dall'osservatore. Questi fattori possono essere suddivisi, a loro volta, in altre due classi:

- *Fattori intrapersonali*: riguardano la variazione di aspetto di una stessa persona (ad esempio, età, espressione, presenza di barba, occhiali, trucco, ecc...);
- *Fattori interpersonali*: riguardano la variazione di aspetto di un insieme di persone rispetto a un altro; ricadono dunque all'interno di questo gruppo l'appartenenza ad una particolare etnia e il genere sessuale.

B) Fattori estrinseci: dipendono dall'interazione della luce sul volto dell'individuo osservato e sull'osservatore. Questi fattori includono l'illuminazione, la posa, il fattore di scala e i parametri di imaging (ad esempio, la risoluzione, la messa a fuoco, la presenza di rumore, ecc...).

Nello scenario di face recognition, in cui ogni individuo presente nel training set costituisce una classe, le problematiche generate dalle cause sopraelencate si dividono sostanzialmente in due gruppi: quello della similarità *inter-classe* e quello della variabilità *intra-classe*. La prima consiste nel dover fare i conti con la presenza, nel training set, di due o più individui i cui volti siano molto simili - l'esempio più eclatante e naturale, in questo senso, è rappresentato dal caso dei gemelli. La seconda, invece, si verifica quando alcuni dei fattori di disturbo risultano così accentuati da rendere apparentemente molto diverse due rappresentazioni del volto di uno stesso soggetto.

(a) (b)

Figura 1.2: a) Similarità inter-classe b) Variabilità intra-classe

È dunque evidente che, per ottimizzare la correttezza della classificazione sarà necessario da una parte massimizzare la variabilità inter-classe e, al contempo, minimizzare quella intra-classe. Un altro problema, in cui si può incorrere quando la dimensione dell'immagine di input è molto grande, è quello che prende il nome di *curse of dimensionality*, ma, come già sottolineato precedentemente, questo ostacolo può essere aggirato proiettando le immagini in un opportuno sottospazio di dimensione inferiore.

2 Estrazione delle feature

Nel paragrafo precedente si è già accennato al fatto che, nel corso degli anni, la ricerca nell'ambito del problema di feature extraction per il riconoscimento di volti ha prodotto diversi risultati interessanti; in questa sezione verranno dunque riassunte le principali metodologie scoperte.
Innanzitutto, si ricorda che l'intento fondamentale di queste tecniche è quello di proiettare le immagini originali che rappresentano i volti - sia del training set, sia del test set - in un opportuno sottospazio, detto feature space.
Per operare questo calcolo esistono diverse possibilità:

- **Generare un unico sottospazio, valido per tutti gli individui:** in base a questo approccio, si utilizza un'unica matrice di proiezione per trasformare le immagini di partenza (compresa quella che contiene il volto da riconoscere) nell'unico sottospazio considerato, per poi procedere con la classificazione. Lo svantaggio più evidente delle tecniche di feature extraction che ricadono in questa categoria consiste nella necessità di ricalcolare l'intera matrice di proiezione, qualora si ritenesse opportuno aggiungere altri volti al dizionario precedentemente utilizzato, nel tentativo di rendere più efficiente la ricerca della classe corretta. I metodi principali che si basano su questo sistema sono il metodo *eigenfaces* [9] e il metodo *fisherfaces* [10], che verranno meglio descritti in seguito.

- **Generare un sottospazio diverso per ogni soggetto:** questa opzione permette di superare la problematica evidenziata per le tecniche precedenti, in quanto, dopo aver aggiunto nuovi individui al training set, risulterà indispensabile unicamente calcolare la matrice di proiezione per le nuove classi aggiunte.
  La successiva fase di riconoscimento richiederà, in questo caso, di associare il volto da identificare alla classe la cui distanza fra i relativi sottospazi sia minima.

- **Generare più sottospazi diversi per ogni individuo:** in questo modo, si ottiene generalmente una suddivisione migliore delle classi, poiché si riesce a tenere in considerazione un maggior numero di variazioni. Come le tecniche della categoria precedente, anche quelle contenute in questo insieme eseguono il riconoscimento associando il volto, inizialmente sconosciuto, alla classe il cui spazio ha distanza minima rispetto a quello dell'immagine di test.
  Il calcolo della distanza fra i diversi sottospazi può chiaramente essere effettuato sulla base della misura che si ritiene più adeguata, fra tutte quelle possibili.

I metodi di feature extraction possono, inoltre, essere suddivisi sulla base delle caratteristiche che cercano di estrarre, ottenendo la seguente classificazione:

- **Tecniche olistiche:** per estrarre le caratteristiche utili alla classificazione, viene considerata l'immagine del volto nella sua interezza.
  Il principale vantaggio messo a disposizione da questi metodi è quello di risultare efficaci anche in presenza di immagini di input di piccole dimensioni.

- **Tecniche locali:** estraggono le feature utilizzando soltanto alcune zone dell'immagine come, ad esempio, la regione degli occhi, del naso e/o della bocca.

- **Tecniche ibride:** com'è possibile dedurre dal nome, queste tecniche estraggono caratteristiche sia globali che locali.
  Garantiscono, potenzialmente, migliori prestazioni nella successiva fase di classificazione, poiché forniscono un'informazione più completa, ma sono anche più difficili da gestire e interpretare.

Scendendo maggiormente nel dettaglio, è possibile catalogare le varie tecniche di feature extraction esistenti anche secondo la suddivisione seguente:

- **Metodi basati sull'estrazione di caratteristiche geometriche del volto:** le tecniche comprese in questa categoria ricavano le proprietà e le relazioni geometriche - distanze, aree, angoli, ecc... - che collegano le varie componenti del volto - ovvero, il naso, la bocca, gli occhi - e le sfruttano nella fase di classificazione. Questi metodi sono quelli proposti per primi, ma, grazie alla loro estrema semplicità ed efficienza continuano ad essere utilizzati anche in alcuni sistemi odierni. Nonostante tutto, però, l'uso delle sole relazioni geometriche spesso si rivela insufficiente per l'ottenimento di una percentuale soddisfacente di riconoscimenti corretti, e ciò avviene perché queste feature non tengono conto né della trama del volto né del suo aspetto.

- **Metodi basati sull'apprendimento statistico:** queste tecniche fondano il loro funzionamento sull'esistenza di un vasto database di apprendimento - che può contenere immagini o direttamente feature -, sfruttando il quale tentano di *imparare* quali siano le caratteristiche più adatte alla definizione di un classificatore che si avvicini quanto più possibile a quello ottimo. Per ottenere il risultato sperato, è quindi necessario che questi metodi sfruttino al massimo la conoscenza a priori sui campioni di learning. Grazie alla sua adattabilità, questo insieme di tecniche è oggi largamente adoperato in molteplici algoritmi di riconoscimento di volti.

- **Metodi lineari basati sull'aspetto globale del volto:** queste tecniche proiettano gli esempi inclusi nel training set - che generalmente vengono rappresentati utilizzando vettori n-dimensionali contenenti l'intensità dei vari pixel - direttamente nel sottospazio delle feature, per effettuare poi la classificazione. È dunque evidente come questi metodi ricadano nella categoria delle tecniche olistiche. Tuttavia, la loro linearità, pur garantendo una maggiore stabilità rispetto alle tecniche basate sull'estrazione di feature geometriche, non consente di gestire al meglio lo spazio di rappresentazione dei volti che, per sua natura, risulta essere non lineare e non convesso; ragion per cui, a volte, questi metodi non permettono di raggiungere buoni risultati.
  Gli esempi più noti di tecniche di questo tipo sono: *principal component analysis (PCA)* [9],[12], *linear discriminant analysis (LDA)* [10], [13], *locality preserving projection (LPP)* [14], [19] e *indipendent component analysis (ICA)* [15], [16].

- **Metodi basati su kernel non lineari:** questi metodi costituiscono un'estensione di quelli precedentemente descritti. La loro peculiarità consiste nell'utilizzo di una proiezione non lineare che conduca ad un nuovo sottospazio di rappresentazione delle feature. Data la non linearità della trasformazione, lo spazio risultante non sarà necessariamente di dimensioni inferiori rispetto all'originale. Tuttavia, ciò non costituisce un problema in quanto, in questo caso, l'obiettivo principale è

quello di preservare tutte quelle informazioni che sono indispensabili per ottimizzare la discriminazione dei volti. Queste tecniche risultano effettivamente più flessibili, ottenendo ottimi risultati nel caso in cui i volti sottoposti al sistema appartengano ad individui che si trovano nel training set; d'altra parte le loro prestazioni si riducono in presenza di volti di soggetti ignoti.
Appartengono a questa categoria le metodologie *Kernel PCA* [17] e *Kernel LDA* [18].

- **Metodi basati sull'aspetto locale delle componenti del volto:** lo scopo di questi metodi è quello di ottenere una maggiore robustezza nei confronti delle possibili variazioni. Per raggiungere questo risultato sfruttano l'utilizzo di opportune tecniche di filtraggio. Fra tutte le metodologie che ricadono in questo gruppo, possiamo ricordare la *local feature analysis (LFA)* [20], [21], l'*elastic graph bunch matching (EGBM)* [22] e, soprattutto, il *local binary pattern (LBP)* [23].

Nei prossimi paragrafi verrà esaminato più approfonditamente il funzionamento di alcune delle tecniche precedentemente annoverate.

2.1 Metodi lineari basati sull'aspetto globale del volto

L'idea di partenza di queste tecniche consiste nella generazione di una base vettoriale che permetta di descrivere qualsiasi volto all'interno di un spazio ideale di rappresentazione dei volti. Per costruire questa base è possibile utilizzare il metodo di fattorizzazione di una matrice, noto come *singular value decomposition (SVD).*
Sia M una matrice di dimensioni $m \times n$ i cui valori appartengano ad un campo K, reale o complesso; allora è possibile fattorizzare la matrice M nel seguente modo:

$$M = U\Sigma V^*$$

dove U è una matrice unitaria di dimensione $m \times n$, V^* è la trasposta coniugata della matrice unitaria V di dimensione $n \times n$ e, infine, Σ è una matrice

diagonale $n \times n$ di numeri reali non negativi. Le colonne di U prendono il nome di vettori singolari sinistri e costituiscono gli autovettori della matrice M^*M; mentre le colonne di V vengono denominate vettori singolari destri e coincidono con gli autovettori della matrice MM^*. Gli elementi della diagonale di Σ vengono detti *valori singolari* e definiscono la rilevanza dei relativi vettori singolari.
I metodi PCA e LDA si basano proprio su questa decomposizione per ottenere una base ortonormale che permetta di generare una rappresentazione più compatta dei volti. Diversamente, il metodo ICA sfrutta un'ottimizzazione più ad alto livello per procurarsi una descrizione più precisa della variabilità inter-classe.
Analizzeremo ora in dettaglio alcune di queste tecniche.

Principal Component Analysis (PCA) e Eigenfaces

Lo scopo della *principal component analysis* è quello di proiettare le immagini originali, sia del training set, sia del test set, in un sottospazio di dimensione inferiore. Per ottenere questo risultato viene utilizzata la trasformata di *Karhunen-Loève (KL)*, la cui formulazione coincide con quella della PCA, se si ipotizza che i dati abbiano media nulla. Il metodo di trasformazione Karhunen-Loève riduce la dimensionalità di uno spazio n-dimensionale portandolo ad una dimensione k, dove chiaramente si avrà $k<n$. Il calcolo della base ortonormale che permette questa proiezione avviene, come visto in precedenza, sfruttando la teoria della decomposizione SVD, la quale consente appunto di trovare quegli autovalori e autovettori che andranno a costituire la matrice di trasformazione e che, nell'ambito specifico della face recognition, prendono il nome di *autofacce* (*eigenfaces*).
Vediamo ora nel dettaglio come avviene il calcolo della matrice di proiezione. Si supponga che il training set X sia composto da m immagini, ognuna memorizzata concatenando le sue colonne per ottenere un vettore n-dimensionale:

$$X = \{x_1, x_2, ..., x_m\}$$

A questo punto è possibile calcolare la faccia media di tutto il dizionario nel seguente modo:

$$\hat{x} = \frac{1}{m} \sum_{x \in X} x$$

Così come la matrice di covarianza:

$$C = \frac{1}{m} \sum_{x \in X} (x - \hat{x}) (x - \hat{x})^T$$

Una volta ottenuta la matrice C, è finalmente possibile calcolarne autovettori e autovalori, per poi disporli in ordine decrescente e costruire così la matrice di proiezione Φ_k come:

$$\Phi_k = [\phi_1, \phi_2, ..., \phi_k], \; con \, \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_k$$

Dove i ϕ_j rappresentano gli autovettori - o *eigenfaces* - e i λ_j rappresentano, invece, i corrispondenti autovalori. Dunque, una volta scelta la desiderata dimensione k per il sottospazio delle feature, si avrà che la matrice di proiezione sarà costituita dai k autovettori maggiori.
Una volta generata la matrice di proiezione, si può procedere all'estrazione delle feature utilizzando la formula:

$$y_i = \Phi_k^T (x_i - \hat{x})$$

Quando viene sottoposto al sistema di face recognition un nuovo volto per l'identificazione, anch'esso dovrà essere proiettato nel medesimo sottospazio, utilizzando la stessa trasformazione.
Il metodo di classificazione solitamente affiancato alla tecnica eigenfaces è il *nearest neighbour*, in base al quale il volto da riconoscere viene associato alla classe per cui risulta minima la distanza euclidea fra la proiezione di una delle sue immagini y_i, contenute nel training set, e la proiezione dell'immagine di test. Chiaramente, al posto del metodo nearest neighbour è possibile introdurre classificatori alternativi come, ad esempio, il *K-nearest neighbour*, in cui vengono considerati i k volti la cui proiezione è più vicina a quella del viso da riconoscere, al fine di eseguire l'identificazione scegliendo una determinata regola di voto fra questi k volti.
Come si vedrà poi anche in seguito, il metodo di estrazione delle feature

eigenfaces è stato sfruttato, assieme ad altre tecniche, in fase di sperimentazione.

Linear Discriminant Analysis (LDA) e Fisherfaces

Il principale inconveniente del metodo eigenfaces si concretizza in una separazione - o *scattering* - delle classi non sempre ottimale, e ciò si deve al fatto che questa tecnica non considera soltanto la variabilità inter-classe - che, come abbiamo visto, deve essere massimizzata per favorire il riconoscimento - ma considera anche quella intra-classe - che, si ricorda, deve essere invece minimizzata - generando, in questo modo, dei raggruppamenti errati, che possono dipendere dalla presenza di forti variazioni nelle condizioni di illuminazione o nell'espressione, fra le diverse rappresentazioni del volto di uno stesso soggetto.
Al fine di superare la problematica sopra evidenziata, è stata proposta, come alternativa alla trasformata PCA, la trasformata *linear discriminant analysis (LDA)*, nota anche col nome di *Fisher's linear discriminant (FLD)* - dal suo inventore Fisher; anche se la formulazione originale di quest'ultimo non prevedeva alcune assunzioni sui dati, che, invece, vedremo presenti nella formulazione dell'LDA.
Un'implementazione esplicita del metodo FLD, applicata al riconoscimento dei volti, prende il nome di *fisherfaces*. Questo algoritmo consente di ottenere una riduzione di dimensionalità lineare e supervisionata con lo scopo di massimizzare la separazione tra le classi.
Per il calcolo della trasformata LDA è necessario disporre di un training set, le cui classi siano già state etichettate. Si supponga, dunque, come visto per il caso precedente, che X sia un training set, composto da m immagini sotto forma di vettori n-dimensionali, $X = \{x_1, x_2, ..., x_m\}$, inoltre, sia $\wp = \{X_1, X_2, ..., X_s\}$ la partizione di X in s classi, determinata sulla base dell'etichettatura degli esempi di training. Infine, sia m_i la cardinalità relativa alla classe X_i. Allora è possibile cercare di massimizzare la separazione tra le classi, prendendo in considerazione le seguenti relazioni:

- *Within-class:*

$$S_w = \sum_{i=1}^{s} m_i C_i,\ C_i = \frac{1}{m_i} \sum_{x \in X_i} (x - \hat{x}_i)(x - \hat{x}_i)^T,\ \hat{x}_i = \frac{1}{m_i} \sum_{x \in Tr_i} x$$

- *Between-class:*

$$S_b = \sum_{i=1}^{s} m_i (\hat{x}_i - \hat{x}_0)(\hat{x}_i - \hat{x}_0)^T,\ \hat{x}_0 = \frac{1}{m} \sum_{i=1}^{s} m_i \hat{x}_i,\ \hat{x}_i = \frac{1}{m_i} \sum_{x \in X_i} x$$

La relazione *within-class* descrive lo scattering di ogni volto rispetto al centro della propria classe, mentre quella *between-class* descrive lo scattering presente fra i centri delle varie classi e il centro del sottospazio usato come rappresentazione per tutti i volti. Traducendo quanto detto finora in altri termini, si può dunque affermare che, per migliorare le prestazioni del sistema di face recognition, è indispensabile massimizzare lo scattering *between-class*, cercando al contempo di minimizzare quello *within-class*.

Per costruire un'opportuna matrice di proiezione che permetta di ottenere

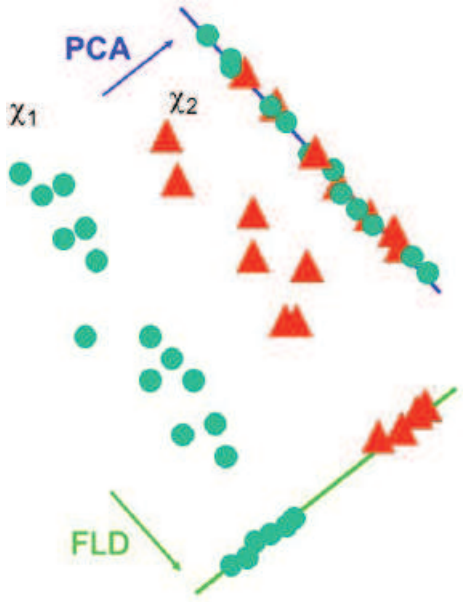


Figura 1.3: Confronto fra PCA e FLD

questo risultato, si procede, questa volta, con il calcolo degli autovalori e degli autovettori della matrice $S_w^{-1} S_b$, anziché della matrice C, come invece avevamo precedentemente visto nel metodo eigenfaces. Anche in questo caso, però, è opportuno considerare solamente i primi k maggiori autovettori,

tenendo presente che, con questa tecnica si ha necessariamente $k \leq s - 1$, dove, si ricorda, s è il numero di classi presenti nel training set.
Come già visto per il metodo eigenfaces, pure per la tecnica fisherfaces, dopo aver generato la matrice di trasformazione, si esegue il riconoscimento proiettando nel relativo sottospazio tutte le immagini del training set e quella che contiene il volto da riconoscere, concludendo il procedimento con l'utilizzo del classificatore che si ritiene più consono allo scopo.
Anche il metodo fisherfaces è stato considerato nella successiva fase di sperimentazione.

Locality Preserving Projection (LPP) e Laplacianfaces

Come si può dedurre dal nome, il metodo di *locality preserving projection (LPP)* intende ricavare una matrice di proiezione, che traduca le immagini dei volti in un nuovo sottospazio, senza perdere però le informazioni di località presenti nell'image space; informazioni che permettono di valutare al meglio la similarità intra-classe. L'algoritmo, che applica questa metodologia specificatamente nell'ambito della face recognition, prende il nome di *laplacianfaces*.
Questa trasformazione rappresenta la miglior approssimazione lineare della *eigenfunction* dell'operatore di *Laplace-Beltrami*; verrà dunque ora specificato come ottenere la matrice di proiezione sulla base di queste premesse.
Sia X un dizionario composto da m immagini, rappresentate sotto forma di vettori n-dimensionali ($n = w \times h$, dove w e h sono rispettivamente larghezza e altezza delle immagini); è allora possibile costruire la matrice di proiezione W_{PCA} utilizzando il metodo PCA, introdotto precedentemente. Si costruisca, a questo punto, un *nearest-neighbour graph* G per descrivere la similarità locale a livello di image space. Il grafo G sarà dunque composto da m nodi - le immagini - e presenterà un arco tra un nodo i e un nodo j soltanto nel caso in cui l'immagine *i-esima* e quella *j-esima* risultino *vicine* fra loro.
Per generare il suddetto grafo è possibile attenersi ad una delle due seguenti strategie:

- ϵ-neighbourhood: i nodi i e j sono connessi da un arco solo se $\|x_i - x_j\|^2 < \epsilon$, con $\epsilon \in \mathbb{R}$;
- K-nearest neighbour: con $k \in \mathbb{N}$, i nodi i e j sono connessi da un arco se i è tra i k nodi più vicini a j o viceversa, utilizzando come misura ancora la distanza euclidea.

Dopo aver costruito il grafo G, si può procedere alla creazione della matrice dei pesi degli archi S, in questo modo: se i nodi i e j sono collegati da un arco s'imposterà il costo dell'arco ij come

$$S_{ij} = e^{-\frac{\left\|x_i - x_j\right\|^2}{t}}$$

dove t è una costante, altrimenti si avrà $S_{ij} = 0$.
Giunti a questo punto, è necessario calcolare gli autovettori e gli autovalori del seguente problema generico:

$$XLX^T w = \lambda XDX^T w$$

dove D è la matrice diagonale i cui elementi diversi da 0 corrispondono con la somma delle varie colonne di S,

$$D_{ii} = \sum_j S_{ij}$$

mentre $L = D - S$ è la matrice *laplaciana*. I vettori w costituiscono, invece, gli autovettori soluzione del sistema, relativi ai λ autovalori.
Ordinando gli autovalori e gli autovettori, così ottenuti, in ordine decrescente e scegliendo la dimensione desiderata k per il sottospazio, è ora possibile calcolare una nuova matrice W_{LPP} utilizzando come colonne i primi k autovettori. Le ultime due operazioni previste consistono nel calcolo della matrice di proiezione finale come,

$$W = W_{PCA} W_{LPP}$$

e nella proiezione di tutte le immagini x nel corrispondente sottospazio, utilizzando la formula:

$$y = W^T x.$$

Anche il metodo di estrazione delle feature *laplacianfaces* è stato usato nella sperimentazione.

Indipendent Component Analysis (ICA)

Come già accennato in precedenza, la tecnica *indipendent component analysis (ICA)* utilizza un'ottimizzazione ad alto livello che permette una descrizione più accurata della variabilità inter-classe. Questo metodo, largamente sfruttato nell'ambito dell'analisi di segnali digitali e, più specificatamente, nel problema di *source separation*, ha l'obiettivo di ricostruire i segnali di origine S_i a partire dalle osservazioni X_j, dove si suppone che ogni osservazione rappresenti un mix ignoto dei segnali originali. Assumendo che i suddetti segnali risultino fra loro statisticamente indipendenti, è allora possibile rigenerarli sfruttando esclusivamente le osservazioni in cui essi risultano combinati. Esistono diverse varianti della tecnica ICA, alcune delle quali superano il limite imposto dalla linearità, sfruttando combinazioni anche di grado superiore.
Nell'ambito della face recognition sono state prodotte, a loro volta, differenti implementazioni della metodologia ICA, fra le quali, si ricordano, a titolo di esempio, quelle note col nome di *Architettura I* e *Architettura II*.
Nell'Architettura I, le immagini che rappresentano i volti da riconoscere vengono considerate come una combinazione lineare di più immagini di base, fra loro statisticamente indipendenti e *mescolate* secondo una matrice A ignota. In questo contesto, le immagini dei volti costituiscono, dunque, le variabili, mentre i valori d'intensità dei pixel rappresentano le osservazioni; quindi, l'algoritmo ICA calcola degli opportuni pesi, che verranno poi utilizzati per ricostruire un insieme di immagini base fra loro indipendenti.
Nell'Architettura II, invece, si risolve il duale del problema precedente, ovvero, questa volta le intensità dei pixel costituiscono le variabili, mentre le immagini dei volti rappresentano le osservazioni. Quindi, l'impostazione dell'Architettura I preserva maggiormente le caratteristiche locali delle immagini, al contrario, lo schema dell'Architettura II favorisce la conservazione delle informazioni di carattere globale.
Da una certa prospettiva, si può considerare il metodo ICA come una generalizzazione del metodo PCA, infatti, mentre proprio quest'ultima tecnica viene utilizzata in fase di pre-processing, successivamente ICA esegue un'ulteriore discriminazione delle dipendenze di ordine superiore presenti tra le immagini, col fine di separare al meglio le diverse classi.

2.2 Metodi basati su kernel non lineari

È stato già precedentemente accennato come queste tecniche cerchino di ottenere una migliore separazione fra le classi sfruttando delle funzioni di proiezione non lineari e il cui spazio risultante potrebbe anche avere una dimensione addirittura superiore a quello di partenza.
Un esempio di algoritmo appartenente a questa categoria consiste nel cosiddetto *Kernel PCA*, il quale rappresenta un'estensione del metodo PCA classico in un contesto appunto non lineare. Più precisamente, questa tecnica sfrutta una funzione kernel per effettuare un mapping non lineare che conduce ad uno spazio la cui dimensionalità risulta più elevata rispetto a quella dell'image space. L'intento è, infatti, quello di usufruire del maggior numero di coordinate dei vettori-volto per discriminare più marcatamente le varie classi, semplificando di conseguenza il processo di riconoscimento. Per ottenere questo risultato, è comunque indispensabile utilizzare anche classificatori particolarmente robusti, capaci di trattare in maniera opportuna la dimensionalità dello spazio e la non linearità della trasformazione.

2.3 Metodi basati sull'aspetto locale delle componenti del volto

Queste tecniche sfruttano filtri e maschere locali per descrivere, al meglio e con la minor quantità di informazione possibile, le caratteristiche dei volti.
Un esempio emblematico di filtri adatti a questo scopo è rappresentato dai filtri di Gabor; questi consistono nell'applicazione di una funzione sinusoidale, attenuata da una funzione gaussiana. Per descriverli è quindi necessario definire la frequenza e l'orientazione della sinusoide, oltreché l'ampiezza della gaussiana. I vettori di feature possono essere successivamente ricavati eseguendo la convoluzione delle immagini dei volti con un banco di filtri, in modo da considerarne diverse combinazioni.
Alcuni algoritmi che utilizzano i filtri di Gabor sono: l'*elastic bunch graph matching (EBGM)*, il *Gabor Fisher classifier (GFC)* [24] e l'*Adaboost Gabor Fisher classifier (AGFC)* [25].

Si proseguirà, dunque, la dissertazione esaminando il metodo EBGM.

Elastic Bunch Graph Matching (EBGM)

Le immagini dei volti vengono rappresentate utilizzando una struttura *ad hoc*, denominanta *labeled graph*, che permette di memorizzare sia informazioni locali, sia informazioni topografiche globali del viso. I nodi di questo grafo coincidono con le caratteristiche locali, che vengono rappresentate attraverso la distribuzione dei livelli di grigio, ricavata mediante l'applicazione dei filtri di Gabor, diversificando scala e orientazione; gli archi, invece, forniscono indicazioni sulla topologia generale del volto. Un grafo così costruito permette di rappresentare potenzialmente tutte le variazioni del volto di un individuo, indipendentemente dalla presenza di eventuali distorsioni e/o variazioni di illuminazione e di scala. La fase di riconoscimento si basa su una misura di similarità fra grafi, selezionando per ogni nodo la distribuzione di livelli di grigio che meglio descrive localmente il volto, per poi ricavarne una rappresentazione della geometria globale.

Local Binary Pattern (LBP)

Questo metodo utilizza delle maschere di dimensione 3×3, per assegnare ad ogni pixel un valore binario 0 o 1, determinato analizzando le diverse intensità dei pixel stessi. Queste maschere vengono passate su tutta l'immagine, che, per lo scopo, sarà stata preventivamente suddivisa in più finestre di ricerca, eventualmente anche sovrapposte. I valori generati dall'applicazione di queste maschere permettono di identificare diversi pattern, utili all'evidenziazione di particolari caratteristiche del volto. L'output del processo di filtraggio consiste in un vettore di istogrammi (uno per ogni regione), che rappresenterà quindi il vettore di feature desiderato.
Recenti estensioni di questa tecnica permettono l'utilizzo anche di maschere di dimensione maggiore, che rendono possibile quindi l'analisi di un intorno circolare di pixel più ampio.

3 Principlali tecniche di classificazione

Nel paragrafo precedente sono stati passati in rassegna alcuni dei metodi più comuni per l'estrazione delle feature. In questa sezione, invece, verranno presentate le metodologie maggiormente utilizzate per la successiva fase di classificazione dei volti.
Innanzitutto, possiamo suddividere le diverse tecniche esistenti nelle seguenti categorie:

- metodi basati su sottospazi;
- reti neurali;
- modelli deformabili;
- hidden Markov model;
- support vector machine.

Alcuni di questi classificatori risultano molto semplici da realizzare, ma, al contempo, si rivelano spesso fortemente sensibili alle eventuali variazioni presenti nelle immagini e il loro esito non è quindi sempre soddisfacente - esempi emblematici di tecniche di questo tipo consistono nel *nearest neighbour* e nel *nearest subspace* -; altri classificatori, al contrario, sono pensati per essere più robusti e dunque più resistenti nei confronti delle possibili variazioni, ma, d'altra parte, la loro realizzazione risulta meno immediata - ad esempio, si ricorda il caso del *support vector machine*.
Nel seguito, si esamineranno con maggiore dettaglio gli ultimi tre metodi citati. Fra questi, i primi due sono stati utilizzati in fase di sperimentazione, come termini di confronto con il framework di classificazione presentato in questa tesi.

3.1 Nearest Neighbour (NN)

Il metodo nearest neighbour (NN), come già antecedentemente sottolineato, rappresenta la tecnica di classificazione storicamente associata all'estrazione delle feature con eigenfaces. Questo criterio di discriminazione richiede innanzitutto il calcolo delle distanze fra il vettore che rappresenta il volto da

riconoscere e tutti i vettori contenuti nel training set - chiaramente, qualora sia stata impiegata una tecnica per la riduzione della dimensionalità, dovranno essere considerate, invece, le distanze fra i relativi vettori di feature. Dopodiché, la classificazione vera e propria è eseguita associando l'immagine di input alla classe che contiene il volto che rende minima la distanza.
Come già accennato nel paragrafo precedente, esistono diversi modi per misurare il divario che separa le varie rappresentazioni dei volti; il più comune dei quali consiste notoriamente nel calcolo della distanza euclidea. Nel corso del tempo, tuttavia, sono state proposte numerose alternative come, ad esempio, l'utilizzo della metrica di *Mahalanobis*.
Inoltre, è possibile estendere il concetto di fondo della tecnica nearest neighbour, eseguendo la classificazione non soltanto sulla base del nodo (volto) più vicino, ma valutando nel complesso l'apporto di k nodi limitrofi, attraverso l'uso di un'opportuna regola di voto; questa evoluzione dell'impostazione originale del metodo, prende il nome di *K-nearest neighbour*.

3.2 Nearest Subspace (NS)

Il metodo *nearest subspace (NS)* richiede in primo luogo la determinazione di un sottospazio diverso per ogni classe, nel senso che non si disporrà più, come nel caso del nearest neighbour, di un'unica matrice di proiezione calcolata sulla base dell'intero training set, ma, supponendo che s costituisca il numero di classi in esso presenti, esisteranno s matrici di proiezione, ovvero una per ogni individuo. Per ricavare le matrici di trasformazione, resta comunque possibile utilizzare uno dei tanti metodi visti nel paragrafo precedente.
L'entità della distanza, fra un volto e uno specifico sottospazio, viene determinata misurando il divario presente fra il vettore del volto, rappresentato nell'image space, e la sua proiezione nel sottospazio stesso; anche in questo caso, è possibile usufruire delle medesime metriche utilizzate dal metodo nearest neighbour.
Giunti a questo punto, si può procedere con la classificazione, associando il volto da riconoscere alla classe il cui relativo sottospazio minimizza la distanza.

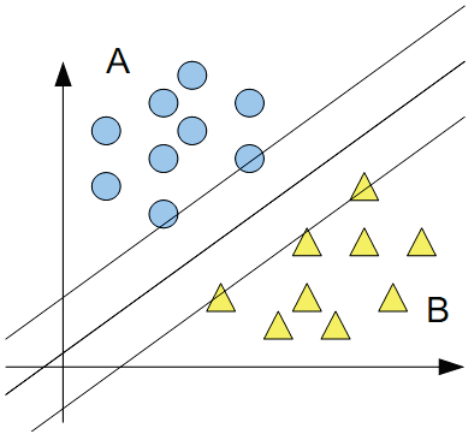


Figura 1.4: Esempio di iperpiano di separazione nel metodo SVM

3.3 Support Vector Machine (SVM)

Il metodo *support vector machine (SVM)* [26] è stato introdotto per la prima volta negli anni '60, ma, grazie agli ulteriori sviluppi raggiunti nel corso delle decadi successive, costituisce ancora oggi uno dei classificatori più efficienti ed utilizzati.

Il nucleo di questa tecnica consiste nel cercare di separare due classi attraverso un opportuno iperpiano (Figura 1.4). La minima distanza fra l'iperpiano di separazione e uno dei volti del training set viene denominata *margine*; questo metodo deve dunque cercare l'equazione che descrive quell'iperpiano che massimizza il margine fra gli esempi delle due classi.

Per ottenere questo scopo è sufficiente utilizzare soltanto alcuni vettori del dizionario di training, ovvero quelli che giacciono sul margine della rispettiva classe e che prendono appunto il nome di *support vector*; la soluzione del problema può dunque essere espressa anche soltanto in funzione di questi vettori, indipendentemente dalle dimensioni originali del training set.

Se i support vector delle due classi sono linearmente separabili, è possibile procedere direttamente alla determinazione dell'equazione dell'iperpiano che massimizza la distanza fra i margini; se, al contrario, i pattern non sono linearmente separabili, diventa necessario rimapparli in uno spazio di dimensione più grande, dove, grazie ai maggiori gradi di libertà, è possibile ottenere una separazione più accurata, utilizzando sempre e comunque un opportuno iperpiano (Figura 1.5). Procedendo in questo modo, si ottiene un risultato

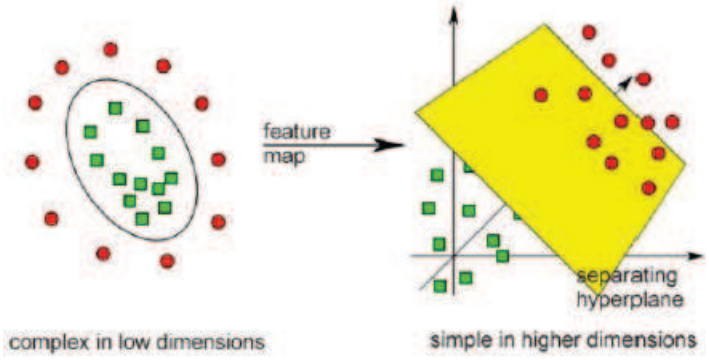


Figura 1.5: SVM - Esempio di mapping in uno spazio più grande

equivalente all'utilizzo di una superficie più complessa nello spazio di dimensione inferiore.

Finora è stato preso in considerazione soltanto il caso in cui le classi da suddividere siano due; infatti, in presenza di un numero più elevato di classi il metodo si complica notevolmente, risultando ancora oggi oggetto di fervente ricerca. Tuttavia, in questi casi, si riescono a raggiungere buoni risultati attraverso la tecnica *one-against-all*, in base alla quale la separazione viene eseguita iterativamente distinguendo gli elementi di ciascuna classe da quelli di tutte le altre.

Come già sottolineato in apertura del paragrafo, il metodo SVM si dimostra particolarmente robusto e garantisce il raggiungimento di ottimi risultati, anche se, d'altra parte, la scelta della giusta funzione di separazione richiede una preventiva e attenta parametrizzazione dell'algoritmo.

Nel prossimo capitolo vedremo un framework di classificazione, alternativo a quelli finora presentati, basato sulla recente e sorprendente teoria del compressed sensing.

Capitolo 2

Riconoscimento di volti basato sulla rappresentazione sparsa

Nel capitolo precedente è stato presentato il problema del riconoscimento di volti, esaminandone le diverse sfaccettature; è stato innanzitutto dimostrato come questo costituisca uno dei campi di ricerca più ferventi degli ultimi anni, non soltanto nell'ambito dell'analisi di immagini, ma anche in quello delle neuroscienze e della psicologia; in seguito, è stato evidenziato il ruolo di estrema importanza ricoperto dalla fase di estrazione delle feature, nell'approccio tradizionale al problema; infine, sono state passate in rassegna alcune delle principali tecniche di classificazione esistenti.
La criticità dello stadio di feature extraction ha fatto sì che, per molti anni, gli studiosi impegnati nella ricerca di sistemi di face recognition sempre più robusti si fossero concentrati principalmente sulla scoperta di nuove metodologie in questo settore; in questo capitolo, invece, verrà presentato un ottimo classificatore - introdotto per la prima volta da John Wright ed altri in [1], che, sfruttando il concetto di sparsità, è in grado di produrre risultati molto interessanti, a prescindere dal tipo di caratteristiche estratte.

Lo spunto per l'ideazione di questo nuovo metodo prende origine dal cosiddetto *principio di parsimonia*, e più precisamente da una delle sue istanze più famose, ovvero quella del principio di selezione di un modello sulla base della sua rappresentazione più compatta. Questa teoria giunge alla conclusione che, considerando una gerarchia di classi di formulazioni matematiche, il

modello che considera, fra tutte le rappresentazioni possibili, quella più semplice deve essere preferito agli altri, quando il suo utilizzo riguarda processi decisionali, come ad esempio la classificazione. In un certo senso, le tecniche di feature exctraction, studiate nel capitolo precedente, si basano a loro volta su questo principio, anche se in maniera più ingenua.
L'importanza dei fondamenti metodologici legati alla parsimonia è stata inoltre rafforzata dagli studi effettuati sul sistema visivo umano; infatti, i ricercatori di questo settore hanno recentemente dimostrato che anche il nostro cervello sfrutta il principio sopra descritto, nel processo di visione e riconoscimento di oggetti. In particolare, è stato evidenziato che molti neuroni legati alla vista risultano particolarmente selettivi, reagendo esclusivamente ad un tipo specifico di stimoli - colore, trama, orientazione, fattore di scala, ecc. Considerando che questi neuroni formano, ad ogni passo del processo visivo, un dizionario più che completo di elementi base di un segnale, è allora possibile concludere che la loro attivazione, nei confronti di una data immagine di input, risulterà essere tipicamente molto sparsa.
Nella comunità di elaborazione statistica dei segnali, il problema algoritmico di calcolare linearmente una rappresentazione sparsa, rispetto a un dizionario più che completo di elementi base di un segnale, ha ultimamente subìto una forte ondata di interesse. Gran parte di questa eccitazione è dovuta alla scoperta secondo cui, qualora la rappresentazione ottima di un segnale risulti essere sufficientemente sparsa, è allora possibile calcolarla in maniera efficiente sfruttando metodi di ottimizzazione convessi, nonostante, nel caso generale, questo problema possa risultare estremamente difficile da risolvere. Nel prossimo paragrafo verranno dunque esaminati, in maniera introduttiva, i fondamenti di questa nuova teoria, che prende il nome di *compressed sensing* [27], [28].

1 Introduzione al compressed sensing

Nonostante le origini del compressed sensing possano essere fatte risalire agli anni '70, le scoperte che hanno rivoluzionato il suo approccio sono state formulate soltanto in anni recenti da D. Donoho [27], E. Candès [28] ed altri.

L'apporto fondamentale di questi studiosi è stato quello di aver trovato il modo di determinare il numero minimo di dati necessari per la ricostruzione di un generico segnale; infatti, l'obiettivo originale di queste ricerche non consisteva tanto nel trovare regole d'inferenza o nella classificazione fine a se stessa, ma piuttosto nella rappresentazione e compressione dei segnali, utilizzando - ed è qui che entra in gioco la straordinarietà di questa teoria - un numero di campioni inferiore a quello determinato dal teorema di Nyquist-Shannon [29]. Sulla base di questo principio, infatti, per poter ricostruire in maniera esatta un segnale a banda limitata, risulta necessario utilizzare una frequenza di campionamento almeno doppia rispetto alla frequenza massima presente nel segnale stesso e, data la sua presunta inviolabilità, esso ha costituito per decenni la regola fondamentale seguita da tutti i protocolli di acquisizione.
La nuova teoria del compressed sensing definisce, invece, sotto determinate condizioni, nuovi limiti per il campionamento dei segnali, sfruttando i due concetti di *sparsità* e *incoerenza*.
La sparsità esprime l'idea di poter rappresentare un segnale continuo utilizzando appunto una quantità di misurazioni significativamente minore rispetto a quella suggerita dal teorema di Nyquist-Shannon.
L'incoerenza, invece, esprime l'idea che i segnali che hanno una rappresentazione sparsa in un certo dominio, diverso da quello di acquisizione, devono risultare, d'altra parte, estesi (densi) in quest'ultimo.
In definitiva, la teoria del compressed sensing formalizza un nuovo protocollo di acquisizione, che permette di condensare tutte le informazioni significative per la ricostruzione perfetta di un segnale in un piccolo gruppo di acquisizioni lineari.
Nella prossima sezione, si mostreranno dunque in maggiore dettaglio le fondamenta matematiche su cui poggia questa innovativa teoria.

1.1 Formulazione del compressed sensing

Sia x un segnale sconosciuto, monodimensionale e discreto di dimensione N:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ \cdots \\ x_{N-1} \\ x_N \end{pmatrix} \quad t.c. \;\; x_i \in \mathbb{R} \;\; \forall i = 1...N \,. \tag{2.1}$$

Si noti che, se il segnale di interesse dovesse essere bidimensionale, di dimensione $w \times h$ - come nel caso delle immagini - è comunque possibile rappresentarlo sotto forma di vettore (con $N = w \times h$), concatenando le sue colonne una dopo l'altra. Per semplicità, si continuerà dunque la dissertazione considerando solamente il caso monodimensionale.
Data la matrice di acquisizione Φ di dimensione $M \times N$, con $M < N$, e dato il vettore di misurazioni $y = \Phi x$, sarà allora possibile ricostruire il segnale x risolvendo un sistema sottodeterminato di equazioni della seguente forma:

$$\begin{pmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} & \cdots & \cdots & \phi_{1,N} \\ \phi_{2,1} & \phi_{2,2} & \phi_{2,3} & \cdots & \cdots & \phi_{2,N} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \phi_{M,1} & \phi_{M,2} & \phi_{M,3} & \cdots & \cdots & \phi_{M,N} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ \cdots \\ x_N \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_M \end{pmatrix} \tag{2.2}$$

Tuttavia, com'è noto, il sistema (2.2) ammette infinite soluzioni, ragion per cui si rivelerà indispensabile la ricerca di un'ulteriore condizione che permetta di selezionare, fra tutte quelle esistenti, l'unica ricostruzione desiderata del segnale.
Assunto che x sia K-sparso, ovvero tale da poter essere rappresentato come un vettore di soli K elementi diversi da zero, con $K \ll N$, è allora possibile formulare il problema del compressed sensing nel seguente modo:

$$\min_{u \in \mathbb{R}^N} F(u) \;\; s.t. \;\; \Phi u = y \tag{2.3}$$

dove $F(u)$ rappresenta un'opportuna funzione sparsificante, che descrive il criterio di selezione della soluzione. Una possibile strategia è, ad esempio, quella di porre $F(u)$ uguale ad una certa norma p:

$$F(u) = \|u\|_p = \sqrt[p]{\sum_{i=1}^{N} |u_i|^p} \tag{2.4}$$

La norma più indicata a questo scopo, tale da poter quindi garantire che la soluzione trovata sia effettivamente sparsa, risulta essere la norma l_0; infatti, essa fornisce come risultato il numero di elementi non nulli del vettore:

$$\|u\|_0 = \sum_{i=1}^{N} |u_i|^0 \tag{2.5}$$

Detto questo, è possibile riformulare il problema di ricostruzione del segnale nel modo seguente:

$$\min_{u \in \mathbb{R}^N} \|u\|_0 \quad s.t. \;\; \Phi u = y \tag{2.6}$$

Quello descritto nella (2.6), purtroppo, però, si rivela essere un problema NP-hard, dal momento che non si conosce nessun metodo alternativo per la sua risoluzione, oltre all'esplorazione esaustiva di tutti i possibili vettori u, K-sparsi, che soddisfano il vincolo $\Phi u = y$. Per questo motivo, risulterà necessario cercare un rilassamento del problema sopra descritto, che permetta di ricavare la soluzione corretta in maniera più trattabile. L'approccio generalmente sfruttato per ovviare a questa difficoltà consiste nel sostituire la minimizzazione della norma l_0 con una sua approssimazione convessa consistente nella minimizzazione della norma l_1; il che conduce al seguente nuovo problema di ottimizzazione:

$$\min_{u \in \mathbb{R}^N} \|u\|_1 \quad s.t. \;\; \Phi u = y \tag{2.7}$$

Infatti, è stato dimostrato che i due problemi (2.6) e (2.7) si rivelano equivalenti, se:

- la matrice di acquisizione Φ gode della *restricted isometry property* (RIP), ovvero se ogni insieme di K colonne di Φ forma approssimativamente un sistema ortogonale;

- il numero di acquisizioni M è tale da soddisfare la seguente relazione:

$$M \geq a \cdot K \log\left(\frac{N}{K}\right) \tag{2.8}$$

  per una costante piccola $a > 1$.

Il nuovo problema (2.7), così ottenuto, può dunque essere risolto attraverso l'uso di metodi classici di programmazione lineare.

Nel prossimo capitolo sarà presentato un nuovo framework che permette di considerare come funzione obiettivo, al posto della norma l_1, altre funzioni sparsificanti non convesse, che garantiscono di ottenere risultati generalmente migliori, in quanto forniscono approssimazioni della soluzione più vicine alla norma l_0. Tuttavia, nel prossimo paragrafo, si riprenderà la trattazione mostrando i principi cardine del nuovo classificatore nella sua versione originale, che, come vedremo, utilizza la minimizzazione della norma l_1.

2 Classificazione basata sulla rappresentazione sparsa (SRC)

Il nuovo metodo, che verrà presentato in questo paragrafo, realizza l'operazione di classificazione, sfruttando la capacità di discriminazione intrinseca nella sparsità. L'idea, che sta alla base di questo nuovo approccio, è quella di cercare di rappresentare l'immagine di test, approfittando di un dizionario più che completo, i cui elementi base saranno costituiti dalle immagini del training set stesse. Qualora si disponga di un numero sufficiente di esempi di training per ogni classe, sarà allora possibile rappresentare le immagini di test come una combinazione lineare dei soli elementi che appartengono alla medesima classe. Questa rappresentazione risulterà essere naturalmente sparsa, dal momento che coinvolgerà esclusivamente una piccola porzione dell'intero database di training, e potrà dunque essere determinata in modo efficiente utilizzando la minimizzazione della norma l_1, come abbiamo già visto nella sezione precedente. La ricerca della rappresentazione più sparsa,

fra tutte quelle possibili, inoltre, funge automaticamente da fattore di separazione per le diverse classi. Come si vedrà meglio in seguito, questo tipo di soluzione fornisce anche un criterio straordinariamente efficace per riuscire a rigettare quegli esempi di test che non appartengono a nessuna delle classi conosciute; infatti, il risultato ottenuto in questi casi tenderà a coinvolgere molti elementi del dizionario, estesi su un numero elevato di classi.
L'utilizzo della sparsità per la classificazione si allontana significativamente dall'utilizzo del principio di parsimonia che fanno i metodi di feature extraction, introdotti nel capitolo precedente; infatti, anziché sfruttare la sparsità per ottenere un opportuno modello di rappresentazione, che possa essere successivamente utilizzato per classificare tutti gli esempi di test, in questo contesto, viene eseguita la classificazione utilizzando direttamente la rappresentazione più compatta dell'immagine di test, selezionando in maniera adattiva quegli esempi di training che meglio la ricostruiscono. Questo tipo di classificatore può essere considerato una generalizzazione dei più noti classificatori nearest neighbour e nearest subspace, visti in precedenza; infatti, questo metodo cerca di ottenere una sorta di bilanciamento fra le due tecniche sopra menzionate, considerando tutti i possibili apporti - presenti sia all'interno di ogni classe, sia fra tutte le classi - al fine di determinare il numero minimo di esempi di training, atti a rappresentare ogni immagine di test [1].
Sebbene questo approccio possa essere impiegato in diversi contesti specifici di pattern recognition, nel seguito si discuterà della sua applicazione nell'ambito del problema di face recognition.

2.1 Modellazione matematica

Come abbiamo già visto, il problema basilare del riconoscimento di oggetti prevede l'utilizzo di alcuni esempi di training già etichettati, relativi a k classi distinte, al fine di determinare quella a cui appartiene un nuovo esempio di test.
Si dispongano, dunque, gli n_i esempi di training dell'i-esima classe come colonne di una matrice $A_i \doteq [v_{i,1}, v_{i,2}, ..., v_{i,n_i}] \in \mathbb{R}^{m \times n_i}$. Nel contesto del riconoscimento di volti, si considerino quindi le diverse immagini grayscale di dimensione $w \times h$ come dei vettori $v \in \mathbb{R}^m$ (con $m = w \times h$), ottenuti

concatenando le loro colonne una di seguito all'altra. Le colonne della matrice A_i saranno dunque costituite dalle immagini dei volti di training dell'i-esima classe, riadattati in questo modo.
Nel corso degli anni sono stati prodotti numerosi studi riguardo a modelli statistici, generativi e discriminativi per cercare di stabilire quale struttura risulti più adatta per la matrice A_i in contesti legati al riconoscimento di oggetti e, sebbene l'approccio qui presentato possa essere applicato anche a database con distribuzioni non lineari e multimodali, per semplificare la presentazione, si assumerà d'ora in poi che gli esempi di training di ogni singola classe risiedano tutti in uno stesso sottospazio. Questa costituisce l'unica conoscenza a priori che si assume sui dati in input e verrà quindi sfruttata successivamente nella soluzione del problema.
Dato, dunque, un numero sufficiente di immagini di training per la classe i, organizzati nella matrice $A_i = [v_{i,1}, v_{i,2}, ..., v_{i,n_i}] \in \mathbb{R}^{m \times n_i}$, una qualunque nuova immagine $y \in \mathbb{R}^m$ della stessa classe potrà essere approssimativamente ricostruita come combinazione lineare degli esempi di training, nel seguente modo:

$$y = \alpha_{i,1} v_{i,1} + \alpha_{i,2} v_{i,2} + \cdots + \alpha_{i,n_i} v_{i,n_i} \tag{2.9}$$

per un qualche scalare $\alpha_{i,j} \in \mathbb{R}$, con $j = 1, 2, ..., n_i$. Considerando che, di solito, l'identità dell'immagine di test è inizialmente sconosciuta, risulta necessario definire una nuova matrice A di dimensione $m \times n$ con $n = \sum_{i=1}^{k} n_i$, che rappresenti l'intero training set come concatenazione di tutte le n immagini di training, relative a tutte le k classi conosciute:

$$A \doteq [A_1, A_2, ..., A_k] = [v_{1,1}, v_{1,2}, ..., v_{k,n_k}] \tag{2.10}$$

Allora, la rappresentazione lineare di y può essere riscritta in termini di tutti gli esempi di training, in questo modo:

$$y = Ax_0 \quad \in \mathbb{R}^m, \tag{2.11}$$

dove $x_0 = [0, ..., 0, \alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,n_i}, 0, ..., 0]^T \in \mathbb{R}^n$ è un vettore di coefficienti con valori diversi da zero solo in corrispondenza dell'i-esima classe.
Dal momento che gli elementi del vettore x_0 esprimono l'identità del volto di test y, risulta spontaneo cercare di ricavarlo risolvendo il sistema lineare di

equazioni $y = Ax$. Si noti, innanzitutto, che utilizzare l'intero training set per risolvere il sistema per x costituisce un significativo allontanamento dai metodi classici come nearest neighbour e nearest subspace, che usano a questo scopo soltanto un'immagine o una classe alla volta. In seguito, si dimostrerà come sfruttare questa rappresentazione globale per costruire un classificatore largamente più discriminativo rispetto a quelli classici, sia nell'identificazione della classe corretta per quei volti di cui si ha qualche esempio nel training set, sia nel rifiuto di quegli individui rispetto ai quali non esiste alcuna corrispondenza. Questi vantaggi possono essere ottenuti, inoltre, senza incorrere in un eccessivo aggravio del costo computazionale, che, come vedremo, manterrà la sua linearità nei confronti della dimensione del training set A.
Ovviamente, qualora si avesse $m > n$, il sistema di equazioni $y = Ax$ risulterebbe sovradeterminato e sarebbe possibile ricavare x_0 in qualità di unica soluzione esistente. Tuttavia, abbiamo visto che, nel contesto di riconoscimento dei volti, questo sistema si rivela in realtà sottodeterminato e, dunque, ammetterà infinite soluzioni. Generalmente, questa difficoltà viene superata scegliendo la soluzione che minimizza la norma l_2:

$$\hat{x}_2 = arg\,min \|x\|_2 \quad s.t. \;\; Ax = y \tag{2.12}$$

Sebbene questo problema di ottimizzazione possa essere risolto semplicemente - attraverso il calcolo della pseudoinversa di A - la soluzione $\hat{x}_2$ non risulta particolarmente informativa, al fine della classificazione dell'immagine di test y; infatti, $\hat{x}_2$ è generalmente molto *densa* e contiene elementi molto maggiori di zero in corrispondenza di diversi esempi di training, appartenenti a diverse classi. Per affrontare questa ulteriore difficoltà è allora possibile sfruttare la seguente semplice osservazione: un'immagine di test valida y può essere rappresentata in maniera sufficientemente corretta utilizzando, per la ricostruzione, esclusivamente esempi di training provenienti dalla sua stessa classe. Una rappresentazione di questo tipo risulterà essere naturalmente *sparsa*, ammesso che il numero k di classi sia ragionevolmente grande. Ad esempio, se $k = 20$, soltanto il 5% degli elementi del vettore x_0 desiderato saranno diversi da zero. In generale, più la soluzione x_0 risulterà essere sparsa, maggiore sarà la semplicità nel determinare accuratamente l'identità del volto di test y.

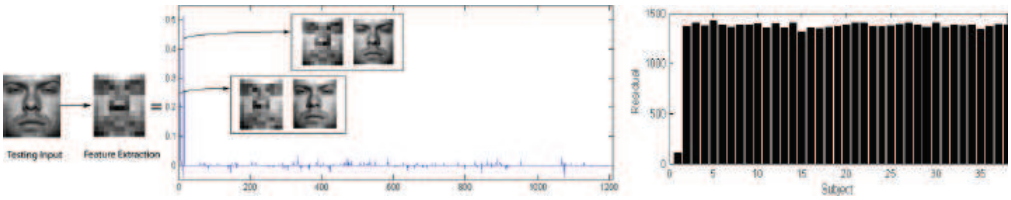


Figura 2.1: Esempio di sparsità della soluzione per un'immagine di test valida dell'Extended Yale B database, ottenuta con la minimizzazione l_1.

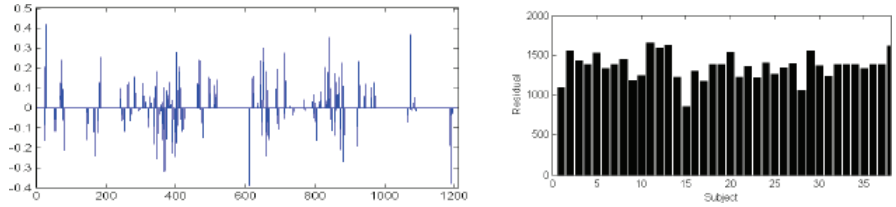


Figura 2.2: Esempio di non sparsità della soluzione, ottenuta con la minimizzazione l_2.

Quanto finora affermato, ci spinge a cercare, fra tutte le soluzioni del sistema $y = Ax$, quella più sparsa, che, com'è stato già dimostrato nel paragrafo precedente, può in linea teorica essere calcolata minimizzando la norma l_0, ottenendo il seguente problema di ottimizzazione:

$$\hat{x}_0 = arg\,min \|x\|_0 \quad s.t. \;\; Ax = y \tag{2.13}$$

Tuttavia, si è già visto che trovare la soluzione più sparsa di un sistema sottodeterminato di equazioni lineari risulta essere un problema NP-hard e, dunque, richiamando la teoria del compressed sensing presentata nel paragrafo precedente, possiamo riformulare il problema sopra descritto sostituendo alla norma l_0 la norma l_1:

$$\hat{x}_1 = arg\,min \|x\|_1 \quad s.t. \;\; Ax = y \tag{2.14}$$

2.2 Robustezza in presenza di rumore

Finora, si è assunto che l'equazione (2.11) valesse in maniera esatta. Considerando che, invece, i dati reali sono generalmente affetti da rumore, potrebbe non essere più possibile esprimere correttamente l'esempio di test come combinazione sparsa degli elementi di training. Tuttavia, per aggirare questo ostacolo, si rivela sufficiente apportare qualche modifica al modello (2.11), affinché esso possa trattare esplicitamente immagini di test affette da una

piccola quantità di rumore - auspicabilmente denso -, ottenendo la seguente nuova formulazione:

$$y = Ax_0 + z \tag{2.15}$$

dove $z \in \mathbb{R}^m$ rappresenta il termine di perturbazione, con energia limitata dalla formula $\|z\|_2 < \epsilon$.
La soluzione sparsa x_0 può ancora essere approssimativamente ricavata risolvendo il seguente problema *stabile* di minimizzazione l_1:

$$\hat{x}_1 = arg\,min \|x\|_1 \quad s.t. \quad \|Ax - y\|_2 \leq \epsilon \tag{2.16}$$

Questo problema convesso può essere risolto efficientemente utilizzando metodi di ottimizzazione del second'ordine. La risoluzione del problema (2.16) garantisce di trovare soluzioni approssimativamente sparse per insiemi di matrici random A [31]: in dettaglio, esistono due costanti ρ e ζ tali per cui con altissima probabilità, se valgono le condizioni $\|x_0\|_0 < \rho\zeta$ e $\|z\|_2 \leq \epsilon$, è possibile calcolare una soluzione $\hat{x}_1$ che soddisfi la seguente disuguaglianza:

$$\|\hat{x}_1 - x_0\|_2 \leq \zeta\epsilon \tag{2.17}$$

La versione stabile del problema di minimizzazione l_1, presentata in (2.16), è nota in letteratura statistica con la denominazione *Lasso* [32]; ed effettivamente permette di regolarizzare problemi di regressione lineare fortemente sottodeterminati, nei casi in cui la soluzione desiderata debba essere sparsa.

2.3 Algoritmo di classificazione

Data una nuova immagine di test y, relativa ad una classe presente nel training set, si dovrà innanzitutto calcolare la sua rappresentazione sparsa $\hat{x}_1$ attraverso la risoluzione di uno dei due problemi di minimizzazione l_1 (2.14) o (2.16), visti precedentemente. In linea teorica, gli elementi diversi da zero dell'approssimazione $\hat{x}_1$ saranno relativi alle colonne di A, corrispondenti all'individuo che costituisce una certa classe i e, dunque, risulterà immediato assegnare l'esempio di test y a quella determinata classe. Tuttavia, la presenza di rumore e di errori nella definizione del modello possono produrre

elementi diversi da zero, di piccola entità, anche in corrispondenza di immagini di training appartenenti ad altre classi. Sfruttando l'approccio globale di rappresentazione sparsa, di cui si è fin qui discusso, è possibile progettare una grande varietà di classificatori che permettano di gestire questa complicazione; ad esempio, si potrebbe semplicemente associare y al soggetto a cui corrisponde il singolo valore più elevato, presente nella soluzione $\hat{x}_1$. D'altra parte, un approccio euristico di questo tipo non sfrutterebbe a dovere la struttura del sottospazio associato alle immagini dei volti. Per approfittare al meglio di questa struttura lineare, si potrebbe, invece, classificare y basandosi su quanto i coefficienti, associati a tutti gli esempi di training di una classe, riproducano accuratamente l'immagine di test stessa.
Per ogni classe i, sia dunque $\delta_i : \mathbb{R}^n \to \mathbb{R}^n$ la funzione caratteristica che seleziona esclusivamente i coefficienti del vettore soluzione, associati alla classe i-esima. Allora, per $x \in \mathbb{R}^n$, si avrà un nuovo vettore $\delta_i(x) \in \mathbb{R}^n$ i cui elementi diversi da zero saranno contenuti solamente nelle posizioni corrispondenti alle immagini della classe i. A questo punto, è dunque possibile approssimare l'esempio di test y, dato in input, utilizzando per lo scopo solo i valori della soluzione associati all'i-esima classe, in questo modo: $\hat{y}_i = A\delta_i(\hat{x}_1)$.
Una volta calcolati tutti gli $\hat{y}_i$ con $i = 1, 2, ..., k$, sarà finalmente possibile classificare l'esempio di test y, sulla base di queste approssimazioni, assegnandolo alla classe che minimizza il residuo fra y e $\hat{y}_i$:

$$\min_i r_i(y) \doteq \|y - A\delta_i(\hat{x}_1)\|_2 \tag{2.18}$$

L'intero processo di riconoscimento è riassunto in Tabella 2.1.

2.4 Algoritmo SRC ed estrazione delle feature

Nel capitolo precedente è stato già ampiamente trattato il ruolo fondamentale ricoperto dalla fase di estrazione delle feature nei metodi di classificazione tradizionali. Vedremo invece ora che tipo di benefici si possono ottenere da queste tecniche, nel contesto del compressed sensing.
Il primo vantaggio apportato dai metodi di feature extraction, che continua ad essere valido anche per il framework basato sulla rappresentazione sparsa

Algoritmo: Sparse Representation-based Classification (SRC)

1: **Input:** una matrice di esempi di training $A = [A_1, A_2, ..., A_k] \in \mathbb{R}^{m \times n}$ per k classi di volti, un'immagine di test $y \in \mathbb{R}^m$, (ed eventualmente il parametro di tolleranza all'errore $\epsilon > 0$).

2: Normalizzare le colonne di A e il vettore y per ottenere norma l_2 unitaria.

3: Risolvere il problema di minimizzazione l_1:

$$\hat{x}_1 = arg\ \min_x \|x\|_1 \quad s.t.\ \ Ax = y$$

o, in alternativa, risolvere:

$$\hat{x}_1 = arg\ \min_x \|x\|_1 \quad con\ \ \|Ax - y\|_2 \leq \epsilon$$

4: Calcolare il residuo $r_i(y) = \|y - A\delta_i(\hat{x}_1)\|_2$, per $i = 1, ..., k$.

5: **Output:** Identità $(y) = arg\ \min_i r_i(y)$.

Tabella 2.1: Versione base dell'algoritmo SRC

presentato sopra, è quello di ridurre la dimensione del problema e, conseguentemente, il costo computazionale della sua soluzione; infatti, per le immagini di volti in formato originale, il corrispondente sistema lineare $y = Ax$ risulta tipicamente molto grande - per esempio, se le immagini di input avessero la risoluzione tipica di 640 x 480 pixel, la matrice del sistema risultante presenterebbe un numero di righe m nell'ordine di 10^5. Sebbene l'algoritmo presentato in Tabella 2.1 possa usufruire, per la fase di minimizzazione, di metodi scalabili di programmazione lineare, applicarlo direttamente su un database di immagini con una risoluzione così alta resta comunque un'operazione che va al di là delle capacità di calcolo dei computer più diffusi.
Dato che la maggior parte delle trasformazioni utilizzate per l'estrazione delle

Figura 2.3: Alcuni esempi di estrazione delle feature di dimensione 120 applicati su un volto del database Extended Yale B. **(a)** Immagine originale. **(b)** Da sinistra verso destra: eigenfaces, laplacianfaces, downsampling e randomfaces.

feature coinvolgono esclusivamente calcoli lineari (o delle loro approssimazioni), l'operatore che permette di proiettare i volti dall'image space al feature space può essere rappresentato sotto forma di una matrice $R \in \mathbb{R}^{d \times m}$ con $d \ll m$. Applicando, dunque, R a entrambi i membri dell'equazione (2.11) si ottiene:

$$\tilde{y} \doteq Ry = RAx_0 \quad \in \mathbb{R}^d \tag{2.19}$$

In pratica, la dimensione d dello spazio delle feature è generalmente scelta in modo da risultare molto minore di n. In questo caso, il sistema di equazioni $\tilde{y} = RAx \quad \in \mathbb{R}^d$ sarà sottodeterminato con vettore delle incognite $x \in \mathbb{R}^n$. Nonostante questo, dal momento che la soluzione desiderata x_0 è sparsa, possiamo comunque pensare di ricavarla risolvendo il seguente problema *ridotto* di minimizzazione l_1:

$$\hat{x}_1 = arg\,min \|x\|_1 \quad s.t. \quad \|RAx - \tilde{y}\|_2 \leq \epsilon \tag{2.20}$$

per una data tolleranza all'errore $\epsilon > 0$. Quindi, nell'algoritmo in Tabella 2.1, la matrice A che contiene le immagini di training verrà ora sostituita dalla matrice $RA \in \mathbb{R}^{d \times n}$, dove d è la dimensione delle feature. Allo stesso modo, l'immagine di test y dovrà essere rimpiazzata dal vettore delle sue feature $\tilde{y}$. Studi empirici hanno evidenziato che, per i metodi di face recognition esistenti, l'aumento della dimensione d dello spazio delle feature, generalmente, permette di ottenere percentuali di riconoscimento più elevate - chiaramente, finché la distribuzione delle feature RA_i non comincia a degenerare. Questa degenerazione non rappresenta comunque un problema per la minimizzazione l_1, che richiede solamente che $\tilde{y}$ si trovi in prossimità dell'intervallo di RA_i; infatti, essa non dipende dalla necessità che la matrice di covarianza $\Sigma_i = A_i^T R^T RA_i$ risulti non singolare, come invece accade per i metodi classici

di *discriminant analysis*.

Nel paragrafo precedente, si è visto come la nuova teoria del compressed sensing abbia introdotto nuovi limiti, che permettono di ricostruire alcuni segnali sulla base di un numero di campioni significativamente inferiore, rispetto a quello che si era ritenuto fino a quel momento un valore inviolabile. Per stabilire sotto quali condizioni i metodi di estrazione delle feature possano inficiare il funzionamento dell'algoritmo SRC è, dunque, necessario esplorare questi nuovi limiti, applicati al contesto del riconoscimento di oggetti. Sfruttando la rappresentazione geometrica del problema di minimizzazione della norma l_1, è, infine, possibile determinare un limite inferiore piuttosto soprendente; infatti si può dimostrare che, se la soluzione x_0 risulta essere sufficientemente sparsa, allora può essere correttamente ricostruita attraverso l'uso della minimizzazione l_1 a partire da un numero di campioni d sufficientemente grande, rappresentati da $\tilde{y} = RAx_0$. Più precisamente, se x_0 ha $t \ll n$ elementi diversi da zero, allora d misurazioni, anche casuali, sono sufficienti per ricostruire correttamente la soluzione x_0, se vale la seguente relazione:

$$d \geq 2t \log\left(\frac{n}{d}\right). \tag{2.21}$$

Questo sorprendente fenomeno permette di affermare che anche caratteristiche estratte in modo random, purché rispettino la condizione (2.21), sono sufficienti per ottenere la soluzione corretta.
Le feature estratte casualmente possono essere viste come una controparte meno strutturata delle classiche feature applicate nell'ambito della face recognition, come eigenfaces e fisherfaces; dunque, per mantenere una sorta di correlazione fra i nomi dei metodi convenzionali e questa nuova tecnica di estrazione, definiremo *randomfaces* la proiezione lineare generata da una matrice di valori casuali, con distribuzione gaussiana.

Randomfaces: *si consideri una matrice di trasformazione $R \in \mathbb{R}^{d \times m}$ i cui elementi siano stati indipendentemente campionati a partire da una distribuzione normale con media nulla, e le cui righe siano state normalizzate in maniera unitaria. I vettori riga della matrice R potranno dunque essere considerati come d volti random in $\mathbb{R}^m$.*

Uno dei maggiori vantaggi del metodo randomfaces è quello di risultare estremamente efficiente nel generare la matrice di proiezione, dal momento che essa è del tutto indipendente dalle informazioni contenute nel training set. Questo beneficio si può rivelare particolarmente importante per quei sistemi di face recognition in cui, per varie ragioni, non si dispone di un database sufficientemente ampio per rappresentare adeguatamente i volti in un sottospazio di feature classico, generato sulla base dei dati di training; o ancora per quei contesti in cui i soggetti contenuti nel database possono variare nel tempo. In questo ultimo caso, il vantaggio consiste nel non dover ricalcolare la matrice di trasformazione R, dopo l'aggiunta dei nuovi individui.
Quanto finora sostenuto ci permette di concludere che, in questo contesto di classificazione, se la dimensione d dei vettori delle feature rispetta il limite (2.21), qualsiasi metodo di estrazione delle caratteristiche - anche non convenzionale, come il sopra descritto randomfaces, o come il semplice downsampling - si rivela sufficientemente adatto allo scopo di ricostruire correttamente la soluzione x_0.

2.5 Robustezza alle occlusioni

In numerosi scenari pratici legati al riconoscimento di volti, l'immagine di test y potrebbe presentare forti perturbazioni o essere addirittura parzialmente occlusa. In questi frangenti, si rivela indispensabile modificare il modello lineare, presentato nella (2.11), in questo modo:

$$y = y_0 + e_0 = Ax_0 + e_0 \tag{2.22}$$

dove $e_0 \in \mathbb{R}^m$ è un vettore che rappresenta l'errore esistente, costituito da una certa quantità ρ di elementi diversi da zero. La posizione della perturbazione può variare da immagine a immagine e non può essere quindi predetta in maniera automatica; per di più, gli errori potrebbero essere anche piuttosto rilevanti, rendendo di fatto inefficaci le tecniche di stabilizzazione del problema presentate precedentemente.

Uno dei principi fondamentali della teoria dell'informazione è quello di sfruttare la *ridondanza* dei dati per individuare e correggere gli errori di grandi

dimensioni. Nel contesto del riconoscimento di volti, la ridondanza dipende dal fatto che il numero di pixel che costituiscono le immagini è generalmente molto più elevato rispetto al numero di soggetti in esse rappresentati. In questo scenario, anche se una porzione di pixel risulta completamente occlusa, il riconoscimento potrà ancora essere eseguito basandosi sull'informazione apportata dai restanti pixel. D'altra parte, le tecniche di estrazione delle feature riducono la quantità di dati a disposizione, diminuendo conseguentemente anche la ridondanza in essi presente, che potrebbe invece essere utilizzata per compensare, appunto, la presenza di eventuali occlusioni. In questo senso, dunque, nessuna rappresentazione risulterà essere più ridondante, robusta o informativa delle immagini originali dei volti stesse. Detto questo, quando si assume di dover prendere in considerazione il problema delle occlusioni o di forti perturbazioni, si dovrebbe altresì considerare la possibilità di lavorare su immagini con la più alta risoluzione possibile - effettuando un sottocampionamento o estraendo un qualche altro tipo di feature soltanto nel caso in cui la dimensione originale dei dati rendesse particolarmente inefficiente la computazione.

Chiaramente, il mantenimento della ridondanza risulterebbe del tutto inutile, senza l'introduzione di opportune metodologie, che permettano di sfruttarla nel processo di ricostruzione. Purtroppo, le difficoltà intrinseche nella gestione diretta della ridondanza presente nelle immagini originali ha condotto i ricercatori a concentrarsi piuttosto sul concetto di *località spaziale*, dando vita, in alternativa, ad alcune tecniche locali di estrazione delle feature, che sono già state presentate nel capitolo precedente. Si noti che questi metodi, tuttavia, trasformano il dominio relativo al problema delle occlusioni, senza eliminare, invece, le occlusioni stesse; per di più, gli errori presenti nei pixel originali diventano errori nel sottospazio di proiezione, perdendo anche la loro località. Su queste basi, si potrebbe, dunque, mettere in discussione il ruolo stesso delle tecniche di feature extraction nell'ottenere informazioni relative alla località spaziale; infatti, *nessuna base o insieme di feature risulta più spazialmente localizzato dei pixel stessi dell'immagine originale* [1].

Si mostrerà ora come estendere il metodo di classificazione basato sulla rappresentazione sparsa, in modo che possa anch'esso trattare immagini di volti affetti da occlusione.

Si assuma che il numero di pixel perturbati costituisca una porzione relativamente piccola dell'immagine di test. Il vettore dell'errore e_0 risulterà dunque essere sparso, così come il vettore soluzione x_0. Dal momento che $y_0 = Ax_0$, è possibile riscrivere la relazione (2.22) nel seguente modo:

$$y = [A, I] \begin{bmatrix} x_0 \\ e_0 \end{bmatrix} \doteq B\, w_0 \tag{2.23}$$

Nella formula sopra riportata si ha $B = [A, I] \in \mathbb{R}^{m\times(n+m)}$, dunque, il sistema $y = Bw$ continua ad essere sottodeterminato e non ammetterà un'unica soluzione w. Inoltre, sulla base di quanto finora affermato riguardo alla sparsità di x_0 ed e_0, è possibile intuire che il vettore corretto $w_0 = [x_0, e_0]$ dovrà avere al più $n_i + \rho m$ elementi diversi da zero; possiamo quindi continuare a sperare di ricostruire w_0 come soluzione più sparsa del sistema $y = Bw$. Infatti, nel caso generale, se vale la relazione $y = B\tilde{w}$, per un qualche vettore $\tilde{w}$ con meno di $m/2$ elementi diversi da zero, allora $\tilde{w}$ rappresenta l'unica soluzione sparsa. In definitiva, se l'occlusione e copre meno di $\frac{m-n_i}{2}$ pixel, che corrisponderebbe circa al 50% dell'immagine, la soluzione più sparsa $\tilde{w}$ coincide con la soluzione $w_0 = [x_0, e_0]$ del sistema descritto nella (2.23).
Più in generale, si può ammettere che esista una rappresentazione sparsa della perturbazione e_0, rispetto ad una qualche base $A_e \in \mathbb{R}^{m\times n_e}$; il che corrisponde ad avere $e_0 = A_e u_0$ per un qualche vettore sparso $u_0 \in \mathbb{R}^m$.
Nella (2.23), è stato considerato il caso speciale in cui $A_e = I \in \mathbb{R}^{m\times m}$, e ciò si deve all'assunzione che l'errore e_0 sia sparso rispetto alle coordinate originali dei pixel. Nel caso in cui, invece, l'errore e_0 risultasse essere maggiormente sparso rispetto ad un'altra base - come, ad esempio, quella di Fourier o di Haar - basterà semplicemente ridefinire la matrice B concatenando una qualche altra matrice A_e alla matrice A, ottenendo il nuovo sistema di equazioni:

$$y = Bw \;\; con \;\; B = [A, A_e] \;\; \in \mathbb{R}^{m\times(n+n_e)} \tag{2.24}$$

In questo modo, è possibile gestire attraverso la medesima formulazione un numero maggiore di classi generiche di perturbazione.
Come già visto per la versione base dell'algoritmo, anche in questo caso si

procede cercando di ricavare la soluzione più sparsa w_0, risolvendo il seguente problema *esteso* di minimizzazione della norma l_1:

$$\tilde{w}_1 = arg\,min \|w\|_1 \quad s.t. \;\; Bw = y \tag{2.25}$$

Il che consiste nel sostituire, nell'algoritmo presentato in Tabella 2.1, la matrice A con la matrice $B = [A, I]$ e il vettore x con $w = [x, e]$.
In questo scenario non risulta più necessario utilizzare la versione stabile vista in (2.16), poiché, con l'introduzione della matrice estesa $B = [A, I]$ e del vincolo esatto $Bw = y$, è possibile gestire non soltanto la presenza di occlusioni, ma anche perturbazioni più moderate.
Dopo aver calcolato la soluzione sparsa $\tilde{w}_1 = [\tilde{x}_1, \tilde{e}_1]$, si può ottenere una ricostruzione dell'immagine, liberata dall'occlusione attraverso l'uso della compensazione dell'errore, sfruttando la formula $y_r \doteq y - \tilde{e}_1$. Per identificare il volto del soggetto fornito in input, è necessario apportare una leggera modifica anche alla formula di calcolo dei residui introdotta per l'algoritmo precedente; in particolare si avrà:

$$r_i\,(y) = \|y_r - A\delta_i\,(\tilde{x}_1)\|_2 = \|y - \tilde{e}_1 - A\delta_i\,(\tilde{x}_1)\|_2 \tag{2.26}$$

2.6 Meccanismo di validazione

Prima di procedere con la classificazione di un volto di test, è necessario stabilire se esso costituisca un esempio valido - ovvero, relativo ad una classe conosciuta - oppure se appartenga ad un soggetto non presente nel training set. La capacità di individuare e rigettare gli esempi di test non validi, o *outlier*, è una caratteristica cruciale per i sistemi di face recognition utilizzati in contesti reali - potrebbe infatti accadere che vengano sottoposte ad un sistema immagini che rappresentano volti di individui sconosciuti, o addirittura immagini che non contengono alcun volto.
Impianti di riconoscimento di volti, basati sull'uso di classificatori convenzionali come nearest neighbour o nearest subspace, spesso utilizzano i valori dei residui $r_i\,(y)$, sia per il processo di validazione che per quello di identificazione: ovvero, accettano o rifiutano un'immagine, in base a quanto risulta essere piccolo il residuo di minore entità. Tuttavia, ogni residuo $r_i\,(y)$ viene calcolato senza sfruttare nessuna ulteriore conoscenza, che coinvolga anche

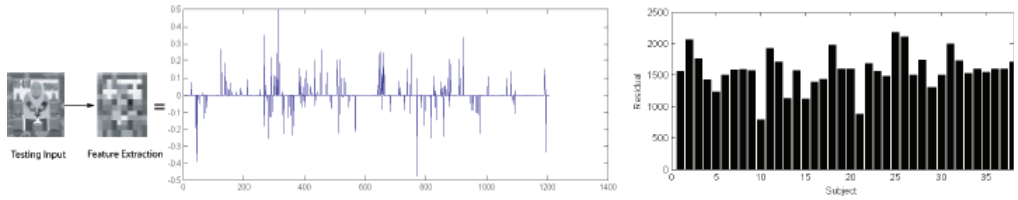


Figura 2.4: Soluzione di un'immagine non valida.

le altre immagini del training set, ma misura esclusivamente la somiglianza fra l'esempio di test ed ogni singola classe.

Nel paradigma della rappresentazione sparsa, i coefficienti della soluzione $\tilde{x}_1$ vengono invece calcolati globalmente, prendendo in considerazione tutte le immagini di tutte le classi, e possono dunque sfruttare, per la fase di validazione, la loro distribuzione congiunta. Per questo motivo, è possibile affermare che i coefficienti di $\tilde{x}$ costituiscono un'informazione statisticamente più significativa rispetto ai residui, per l'espletamento di questa funzione. Si dimostrerà quest'affermazione attraverso un esempio.

A questo scopo è stata selezionata un'immagine casuale, che non rappresenta alcun volto, ed è stata sottocampionata alla dimensione 12×10. Dopodiché, si è proceduto alla ricostruzione dell'immagine, sulla base di alcuni esempi di training estratti dall'Extended Yale B database. In Figura 2.4 a sinistra sono rappresentati i coefficienti della soluzione trovata, mentre a destra si trovano i corrispondenti residui. Se si confrontano i coefficienti ottenuti sottoponendo al sistema un'immagine di test valida - come in Figura 2.1 - con quelli ricavati in questo esperimento, si potrà notare che, mentre nel primo caso i coefficienti maggiori in modulo sono concentrati in corrispondenza di immagini appartenenti sempre alla stessa classe, nel secondo, d'altra parte, sono distibuiti su tutto il training set.

Dunque, si presume che un'immagine di test valida debba avere una rappresentazione sparsa i cui elementi diversi da zero siano concentrati, per la maggior parte, in corrispondenza delle immagini di training di uno stesso soggetto.

Al fine di quantificare questa osservazione, verrà definita una nuova unità di misura, che descriva quanto siano concentrati i coefficienti della soluzione, rispetto ad ogni singola classe presente nel database di training, e che verrà denominata **SCI** (da *Sparsity Concentration Index*, o *indice di concentrazio-*

ne della sparsità).
L'indice SCI di un vettore $x \in \mathbb{R}^n$ può essere descritto come:

$$SCI\left(x\right) \doteq \frac{k \cdot \max_i \left\|\delta_i\left(x\right)\right\|_1 / \left\|x\right\|_1 - 1}{k-1} \in [0,1] \tag{2.27}$$

Quindi, se la soluzione $\tilde{x}$ trovata dall'algoritmo in Tabella 2.1 è tale per cui $SCI\left(\tilde{x}\right) = 1$, l'immagine di test sarà stata ricostruita utilizzando gli esempi di training relativi ad un unico individuo, viceversa, se $SCI\left(\tilde{x}\right) = 0$, significa che i coefficienti della soluzione trovata saranno distribuiti su tutte le classi. Si rivelerà, dunque, necessario stabilire un nuovo valore soglia $\tau \in (0,1)$, sfruttando il quale il sistema accetterà come valida un'immagine di test, secondo la seguente regola:

$$SCI\left(\tilde{x}\right) \geq \tau \tag{2.28}$$

In caso contrario, l'esempio di test sarà rigettato. Si potrà quindi procedere con la modifica del passo 5 dell'algoritmo in Tabella 2.1, producendo in output l'identità presunta del soggetto rappresentato in y, soltanto se la ricostruzione dell'immagine soddisfa il criterio sopra descritto.
Si torna a sottolineare il fatto che, contrariamente a quanto avviene nei metodi classici di face recognition, nearest neighbour e nearest subspace, l'uso di questa nuova condizione evita di dover considerare il residuo $r_i\left(y\right)$ anche per la fase di validazione; l'approccio basato sulla rappresentazione sparsa, potrà infatti sfruttare due tipi differenti d'informazione: i residui per l'identificazione e la concentrazione dei coeffcienti sparsi per la validazione. In particolare, si può affermare che, mentre i residui descrivono con quale precisione la rappresentazione trovata approssima l'immagine di test, l'SCI, invece, misura la qualità della soluzione in se stessa, in termini di localizzazione.
Il vantaggio principale raggiunto attraverso l'utilizzo di questo nuovo criterio consiste nel riuscire a migliorare le performance generali del sistema di face recognition, rendendolo più robusto nei confronti di quelle immagini di volti sconosciuti, che risultano, però, molto simili ad alcuni soggetti noti. Si può quindi concludere questa sezione, affermando che la nuova regola presentata permette di giudicare più correttamente se una data immagine di test rappresenti un viso generico o un volto di un particolare individuo presente nel database di training. Questa affermazione è rafforzata dai risultati empirici

ottenuti e presentati nel capitolo dedicato alla sperimentazione; lì verrà mostrato anche come questo nuovo criterio di validazione risulti ulteriormente potenziato dall'utilizzo di funzioni sparsificanti, alternative alla minimizzazione della norma l_1, che verranno introdotte nel prossimo capitolo.
Prima di procedere oltre, tuttavia, verrà riportata in Tabella 2.2, la nuova versione *robusta* dell'algoritmo SRC, ottenuta introducendo l'estensione della matrice per la compensazione dell'errore e il nuovo metodo di validazione.

Algoritmo: Sparse Representation-based Classification (SRC)
(versione robusta)

1: **Input:** una matrice di esempi di training $A = [A_1, A_2, ..., A_k] \in \mathbb{R}^{m \times n}$ per k classi di volti, un'immagine di test $y \in \mathbb{R}^m$, una matrice A_e che descrive la base di sparsità dell'errore, una matrice R di trasformazione, per l'estrazione delle feature (ed eventualmente il parametro di tolleranza all'errore $\epsilon > 0$).

2: Estrarre le feature: $\tilde{A} = RA$ e $\tilde{y} = Ry$.

3: Estendere la matrice: sia $B = \left[\tilde{A}, A_e\right]$.

4: Normalizzare le colonne di B e il vettore $\tilde{y}$ per ottenere norma l_2 unitaria.

5: Risolvere il problema di minimizzazione l_1:
$$\hat{w}_1 = arg\ \min_w \|w\|_1 \quad s.t.\ \ Bw = \tilde{y}$$

6: Separare le due componenti della soluzione $\hat{w}_1 = [\hat{x}_1, \hat{e}_1]$, ottenendo $\hat{x}_1$ e $\hat{e}_1$.

7: Calcolare l'indice SCI:
$$SCI\left(\hat{x}_1\right) = \frac{k\, max_i \|\delta_i(\hat{x}_1)\|_1 / \|\hat{x}_1\|_1 - 1}{k-1}$$

8: Se $\left(SCI\left(\hat{x}_1\right) \geq \tau\right)$
Calcolare il residuo $r_i\left(\tilde{y}\right) = \left\|\tilde{y} - \hat{e}_1 - \tilde{A}\delta_i\left(\hat{x}_1\right)\right\|_2$,
per $i = 1, ..., k$.

9: **Output:**
Se $\left(SCI\left(\hat{x}_1\right) \geq \tau\right)$
Identità $(y) = arg\ \min_i r_i\left(\tilde{y}\right)$.
altrimenti
rifiutare l'immagine.

Tabella 2.2: Versione robusta dell'algoritmo SRC

Capitolo 3

Risoluzione di approcci non convessi al problema del compressed sensing

Nel capitolo precedente è stato dimostrato come sia possibile sfruttare la teoria del compressed sensing nel contesto del riconoscimento di volti. In particolare, è stato presentato un framework di classificazione basato sulla rappresentazione sparsa (SRC), in grado di ottenere percentuali di riconoscimento molto più elevate, rispetto ai tradizionali metodi di face recognition, noti come nearest neighbour e nearest subspace. Inoltre, è stata dimostrata l'indipendenza di questo nuovo metodo dalle tecniche di feature extraction introdotte nel primo capitolo e si è infine evidenziata la possibilità di estendere il modello base, in modo da rendere questo classificatore più robusto rispetto alla presenza di eventuali occlusioni o forti perturbazioni.
In questo capitolo, invece, verrà presentata una nuova infrastruttura algoritmica, denominata WNFCS - da *Weighted Nonlinear Filters for Compressed Sensing* - che permetterà di sostituire il passo di minimizzazione della norma l_1, con la minimizzazione di funzioni sparsificanti alternative, generalmente non convesse, che consentano di ottenere soluzioni più sparse e dunque più vicine alla soluzione del problema originale di minimizzazione della norma l_0. In seguito, si mostrerà come sfruttare questo nuovo framework per introdurre un successivo miglioramento, in grado di incoraggiare ulteriormente la sparsità della soluzione, sfruttando il concetto di sparsità strutturata.

1 Funzioni sparsificanti non convesse

Nel capitolo precedente, si è visto come la teoria del compressed sensing permetta di ricostruire un segnale K-sparso, a partire da un numero molto basso di misurazioni lineari, attraverso la risoluzione di un problema di minimizzazione vincolato. In particolare, si è mostrato come un vettore sconosciuto $x \in \mathbb{R}^N$, con soli K elementi diversi da zero, dove tipicamente $K \ll N$, possa essere esattamente ricostruito a partire dalle sue misurazioni $y = \Phi x$, essendo nota la matrice di acquisizione Φ di dimensione $M \times N$, risolvendo il problema (2.7).
Tuttavia, recentemente sono state proposte in letteratura molteplici alternative al problema di minimizzazione della norma l_1 che permettono di ricostruire un segnale sparso a partire da un numero di campioni ancora più basso. In questo capitolo, si prenderanno in considerazione funzioni sparsificanti non convesse, poiché sembra che siano in grado di riprodurre in maniera più fedele la capacità della minimizzazione della norma l_0 di indurre la sparsità nella soluzione. Fra queste, si ricordano la norma l_q, per $0 < q < 1$ [33] [34], le funzioni *log-sum* e *atan* [35] [36] e, infine, la funzione *log-exp* [2]. In Tabella 3.1 sono riportate tutte queste funzioni sparsificanti, insieme alle loro derivate prime.
Per poter introdurre queste nuove funzioni obiettivo nella formulazione del problema di ricostruzione di un segnale K-sparso, sarà necessario sfruttare l'enunciato (2.3), riprodotto anche in questa sezione per comodità del lettore:

$$\min_{u \in \mathbb{R}^N} F(u) \quad s.t. \;\; \Phi u = y$$

dove, si ricorda, $F(u)$ rappresenta un'opportuna funzione sparsificante. Sostituendo ad $F(u)$ una delle funzioni sopra citate, si otterrà dunque un problema di minimizzazione non convesso.
Nella ricerca di funzioni sparsificanti alternative alla minimizzazione delle due norme l_0 e l_1, si è sempre cercato di mantenere le proprietà di separabilità presentate da entrambe, concentrandosi dunque maggiormente su funzioni

$F(u)$	ψ	ψ'
log-sum	$\psi(\|u_i\|) = \frac{1}{\log(1+\epsilon)} \log(\|u_i\| + \epsilon)$	$\psi'(\|u_i\|) = \frac{1}{\|u_i\|+\epsilon}$
atan	$\psi(\|u_i\|) = atan\left(\frac{\|u_i\|}{\epsilon}\right)$	$\psi'(\|u_i\|) = \frac{\epsilon}{\epsilon^2+\|u_i\|^2}$
l_q	$\psi(\|u_i\|) = \|u_i\|_q$	$\psi'(\|u_i\|) = \frac{q}{\|u_i\|^{1-q}+\epsilon}$
log-exp	$\psi_\mu(\|u_i\|) = \frac{1}{\log(2)} \log\left(\frac{2}{1+e^{\frac{-\|u_i\|}{\mu}}}\right)$	$\psi'_\mu = \frac{1}{\mu \log(2)} \cdot \frac{1}{1+e^{\frac{\|u_i\|}{\mu}}}$

Tabella 3.1: Alcune funzioni sparsificanti non convesse.

obiettivo che potessero essere espresse nel modo seguente:

$$F(u) = \sum_{i=1}^{N} \psi(|u_i|) \quad u \in \mathbb{R}^N \tag{3.1}$$

dove $\psi : \mathbb{R}_+ \longrightarrow \mathbb{R}_+$ deve garantire che la funzione $F(u)$ riproduca al meglio l'azione sparsificante della norma l_0, pur mantenendo alcune comode proprietà della norma l_1, come la continuità e la differenziabilità (per $u \neq 0$).

Fra tutte le possibilità presentate in Tabella 3.1, in questa tesi ci si concentrerà principalmente sulla funzione *log-exp*, introdotta in [2] poiché, come verrà presto evidenziato, essa presenta alcune caratteristiche interessanti.
La funzione *log-exp* è innanzitutto definita nel seguente modo:

$$\psi_\mu(|t|) = \frac{1}{\log(2)} \log\left(\frac{2}{1+e^{\frac{-|t|}{\mu}}}\right), \quad \mu > 0 \tag{3.2}$$

e gode delle seguenti proprietà:

- $\psi_\mu(t)$ è concava e non decrescente per $t \in [0, \infty)$;
- $\psi_\mu(t)$ ha derivata continua e limitata, per $t \in (0, \infty)$ ed è singolare per $t = 0$;
- per la sua derivata

$$\psi'_\mu(t) = \frac{1}{\mu \log(2)} \cdot \frac{1}{1+e^{\frac{|t|}{\mu}}}$$

si ha:

$$\begin{cases} \psi'_\mu(t) \to 0 \ \ per\ t \gg \mu; \\ \psi'_\mu(t)\ è\ grande \ \ per\ t \ll \mu. \end{cases}$$

Le proprietà sopraelencate sono proprio quelle richieste affinché una funzione abbia una buona capacità sparsificante ed è dunque per questo motivo che si è deciso di concentrarsi su di essa. Per di più, la funzione *log-exp* si differenzia dalle altre funzioni non convesse, presentate in Tabella 3.1, per la sua dipendenza dal parametro μ, il cui ruolo risulta essere ambivalente: esso agisce innanzitutto come strumento di modellazione della forma, dal momento che, più $\mu \longrightarrow 0$, più la funzione (3.2) approssima in maniera continua la norma l_0; inoltre, durante il processo di ricostruzione, esso permette di discriminare i coefficienti di grande entità - ovvero quelli maggiori di μ - che si vogliono mantenere diversi da zero, da quelli di piccola entità che, al contrario, si vogliono sopprimere, rendendoli uguali a zero. Questo procedimento, come verrà meglio evidenziato in seguito, costituisce una delle basi fondamentali dell'algoritmo WNFCS.

2 Metodi iterativi pesati per problemi di minimizzazione non convessi

Com'è stato già evidenziato, le funzioni sparsificanti non convesse presentano, rispetto alla norma l_1, un comportamento che approssima in maniera più precisa l'azione d'induzione della sparsità, tipica della norma l_0. Tuttavia, questo vantaggio rende il problema di ottimizzazione risultante decisamente più difficile da risolvere.
Un approccio classico, che permette di affrontare la non convessità della funzione $F(u)$ nel problema (2.3), consiste nel sostituire iterativamente la funzione obiettivo con un suo upper bound $f(u)$, più semplice da risolvere, il quale viene calcolato, ad ogni passo, sfruttando la soluzione ottenuta all'iterazione precedente. Una possibile scelta di $f(u)$ consiste nell'approssimazione quadratica locale di $F(u)$, calcolata per $u \in I_{\overline{u}}$:

$$f(u) = \sum_{i=1}^{N} \phi(|u_i|) = \sum_{i=1}^{N} (\psi(|\overline{u}_i|) + \psi'(|\overline{u}_i|)(|u_i| - |\overline{u}_i|)) + \beta_0(|u_i| - |\overline{u}_i|)^2 \tag{3.3}$$

dove $\beta_0 \geq 0$ è una costante scelta opportunamente. Questo approccio conduce all'algoritmo denominato *iteratively reweighted least squares (IRLS)*, il cui utilizzo consente di minimizzare con successo un'ampio insieme di funzioni obiettivo non convesse.
Un'altra proposta per la rappresentazione dell'upper bound convesso di $F(u)$ si basa sull'uso dell'approssimazione lineare locale della funzione non convessa, in prossimità del punto $\overline{u}$, riportata sotto:

$$f(u) = \sum_{i=1}^{N} \phi(|u_i|) = \sum_{i=1}^{N} (\psi(|\overline{u}_i|) + \psi'(|\overline{u}_i|)(|u_i| - |\overline{u}_i|)) \tag{3.4}$$

La funzione lineare sopra descritta può facilmente presentare le proprietà di maggiorizzazione richieste, sfruttando la concavità della funzione ψ, la quale sta sempre sotto la sua tangente.
Inserendo l'approssimazione lineare locale appena presentata nel contesto di un approccio di tipo Lagrangiano, sarà possibile rilassare il problema di minimizzazione non convesso della (2.3), ottenendo un problema convesso non vincolato, che possa essere quindi risolto iterativamente sfruttando la procedura seguente:

$u^0 \leftarrow arg\, min_{u \in \mathbb{R}^N} \{\lambda \sum_{i=1}^{N} |u_i| + \frac{1}{2} |\Phi u - y|_2^2\}$
$k \leftarrow 1$
`repeat`
  $w^k \leftarrow \psi'(|u^{k-1}|)$
  $u^k \leftarrow arg\, min_{u \in \mathbb{R}^N} \{\lambda \sum_{i=1}^{N} w_i^k |u_i| + \frac{1}{2} |\Phi u - y|_2^2\}$
  $k \leftarrow k + 1$
`until convergence`

L'algoritmo iterativo sopra riportato, al primo passo si comporta come la classica minimizzazione l_1; nel seguito della computazione, tuttavia, esso risolve una sequenza di problemi di mininimizzazione l_1 pesati, utilizzando come pesi, ad ogni passo, il gradiente della funzione obiettivo originale, valutato rispetto alla soluzione trovata all'iterazione precedente. Questa procedura, nota in letteratura come *iterative reweighted* l_1 *(IRl*$_1$*)*, rappresenta uno degli algoritmi più utilizzati per la risoluzione del problema (2.3) nei

casi in cui la funzione $F(u)$ risulti essere non convessa. La garanzia di convergenza di questo metodo iterativo pesato verso un minimo della funzione obiettivo di partenza può essere ricavata facilmente; infatti, dal momento che l'approssimazione lineare locale (3.4) soddisfa le seguenti relazioni,

$$\phi(|\overline{u}_i|) = \psi(|\overline{u}_i|) \;\; e \;\; \phi(|u_i|) \geq \psi(|u_i|) \tag{3.5}$$

l'algoritmo IRl_1 rientrerà nella classe più generale dei metodi di *majorize-minimization (MM)*, ereditando dunque le loro ben note proprietà di convergenza.
Tuttavia, è importante sottolineare che, data la non convessità della funzione obiettivo originale, entrambe le procedure iterative IRLS e IRl_1 potrebbero convergere in un minimo locale; inoltre, queste tecniche hanno lo svantaggio di essere particolarmente pesanti, dal punto di vista computazionale.

3 Una nuova strategia di minimizzazione

In questa sezione verrà esposta una nuova strategia, presentata per la prima volta in [2], per la soluzione del seguente problema di minimizzazione vincolato e non convesso:

$$\min_{u \in \mathbb{R}^N} \sum_{i=1}^{N} \psi(|u_i|) \;\; s.t. \;\; \Phi u = y \tag{3.6}$$

dove $\psi : \mathbb{R}_+ \longrightarrow \mathbb{R}_+$ è, appunto, una funzione non convessa. Il nuovo metodo proposto sfrutta la tecnica iterativa pesata, basata sulla LLA, introdotta nel paragrafo precedente, integrandola all'interno di un metodo di penalizzazione basato sullo splitting, che sarà presentato nella prossima sezione. Inoltre, sebbene la tecnica che ci si appresta a mostrare costituisca un framework algoritmico di carattere generale in grado di considerare tutte le funzioni non convesse raffigurate in Tabella 3.1,nel seguito della dissertazione si discuterà della sua applicazione specifica considerando la funzione sparsificante *log-exp*, descritta nella formula (3.2).

3.1 Strategia di penalizzazione basata sullo splitting

Nella letteratura recente, sono stati proposti numerosi metodi di *proximal splitting* per la risoluzione di un problema generale di minimizzazione della somma di due funzionali convessi $H(u)$ e $G(u)$, di cui uno risulta essere non derivabile. In particolare, si ricorda che l'operatore di prossimità di un funzionale $G(u)$ non derivabile è definito come la funzione di v che soddisfa la relazione:

$$prox_G(v) = \min_u \left\{ G(u) + \frac{1}{2\beta} \|u - v\|_2^2 \right\}$$

con $\beta > 0$.

È stato inoltre dimostrato che, sotto l'assunzione che il funzionale $H(u)$ sia convesso e L-Lipschitz continuamente differenziabile, la soluzione del problema di minimizzazione originale è caratterizzata dalla seguente equazione di punto fisso:

$$u = prox_{\beta G}(u - \beta \nabla H(u)). \tag{3.7}$$

Per $\beta < 2/L$, la soluzione dell'equazione precedente può essere valutata mediante il seguente schema iterativo di splitting *forward-backward* (FBS):

$$u_{n+1} = prox_{\beta G}(u_n - \beta \nabla H(u_n)) \tag{3.8}$$

dove l'operazione di aggiornamento esplicito rappresenta il passo in avanti, mentre la valutazione eseguita dall'operatore di prossimità rappresenta l'implicito passo a ritroso. La convergenza di questo schema iterativo è stata dimostrata in [37].

Detto questo, per risolvere il problema convesso (2.7) è possibile integrare lo schema FBS all'interno di un metodo di penalizzazione che permetta di approssimare il problema vincolato originale con una sequenza di problemi non vincolati, della seguente forma:

$$\min_{u \in \mathbb{R}^N} \left\{ \sum_{i=1}^{N} |u_i| + \frac{1}{2\lambda_k} \|\Phi u - y\|_2^2 \right\} \tag{3.9}$$

dove $\lambda_1 > \lambda_2 > ... > \lambda_{min}$ rappresenta una sequenza decrescente di valori del parametro di penalizzazione, la quale assicura la convergenza della soluzione penalizzata con quella del problema originale. I problemi di minimizzazione (3.9) possono, a questo punto, essere risolti efficientemente utilizzando

il seguente schema di splitting FBS con $\beta < \frac{2}{\lambda_{max}\Phi^T\Phi}$:

```
Dati λ_k, β, y, Φ
Inizializza u_0 ← Φ^T y
n ← 0
repeat
  Passo forward di aggiornamento
  v_{n+1} ← u_n + βΦ^T(y − Φu_n)
  Passo backward di minimizzazione
  u_{n+1} ← arg min_{u∈R^N} { Σ_{i=1}^N |u_i| + 1/(2λ_kβ) ‖u − v_{n+1}‖_2^2 }
  n ← n + 1
until convergence
```

È interessante notare come il passo backward di minimizzazione, che coinvolge l'operatore di prossimità del funzionale non derivabile $\lambda_k G(u)$, consista nel risolvere la formulazione variazionale di un classico problema di denoising, in cui il segnale perturbato è costituito dall'output del precedente passo di aggiornamento. Nel caso in cui $G(u) = \sum_{i=1}^{N} |u_i|$, la soluzione esatta del passo a ritroso sarà data dal noto operatore non lineare di soft thresholding $S_{\lambda\beta}(\cdot)$, in base al quale si avrà:

$$u_{n+1,i} = S_{\lambda\beta}(v_{n+1,i}) = \ sign(v_{n+1,i}) \max\{|v_{n+1,i}| - \lambda\beta, 0\}, \ i = 1, ..., N. \tag{3.10}$$

3.2 Algoritmo WNFCS

Come è già stato accennato precedentemente, nel caso in cui la funzione sparsificante $F(u) = \sum_{i=1}^{N} \psi(|u_i|)$ risulti essere non convessa, il conseguente problema di minimizzazione si rivelerà più difficile da risolvere. In questa sezione si mostrerà, dunque, una rappresentazione schematica dell'algoritmo WNFCS (*Weighted Nonlinear Filters for Compressed Sensing*), in quanto esso concretizza con successo l'auspicata fusione fra gli approcci iterativi pesati e le strategie di penalizzazione basate sullo splitting. Questo nuovo metodo, successivamente utilizzato in fase di sperimentazione, consiste in un'infrastruttura algoritmica di carattere generale, la cui struttura è mostrata in Tabella 3.2.

Algoritmo: WNFCS

1: **Input:** la matrice di acquisizione Φ, il vettore delle misurazioni lineari y, la derivata della funzione sparsificante non convessa ψ'_μ, un algoritmo di accelerazione di tipo FISTA A_{τ_n}, $\gamma > 0$, $\mu > 0$, $\lambda_0 > 0$, $\beta > 0$, il fattore di riduzione $r_\lambda \in (0, 1)$

2: *Inizializzazione*: $u_0^0 \leftarrow \tilde{u}_0^0 \leftarrow \Phi^T y$, $w^0 \leftarrow \psi'_\mu(|u_0^0|)$

Ciclo esterno di penalizzazione

3: `for k = 0, 1, ... do`
$n \leftarrow 0$

Minimizzazione l_1 pesata

4: {*Passo forward di aggiornamento*}
$v_{n+1}^k \leftarrow u_n^k + \beta\Phi^T(y - \Phi u_n^k)$

5: {*Passo backward di minimizzazione*}
$\tilde{u}_{n+1}^k \leftarrow arg\ \min_{u\in\mathbb{R}^N} \left\{ \sum_{i=1}^N \lambda_k w_i^k |u_i| + \frac{1}{2\beta} \left\| u - v_{n+1}^k \right\|_2^2 \right\}$

6: {*Strategia di accelerazione*}
$u_{n+1}^k \leftarrow A_{\tau_n}(\tilde{u}_{n+1}^k, \tilde{u}_n^k)$

7: {*Ciclo interno + test di uscita*}
$F(u_{n+1}^k, \lambda_k) \leftarrow \sum_{i=1}^N \lambda_k w_i^k |u_{n+1,i}^k| + \frac{1}{2\beta} \left\| \Phi u_{n+1}^k - y \right\|_2^2$

`if` $\frac{|F(u_{n+1}^k,\lambda_k) - F(u_n^k,\lambda_k)|}{|F(u_{n+1}^k,\lambda_k)|} > \gamma \cdot \lambda_k$
$n \leftarrow n + 1$
`goto 4.`

`else`
$\lambda_{k+1} \leftarrow \lambda_k \cdot r_\lambda$ {*Aggiornamento di λ_k*}
$w^{k+1} \leftarrow \psi'_\mu(|u_{n+1}^k|)$ {*Aggiornamento dei pesi*}
$u_0^{k+1} \leftarrow u_{n+1}^k$

`until convergence`

9: **Output:** u_0^{k+1} come approssimazione di x.

Tabella 3.2: Struttura dell'algoritmo WNFCS

Come si può notare, esso è composto da due cicli innestati: uno esterno che decrementa il parametro di penalizzazione λ_k - operazione indispensabile per la convergenza del metodo - ed esegue l'aggiornamento dei pesi $w_k = \psi'_\mu(|u^{k-1}|)$ - com'è invece richiesto dalla tecnica iterativa pesata -; e uno interno che, d'altra parte, risolve una sequenza di sottoproblemi non vincolati di minimizzazione pesata della norma l_1, attraverso una serie di passi di aggiornamento, seguiti da altrettanti passi di filtraggio non lineare accelerati dalla tecnica nota come Fista. Quest'ultima fase è stata realizzata sfruttando un operatore di thresholding pesato, definito nel seguente modo:

$$S_{w_i^k \lambda_k}(u_i^k) = sign(u_i^k) \max\{|u_i^k| - w_i^k \lambda_k, 0\} \tag{3.11}$$

dove $w_i^k = \psi'(|u_i^{k-1}|)$.
La semplice struttura del framework WNFCS consente di accorgersi facilmente che la sua complessità computazionale è principalmente dovuta al costo del ciclo di minimizzazione pesata della norma l_1. Utilizzando la strategia di splitting forward-backward, si ha infatti che il costo computazionale è dato da: $CC = outit$ x $init$ x $O(MN)$; dove $init$ e $outit$ descrivono rispettivamente il numero di iterazioni interne ed esterne. La quantità di passi interni influenza, dunque, in maniera decisiva l'efficienza dell'algoritmo, ma, in combinazione con l'approccio di penalizzazione, costituisce al contempo il suo punto di forza, giustificandone le migliori prestazioni, rispetto ai metodi iterativi pesati classici, come i già citati IRLS e IRl_1. Infatti, l'uso della minimizzazione pesata della norma l_1, come nucleo centrale del metodo di penalizzazione, permette di scegliere il livello di precisione richiesto dal metodo di splitting prossimale, sulla base del valore del parametro di penalizzazione λ_k. Il vantaggio che si ottiene, attraverso l'uso di questa strategia adattiva, sembra essere ambivalente: da una parte, essa causa una riduzione del numero di iterazioni interne, quando la soluzione intermedia è lontana da quella ottima - ad esempio, per i primi valori di λ_k -; dall'altra, permette di evitare che il metodo si blocchi in un minimo locale sub-ottimo.
In conclusione, si torna a sottolineare che lo schema dell'algoritmo WNFCS è stato lasciato intenzionalmente molto generale, in modo da incoraggiare l'introduzione di eventuali migliorie e/o vincoli, che potrebbero rivelarsi necessari al variare dell'ambito applicativo[2]; ad esempio, in alcuni casi potrebbe

risultare indispensabile introdurre il cosiddetto *vincolo di positività*, oppure potrebbe essere conveniente sfruttare alcune tecniche di regolazione dei parametri delle funzioni sparsificanti, come quello mostrato nella sezione seguente.

3.3 Regolazione dei parametri della funzione sparsificante

Dal momento che, come si è già accennato precedentemente, la funzione sparsificante non convessa *log-exp* approssima tanto più fedelmente la norma l_0 quanto più il parametro μ tende a zero, è stato introdotto nell'algoritmo WNFCS un ulteriore miglioramento che si basa proprio su questa considerazione. Infatti, è stato inserito nel ciclo più esterno, subito prima dell'aggiornamento dei pesi, un'opportuna regolazione del parametro μ, in modo tale che, con l'approssimarsi della soluzione intermedia alla soluzione reale ottima, la funzione che induce la sparsità si avvicini sempre più al suo limite, costituito dalla norma l_0. L'algoritmo comincia dunque la sua computazione utilizzando un valore di μ relativamente alto per poi decrementarlo lentamente, col procedere dell'esecuzione, in modo da ridurre la penalizzazione di quei coefficienti che risultano già particolarmente elevati in quanto la loro ricostruzione è molto vicina a quella esatta - e dei quali, quindi, non si deve più incoraggiare l'annullamento.
La riduzione iterativa del parametro μ, anche se al momento non risulta ancora supportata da una giustificazione teorica, migliora le prestazioni dell'algoritmo WNFCS, perché agisce come fattore di perturbazione, evitando che il processo di risoluzione si blocchi sempre nell'intorno di uno stesso minimo locale. Chiaramente, questa riduzione del parametro μ viene eseguita in maniera tale da mantenere invariata la proprietà discendente dell'algoritmo. Un approccio di riduzione simile può essere apportato anche sui parametri delle altre funzioni sparsificanti, riportate in Tabella 3.1, ottenendo un comportamento analogo a quello descritto per la funzione *log-exp*.

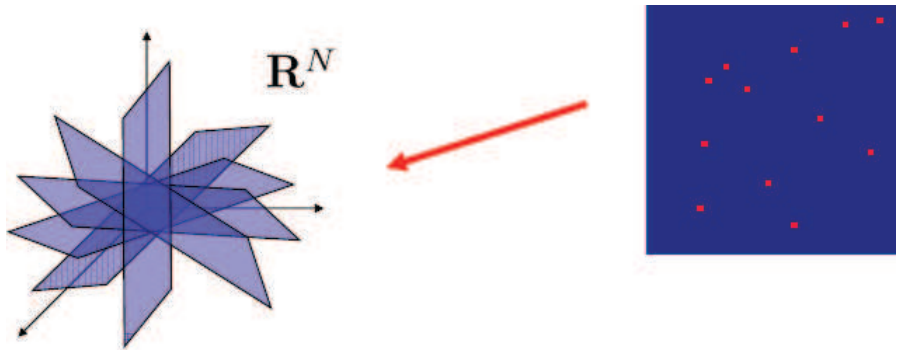


Figura 3.1: Allineamento dei sottospazi di dimensione K che compongono l'insieme Σ_K.

4 Model-based compressed sensing

Come abbiamo visto nel capitolo precedente, nella teoria classica del compressed sensing l'unico vincolo imposto sul vettore soluzione x è che esso sia K-sparso. Recentemente, tuttavia, sono stati presentati in letteratura nuovi approcci di ricostruzione che, sfruttando l'introduzione di opportuni modelli di sparsità, sono in grado di ricavare la soluzione x a partire da un numero di misurazioni lineari ancora inferiore. Infatti, l'uso di questi modelli riduce i gradi di libertà di un segnale sparso, vincolando i coefficienti di grande entità a disporsi soltanto secondo particolari configurazioni.
Queste nuove tecniche si basano sulla considerazione che un vettore x, che rappresenta un segnale K-sparso, risiede in un insieme Σ_K (Figura 3.1), costituito dall'unione di tutti gli $\binom{N}{K}$ sottospazi di dimensione K, allineati fra loro sugli assi delle coordinate in $\mathbb{R}^N$. In questo contesto, un *modello di sparsità strutturata* fornirà dunque al segnale K-sparso x una determinata conformazione, costringendolo a risiedere soltanto in un sottoinsieme di sottospazi dell'insieme Σ_K.
Sia dunque $x|_\Omega$ la rappresentazione degli elementi di x corrispondenti all'insieme di indici $\Omega \subseteq \{1, ..., N\}$ e sia Ω^C l'insieme complementare di Ω; è allora possibile definire formalmente il concetto di *modello di sparsità strutturata* M_K come l'unione di m_K sottospazi canonici di dimensione K, nel seguente modo:

$$M_K = \bigcup_{m=1}^{m_K} \chi_m \;\; s.t. \;\; \chi_m = \{x : x|_{\Omega_m} \in \mathbb{R}^K, x|_{\Omega_m^C} = 0\} \tag{3.12}$$

dove $\{\Omega_1, ..., \Omega_{m_K}\}$ rappresenta l'insieme di tutti i supporti ammissibili, con $|\Omega_m| = K$ per ogni $m = 1, ..., m_K$, e dove ogni sottospazio χ_m contiene tutti i segnali x tali che $supp(x) \subseteq \Omega_m$.
I segnali appartenenti all'insieme M_K sono detti *segnali sparsi K-strutturati*. Chiaramente, si avrà che $M_K \subseteq \Sigma_K$ sarà composto da $m_K \leq \binom{N}{K}$ sottospazi.

Nelle prossima sezione si mostrerà come introdurre questo nuovo approccio nell'infrastruttura algoritmica WNFCS.

4.1 Model-based WNFCS

Come si è visto nel precedente paragrafo, data la sua grande generalità, il framework algoritmico WNFCS può essere utilizzato per la ricostruzione di un segnale K-sparso, sostituendo alla classica minimizzazione l_1 una qualsiasi altra funzione obiettivo sparsificante. Sfruttando la teoria del model-based compressed sensing, risulterà quindi possibile introdurre nel processo di risoluzione opportuni modelli di sparsità, modificando in maniera idonea la funzione sparsificante. In particolare, il modello utilizzato in questa tesi cercherà di ricostruire la soluzione valutando invece dell'apporto dei singoli coefficienti, come avveniva in precedenza, il contributo di tutti gli intorni di dimensione $2d + 1$ di ogni coefficiente, dove $d \in \mathbb{N}$ dovrà essere opportunamente selezionato in base al contesto applicativo.
Verranno dunque considerate funzioni sparsificanti separabili, che possano essere rappresentate nel seguente modo:

$$F(u) = \sum_{i=1}^{N} \psi_i(|u_{i+h}|,\ per\ h = -d, ..., d) \tag{3.13}$$

Scegliendo come punto di riferimento l'applicazione *log-exp* descritta nella (3.2), sarà possibile ridefinire ogni $\psi_i(|u_{i+h}|,\ per\ h = -d, ..., d)$, come:

$$\psi_i(|u_{i+h}|,\ per\ h = -d, ..., d) = \frac{1}{\log(2)} \log\left(\frac{2}{1 + e^{-\frac{\sum_{h=-d}^{d} |u_{i+h}|}{\mu}}}\right) \tag{3.14}$$

Sulla base di queste considerazioni, al passo 7 dell'algoritmo in Tabella 3.2, sarà necessario calcolare i nuovi pesi w_i^{k+1} come la derivata parziale della

funzione $F(u)$ rispetto alla direzione $|u_i|$, ottenendo la seguente equazione:

$$w_i^{k+1} = \left.\frac{\partial F(u)}{\partial |u_i|}\right|_{u=u^k} = \left.\frac{\partial}{\partial |u_i|} \sum_{l=i-d}^{i+d} \psi_l(|u_{l+h}|,\ per\ h = -d, ..., d)\right|_{u=u^k} \tag{3.15}$$

Da cui, portando fuori la sommatoria si otterrà:

$$w_i^{k+1} = \left.\frac{\partial F(u)}{\partial |u_i|}\right|_{u=u^k} = \sum_{l=i-d}^{i+d} \frac{\partial}{\partial |u_i|} \psi_l(|u_{l+h}^k|,\ per\ h = -d, ..., d) =$$

$$= \sum_{l=i-d}^{i+d} \frac{1}{\mu \log(2)} \left(\frac{1}{1 + e^{\frac{\sum_{h=-d}^{d} |u_{l+h}^k|}{\mu}}} \right)$$

Nel prossimo capitolo verranno mostrati i risultati ottenuti, nell'ambito della face recognition, utilizzando l'algoritmo WNFCS in entrambe le sue varianti. In particolare, si eseguirà un confronto fra le prestazioni di questo nuovo framework e quelle ottenute da un metodo classico di programmazione lineare primale-duale di punto interno, implementato nella libreria SparseLab [45].

Capitolo 4

Sperimentazione e analisi dei risultati

Nel capitolo precedente è stato presentato l'algoritmo WNFCS, dimostrando come esso consenta di risolvere il problema del compressed sensing sfruttando funzioni obiettivo non convesse e permettendo anche l'introduzione di opportuni modelli di sparsità.
In questo capitolo saranno invece riportati ed analizzati i risultati prodotti dai diversi test eseguiti, che hanno coinvolto, fra tutti quelli possibili, due database pubblici di volti; l'Extended Yale B database [42] e l'AR database [44]. La scelta è ricaduta proprio su questi archivi, in quanto essi costituiscono i principali punti di riferimento anche per i test riportati in letteratura e, allo stesso tempo, risultano liberamente scaricabili dalla rete.

Questo capitolo sarà suddiviso in due parti principali:

- in primo luogo saranno mostrate le percentuali nette di riconoscimento raggiunte, assumendo che tutti i volti del test set appartengano a individui conosciuti;
- successivamente, verrà focalizzata l'attenzione sul processo di validazione, eliminando la precedente assunzione e sottoponendo ai diversi classificatori anche immagini di test relative a soggetti non rappresentati nel training set.

In entrambe queste categorie di test verranno messi a confronto i risultati ottenuti dal framework SRC nella sua modellazione classica, con quelli

ottenuti sostituendo all'algoritmo di minimizzazione l_1 l'infrastruttura algoritmica WNFCS, descritta nel Capitolo 3; inoltre, verranno forniti anche i risultati raggiunti dai metodi tradizionali *nearest neighbour* e *nearest subspace*.
Prima di procedere in questo senso, però, verrà introdotto un ulteriore paragrafo, al fine di evidenziare quali valori sono stati scelti per i parametri liberi dell'algoritmo WNFCS.

1 Parametrizzazione dell'algoritmo WNFCS

Dalla descrizione schematica dell'algoritmo WNFCS, riportata nel terzo capitolo, è possibile notare che la sua inizializzazione si basa sull'opportuna impostazione di determinati parametri liberi. Una parte consistente del lavoro di sperimentazione, effettuato in questa tesi, è quindi consistita proprio nella ricerca di una configurazione di questi valori che permettesse di ottenere una percentuale di successi il più elevata possibile, in termini di riconoscimento di volti. In effetti, è emerso sperimentalmente che, in questo contesto, una soluzione più sparsa non conduce in assoluto ad una classificazione più corretta, in quanto un'errata parametrizzazione dell'algoritmo potrebbe condurre al raggiungimento di minimi locali, ricostruiti utilizzando immagini di training appartenenti al soggetto sbagliato. Dunque, verranno sotto riportati i valori di parametrizzazione dell'algoritmo WNFCS determinati empiricamente e successivamente utilizzati nei test di riconoscimento.
A seguito della lunga sperimentazione, è emerso che il valore iniziale più adatto per il parametro di penalizzazione risulta essere $\lambda_0 = 0,8 \cdot \left\|\Phi^T y\right\|_\infty$ e che il miglior compromesso per il relativo fattore di riduzione consiste in $r_\lambda = 0,5$. Un altro valore importante è senz'altro quello del cosiddetto *parametro di rilassamento* β, il quale, è stato inizializzato con $\beta = 0,7 \cdot \frac{2}{|\sigma|}$, dove σ descrive l'autovalore di modulo massimo della matrice $\Phi^T\Phi$.
Con l'utilizzo della funzione non convessa *log-exp*, si è rivelato inoltre necessario definire un valore iniziale appropriato per il parametro μ, impostandolo come $\mu = 0,8 \cdot \max(|\Phi^T y|)$, e successivamente, sulla base di quanto affermato nel paragrafo 3.3, si è deciso di introdurre un opportuno criterio di riduzione

per μ, fissando $r_\mu = 0,8$.
Il parametro più influente nel determinare la velocità di convergenza dell'algoritmo WNFCS, tuttavia, è senz'altro γ; esso, infatti, rappresenta lo strumento di aggiustamento adattivo della precisione, per l'approccio iterativo di splitting. Le prove sperimentali hanno permesso di verificare che, nel contesto del riconoscimento di volti, il valore di γ più appropriato risulta essere $0,5$.
Infine, rimane da chiarire la scelta dell'ultimo parametro libero, consistente nel criterio di arresto. In particolare, si è deciso di utilizzare a questo scopo un approccio ibrido, basato principalmente sull'utilizzo di un valore di tolleranza sulla qualità della ricostruzione, affiancato, però, dall'imposizione di un numero massimo di iterazioni. Questo secondo vincolo è stato introdotto per evitare la levitazione del tempo di esecuzione in presenza di casi difficili. I valori scelti per il parametro di tolleranza e per il numero massimo di iterazioni sono rispettivamente $5 \cdot 10^{-3}$ e 3000.
A partire dal prossimo paragrafo, verranno finalmente mostrati i risultati così ottenuti.

2 Test eseguiti senza validazione

Nei primi capitoli si è parlato del processo di feature extraction e di quanto esso risulti fondamentale per i metodi classici di riconoscimento dei volti. Tuttavia, è stato anche dimostrato che il nuovo framework basato sulla rappresentazione sparsa risulta in qualche modo meno subordinato a questa fase del processo di face recognition. I test riportati in questa sezione tenteranno dunque innanzitutto di evidenziare questo fenomeno, lasciando per il momento in disparte l'utilizzo di una strategia di validazione e sottoponendo a tutti i classificatori, presentati nelle sezioni precedenti, scenari di riconoscimento che implichino l'utilizzo delle principali tecniche di estrazione delle feature discusse innanzi. In particolare, sono stati presi in considerazione metodi di feature extraction tradizionali - come Eigenfaces, Fisherfaces e Laplacianfaces - e altri metodi meno usuali - come il downsampling e Randomfaces. In seguito, si presenteranno invece alcuni test effettuati in presenza di occlusioni.

In tutti gli esperimenti è stata utilizzata la versione robusta del framework SRC in quanto, come precedentemente discusso, questa permette di gestire al meglio eventuali perturbazioni esistenti nelle immagini di test, siano esse dovute a presenza di rumore, a condizioni di illuminazione esasperate, o a occlusioni di diversa entità.
Si sottolinea, infine, che tutti i test sono stati effettuati su una macchina con processore Intel Core i7 Q720 a 1.60 Ghz, con 8 GB di RAM.

2.1 Extended Yale B database

L'Extended Yale B database è composto da immagini frontali di 38 individui, per ognuno dei quali sono state raccolte circa 64 immagini - alcune hanno subìto dei disturbi in fase di acquisizione e sono state quindi scartate. Questo database contiene dunque un totale di 2414 volti. La caratteristica fondamentale delle immagini racchiuse in questo archivio è quella di essere state ottenute sotto diverse condizioni di illuminazione, alcune delle quali risultano così estreme da rendere molto difficile il riconoscimento del volto anche da parte di un essere umano. Fra le diverse versioni di questo archivio disponibili in rete, è stata scelta quella in cui le immagini risultano preventivamente segmentate e allineate alla dimensione di 192x168 pixel [43].
I risultati riportati in seguito sono stati ottenuti costruendo i database di training e di test nel seguente modo: per ogni individuo, sono state selezionate casualmente 32 immagini da inserire nel training set, lasciando le rimanenti a costituzione del test set. Il database di training conterrà dunque 1216 immagini, mentre quello di test ne conterrà 1198.
Per la dimensione dello spazio delle feature sono stati scelti, in conformità coi test riportati in letteratura, i seguenti fattori di ridimensionamento: 1/8, 1/16, 1/24 e un 1/32; corrispondenti rispettivamente alle seguenti dimensioni: 504, 120, 56 e 30. Dal momento che il metodo Fisherface impone che la dimensione dello spazio delle feature non possa superare il numero di classi meno uno - in questo caso, $38 - 1 = 37$ - l'unico fattore di downsampling utilizzabile con questa tecnica sarà 1/32.
Detto questo, è finalmente possibile riportare i risultati ottenuti.

Eigenfaces				
Dimensione	30	56	120	504
SRC + Model-based WNFCS	87,56%	**92,99%**	94,74%	96,66%
SRC + WNFCS	**88,40%**	92,82%	**95,16%**	**96,83%**
SRC + l_1 min	86,14%	92,49%	94,82%	96,49%
NN	49,83%	60,52%	69,70%	73,04%
NS	74,21%	82,22%	86,23%	88,15%

Fisherfaces	
Dimensione	30
SRC + Model-based WNFCS	79,80%
SRC + WNFCS	80,05%
SRC + l_1 min	**80,30%**
NN	77,30%
NS	76,79%

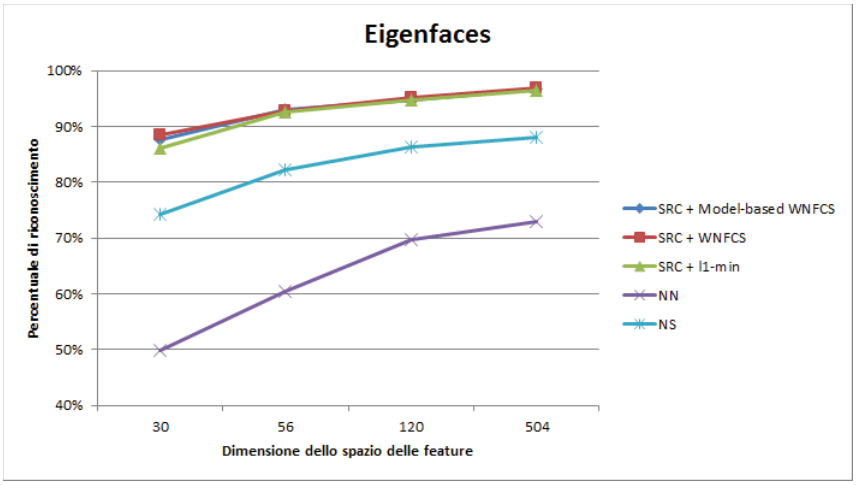
Laplacianfaces				
Dimensione	30	56	120	504
SRC + Model-based WNFCS	**86,98%**	93,32%	95,99%	96,08%
SRC + WNFCS	86,31%	**93,41%**	**96,33%**	**96,41%**
SRC + l_1 min	86,81%	93,24%	95,91%	95,99%
NN	80,13%	86,64%	89,90%	89,57%
NS	70,70%	86,39%	92,99%	93,82%

Downsampling				
Dimensione	30	56	120	504
SRC + Model-based WNFCS	75,79%	87,40%	92,74%	**96,49%**
SRC + WNFCS	**76,63%**	**88,65%**	92,82%	95,99%
SRC + l_1 min	74,54%	88,48%	**93,07%**	95,66%
NN	48,66%	61,44%	69,20%	72,95%
NS	61,69%	72,04%	83,64%	87,40%

Randomfaces				
Dimensione	30	56	120	504
SRC + Model-based WNFCS	85,23%	90,57%	93,57%	**96,99%**
SRC + WNFCS	**85,48%**	**90,73%**	**94,24%**	96,91%
SRC + l_1 min	81,89%	88,73%	94,16%	96,41%
NN	53,92%	64,27%	69,45%	71,70%
NS	71,70%	75,21%	78,63%	86,89%

Tabella 4.1: Risultati ottenuti dai diversi classificatori sull'Extended Yale B database, al variare del metodo di estrazione delle feature.

Il primo dato a risultare evidente, esaminando i risultati mostrati in Tabella 4.1, consiste nel predominio del classificatore SRC sui metodi tradizionali nearest neighbour e nearest subspace. Le diverse varianti di questo framework, d'altra parte, ottengono in questo scenario risultati molto simili fra loro, sebbene il maggior numero di *vittorie* sia stato raggiunto dalla versione che utilizza, al posto della minimizzazione l_1, lo schema base dell'algoritmo WNFCS. La variante che sfrutta la sparsità strutturata si rivela comunque competitiva e, come sarà evidenziato da alcuni test successivi, risulterà largamente vincente in presenza di occlusioni.
La superiorità del metodo SRC rispetto agli altri classificatori può essere meglio apprezzata esaminando i grafici seguenti.



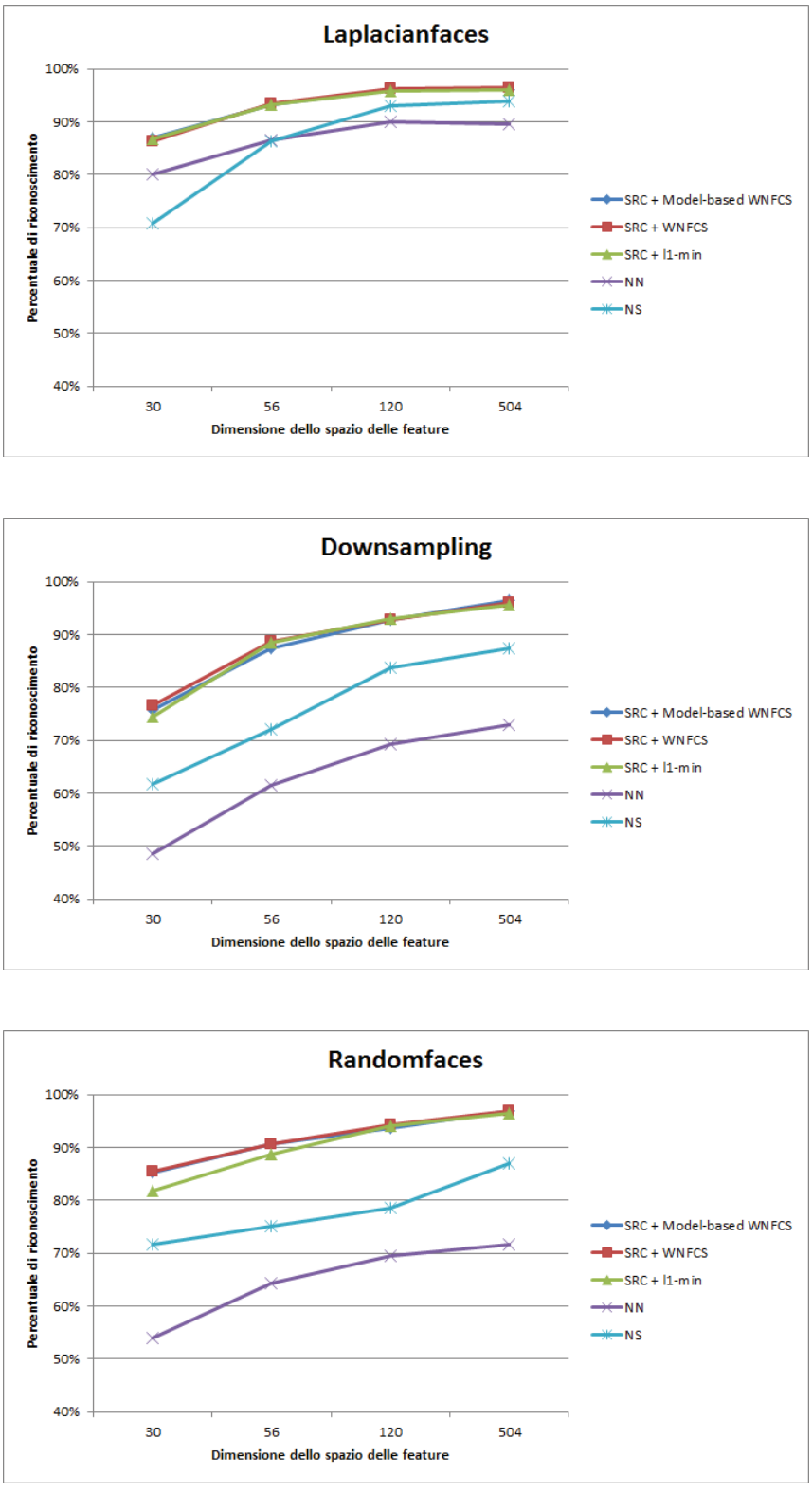


Figura 4.1: Grafici dei risultati ottenuti dai diversi classificatori sull'Extended Yale B database, al variare del metodo di estrazione delle feature.

Il secondo aspetto interessante che emerge dai risultati sopra riportati consiste nella dimostrazione di quanto affermato nel secondo capitolo, ovvero la quasi totale indipendenza dell'algoritmo SRC dalle varie tecniche di feature extraction; infatti, questo framework ottiene, in tutte le sue varianti, ottimi risultati, sia utilizzando i metodi tradizionali, sia utilizzando quelli meno convenzionali. La tabella e il grafico sotto riportati offrono un riepilogo delle percentuali prodotte dai test, atto a chiarificare meglio questo fenomeno. A questo scopo è stata scelta, come punto di riferimento, la versione dell'SRC che utilizza l'algoritmo WNFCS nella sua variante base.

SRC + WNFCS				
Dimensione	30	56	120	504
Eigenfaces	**88,40%**	92,82%	95,16%	96,83%
Fisherfaces	80,05%	-	-	-
Laplacianfaces	86,31%	**93,41%**	**96,33%**	96,41%
Downsampling	76,63%	88,65%	92,82%	95,99%
Randomfaces	85,48%	90,73%	94,24%	**96,91%**

Tabella 4.2: Riepilogo dei risultati ottenuti sull'Extended Yale B.

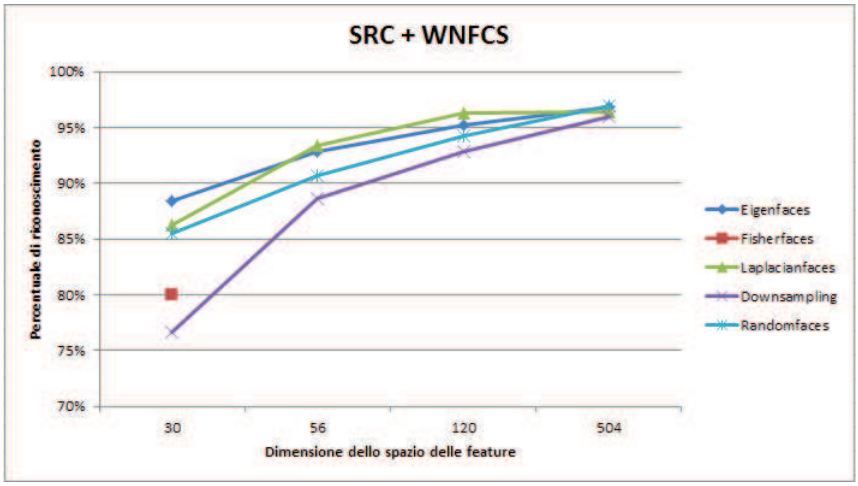


Figura 4.2: Grafici dei risultati ottenuti sull'Extended Yale B database dal classificatore SRC + WNFCS.

Analizzando il riepilogo in Tabella 4.2 è altresì possibile notare come le percentuali di riconoscimento aumentino al crescere della dimensione dello spazio delle feature. Nella scelta di questo parametro incideranno dunque due

fattori fondamentali: il dominio applicativo e l'infrastruttura hardware a disposizione. Infatti, nel secondo capitolo si è visto che le informazioni di località spaziale, utili per l'isolamento di perturbazioni e occlusioni, vengono tanto più preservate quanto più grande risulta essere lo spazio delle feature; d'altra parte, è stato anche mostrato che il processo di riconoscimento, nel contesto del framework SRC, è basato sulla risoluzione di un sistema lineare sottodeterminato, la cui corrispondente matrice avrà una dimensione tanto maggiore quanto più grande sarà il numero delle immagini che la compongono, oltreché, appunto, la dimensione delle rispettive feature.
Risulterà dunque evidente che all'utilizzo di un spazio delle feature di dimensioni più elevate corrisponderà, inevitabilmente, un aggravio del costo computazionale.
Nella tabella seguente, vengono messi a confronto i tempi medi, per il riconoscimento di una singola immagine, richiesti dai classificatori utilizzati nei test; come si può notare il metodo SRC che impiega al suo interno l'algoritmo WNFCS, purtroppo, risulta sempre più lento rispetto al suo diretto concorrente (la variante che usa la minimizzazione della norma l_1); i tempi impiegati rimangono comunque compresi all'interno di un range accettabile e tale da giustificare in ogni caso l'utilizzo di questo nuovo algoritmo, visti anche i migliori risultati da esso raggiunti. Inoltre, questo test è stato eseguito considerando uno scenario piuttosto controllato, in quanto l'unico fattore di difficoltà considerato consiste nella variazione di illuminazione; d'altra parte, verrà presto mostrato che, aggiungendo ulteriori problematiche - come l'occlusione o la necessità di validazione - i nuovi metodi presentati raggiungeranno percentuali di successo spesso di gran lunga superiori a quelle della minimizzazione l_1, sminuendo ulteriormente l'importanza del fattore tempo.

Tempi medi di esecuzione per singola immagine				
Dimensione	30	56	120	504
SRC + Model-based WNFCS	1,151 s.	1,364 s.	1,575 s.	3,418 s.
SRC + WNFCS	0,655 s.	0,916 s.	1,151 s.	2,515 s.
SRC + l_1 min	0,065 s.	0,092 s.	0,173 s.	1,415 s.
NN	0,001 s.	0,001 s.	0,003 s.	0,015 s.
NS	0,033 s.	0,042 s.	0,056 s.	0,163 s.

Tabella 4.3: Tempi richiesti dai diversi classificatori sull'Extended Yale B.

2.2 AR database

L'AR database è composto da immagini frontali di 126 individui, per ognuno dei quali sono state raccolte circa 26 immagini. Questo database contiene dunque un totale di circa 3276 volti. Le 26 immagini relative ad ogni soggetto sono state acquisite in due diverse sessioni - 13 per ognuna - e dunque presentano fattori di variabilità intra-classe. Delle 13 immagini catturate durante una particolare sessione, 7 rappresentano diverse espressioni del volto e diverse condizioni di illuminazione, mentre le restanti 6 sono state scattate aggiungendo dei fattori di occlusione realistici, facendo indossare ai soggetti rappresentati degli occhiali da sole o una sciarpa.
In conformità coi test presentati in letteratura, i risultati riportati in seguito sono stati ottenuti considerando fra i 126 individui totali, soltanto 50 uomini e 50 donne. Per ognuno di essi, sono state inserite nel training set le relative 7 immagini della prima sessione che non presentano occlusioni; mentre le 7 immmagini senza occlusione, acquisite durante la seconda sessione, sono state selezionate come immagini di test (vedi Figura 4.3). Dunque, entrambi i database saranno composti da 700 volti. Fra le diverse versioni esistenti di questo database, è stata scelta quella in cui le immagini sono già state convertite in scala di grigi e segmentate alla dimensione di 165x120 pixel. Purtroppo, però, questa versione presenta forti problemi di allineamento fra le varie immagini dei volti e questo fattore, assieme al ridotto numero di rappresentazioni nel training set per individuo - soltanto 7, contro le 32 dell'Extended Yale B database - conduce ad ottenere risultati considerevolmente inferiori rispetto a quelli già presentati nella sezione precedente. Inoltre, la

maggior variabilità esibita dall'AR database, costituisce un ulteriore fattore di difficoltà.

Figura 4.3: Esempio di immagini per un soggetto dell'AR database. Nella prima fila si trovano le immagini senza occlusione acquisite nella prima sessione, mentre sotto sono riportate quelle relative alla seconda.

Prima di procedere presentando i risultati ottenuti, è necessario specificare che i fattori di ridimensionamento utilizzati per l'estrazione delle feature sono 1/6, 1/12, 1/18 e 1/24, relativi rispettivamente ai sottospazi di grandezza 540, 130, 54 e 30.
Inoltre, a causa del ridotto numero di immagini per soggetto, il metodo nearest subspace si è rivelato particolarmente inefficace e i suoi risultati sono stati dunque ignorati.

Eigenfaces				
Dimensione	30	54	130	540
SRC + Model-based WNFCS	**62,00%**	**72,00%**	**78,86%**	81,14%
SRC + WNFCS	59,00%	71,14%	78,00%	**82,43%**
SRC + l_1 min	55,29%	64,29%	72,43%	77,86%
NN	54,14%	57,43%	59,43%	58,86%

Downsampling				
Dimensione	30	54	130	540
SRC + Model-based WNFCS	41,29%	54,71%	**72,86%**	83,29%
SRC + WNFCS	**44,00%**	**55,00%**	72,43%	**83,71%**
SRC + l_1 min	35,14%	48,71%	68,29%	81,57%
NN	34,43%	41,86%	52,14%	58,71%

Randomfaces				
Dimensione	30	54	130	540
SRC + Model-based WNFCS	**40,86%**	**50,86%**	**70,00%**	**80,57%**
SRC + WNFCS	39,29%	49,43%	68,14%	79,86%
SRC + l_1 min	35,57%	42,29%	64,86%	77,00%
NN	33,23%	41,00%	58,00%	57,71%

Tabella 4.4: Risultati ottenuti dai diversi classificatori sull'AR database.

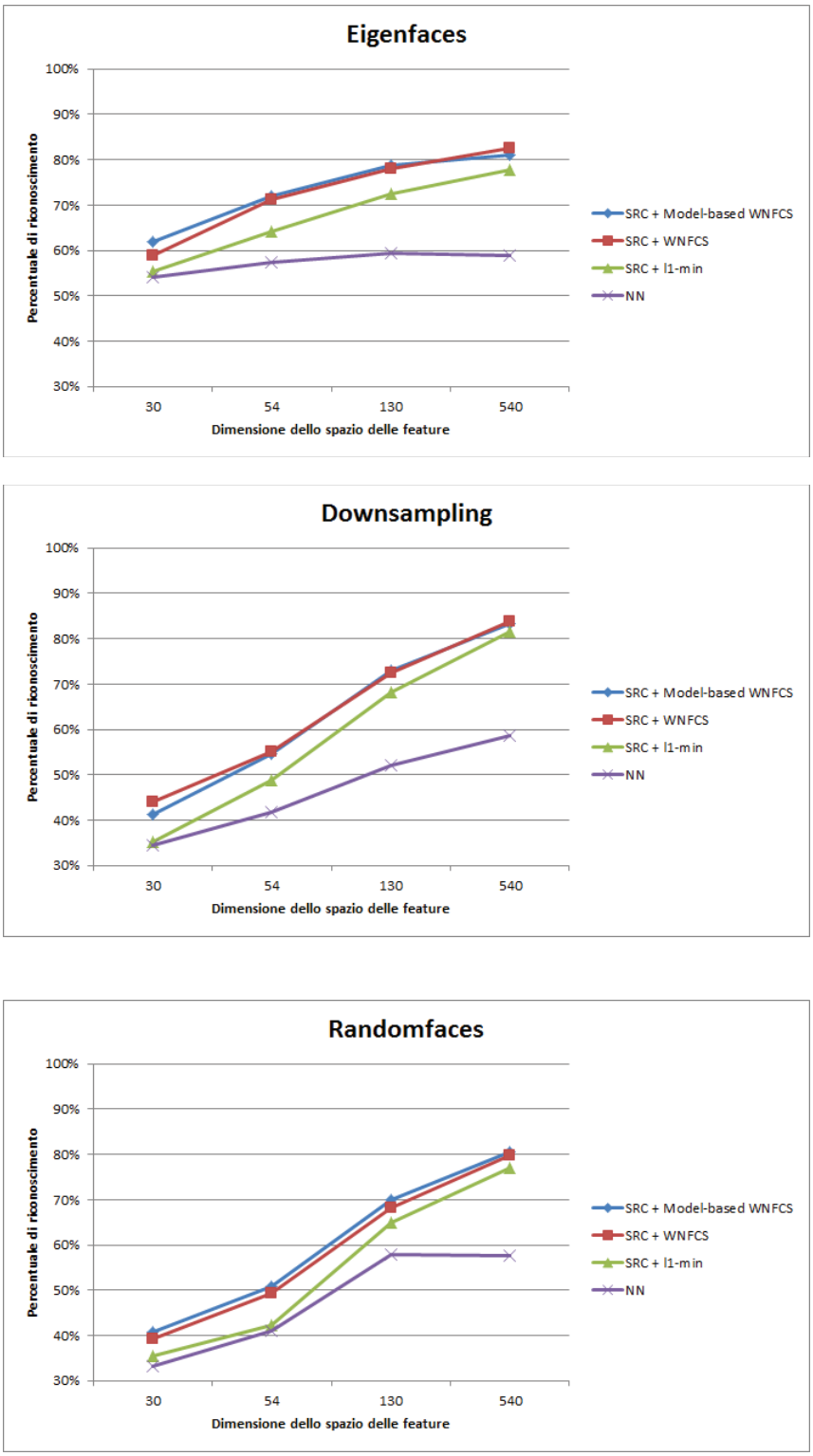


Figura 4.4: Grafici dei risultati ottenuti dai diversi classificatori sull'AR database, al variare del metodo di estrazione delle feature.

I risultati sopra presentati, sebbene inferiori a quelli ottenuti sull'Extended Yale B database, confermano comunque quanto già precedentemente concluso; infatti, anche in questo caso il framework SRC risulta sempre superiore al metodo tradizionale nearest neighbour. Inoltre, in questo scenario il maggior numero di successi è stato raggiunto, quasi sempre, utilizzando la versione model-based del WNFCS, la quale si è rivelata migliore del metodo di minimizzazione l_1 in ogni situazione, primeggiando in alcuni casi per una percentuale superiore al 7%. Il divario fra l'algoritmo WNFCS (in entrambe le sue versioni) e il metodo di risoluzione classico si manifesta soprattutto quando lo spazio delle feature è di piccole dimensioni, in perfetta armonia con le fondamenta teoriche del compressed sensing.
In questo scenario, dunque, l'impiego della nuova infrastruttura algoritmica WNFCS risulta ulteriormente giustificato, nonostante i suoi tempi di esecuzione continuino a rivelarsi maggiori - come riportato in Tabella 4.5.

Tempi medi di esecuzione per singola immagine				
Dimensione	30	54	130	540
SRC + Model-based WNFCS	0,708 s,	0,771 s,	0,823 s,	2,479 s,
SRC + WNFCS	0,387 s,	0,542 s,	0,658 s,	2,366 s,
SRC + l_1 min	0,052 s,	0,063 s,	0,111 s,	1,218 s,
NN	0,001 s,	0,001 s,	0,001 s,	0,010 s,

Tabella 4.5: Tempi richiesti dai diversi classificatori sull'AR database.

AR Database ed espressioni facciali

Come si è detto, per l'esecuzione dei test sull'AR database, descritti nella sezione precedente, sono state inserite nel training set anche le immagini relative alle espressioni più intense. Tuttavia, queste rappresentazioni non fanno altro che aumentare la variabilità intra-classe, diminuendo contemporaneamente quella inter-classe e rendendo dunque ancora più difficile il processo di riconoscimento. In questo paragrafo saranno invece mostrati i risultati ottenuti, creando i database di training e di test in maniera differente, con l'obiettivo di ovviare al fenomeno appena descritto. In particolare, sono state inserite nel training set le quattro immagini di ogni sessione in

cui gli individui sono rappresentati con un'espressione neutra, per un totale di 8 immagini a persona, mentre il test set è stato composto utilizzando le rimanenti tre immagini per sessione in cui i soggetti hanno espressioni più accentuate, per un totale di 6 immagini. Il database di training e quello di test conterranno dunque rispettivamente 800 e 600 immagini.
Questo esperimento è stato eseguito utilizzando come metodo di estrazione delle feature il semplice downsampling con fattore 1/6. Di seguito sono forniti i risultati raggiunti.

Test espressioni facciali	
SRC + Model-based WNFCS	**87,83%**
SRC + WNFCS	87,00%
SRC + l_1 min	86,00%
NN	75,83%

Tabella 4.6: Risultati ottenuti sull'AR database, inserendo nel training set soltanto immagini con espressioni neutre.

Confrontando le percentuali riportate in Tabella 4.6 con quelle descritte nel paragrafo precedente, risulta immediatamente visibile il maggior numero di successi ottenuto attraverso questa nuova configurazione di training e test. Inoltre, anche in questo esperimento, la versione del framework SRC a rivelarsi vincente è quella che fa uso dell'algoritmo WNFCS e della sparsità strutturata.

2.3 Robustezza nei confronti delle occlusioni

Nei primi capitoli si è visto che una delle problematiche principali, che devono essere affrontate da un sistema di face recognition robusto, consiste nella possibile presenza di occlusioni. In questi casi, l'immagine del volto da riconoscere risulta infatti parzialmente coperta da effetti di perturbazione o da travestimenti, tali da rendere molto complicato il processo di identificazione anche per il cervello umano. Tuttavia, è stato anche mostrato come il metodo SRC risulti perfettamente in grado di gestire scenari di questo genere.

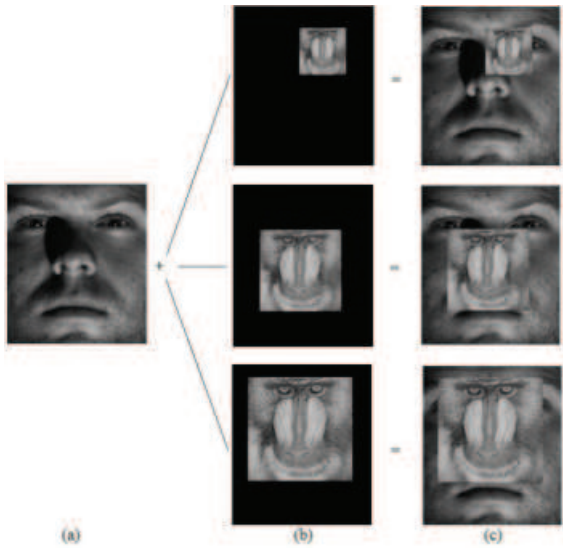


Figura 4.5: Esempi di random block occlusion. (a) Immagine di test originale. (b) Random block occlusion. Percentuale di occlusione dall'alto verso il basso: 10%, 30% e 50%. (c) Immagini occluse sottoposte al sistema.

In questa sezione saranno dunque riportati alcuni test eseguiti sia sull'Extended Yale B che sull'AR database, che metteranno ancora una volta a confronto le prestazioni dell'infrastruttura algoritmica WNFCS con quelle della classica minimizzazione l_1. Per completezza si forniranno inoltre anche i risultati ottenuti dai metodi tradizionali nearest neighbour e nearest subspace.

Extended Yale B e random block occlusion

Dal momento che l'Extended Yale B database non mette a disposizione immagini di volti con occlusioni realistiche, al fine di testare le performance dei diversi algoritmi con questo archivio, si è rivelato indispensabile introdurre un tipo di occlusione sintetica. In particolare, il disturbo che si è scelto di considerare - in maniera coerente con quanto riportato in letteratura - consiste nel sovrapporre all'immagine del volto un'altra immagine scelta casualmente e di cui il sistema di face recognition non deve avere alcuna informazione. La zona di occlusione deve consistere, inoltre, in una sottoporzione quadrata dell'immagine di test - anch'essa scelta in maniera casuale. In Figura 4.5, viene fornito un esempio di questo tipo di occlusione, che prende il nome di *random block occlusion*.

L'Extended Yale B database può essere suddiviso in cinque sottoinsiemi, determinati considerando le diverse condizioni di illuminazione; dunque, per questo test, il training set è stato costruito selezionando le rappresentazioni dei volti dei primi due sottoinsiemi, che contengono le immagini con condizioni di illuminazione più moderate, mentre il test set è stato composto soltanto dai volti del terzo sottoinsieme, che contiene, invece, immagini riprese in condizioni di illuminazione più critiche. Il database di training sarà quindi composto da 722 immagini, mentre quello di test ne conterrà 450.
Dal momento che, come si è detto nel secondo capitolo, in presenza di occlusioni è conveniente mantenere uno spazio delle feature di grandi dimensioni, i risultati riportati di seguito sono stati ottenuti utilizzando come metodo di feature extraction il downsampling con fattore di ridimensionamento 1/6, corrispondente ad una grandezza del sottospazio pari a 896.

Random block occlusion					
% Occlusione	SRC + Model-based WNFCS	SRC + WNFCS	SRC + l_1 min	NN	NS
0%	**100,00%**	**100,00%**	**100,00%**	77,78%	99,78%
10%	**100,00%**	**100,00%**	**100,00%**	80,00%	97,78%
20%	**99,78%**	**99,78%**	99,11%	77,11%	88,67%
30%	**98,67%**	97,78%	95,78%	67,33%	72,67%
40%	**93,11%**	89,78%	80,22%	56,67%	56,44%
50%	**76,22%**	68,67%	58,00%	41,78%	38,00%
60%	**46,67%**	38,89%	30,44%	29,78%	23,33%
70%	**23,33%**	20,44%	17,11%	20,67%	13,33%

Tabella 4.7: Risultati random block occlusion su Enxtended Yale B database.

Analizzando i risultati in Tabella 4.7, si nota in maniera evidente come l'utilizzo dell'algoritmo WNFCS, nella sua variante che considera la struttura della sparsità, si riveli sempre migliore dei suoi concorrenti; confermando la sua importante capacità nel ricostruire correttamente il segnale anche in presenza di una quantità inferiore di misurazioni lineari. In particolare, tramite questo metodo, è stato possibile raggiungere percentuali di riconoscimento superiori al 90% fino ad un fattore di occlusione pari al 40%; questo fenomeno rappresenta un significativo miglioramento rispetto al risultato ottenuto dal metodo di minimizzazione l_1, che, come si può vedere, in questo frangente ottiene una percentuale di successi di poco superiore all'80%.

La migliore efficacia dell'algoritmo WNFCS potrà essere meglio apprezzata, esaminando il grafico in Figura 4.6.

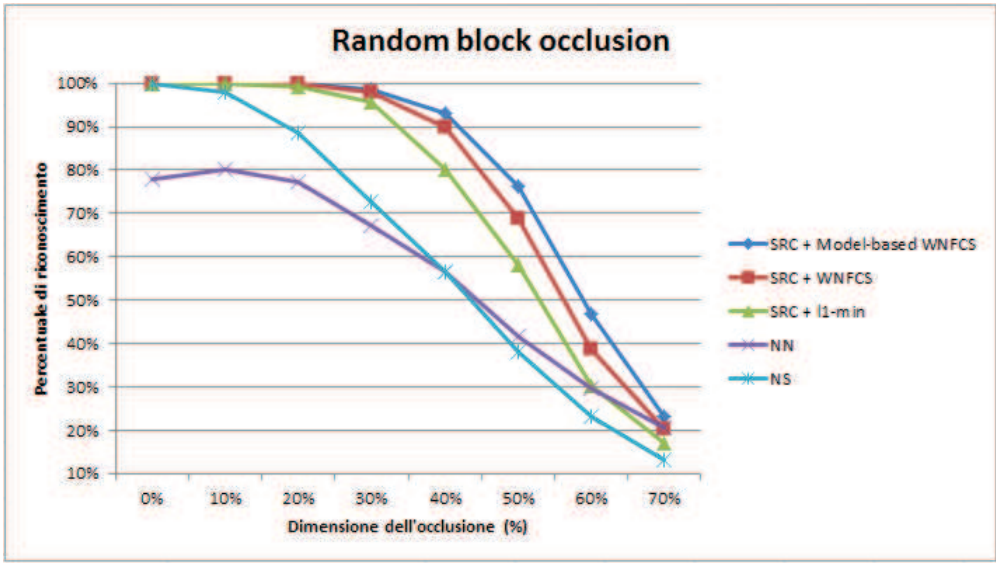


Figura 4.6: Grafico dei risultati ottenuti con random block occlusion su Enxtended Yale B database.

AR database e occlusione da occhiali da sole

Com'è stato già accennato nel paragrafo 2.2, alcune immagini dell'AR database sono state acquisite facendo indossare degli occhiali da sole agli individui rappresentati. In questa sezione verranno dunque mostrati i risultati ottenuti, sottoponendo ai diversi algoritmi di riconoscimento queste immagini di volti occlusi in maniera realistica.
Per l'esecuzione di questo esperimento, è stato fatto uso dello stesso training set già utilizzato per il test relativo alle espressioni, presentato nel paragrafo 2.2; mentre il test set è stato costruito selezionando soltanto le immagini con occhiali da sole, relative ai 100 soggetti considerati nel database di training. Il training set conterrà dunque 800 immagini, mentre il test set ne comprenderà 600.
I risultati presentati in Tabella 4.8 sono stati ottenuti, inoltre, estraendo feature di grandezza 1230 con il metodo dowsampling e fattore di ridimensionamento 1/4.

Test occlusione da occhiali da sole	
SRC + Model-based WNFCS	**75,17%**
SRC + WNFCS	74,67%
SRC + l_1 min	73,00%
NN	39,00%

Tabella 4.8: Risultati ottenuti sull'AR database, in presenza di occlusione da occhiali da sole.

Anche in questo scenario l'algoritmo WNFCS ottiene risultati migliori rispetto alla minimizzazione l_1, sebbene il divario risulti meno accentuato di quello emerso nel test riportato nella sezione precedente.
Le percentuali di riconoscimento sono, inoltre, generalmente più basse sia per i problemi di allineamento e variabilità che affliggono l'AR database, sia perchè l'occlusione da occhiali da sole costituisce un caso particolarmente arduo da gestire, in quanto gli occhi rappresentano un'area del volto tipicamente discriminativa.

3 Test eseguiti attivando il meccanismo di validazione

Nel secondo capitolo è stato evidenziato come il classificatore SRC, contrariamente a tutti gli altri metodi esistenti, possa basare i processi di validazione e riconoscimento su due informazioni statistiche differenti. In particolare, si è visto che, mentre l'operazione di classificazione viene eseguita sfruttando i residui, quella di validazione è basata sull'utilizzo di un nuovo indice che prende il nome di SCI e che può essere calcolato come mostrato nella formula (2.27). Si ricorda, inoltre, che il processo di validazione consiste nello stabilire se l'immagine di test, sottoposta al sistema, rappresenti o meno il volto di uno degli individui contenuti nel training set.
Dal momento che l'indice SCI risulta compreso nell'intervallo $[0, 1]$, attivare il meccanismo di validazione si tradurrà nello stabilire una valore di soglia τ che possa discriminare al meglio le immagini valide da quelle che non lo sono. Inoltre, dato che l'indice SCI fornisce una misura della qualità della

soluzione, la robustezza del meccanismo di validazione risulterà tanto più accentuata quanto più $\tau \longrightarrow 1$.
In un contesto di questo genere la percentuale di successi non sarà più calcolata esclusivamente sulla base dei volti correttamente classificati, ma dovrà tener conto anche delle immagini giustamente rigettate; d'altra parte, la percentuale di fallimenti comprenderà non soltanto i volti mal classificati, ma anche due nuovi fenomeni: il primo consiste nel rifiuto di un'immagine valida - misurato dall'indice FRR (*False Rejection Rate*) - mentre il secondo si concretizza con l'accettazione di un volto sconosciuto - misurato dall'indice FAR (*False Acceptance Rate*).
Per testare il comportamento delle diverse versioni dell'algoritmo SRC in presenza di questo nuovo meccanismo, sono stati eseguiti due esperimenti; in primo luogo si è valutato il numero di false reiezioni ottenute al variare della soglia sull'indice SCI, senza introdurre nel database di test alcun *impostore*; in seguito si sono invece valutate le prestazioni generali delle diverse varianti dell'algoritmo anche in presenza di immagini non valide.
Verranno ora mostrati i risultati ottenuti in questi nuovi scenari.

3.1 False reiezioni in assenza di impostori

Per eseguire questo esperimento, è stato scelto come database l'Extended Yale B, utilizzando come metodo di estrazione delle feature il semplice downsampling alla dimensione 504 e costruendo il training set e il test set come già avvenuto per le prove presentate in Sezione 2.1. Le tabelle seguenti rappresentano i risultati ottenuti per ogni metodo, al variare della soglia τ. La colonna con intestazione *Errori*, contiene le percentuali relative alle immagini accettate come valide, ma mal classificate.

Model-based WNFCS			
τ	Successi	Errori	FRR
0,1	96,49%	3,51%	0,00%
0,2	96,33%	2,92%	0,75%
0,3	94,32%	1,25%	4,42%
0,4	92,15%	0,58%	7,26%
0,5	88,40%	0,33%	11,27%
0,6	82,80%	0,17%	17,03%
0,7	75,13%	0,17%	24,71%
0,8	57,51%	0,17%	42,32%
0,9	33,89%	0,17%	65,94%

WNFCS			
τ	Successi	Errori	FRR
0,1	95,99%	4,01%	0,00%
0,2	94,57%	1,92%	3,51%
0,3	91,90%	0,58%	7,51%
0,4	87,73%	0,17%	12,10%
0,5	81,39%	0,17%	18,45%
0,6	73,71%	0,17%	26,13%
0,7	57,85%	0,08%	42,07%
0,8	38,06%	0,08%	61,85%
0,9	15,86%	0,00%	84,14%

l_1-min			
τ	Successi	Errori	FRR
0,1	95,66%	4,09%	0,25%
0,2	93,24%	1,34%	5,43%
0,3	88,98%	0,25%	10,77%
0,4	80,05%	0,08%	19,87%
0,5	66,69%	0,00%	33,31%
0,6	43,32%	0,00%	56,68%
0,7	17,70%	0,00%	82,30%
0,8	1,59%	0,00%	98,41%
0,9	0,00%	0,00%	100%

Tabella 4.9: Risultati ottenuti dalle tre varianti del framework SRC attivando il meccanismo di validazione in assenza di impostori.

Dai risultati sopra riportati è possibile vedere come le soluzioni, trovate utilizzando l'infrastruttura algoritmica WNFCS, dimostrino la maggiore robustezza di questo framework, rispetto alla minimizzazione l_1; infatti, la percentuale di false reiezioni ottenuta dal nuovo metodo risulta inferiore al 20% fino ad un valore soglia compreso fra $0,5$ e $0,7$. Questo risultato potrà essere meglio apprezzato esaminando i grafici in Figura 4.7.

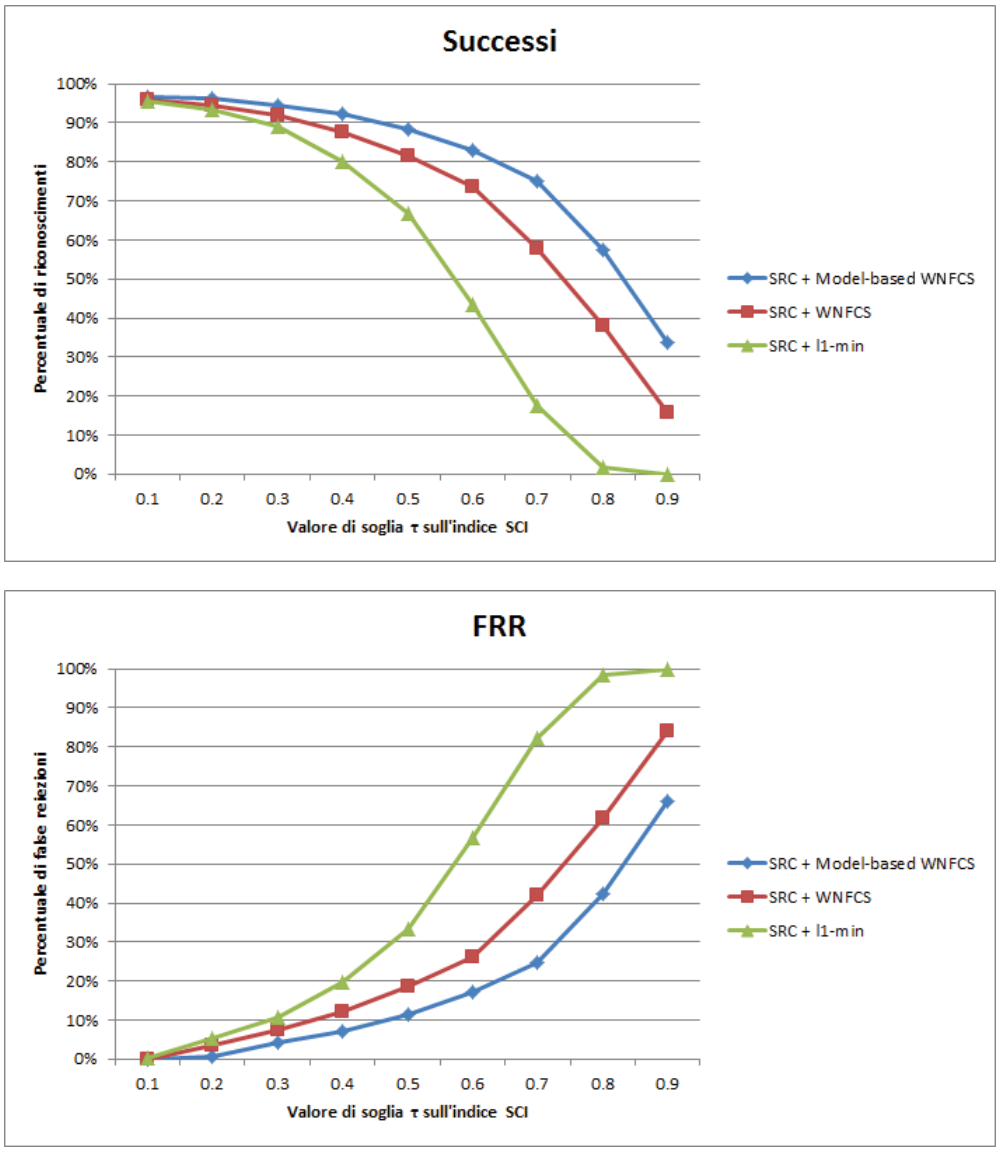


Figura 4.7: Andamento della percentuale di successi e dell'FRR al variare della soglia sull'indice SCI.

3.2 Test di validazione con impostori

In questa sezione verranno infine mostrati i risultati ottenuti dai tre diversi metodi di minimizzazione, sottoponendo al framework SRC un test set composto non soltanto da volti di individui noti, ma anche da rappresentazioni di cosiddetti *impostori*, ovvero di soggetti non contenuti nel training set e dunque sconosciuti al sistema di face recognition. A questo scopo è stato usato

ancora una volta l'Extended Yale B database, sfruttando la stessa composizione di training e test utilizzata anche nella sezione precedente, con l'unica differenza che questa volta sono state eliminate dal training set le immagini relative a 19 dei 38 individui totali. In questo scenario, dunque, il database di training conterrà 608 immagini relative a 19 individui, mentre il test set sarà sempre composto da 1198 immagini appartenenti a 38 soggetti. Anche in questo caso il metodo di estrazione delle feature utilizzato è il downsampling con fattore di ridimensionamento 1/8, corrispondente ad una dimensione del sottospazio pari a 504. Non si forniscono esempi relativi ad altri metodi di feature extraction, in quanto i risultati prodotti al variare di questo fattore, come già dimostrato nel paragrafo 2.1, sono fra loro molto simili.

Model-based WNFCS				
τ	Successi	Errori	FAR	FRR
0,5	91,57%	0,17%	3,34%	4,92%
0,6	91,49%	0,08%	1,09%	7,35%
0,7	90,07%	0,00%	0,17%	9,77%
0,8	85,31%	0,00%	0,00%	14,69%
0,9	80,38%	0,00%	0,00%	19,62%

WNFCS				
τ	Successi	Errori	FAR	FRR
0,5	92,07%	0,00%	0,33%	7,60%
0,6	89,40%	0,00%	0,00%	10,60%
0,7	85,64%	0,00%	0,00%	14,36%
0,8	80,22%	0,00%	0,00%	19,78%
0,9	72,29%	0,00%	0,00%	27,71%

l_1-min				
τ	Successi	Errori	FAR	FRR
0,5	90,82%	0,00%	0,00%	9,18%
0,6	85,23%	0,00%	0,00%	14,77%
0,7	78,21%	0,00%	0,00%	21,79%
0,8	67,61%	0,00%	0,00%	32,39%
0,9	52,00%	0,00%	0,00%	48,00%

Tabella 4.10: Risultati ottenuti dalle tre varianti del framework SRC attivando il meccanismo di validazione in presenza di impostori.

Inoltre, dal momento che, come già sostenuto, la procedura di validazione risulterà tanto più robusta, quanto più il valore di $\tau \longrightarrow 1$, si è deciso di impostare come soglia minima il valore $\tau = 0,5$. In Tabella 4.10 sono riportati

i risultati così ottenuti, dove la colonna *Successi* descrive la percentuale calcolata considerando sia il totale delle immagini ben classificate, sia il totale delle immagini giustamente rigettate, in quanto non valide; la colonna *Errori* contiene la percentuale di immagini mal classificate e le rimanenti colonne descrivono rispettivamente i valori di FAR e FRR.
Analizzando gli esiti di questi test, si nota innanzitutto che la percentuale di successi raggiunta attraverso l'uso dell'algoritmo WNFCS è sempre maggiore di quella ottenuta con il metodo di minimizzazione l_1.
Con soglia $\tau = 0,5$, la versione che sfrutta la struttura della sparsità produce ancora un numero di false accettazioni superiore agli altri due metodi, non raggiungendo il primato di successi; tuttavia, già con $\tau = 0,6$ questo fenomeno viene largamente ridotto, facendo risultare questo metodo il migliore. Infatti, anche al crescere del valore di soglia, il numero di esiti favorevoli non scende mai sotto l'80%. D'altra parte, il metodo di minimizzazione l_1 subisce un'esplosione della percentuale di false reiezioni quando τ viene impostato con un valore maggiore o uguale a $0,8$, a dimostrazione della peggiore qualità delle soluzioni da esso prodotte; mentre l'utilizzo dell'algoritmo WNFCS, nella sua implementazione base, rappresenta un compromesso fra gli altri due metodi.

Conclusioni e sviluppi futuri

Il classificatore basato sulla rappresentazione sparsa (SRC) ha aperto una nuova frontiera di ricerca nell'ambito del riconoscimento di volti, proponendo un approccio radicalmente diverso da quello usato dai metodi classici di face recognition. Ciò lo ha condotto ben presto a ricoprire il ruolo di stato dell'arte in questo settore. In particolare, come è stato dimostrato nel capitolo precedente, questa nuova tecnica permette di raggiungere percentuali di riconoscimento spesso di gran lunga superiori a quelle ottenute dai metodi nearest neighbour e nearest subspace, risultando al contempo anche meno dipendente dalla fase di feature extraction. Oltretutto, la grande semplicità e flessibilità del modello permettono di introdurre modifiche ad hoc per la gestione delle occlusioni; inoltre, l'introduzione del concetto di sparsità garantisce la possibilità di sfruttare questa nuova caratteristica per progettare indici alternativi - come l'SCI - che consentano di irrobustire anche il meccanismo di validazione, fondandolo su informazioni statistiche più significative rispetto alla solita valutazione dei residui.

Proprio per le grandi potenzialità dimostrate dal metodo SRC, dal 2009 - anno in cui esso è stato proposto per la prima volta - ad oggi numerosi gruppi di ricerca hanno tentato di migliorare ulteriormente le sue prestazioni, in modo da renderlo applicabile in un ventaglio sempre più ampio di contesti pratici.

Una delle diverse strade percorribili in tal senso è quella intrapresa in questa tesi. Infatti, come si è visto, la sostituzione della norma l_1 con altre funzioni non convesse e l'introduzione di modelli di sparsità strutturata permettono di ottenere soluzioni qualitativamente migliori rendendo di conseguenza il processo di riconoscimento ancora più affidabile e robusto, specialmente in presenza di occlusioni.

Tuttavia, in futuro, potrebbero essere attuate ulteriori modifiche per rafforzare maggiormente l'efficacia di questo classificatore nei confronti delle occlusioni. Un possibile sviluppo potrebbe essere, a questo proposito, l'utilizzo di un nuovo approccio iterativo che cerchi di eseguire la ricostruzione dell'immagine di test pesando diversamente i vari vincoli del sistema lineare, in maniera tale che quelli corrispondenti ai pixel perturbati abbiano via via sempre meno influenza nel procedimento.

I test effettuati sull'AR database hanno manifestato, inoltre, una notevole vulnerabilità del classificatore SRC nei casi in cui l'immagine di test e quelle di training risultino fra loro disallineate per la presenza di espressioni o diverse pose del volto o ancora per errori avvenuti durante la fase di face detection. Per questo motivo, potrebbe risultare particolarmente vantaggioso apportare alcune modifiche al framework di riconoscimento, ad esempio introducendo al suo interno delle opportune metodologie per la stima della deformazione presente fra le varie immagini dei volti.

Un altro aspetto migliorabile riguarda il criterio di classificazione; infatti, come si è visto nel secondo capitolo, è possibile identificare il volto di test interpretando la struttura sparsa della soluzione in diversi modi. I risultati raggiunti in fase di sperimentazione testimoniano come la regola scelta in questa tesi permetta di ottenere un alto numero di successi sia in termini di classificazione che in termini di validazione; tuttavia, ciò non esclude la ricerca di strategie alternative che possano ulteriormente rafforzare anche questa fase dell'algoritmo.

Infine, sebbene il framework SRC sia stato proposto principalmente come metodo di classificazione per la face recognition, potrebbe risultare interessante valutare le sue performance anche in altri contesti specifici del più generale problema di pattern recognition.

Bibliografia

[1] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, *Robust face recognition via sparse representation*, IEEE transactions on pattern analysis and machine intelligence, Vol. 31, No. 2, 2009.

[2] L.B. Montefusco, D. Lazzaro, S. Papi, *A fast algorithm for nonconvex approaches to sparse recovery problems*, accettato per la pubblicazione su Signal Processing, 2013.

[3] D. Lazzaro, L.B. Montefusco, S. Papi, *Blind cluster structured sparse signal recovery: a deterministic approach*, preprint, 2013.

[4] R.G. Baraniuk, V. Cevher, M.F. Duarte, C. Hegde, *Model-based compressive sensing*, IEEE transactions on information theory, Vol. 56, No. 4, pp. 1982-2001, 2010.

[5] W. Zhao, R. Chellappa, J. Phillips and A. Rosenfeld, *Face recognition: a literature survey*, ACM Computing Surveys, pp. 399-458, 2003.

[6] E. Singer, *Don't I know you?*, MIT Technology Review, 2007 - http://www.technologyreview.com/news/408205/dont-i-know-you/

[7] R. Jafri and R. Arabnia, *A survey of face recognition techniques*, Journal of information processing systems, Vol. 5, No. 2, 2009.

[8] A.S. Tolba, A.H. El-Baz and A.A. El-Harby, *Face recognition: a literature review*, International journal of information and communication engineering, 2006.

[9] M. Turk and A. Pentland, *Eigenfaces for recognition*, Proc. IEEE Int'l conf. computer vision and pattern recognition, 1991.

[10] P. Belhumeur, J. Hespanda and D. Kriegman, *Eigenfaces versus Fisherfaces: recognition using class specific linear projection*, IEEE trans. pattern analysis and machine intelligence, Vol. 19, No. 7, pp. 711-720, 1997.

[11] K. Delac, M. Grgic, P. Liatsis, *Appearence-based statistical methods for face recognition*, 47^{th} international symposium ELMAR, 2005.

[12] I. Jolliffe, *Principal component analysis*, Springer Verlag, New York, 1986.

[13] A. Martinez, A. Kak, *PCA versus LDA*, IEEE trans. on pattern analysis and machine intelligence, Vol. 23, No. 2, pp. 228-233, 2001.

[14] X. He, P. Niyogi, *Locality preserving projection*, Advances in neural information processing systems, Cambridge, 2003.

[15] A. Hyvärinen, E. Oja, *Indipendent component analysis: algorithms and applications*, Neural networks, Vol. 13, pp. 411-430, 2000.

[16] M.S. Bartlett, J.R. Movellan and T.J. Sejnowski, *Face recognition by indipendent component analysis*, IEEE trans. neural networks, Vol. 13, No. 6, pp. 1450-1464, 2002.

[17] B. Schölkopf, A. Smola, K.R. Müller, *Kernel principal component analysis*, Lecture notes in computer science, Vol. 1327, pp. 583-588, 1997.

[18] S. Mika, G. Ratsch, J. Weston, B. Schölkopf, K.R. Müller, *Fisher discriminant analysis with kernels*, Neural networks for signal processing, pp. 41-48, 1999.

[19] X. He, S. Yan, Y. Hu, P. Niyogi and H. Zhang, *Face recognition using laplacianfaces*, IEEE trans. pattern analysis and machine intelligence, Vol. 27, No. 3, pp. 328-340, 2005.

[20] P.S. Penev, J.J. Atick, *Local feature analysis: a general statistical theory for object represantation*, Computation in neural system, No. 7, pp. 477-500, 1996.

[21] Z. Qian, P. Su, D. Xu, *Face recognition based on local feature analysis*, International symposium on computer science and computational technology, 2008.

[22] L. Wiskott, J. Fellous, N. Kruger, C. Von der Malsburg, *Face recognition by elastic bunch graph matching*, Intelligent biometric techniques in fingerprint and face recognition, pp. 355-396, 1999.

[23] T. Ahonen, A. Hadid, M. Pietikainen, *Face recognition with local binary pattern*, Proc. 8^{th} european conference on computer vision, 2004.

[24] N. Sang, J. Wu, K. Yu, *Local Gabor Fisher classifier for face recognition*, 4^{th} international conference on image and graphics, 2007.

[25] S. Shan, P. Yang, S. Chen, W. Gao, *AdaBoost Gabor Fisher classifier for face recognition*, Analysis and modelling of faces and gestures, Vol. 3723, pp. 279-292, 2005.

[26] C. Burges, *A tutorial on support vector machines for pattern recognition*, Data mining and knowledge discovery, Vol. 2, No. 2, pp. 121-167, 1998.

[27] D.L. Donoho, *Compressed sensing*, IEEE transactions on information theory, Vol. 52, pp. 1289-1306, 2006.

[28] E.J. Candès, M.B. Wakin, *An introduction to compressed sampling*, IEEE signal processing magazine, Vol. 21, 2008.

[29] C.E. Shannon, *Communication in the presence of noise*, Proc. institute of radio engineers, Vol. 37, No. 1, pp. 10-21, 1949. Reprint as classic paper in: Proc. IEEE, Vol. 86, No. 2, 1998.

[30] D. Mackenzie, *Compressed sensing makes every pixel count*, in: *What's happening in mathematical sciences*, American mathematical society, Vol. 7, pp. 114-127, 2009.

[31] D. Donoho, *For most large underdetermined systems of linear equations the minimal l_1-norm near solution approximates the sparsest solution*, preprint, 2004.

[32] R. Tibshirani, *Regression shrinkage and selection via the LASSO*, J. Royal statistical society B, Vol. 58, No. 1, pp. 267-288, 1996.

[33] R. Chartrand, W. Yin, *Iteratively reweighted algorithms for compressive sensing*, in Proc. Acoustics, speech and signal processing, pp. 3869-3872, 2008.

[34] R. Chartrand, *Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data*, IEEE international symposium on biomedical imaging: from nano to macro, pp. 262-265, 2009.

[35] E.J. Candès, M.B. Wakin, S.P. Boyd, *Enhancing sparsity by reweighted l_1 minimization*, J. Fourier analysis appl., Vol. 14, pp. 877-905, 2008.

[36] N. Mourad, J.P. Reilly, *Minimizing nonconvex functions for sparse vector reconstruction*, IEEE transactions on signal processing, Vol. 58, No. 7, pp. 3485-3496, 2010.

[37] P.L. Combettes, V.R. Wajs, *Signal recovery by proximal forward-backward splitting*, SIAM journal on multiscale modelling and simulation, Vol. 4, No. 4, pp. 1168-1200, 2005.

[38] A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, *Fast L1-minimization algorithms and an application in robust face recognition: a review*, International conference on image processing, 2010.

[39] A.Y. Yang, A. Ganesh, Z. Zhou, S.S. Sastry, Y. Ma, *A review of fast l_1-minimization algorithms for robust face recognition*, preprint, 2010.

[40] E. Elhamifar, R. Vidal, *Robust classification using structured sparse representation*, IEEE conference on computer vision and pattern recognition, pp. 1873-1879, 2011.

[41] K. Jia, T.-H. Chan, Y. Ma, *Robust and practical recognition via structured sparsity*, European conference on computer vision, 2012.

[42] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, *From few to many: illumination cone models for face recognition under variable lighting and*

pose, IEEE transactions on pattern analysis and machine intelligence, Vol. 23, No. 6, pp. 643-660, 2001.

[43] K.C. Lee, J. Ho, D.J. Kriegman, *Acquiring linear subspaces for face recognition under variable lighting*, IEEE transactions on pattern analysis and machine intelligence, Vol. 27, No. 5, pp. 684-698, 2005.

[44] A.M. Martinez, R. Benavente, *The AR face database*, CVC technical report, No. 24, 1998.

[45] D. Donoho, V. Stodden, Y. Tsaig, *About SparseLab*, Technical report, Stanford University, 2007.