



***TECNICHE INNOVATIVE PER  
L'INDIVIDUAZIONE AUTOMATICA DEGLI  
OGGETTI SULLE IMMAGINI SATELLITARI***

di

Giancarlo Santilli

**Tesi di Dottorato del XXIV Ciclo**

**Scuola di Ingegneria Aerospaziale  
Sapienza – Università di Roma**

Tutore: Prof. Giovanni Laneve

**2012**



## RINGRAZIAMENTI

*A mis padres, Lili, Giovanni y todas las personas que me acompañaron durante este recorrido...*



## INDICE DELLA TESI

<b>RINGRAZIAMENTI</b>	<b>2</b>
<b>INTRODUZIONE ALLA TESI</b>	<b>6</b>
<b>CAPITOLO I: Introduzione al Pattern Recognition</b>	<b>13</b>
1.1 CAMPI D'APPLICAZIONE	14
1.2 SISTEMA DI PATTERN RECOGNITION	16
1.3 APPROCCIO STATISTICO E STRUTTURALE	21
1.3.1 PATTERN RECOGNITION STATISTICO	22
1.3.2 PATTERN RECOGNITION STRUTTURALE	24
1.3.3 RAPPRESENTAZIONE IN TERMINI DI ARG	26
1.4 QUALE APPROCCIO: STATISTICO O STRUTTURALE ?	27
<b>CAPITOLO II: Le Fasi della Descrizione</b>	<b>29</b>
2.1 ACQUISIZIONE	31
2.2 FILTAGGIO DEL RUMORE E SOGLIATURA	31
2.3 THINNING O ASSOTIGLIAMENTO DELL'IMMAGINE	32
2.4 PRUNING	33
2.5 LABELLING DELLO SCHELETRO	33
2.6 APPROSSIMAZIONE POLIGONALE	34
<b>CAPITOLO III: Pattern Recognition basato sul Modello</b>	<b>37</b>
3.1 TEMPLATE MATCHING	39
3.1.1 SOMMA DELLE DIFFERENZE QUADRATE (SDQ)	40
3.1.2 SOMMA DELLE DIFFERENZE ASSOLUTE (SAD)	40
3.1.3 CROSS-CORRELATION NORMALIZZATA (NCC)	40
3.1.4 CROSS-CORRELATION NORMALIZZATA A MEDIA NULLA (ZNCC)	41
3.2 OTTIMIZZAZIONE DELLA TECNICA	42
3.3 SISTEMA DI PATTERN RECOGNITION RECOGNITION BASATO SUL TEMPLATE MATCHING	43
<b>CAPITOLO IV: Pattern Recognition Statistico</b>	<b>54</b>
4.1.1 PROCESSO DI RICONOSCIMENTO	55
4.1.2 ESTRAZIONE DELLE FEATURES	56
4.1.3 CLASSIFICAZIONE DI PATTERN	57
4.2 LA TEORIA DEI MOMENTI	57
4.2.1 I MOMENTI REGOLARI	57
4.2.2 MOMENTI CENTRALI	58
4.2.3 MOMENTI ORTOGONALI: MOMENTI DI ZERNIKE	60
4.2.4 ALTRI TIPI DI MOMENTI	63
4.3 SISTEMA DI PATTERN RECOGNITION BASATO SUI MOMENTI GEOMETRICI INVARIANTI	64
4.4 ULTERIORI CONSIDERAZIONI SUI MOMENTI	70
4.4.1 AFFIDABILITA' DEI MOMENTI	70
4.4.2 SENSIBILITA' AL RUMORE	71
4.4.3 RAPPRESENTATIVITA' DEI MOMENTI	72
<b>CAPITOLO V: Pattern Recognition Strutturale – Sintattico</b>	<b>75</b>
5.1 MORFOLOGIA MATEMATICA	77



5.2	OPERATORI D'INSIEME APPLICATI ALLE IMMAGINI	78
5.3	RELAZIONI D'ORDINE	79
5.4	TRASFORMAZIONI DA IMMAGINE A IMMAGINE	80
5.5	TRASFORMAZIONI MORFOLOGICHE	81
5.5.1	ELEMENTO STRUTTURANTE (ES)	81
5.5.2	OPERATORE EROSIONE	82
5.5.3	OPERATORE DILATAZIONE	84
5.5.4	OPERATORI OPENING E CLOSING	85
5.6	ALGORITMI MORFOLOGICI DI BASE IN SCALA DI GRIGIO	86
5.7	SISTEMA DI PATTERN RECOGNITION BASATO SULLA MORFOLOGIA MATEMATICA	87
5.8	MONITORAGGIO CAMPI DI RIFUGIATI IN AFRICA	87
5.9	INDIVIDUAZIONE AUTOMATICA DI STRADE SU IMMAGINI OTTICHE E SAR	98
<b>CAPITOLO VI: Pattern Recognition basato sulle Reti Neurali</b>		<b>102</b>
6.1	IL NEURONE (MODELLO DI McCULLOCH-PITTS)	103
6.2	FUNZIONI D'ATTIVAZIONE	105
6.2.1	FUNZIONE THRESHOLD	105
6.2.2	FUNZIONE PIECEWISE-LINEAR	105
6.2.3	FUNZIONE SIGMOIDE	105
6.3	ARCHITETTURA DELLE RETI NEURALI	106
6.3.1	RETE FEEDFORWARD AD UNO STRATO	106
6.3.2	RETE FEEDFORWARD A PIU' STRATI	106
6.3.3	RETE RICORRENTE O FEEDBACK	107
6.4	PROCESSI DI APPRENDIMENTO	107
6.4.1	APPRENDIMENTO CON CORREZIONE DI ERRORE	107
6.4.2	APPRENDIMENTO BASATO SULLA MEMORIA	108
6.5	APPRENDIMENTO DI HEBBIAN	109
6.6	APPRENDIMENTO COMPETITIVO	109
6.7	ATTIVITA' DI APPRENDIMENTO	111
6.7.1	ASSOCIAZIONE DI PATTERN	111
6.7.2	RICONOSCIMENTO DI PATTERN	111
6.7.3	APPROSSIMAZIONE DI FUNZIONI	112
6.8	SINGLE LAYER PERCEPTRON (SLP)	113
6.8.1	TEOREMA DI CONVERGENZA DEL PERCETTRONE	113
6.8.2	ALGORITMO DI APPRENDIMENTO DEL PERCETTRONE	115
6.9	ESEMPIO DI APPLICAZIONE: OPERATORI BOOLEANI	116
6.9.1	NOT	116
6.9.2	AND	117
6.9.3	OR	117
6.9.4	XOR	117
6.10	MULTILAYERS PERCEPTRONS	117
6.10.1	ALGORITMO DI BACK-PROPAGATION	118
6.10.2	NEURONE J DI OUTPUT	118
6.10.3	NEURONE J NASCOSTO	120
6.11	IN GENERALE	121
6.12	FATTORE DI APPRENDIMENTO	121
6.13	CRITERI D'ARRESTO	122
6.14	REGOLE EURISTICHE PER MIGLIORARE L'ALGORITMO DI BACK.-PROPAGATION	122
6.15	GENERALIZZAZIONE	125
6.16	APPROSSIMAZIONE DI FUNZIONI	126
6.17	CROSS-VALIDATION	126
6.17.1	SELEZIONE DEL MODELLO	127



6.18	RETI DI HOPFIELD	128
6.18.1	CASO DISCRETO	128
6.18.2	CASO CONTINUO	130
6.18.3	PROBLEMI	131
6.19	INDIVIDUAZIONE DI VERSAMENTI DI PETROLIO IN MARE SULLE IMMAGINI SAR	131
	<b>CONCLUSIONI</b>	<b>149</b>
	<b>REFERENZE BIBLIOGRAFICHE</b>	<b>153</b>



## Introduzione

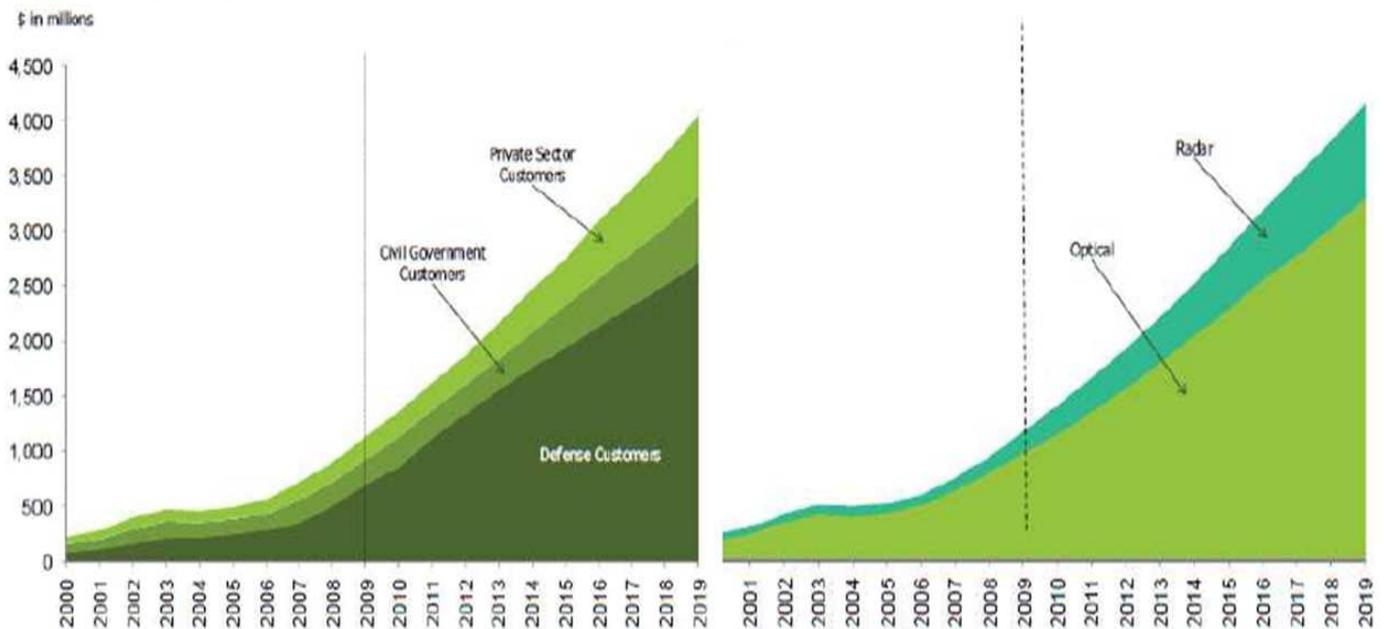
I satelliti in orbita terrestre, grazie alla loro capacità di avere una visione globale di vaste aree della Terra, senza vincoli politici o geografici, offrono considerevoli vantaggi rispetto ai sistemi terrestri. Ciò ha portato all'applicazione, e da circa un decennio anche commerciale, dei satelliti in numerose aree quali ad esempio, le telecomunicazioni, il telerilevamento e la meteorologia.

Il **telerilevamento** rappresenta l'insieme delle tecniche che consentono di ottenere informazioni qualitative e quantitative relative ad un oggetto, un'area o un fenomeno tramite l'analisi dei dati acquisiti da opportuni sensori che non sono in contatto con l'oggetto, l'area o il fenomeno investigato stesso.

Con il lancio nel 1972 del primo satellite americano della serie LANDSAT, si apriva l'era dell'osservazione della Terra per scopi civili mediante l'uso di satelliti artificiali (EOS). Da allora, visto il successo di tale missione e le grandissime potenzialità offerte dall'osservazione satellitare della Terra, la crescita del numero di satelliti destinati a questo compito e la corrispondente quantità di dati generati, non ha più conosciuto sosta.

L'osservazione della Terra da satellite, mediante l'acquisizione remota di dati scientifici e di immagini, è stata guidata per quasi tre decenni, dagli investimenti compiuti dal settore pubblico dei paesi leader nel settore aerospaziale. Ma nell'ultimo decennio lo scenario di questo settore è notevolmente cambiato, grazie all'ingresso di un nuovo protagonista, il settore commerciale.

Dalla sua comparsa, circa 12 anni fa, ad oggi il settore commerciale ha vivacizzato notevolmente il mercato dei dati satellitari EO, imprimendogli una notevole crescita (figura 1), attraverso la fornitura di nuovi dati, prodotti e servizi creati ad hoc per meglio incontrare le esigenze e le richieste del mercato. E il trend è ancora molto sostenuto. Infatti come dimostrano i dati di settore (studio Euroconsult, “*Satellite-based Earth Observation*” report, 4th Edition 2011), la tendenza si manterrà forte e solida almeno per un altro decennio, tanto in termini di ricavi (figura 1) quanto in termini di volume di dati generati, grazie alle previsioni relative al numero dei satelliti EO che saranno messi in orbita fino al 2020 (figura 2).



**Figura 1.** Andamento nel periodo 2000–2019 dei ricavi dati satellitari commerciali. Domanda di dati: clienti per la vendita dati commerciali (sinistra). Domanda dati satellitari EO per tipo: Ottici e Radar (destra).

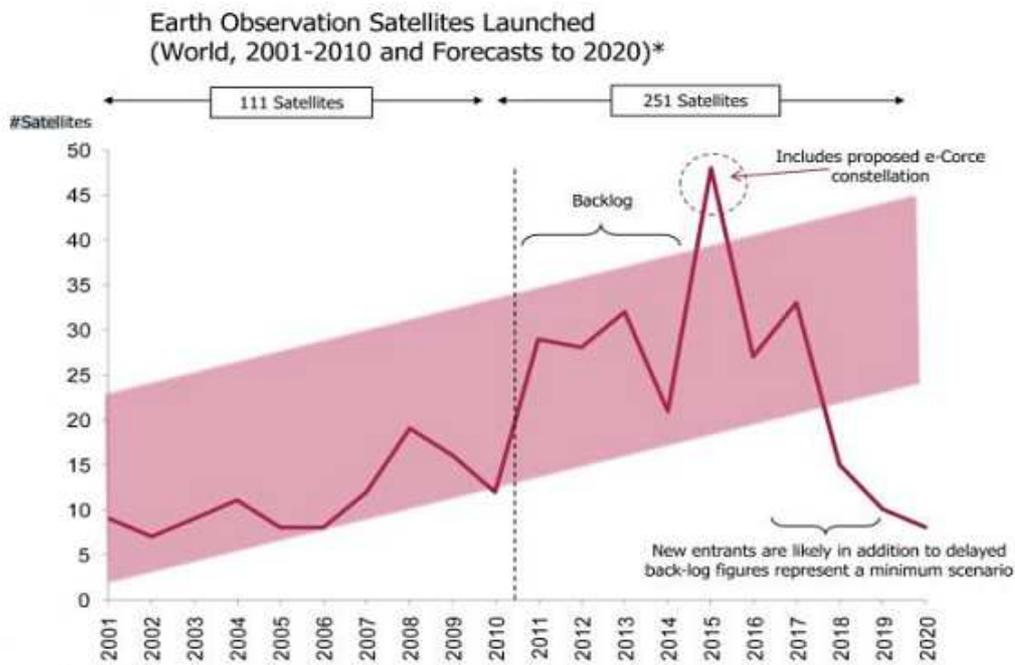
Questa notevole crescita del settore, prevista per il prossimo decennio, è dovuta principalmente ai seguenti fattori:

- miglioramento delle capacità tecnologiche dei moderni satelliti/sensori;
- miglioramento delle stazioni di terra e delle reti per la distribuzione dei dati/immagini;
- maggiori incentivi e sensibilizzazione verso le scuole e gli utenti finali;
- servizi più efficienti e su misura rivolti alle esigenze degli utenti finali;
- i satelliti diventano sempre più specializzati, realizzando missioni ad hoc tanto per la scienza e la ricerca quanto per scopi operativi (marittimo, oceanografici, etc);
- maggiori e più efficienti portali per la visualizzazione e l'acquisto delle immagini e prodotti;

La figura 2 dimostra come il numero di satelliti per l'osservazione della Terra (EOS), è destinato a raddoppiare nel decennio 2011-2020 raggiungendo una cifra record superiore ai 250 satelliti (previsione conservativa), rispetto ai circa 120 operativi nel decennio 2001-2010, generando di conseguenza un aumento esponenziale di dati e di

immagini satellitari disponibili, di fronte al quale i settori dedicati non dovranno farsi trovare impreparati.

La parte dei dati per per l'osservazione della terra (EO) che suscita più interesse, è principalmente quella relativa alla vendita di immagini HR (High Resolution) e VHR (Very High Resolution), le quali sono trainate dalle esigenze governative. Infatti, proprio la Difesa e la Sicurezza dominano il mercato dell'acquisto dei dati, con circa il 66% del mercato, mentre le applicazioni civili (pubbliche amministrazioni, ricerca, settore privato, ecc) gestiscono maggiori volumi di dati. Questi dati ci portano a concludere che, nonostante la comparsa di un altro player sul mercato dei satelliti EO, è la domanda governativa a dominare e guidare l'intero processo.



**Figura 2.** Numero di satelliti EO (commerciali, governativi) lanciati nel decennio 2001-2010 e la previsione di quelli che saranno messi in orbita (ottici e RADAR) nel prossimo decennio 2011-2020. Ricerca della Euroconsultant: "Satellite-based Earth observation" report, 4th Edition 2011.

La grande crescita di dati satellitari disponibili, corrente e attesa per il prossimo decennio, deve essere accompagnata da una crescita altrettanto robusta dei servizi a valore aggiunto associati agli stessi. Tuttavia, fino ad oggi le immagini e i prodotti derivati da satellite o da altre tipologie di piattaforma sono stati utilizzati sistematicamente, per l'analisi e il monitoraggio del territorio e dell'ambiente, quasi esclusivamente da enti di ricerca e i settori specializzati. Problemi tecnici, logistici e politici hanno limitato l'uso e la diffusione massiva del dato satellitare finora. Inoltre, l'offerta commerciale e istituzionale di servizi a valore aggiunto



associati ai dati satellitari EO, non ha registrato la stessa crescita sperimentata nell'acquisizione e nella vendita dei dati.

Ma il settore dei servizi a valore aggiunto potrebbe recuperare il proprio ritardo nei prossimi anni, poiché i cambiamenti economici e politici in atto dovrebbero incidere positivamente sulla catena di valore aggiunto, per esempio attraverso:

- la comparsa di fornitori di soluzioni e servizi ad elevato valore aggiunto regionali, in grado di avere migliore accesso al mercato locale;
- la crescita dei dati operativi low-cost/gratuiti (come il Charter) e maggiore interazione fornitore/utente finale (come i progetti europei GMOSS, GMOSAIC, etc)
- i meccanismi per lo sviluppo di servizi ad elevato valore aggiunto e la diffusione dei dati alla comunità come GMES (Global Monitoring of Environment and Security);

Dalle considerazioni precedenti appare evidente che, vista la grande disponibilità di dati e immagini satellitari corrente e futura, sarà necessario un grande sforzo da parte del mondo della ricerca e delle società di servizi nello sviluppo di algoritmi e applicazioni che siano in grado di elaborare, nel modo più accurato e celere possibile, grandi quantità di dati per estrarre le informazioni in essi contenute e incontrando così meglio le esigenze degli utenti finali. Infatti, il vero valore del dato satellitare risiede nelle informazioni che esso contiene e che è in grado di svelare solo attraverso un adeguato processo di manipolazione, elaborazione ed estrazione. In definitiva, il dato satellitare di per se non è in grado di fornire grandi informazioni, se non solo attraverso l'applicazione di opportuni algoritmi, metodi e procedure automatiche, semi-automatiche e manuali create ad hoc, per ottimizzare il processo di estrazione delle informazioni contenute nel dato.

Il grande livello tecnologico raggiunto nella costruzione dei satelliti/sensori per l'osservazione della Terra e il progresso della scienza del telerilevamento, rende il loro impiego unico e indispensabile in tanti campi applicativi della vita quotidiana. Settori come la difesa e la sicurezza, il monitoraggio ambientale e delle risorse naturali, l'energia, la gestione e il monitoraggio dei disastri naturali e antropici, ecc (tabella 1), avrebbero un'efficacia notevolmente ridotta senza l'uso del satellite, i dati derivati e le tecniche del telerilevamento largamente diffuse e impiegate oggi.

Il CRPSM (Centro di Ricerca Progetto San Marco), ha rivolto il suo interesse verso questa problematica, in seguito al suo coinvolgimento in diversi progetti nazionali ed internazionali, quali: GMOSAIC, GMOSS,

SIGRI, OSDML, ecc. I compiti specifici del CRPSM, nell'ambito delle proprie attività, hanno riguardato il monitoraggio dei confini nazionali, delle popolazioni e degli obiettivi/infrastrutture sensibili presenti in certi scenari. Questo ha rappresentato un'opportunità, e inoltre ci ha fornito lo spunto per sviluppare diversi algoritmi e tecniche, atte all'individuazione automatica di oggetti sulle immagini satellitari.

	Defense & Security	Environment monitoring	Energy	Natural resources monitoring	Maritime	Disaster management	Location-based services
Optical data needs	HR data	Instruments diversity for atmospheric study, land-sea interact, ozone...	HR (infrastructure, logistics) & MR (geological maps)	MR (20-30m) in IR for wide agriculture; HR in precision agri. & forest	MR/LR for algae. Fisheries & costal zones	HR for logistics and crisis mapping	HR high accuracy
Radar data needs	HR data with night imaging and no cloud	Ocean currents model, tidal info from MR	HR/MR (offshore seepage) + optical for geology maps. InSAR for subsidence	Surface texture/water content for land (geology, agriculture)	HR/MR for sea ice. HR for ship track, transport	HR in flood mapping	Few needs
Revisit needs	Max. for 1m is 1 day	Depend on scope of the study, incl. real time meteo, archive & continuity	Recent & NRT. Archive good for geological map	Low for monitoring (agriculture, forest)	Low for ship track	HR for responsiveness (before & after mapping)	Recent archive for most of needs
Data sources	Proprietary systems + HR commercial	Scientific + low cost MR	Low cost MR/HR + HR commercial	Low cost MR + HR	Low cost MR + HR commercial	HR commercial + Charter on Space	HR commercial

**Tabella 1.** *Importanza del dato satellitare EO in diversi campi con le relative necessità in termini di risoluzione temporale e spaziale.*

Il riconoscimento degli oggetti (*Pattern Recognition*) presenti nelle immagini satellitari rappresenta una delle applicazioni più importanti e antiche del telerilevamento. Alla luce delle considerazioni precedenti, e vista la grandissima importanza strategica di questo argomento in tanti campi della vita quotidiana, risulta evidente che la ricerca e il settore dei servizi a valore aggiunto dovranno dedicare molti sforzi e risorse al problema dell'*automatizzazione* del processo di *Pattern Recognition*, in modo da sfruttare efficientemente l'enorme quantità di immagini satellitari derivate da missioni passate, presenti e future.

L'attività del riconoscimento degli oggetti presenti nelle immagini, può essere considerata semplice finché non si tenta di automatizzarla, ovvero di far riconoscere automaticamente oggetti, modelli e simboli ad un computer. E infatti, la complessità e le difficoltà del *riconoscimento automatico degli oggetti* nelle immagini aumentano notevolmente, rispetto al riconoscimento manuale tradizionale.



Questo studio rappresenta la sintesi del lavoro svolto durante la mia attività di dottorato. La sfida è stata dunque quella di risolvere il problema dell' *individuazione automatica degli oggetti presenti nelle immagini satellitari*, attraverso la descrizione del problema, lo studio delle tecniche e i metodi più adatti a svolgere il compito (evidenziando pregi e difetti di ciascuno), e l'implementazione di numerosi algoritmi che, in modo automatico, riconoscono ed estraggono gli oggetti desiderati dalle immagini satellitari oggetto di studio.

Il presente lavoro di tesi e l'intera attività di dottorato sul *Pattern Recognition* è stata svolta presso il Centro di Ricerca Progetto San Marco (CRPSM) della Sapienza – Università di Roma.

La presente tesi è stata così articolata:

- *Capitolo 1*: in questo capitolo si inquadra il contesto generale in cui si sviluppa il presente lavoro di tesi. In particolare, si dà una descrizione generale del problema e dei passi che portano al riconoscimento automatico degli oggetti presenti sulle immagini satellitari, evidenziando i le metodologia e tecniche usate per raggiungere questo scopo.
- *Capitolo 2*: dopo aver inquadrato il problema e il contesto applicativo, si espone con maggiore dettaglio, la fase di descrizione di una immagine digitale e le principali fasi di pre-processing adottate. In particolare, quest'ultima fase è fondamentale nel processo di estrazione delle caratteristiche salienti, utili nel successivo passo di riconoscimento degli oggetti presenti in una immagine digitale.
- *Capitolo 3*: si introduce la teoria del *Template Matching*, e i principi di funzionamento della tecnica che sono in grado di condurre al riconoscimento automatico degli oggetti. Inoltre, vengono evidenziati pregi e difetti assoluti e relativi ad altre tecniche, focalizzando i criteri che portano a un buon funzionamento della stessa nei vari contesti applicativi. Il capitolo si conclude con la presentazione dei casi trattati e i risultati raggiunti mediante l'implementazione di propri algoritmi di *template matching*.
- *Capitolo 4*: vengono introdotti i concetti relativi alla tecnica dei *Momenti Invarianti* (tecnica di classificazione statistica) e i principi di funzionamento della tecnica che sono in grado di condurre al riconoscimento automatico degli oggetti. Vengono inoltre, evidenziati i pregi e difetti assoluti e relativi ad altre tecniche, stabilendo i criteri che portano a un buon



funzionamento della stessa nei vari contesti applicativi. Il capitolo si conclude con la presentazione dei casi trattati, e i risultati raggiunti mediante l'implementazione di propri algoritmi sui *momenti invarianti*.

- *Capitolo 5*: in questo capitolo viene introdotta e descritta la tecnica della *Morfologia Matematica* (metodo strutturale), presentando i principi di funzionamento che rendono tale tecnica così potente e versatile. Inoltre, vengono evidenziati i pregi e difetti assoluti e relativi ad altre tecniche, stabilendo i criteri che portano a un corretto funzionamento della stessa nei vari contesti applicativi. Il capitolo si conclude con la presentazione dei molti casi trattati, e i risultati raggiunti mediante l'implementazione di propri algoritmi di *morfologia matematica*.
- *Capitolo 6*: questo capitolo chiude l'insieme delle tecniche adottate in questo lavoro, per l'individuazione automatica degli oggetti nelle immagini satellitari/aeree, introducendo le *Reti Neurali* e il loro principio di funzionamento in questo contesto. Vengono evidenziati, come fatto pure per le altre tecniche, pregi e difetti assoluti e relativi ad altre tecniche, stabilendo i criteri che portano ad un corretto funzionamento della stessa nei vari contesti applicativi. Il capitolo si conclude con la presentazione dei casi trattati, e i risultati raggiunti mediante l'implementazione di propri algoritmi basati sulle *reti neurali*.



## Capitolo 1

# Introduzione al Pattern Recognition

Gli esseri umani e le specie animali, nel corso dell'evoluzione hanno sviluppato notevoli attitudini, che gli hanno permesso di imparare a riconoscere: forme, suoni, odori, etc. Distinguere i “rumori” dei predatori, riconoscere gli amici ed i nemici, . . . , riconoscere la scrittura, etc., sono processi che vengono svolti in massima parte a livello inconscio. Gli studiosi, nel corso degli anni hanno impiegato notevoli sforzi nella ricerca di tecnologie che gli permettessero di creare una macchina che avesse le stesse capacità. Molte ricerche sono state condotte in ambito tecnologico per creare una macchina in grado di riconoscere particolari configurazioni in maniera automatica.

*“Pattern Recognition can be defined as the scientific discipline that studies theories and methods for designing machines that are able to recognise patterns in noisy data . . . . . Pattern Recognition has an “engineering” nature, as its final goal is the design of machines” [1].*

Il Pattern Recognition (“riconoscimento di forme”) è lo studio su come le macchine possano osservare l'ambiente, imparare a distinguere *patterns* di interesse dalle osservazioni acquisite in precedenza e classificarli in maniera opportuna [1]. Ma che cosa si intende esattamente per “*Pattern*”? Watanabe [2] afferma che: un “*pattern*” è l'opposto del caos; è una entità, vagamente definita alla quale può essere attribuito un nome. Esempi di pattern possono essere: l'immagine di una impronta digitale, una parola scritta in corsivo, l'immagine di un volto umano, oppure un segnale acustico. Dunque il Pattern Recognition implica, dato un determinato oggetto, la sua identificazione in base alle conoscenze relative ad un modello cui ricondurlo [3].

Occorre fare una distinzione netta tra processi di descrizione, classificazione, riconoscimento ed interpretazione. Tali concetti sono difficili da separare e sono attuati progressivamente in modo inconscio nella mente umana. Essi, tuttavia presentano delle differenze, ed in particolare gli possono essere assegnate le seguenti definizioni:

- *Descrizione*: individuare, all'interno di una forma un insieme di primitive descrittive in grado di rappresentarla adeguatamente, in



modo da facilitare il successivo passo di classificazione (es. descrivere la lettera “P” come composta da due primitive strutturali);

- *Classificazione*: assegnazione di una “forma” (“*pattern*”) incognita ad una classe (es. classificare le varie “lettere” in un testo);
- *Riconoscimento*: confronto di un oggetto, che può essere composto da più “*pattern*”, con un modello predefinito (es. riconoscere una “parola” dopo aver classificato le lettere che la compongono);
- *Interpretazione*: analisi volta ad evidenziare gli aspetti semantici dei dati di ingresso (ossia a dargli un “significato”, anche legato alla specifica applicazione). Ad es. interpretare un testo, dopo averne riconosciuto tutte le parole, e decidere che tratta di “economia” e non di “sport”.

Dato un *pattern*, il suo riconoscimento (detto anche classificazione) può essere principalmente effettuato in due modi [2]:

1. *supervisionato*, in cui il *pattern* in input viene identificato come membro di una classe tra quelle predefinite, ovvero viene classificato in maniera supervisionata avendo l'utente o il progettista definito a priori le classi di interesse;
2. *non supervisionato*, in cui il *pattern* in input viene assegnato ad una classe sconosciuta a priori, ovvero i *pattern* vengono raggruppati nei cosiddetti cluster (tecniche di clustering).

Il problema del *pattern recognition* quindi viene posto nella forma di classificazione o identificazione delle categorie di appartenenza, dove le classi o categorie possono essere sia definite dal progettista del sistema (nei metodi supervisionati), sia apprese dalle similarità tra i *pattern* (nei metodi non supervisionati).

## 1.1 Campi di applicazione

I campi di applicazione del “*Pattern Recognition*” sono svariati e toccano diverse branche della tecnologia. Si può andare dalla gestione di un processo industriale, all'ambito medico, al riconoscimento visivo e sonoro. In particolar modo, il riconoscimento di forme può essere impiegato:

- *nel campo del telerilevamento*: per l'individuazione automatica degli oggetti in diversi contesti che vanno dalla sicurezza, al monitoraggio dei confini e popolazioni, ecc come sarà dimostrato nel corso di questo lavoro.



- *nel campo biomedico*: per interpretare le immagini prodotte da diverse apparecchiature di diagnosi, allo scopo di localizzare meglio alcune zone di tali immagini, facilitando in questo modo la diagnosi medica;
- *nell'elaborazione di testi*: realizzazione di sistemi di conversione automatica di documenti su supporto cartaceo; riconoscimento di caratteri manoscritti;
- *in biometria*: identificazione personale basata su vari attributi fisici, quali viso, impronta digitale, etc.
- *nella sicurezza delle reti di calcolatori*: rilevazione di attacchi intrusivi su sistemi informatici grazie allo studio del traffico di rete [4].

I sistemi di pattern recognition creano, nuovi scenari applicativi sia a livello accademico che industriale. Si pensi, per esempio, ai vantaggi per un processo industriale che, impiegando il riconoscimento automatico di oggetti, può beneficiare di un controllo qualità automatizzato. Sempre nell'ambito industriale, l'impiego di algoritmi di riconoscimento automatico, porta ad arricchire le funzionalità dei robot impiegati, come ad esempio la possibilità di muoversi in determinati ambienti riconoscendo automaticamente gli ostacoli.

Già da questi primi esempi appare chiaro come l'uso di tecniche di riconoscimento automatico possa notevolmente migliorare le attuali applicazioni tecnologiche in diversi campi, dal settore scientifico a quello sociale ed industriale. Di seguito (tabella 1.1) si riporta una sintesi dei principali domini applicativi dei sistemi di riconoscimento automatico evidenziando anche le classi di oggetti/entità coinvolte nei suddetti processi.

Il rapido sviluppo della potenza di calcolo ha permesso l'elaborazione di grandi quantità di informazioni in tempi rapidi, facilitando l'uso di diversi ed elaborati metodi di analisi e di classificazione di dati. Allo stesso tempo, la crescita delle dimensioni dei database e l'esigenza di maggiori performance (velocità, accuratezza, costi, ecc ) hanno portato all'aumento della richiesta di sistemi automatici di pattern recognition. Tuttavia, in cinquanta anni di ricerca si è arrivati alla conclusione che non esiste un unico approccio “ottimale” e “generale” alla classificazione, ma esistono molti metodi che possono essere utilizzati, e la scelta di integrarli è la migliore da perseguire. Se il riconoscimento di forme “automatizzato” è, dunque possibile, molto dipende da come il sistema viene costruito e definito.

|

Dominio	Applicazioni	Pattern	Classi
Analisi di documenti	Ricerca su internet e Web Semantica	Documenti testuali / corpus di documenti digitali o digitalizzati	Categorie semantiche (esempio, affari, sport, scienza, ecc.)
Analisi di immagini di documenti	Strumento di supporto per i diversamente abili, digitalizzazione automatica di documenti.	Immagini di documenti (esempio, acquisite da scanner).	Riconoscimento dei caratteri (OCR), parole, regioni di testo
Automazione Industriale	Verifica delle schede dei circuiti, verifica del software	Immagini all'infrarosso o spettroscopiche, e moduli software)	Rilevamento di componenti hardware e software difettosi
Data Mining	Ricerca di pattern significativi in insiemi di dati / documenti	Punti nello spazio multidimensionale / vettori multidimensionali	Gruppi omogenei e ben separati (cluster)
Bioinformatica	Analisi sequenze DNA	Sequenze DNA / Proteine	Tipi di geni / proteine
Information Retrieval su basi dati multimediali.	Ricerca su internet o in reti aziendali di immagini ed audio / video	Video clip ed immagini	Generi video, tipi di immagini o tipologie di inquadrature (ed. nello sport: goal, rigore, fuorigioco, ecc.)
Riconoscimento Bio-Metrico	Identificazione personale attraverso rilevazione dati biometrici	Immagini del volto, dell'iride, dell'impronta digitale	Gestione controllata accessi
Riconoscimento del parlato	Inserimento dati per persone diversamente abili, interfaccia audio per l'uso di telefonia	Onde sonore della Voce.	Parole pronunciate, suoni modulati
Sensor Data Fusion	Riconoscimento della persona e della posizione geografica attraverso diversi sensori.	Informazioni provenienti da sensori (esempio, accesso mediante RFID - identificazione a radio frequenza)	Posizione, risorse disponibili, contesto dell'utente
Computer Network	Detezione di intrusioni informatiche	Log di connessione	Rilevazione di intrusioni

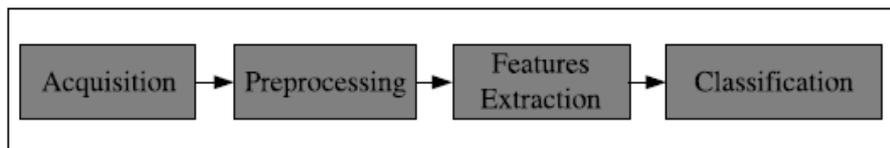
*Tabella 1.1 Applicazioni del Pattern Recognition.*

## 1.2 Sistema di Pattern Recognition

La progettazione di un sistema di “*pattern recognition*”, per l'elaborazione di immagini digitalizzate può essere scomposto in più fasi distinte (*acquisizione, pre-elaborazione e normalizzazione, estrazione delle caratteristiche o rappresentazione, decisione e classificazione*), come evidenziato nella figura 1.1

Una immagine digitale può essere interpretata come una funzione bidimensionale  $f : R \times R \rightarrow R$  il cui valore ne rappresenta l'intensità luminosa. Il blocco di “*acquisizione*”, in termini semplificati, è

composto da tutte quelle fasi che portano da una immagine ad una sua rappresentazione sotto forma di matrice numerica (valida sia per le immagini pancromatiche che per le multispettrali). L'immagine viene suddivisa in piccole regioni, chiamate *pixel*, individuati nella matrice da un numero di riga e un numero di colonna. Ad ogni pixel viene associato un valore che corrisponde all'intensità luminosa della componente della relativa matrice, nell'area elementare corrispondente. Tale blocco comprende, in generale, i passi di campionamento e di quantizzazione in livelli di grigio, necessari per la sua discretizzazione nello spazio e nell'ampiezza.



**Figura 1.1.** Schema a blocchi di un sistema di Pattern Recognition per il riconoscimento di immagini digitali.

L'insieme dei *pattern* (modelli) di interesse, è, infatti, rappresentato da uno spazio  $S_x$ , generalmente di tipo continuo, e per ottenerne una rappresentazione informatica occorre effettuare una discretizzazione come mostrato nella figura 1.1.

Il blocco di “*pre-processing*” permette, attraverso una serie di elaborazioni preliminari, di rappresentare l'immagine originale in una maniera più adatta ad essere trattata nelle fasi successive di elaborazione, determinando il buon funzionamento del riconoscitore.

Nel caso delle applicazioni trattate in questa tesi di dottorato, lo scopo principale del *pre-processing* è segmentare la regione da riconoscere (tende, aerei, macchine, ecc) dallo sfondo dell'immagine rispetto ad una o più proprietà discriminanti, quali ad esempio l'intensità del tono di grigio, e da questa ricavare l'immagine binarizzata dell'oggetto d'interesse .

Questa fase è estremamente delicata poiché, come detto precedentemente, influisce sulle prestazioni e buon funzionamento del sistema di riconoscimento. In generale, non esistono procedure di *pre-processing* automatiche, in grado di garantire i migliori risultati in tutti i campi di applicazione e per tutti i sensori.

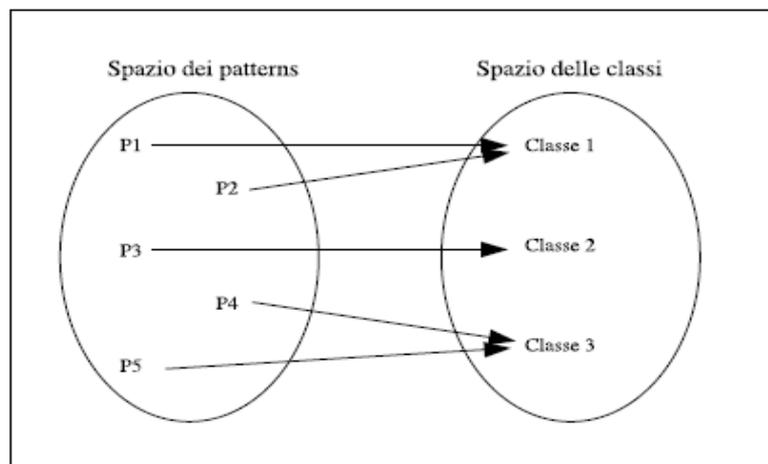
Anche se non esiste una procedura generale di *pre-processing* automatica, tale risultato può essere ottenuto costruendo una opportuna sequenza dedicata di passi che sono funzione delle caratteristiche delle immagini generate da un opportuno sensore e degli obiettivi che si

desidera centrare nelle fasi successive. Per fare ciò, possono essere adottate alcune operazioni che riguardano per esempio:

- **Applicazioni di opportuni filtri** per la riduzione del rumore, per il migliore sfruttamento della dinamica, ecc, che portano ad una migliore segmentazione dell'oggetto desiderato dall'immagine stessa;
- **Segmentazione dell'immagine** in modo da isolare i pattern dalla altre aree omogenee presenti sulla scena;
- **Binarizzazione dell'immagine segmentata** per isolare gli oggetti desiderati, ottenendo una sagoma degli stessi che consenta una caratterizzazione meno affetta da rumore.

Attraverso il blocco di “*estrazione delle caratteristiche*” (*features*) si ottiene una descrizione tramite primitive strutturali dell'immagine originale. In questo modo si riesce a concentrare l'informazione in pochi dati, ottenendo un maggiore potere discriminante ed una diminuzione dei dati utilizzati.

Selezionare delle caratteristiche (*features*) “ottime” è un passo cruciale, poiché il processo decisionale che classificherà gli oggetti si basa soltanto sulle informazioni, quindi sui valori, forniti da queste *features*. Le caratteristiche selezionate devono possedere il più possibile informazioni utili per permettere di discriminare un modello rispetto ad un'altro. Utilizzeremo il concetto di distanza *interclasse* e *intraclasse* [5].



**Figura 1.2** Mappatura dallo spazio dei pattern in quello dei significati.

In condizioni ideali, all'interno di una determinata classe, le *features* assumeranno sempre lo stesso valore (quindi con varianza zero). In realtà non sarà mai così: i valori oscillano a causa del campionamento dell'immagine, del rumore sempre presente nelle stesse e nelle sequenze video riprese da una telecamera, ecc.



Le features che verranno utilizzate nei nostri *riconoscitori automatici* dovranno avere:

- *una separazione interclasse molto grande*, il che significa che una feature in classi diverse deve assumere valori numerici molto distanti (quindi varianza molto grande). Due valori molto diversi di una feature rappresentano oggetti che hanno caratteristiche diverse fra di loro;
- *una distanza intraclasse piccola*, che significa che una feature all'interno della stessa classe deve dare valori numerici molto vicini fra di loro (ovvero varianza molto piccola). Valori numerici vicini rappresentano oggetti con caratteristiche generali simili.

La capacità discriminativa di una determinata feature, che indichiamo con il valore  $\Delta$ , sarà proporzionale alla distanza fra le medie di ogni feature in classi distinte (distanza interclasse), e inversamente proporzionale alla varianza dei valori assunti da ogni feature all'interno della stessa classe (distanza intraclasse). Il grado di discriminazione può essere quindi espresso da:

$$\Delta_i = \frac{Var_j(E_i \mathcal{F}_{i_k})}{E_j(Var_i \mathcal{F}_{i_k})} \quad (1.1)$$

dove  $F_{ik}$  rappresenta il  $k$ -mo campione del momento  $i$ -mo,  $E$  e  $Var$  denotano, rispettivamente, gli operatori di media e varianza. Questa quantità viene calcolata su tutte le features, fino ad un certo valore massimo. Le features che forniranno i valori più alti di  $\Delta_i$  saranno utilizzati dal riconoscitore.

Bisogna anche dire che questo valore non sempre rispecchia fedelmente la capacità discriminativa assoluta di una determinata feature, in quanto, per alcune coppie di classi, può assumere un valore molto alto, mascherando il contributo riportato da altre coppie di classi. Per evitare ciò, l'equazione (1.1) verrà calcolata per tutte le coppie di classi distinte, e le migliori features che scaturiranno da ogni valutazione andranno a far parte del *riconoscitore*.

Il blocco di “*classificazione*” è l'ultima fase del processo di riconoscimento, dove viene associata una classe ad una immagine sconosciuta. La classificazione viene eseguita esaminando le features estratte e confrontandole con quelle di un modello definito durante la fase di training (preso come riferimento) in modo da giungere all'attribuzione della componente in esame, come appartenente ad una delle classi possibili.

Ma per fare questo, bisogna definire un criterio di somiglianza che valuti quanto dista un'immagine rappresentata da un vettore di features ad  $n$  dimensioni  $\mathbf{X} = [x_1, x_2, \dots, x_n]$  da tutti quelli appartenenti ad un training set. Il training set sarà costituito da vettori campioni, ognuno dei quali rappresenta un determinato pattern.

Il classificatore utilizzato è chiamato *nearest-neighbour rule*. Quando un pattern  $\mathbf{X}$  deve essere classificato, viene ricercato l'elemento del training set più vicino a  $\mathbf{X}$ , e il pattern viene classificato appartenente alla classe dell'elemento trovato. La distanza tra  $\mathbf{X}$  ed i componenti del training set viene misurata mediante la distanza Euclidea. Comunque, poiché le componenti del vettore di features hanno range di valori di ordine molto variabile, è possibile che un sottogruppo di componenti prevalga sugli altri perché hanno valori molto grandi. Per evitare ciò e per bilanciare il peso della distanza per ogni componente del vettore di features, queste vengono normalizzate. La normalizzazione consiste nel sottrarre la media e dividere per la deviazione standard della corrispondente classe.

Sia  $d(\mathbf{X}, t_k^{(i)})$  la distanza tra l'immagine da classificare, rappresentata dal vettore di features  $\mathbf{X}$  e  $t_k^{(i)} = [t_1^{(i)}, t_2^{(i)}, \dots, t_n^{(i)}]$  che denota il  $k$ -esimo vettore di training della  $i$ -esima classe. Si ha:

$$d(\mathbf{X}, t_k^{(i)}) = \sqrt{\sum_{m=1}^n (\tilde{x}_m - \tilde{t}_{k_m}^{(i)})^2} \quad (1.2)$$

$$\tilde{x}_m = \frac{x_m - \bar{t}_m^{(i)}}{\sigma_{t_m}^{(i)}} \quad \tilde{t}_{k_m} = \frac{t_{k_m} - \bar{t}_m^{(i)}}{\sigma_{t_m}^{(i)}}$$

con  $\bar{t}_m^{(i)}$  e  $\sigma_{t_m}^{(i)}$  rappresentanti la media e la deviazione standard del  $m$ -mo elemento del vettore di features ad  $n$  dimensioni di classe  $i$ . L'equazione può essere semplificata, ottenendo:

$$d(\mathbf{X}, t_k^{(i)}) = \sqrt{\sum_{m=1}^n \frac{(x_m - t_{k_m}^{(i)})^2}{(\sigma_{t_m}^{(i)})^2}} \quad (1.3)$$

$\mathbf{X}$  viene classificato appartenente alla classe  $i^*$ , dove,

$$i^* = \min\{d(\mathbf{X}, t_k^{(i)}), \quad i = 1, 2, \dots, c\}$$

dove  $c$  è il numero totale di classi.

Come si può facilmente intuire, il blocco di estrazione delle componenti descrittive (“features”) rappresenta il cuore dell'intero sistema [5], in quanto una buona descrizione dell'immagine permette una più semplice classificazione della stessa.

### 1.3 Approccio Statistico e Strutturale - Sintattico

L'obiettivo di un sistema di riconoscimento automatico per l'analisi di immagini è fornire una descrizione plausibile di una scena del mondo reale in tempi utili a prendere una decisione. Le immagini permeano il nostro mondo percettivo-sperimentale, spesso sono confuse ed indeterminate, e la ragione di tale confusione può essere legata alla natura casuale dei dati, ai criteri di rappresentazione utilizzati, o dalla scarsa cognizione del contesto conoscitivo in cui le immagini sono prodotte.

Il riconoscimento di forme comporta la capacità, da parte di chi lo esercita, di attribuire un significato ad un insieme di informazioni, che si presume abbiano un qualche tipo di organizzazione interna (una determinata struttura). La descrizione o il riconoscimento di tali forme può condurre a delle decisioni circa azioni successive da intraprendere. Per riconoscere quindi, da un lato è necessario aver fatto esperienza su di un certo numero di situazioni che si è imparato a classificare, dall'altro in presenza di un qualcosa che non sia direttamente riconducibile ad un'esperienza nota, è indispensabile fare delle ipotesi o ricevere dei suggerimenti che orientino su certe scelte piuttosto che su altre.

I quattro principali approcci al pattern recognition [1] sono: il Template Matching (approcci basati su modello), lo Statistical Classification (approcci con classificazione statistica), il Syntactic or Structural Matching (approcci sintattici o strutturali) e il Neural Networks (approcci basati su reti neurali). La tabella 1.2 sintetizza ciascuno di questi 4 approcci con le rispettive caratteristiche.

Approccio	Rappresentazione	Riconoscimento	Criterio di classificazione.
Basato su modello (Template Matching)	Dati campionati, pixel, curve introduzione di distanze.	Correlazione Statistica.	Basato su stima dell'errore di classificazione.
Classificazione statistica ( <i>Statistical Classification</i> )	Misure o caratteristiche (Features).	Funzioni Discriminanti.	Basato su stima dell'errore di classificazione.
Sintattico o strutturale ( <i>Syntactic or Structural Recognition</i> )	Primitive.	Regole, grammatiche.	Basato su stima dell'errore di accettazione.
Reti Neurali ( <i>Neural Network</i> )	Dati campionati, pixel, Caratteristiche (Features).	Funzioni sinaptiche e di reti.	Basato su stima dell'errore quadratico medio.

**Tabella 1.2** Approcci e modelli per i sistemi di Pattern Recognition.

L'approccio più immediato al "pattern recognition" è il *template matching*". Il pattern da classificare è confrontato con un insieme di

prototipi (templates), ognuno rappresentativo di una delle classi di interesse. La classificazione è basata su un opportuno criterio di “*best matching*”, in cui si definisce una funzione  $F$  di valutazione del matching, che assume valore massimo per il miglior accoppiamento esemplare-prototipo. Tale approccio ha fornito risultati apprezzabili quando il numero delle classi non è grande, e gli esemplari appartenenti ad una stessa classe presentano una ridotta variabilità tra loro.

I modelli basati su reti neurali cercano di utilizzare principi di organizzazione dei dati, come: addestramento, generalizzazione, tolleranza ai guasti e distribuzione dei dati, in una rete di grafi pesati.

Approcci più sofisticati al problema sono quello statistico (di tipo parametrico o non parametrico) [6], [7] e quello strutturale o sintattico. Nei successivi paragrafi verrà fatta una comparazione tra l'approccio di tipo statistico e quello di tipo strutturale per evidenziarne le analogie e differenze.

### 1.3.1 Pattern Recognition Statistico.

Secondo l'approccio statistico, un pattern viene caratterizzato attraverso un insieme di misure eseguite su di esso (vettore delle features), e quindi rappresentato da un punto in un iperspazio delle features (figura 1.3).

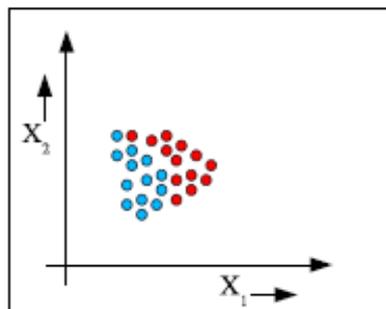


Figura 1.3. Rappresentazione delle features.

Il problema della classificazione può essere ricondotto a quello di suddividere lo spazio delle features in regioni, ognuna di pertinenza di una diversa classe (approccio non parametrico). Un esemplare è assegnato ad una classe se il punto che lo rappresenta cade nella regione di pertinenza di quella classe. Ovviamente per avere una rappresentazione ottimale del problema, gli oggetti di una stessa classe devono essere “*simili*” tra di loro ed il più possibile “*diversi*” da quelli appartenenti alle altre classi.

Alla base dell'approccio statistico vi è, infatti, l'idea che le forme di oggetti appartenenti alla stessa classe differiscano di poco tra loro, e per



poter valutare la “lontananza” tra le forme, è possibile fare delle misure delle loro caratteristiche.

In questa sezione la parola “*feature*” denota un'entità che è derivata da alcune misure iniziali. Watanabe, insieme ad altri studiosi ha utilizzato un metodo di valutazione che usa uno spazio detto delle configurazioni o delle misure, in cui un oggetto esaminato è descritto attraverso un opportuno vettore, detto “*feature vector*” di variabili aleatorie associate ai risultati di un insieme di misure effettuate sull'oggetto stesso. Il pattern rappresentante, l'oggetto è, dunque un insieme di misure eseguite sull'immagine. L'aspetto importante, consiste nello scegliere un insieme di misure adeguato, per le quali due punti poco distanti corrispondano a due oggetti con caratteristiche molto simili. Benché lo sembri, la valutazione delle similitudini tra forme non è un problema banale ed il concetto di distanza tra forme coinvolge la variazione di molti fattori. Nell'approccio statistico ogni pattern è rappresentato in termini di  $d$  features o caratteristiche ed è visto come un punto in uno spazio  $d$ -dimensionale. L'efficacia dello spazio di rappresentazione (feature set) è determinato da quanto bene pattern provenienti da classi differenti possono essere distinti.

Dato un set di pattern di prova per ogni classe, l'obiettivo è quello di stabilire dei contorni di decisione nello spazio delle features con cui separare pattern appartenenti a classi differenti. I contorni sono determinati dalla distribuzione di probabilità dei pattern appartenenti ad ogni classe [8].

Detto  $V$  il vettore delle caratteristiche e  $D$  la sua cardinalità, un pattern può essere rappresentato da un punto in uno spazio  $S$  a  $D$  dimensioni: se si dispone di un insieme sufficientemente ampio di pattern campioni appartenenti a tutte le possibili classi (training set) e per ognuno di questi è nota la classe di appartenenza, è possibile definire in  $S$  delle regioni di pertinenza delle singole classi.

Detto  $X$  il vettore delle caratteristiche, ottenuto dalle misure effettuate su un pattern incognito  $P$ , è possibile calcolare la distanza:

$$D_i = |X - C_i| \quad \text{con } i = 1, \dots, n \quad (1.4)$$

Dove le  $C_i$  sono dei vettori modello che individuano le varie classi. La regola di decisione ottimale consiste nell'assegnare il pattern alla classe  $i$ -esima se risulta verificata la seguente condizione:

$$D_i < D_j \quad \forall j \neq i \quad (1.5)$$

Nell'approccio di tipo statistico, il sistema di riconoscimento è realizzato mettendo insieme due fasi: una di training/learning ed una di classification/testing (figura 1.4).

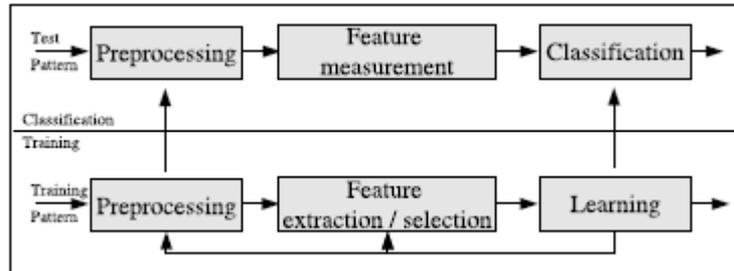


Figura 1.4. Esempio di Pattern Recognition di tipo Statistico.

Nella modalità *training* si selezionano le caratteristiche (*features*) adeguate per rappresentare i pattern di input ed il classificatore è addestrato a partizionare lo spazio delle caratteristiche (*features*). Nella modalità di classificazione il classificatore assegna i pattern ad una delle classi possibili sulla base di valutazioni di misura effettuate sulle *features*.

### 1.3.2 Pattern Recognition Strutturale

Nell'analisi di configurazioni complesse una soluzione ottimale può essere quella di considerare una prospettiva gerarchica al problema [9], [10], dove i pattern vengono visti come insiemi di componenti elementari. Il metodo di riconoscimento di tipo sintattico permette di semplificare la struttura del pattern [11]. Un *pattern* complesso è scomposto in un insieme di *subpatterns* più semplici, i quali a loro volta possono essere suddivisi in sotto-elementi.

La componente più piccola che si riesce ad individuare, all'interno di un pattern complesso è definita "*primitiva strutturale*".

Le primitive strutturali che compongono un pattern complesso sono legate tra di loro da relazioni; occorre quindi, individuare delle regole che riescano a rappresentarle e che ne diano una descrizione. L'approccio strutturale, vede, infatti le figure come costituite da un insieme complesso di primitive più semplici, e le relazioni tra di esse sono definite usando delle regole sintattiche o morfologiche.

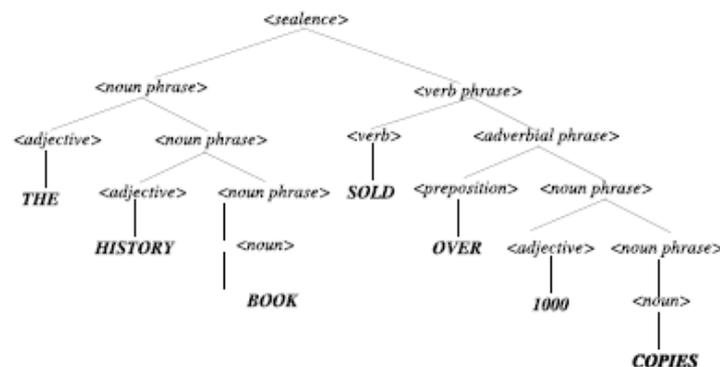
Per costruire una classificazione di questo tipo, si devono possedere tutte le tecniche necessarie per il riconoscimento e l'estrazione di componenti da un oggetto. Rispetto alle *features* di tipo statistico, le primitive strutturali costituiscono degli oggetti di più alto livello. Nell'approccio di tipo strutturale (e/o sintattico) si dà, infatti, maggiore rilievo al problema della scelta delle *features*; *features* significative dal punto di vista percettivo, geometrico o morfologico, che rappresentano componenti di

un pattern, sono ritenute più affidabili. Il principale obiettivo è di giungere a descrizioni efficaci ed efficienti.

Le features estratte utilizzando l'approccio di tipo sintattico possono essere usate all'interno di vari schemi di classificazione; dal template matching ai classificatori statistici. Esse si rivelano particolarmente adatte a conseguire descrizioni dei pattern complete, evidenziando, sia le primitive di cui il pattern si compone, sia il modo in cui esse sono correlate tra loro a formare strutture più complesse. Negli anni ottanta, Fu [10] introdusse la nozione di grammatica con attributi che unifica i concetti di *pattern recognition* sintattico con quello basato su classificazione statistica.

L'approccio strutturale consente di descrivere un ampio insieme di pattern complessi, usando un piccolo insieme di pattern assunti a primitive, e di regole di composizione per esse. L'approccio strutturale coinvolge l'insieme di tre processi indipendenti:

1. estrazione ed identificazione delle primitive;
2. identificazione delle relazioni esistenti tra le primitive;
3. identificazione di strutture valide in termini di primitive e di regole tra di esse.



**Figura 1.5.** Esempio di grammatica della lingua inglese.

Una analogia potrebbe essere fatta tra l'approccio sintattico in pattern recognition e la sintassi del linguaggio. I pattern sono visti come frasi di un linguaggio, mentre le primitive vengono rappresentate come l'alfabeto del linguaggio e le frasi quindi sono generate a partire da una grammatica per questo linguaggio. Un "pattern" deve rispettare la grammatica di una determinata classe per appartenerele.

Così un vasto insieme di pattern complessi può essere descritto con un piccolo numero di primitive e regole grammaticali (figura 1.5).



Un linguaggio può avere diversi tipi di rappresentazioni: quello che interessa, comunque, è la possibilità di risalire ad una delle sue categorie, a partire dalle componenti elementari ricavate. Nel metodo sintattico (e/o strutturale) il riconoscimento di un esemplare come appartenente ad una tra tutte le classi possibili può essere ricondotto, quindi, al riconoscimento di una frase come appartenente ad uno tra un certo numero di linguaggi definiti. Il problema si può risolvere utilizzando una grammatica per la rappresentazione del linguaggio che, tramite la composizione delle primitive, permetta la costruzione di una rappresentazione formale della configurazione iniziale. La fase di apprendimento del sistema di riconoscimento coincide con la determinazione di tale grammatica.

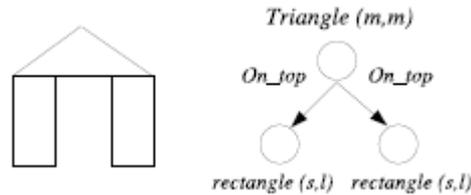
Nell'approccio di tipo strutturale sintattico esistono diverse tecniche per individuare ed estrarre le primitive (caratteristiche) elementari che costituiscono il pattern, tuttavia la maggior parte degli studi sono stati indirizzati alle rappresentazioni derivanti dalla teoria dei grafi. In particolare sono state sviluppate rappresentazioni in termini di alberi, di grafi e di ipergrafi, che consentono di rappresentare in maniera semplice una vasta classe di relazioni. Tra le rappresentazioni di questo genere una delle più utilizzate è costituita dai grafi relazionali con attributi (A. R. G.: Attributed Relational Graphs) [12], in cui ai nodi e agli archi di un grafo sono associate informazioni che caratterizzano rispettivamente le primitive e le relazioni esistenti tra di esse.

### 1.3.3 Rappresentazione in termini di ARG

Il metodo basato sui grafici relazionali con attributi consiste nell'ottenere un insieme di prototipi generali, partendo da un set di figure ben identificato. Un grafico relazionale è composto da un insieme di nodi e di archi (rami), rappresentativi delle relazioni tra due nodi. Nel caso della descrizione di figure, gli attributi dei nodi e dei rami, rappresentano rispettivamente le proprietà delle componenti primitive delle figure e le relazioni tra di esse. Quindi, data una curva originale, la sua rappresentazione mediante ARG consiste, nell'individuare le primitive strutturali che la compongono e le relazioni ritenute significative esistenti tra di esse, e nel tradurle in attributi dell'ARG. Una volta individuate le primitive e le relazioni si costruisce un grafo ai cui nodi sono associate le informazioni che caratterizzano le primitive e ai cui rami sono associate le informazioni che caratterizzano le relazioni esistenti tra le stesse.

Un grafo relazionale con attributi può, quindi essere definito come una sestupla  $(N, E, A_N, A_E, \alpha_N, \alpha_E)$ , dove  $N$  ed  $E$  sono rispettivamente l'insieme dei nodi e l'insieme degli archi,  $A_N$ , ed  $A_E$ , sono gli insiemi degli attributi dei nodi e degli archi, mentre  $\alpha_N$  e  $\alpha_E$  sono le funzioni che associano ad ogni nodo o ad ogni arco l'attributo corrispondente. Lo

stesso tipo di nodo può essere utilizzato per rappresentare diverse parti, anche di forma diversa di uno stesso oggetto (figura 1.6).



**Figura 1.6.** Attributed Relational Graph (ARG).

Il problema del riconoscimento, consiste nell'attribuire un esemplare come appartenente ad una classe. L'algoritmo di apprendimento cerca di trovare una legge di corrispondenza tra nodi ed archi di due grafi, l'uno rappresentante una descrizione dell'esemplare e l'altro la descrizione di un prototipo di una classe. Nella pratica è molto difficile stabilire una perfetta corrispondenza tra un prototipo e la rappresentazione di una figura, in quanto l'insieme delle possibili figure è praticamente illimitato, mentre quello dei prototipi è un insieme finito. Bisogna quindi definire una metrica, che consenta di valutare le similitudini tra grafi in modo da poter stabilire quale dei prototipi è più vicino all'esemplare.

Il problema si può quindi risolvere ricorrendo ad una tecnica di “*matching*” tra grafi, tecnica ampiamente usata in pattern recognition ed in particolar modo per l'OCR. Esistono diversi tipi di algoritmi per ottenere il matching tra grafi, alcuni di tipo deterministico ed altri di tipo euristico. Per alcuni di essi si hanno dei tempi di elaborazione esponenziali, mentre per altri il tempo di risposta è del tipo  $O(N \log N)$ . Un tempo di risposta così conveniente è stato ottenuto realizzando un algoritmo di matching tra grafi che confronta le loro geometrie, come gli angoli (tra gli archi), le posizioni (espresse come coordinate  $(x,y)$ ) e le lunghezze [13].

Metodi per ottenere delle rappresentazioni strutturali più robuste, e degli algoritmi con un basso costo computazionale per il matching tra grafi, sono oggetto di una continua ricerca nel settore del “*pattern recognition*” e dell'OCR.

#### **1.4 Quale approccio: Statistico o Strutturale ?**

Come già esposto nei paragrafi precedenti, il *pattern recognition*, cerca di identificare un oggetto in base alle conoscenze relative ad un modello cui ricondurre l'oggetto in esame. Storicamente i due principali approcci al riconoscimento sono quello “*statistico*” e quello “*sintattico* o *strutturale*”, mentre una tecnica emergente è legata alle “*reti neurali*”.



Si ricorda che l'approccio statistico utilizza delle tecniche di classificazione, che sono intrinsecamente statistiche, poiché i vari pattern hanno delle distribuzioni controllate da leggi probabilistiche. Per valutare la lontananza tra forme (le loro differenze) si effettuano delle misure sulle loro caratteristiche. Ogni oggetto viene quindi descritto attraverso un opportuno vettore, detto "*Feature Vector*" di variabili aleatorie associate ai risultati di un insieme di misure effettuate sull'oggetto stesso. Il feature vector (vettore delle caratteristiche) è rappresentato da un punto  $P$  in uno spazio  $d$ -dimensionale le cui coordinate sono date dai valori assunti dalle  $d$  variabili aleatorie. Nel caso ideale, ad oggetti appartenenti alla stessa classe corrisponderanno feature vector identici, mentre a quelli appartenenti a classi diverse corrisponderanno feature vector differenti. Il processo di classificazione consiste nell'individuare la regione di appartenenza del feature vector associato al campione in esame, risolvendo eventualmente ambiguità generate dalla contemporanea appartenenza a più regioni di alcuni punti.

I modelli statistici sono decisamente più potenti. Essi mettono a disposizione strumenti estremamente sofisticati per procedere alla classificazione di un pattern una volta che questo sia stato opportunamente caratterizzato per mezzo di un vettore di features adeguatamente rappresentativo. Tuttavia questo tipo di algoritmo incontra notevoli difficoltà quando i pattern sono molto complessi od il numero di classi è elevato, ovvero quando è presente una larga variabilità tra membri di una stessa classe. In questo modo si avranno sempre più punti comuni a regioni di decisione diverse, generando una notevole ambiguità. Pertanto il metodo statistico diviene inadeguato quando la variabilità degli oggetti da rappresentare ed il loro numero crescono.

Il metodo di riconoscimento di tipo sintattico è indicato per configurazioni complesse. Un pattern complesso può essere scomposto in un insieme di sub-patterns più semplici, i quali a loro volta sono suddivisi in sottocomponenti. In tal modo ci si può avvicinare alla maniera di descrivere un oggetto di un essere umano. Il primo fondamentale vantaggio di una tale impostazione è che la ricerca di un oggetto complesso può essere ricondotto alla sensibilità percettiva degli esseri umani. Un secondo vantaggio è la possibilità di spostare il problema della modellizzazione della variabilità delle forme in ingresso a quella della variabilità dei descrittori primitivi utilizzati. Lo svantaggio è che può generare una esplosione di combinazioni da investigare per inferire dalle primitive e dalle loro relazioni la grammatica conseguente; per cui necessita di grandi insiemi di pattern per l'apprendimento e quindi grosse risorse di elaborazione dati.



## Capitolo 2

# Le Fasi della Descrizione.

In generale il processo di riconoscimento può essere scomposto in quattro fasi, come di seguito descritto:

1. **scanning** o **acquisizione**: localizzazione e digitalizzazione delle immagini da analizzare e interpretare. Solitamente questa fase è svolta dalle apparecchiature atte all'acquisizione delle immagini, tramite meccanismi che tendono ad impedire di fornire all'utente immagini distorte;
2. **pre-processing** o **pre-elaborazione**: rappresentazione dell'immagine digitalizzata in un formato più conveniente per le successive elaborazioni;
3. **feature extraction**: decomposizione dell'immagine in componenti primitive e descrizione in termini di caratteristiche strutturali, ritenute descrittive dell'immagine originale;
4. **classification**: confronto della descrizione, ottenuta tramite primitive semplici con descrizioni di riferimento allo scopo di attribuire l'immagine in esame ad una delle possibili classi.

La tesi di dottorato, non si pone come obiettivo la ricostruzione perfetta di una immagine digitale (“*Curve Fitting*”), bensì la descrizione della stessa in termini di componenti elementari, in modo da facilitare il successivo passo di riconoscimento degli oggetti in essa contenuti. La descrizione ottenuta deve essere tale che, attraverso le primitive strutturali individuate e le relazioni esistenti tra le stesse si riesca a determinare la classe di appartenenza dell'oggetto da riconoscere, mediante il confronto con i modelli di riferimento.

Consideriamo ad esempio il carattere “A”. L'obiettivo è quello di descriverlo come la decomposizione di tre primitive strutturali, e di creare le relazioni esistenti tra le stesse, come rappresentato nella figura 2.1.



**Figura 2.1.** Esempio di descrizione della lettera “A” in termini di primitive strutturali semplici. Il carattere è rappresentato in termini di tre “features”.

Un buon metodo di descrizione, deve possedere la capacità di conservare le informazioni utili per il riconoscimento dell'oggetto descritto ed essere capace di filtrare le informazioni non necessarie (in particolare il rumore introdotto nel processo di acquisizione), producendo una descrizione compatta e maneggevole che rappresenti in maniera simile oggetti che differiscono per dettagli non significativi [14].

Il punto di partenza è una adeguata rappresentazione dell'oggetto originale, che faciliti il processo di estrazione delle “features”. In particolare, si parte da una rappresentazione in termini di segmenti di retta connessi a formare una “poligonale” dell'immagine originale. La rappresentazione poligonale deve, ovviamente essere libera da ogni fonte di rumore e deve contenere le sole informazioni necessarie.

Tale rappresentazione è ottenuta sottoponendo l'immagine da descrivere alle seguenti fasi:

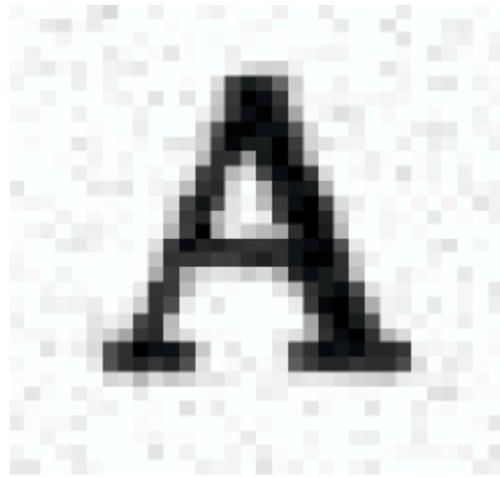
1. acquisizione;
2. filtraggio del rumore e sogliaatura;
3. thinning o assottigliamento dell'immagine;
4. labelling dello scheletro;
5. pruning;
6. approssimazione poligonale.

Nei paragrafi seguenti vogliamo dare una breve descrizione delle fasi elencate precedentemente, in modo che le stesse possano essere comprese meglio attraverso un semplice esempio, che ben illustra quello che succede durante le varie fasi del pre-processing e del processing di un sistema di *pattern recognition*. Queste ovviamente non sono tutte e le uniche fasi che vengono usate, ne esistono tante altre (soprattutto in fase di pre-elaborazione), che possono essere adottate per raggiungere l'obiettivo del riconoscimento automatico di un oggetto su un'immagine satellitare/aerea.

## 2.1 Acquisizione

Il primo componente di un sistema di riconoscimento è l'acquisizione dell'immagine da trattare. Essa è condotta utilizzando particolari dispositivi come scanner, sensori satellitari/aerei, ecc, e attraverso questa fase si ottiene una rappresentazione digitale dell'immagine, denominata *rappresentazione raster*.

Mediante un campionamento spaziale dell'immagine si ottiene una matrice numerica in cui ogni elemento rappresenta un pixel. La distanza (orizzontale e/o verticale) tra due pixel, definisce la risoluzione spaziale alla quale l'acquisizione è stata effettuata. Nella seguente figura 2.2 si riporta un esempio di immagine relativa a un carattere stampato.



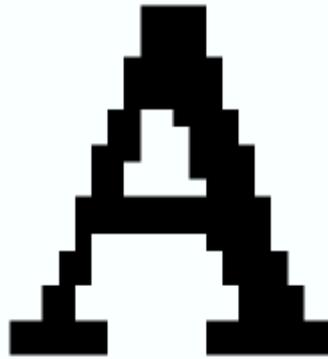
*Figura 2.2. Immagine “raster” di un carattere stampato, contenuto all'interno di un testo.*

## 2.2 Filtraggio del rumore e sogliatura

Le fasi successive a quella di acquisizione, lavorano su un'immagine in formato bitmap dell'immagine originale poiché nel processo di riconoscimento non è significativo il valore esatto della luminosità di un pixel. Quindi, la prima elaborazione a cui l'immagine è sottoposta è una binarizzazione, il cui scopo è quello di trasformare l'immagine in una matrice di bit (bitmap), dove ciascun pixel assume il valore 0 se appartiene allo sfondo e 1 se appartiene all'immagine. Per arrivare alla bitmap occorre, quindi effettuare una operazione di sogliatura (*thresholding*), che attribuisca ogni pixel come appartenente allo sfondo oppure all'immagine a seconda che superi o meno una soglia predefinita [15].

Sulla bitmap dell'immagine originale potrebbero essere presenti dei punti spuri (pixel dello sfondo con valore 1) o dei buchi (pixel

dell'immagine con valore 0), causati da fonti di rumore generate durante la precedente fase di acquisizione. Per ovviare a questi inconvenienti è necessario sottoporre la rappresentazione ottenuta, ad una operazione di *filtraggio*, compiuta attraverso filtri digitali. Nella figura 2.3 è riportata la bitmap dell'immagine mostrata nella precedente figura 2.2.



*Figura 2.3. Esempio di bitmap relativa ad un carattere stampato, ottenuta dalla rappresentazione "raster" dello stesso.*

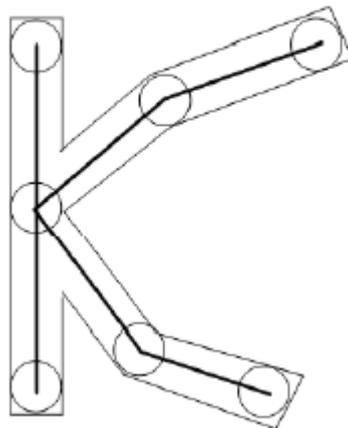
### 2.3 Thinning o assottigliamento dell'immagine

La "bitmap" dell'immagine ottenuta a valle delle fasi di *acquisizione*, di *filtraggio* e di *sogliatura* contiene una grande mole di informazioni, non tutte rilevanti al fine della descrizione e del successivo riconoscimento. Per ridurre la quantità di queste informazioni è possibile effettuare una operazione di assottigliamento, in modo da passare dalla bitmap a delle linee di spessore unitario.

Tale trasformazione è detta "*thinning*" o "*scheletrizzazione*". La figura 2.4 mostra il processo di assottigliamento (scheletrizzazione) della lettera "K".

Un algoritmo di thinning ottimale, deve permettere di ricostruire la struttura lineare dell'immagine senza perderne la connessione, consentendo una riduzione dei dati, senza perdita di informazioni basilari [16], [17], [18]. La fase di thinning si basa, infatti sull'ipotesi che lo spessore dei tratti non sia un'informazione necessaria al riconoscimento dell'oggetto. Ovviamente il thinning dell'immagine deve soddisfare alcuni requisiti fondamentali [19]:

- non deve dare origine a connessioni se la figura di partenza non era connessa;
- non deve operare fusioni tra parti non connesse.



*Figura 2.4. Scheletro dell'immagine di un carattere, ottenuto attraverso la fase di scheletrizzazione e/o "thinning"*

## 2.4 Pruning

La fase di thinning (o assottigliamento) potrebbe produrre, in alcuni casi uno scheletro che non riflette in maniera puntuale l'andamento dell'immagine. Lo scheletro potrebbe presentare delle distorsioni. La fase di "pruning" è necessaria per depurare lo scheletro da tutte queste imperfezioni, quali per esempio eventuali rami spuri che non trovano un corrispondente nell'immagine originale.

## 2.5 Labelling dello scheletro

L'approssimazione ottenuta a valle delle precedenti fasi, presenta però ancora un fondamentale difetto per poter essere considerata un buon punto di partenza per il riconoscimento. Essa infatti, è ancora una curva complessa e difficilmente gestibile. Il nostro obiettivo è quello di ottenere delle linee elementari da utilizzare nella fase di descrizione. Per poter suddividere la curva (complessa), ottenuta in un insieme di linee elementari, bisogna individuare dei punti notevoli rappresentanti i punti di incrocio, al fine di poter sbrogliare lo scheletro ed ottenere la poligonale su cui effettuare il processo di descrizione.

Si possono individuare cinque classi differenti per i punti appartenenti ad uno scheletro, come di seguito descritto [20] :

- "normal point (NP)": è un punto il cui cerchio di raggio massimo tocca i bordi dell'oggetto in due insiemi separati e continui di punti;
- "branch point (BP)": è un punto il cui cerchio di raggio massimo tocca i bordi dell'oggetto in tre o più insiemi separati e continui di punti;



- “*end point (EP)*”: è un punto il cui cerchio di raggio massimo tocca il bordo in un unico insieme continuo di punti;
- “*terminal point (TP)*”: è un punto con la stessa definizione di un NP, con il requisito aggiuntivo di essere adiacente ad un BP;
- “*adjacent branch point (ABP)*”: è un punto con la stessa definizione di un BP con il requisito aggiuntivo di essere vicino ad un altro BP.

## 2.6 Approssimazione poligonale

La rappresentazione mediante scheletro, consente di ridurre notevolmente l'enorme quantità di dati contenuti nell'immagine originale. Tale rappresentazione risulta notevolmente più sintetica rispetto alla bitmap, ma è ancora poco adatta ad essere trattata all'interno di un sistema di riconoscimento.

Una volta estratto lo scheletro dell'immagine, occorre compiere un altro passo per sfoltire ulteriormente il suo contenuto informativo, usando degli algoritmi di approssimazione poligonale. L'obiettivo di tale approssimazione è quello di rimuovere il rumore generatosi nelle precedenti fasi, comprimere i dati, e descrivere l'immagine in maniera opportuna (es. Spline ed Archi di Cerchio), mantenendone il contenuto informativo. La scelta delle primitive da utilizzare dovrebbe permettere di rappresentare in maniera compatta i tratti che compongono la poligonale. Ad esempio la lettera “A” dovrebbe poter essere rappresentata mediante tre strati rettilinei, mentre la lettera “B” dovrebbe essere descritta da un tratto rettilineo e due tratti curvilinei con concavità rivolta verso sinistra.

Prima di procedere alla descrizione dei passi che compongono tutto il processo di decomposizione, introduciamo alcune componenti che saranno utilizzate nel seguito di questo lavoro. Come già detto in precedenza, un possibile approccio o punto di partenza (ma non l'unico come vedremo nei capitoli successivi), potrebbe essere la rappresentazione poligonale dell'immagine digitale originale (quando la complessità del pattern da riconoscere lo consente), cioè in altre parole un insieme di segmenti di retta che uniscono due punti consecutivi.

Definiamo tali componenti “*CPL o Componenti di Primo Livello*”. Da tale rappresentazione si passa, poi ad una costituita da componenti dalle più ampie capacità descrittive, che verranno descritte in seguito, e che definiamo “*CSL o Componenti di secondo Livello*”.

Esistono numerosi algoritmi di approssimazione poligonale [19], che permettono di passare dalla descrizione tramite CPL a quella in termini



di CSL, classificabili in funzione della strategia di accorpamento dei punti che essi adottano.

Sono note dalla letteratura, varie metodologie di decomposizione basate su valutazione locale dell'errore di approssimazione, ed una basata su valutazione globale dello stesso. In particolare:

- “Merge”;
- “Split”;
- “Split and Merge”;

L'algoritmo di “*merge*” [21] procede in maniera “sequenziale. Partendo da un estremo della curva, assumendolo come primo vertice della poligonale, tenta di approssimare con un unico segmento (arco di cerchio o spline), il tratto di curva compreso tra il primo vertice ed il punto successivo. Punti consecutivi della curva vengono considerati fino a quando il criterio di approssimazione è soddisfatto. Il primo punto in corrispondenza del quale l'errore di approssimazione supera la soglia prefissata viene assunto come nuovo vertice e da esso riparte il procedimento di approssimazione, fino a quando non è stato esaminato l'ultimo punto della curva. Quando è stato raggiunto, quindi, l'altro estremo della poligonale, il procedimento di decomposizione termina.

L'algoritmo di “*split*” considera inizialmente l'intera curva, cercando di approssimarla con un unico segmento (arco di cerchio o spline). Nel caso in cui l'errore di approssimazione risulti maggiore di una soglia prefissata, la curva viene suddivisa in due parti in corrispondenza del punto che produce l'errore massimo. In tale punto si pone un nuovo vertice e si riconsidera l'approssimazione su ognuna delle parti ottenute. Il procedimento viene ripetuto fino a quando l'errore di approssimazione non risulta inferiore della soglia per tutti i tratti in cui è stata divisa la curva originale, ed essa non ha subito mutamenti nel precedente passo.

L'algoritmo di “*split and merge*” [22] combina i due algoritmi precedenti, alternando un passo di split ad uno di merge che fonde coppie di segmenti successivi generate dal precedente passo di split, ma che comunque soddisfano il criterio di approssimazione.

L'implementazione dei metodi appena descritti, richiede la definizione di un criterio di accettazione delle approssimazioni effettuate. Esistono numerose tecniche per la definizione del criterio di accettazione, alcune delle quali segmentano la curva nei punti di massima deviazione, altre in corrispondenza dei punti di massima curvatura, mentre altre utilizzano tecniche miste.

Purtroppo esse hanno l'inconveniente di utilizzare una soglia fissa, in questo modo la descrizione dell'immagine, cambia al variare delle



dimensioni e della conformazione della stessa. Alcuni algoritmi sopperiscono al fatto di avere una soglia fissa con una misura ottenuta normalizzando la stessa rispetto alla lunghezza totale del tratto approssimante [23], il che permette di ottenere l'invarianza dalla scala della soglia.

Vedremo nel seguito di questo lavoro, quanto importante e delicate siano la fasi di *pre-processing* e *descrizione* degli oggetti da riconoscere, poiché questo determina il successo o meno del *pattern recognition* progettato.



## Capitolo 3

# Pattern Recognition basati sul Modello: *Template Matching*.

Uno dei primi e più semplici approcci alla pattern recognition è quello basato sul modello, il cosiddetto *template matching*. I sistemi di *pattern recognition* basati sul modello utilizzano la cosiddetta tecnica del “matching”; il *matching* è definito come l’operazione che viene usata per determinare la *similarità* tra due entità dello stesso tipo: punti, curve o forme ad esempio.

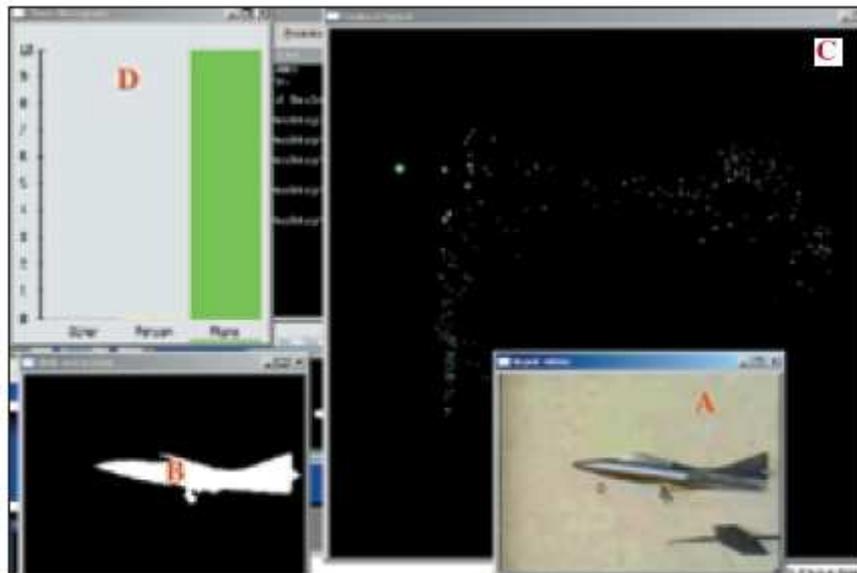
Nell’approccio basato su modello, quest’ultimo (appunto il modello) è tipicamente un prototipo del *pattern* od oggetto da riconoscere e classificare, tipicamente una forma bidimensionale. L’oggetto da riconoscere (il *pattern*) viene confrontato con tutti i modelli memorizzati nel sistema considerando i cambiamenti di scala e le possibili invarianze per rotazione o traslazione. Naturalmente ci saranno casi in cui sarà utile l’invarianza di scala, cioè ci interessa riconoscere l’oggetto semplicemente dalla forma, mentre in altri casi sarà necessario distinguere non solo rispetto alla forma ma anche rispetto alla dimensione. Nella figura 3.1 è riportato un esempio di sistema basato su modello dove si mostra un *pattern*, l’aereo in input (A), e l’estrazione della forma (B). Nella stessa figura è presente anche una finestra sullo sfondo (C) con dei punti rappresentanti i valori delle caratteristiche per i modelli di aereo presenti nella base dati. I punti di dimensione più grande rappresentano i modelli che sono più simili all’aereo in input (risultato della fase di *matching*). L’istogramma di colore verde (D), mostra come l’oggetto sia classificato “aereo” dal sistema, perché la maggior parte (tutti in questo caso) dei punti della base dati di oggetti nel suo intorno erano stati classificati (in fase di addestramento) come aerei (colore verde).

La misura di similarità utilizzata nei sistemi basati su modello, è solitamente una correlazione statistica e può essere ottimizzata apprendendo i parametri dal *training set*, ovvero da un insieme di esempi di oggetti forniti in input al sistema. I sistemi basati su modello sono computazionalmente inefficienti, richiedono molte risorse di calcolo per confrontare il modello del pattern in input con tutti gli altri

presenti nella base dati, ma grazie alla continua crescita della potenza di calcolo questo approccio viene sempre più utilizzato soprattutto in ambito industriale.

Per come abbiamo definito il modello fino ad ora si dovrebbe parlare di sistemi basati su modelli rigidi, ovvero sistemi dove il modello è descritto con parametri ed insiemi di valori fissati. Questo tipo di sistemi tende a non essere efficiente in presenza di distorsioni dovute all'acquisizione dell'oggetto (per esempio: effetti di errori sulle immagini), cambiamento dei punti di vista, o grosse variazioni di valori delle caratteristiche all'interno della stessa classe (*inter-class variance*).

In questi ultimi casi si tende ad utilizzare tecniche di modelli deformabili o elastici [24], al fine di consentire un confronto di similarità (*matching*) tra modelli le cui deformazioni non possono essere rappresentate esplicitamente a priori. Per intenderci, questi sistemi sono utili quando non possiamo modellare a priori tutte le possibili deformazioni che un oggetto possa avere, ed allora si agisce creando un modello dell'oggetto e studiando quali agenti esterni lo possono deformare (modellando quindi le possibili deformazioni) a volte simulando anche forze fisiche.



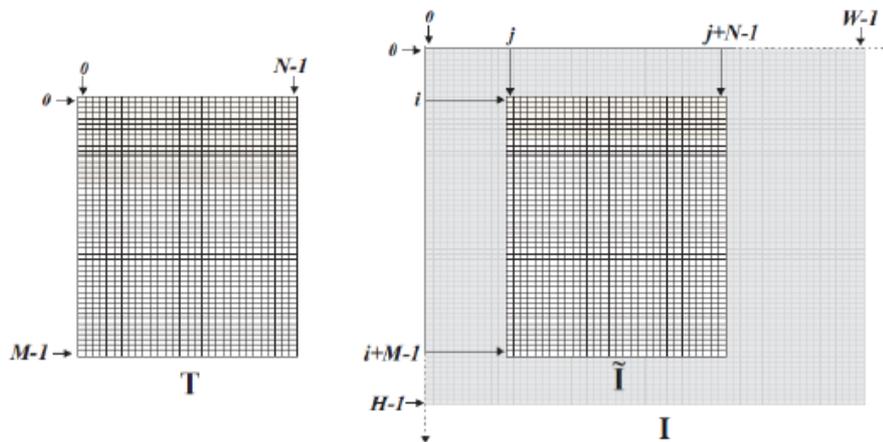
**Figura 3.1.** Esempio di sistema di pattern recognition basato su modello: (A) pattern in input (aereo), (B) estrazione forma, (C) base dati di addestramento (training set), (D) istogramma di classificazione.

### 3.1 Template Matching

Il *Template Matching* è una tecnica molto potente, che permette di identificare ed individuare la posizione di una porzione di immagine (*pattern*) all'interno dell'immagine stessa. Questa tecnica è molto diffusa nell'ambito della *computer vision*, poiché permette di individuare posizione, rotazione e fattore di scala di un oggetto o una parte di esso, all'interno dell'immagine in cui questo è raffigurato. Il fattore di scala e la rotazione individuano rispettivamente eventuali variazioni della distanza tra la telecamera e target e possibili rotazioni relative tra telecamera e oggetto.

Nelle analisi effettuate si è ipotizzato di misurare un moto bidimensionale puramente traslatorio del target, situato in un piano perpendicolare all'asse ottico della telecamera. Le ipotesi introdotte, permettono di tracciare la posizione del target individuando la sola posizione del pattern all'interno dell'immagine.

La posizione del pattern può essere determinata mediante diversi algoritmi, e quelli più diffusi sono la *Somma delle Differenze al Quadrato* (Sum of Squared Differences, SSD), la *Somma delle Differenze Assolute* (Sum of Absolute Differences, SAD), e la *Cross-Correlazione Normalizzata* (Normalised Cross-Correlation, NCC). Tutti e tre gli algoritmi sono caratterizzati da una prima fase comune, che consiste nel far scorrere il pattern sull'immagine da analizzare al fine di stimare, per ogni posizione (figura 3.2), la *similarità* fra pattern e porzione dell'immagine sotto analisi.



**Figura 3.2.** Rappresentazione del pattern desiderato (sinistra, dimensioni  $N \times M$ ) e dell'immagine originale, sotto analisi, dove lo stesso viene cercato (destra).

Gli algoritmi si differenziano nella scelta della funzione utilizzata per la stima della *similarità*, cioè della grandezza che misura quantitativamente l'*uguaglianza* tra il *pattern* da riconoscere e l'immagine originale [25].



Definiamo ora brevemente queste grandezze, per passare successivamente alle descrizioni degli algoritmi di TM sviluppati nelle applicazioni affrontate in questa sede, evidenziando di volta in volta pregi e difetti della tecnica di *pattern recognition*.

### 3.1.1 Somma delle Differenze Quadrate (SSD)

Interpretando  $\tilde{I}(i, j)$  (la sottoimmagine associata alla posizione  $(i, j)$ ) ed avente le stesse dimensioni di  $T$ ) e  $T$  come vettori di uno spazio a  $M \times N$  dimensioni, la *SSD* rappresenta il quadrato della norma  $L2$  (norma euclidea) della differenza dei due vettori.

$$SSD(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I(i+m, j+n) - T(m, n))^2 \quad (3.1)$$

### 3.1.2 Somma delle Differenze Assolute (SAD)

Interpretando  $\tilde{I}(i, j)$  e  $T$  come vettori, la *SAD* rappresenta la norma  $L1$  della differenza dei due vettori. Se il pattern rispecchia fedelmente una porzione di immagine, vi sarà una posizione  $(i, j)$  in cui sia la *SAD* che la *SSD* sono nulli. Se invece il pattern non viene estratto dall'immagine in analisi, non vi sarà possibilità di annullare i due parametri poiché nessuna zona dell'immagine sarà esattamente identica al pattern per via del rumore di acquisizione; in tal caso si sceglie come posizione di somiglianza tra pattern e immagine, quella che presenta un valore *SAD* o *SSD* più basso.

$$SAD(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |I(i+m, j+n) - T(m, n)| \quad (3.2)$$

### 3.1.3 Cross-Correlation Normalizzata (NCC)

Interpretando  $\tilde{I}(i, j)$  e  $T$  come vettori, la *NCC* rappresenta il coseno dell'angolo fra  $\tilde{I}(i, j)$  e  $T$ :

$$NCC(i, j) = \frac{\tilde{I}(i, j) \cdot T}{|\tilde{I}(i, j)| \cdot |T|} = \frac{|\tilde{I}(i, j)| \cdot |T| \cdot \cos \theta}{|\tilde{I}(i, j)| \cdot |T|} = \cos \theta \quad (3.3)$$

ed è quindi dipendente dalla sola direzione dei due vettori, assumendo il valore massimo (pari a 1) quando i due vettori sono paralleli ( $\tilde{I}(i, j)=T$ ).

Rispetto alla *SSD* e alla *SAD*, la *NCC* costituisce una misura di similarità più robusta rispetto a possibili variazioni fotometriche dell'immagine. Infatti, a differenza della *SSD* e della *SAD*, la *NCC* risulta insensibile a variazioni proporzionali dell'intensità luminosa dell'immagine (ad esempio, dovute a variazioni nelle condizioni di illuminazione dell'oggetto di interesse):

$$\tilde{I}(i, j) = \alpha \cdot T$$

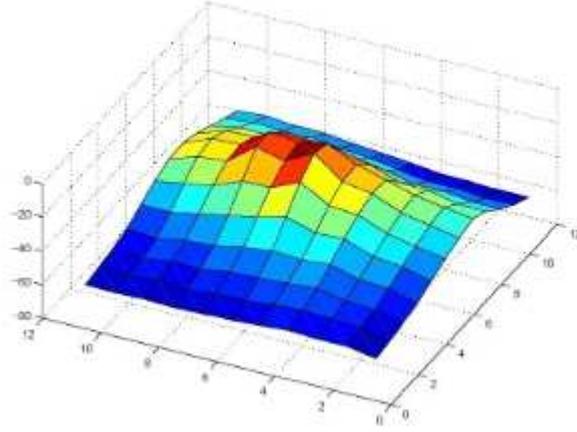


Figura 3.3. Esempio di superficie di correlazione normalizzata.

### 3.1.4 Cross-Correlation Normalizzata a Media Nulla (ZNCC)

$$ZNCC(i, j) = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [I(i+m, j+n) - \mu(\tilde{I})] \cdot [T(m, n) - \mu(T)]}{\sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [I(i+m, j+n) - \mu(\tilde{I})]^2} \cdot \sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [T(m, n) - \mu(T)]^2}} \quad (3.4)$$

dove  $\mu(\tilde{I})$  e  $\mu(T)$  rappresentano rispettivamente il valore medio dell'intensità della sottoimmagine  $\tilde{I}$  associata alla posizione  $(i, j)$  e del template  $T$ :

$$\mu(\tilde{I}) = \frac{\sum_{m=0}^{M-1} \sum_{j=0}^{N-1} I(i+m, j+n)}{M \cdot N} \quad \mu(T) = \frac{\sum_{m=0}^{M-1} \sum_{j=0}^{N-1} T(m, n)}{M \cdot N}$$

Anche la ZNCC costituisce una misura di similarità molto robusta rispetto a possibili variazioni fotometriche dell'immagine. Difatti, la ZNCC risulta insensibile a variazioni lineari dell'intensità luminosa dell'immagine:

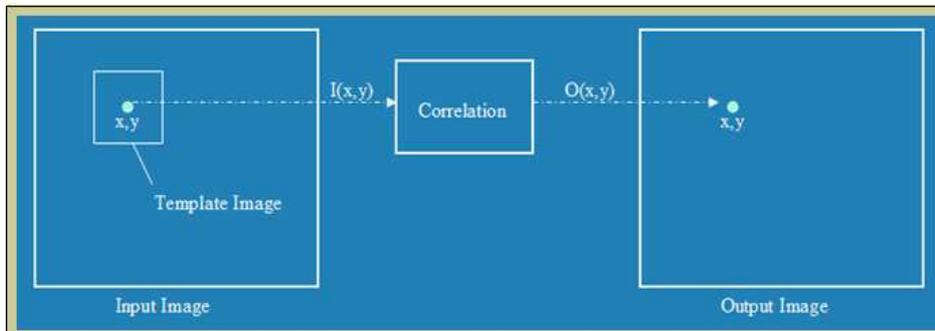
$$\tilde{I}(i, j) = \alpha \cdot T + \beta$$

### 3.2 Ottimizzazione della Tecnica

Un problema associato all'utilizzo di questi algoritmi, per il posizionamento del pattern all'interno dell'immagine, è l'onerosità computazionale che questi comportano. La complessità computazionale di un algoritmo di *template matching* (figura 3.4) è pari al prodotto delle dimensioni dell'immagine e del pattern:

$$\text{Noperazioni} = (H * W) * (M * N) \quad (3.5)$$

Di conseguenza, quando queste sono significative (ad esempio immagine 1920\*1088 pixel e pattern 256\*256 pixel) il costo computazionale diviene enorme. E' quindi necessario individuare dei metodi che consentano di velocizzare l'esecuzione e renderla compatibile con i vincoli sui tempi di elaborazione eventualmente imposti dall'applicazione. Fra i metodi proposti ricordiamo:



**Figura 3.4.** Rappresentazione del pattern (piccolo quadrato a sinistra), dell'immagine d'input e dell'immagine "correlazione" d'output dove cerchiamo i massimi per identificare l'oggetto desiderato.

- **Ricerca Multi-resolution:** si cerca prima in una immagine a risoluzione ridotta e poi si raffina la ricerca alla risoluzione piena (è possibile anche lavorare con più di due livelli di risoluzione);
- **Ricerca basata su Sub-template:** Si fa una prima ricerca con un sub-template e poi si fa la ricerca con l'intero pattern solo nei punti che presentano un buon match con il *sub-template*;
- **SAD-SSD con opzione di Call-out:** è possibile interrompere il calcolo della SAD (o della SSD) quando il suo valore è maggiore del minimo corrente;
- **Uso di soglie calcolabili in modo efficiente:** ad esempio per la SAD è possibile sfruttare la relazione:

$$SAD(i, j) \geq \left| \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n) - \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} T(m, n) \right|$$



La scelta del pattern da utilizzare è fondamentale per ottimizzare la ricerca ed ottenere dei buoni risultati di posizionamento.

### **3.3 Sistema di Pattern Recognition basato sul *Template Matching***

Alla luce di quanto esposto nei paragrafi precedenti, in questa sezione vogliamo presentare il sistema di *pattern recognition* basato sulla tecnica del *template matching*, sviluppato durante gli anni di dottorato per affrontare vari tipi di problemi legati alla sicurezza.

La sicurezza dei confini e delle popolazioni è un argomento molto complesso e sensibile in seno alle istituzioni europee (e non solo), come dimostrano alcuni eventi che, negli ultimi anni, hanno cambiato radicalmente la nostra percezione del mondo. Proprio per questo la Commissione Europea si mostra particolarmente attenta a queste problematiche, e cerca in tutti i modi di incentivare la ricerca scientifica e tecnologica in questi campi. In particolare, investendo enormi risorse su progetti che mirano a raggiungere l'obiettivo della sicurezza e della sorveglianza dei nostri territori, mediante l'uso dei dati dei sistemi per l'osservazione della terra (EOS). E infatti, gran parte degli spunti che ci hanno motivato ad occuparci di questo tipo di problematiche, è arrivato grazie al coinvolgimento del CRPSM (centro presso il quale ho svolto la mia attività di dottorato), nei vari progetti finanziati dall'UE sulla sicurezza, sorveglianza e monitoraggio come GMOSS (Global Monitoring for Security and Stability), G-MOSAIC (GMES services for Management of Operations, Situation Awareness and Intelligence for regional Crises), ecc.

I sistemi EO, a causa dei limiti imposti dalla dinamica orbitale, non permettono un controllo continuo (a causa dello sfavorevole compromesso tra la risoluzione spaziale e quella temporale necessario) delle frontiere. Tale controllo può essere comunque esercitato attraverso un'analisi delle immagini telerilevate, sulla base di serie temporali di immagini acquisite con un periodo temporale determinato o regolato in funzione delle esigenze che di volta in volta possono sorgere. In generale, vista la loro estensione, in alcuni casi il controllo delle frontiere avente l'obiettivo di individuare nuove strutture artificiali o "tracce" che attestino e/o permettano la previsione di eventi particolari (flussi migratori, concentrazione di truppe, ecc) richiede il trattamento di mosaici d'immagini satellitari enormi (per esempio, il confine Libia-Chad per essere interamente coperto richiederebbe la costruzione di un mosaico di più di 40 immagini ASTER).

L'analisi di dati di enormi dimensioni (quali i mosaici d'immagini satellitari), nella ricerca di oggetti sconosciuti in termini di dimensione,



posizione e caratteristiche spettrali, non può essere affrontato mediante processi manuali di foto-interpretazione. Questo problema di *data mining* mette in evidenza la necessità di sviluppare nuove tecniche e strumenti per l'estrazione e la gestione delle informazioni contenute nelle immagini satellitari. Questo risulta tanto più vero alla luce della crescente disponibilità di dati EO usati per il rilevamento dei cambiamenti (*change detection*), il monitoraggio globale, il supporto alla gestione delle catastrofi, ecc.

Le difficoltà che si incontrano nel processo di sviluppo di una tecnica con elevato grado di automatizzazione, che consenta di ridurre la partecipazione umana all'elaborazione delle immagini, risiedono principalmente nella grande varietà di applicazioni che coinvolgono il trattamento di grandi quantità di dati. Così, in questo ambito, le applicazioni più complesse sono quelle in cui l'oggetto d'interesse cercato non è spettralmente ben definito, per il quale è quindi richiesta una rilevazione basata soprattutto sulle sue caratteristiche morfologiche (forma, dimensione, ecc). La soluzione ottima in generale, impone di considerare una combinazione di caratteristiche spettrali e morfologiche per individuare l'oggetto desiderato.

Questo problema ovviamente è tanto più grande quanto migliore è la risoluzione spaziale del sensore usato, poiché a parità di area è maggiore il numero di pixel da analizzare. Per esempio, la superficie coperta da una singola immagine Landsat (185 x 185 km) per essere interamente coperta richiede l'utilizzo di circa 340 immagini Ikonos (risoluzione spaziale di 1m e area coperta da singola immagine pari a 10x10 km) !!!

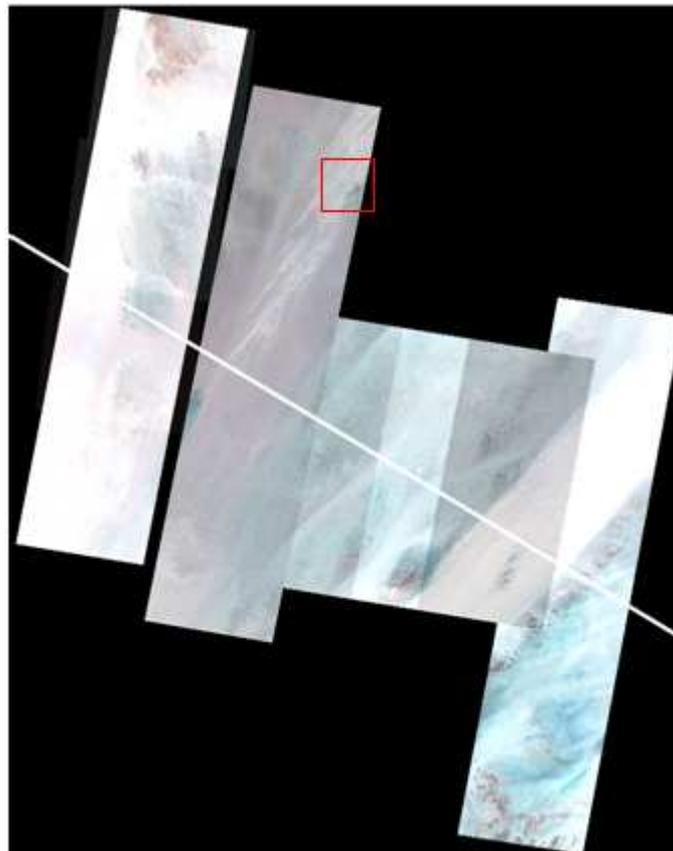
L'obiettivo principale in questo caso è quindi quello di creare un sistema di pattern recognition, basato sulla tecnica del template matching, in grado di processare immagini satellitari di grandi dimensioni (mosaici per esempio), per individuare efficacemente gli oggetti d'interesse. Gli algoritmi sviluppati qui, sono stati realizzati usando immagini satellitari del sensore ASTER, poiché questo presenta un buon compromesso tra la risoluzione spaziale e la dimensione dell'oggetto che si vuole individuare (infrastrutture quali strade, aeroporti, ecc.). Questo sensore fornisce immagini in 14 canali spettrali, ma poiché solo i primi 3 canali VIS/NIR hanno una risoluzione spaziale di 15m, adatta alle nostre esigenze, solo questi vengono utilizzati. Gli altri canali sono 6 nello SWIR (Short-Wave Infrared) con risoluzione spaziale di 30 m e 5 canali nel TIR (Thermal Infrared), con risoluzione spaziale di 90 m.

Due grandi mosaici di immagini ASTER sono stati creati per condurre il nostro studio, uno copre il confine Libia/Ciad e l'altro si riferisce all'area di frontiera Pakistan/India (regione del Kashmir). Nella figura 3.5 viene mostrata l'estensione della zona di interesse in Africa, e il

quadrato rosso indica la parte coperta dal mosaico ASTER creato per realizzare l'analisi della zona e riportata nella figura 3.6.



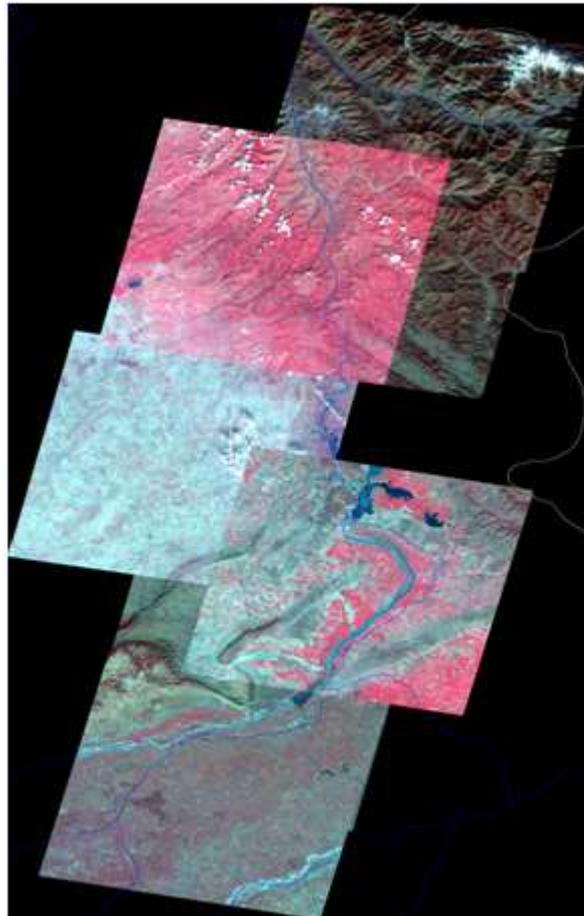
**Figura 3.5.** Confine Libia/Chad con l'aria di studio (quadrato rosso).  
Le tracce sovrainposte (linee rosse) corrispondono ai passaggi  
dei satelliti SPOT 4 e 5.



**Figura 3.6.** Uno dei mosaici ASTER relativo al confine Libia/Chad  
usati per svolgere lo studio. Questo è composto da 20 immagini.

Come accennato precedentemente, per coprire questa parte del confine sono state necessarie circa 40 immagini ASTER.

La figura 3.7 mostra una parte del mosaico ASTER coprente la zona del confine tra il Pakistan e l'India nella regione del Kashmir. Le differenze nelle caratteristiche paesaggistiche, rispetto al caso precedente, possono essere chiaramente osservate, e questo porta ad un approccio diverso in fase di pre-processing come vedremo in seguito (montagna e vegetazione folta contro un'area desertica).



**Figura 3.7.** Parte del mosaico d'immagini ASTER relative al confine Pakistan/India (regione del Kashmir) usato per condurre lo studio.

L'idea principale del nostro lavoro, così come già visto in altri capitoli usando altri approcci, è quello di costruire un sistema (basato sulla tecnica del *Template Matching* esposta precedentemente), per estrarre in modo automatico, le informazioni (in questo caso aeroporti) dalle immagini mosaico considerate, sfruttando le caratteristiche spettrali e morfologiche degli oggetti, in modo da poter ottenere i migliori risultati nelle condizioni date.



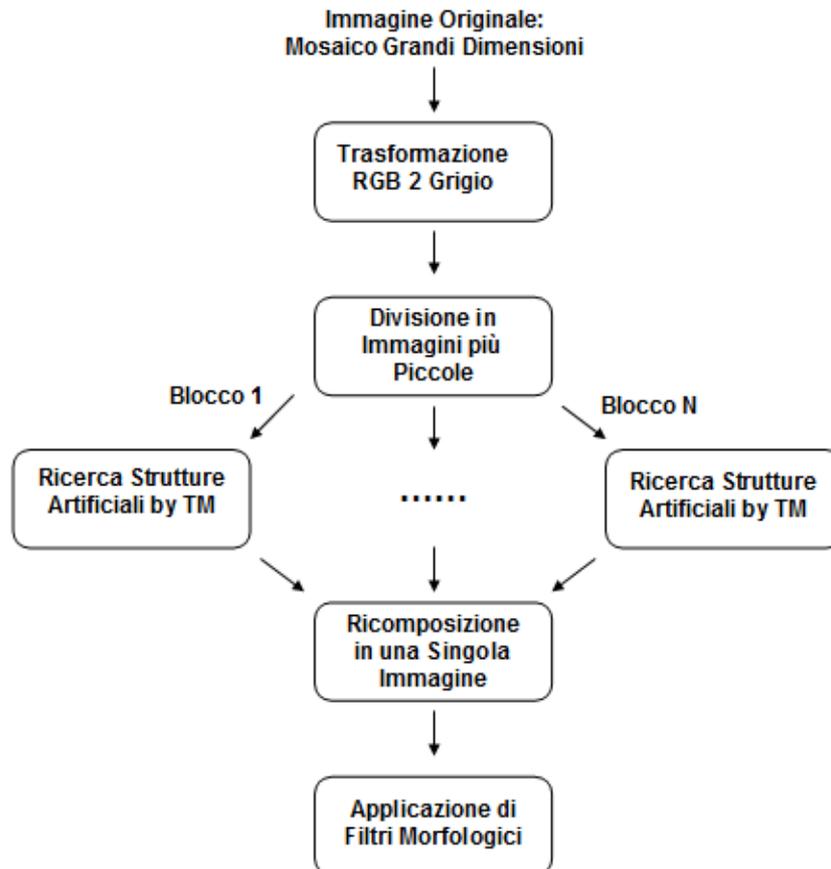
L'obiettivo descritto sopra viene ottenuto utilizzando un algoritmo comprende le seguenti fasi (vedi figura 3.8):

- Le immagini mosaico RGB vengono convertite in immagini in toni di grigio, poiché in questo modo si riesce ad applicare meglio la *template matching* (meno ambiguità in grado di generare errori di varia natura. L'immagine in toni di grigio può essere ottenuta attraverso varie vie:
  1. selezione della banda più adeguata (quella che evidenzia meglio gli oggetti cercati);
  2. applicazione di una tecnica di trasformazione chiamata PCA (Principal Component Analysis), la quale è in grado di sintetizzare il contenuto spettrale dell'immagine originale (con tutte le sue bande) in un numero ridotto di bande (al limite solo una), senza particolari perdite d'informazioni spettrali;
  3. uso di funzioni dedicate in grado di passare dal dominio RGB a quello monocromatico (rgb2gray per esempio in ambiente Matlab). Questa funzione converte l'immagine RGB originale in una a scala di grigi attraverso l'eliminazione dell'informazioni di tonalità e di saturazione, pur mantenendo la luminanza.

Il risultato di questa processo è mostrato nella figura 3.9. Tale immagine contiene una parte del mosaico contenente 10 scene ASTER relative al confine Libia/Ciad. Sull'immagine in toni di grigio viene compiuto una pre elaborazione, volta ad ottenere un migliore contrasto tra gli oggetti desiderati (per esempio tracce) e lo sfondo.

- L'immagine originale in scala di grigi, viene divisa in più parti (settori) di dimensioni adeguate, su ciascuna delle quali agisce l'algoritmo di *template matching* che realizza l'individuazione automatica degli oggetti che potenzialmente possono essere strutture artificiali. L'analisi automatica di ogni parte del mosaico, è realizzato utilizzando gli strumenti forniti dalla teoria del template matching descritta in questo capitolo. Nel nostro caso, visto che siamo interessati all'individuazione di aeroporti o simili, possiamo usare *templates* ("immagini sonda") contenenti segmenti lineari, piste estratte da altre immagini campione o costruite ad hoc (lunghezza, larghezza, inclinazione, ecc) per questa applicazione. La cosa importante da sottolineare è che l'intero processo (filtraggio, determinazione delle soglie, estrazione degli oggetti, ecc) avviene in modo completamente automatico senza intervento da parte degli operatori umani.
- Il processo precedente viene ripetuto per ogni blocco dell'immagine considerato (vedere figura 3.8). Infine, tutti i

blocchi processati vengono riorganizzati in una sola immagine uguale, per dimensioni, a quella originale. In altre parole, attraverso opportune funzioni Matlab possiamo mosaicare tutti i blocchi processati per dare luogo ad una immagine processata di dimensioni identiche a quella di partenza.



**Figura 3.8.** Schema a blocchi del sistema di pattern recognition basato sulla Template Matching per l'individuazione delle strutture artificiali.

- In generale, alla fine del processing precedente l'immagine elaborata che si ottiene (dopo mosaico), presenta diversi oggetti: i nostri oggetti di interesse più alcuni oggetti indesiderati (il rumore). Questi ultimi possono essere rimossi mediante un filtraggio morfologico che, sfruttando le caratteristiche abbastanza note degli "oggetti ricercati", consente di raggiungere lo scopo finale.

L'algoritmo qui descritto è stato applicato in due scenari molto diversi tra di loro: nell'individuazione di strutture artificiali in una zona deserta (confine Libia/Ciad), quando non sono disponibili informazioni sulle caratteristiche spettrali dell'oggetto di destinazione e nell'

individuazione di piste di atterraggio in zone abitate nella regione del Kashmir (confine Pakistan/India). In entrambi i casi, a causa delle notevoli dimensioni delle immagini che coprono l'area di interesse, abbiamo avuto problemi legati alle risorse di calcolo e la gestione di questi dati in ambiente Matlab. Per questo motivo, abbiamo introdotto nella catena di processing una suddivisione del dato di partenza in una serie di immagini più piccole, di dimensione opportuna e adatte ad essere gestite in modo più comodo ed efficiente dall'algoritmo di base.

Inoltre, vale la pena sottolineare che i casi studiati in questo contesto presentano caratteristiche dello sfondo (background), molto diverse tra di loro e quindi rappresentano casi test estremi atti a verificare le prestazioni dei metodi sviluppati. In entrambi i casi, gli oggetti di interesse sono stati rilevati correttamente utilizzando, insieme alla tecnica del template matching, alcune informazioni accessorie riguardanti la forma prevista e le dimensioni caratteristiche dell'oggetto. Naturalmente, nel caso del Kashmir queste informazioni sono necessarie per discriminare piste d'atterraggio da strade, fiumi, canali di irrigazione e così via.

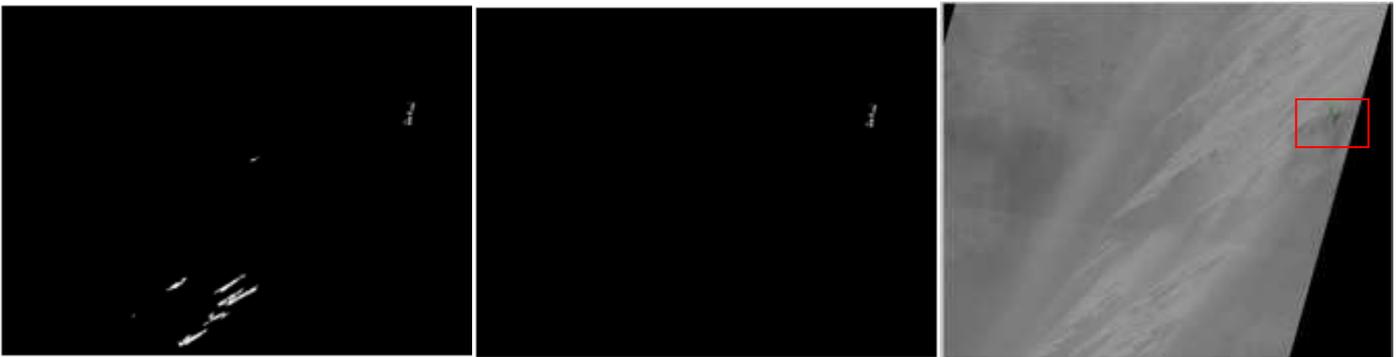


*Figura 3.9. Mosaico ASTER: immagine in toni di grigio ottenuta come descritto nel testo. Il rettangolo rosso evidenzia l'area dell'aeroporto.*

La figura 3.10 mostra l'immagine binaria ottenuta dopo la frammentazione e l'elaborazione dell'immagine originale, come uscita

dalla catena di elaborazione. Gli oggetti non desiderati vengono rimossi sfruttando le loro caratteristiche geometriche (area, lunghezza, larghezza, solidità, baricentro, ecc). Queste proprietà geometriche, vengono sfruttate nella costruzione di opportuni filtri morfologici, in grado di separare nello stadio finale gli oggetti "buoni" da quelli "falsi" (figura 3.10).

L'immagine finale, contenente solo l'oggetto desiderato mostrato nella figura 3.10 centrale, può essere sovrapposta sull'immagine originale. In questo modo, le strutture desiderate estratte possono essere enfatizzate (figura 3.10 di destra). In generale, possiamo dire in questo caso che l'algoritmo è semplice e robusto, ma al fine di aumentare la sua efficienza quando si utilizzano grandi mosaici, è richiesto un hardware molto potente.

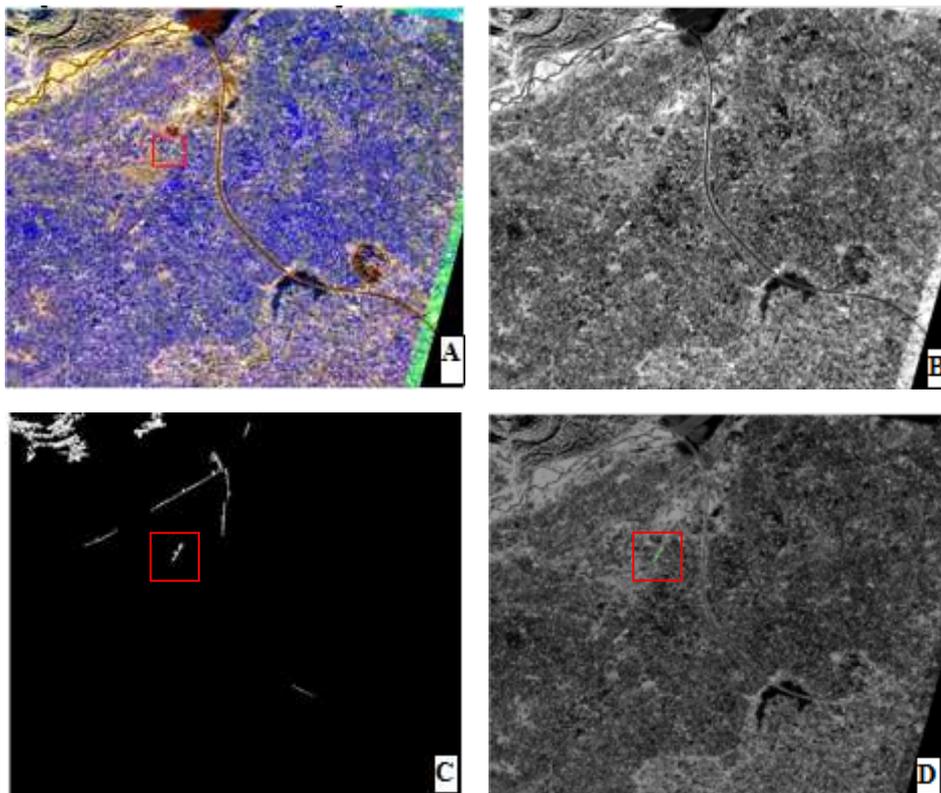


**Figura 3.10.** Risultati del processing sul mosaico della figura 3.9. Oggetti rimanenti dopo il processing (sinistra), successivo al filtraggio morfologico finale (centro) e sovrainposto sull'immagine originale (rettangolo a destra).

La figura 3.11 (A) mostra un dettaglio del mosaico dato nella figura 3.7, relativo all'area di studio nel Kashmir. Questa immagine corrisponde a uno dei settori selezionati per condurre la ricerca poiché è presente uno degli obiettivi. Come si può notare, lo sfondo (*background* dell'immagine) è molto diverso dal caso test studiato precedentemente. La procedura di rilevamento creata su quest'area, è stata poi applicata sull'intero mosaico portando all'individuazione di tutte le piste di atterraggio esistenti nella zona e oggetto di studio. In figura 3.11 (C) viene riportato il risultato delle varie trasformazioni applicate al settore 3.11 (A). È evidente in quest'immagine come, al termine della catena di processing e prima dell'applicazione dei filtri morfologici, gli oggetti estratti siano molti: l'oggetto desiderato (aeroporto di Mandi) più tanti altri oggetti indesiderati, aventi forma e dimensioni molto diverse rispetto agli oggetti ricercati (aeroporti). Questo fenomeno, quasi sempre presente, è conseguenza della non idealità del processo di elaborazione,

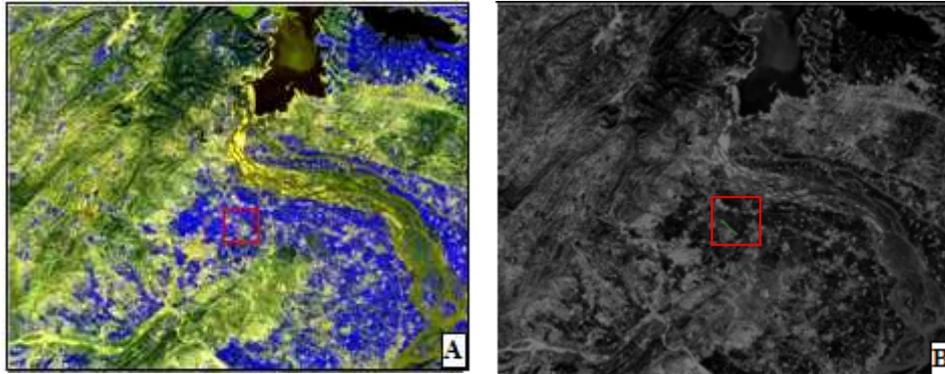
della complessità dell'applicazione e della difficoltà di mettere a punto una fase di *pre-processing* universale, valida cioè sempre, indipendentemente dal sensore, risoluzione spaziale, background, ecc. Questo difetto, intrinseco nell'*image processing* è indipendente dall'approccio usato, ma la scelta di uno approccio rispetto ad un altro può aiutare a generare meno rumore nella fase finale. Fondamentale resta, come accennato in altri capitoli, la fase di *pre-processing* compiuta sul dato originale, poiché questa fase crea le condizioni per fare funzionare in modo ottimale l'algoritmo di individuazione degli oggetti nelle fasi successive.

A questo punto l'aggiunta di un ulteriore passo basato sui filtri morfologici, in grado di sfruttare le caratteristiche degli oggetti cercati (lunghezza, forma, ecc) può aiutare nella loro classificazione finale. I filtri morfologici possono essere applicati sfruttando la disponibilità, *a priori*, di informazioni relative all'oggetto di interesse. La stessa sequenza di passi (algoritmo) viene ripetuta su tutte le zone del mosaico.

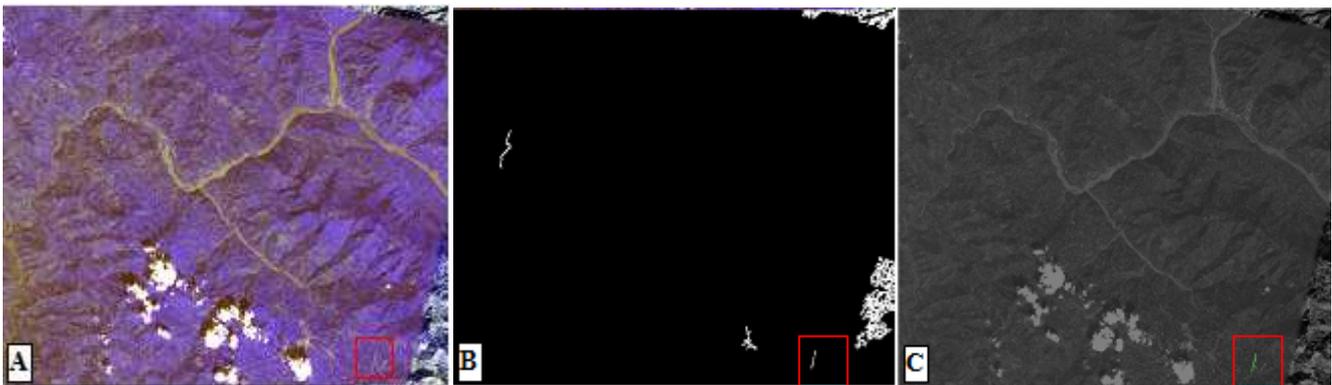


**Figura 3.11.** Immagine in falsi colori di una frazione del mosaico ASTER sull'area del Kashmir studiata (A). Immagine in tonalità di grigio derivata (B) e risultati ottenuti dall'algoritmo prima del filtraggio finale (C), contenendo la pista desiderata più altri oggetti identificati dalla catena di processing. L'oggetto estratto (aeroporto Mandi) sovrainposto sull'immagine originale (D) racchiuso dal rettangolo rosso.

Le figure 3.12 e 3.13 mostrano i risultati ottenuti su altre parti del mosaico ASTER originale, evidenziando anche in questi casi l'efficacia dell'algoritmo implementato.



**Figura 3.12.** Immagine in falsi colori di un'altra porzione del mosaico ASTER del Kashmir (A) e relativo risultato (B) dopo rimozione del rumore con opportuni filtri morfologici (aeroporto Mangla).



**Figura 3.13.** Immagine a falso colore della 3° porzione di studio estratta dal mosaico (A) e gli oggetti individuati (desiderati e non) mediante *template matching* sulla scena (B). Dopo filtraggio morfologico si arriva finalmente all'aeroporto cercato (aeroporto Mangla).

I progetti finanziati dall'Unione Europea nei quali sono stato coinvolto durante l'attività di dottorato, quali GMOSS, G-MOSAIC, ecc, mi hanno dato l'opportunità di affrontare problemi legati all'individuazione di infrastrutture artificiali, nel controllo e monitoraggio delle frontiere nazionali attraverso l'uso delle immagini satellitari.

E questo argomento ha guidato lo sviluppo di questo capitolo, attraverso la ricerca di tecniche in grado di rilevare in modo automatico, strutture artificiali presenti su immagini satellitari di grandi dimensioni (diversi GB di dati), quali i mosaici. Per soddisfare questo obiettivo, sono stati creati e applicati algoritmi basati sul *template matching*. In particolare, questa tecnica è stata utilizzata per rilevare aeroporti e piste di atterraggio in zone caratterizzate da un background (sfondo, paesaggio)



molto diversi tra di loro, consentendo così di testare la robustezza degli algoritmi e rendendoli il più generali possibile.

Infine, la catena di processing si conclude sfruttando la conoscenza *a priori* che si ha sugli oggetti desiderati (forma, dimensione, asse maggiore, asse minore, ecc), per costruire filtri morfologici in grado di discriminare correttamente (o con il minimo margine di errore) gli oggetti veri da quelli falsi.

La tecnica del *Template Matching* e gli algoritmi derivati e sviluppati per questo scopo, sono stati applicati a due grandi mosaici di immagini satellitari ASTER relativi alle aree di frontiera tra la Libia e il Ciad (in Africa) e tra il Pakistan e l'India (nella regione del Kashmir). I risultati mostrano che la procedura generale messa a punto, applicata ad entrambi gli scenari, funziona bene nonostante caratteristiche paesaggistiche così diverse tra di loro come evidenziato e descritto nei paragrafi precedenti.



## Capitolo 4

# Pattern Recognition: Approccio Statistico

L'uso del calcolatore come strumento atto ad interpretare le immagini, in qualche modo rilevate e digitalizzate, è un'applicazione relativamente recente. E questa attività, di interpretazione ed analisi di immagini, è una operazione piuttosto complessa poiché, infatti, l'immagine è disponibile semplicemente come una tavola di valori di grigio (i livelli di grigio, formulati in una matrice di pixel), con l'aggiunta di eventuali disturbi, chiamati rumore, provocati dagli strumenti di rilevamento.

Di solito l'attività di interpretazione di una immagine è influenzata dalla tipologia dell'immagine da riconoscere, di conseguenza, l'attività di scelta e messa a punto di *algoritmi di riconoscimento* deve tener conto della tipologia dell'oggetto da riconoscere. Il compito dell'*image processing & analysis* è, dunque, quello di eseguire delle misure quantitative da una immagine (oggetto o pattern) per produrre una descrizione di questa, e dalla descrizione interpretarne il contenuto [26].

Come già visto nei capitoli precedenti, in genere un sistema di *pattern recognition* riceve in ingresso la descrizione di un oggetto, ovvero un insieme di misure che lo caratterizzano (*feature*), e sulla base di queste “decide” a quale classe l'oggetto appartiene. La decisione della classe di appartenenza ha un costo associato all'assegnazione dell'oggetto alla classe sbagliata, e l'obiettivo con cui realizzare un buon sistema di *pattern recognition* è quello di minimizzare il costo di classificazione.

Un sistema per il riconoscimento di oggetti è in generale formato dai seguenti elementi:

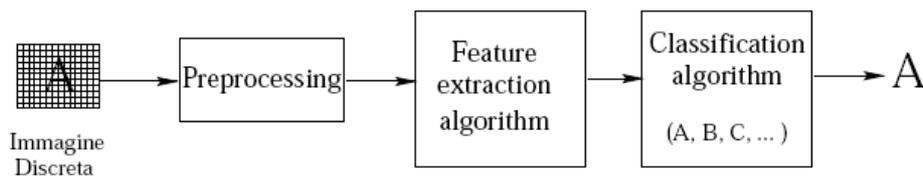
1. Acquisizione dell'Immagine;
2. Sistema di Pre-processing;
3. Estrazione delle Caratteristiche (*features*);
4. Classificazione degli Oggetti (pattern).

Descriviamo ora meglio i singoli blocchi costitutivi del sistema, per comprendere con maggiore chiarezza il loro funzionamento e le

difficoltà di progettazione che si incontrano nel affrontare questo problema.

#### 4.1.1 Processo di Riconoscimento

In generale un'immagine è una funzione continua bidimensionale che associa ad una scena la sua rappresentazione bidimensionale. Ad esempio una fotografia in bianco e nero (monocromatica) può essere associata ad una funzione  $f(x, y)$ , il valore che questa funzione assume in un punto individua il tono di grigio dell'immagine in quel punto. Il processo di conversione di un'immagine analogica (come è in natura) ad una digitale, adatta ad essere elaborata dal calcolatore, è detto digitalizzazione. Con questa tecnica ad ogni pixel viene associato un valore, che corrisponde generalmente al valor medio di grigio nell'area corrispondente (quantizzazione). L'attività di una applicazione di *pattern recognition* è proprio quella di rappresentare in un modello interno la scena rappresentata in una immagine digitale, utilizzando un set finito di features estratte dall'immagine. Una feature è, generalmente parlando, una grandezza misurabile che può essere estratta dall'immagine, e la caratterizza in qualche modo [27]. Le features possono essere simboliche o numeriche.



**Figura 4.1.** Schema di un sistema di pattern recognition per il riconoscimento di caratteri.

Per esempio, in un sistema di riconoscimento posto a riconoscere diverse monete, una feature simbolica può essere il colore, mentre una feature numerica può essere l'area, o il perimetro. Mediante questa descrizione interna data dalle features, un opportuno metodo di classificazione assegnerà all'immagine una classe di appartenenza.

Schematicamente, la figura 4.1 mostra come è suddiviso un sistema di pattern recognition per il riconoscimento di caratteri. Questo consiste in una fase di *pre-processing*, dove vengono eseguite alcune elaborazioni preliminari sull'immagine per migliorarne la qualità e minimizzare il rumore, un *algoritmo di estrazione delle features*, con le quali ci costruiamo una rappresentazione interna dell'immagine digitale, e un *algoritmo di classificazione*. Quest'ultimo per poter classificare correttamente un pattern, ha bisogno di alcuni modelli campione già classificati, chiamato *training set*, in modo da allenare il sistema o fornire un modello di paragone (in questo caso abbiamo un modello interno che rappresenta ogni lettera dell'alfabeto).



#### 4.1.2 Estrazione delle Features

Di enorme aiuto per le applicazioni di *image analysis* è quindi l'estrazione di features a partire dalle immagini, con le quali può essere data una descrizione e una interpretazione di una scena. Le features di una immagine vengono ricavate applicando un *algoritmo di estrazione delle features* alle immagini in input. Inoltre una feature può essere un oggetto di alto livello: esempio di features ad alto livello può essere un descrittore geometrico di una regione dell'immagine o persino di un oggetto geometrico in 3D. Le features possono essere rappresentate da funzioni continue, discrete o binarie. Quelle binarie possono essere utilizzate per rappresentare la presenza o l'assenza di una particolare feature. Bisogna notare, comunque, che:

1. il processo di estrazione di features può richiedere grandi quantità di calcolo;
2. le features estratte possono contenere errori, dovuti a rumore oppure ad una non corretta applicazione degli algoritmi di estrazione.

Bisogna quindi stare attenti a scegliere l'algoritmo giusto per il problema in questione. Il processo di selezione delle features è uno dei problemi chiave di ogni sistema di *pattern recognition*, e consiste nel decidere quali features utilizzare, fra tutte quelle disponibili, per uno specifico problema. E' importante, per la buona riuscita di tali sistemi, scegliere (e quindi estrarre) features che:

1. siano rilevanti per il problema che si sta trattando;
2. siano computazionalmente realizzabili;
3. diano il minimo numero di false classificazioni;
4. riducano il problema in un insieme ridotto di dati senza scartare informazioni importanti.

In alcuni casi ci sono degli strumenti matematici che ci aiutano a selezionare le features più appropriate, in altri casi possono essere di aiuto alcune simulazioni. Comunque la maggior parte delle volte ci si affida all'esperienza, e la scelta delle features da utilizzare è spesso decisa procedendo a tentativi.

Le features vengono raggruppate in un *vettore di features* a  $n$  dimensioni. In questo modo, se ad ogni feature viene fatto assumere un valore reale, i vettori di features vengono mappati nello spazio  $R^n$ .

Sono molte le misure che possono essere compiute sugli oggetti per caratterizzarli, e diversi gli approcci statistici che possono essere adottati per realizzare un sistema di *pattern recognition statistico*. Per ragioni che saranno chiarite nel prosieguo del capitolo, il sistema di pattern recognition statistico seguito in questo studio, è basato sui *Momenti Invarianti* (MI). Queste grandezze (MI) sono un insieme di sette



momenti (misure), calcolati su ogni oggetto, che risultano invarianti per traslazione, rotazione e scala, conferendo per questo motivo molta robustezza al sistema stesso.

### 4.1.3 Classificazione di Pattern

L'attività di classificazione delle immagini digitali, ha come scopo stabilire una relazione fra le strutture presenti in una immagine e gli oggetti presenti nella scena reale o in un database di riferimento. Meglio si può definire dicendo che, stabilito un insieme di classi, la classificazione è una regola che assegna ad una forma presente nell'immagine una classe di appartenenza, quando questa soddisfa una certa descrizione.

Se la rappresentazione interna di un pattern è composta da un vettore di caratteristiche ad  $n$ -dimensioni, prima di eseguire la classificazione bisogna partizionare lo spazio delle caratteristiche (*features*) in regioni, dove ad ogni regione corrisponde una classe. In questo modo, la classificazione di un vettore di features  $X$  diventa molto semplice: *bisogna determinare la regione di  $R^n$  nella quale ricade  $X$ , ed assegnare ad  $X$  quella classe.*

## 4.2 La Teoria dei Momenti

Funzionalità chiave nei sistemi d'*image processing & analysis* è il riconoscimento automatico di un oggetto in una scena, indipendentemente dalla sua posizione, dimensione e orientamento. I **momenti** e funzioni di momenti sono stati utilizzati per ricavare *features* in numerose applicazioni, adatte al riconoscimento di modelli in immagini bidimensionali [28]. Hu [29] è stato il primo ad introdurre i momenti invarianti già nel 1961.

Utilizzando una combinazione lineare di *momenti regolari*, egli ha derivato un insieme di *momenti invarianti*. Questi sono un insieme di funzioni non lineari, i quali hanno la proprietà di essere **invarianti per traslazione, rotazione e scala**. Queste features catturano informazioni globali sugli oggetti (*pattern*) presenti nell'immagine, usate poi per il loro riconoscimento.

Diversamente da quanto viene fatto da altri metodi, quali ad esempio i descrittori di Fourier [26], vengono applicati esclusivamente al contorno dell'oggetto da riconoscere, rendendolo molto più sensibile al rumore e alle perturbazioni sul contorno.

### 4.2.1 Momenti Regolari

I *momenti regolari* sono stati introdotti per la prima volta da Hu [29] nel campo dell'*image processing* per il riconoscimento di immagini. Durani

e altri [30] li hanno utilizzati con successo per il riconoscimento vari tipi di forme, mentre Wong e Hall [31] hanno utilizzato i momenti per il riconoscimento automatico di immagini radar.

Sia  $f(x,y)$  una funzione continua rappresentante l'intensità dell'immagine nel punto  $(x,y)$ ; i **momenti regolari** (oppure chiamati generici)  $m_{pq}$  di ordine  $(p + q)$  vengono definiti come [29].

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (4.1)$$

dove  $\{p, q = 0, 1, 2, \dots, \infty\}$ . Per le immagini digitali l'integrale viene sostituito dalla sommatoria, e  $m_{pq}$  diventa

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y). \quad (4.2)$$

#### 4.2.2 Momenti Centrali

Per rendere i momenti regolari invarianti alla traslazione, vengono introdotti i momenti centrali  $\mu_{pq}$ , definiti da:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy \quad (4.3)$$

dove

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad p, q = 0, 1, 2, \dots, \infty \quad (4.4)$$

Il punto  $(\bar{x}, \bar{y})$  viene chiamato *centroide* dell'immagine, e viene calcolato mediante i momenti regolari definiti nella formula (4.2). I momenti centrali  $\mu_{pq}$  sono equivalenti ai momenti regolari  $m_{pq}$  di una immagine che è stata spostata, in modo tale che il centroide dell'immagine  $(\bar{x}, \bar{y})$ , coincida con l'origine del sistema di riferimento. Ciò comporta che *i momenti centrali  $\mu_{pq}$  siano invarianti per traslazione*. Ovviamente per le immagini digitali  $\mu_{pq}$  diventa:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y). \quad (4.5)$$

Mediante la combinazione lineare di momenti, si possono derivare parecchie funzioni non lineari che sono *invarianti per traslazione, scala e rotazione*. Questi vengono chiamati **momenti invarianti** [32]. Ad esempio, le seguenti combinazioni di momenti sono *invarianti per traslazione e rotazione*,

$$\begin{aligned}\psi_1 &= \mu_{20} + \mu_{02} \\ \psi_2 &= (\mu_{20} - \mu_{02}) + 4\mu_{11}^2 \\ \psi_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2\end{aligned}\quad (4.6)$$

dove  $\mu_{20}, \mu_{02}, \dots$  sono ricavati con la formula (4.3) nel caso di immagini continue, altrimenti con la formula (4.5) nel caso di immagini digitali.

Le funzioni dei momenti centrali sopra definite **non** sono *invarianti per scala*. I momenti centrali possono essere normalizzati e **diventare invarianti per scala** definendo [29] [32]:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \quad \gamma = \frac{p+q}{2} + 1. \quad (4.7)$$

dove  $\{p, q = 0, 1, 2, \dots, \infty\}$ . Un insieme di funzioni **non lineari** che **risultano invarianti per rotazione, traslazione e cambiamento di scala** sono l'insieme dei sette momenti di Hu:

$$\begin{aligned}\phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\ &\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ \phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\ &\quad (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].\end{aligned}\quad (4.8)$$

I valori numerici di  $\Phi_1, \dots, \Phi_4$  in genere sono molto grandi. Per evitare problemi di precisione e di overflow, spesso viene utilizzato il logaritmo del modulo  $\{\log(|\Phi_i|), i=1, \dots, 4\}$ , e queste grandezze sono usate come *features* rappresentanti l'oggetto.

Le features illustrate sopra, possono essere applicate sia sulle immagini binarie che su quelle a livelli di grigio. In entrambi i casi, comunque, si assume che la regione che rappresenta l'oggetto da analizzare sia stata estratta dalla scena e che lo sfondo abbia intensità zero. L'utilizzo dei momenti regolari nel campo della *pattern recognition*, è stato suggerito dal seguente teorema [26]:

**Teorema:** *l'insieme infinito di momenti  $\{m_{p,q} \mid p, q = 0, 1, 2, \dots, \infty\}$  determina una univocamente  $f(x, y)$  e viceversa.*

Ciò significa che, i momenti possiedono la proprietà di rappresentare in maniera univoca una determinata immagine o pattern. Il teorema può essere dimostrato mediante l'espansione in serie di potenze della trasformata di Fourier di  $f(x,y)$  [33].

### 4.2.3 Momenti Ortogonali: Momenti di Zernike

I momenti regolari possono essere visti come la proiezione della funzione  $f(x,y)$  nella base  $x^p y^q$ . Sfortunatamente la base  $x^p y^q$  non è ortogonale, e di conseguenza vi è un certo grado di ridondanza nell'informazione contenuta nei diversi  $m_{p,q}$ .

Teague [34] ha suggerito l'uso di **momenti ortogonali**, basati sulla teoria dei polinomi ortogonali per ovviare al problema associato con i momenti regolari. Utilizzando un insieme di polinomi complessi, che risultano essere ortogonali all'interno del cerchio unitario  $\{x^2 + y^2 = 1\}$ , ha introdotto i **momenti di Zernike**, i quali consentono di ricavare momenti invarianti indipendenti in maniera semplice, fino ad un ordine arbitrario. Questi momenti hanno la proprietà di essere *invarianti per rotazione*, quindi il valore ricavato dai momenti applicati ad una immagine, ed alla stessa ma ruotata, non varia. Inoltre la proprietà di ortogonalità ci permette di dare una descrizione degli oggetti da riconoscere, utilizzando un numero limitato di features.

I momenti di Zernike bidimensionali di ordine  $n$ , applicati ad una immagine rappresentata da una funzione di intensità  $f(x,y)$ , vengono definiti come:

$$Z_{nm} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 [V_{nm}(r, \theta)]^* f(r \cos \theta, r \sin \theta) r dr d\theta \quad (4.9)$$

dove

$$r = \sqrt{x^2 + y^2}, \quad \theta = \tan^{-1}\left(\frac{y}{x}\right), \quad -1 < x, y < 1,$$

e i valori  $n$  ed  $m$  sono interi tali che soddisfino la condizione,

$$0 \leq |m| \leq n, \quad n - |m| = \text{pari}, \quad n > 0$$

i polinomi di Zernike

$$V_{nm}(r, \theta) = R_{nm}(r) e^{im\theta} \quad (4.10)$$

dove  $i = \sqrt{-1}$  sono un insieme completo di funzioni ortogonali a valori complessi all'interno del cerchio unitario  $x^2 + y^2 \leq 1$ , e vengono definiti come,

$$R_{nm}(r) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} r^{n-2s} \quad (4.11)$$

dove risulta che  $R_{n(-m)}(r) = R_{nm}(r)$ .

Questi polinomi sono ortogonali, in quanto soddisfano la condizione,

$$\frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 [V_{nm}(r, \theta)]^* V_{pq}(r, \theta) dr d\theta = \begin{cases} 1 & \text{se } n = p \text{ e } m = q \\ 0 & \text{altrimenti} \end{cases}$$

I momenti di Zernike sono la proiezione della funzione dell'immagine  $f$  in questa base ortogonale. Per le immagini digitali, gli integrali sono sostituiti da sommatorie, ottenendo:

$$Z_{nm} = \frac{n+1}{\pi} \sum_x \sum_y [V_{nm}(r, \theta)]^* f(x, y) \quad (4.12)$$

con

$$x^2 + y^2 \leq 1$$

Per calcolare i momenti di Zernike di una immagine, il centro dell'immagine viene preso come l'origine, e le coordinate dei pixel vengono mappate all'interno del cerchio unitario  $\{x^2 + y^2 \leq 1\}$ . Quei pixel che si trovano all'esterno del cerchio unitario non verranno usati nel calcolo.

Poichè è più conveniente lavorare con funzioni reali piuttosto che con numeri complessi, considerando la funzione (4.10), e che  $\{e^{im\theta} = \cos(m\theta) + i\sin(m\theta)\}$ , la funzione  $Z_{nm}$  è spesso spezzata nelle parti reale e immaginaria, come mostrato sotto:

$$\begin{aligned} Re[Z_{nm}] &= \frac{n+1}{\pi} \sum_x \sum_y R_{nm}(r) \cos(m\theta) f(x, y) \\ Im[Z_{nm}] &= \frac{n+1}{\pi} \sum_x \sum_y R_{nm}(r) \sin(m\theta) f(x, y). \end{aligned} \quad (4.13)$$

Le features derivate dai momenti di Zernike sono ricavate utilizzando le proprietà rotazionali di questi momenti [35]. Infatti, Sia  $f(x,y)$  la funzione rappresentante una immagine, e  $f'(x,y)$  la funzione che rappresenta la stessa, ma ruotata di un angolo  $\alpha$ , la relazione tra l'immagine ruotata e quella non ruotata nelle stesse coordinate polari è:

$$f'(r, \theta) = f(r, \theta - \alpha).$$

I momenti di Zernike dell'immagine ruotata sono:

$$Z'_{nm} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r, \theta - \alpha) R_{nm}(r) e^{-im\theta} r dr d\theta.$$

Effettuando il cambiamento di variabile  $\theta_1 = \theta - \alpha$  si ottiene:

$$\begin{aligned} Z'_{nm} &= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r, \theta_1) R_{nm}(r) e^{-im(\theta_1 + \alpha)} r dr d\theta_1 \\ &= \left[ \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r, \theta_1) R_{nm}(r) e^{-im\theta_1} r dr d\theta_1 \right] e^{-im\alpha} \\ &= Z_{nm} e^{-im\alpha}. \end{aligned} \quad (4.14)$$

L'equazione sopra mostra come i momenti di Zernike possiedano una semplice proprietà di trasformazione per rotazione; ogni momento di Zernike acquisisce un cambiamento di fase in una rotazione. Questa semplice proprietà porta alla conclusione che il modulo dei momenti di Zernike di una funzione di una immagine ruotata rimangono identici a quelli dell'immagine non ruotata. Quindi  $|Z_{nm}|$  può essere considerata come *feature invariante per rotazione* della funzione descrivente l'immagine.

$$F_i = |Z_{nm}|$$

Al variare di  $n$  ed  $m$  abbiamo quindi un set infinito di features che possono essere utilizzate per descrivere una determinata immagine.

Bisogna notare che, poichè  $R_{nm}(r) = R_{n(-m)}(r)$ , dalla (4.9) risulta anche,

$$\begin{aligned} Z_{n(-m)} &= [Z_{nm}]^* \\ |Z_{nm}| &= |Z_{n(-m)}| \end{aligned}$$

e così per calcolare i vari  $|Z_{nm}|$  ci possiamo limitare ad analizzare quelli con  $m \geq 0$ .

feature					$\mu$	$\sigma$	$\frac{\sigma}{\mu}$
$ Z_{20} $	4804.17	4801.35	4801.87	4803.32	4802.68	1.68	0.0003
$ Z_{22} $	193.33	192.17	192.81	193.27	192.89	0.29	0.0015
$ Z_{31} $	2776.23	2770.25	2776.66	2782.53	2776.42	25.16	0.0090
$ Z_{33} $	37.72	37.34	31.80	31.73	34.65	11.11	0.3208

**Tabella 4.1.** Invarianza rotazionale dei momenti di Zernike.

La proprietà dei momenti di Zernike di essere invarianti per rotazione è dimostrata dal seguente esperimento. La tabella 4.1 include quattro versioni ruotate di una immagine binaria di 240x240 pixel (opportunamente scalata in figura), con angolo di rotazione di 0°; 90°; 180° e 270° rispettivamente.

Nella tabella 4.1 sono riportati i valori dei momenti di Zernike del secondo e terzo ordine, la loro media  $\mu$ , la deviazione standard  $\sigma$  e il valore  $\sigma/\mu$  che indica la distribuzione dei valori di  $|Z_{nm}|$  dalla media corrispondente. Come si può vedere, l'invarianza rotazionale è raggiunta, in quanto i valori di  $\sigma/\mu$  sono molto piccoli. Ovviamente non si è potuto ottenere una invarianza esatta (cioè  $\sigma/\mu = 0$ ), in quanto l'immagine analizzata è digitale, piuttosto che continua. In questa tabella si può inoltre notare che, la distribuzione dei valori dei momenti cresce al crescere dell'ordine, è questa tendenza è valida in generale per tutti i momenti.

Abbiamo visto come le *features* ricavate dai momenti di Zernike sono solo invarianti per rotazione. Per ottenere pure l'invarianza per scala e traslazione l'immagine deve essere prima *normalizzata* rispetto a queste variabili. Quindi le *features* di Zernike devono essere estratte da immagini sottoposte precedentemente alla normalizzazione rispetto scala e traslazione.

In [36] viene dimostrato come una funzione discreta  $\bar{f}(x,y)$ , per la quale i momenti di Zernike coincidono esattamente con quelli di  $f(x,y)$  fino ad un dato ordine  $n_{max}$ , può essere espressa nella forma:

$$\bar{f}(x, y) = \sum_{n=0}^{n_{max}} \sum_m Z_{nm} V_{nm}(r, \theta) \quad (4.15)$$

dove  $m$  ha gli stessi vincoli espressi nell'equazione (4.9) e  $Z_{nm}$  denotano i momenti di Zernike di  $f(x,y)$ . È interessante notare che, per  $n_{max}$  che tende ad infinito,  $\bar{f}(x,y)$  approssima  $f(x,y)$ . Quindi dai momenti di una immagine possiamo ricostruire la stessa. Questo ci garantisce che è *sufficiente utilizzare features derivate solamente dai momenti di Zernike, per descrivere (e riconoscere) una immagine.*

#### 4.2.4 Altri tipi di Momenti

Per una trattazione completa, vengono citati altri tipi di momenti. Un altro tipo di momenti ortogonali sono i **momenti di Legendre** [37] di ordine  $(m + n)$  definiti come

$$\lambda_{nm} = \frac{(2m + 1)(2n + 1)}{4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P_m(x) P_n(y) f(x, y) dx dy \quad (4.16)$$

dove  $\{m, n = 0, 1, 2, \dots\}$ , e  $P_{n,m}$  sono i polinomi di Legendre, i quali formano una base ortogonale nell'intervallo  $[-1, 1]$ , e sono definiti come:

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n \quad (4.17)$$

I **momenti rotazionali** di ordine  $n$  e ripetizione  $m$  sono invece definiti come

$$D_{nm} = \int_0^{2\pi} \int_0^\infty r^n e^{-im\theta} f(r \cos \theta, r \sin \theta) r dr d\theta \quad (4.18)$$

dove  $n$  e  $m$  sono interi, con  $n \geq 0$ .

I **momenti complessi** sono stati introdotti come un modo semplice e diretto per derivare momenti invarianti. I momenti complessi di ordine  $(p+q)$  sono definiti da:

$$C_{pq} = \int_{-\infty}^\infty \int_{-\infty}^\infty (x + iy)^p (x - iy)^q f(x, y) dx dy \quad (4.19)$$

dove  $\{p, q = 0, 1, 2, \dots\}$  e  $i = \sqrt{-1}$ .

### 4.3 Sistema di Pattern Recognition basato sui Momenti Geometrici Invarianti

Come evidenziato nei paragrafi precedenti di questo capitolo, il calcolo dei *momenti geometrici invarianti* degli oggetti, rappresenta uno degli strumenti più potenti e versatili a disposizione nella costruzione di un sistema di *pattern recognition* per il riconoscimento degli stessi. Infatti, come dimostrato precedentemente, l'invarianza di scala, rotazionale e di traslazione sono caratteristiche che rendono estremamente robusto un tale sistema, poiché la loro assenza rappresenta una delle principali cause di errore nei sistemi di riconoscimento automatico di oggetti.

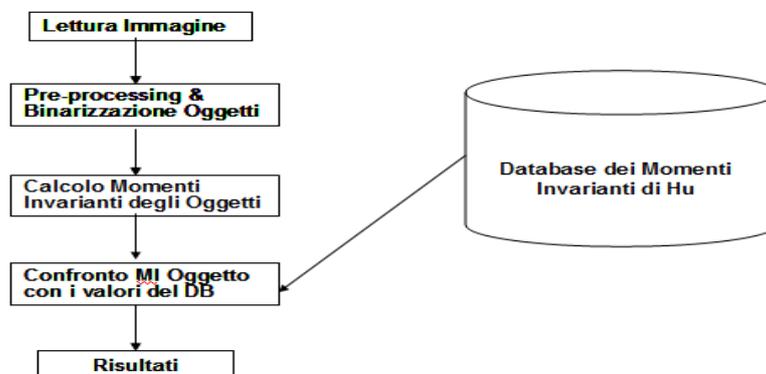


Figura 4.2. Schema di un sistema di pattern recognition statistico basato sui momenti geometrici invarianti.



Sulla base della teoria esposta precedentemente, in questo studio abbiamo implementato un sistema di *pattern recognition* basato sui *momenti geometrici invarianti* di *Hu*, capace di distinguere e classificare correttamente gli oggetti d'interesse. In particolare, usando immagini satellitari relative a due scene completamente diverse (figure 4.3 e 4.4), abbiamo identificato gli oggetti d'interesse su entrambe le immagini, calcolando i momenti invarianti degli stessi e infine, inserendo tali valori in un opportuno database contenente tali caratteristiche. La classificazione dell'oggetto (la sua identificazione), si raggiunge annullando o minimizzando la distanza tra il vettore contenente i valori dei momenti invarianti dell'oggetto, e quelli presenti nel database relativo a tutti i pattern considerati come potenziali oggetti da riconoscere. La figura 4.2 rappresenta lo schema a blocchi del sistema di *pattern recognition statistico* basato sui *momenti geometrici invarianti* di *Hu* qui adottato.

Le immagini prese in considerazione per la realizzazione del database dei momenti, e l'implementazione dell'intero sistema di riconoscimento statistico sono molto diverse tra di loro, in modo da poter realizzare un sistema che sia il più completo possibile, indipendentemente dagli oggetti che si vogliono riconoscere.

La prima immagine (figura 4.3), rappresenta un tratto dell'autostrada Monaco - Salisburgo acquisita dal DLR (Agenzia Spaziale Tedesca) nel 2008, mediante un sensore aereo *multi-spettrale* avente 50 cm di risoluzione spaziale. Lo scopo in questa applicazione è quello di isolare le macchine presenti sull'autostrada per arrivare alla classificazione delle stesse, implementando così un sistema off-line per il monitoraggio del traffico. Il secondo dato, riguarda un'immagine *pan-cromatica* avente risoluzione spaziale di 1,8 m del satellite EROS 1A (figura 4.4), relativa ad un cimitero di aerei civili sito a Tucson, Arizona. L'obiettivo in questo secondo caso, riguarda la possibilità di isolare e riconoscere automaticamente gli aerei presenti nell'immagine satellitare attraverso il matching dei loro momenti e quelli presenti nel database.

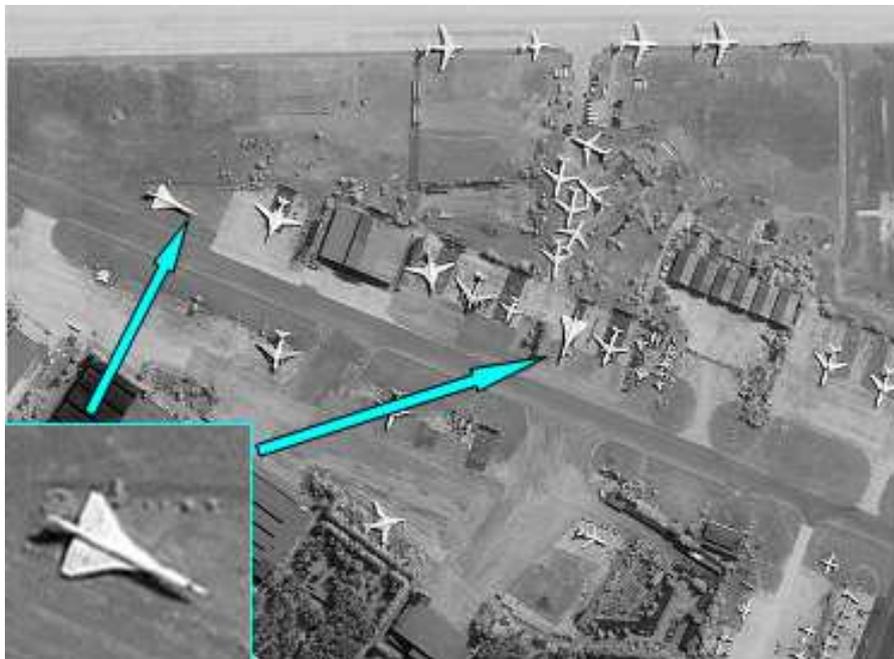
Il database del sistema è quindi costruito inserendo i vettori delle caratteristiche (momenti geometrici invarianti in questo caso), relativo a tutti gli oggetti d'interesse considerati, cioè tipi di aerei e macchine.

Il punto più delicato dell'intera catena di processing, come evidenziato in altri punti del presente lavoro (capitolo 1, paragrafo 1.2), riguarda il pre-processing del dato. Questo perché, proprio attraverso questa fase si riesce a migliorare la qualità dell'immagine, consentendo all'algoritmo di processing che interviene successivamente, di operare nelle migliori condizioni possibili per raggiungere l'obiettivo desiderato. Inoltre, in questo caso è proprio attraverso la fase di *pre-processing* che riusciamo

ad isolare e binarizzare gli oggetti d'interesse, calcolando in questo modo i *momenti geometrici invarianti di Hu* su modelli meno rumorosi.



**Figura 4.3.** Tratto di autostrada Monaco - Salisburgo (da sensore aereo avente 50 cm di risoluzione, 2008) usata per il monitoraggio del traffico. (Cortesia DLR).



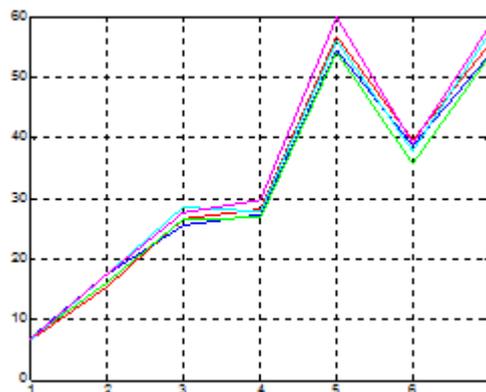
**Figura 4.4.** Immagine pan-cromatica del satellite EROS-A (1,8m di risoluzione spaziale). Cimitero di aerei in Tucson, Arizona del 2005. Il riquadro in basso-sinistra, evidenzia un aereo tipo Concorde da riconoscere automaticamente nei vari punti dell'immagine attraverso i suoi momenti geometrici invarianti.

La tabella 4.2 riporta i valori dei *momenti geometrici invarianti di Hu* (sette), calcolati per 6 tipi di aerei diversi presenti sulla scena. Questi valori, rappresentano il *vettore delle caratteristiche* identificativo di ciascun oggetto. Queste vettore di caratteristiche costituisce, per così dire, la “firma geometrica” dell’oggetto, e vengono usati per costruire il database del sistema e riconoscere l’oggetto stesso. Gli stessi valori, sette per ogni campione, possono essere visualizzati in modo alternativo, attraverso un grafico che collega il momento al logaritmo del suo valore assoluto (figura 4.5).

Momenti Invarianti di Hu (logaritmo)					
$\Phi_1$	6,6472	6,8767	6,8564	6,6359	6,8259
$\Phi_2$	15,3317	17,4817	16,003	17,4693	17,4728
$\Phi_3$	26,6570	25,4601	26,6172	28,4647	27,5590
$\Phi_4$	28,0771	27,2637	26,7054	27,8395	29,6928
$\Phi_5$	56,7061	54,4621	54,0388	56,0229	59,8769
$\Phi_6$	39,5234	38,7543	35,7665	37,7881	39,077
$\Phi_7$	55,5748	53,7531	53,6465	57,3914	58,4260

**Tabella 4.2.** *Momenti Geometrici Invarianti di Hu calcolati per 5 tipi di aerei presenti nell’immagine EROS A di figura 4.4*

Analogamente a quanto fatto per gli aerei, la procedura di pre-processing viene ripetuta anche sull’immagine aerea DLR. Questa sequenza di passi, migliora la qualità dell’immagine e ci consente successivamente, di isolare, binarizzare e calcolare i momenti di Hu relativi a tre tipi di macchine: piccole auto, furgoni e tir. I momenti così calcolati (tabella 4.3), vengono archiviati nel database del sistema.

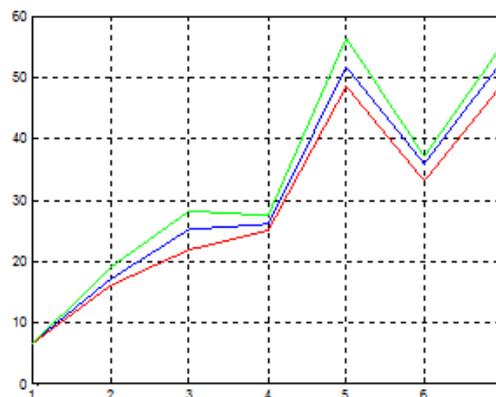


**Figura 4.5.** *Momenti Geometri Invarianti di Hu della tabella 2 riportati sotto forma di grafico (N° del momento vs logaritmo valore assoluto). Colore diverso per i momenti dei vari pattern considerati.*

Momenti Invarianti di Hu (logaritmo)			
$\Phi_1$	6,5224	6,7954	6,5629
$\Phi_2$	16,0084	17,0454	18,9877
$\Phi_3$	21,7922	25,2601	28,1601
$\Phi_4$	24,9854	25,9660	27,5127
$\Phi_5$	48,5011	51,6120	56,3601
$\Phi_6$	33,1443	35,8077	37,0446
$\Phi_7$	49,1220	52,9549	55,5091

**Tabella 4.3.** Momenti Geometrici Invarianti di Hu calcolati per 3 tipi di macchine (tir, furgoni e vetture) presenti nell'immagine di figura 4.3.

A causa della diversità delle immagini considerate (caratteristiche degli oggetti, distribuzione dell'istogramma, rumore, ecc), il pre-processing si ottiene implementando una procedura dedicata semi-automatica, che conduce al calcolo dei momenti per i modelli considerati. Questa considerazione vale in generale, poiché abbiamo generalizzato il sistema a qualunque tipo d'immagine. Ma questo limite può essere superato, costruendo per esempio un sistema dedicato a un certo tipo di sensore, ad un certo tipo di scena (come il controllo di qualità industriale), ecc.



**Figura 4.6.** Momenti Geometrici Invarianti di Hu della tabella 2 riportati sotto forma di grafico (come fig. 4.5). I momenti dei tre oggetti ricercati (macchine, furgoni e tir) sono mostrati insieme sullo stesso grafico con tre colori diversi.

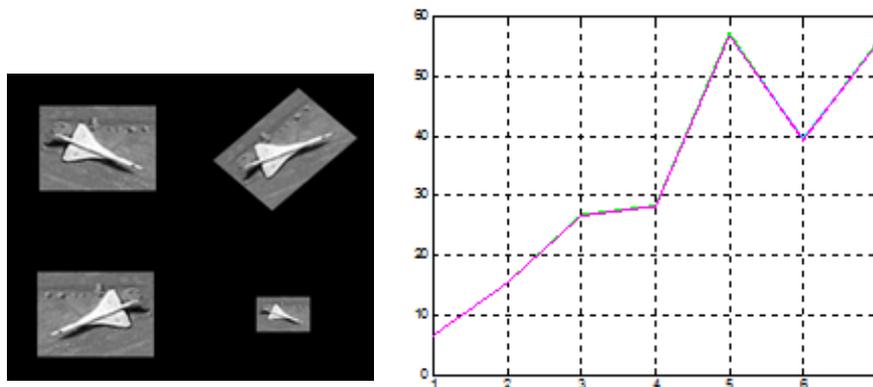
Migliorata la qualità dell'immagini d'input, attraverso opportuna fase di pre-processing, e creato il database del sistema siamo in condizioni di realizzare il sistema di *pattern recognition statistico*, per il riconoscimento automatico dei nostri oggetti d'interesse (tipi di aerei e macchine).

L'implementazione delle fasi successive del *sistema statistico* basato sui *momenti invarianti* (figura 4.2), sono state realizzate interamente in

ambiente MATLAB. Il sistema così realizzato, risulta completamente **automatico** ed è in grado di riconoscere con elevatissima accuratezza, i modelli ricercati in questa sede.

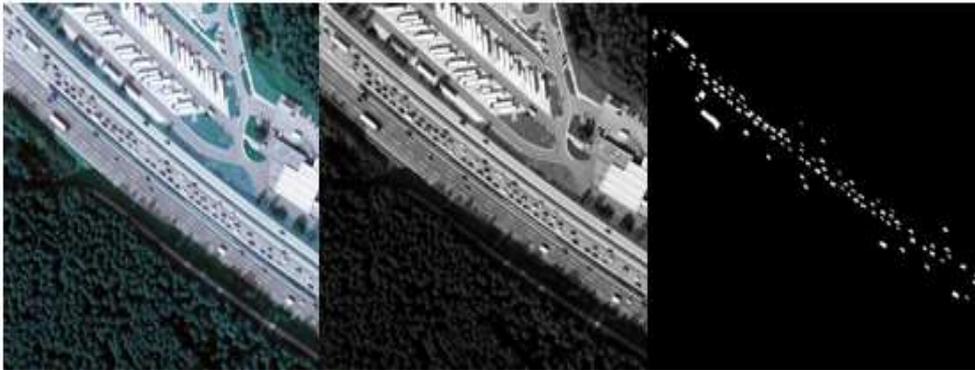
I risultati sono stati ottenuti trovando il match migliore (minima distanza fra vettori), tra i modelli isolati nelle immagini considerate in questo studio, e i vettori delle caratteristiche contenuti nel database. Nel caso degli aerei, il Concorde è stato trovato senza problemi (stesso discorso per gli altri tipi di velivoli) in tutti i punti della scena dove era presente ).

La figura 4.7 mostra il perfetto allineamento dei momenti invarianti di Hu per il Concorde, calcolato nelle varie configurazioni studiate. Questo dimostra, come effettivamente non c'è nessun cambiamento dei momenti di Hu, indipendentemente da variazioni di scala, rotazione e traslazioni, dando notevole potenza alla tecnica di pattern recognition.

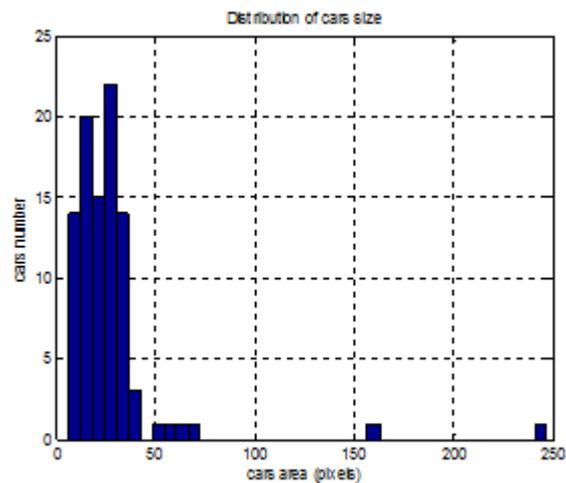


*Figura 4.7. Modello (sinistra) in varie configurazioni (scalato, ruotato e traslato) e il perfetto allineamento dei momenti invarianti di Hu (destra) calcolati in tutte le configurazioni considerate.*

Analogamente si procede nel caso del monitoraggio del traffico presente sulla autostrada, attraverso l'individuazione delle macchine presenti sulla striscia di asfalto interessato, si classificano le stesse per tipo e si esegue il conteggio automatico per arrivare alla stesura delle relative statistiche. La figura 4.8, mostra alcune immagini relative alla sequenza di passi che portano dall'immagine originale fino all'identificazione delle macchine presenti sull'autostrada. Infine, la figura 4.9 rappresenta la distribuzione di macchine individuate in funzione della loro dimensione in termini di pixel, raggiungendo una accuratezza del 100% in questo caso, completando così la catena di monitoraggio automatico del traffico implementata.



**Figura 4.8.** Immagine originale RGB dell'autostrada (sinistra), immagine a toni di grigio ottenuta dalla sola banda verde della stessa (in mezzo) e il risultato del processing con il riconoscimento delle autovetture presenti sulla stessa (destra).



**Figura 4.9.** Distribuzione del numero di macchine individuate sull'autostrada in funzione della dimensione delle stesse in termini di pixel. Il conteggio delle macchine viene realizzato in modo automatico dall'algoritmo MATLAB.

Il sistema di *pattern recognition statistico* basato sui momenti invarianti di Hu presentato in questo studio è il più generale possibile, lasciando quindi aperta la strada all'implementazione di un sistema analogo nei diversi campi tecnico-scientifici, a patto che si costruisca una fase di pre-processing dedicata che renda il sistema completamente automatico.

## 4.4 Ulteriori Considerazioni sui Momenti

### 4.4.1 Affidabilità dei Momenti

I momenti invarianti illustrati nel precedente paragrafo dimostrano di essere perfettamente invarianti nel caso di immagini continue e sotto la condizione di assenza di rumore. *Per immagini discrete, e in presenza di rumore, i valori dei momenti non risultano essere perfettamente invarianti.* Per questo motivo è importante investigare quanto i vari tipi

di momenti sono sensibili al rumore e al processo di discretizzazione dell'immagine continua.

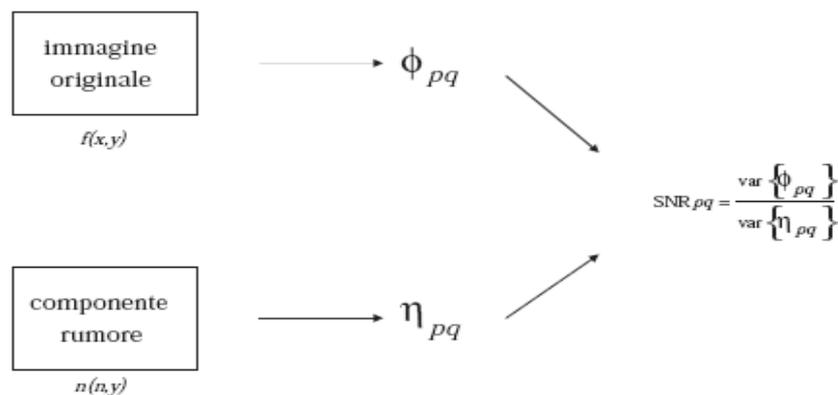
Con un esperimento effettuato per analizzare le proprietà dei vari momenti, Teh e Chin [28] hanno esaminato la sensibilità al rumore e la ridondanza delle informazioni dei vari momenti. In base ad alcune considerazioni, queste prove sperimentali hanno dedotto che i momenti ortogonali (ovvero i momenti di Zernike, di pseudo-Zernike e di Legendre), risultano essere migliori di altri tipi di momenti in termini di ridondanza dell'informazione e di capacità discriminatoria. Ciò è dovuto, come accennato in precedenza, alla base ortogonale di questo tipo di momenti.

#### 4.4.2 Sensibilità al rumore

Al fine di valutare l'effetto provocato dal rumore sui momenti di ordine diverso, e per comparare in che modo i vari tipi di momenti ortogonali sono affetti dal rumore, Teh e Chin hanno derivato [28] delle espressioni basate sulla statistica del secondo ordine che valutano l'impatto che provoca il rumore su questi momenti. Questi hanno considerato una funzione che descrive una immagine  $f(x,y)$ , definita per  $x$  e  $y$  in un piano limitato  $\tau$ , dove per convenienza, i valori che questa assume nel piano abbiano media nulla, quindi

$$\mu_{\tau} \{f(x,y)\} = 0;$$

applicandovi una componente di rumore  $n(x,y)$ , anch'essa con media nulla, viene ricavata l'immagine rumorosa. Vengono ricavati i momenti applicati sia all'immagine originale, denotati con  $\Phi_{pq}$  che alla componente di rumore, denotati con  $\eta_{pq}$  (figura 4.10). E' da notare che anche questi momenti assumono valori a media nulla sul piano.



**Figura 4.10.** Calcolo del rapporto SNR segnale/rumore.

Il rapporto segnale/rumore, denotato con  $SNR_{pq}$ , viene stimato come

$$SNR_{pq} = \frac{var\{\Phi_{pq}\}}{var\{\eta_{pq}\}} \quad (4.20)$$

dove  $var$  denota la varianza. Per un approfondimento dello studio consultare [28]. Diamo una valutazione di queste espressioni, riportando nella figura 4.11 un grafico che mostra l'andamento di queste espressioni per i principali momenti discussi: *Si deduce che la tendenza generale è che i momenti di ordine più alto sono più sensibili al rumore rispetto a quelli di ordine più basso, in quanto il rapporto segnale/rumore è a favore di questi ultimi.*

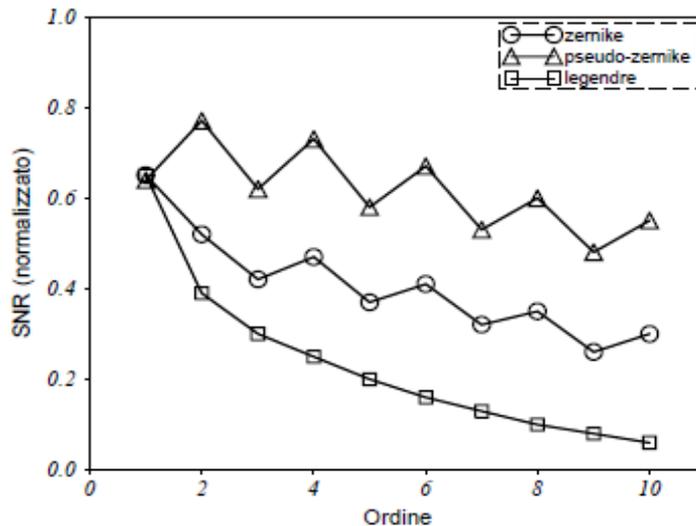
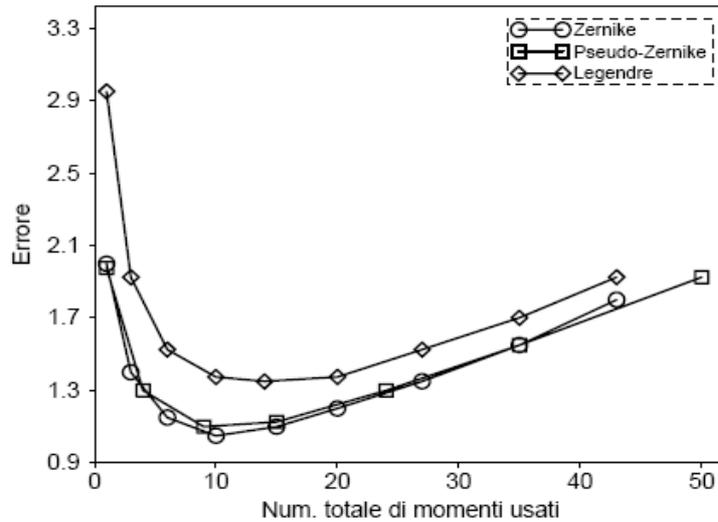


Figura 4.11. Confronto del SNR per differenti momenti.

### 4.4.3 Rappresentatività dei Momenti

Un'altra questione che va analizzata è la seguente: quanto e come una generica immagine può essere descritta da un insieme (piccolo) finito di questi momenti ?. Ancora Teh e Chin hanno eseguito delle prove, considerando come misura di rappresentatività dei momenti l'errore quadratico medio tra l'immagine originale e la ricostruzione di questa attraverso i momenti (abbiamo visto nei par. precedenti che una immagine può essere ricostruita utilizzando un numero infinito di momenti).

Dato che andiamo ad analizzare immagini discrete, e quasi sempre contenenti una certa quantità di rumore, ci si aspetta che i momenti dell'immagine diano dei valori errati, generando un errore quadratico medio grande nella ricostruzione di questa. Poiché abbiamo visto prima che i momenti di ordine più alto sono quelli più sensibili al rumore, è facile ipotizzare che l'immagine viene meglio ricostruita (quindi rappresentata) utilizzando momenti fino ad un certo ordine ottimo. In figura 4.12 [28] è illustrato un grafico, che mostra l'errore quadratico medio che viene provocato nella ricostruzione di un'immagine a partire dai momenti della stessa. Per l'analisi è stata scelta un'immagine, è stata aggiunta una certa quantità di rumore, e infine sono stati calcolati i momenti dell'immagine rumorosa.



**Figura 4.12.** Errore generato durante la ricostruzione di una immagine dai momenti (Teh e Chin).

Poiché per ricostruire un'immagine sono stati utilizzati tutti i momenti disponibili fino ad un dato ordine, nell'asse delle ascisse è riportato  $n_{tot}$ , che è il numero totale di momenti di ordine  $\leq n$  usato per la ricostruzione. Questo è dato da [28]:

$$n_{tot} = \frac{(n+1)(n+2)}{2} \quad (4.21)$$

per i momenti di Legendre e Zernike che devono rispettare, come da definizione, il vincolo  $n - |m| = \text{pari}$ , e da:

$$n_{tot} = (n+1)^2 \quad (4.22)$$

per i momenti di pseudo-Zernike, nei quali il suddetto vincolo non deve essere rispettato. Sull'asse delle ordinate invece, è riportato l'errore quadratico medio, normalizzato fra l'immagine originale e la sua versione ricostruita con i momenti.

Nel grafico appare infatti che l'errore causato dalla ricostruzione di una immagine rumorosa mediante i momenti raggiunge un minimo, e da questo punto in poi utilizzando momenti di ordine più alto inizia ad aumentare. Nel caso riportato in figura 4.12, all'immagine originale è stata aggiunta una quantità di rumore tale che l'immagine rumorosa riporti un rapporto segnale/rumore di 30dB rispetto all'originale, ed appare evidente che utilizzando momenti a partire da un certo  $n_{tot}$  in poi, ad esempio per i momenti di pseudo-Zernike utilizzando più di 9 momenti (ovvero momenti fino al secondo ordine), invece di migliorare la ricostruzione, degrada la qualità dell'immagine ricostruita, e quindi la



rappresentazione. Il grafico riportato in figura 4.12 si riferisce, come accennato, alla funzione di una immagine che ha un rapporto S/R di 30 dB, quindi molto rumorosa. E' da sottolineare comunque che il valore minimo che assume l'errore sull'immagine ricostruita aumenta al crescere del rapporto S/R, ovvero con immagini più pulite.

La conclusione che si trae dall'analisi condotta precedentemente, è che i momenti di ordine più alto sono più sensibili al rumore, ma nello stesso tempo sono i componenti riconoscitivi più cruciali, specialmente se si tratta di riconoscere oggetti che differiscono di poco. D'altro canto, l'utilizzo di momenti di ordine troppo elevato può peggiorare il riconoscimento, piuttosto che migliorarlo, a causa della alta sensibilità al rumore di questi. Questo fattore di conflitto pone un limite superiore sull'ordine massimo dei momenti da usare in un problema di riconoscimento. Esistono altri metodi, diverso da quello proposto da Teh e Chin, che ci consentono di ricavare questo valore in maniera sperimentale.



## Capitolo 5

# Approccio Strutturale – Sintattico: Morfologia Matematica.

In molti problemi di *pattern recognition* che coinvolgono oggetti complessi, una soluzione può essere quella di considerare una prospettiva gerarchica, dove i pattern vengono visti come composti da *subpattern* più semplici [10, 38]. I *pattern elementari* o *subpattern* sono chiamate *primitive*, e i pattern complessi da riconoscere sono rappresentati sotto forma di insiemi e relazioni tra primitivi. Nei sistemi di *pattern recognition* sintattici si crea un'analogia formale fra la struttura dei pattern complessi e la sintassi di un linguaggio.

Quindi un grosso insieme di pattern complessi può essere descritto da un semplice insieme di primitive e regole grammaticali. La grammatica per ogni classe di pattern, può essere inferita o appresa dall'insieme di pattern di addestramento o di esempio per ognuna delle classi di interesse. Inoltre, la *pattern recognition* realizzata mediante l'approccio strutturale (o sintattico) è molto interessante perché, oltre alla classificazione dei pattern, fornisce anche una descrizione di come essi sono costruiti a partire dalle primitive, da qui l'uso della parola *strutturale* [10].

Comprendere e rappresentare la forma delle strutture è di fondamentale importanza in tutti gli ambiti scientifici e tecnologici che necessitano di una caratterizzazione delle proprietà geometriche e morfologiche degli oggetti. La computer vision, la computer graphics, *l'mage processing*, il pattern recognition, ecc sono tipici contesti in cui la conoscenza della forma, trattata in base ad ipotesi molto differenti, è necessaria per risolvere problemi di analisi, riconoscimento o semplificazione, con scopi e requisiti molto simili.

Strumenti di analisi e rappresentazione delle forme sono utili nella compressione di modelli geometrici molto complessi in termini di primitive geometriche utilizzate. In questo contesto, infatti, la conoscenza delle caratteristiche principali della forma, o *features*, e la loro configurazione, è importante per una strategia di semplificazione



che lasci da parte aspetti irrilevanti del modello senza perdere l'aspetto completo.

La ricerca basata sul contenuto nei database ad oggetti e il processing basato sulla formastanno convogliando un'attenzione crescente alla formalizzazione della forma come combinazione di geometria e semantica, e come strumento per valutare la somiglianza tra modelli differenti.

In entrambi i casi, la **topologia**, come studio delle proprietà che sono invarianti per trasformazioni continue di un oggetto, sembra uno strumento appropriato per affrontare questo tipo di problemi. Infatti, queste considerazioni hanno portato negli ultimi anni, ad aumentare l'attenzione rivolta alla cosiddetta topologia computazionale, una branca della topologia che ha l'obiettivo di affrontare i problemi topologici senza perdere di vista il problema della fattibilità computazionale. Oltre a strutture a linea, come la *medial axis* per le immagini, sono state introdotte diverse altre strutture a grafo di tipo lineare, denominate *skeleton*, che riescono a descrivere oggetti anche molto complessi tramite uno schizzo che mantiene le proprietà topologiche del modello di partenza. Il *grafo di Reeb* è un esempio significativo di struttura sia topologica che morfologica, che può essere usato come punto di partenza per la costruzione di modelli avanzati di oggetti geometrici. Il modello di forma ideale dovrebbe infatti preservare le caratteristiche topologiche, e garantire un buon rapporto tra efficienza computazionale e contenuto informativo ed efficacia della rappresentazione.

L'implementazione di un approccio strutturale nella *pattern recognition* è, in generale, piuttosto complessa a causa di diversi problemi. Le difficoltà legate al raggiungimento di una *segmentazione dell'immagine* ottima, può condizionare l'estrazione delle primitive e della grammatica dall'insieme di pattern di addestramento.

Esistono diverse tecniche che consentono di implementare un sistema di *pattern recognition* strutturale, come accennato nel Capitolo. Tuttavia, la ricerca teorica ed sperimentale ci ha condotto ad affrontare e risolvere il problema, ricorrendo ad una tecnica abbastanza innovativa, che va sotto il nome di **Morfologia Matematica** (MM). Nei paragrafi successivi introdurremo le basi teoriche della *Morfologia Matematica* [39], nonché una descrizione del suo funzionamento nell'analisi delle immagini con una formalizzazione matematica rigorosa degli operatori di base, detti morfologici, che possono essere usati per costruire strumenti utili in un'ampia varietà di problemi di analisi di immagini. Infine, concluderemo il capitolo illustrando gli algoritmi automatici di MM costruiti per affrontare diversi tipi di problemi, nonché i relativi risultati.



## 5.1 Morfologia Matematica

La *Morfologia Matematica* è stata definita inizialmente nello spazio euclideo, e la teoria corrispondente era chiamata morfologia euclidea. Tuttavia, la maggior parte delle tecnologie di analisi d'immagine usa dati digitali, e questo ha comportato l'estensione della MM [40] allo spazio discreto. La versione discreta dello spazio euclideo bidimensionale  $\mathbb{R}^2$  è  $\mathbb{Z}^2$ . Il processo di digitalizzazione o il passaggio dallo spazio continuo  $\mathbb{R}^2$  allo spazio discreto  $\mathbb{Z}^2$  è ottenuto campionando  $\mathbb{R}^2$ . In generale, si considera una rete di punti distribuiti uniformemente detti pixel (elementi della figura), e le immagini discrete sono definite associando un valore numerico ad ogni punto della rete di digitalizzazione.

Le immagini binarie si distinguono dalle immagini a toni di grigio per l'intervallo di valori dati ai punti o ai pixel della rete di digitalizzazione. Quando le operazioni di calcolo coinvolgono alcune relazioni di vicinanza, l'uso di una rete di digitalizzazione non è sufficiente. Il concetto di *grafo* permette la definizione di relazioni di vicinanza tra punti della rete di digitalizzazione.

Uno dei campi di applicazione in cui si è rivelato particolarmente efficace l'utilizzo della *morfologia matematica*, è il filtraggio di immagini. Introduciamo il concetto di filtro di immagini presentando innanzitutto gli operatori *point* e *neighbourhood image*. Gli operatori *point image* modificano il valore di un punto dell'immagine indipendentemente dai valori dei punti vicini ad esso. Un tipico esempio è l'operazione che consiste nel calcolare la visione negativa di un'immagine, cioè un'immagine che abbia luci ed ombre invertite. Il valore di output di un operatore *neighbourhood image*, applicato ad un punto dell'immagine dato è ottenuto combinando i valori dell'immagine che giacciono in vicinanza del punto considerato. Per esempio, si può sostituire il livello di grigio di ogni punto dell'immagine con il valore medio dei punti vicini ad esso. Definiamo un filtro di immagini come un operatore *neighbourhood*. In questo senso, i filtri di immagini possono eseguire un'ampia varietà di compiti, come la riduzione del rumore, la rilevazione dei bordi, la compensazione per messa a fuoco scorretta e blurring dovuto al moto. La MM fornisce filtri di immagini [40] per compiere i primi due compiti. Inoltre, i filtri morfologici sono adatti per l'estrazione o la soppressione di oggetti o strutture dell'immagine. In pratica, la maggioranza dei filtri morfologici è realizzata utilizzando alcune conoscenze sulla *forma e sulle proprietà geometriche degli oggetti dell'immagine*.

L'idea che sta alla base della maggior parte degli operatori morfologici è la seguente: ***le strutture dell'immagine sono osservate attraverso un***

*piccolo campione, e sono “mantenute” o “alterate” a seconda che il campione le ricopra oppure no.*

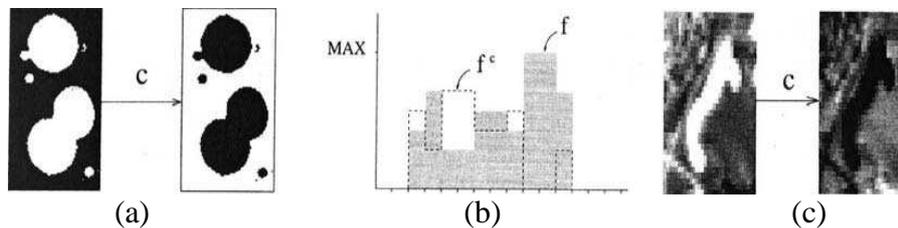
## 5.2 Operatori d'Insieme applicati alle Immagini

Gli operatori di insieme elementari usati sono l'unione  $\cup$  e l'intersezione  $\cap$ . Per le immagini in toni di grigio, l'unione diventa l'operatore *point-wise maximum* e l'intersezione è sostituita dall'operatore *point-wise minimum*. Il point-wise maximum  $\vee$  e il point-wise minimum  $\wedge$  tra due immagini  $f$  e  $g$ , che hanno lo stesso dominio di definizione, sono definiti come segue per ogni punto  $x$ :

$$(f \vee g)(x) = \max [f(x), g(x)] \quad (5.1)$$

$$(f \wedge g)(x) = \min [f(x), g(x)] \quad (5.2)$$

Un altro operatore di insieme di base è il *complemento*. Il complemento di un'immagine  $f$ , indicato con  $f^c$ , è definito per ogni pixel  $x$ , come il massimo valore del tipo di dato utilizzato per memorizzare l'immagine, meno il valore dell'immagine  $f$  nella posizione  $x$ . Il complemento di un'immagine binaria, di un segnale monodimensionale e di un'immagine a livelli di grigio è mostrato nella figura 5.1.



**Figura 5.1.** Complemento di un'immagine binaria (a), di un segnale monodimensionale (b) e di un'immagine 2-D in toni di grigio (c).

Dati due insiemi  $X$  e  $Y$ , la loro differenza,  $X-Y$ , è definita come l'intersezione tra  $X$  e il complemento di  $Y$ . La differenza tra insiemi si applica solo ad immagini binarie. La generalizzazione alle immagini a toni di grigio come  $f \wedge g^c$ , ha la proprietà indesiderabile che i valori di output dipenderanno dal valore  $t_{max}$  utilizzato per complementare  $g$ .

La traslazione di un'immagine  $f$  mediante un vettore  $b$ , è indicata con  $f^b$ . Il valore dell'immagine traslata in un dato pixel  $x$  è uguale al valore dell'immagine originale nella posizione traslata con il vettore opposto:  $f_b(x) = f(x - b)$ .

Per definizione, la traslazione di un vettore  $b$  può essere calcolata se e solo se la rete di digitalizzazione traslata coincide con la rete di

digitalizzazione originale. Di conseguenza, le traslazioni sono applicate solo ad immagini definite su una rete di punti uniformemente distribuiti. L'operatore di traslazione è illustrato nella figura 5.2 per un segnale monodimensionale e per il dominio di definizione di un'immagine 2-D. Tutti gli operatori morfologici si basano sulla sola combinazione degli operatori di intersezione (point-wise minimum), unione (point-wise maximum), complemento e traslazione.

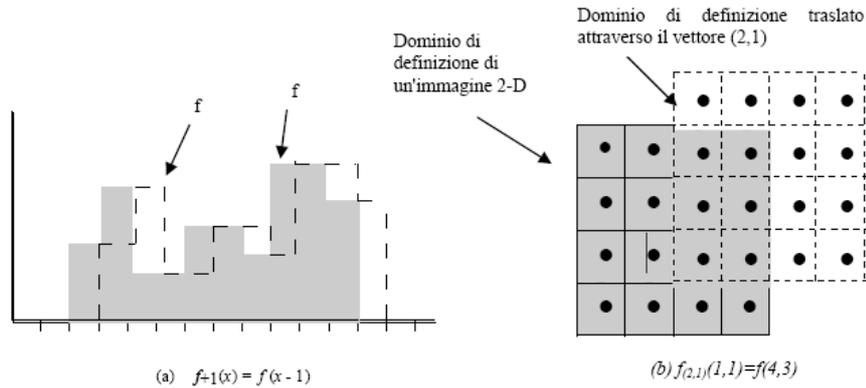


Figura 5.2. Traslazione di un segnale 1-D (a) e di un'immagine 2-D (b).

### 5.3 Relazioni d'Ordine

Un'immagine  $f$  è minore o uguale a un'immagine  $g$  con lo stesso dominio di definizione se il valore di  $f$  è minore o uguale al valore di  $g$  per tutti i pixel  $x$  del dominio di definizione comune. Analogamente, se si definisce la *cross-section* a livello  $t$  di un'immagine in toni di grigio  $f$ , come l'insieme dei pixel dell'immagine i cui valori sono maggiori o uguali a  $t$  e si denota questo insieme con  $CS_t(f)$ , si può dire che per ogni livello di grigio  $t$  la cross-section di  $f$  a livello  $t$ , è inclusa nella cross-section di  $g$  a livello  $t$ :

$$f \leq g \Leftrightarrow \forall x, f(x) \leq g(x) \Leftrightarrow \forall t, CS_t(f) \subseteq CS_t(g) \quad (5.3)$$

In altri termini,  $f$  è minore o uguale a  $g$  se il sottografo di  $f$  è incluso in quello di  $g$ :

$$f \leq g \Leftrightarrow SG(f) \subseteq SG(g) \quad (5.4)$$

L'ordinamento è un concetto chiave in *morfologia matematica*, e verrà utilizzato in seguito per la costruzione di un grafo topologico dell'immagine. *La maggior parte delle trasformazioni morfologiche preserva l'ordinamento sulle immagini.*

L'insieme di tutte le immagini con la relazione d'ordine  $\leq$  non è *totalmente* ordinato, dal momento che esistono coppie di immagini che non sono ordinate, cioè coppie  $\{f, h\}$  tale che ne  $f \leq h$  ne  $f \geq h$ .



Comunque, l'insieme di tutte le immagini soddisfa tutte le proprietà di un insieme *parzialmente* ordinato:

- i)  $f \leq f$ ;
- ii)  $f \leq g, g \leq f \Leftrightarrow f = g$ ;
- iii)  $f \leq g, g \leq h \Rightarrow f \leq h$ .

## 5.4 Trasformazioni da immagine a immagine

Le trasformazioni morfologiche tra immagini sono *trasformazioni da immagine* ad immagine, cioè l'immagine trasformata ha lo stesso dominio di definizione dell'immagine di input ed è ancora una mappatura di questo dominio di definizione nell'insieme degli interi non negativi.

Una trasformazione da immagine ad immagine ampiamente utilizzata è l'*operatore di threshold*, il quale fissa tutti i pixel dell'immagine di input che giacciono in un dato intervallo di valori di grigio, al valore 1 ed i rimanenti al valore 0. Questa tecnica talvolta è chiamata *level slicing*.

In termini matematici, l'operatore di threshold  $T$  assegna ad ogni pixel  $x$  dell'immagine di input  $f$ , il cui valore sia compreso nell'intervallo  $[t_i, t_j]$ , il valore 1 e a tutti gli altri pixel il valore 0:

$$\left[ T_{[t_i, t_j]}(f) \right](x) = \begin{cases} 1, & \text{se } t_i \leq f(x) \leq t_j, \\ 0, & \text{altrimenti} \end{cases} \quad (5.5)$$

Ne consegue che l'*operatore di threshold* mappa ogni immagine a toni di grigio in un'immagine binaria.

Una trasformazione da immagine ad immagine  $\Psi$  è *invariante per decomposizione threshold*, se può essere scritta come somma delle trasformazioni delle cross-section. Molte trasformazioni morfologiche sono invarianti per decomposizione threshold.

Il *dominio di invarianza* della trasformazione di una data immagine è l'insieme di immagini che non sono modificate da questa trasformazione. Per esempio, il dominio di invarianza della trasformazione identità è l'insieme di tutte le immagini. In alcuni casi, è necessario iterare le trasformazioni di immagini: l'immagine trasformata è utilizzata come input per una seconda trasformazione, ecc..

Tali trasformazioni sono *trasformazioni dei punti dell'immagine*, cioè il valore di output di un dato pixel dipende solo dal valore di questo pixel nell'immagine di input [41]. Le trasformazioni morfologiche, invece, sono *trasformazioni degli intorni dell'immagine*. In questo caso, il



valore di output di un dato pixel è funzione dei valori dei pixel che si trovano in una regione contigua centrata nel pixel considerato.

## 5.5 Trasformazioni Morfologiche

Gli operatori morfologici mirano ad estrarre strutture rilevanti dell'immagine, considerata attraverso il suo sottografo di rappresentazione. Questo obiettivo è ottenuto esplorando l'immagine d'input con un altro insieme di forma nota detto *elemento strutturante* (ES). La forma dell'ES, di solito, è scelta in accordo con alcune conoscenze a priori della geometria delle strutture dell'immagine rilevanti ed irrilevanti. Per strutture irrilevanti intendiamo o il rumore o gli oggetti che vorremmo sopprimere.

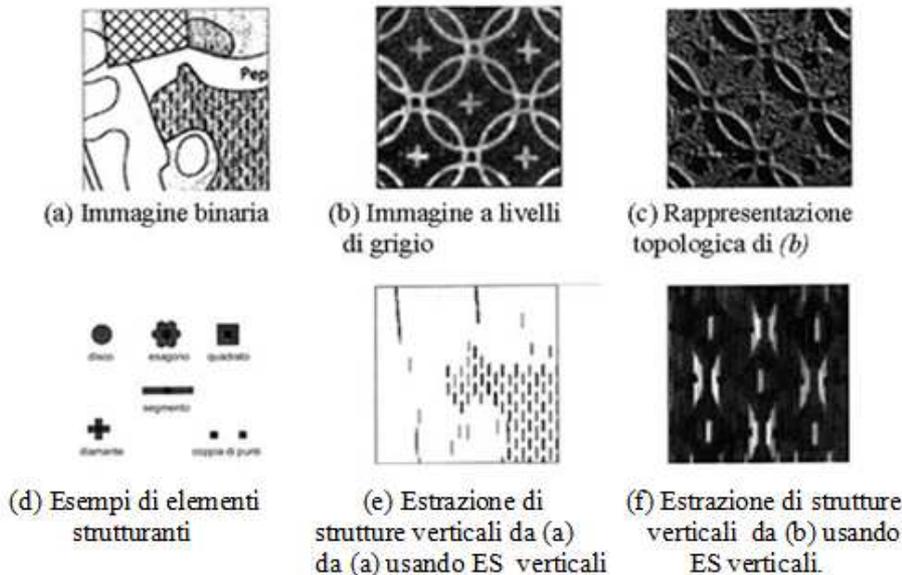
L'approccio morfologico al processing di immagini è illustrato nella figura 5.3. Le figure 3a e 3b sono due immagini d'input, binaria e a toni di grigio rispettivamente. La figura 3c è utilizzata solo per scopi di rendering: rappresenta una visione ombreggiata della figura 3b per arricchire la sua rappresentazione topografica. Nella figura 3d sono mostrati alcuni esempi di elementi strutturanti: un cerchio, un esagono, un quadrato, un segmento lineare orizzontale ed una coppia di punti. Se utilizziamo come sonda un ES verticale, possiamo estrarre tutte le strutture verticali delle immagini binarie e a toni di grigio (figure 3e e 3f). Le trasformazioni che sono state usate in questo esempio, si basano su combinazioni di due operatori morfologici di base: *l'erosione* e la *dilatazione*. Questi operatori duali sono, per così dire, le due lettere dell'alfabeto morfologico, dal momento che tutti gli altri operatori sono espressi in termini di erosioni e dilatazioni.

### 5.5.1 Elemento Strutturante (ES)

Un *elemento strutturante* (ES) [39, 40] non è altro che un piccolo insieme (immagine sonda) utilizzato per indagare l'immagine sotto studio. Analogamente, ES di dimensione  $n+1$  sono chiamati *volumetrici*, *non-flat* o a *toni di grigio*, e corrispondono a piccole immagini a toni di grigio. La forma degli elementi strutturanti *flat* non dipende dalla rappresentazione in scala dell'immagine a livelli di grigio. Quando si utilizzano ES *non-flat*, i loro valori in toni di grigio dovrebbero avere le stesse unità di misura e di scala dell'immagine di input.

Gli operatori morfologici fondamentali richiedono la definizione di un'origine per ogni ES. Questa origine permette il posizionamento dell'ES in un certo punto o pixel: un ES nel punto  $x$  significa che la sua origine coincide con  $x$ . Concentreremo la nostra attenzione sulla morfologia utilizzando ES *flat*, cioè elementi strutturanti che hanno la stessa dimensione topologica del dominio dell'immagine. La forma e la dimensione dell'ES deve essere adatta alle proprietà geometriche degli

oggetti dell'immagine che devono essere processati. Per esempio, elementi strutturanti lineari sono adatti per l'estrazione di oggetti lineari.



*Figura 5.3. Morfologia matematica nell'immagine processing: tanto le immagini binarie quanto le immagini in toni di grigio sono considerate come insieme. Un altro insieme di forma nota, chiamato "elemento strutturante", è utilizzato per studiare l'immagine di partenza e quindi consente di evidenziare le strutture che abbiano forma simile.*

Qualsiasi forma può essere scelta come elemento strutturante. Tuttavia, nelle applicazioni pratiche si utilizza solo un'insieme limitato di elementi strutturanti. A parte la forma, si può anche definire la dimensione e, per alcuni elementi strutturanti, una specifica orientazione. Quando si processano i grafi, il concetto di forma e di orientazione scompare, dal momento che l'ES è definito semplicemente in base alle relazioni di vicinanza di ogni vertice. Perciò non è necessario scegliere un ES.

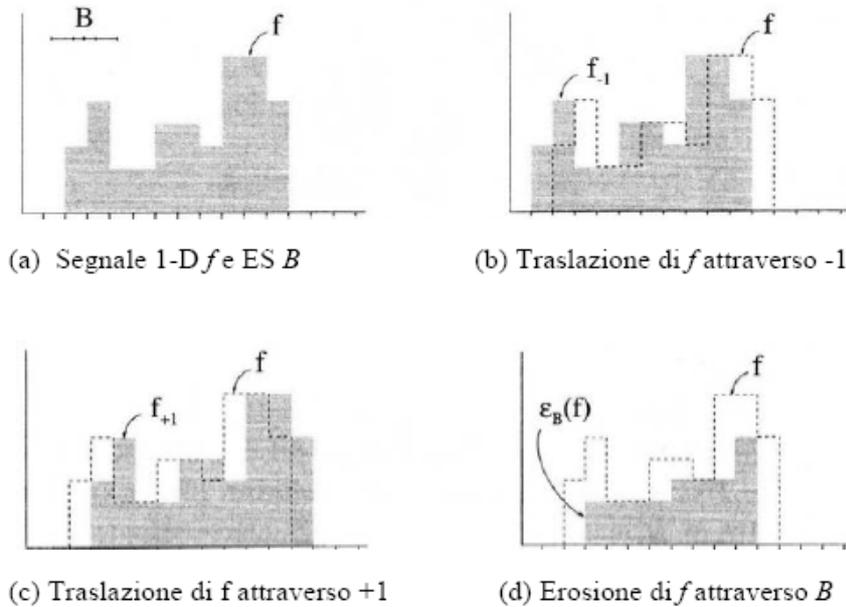
### 5.5.2 Operatore *Erosione*

La prima domanda che può nascere quando si esplora un insieme (immagine) con un elemento strutturante è: "*L'elemento strutturante è incluso nell'insieme*" ? L'insieme eroso è il luogo dei punti dove la risposta a questa domanda è affermativa. L'insieme studiato rappresenta o gli oggetti di un'immagine binaria o il sottografo di un'immagine a toni di grigio.

L'operatore *erosione* di un insieme  $X$  attraverso un elemento strutturante  $B$  è indicato con  $\varepsilon_B(X)$ , ed è definito come il luogo dei punti  $x$  tali che  $B$  è incluso in  $X$  quando la sua origine è in  $x$ :

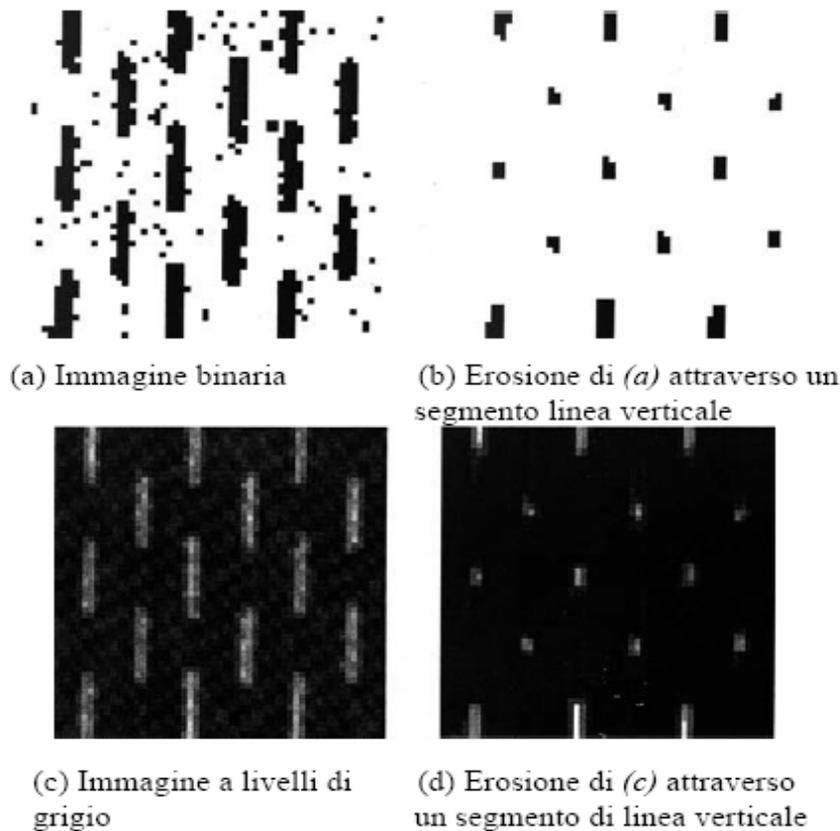
$$\varepsilon_B(X) = \{ x \mid B_x \subseteq X \} \quad (5.6)$$

Analogamente, l'erosione può essere riscritta in termini di intersezione di traslazioni di insiemi e può essere estesa direttamente alle immagini binarie e a livelli di grigio. L'erosione in termini di intersezioni di traslazioni di immagini è illustrata nella figura 5.4. In questa figura l'elemento strutturante è un segmento di tre pixel, e l'origine dell'ES è nel pixel centrale.



**Figura 5.4.** Erosione di un segnale 1-D. L'elemento strutturante  $B$  definisce 3 traslazioni di immagine:  $f$ ,  $f_{-1}$  e  $f_{+1}$ . L'erosione  $\varepsilon$  di  $f$  attraverso  $B$  è uguale al minimo calcolato puntualmente fra le tre immagini. Il valore eroso di un certo pixel corrisponde anche al valore minimo dell'immagine nell'insieme dei pixel vicini definito dall'ES quando è centrato in tale pixel.

Nella figura 5.5 sono mostrati alcuni esempi di erosioni su immagini binarie e a livelli di grigio. In questa figura le barre verticali erose sono ridotte a pochi punti, perché l'elemento strutturante poteva ricoprire le barre solo quando era centrato su questi punti. Le barre connesse ai bordi in alto e in basso non sono così erose come quelle che giacciono all'interno dell'immagine, perché si è assunto che i valori fuori dal dominio di definizione dell'immagine siano tutti uguali a  $t_{\max}$ . Questo è un modo comune per trattare gli effetti di bordo quando si erode un'immagine. Elementi strutturanti non-flat, ad esempio volumetrici, hanno valori in scala di grigio per ogni punto del loro dominio di definizione. I pesi in scala di grigio di un ES volumetrico dovrebbero essere scelti in accordo con i valori di intensità dell'immagine. A causa della difficoltà di avere questa condizione e a causa dei calcoli extra richiesti dalla sottrazione, ES in scala di grigio sono utilizzati raramente nelle applicazioni pratiche.



*Figura 5.5. Erosione di un'immagine binaria e di un'immagine a livelli di grigio attraverso un segmento lineare verticale di 13 pixel.*

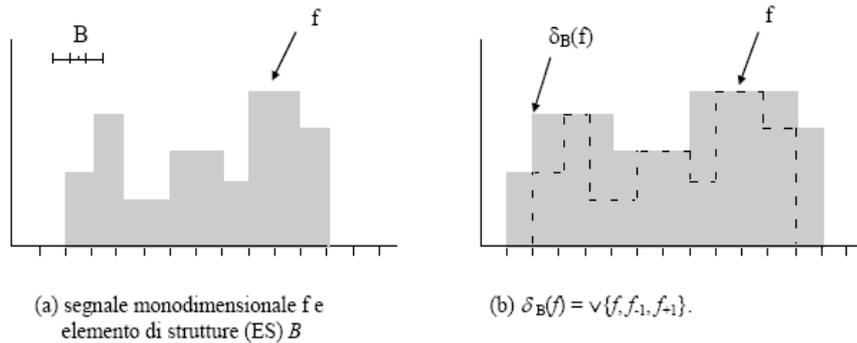
### 5.5.3 Operatore Dilatazione

La *dilatazione* è l'operatore duale dell'erosione e si basa sulla domanda seguente: "*L'elemento strutturante interseca l'insieme*"?. L'insieme dilatato è il luogo dei punti dove la risposta a questa domanda è affermativa.

La dilatazione di un insieme  $X$  attraverso un elemento strutturante  $B$  si indica con  $\delta_B(X)$ , ed è definito come il luogo dei punti  $x$  tali che  $B$  interseca  $X$  quando la sua origine coincide con  $x$ :

$$\delta_B(X) = \{x/B_x \cap X \neq \emptyset\} \quad (5.7)$$

Analogamente a quanto fatto per l'erosione, la dilatazione può essere scritta in termini di unione di traslazioni di insiemi e può essere direttamente estesa alle immagini binarie e a quelle in toni di grigio. In altri termini, il valore dilatato di un pixel  $x$  è il massimo valore dell'immagine nella finestra definita dall'elemento di struttura quando la sua origine è in  $x$ . La dilatazione di un segnale monodimensionale è mostrata nella fig. 5.6.



**Figura 5.6.** Dilatazione rappresentata come point-wise maximum di una serie di traslazioni di immagine. Il valore dilatato di un certo pixel corrisponde al valore massimo dell'immagine all'interno della regione di vicinanza definita dall'ES quando è centrato in quel pixel.

Esempi di dilatazioni di immagini binarie e a livelli di grigio sono mostrate nella figura 5.7. Notare che l'ES è abbastanza lungo da coprire i segmenti lineari della linea tratteggiata quando esso si trova tra due segmenti. Questo spiega perché le lacune sono completamente riempite nell'immagine dilatata.



**Figura 5.7.** Dilatazioni di una immagine binaria (a) e di immagine a livelli di grigio (b), attraverso un segmento verticale di 13 pixel (le immagini originali sono mostrate in fig. 5a e 5c rispettivamente).

A differenza delle immagini dilatate o erose attraverso elementi strutturanti flat, i valori dei livelli di grigio delle immagini dilatate o erose attraverso ES non-flat, non sono limitate dal valore minimo e massimo dei livelli di grigio dell'immagine di input. In particolare le erosioni con ES non-flat possono restituire come output interi negativi.

### 5.5.4 Operatori *Opening* e *Closing*

L'erosione di un'immagine non solo rimuove tutte le strutture che non possono contenere l'elemento strutturante, ma riduce anche le altre. La ricerca di un operatore che recuperi la maggior parte delle strutture perse attraverso l'erosione, conduce alla definizione dell'operatore morfologico *opening*. Il principio consiste nel dilatare l'immagine precedentemente



erosa, utilizzando lo stesso elemento strutturante. In generale, non tutte le strutture sono recuperate [41]. Per esempio, oggetti completamente distrutti dall'erosione non possono essere affatto recuperati. Questo é il comportamento alla base delle proprietà di filtraggio dell'operatore opening: le strutture dell'immagine sono selettivamente filtrate e la selezione dipende dalla forma e dalla dimensione dell'elemento strutturante. L'operatore duale dell'opening morfologico é il *closing morfologico*. Entrambi questi operatori sono alla base dell'approccio morfologico al filtraggio delle immagini.

## 5.6 Algoritmi morfologici di base in scala di grigio

Ora vediamo un elenco di alcuni algoritmi morfologici in scala di grigio largamente usati nell' image processing:

- **Smoothing Morfologico:** abbiamo visto che l'apertura (*opening*) e la chiusura (*closing*) permettono rispettivamente di eliminare i dettagli chiari e i dettagli scuri, quindi possono essere utilizzate come filtri morfologici per lo *smoothing* dell'immagine e la riduzione del rumore. A volte lo *smoothing* si ottiene tramite un *filtraggio sequenziale alternante*, cioè non viene applicata la sequenza apertura-chiusura inizialmente sull'immagine originale, ma i passaggi successivi sui risultati dei passi precedenti;
- **Gradiente morfologico:** è possibile ottenere il gradiente morfologico tramite la dilatazione e l'erosione combinate con la sottrazione dell'immagine. Infatti la dilatazione ispessisce le regioni, l'erosione le rimpicciolisce e la loro differenza permette di enfatizzare i contorni delle regioni ed eliminare le aree omogenee, quindi il risultato finale è un effetto gradiente di tipo derivativo;
- **Ricostruzione morfologica in scala di grigio:** in questo contesto abbiamo due immagini (*marker f*; *maschera g*) di dimensioni uguali, con  $f \leq g$  ed un elemento strutturante.

L'apertura tramite ricostruzione di immagini in scala di grigio, prima erode l'immagine in ingresso e quindi la usa come marker. L'obiettivo è quello di preservare la forma delle componenti dell'immagine dopo il processo di erosione. La chiusura tramite ricostruzione è possibile ottenerla attraverso il complemento dell'immagine, ottenendo l'apertura tramite ricostruzione e complementando il risultato.

## 5.7 Sistema di Pattern Recognition basato sulla Morfologia Matematica

La teoria esposta nei paragrafi precedenti lascia intravedere le enormi potenzialità che la Morfologia Matematica può avere nel campo del riconoscimento automatico degli oggetti. E infatti, anche se per necessità di sintesi tra i casi studiati presenteremo solo quelli più significativi, vale la pena ricordare che i casi affrontati e risolti con successo durante il periodo di dottorato, sono numerosi.

I sistemi di *pattern recognition morfologici* più significativi sviluppati riguardano:

- 1) il *monitoraggio dei campi di rifugiati in Africa*, per costruire una metodologia capace di stimare la popolazione degli stessi in modo accurato e indipendente;
- 2) il *monitoraggio di infrastrutture* (case, strade, ecc) pre e post eventi disastrosi, in modo da avere un fonte di informazione precisa a supporto degli enti e delle autorità preposte al coordinamento degli interventi e la ricostruzione.

## 5.8 Monitoraggio di Campi di Rifugiati in Africa

I dati e relativi prodotti derivati dall'Osservazione della Terra (EO), svolgono un ruolo chiave in molti problemi di natura geopolitica, per la loro capacità di generare informazioni fondamentali a livello politico e decisionale, che altrimenti sarebbe molto difficile da ottenere in modo regolare, imparziale e a bassi costi. I *campi di rifugiati* appartengono a questa sensibile e complessa categoria di problemi geopolitici, e possono trarre vantaggio dall'uso delle tecnologie aerospaziali [42].



**Figura 5.8.** Fotografia panoramica di un parte del Campo Profughi di Djabal (Chad, crisi del Darfour). Si può notare la struttura irregolare dello stesso con diversi tipi di unità abitative.



Il Commissario delle Nazioni Unite per i Rifugiati (UNHCR), ha stimato in 12 milioni il numero dei rifugiati nel 2001 [43]. Dello stesso ordine di grandezza è il numero di sfollati interni.

Di solito i rifugiati sono ospitati in campi che forniscono le infrastrutture essenziali per sopravvivere. Nonostante la scarsa qualità delle infrastrutture, i campi profughi hanno una durata di diversi anni e sono sostenuti, grazie al sostegno finanziario e logistico di donatori internazionali. Queste organizzazioni di supporto richiedono informazioni statistiche sui rifugiati per valutare e giustificare i fondi distribuiti. Tali informazioni sono, inoltre, necessarie ai governi ospitanti locali per valutare l'impatto sociale ed economico della loro presenza sul territorio. Inoltre, in caso di emergenze umanitarie, diversi metodi per la stima della popolazione e della loro distribuzione sono utilizzati, ai fini di generare sistemi di allarmi rapidi, valutare e programmare gli interventi di assistenza e di protezione della popolazione necessari, ecc [43].

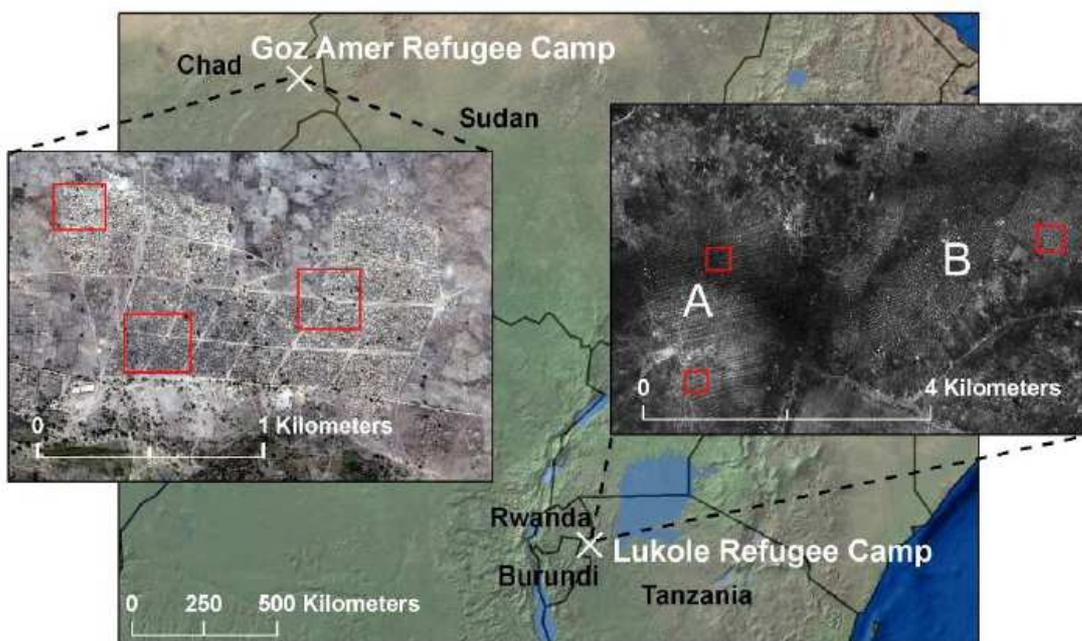
Diverse sono le tecniche che vengono utilizzate per stimare la popolazione totale di un campo profughi. Tali tecniche ricorrono al semplice conteggio di individui o rifugiati, ai documenti amministrativi, alle stime della comunità, alla cartografia, alle immagini aeree e spaziali, ecc. Ma ancora oggi, i metodi più adottati più frequentemente si basano su l'attività che il personale delle organizzazioni internazionali svolge sul campo. Tale personale, appartenente alle organizzazioni non governative (ONG) e alle agenzie governative, realizza il proprio compito attraverso un campionamento spaziale della popolazione, fatto su un sottoinsieme della popolazione di interesse, in modo da ottenere informazioni sull'intera popolazione. Il *campionamento spaziale*, consiste nell'uso di aree geografiche e delle relative densità di popolazione, da cui derivare in seguito una stima della dimensione e della densità totale occupante l'area di studio, spesso utilizzando un palmare con unità GPS e sistemi di informazione geografica (GIS).

In generale, i dati satellitari EO, possono opportunamente integrare e completare le informazioni tradizionalmente raccolte dal personale sul campo. Infatti, le immagini satellitari sono in grado di fornire una visione sinottica del campo, delle unità abitative e delle loro caratteristiche, consentendoci di conseguenza di giungere alla loro distribuzione e alla stima della popolazione. Inoltre, serie temporali di immagini possono essere utilizzate per valutare le variazioni ambientali, provocate dalla pressione esercitata dalla popolazione e le attività umane relative al campo.

Sulla base di queste premesse, la nostra attività di ricerca su questo argomento ha riguardato lo sviluppo e la implementazione (in ambiente

MATLAB) di una serie di algoritmi in grado di localizzare e contare, in modo automatico, le unità abitative presenti nei campi stessi. Gli algoritmi sviluppati si basano sulla teoria della *Morfologia Matematica* [39],[40],[41] descritta precedentemente. Tale applicazione della MM non rappresenta una novità [30], ma lo scopo principale in questo lavoro è quello di migliorare gli algoritmi esistenti (rendendoli più efficienti) e crearne di nuovi, in modo da poter rendere questa metodologia e gli strumenti derivati, quanto più possibile universale nell'affrontare questo tipo di problemi.

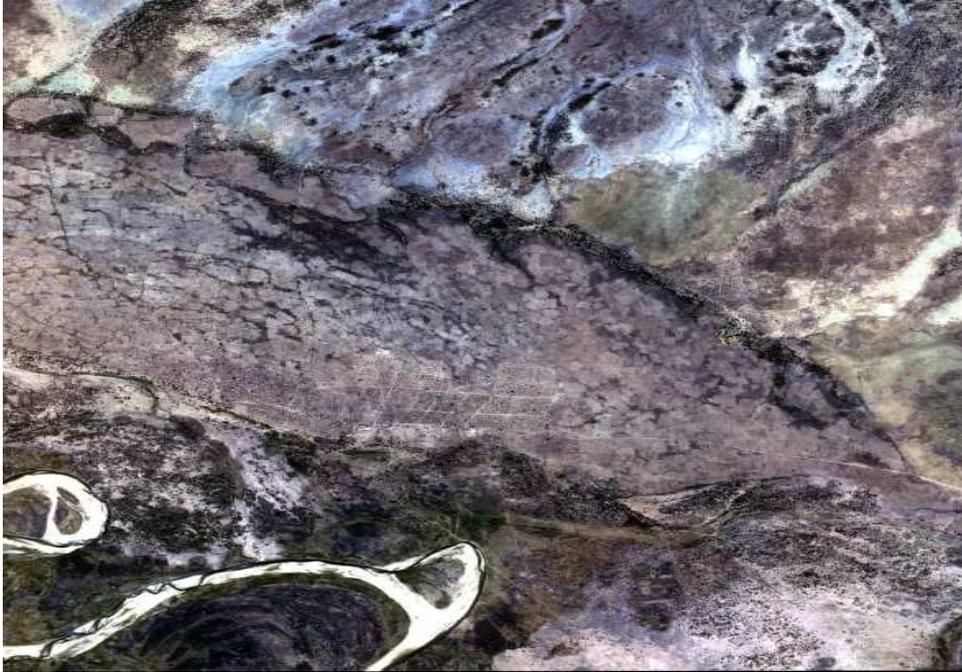
Lo studio è stato condotto su porzioni di due diversi campi profughi dell'Africa sub-sahariana, vale a dire il campo Goz Amer nel Ciad orientale e il campo Lukole A e B, nella Tanzania occidentale. Entrambi i campi presentano strutture regolari, semipermanenti, in termini di unità abitative (tende o case). Le aree di studio usate sono tre sottoinsiemi in ciascuno dei campi, con dimensioni comprese tra 4,3 e 9,0 ettari, necessarie per il confronto intra-e inter-campo (figura 5.9 per la posizione dei campi nel continente africano e le estensioni dei sottoinsiemi usati).



**Figura 5.9.** Overview del campo profughi Goz Amer in Ciad (Quickbird) e il campo profughi Lukole in Tanzania (Ikonos). Le caselle rosse indicano la posizione dei sei sottoinsiemi selezionati nello studio, nonché la localizzazione degli stessi nel continente Africano.

I dati satellitari ad altissima risoluzione (VHSR) usati per condurre lo studio, sono rappresentati da un'immagine QuickBird (12/2004, pansharpened, 4 bande, con 0,6m di risoluzione) per il campo Goz Amer,

e da un'immagine Ikonos (09/2000, pancromatica, avente 1m di risoluzione) per quanto riguarda il campo Lukole (figure 5.10 e 5.11).



*Figura 5.10. Overview del campo profughi Goz Amer in Ciad. Immagine satellitare Quickbird del 12/2004, pan-sharpened avente 60 cm di risoluzione.*

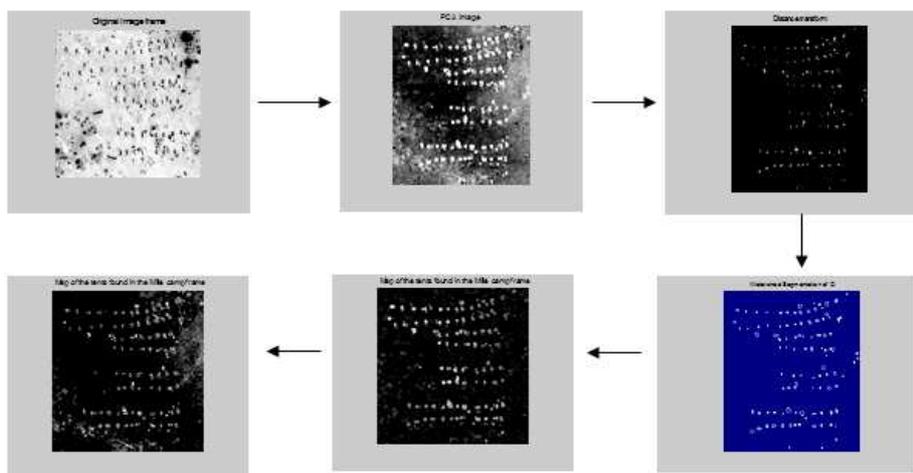


*Figura 5.11. Overview del campo profughi Lukole in Tanzania. Immagine satellitare Ikonos del 09/2000, avente 1m di risoluzione.*

Un'opportuna fase di *pre-processing* dei dati satellitari, analogamente a quanto fatto implementando altri sistemi di *pattern recognition*, consente di migliorare il contrasto delle immagini e le caratteristiche delle unità abitative cui siamo interessati.

I sistemi di *pattern recognition* strutturali (basati sulla *morfologia matematica*), creati ed implementati in questo studio per risolvere il problema dell'individuazione delle unità abitative, sono diversi. Le idee di base utilizzate per lo sviluppo degli algoritmi sono:

- l'osservazione che ciascuna zona di studio considerata ha, grazie al *preprocessing*, un "buon comportamento" nel senso che mostra tutte le caratteristiche (forme, dimensioni e contrasto rispetto allo sfondo) necessaria per ottenere buoni risultati con la MM. Infatti, nei campi le tende appaiono come oggetti con geometria e dimensioni ben definite, ovvero quadrati di circa 4x4 pixel e rettangoli di circa 10x5 pixel, e un numero digitale molto elevato o molto basso, rendendole ben separabili dallo sfondo dell'immagine [41], [44].
- la definizione di una famiglia (base) di elementi strutturanti costruiti in funzione della forma e dimensione delle tende. Il processo consiste nell'individuare, sull'immagine satellitare e per ciascun ES della famiglia, i pattern inclusi in ciascun elemento strutturante della base, sottraendo e isolando loro dall'immagine d'input. Il processo viene ripetuto per ogni elemento della base finché tutte le tende dell'immagine siano state riconosciute.



**Figura 5.12.** Catena di processing morfologico seguita per isolare e contare le tende in un frammento del campo profughi Goz Amer.

Le idee descritte precedentemente sono state tradotte in codici ed implementate interamente in ambiente MATLAB, grazie ad opportuni toolbox [41], [44]. Vale la pena ricordare che, le tecniche morfologiche sono applicabili su immagini a toni di grigio (alle quali è sempre possibile ricondursi con opportune tecniche), risultando così non

sfruttate completamente le informazioni spettrali contenute nelle immagini multispettrali.

Il campo profughi di Goz Amer in Ciad presenta delle ottime caratteristiche come scenario di riferimento per sviluppare e validare gli algoritmi basati sulla teoria di morfologia matematica. La figura 5.13 mostra una porzione di campo Goz Amer analizzata con approccio morfologico, e con il software commerciale *eCognition*.



**Figura 5.13.** Immagine originale del campo Goz Amer (A.). Oggetti verdi dell'immagine (B) rappresentano le tende correttamente rilevate utilizzando una procedura morfologica automatica. Oggetti colorati nell'immagine (C), costituiscono tende correttamente rilevate mediante il software *eCognition*.

Nella tabella 5.1, abbiamo alcuni dei risultati ottenuti analizzando uno dei frammenti del campo Goz Amer presi in considerazione, mediante le tecniche morfologiche descritte sopra. In particolare, sono forniti il numero di oggetti classificati come tende, gli oggetti correttamente classificati come tale e gli errori commessi dall'algoritmo. Tale errore (ultima colonna), calcolato confrontando il numero di tende rilevate dai diversi algoritmi con quello ottenuto da un conteggio visivo, risulta sufficientemente basso. Anche in quella parte del campo Goz Amer in cui le tende sono meno distinguibili dallo sfondo, l'applicazione di un opportuno pre-processing, permette di ottenere un errore intorno al 17%, corrispondente ad una sottostima del numero di tende.

Algoritmo adottato	Oggetti classificati come tende	Oggetti correttamente classificati come tende	Errore di Omissione e Commissione $E_o/E_c$	Numero di errori (%)
Metodo CRPSM 1	148	125 (18 $E_o$ + 23 $E_c$ )	$E_o = 18$ $E_c = 23$	18 (12,6)
Metodo CRPSM 2	139	135 (8 $E_o$ + 4 $E_c$ )	$E_o = 8$ $E_c = 4$	8 (5,60)
Metodo CRPSM 3	143	132 (11 $E_o$ + 11 $E_c$ )	$E_o = 11$ $E_c = 11$	11 (7,69)

**Tabella 5.1.** Risultati dell'analisi del errore ottenuta su un campione del campo di rifugiati Goz Amer analizzato.

I risultati e la densità di tende (da cui derivare la popolazione totale del campo) ottenute dall'analisi, sono riportati nella tabella 5.2.

Metodo	Numero di tende rilevate	Densità di tende calcolata
Morfologico	144	0,004633 tende/m <sup>2</sup>
sw eCognition	163	0,005244 tende/m <sup>2</sup>
Conteggio visuale	143	0,004601 tende/m <sup>2</sup>

**Tabella 5.2.** Sintesi dei risultati ottenuti su una delle Porzioni del campo di rifugiati Goz Amer.

Analogamente a quanto fatto nel caso del campo Goz Amer, abbiamo condotto un'analisi esaustiva relativa al campo profughi Lukole (Tanzania), attraverso le tecniche morfologiche. Tre settori del campo sono stati sottoposti all'analisi, ciascuno avente dimensioni di 300x300 m<sup>2</sup>. La densità media dei 3 settori del campo considerati è pari a 0,002426 tende/m<sup>2</sup>. L'utilizzo di questo dato per la dimensione di tutto il campo (1.119,4 ettari), fornisce un numero totale di tende pari a 27.153.

Settore del campo analizzato	Conteggio visuale di tende	Densità di tende calcolata	Conteggio automatico tramite MM	Densità di tende calcolata
Lukole set. 1	214	0.002377 ten/m <sup>2</sup>	226	0.002511 ten/m <sup>2</sup>
Lukole set. 2	238	0.002644 ten/m <sup>2</sup>	230	0.002555 ten/m <sup>2</sup>
Lukole set. 3	193	0.002144 ten/m <sup>2</sup>	199	0.002211 ten/m <sup>2</sup>

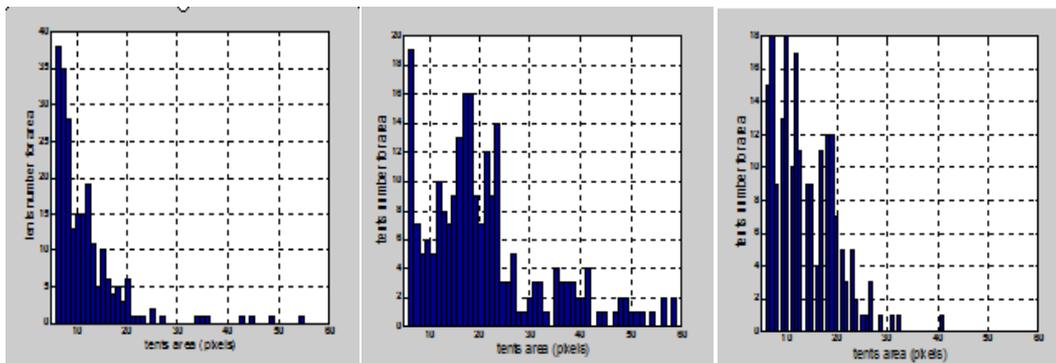
**Tabella 5.3.** Sintesi dei risultati ottenuti assumendo un'area occupata di 2 m<sup>2</sup> per ogni rifugiato.

La densità media di tende, calcolata visivamente, è pari a 0.002388 tende/m<sup>2</sup>. Estendendo questo valore di densità all'intera superficie del campo, si ottiene un numero di tende pari a 26.375. Secondo i dati forniti da ECHO (Ufficio per gli Aiuti Umanitari della Commissione Europea), il numero di rifugiati ospitati in ciascuna unità abitativa è mediamente pari a 5 o 6. Sulla base di questi valori, la popolazione reale del campo risulta compresa tra 131.875 e 158.250 unità [45].

Il JRC ha stimato (utilizzando i metodi descritti in [42]), una popolazione profughi totale compresa tra 121.935 e 146.322 unità. Dal canto nostro, la stima della popolazione di profughi presente nel campo al momento della ripresa satellitare, calcolato con l'algoritmo morfologico migliore (errore più basso) e nelle stesse ipotesi, porta ad una stima compresa tra 135.765 e 162.918 unità. L'errore medio

generato dai nostri *algoritmi morfologici* nel conteggio delle unità abitative, è pari al 2,95% rispetto all'elaborazione visiva.

Un approccio alternativo, forse più accurato, per stimare la popolazione di un campo profughi, tiene conto della dimensione delle tende e della loro distribuzione (figura 5.14). In questo caso la superficie totale coperta dalle tende viene calcolata, e il numero di rifugiati può essere stimato assegnando una determinata superficie dentro il campo a ciascuno di essi. Nel nostro caso, possiamo assumere una superficie occupata da ciascun rifugiato pari a 2 pixel (circa 2m<sup>2</sup>). Noi riteniamo che questo approccio possa fornire una maggiore precisione nella stima della popolazione del campo, dal momento che tiene conto della dimensione reale delle tende presenti nello stesso. Seguendo questo approccio si arriva ad una densità media di 0,0187 rifugiati/m<sup>2</sup>, e questo valore applicato all'intero campo (1119,4 ettari), ci fornisce un numero totale di rifugiati pari a 209.327 unità (tabella 5.4).



**Figura 5.14.** Distribuzione delle unità abitative in funzione della loro dimensione nelle tre aree campioni considerate.

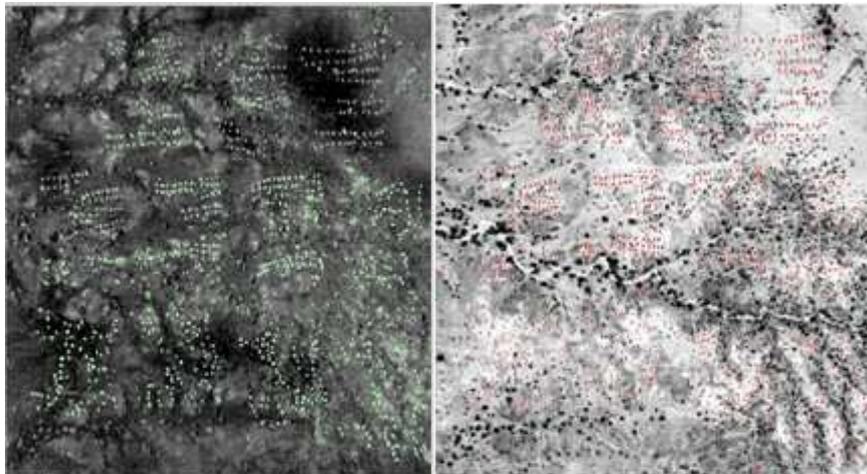
Questo numero naturalmente, può essere considerato come una stima del numero potenziale di rifugiati che possono essere ospitati nel campo, mentre il loro numero effettivo dipende dalla dinamica dei flussi migratori che coinvolgono il campo stesso.

<b>Settore del campo studiato</b>	<b>Superficie coperta dalle tende</b>	<b>Dimensione media tende</b>	<b>N. di rifugiati per ogni settore del campo analizzato</b>	<b>Densità media di rifugiati nel settore considerato</b>
Lukole set. 1	2521 pixel	11,5 pixel	1260,5	0,0140 rifug./m <sup>2</sup>
Lukole set. 2	4845 pixel	21,06 pixel	2422,5	0,02692 rifug./m <sup>2</sup>
Lukole set. 3	2737 pixel	13,75 pixel	1368,5	0,01521 rifug./m <sup>2</sup>

**Tabella 5.4.** Sintesi dei risultati ottenuti assumendo un'area occupata di 2m<sup>2</sup> per ogni rifugiato.

Riassumendo quanto descritto e dimostrato precedentemente, possiamo dire che tutti i sistemi di *pattern recognition morfologici* sviluppati e implementati, producono risultati precisi nell'individuazione delle unità abitative e consentono teoricamente, una stima abbastanza accurata della popolazione totale (e settoriale) ospitata nei campi. Anche se queste considerazioni sono vere, bisogna ricordare comunque che la popolazione effettiva può essere diversa da quanto stimata, a causa del numero di persone per tenda usato (5 o 6 in questa sede) nei calcoli, come visto precedentemente. Sostituendo questo fattore con la superficie occupata minima assegnata ad ogni rifugiato, per la superficie coperta dalle tende, è possibile arrivare con molta accuratezza a stimare il numero massimo di rifugiati che può essere ospitata nel campo.

Per concludere questa sezione della tesi, vogliamo richiamare i risultati di un'analisi effettuata congiuntamente dal CRPSM (con metodi morfologici) e dalla società Definiens (utilizzando il sw commerciale per l'elaborazione delle immagini eCognition). In particolare, le informazioni utili sono state raccolte in due settori dei campi Goz Amer e Mille, particolarmente difficili da analizzare dal punto di vista delle caratteristiche spettrali e del contrasto, anziché estrapolare le loro statistiche (tende, densità, popolazione, ecc) dall'analisi condotta in altri settori del campo come fatto precedentemente.



**Figura 5.15.** Frammento del campo Mille dopo processing con tecniche morfologiche (sinistra) ed con eCognition (destra).

Nel software *eCognition* [46], gli oggetti vengono estratti dall'immagine attraverso un certo numero di livelli gerarchici di segmentazione. Ogni livello successivo estrae oggetti di una dimensione media maggiore combinando gli oggetti di un livello inferiore. Ciò comporta la rappresentazione delle informazioni dell'immagine su scale diverse contemporaneamente. Oggetti (o pixel al primo livello) sono raggruppati

in un oggetto più grande in base alla somiglianza spettrale, il contrasto con gli oggetti vicini, e le caratteristiche di forma.

L'applicazione del metodo morfologico sulla suddetta porzione del campo Goz Amer con caratteristiche difficili (diverse forme e dimensioni delle tende, presenza di recinzioni che delimitano gruppi di unità abitative, ecc), ha generato un errore, rispetto al numero di tende visivamente stimate, dell'ordine di 9,9% mentre il SW *eCognition* [47] dà un errore del 27,5%. L'area oggetto di studio corrisponde a 479x530 pixel, pari a 152.322 m<sup>2</sup> e le tende sono oggetti con dimensione compresa tra 8 e 74 pixel.

Per quanto riguarda il campo Mille (figura 5.15), il confronto è stato fatto su una zona di 1400x1300 pixel, corrispondenti a 1.092.000 m<sup>2</sup>. L'errore commesso dal approccio morfologico è di circa 8,5%, mentre l'applicazione del *SWeCognition* da un errore di circa il 27%. Il campo Mille è caratterizzato da uno schema regolare composto da 21 blocchi. La densità media di tende per blocco, è stato calcolata su un campione di 11 blocchi mediante conteggio visivo e approccio morfologico ottenendo rispettivamente 81,54 tende/blocco e 75 tende/blocco. Estendendo il questo risultato a tutto il campo, possiamo derivare il numero totale di tende. Ciò corrisponde a 1712 tende per il metodo visivo e 1575 tende con il metodo morfologico. Questi numeri sono in accordo con quelli stimati dal DLR (Agenzia Spaziale Tedesca) di 1600 tende in totale su una superficie di 1.100.000 m<sup>2</sup> [44].

<b>Tende individuate mediane ispezione visiva</b>	<b>Tende individuate mediante un metodo morfologico</b>	<b>Tende individuate tramite <i>eCognition</i></b>
<b>campo Goz Amer</b>		
353	318	450
<b>campo Mille</b>		
897	973	1140

*Tabella 5.5. Risultati del confronto tra tecnica morfologica e il sw *eCognition* su una porzione del campo Mille.*

Nella tabella 5.5 sono riportati i risultati ottenuti nelle frazioni del campo Goz Amer e Mille considerate.

I risultati ottenuti per un settore del campo profughi Mille, con tecniche morfologiche e *eCognition*, sono molto importanti poiché il settore del campo selezionato (figura 5.4), rappresenta un sito ottimale per testare i nostri algoritmi vista la sua elevata diversità di caratteristiche (notevole variabilità in termini di contrasto, dimensioni, forme delle tende, ecc).

Vale la pena sottolineare che, anche se la metodologia e analisi qui presentata ha riguardato solo i campi di Lukole, Goz Amer e Mille,

quanto esposto e dimostrato vale in generale, poiché l'analisi è stata estesa (vista la buona disponibilità di dati satellitari) a diversi altri campi. Le figure 14 e 15 mostrano altri due campi studiati durante il ciclo di dottorato.



*Figura 5.14. Overview del campo profughi Bindizi (poligono bordò) in Ciad. Immagine Quickbird multi-spettrale avente 2,4m di risoluzione.*



*Figura 5.15. Overview del campo profughi Mukjar (poligono bordò) in Ciad. Immagine Ikonos multi-spettrale avente 4m di risoluzione.*

## 5.9 Individuazione Automatiche di Strade su Immagini Ottiche e RADAR.

Il coinvolgimento del CRPSM in diversi progetti finanziati dalla Comunità Europea (GMOSS, G-MOSAIC, ecc), dedicati al problema della sicurezza delle popolazioni e delle infrastrutture (centrali nucleari, centri abitati, strade, ecc), ci ha dato la possibilità di sviluppare e implementare un'insieme di algoritmi capaci d'individuare e monitorare anche oggetti costituiti da strade .

Gli stessi principi usati nel caso precedente del monitoraggio dei campi profughi, possono essere nuovamente applicati al problema dell'individuazione automatica delle infrastrutture stradali. La capacità di individuare le strade di un'area/regione è un compito particolarmente importante perché può consentire, come già dimostrato, il riconoscimento delle complesse strade presenti in molti paesi africani consentendo così la creazione di un database aggiornato delle stesse (spesso inesistente in questo area del mondo), o semplicemente di effettuare un controllo veloce del loro stato, in seguito al verificarsi di eventi disastrosi che possono averne compromesso la stabilità e lo stato. L'esperienza maturata nell'individuazione di strade sulle immagini ad alta e media risoluzione, ha portato allo sviluppo di uno specifico pacchetto software, dotato di opportuna interfaccia GUI, adatto a risolvere questo tipo di problemi.

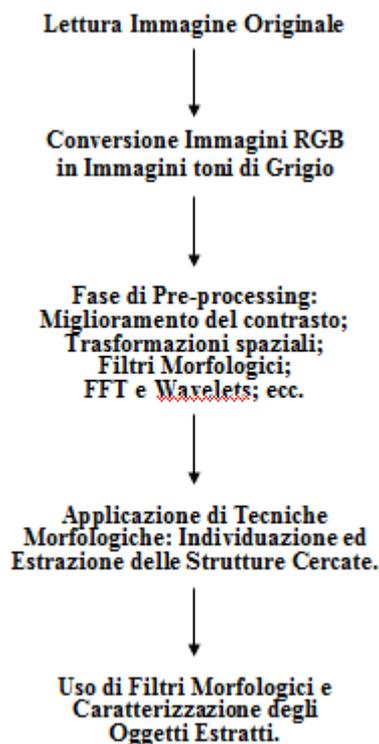
Tra i casi studiati, presentiamo i risultati relativi al monitoraggio della rete stradale in una regione montuosa del Kashmir. Tale regione è stata colpita da un forte terremoto l'8 ottobre del 2005, rendendo necessaria una rapida valutazione dei danni alle infrastrutture per il coordinamento degli interventi e dei soccorsi da prestare alle popolazioni locali. Per fare questo, ci siamo serviti di una coppia di immagini pan-cromatiche SPOT 5 dell'area prima e dopo l'evento (figura 5.16).



**Figura 5.16 .** Immagini ottiche SPOT 5(2,5m di risoluzione spaziale) prima (a sinistra del 6/10/2005) e dopo (a destra del 21/10/2005) il forte terremoto che ha colpito la regione del Kashmir.

L'idea di base ancora è quella di affrontare, vista la grande versatilità e robustezza mostrata dalla *morfologia matematica*, il problema attraverso la creazione di opportuni elementi strutturanti (in termini di lunghezza e larghezza), capaci, dopo adeguato pre-processing, di individuare ed estrarre qualsiasi tipo di strada.

La catena di elaborazione applicata in questo caso e valida in generale comprende due fasi (quella di *pre-processing* e quella di *processing* vera e propria) che possono essere riassunte secondo lo schema della figura 5.17.



**Figura 5.17.** Schema dell' algoritmo morfologico usato per l'estrazione delle strade.

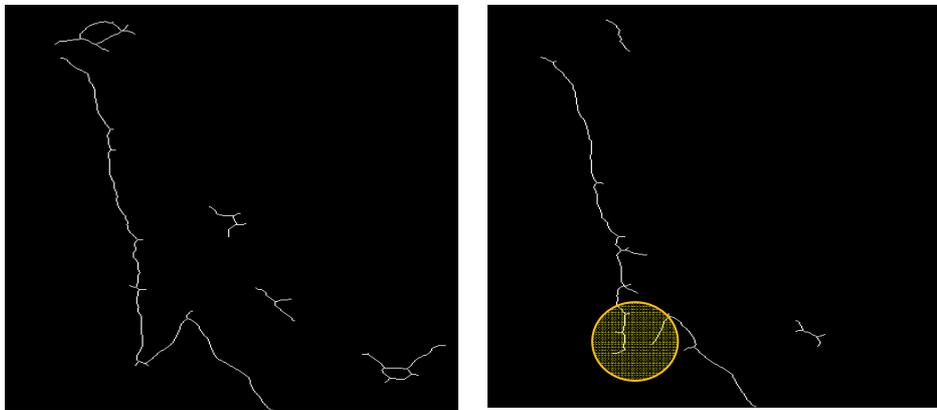
La fase di *pre-processing*, necessaria per rimuovere il rumore presente nell'immagine e per migliorare il contrasto, consente successivamente agli operatori morfologici di comportarsi meglio. Questa fase comprende, in generale, i seguenti passaggi:

- *miglioramento del contrasto*: è ottenuto attraverso una trasformata che esegue una convoluzione tra l'immagine originale e opportuni filtri;
- *equalizzazione dell'istogramma*: questa trasformazione mira a migliorare ulteriormente il contrasto dell'immagine d'input attraverso un'equalizzazione effettuata su aree dell'immagine più piccole rispetto alla dimensione originale;

- *correzione gamma*: questa trasformazione è in grado di generare una nuova immagine in toni di grigio dall'immagine originale, migliorando i toni grigio che hanno le strade.

La fase di processing invece, prevede i seguenti passaggi:

- *definizione di un elemento strutturante* lineare e stretto (ES), poiché queste sono le caratteristiche principali degli oggetti ricercati (strade);
- *analisi delle immagini* mediante l'elemento strutturante definito precedentemente, avente lunghezza appropriata (in accordo ai tratti lineari della strade) e direzione variabile tra  $0^\circ$  e  $180^\circ$ , con passo adatto (in accordo allo spessore stradale). L'immagine completa viene studiata, e le strade con una lunghezza superiore all'ES usato, qualunque sia la loro direzione, vengono individuate.

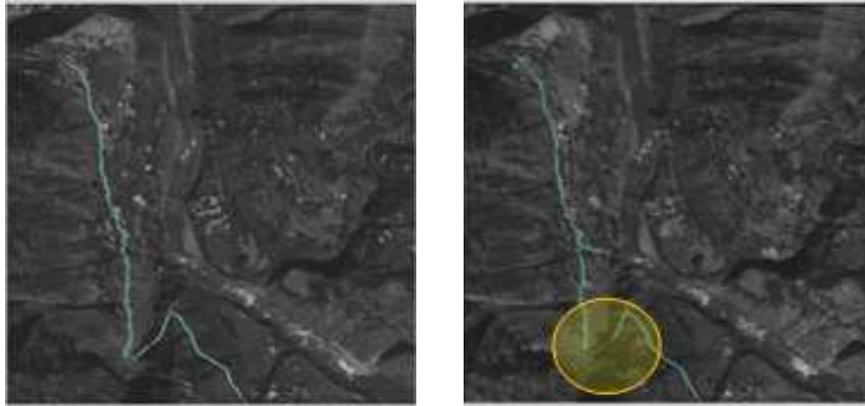


*Figura 5.18. Fase di processing morfologico intermedio dove possono apprezzarsi le strade tortuose di montagna estratte più altri oggetti (rumore presente in entrambi) che vengono rimossi mediante opportuni filtri morfologici applicati*

La figura 5.18 mostra i risultati intermedi, prima dell'applicazione dei filtri morfologici per la rimozione del rumore, ottenuti applicando il *pattern recognition morfologico* descritto precedentemente sulle immagini SPOT 5. Come si può vedere, nelle stesse (prima e dopo) si evidenzia la struttura binarizzata delle strade cercate più altri oggetti (del rumore) frutto della non idealità della catena di processing. Questi oggetti vengono comunque rimossi in uno step successivo, grazie all'utilizzo di *filtri morfologici* adatti a rimuovere queste impurità. La figura 5.19 invece, mostra il passo finale della catena di processing, evidenziando le strade cercate (in sovrapposizione all'immagine originale) e l'area della sede stradale che ha riportato danni (cerchio giallo).

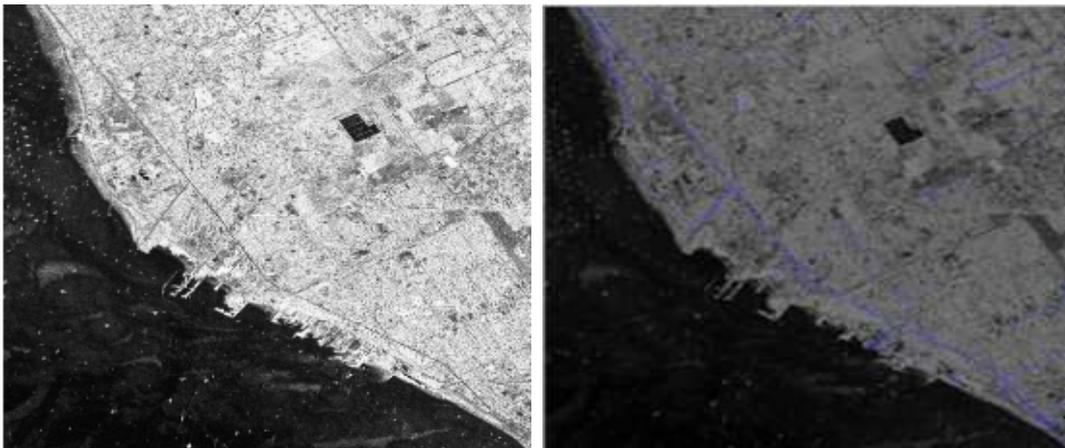
Per concludere questo argomento e l'intero capitolo, vogliamo ricordare che questo *pattern recognition morfologico* può essere applicato non

solo al caso delle immagini satellitari ottiche, ma anche al caso delle immagini SAR.



**Figura 5.19.** Risultato del processing morfologico automatico nella estrazione delle strade del Kashmir. Immagine pre (sinistra) e post evento (destra) con la sovrapposizione di quanto della strada estratta e dell'area danneggiata (circolo giallo dell'immagine di destra).

Infatti, nella figura 5.20 vengono mostrati i risultati ottenuti applicando la catena di elaborazione morfologica sviluppata per immagini ottiche (salvo alcune varianti in fase di pre-processing), ad un'immagine SAR della città di Cabimas, situata sul lago Maracaibo in Venezuela. L'immagine è stata acquisita dal sistema COSMO-SkyMed in modalità Stripmap (cortesia ASI), con risoluzione spaziale di 5m. Anche se la qualità dell'immagine originale non è la migliore possibile, data dalla combinazione della risoluzione spaziale e dalle caratteristiche dello scenario, il sistema automatico è ancora in grado di rilevare la maggior parte delle strade visibili.



**Figura 5.20.** Immagine SAR Cosmo-SkyMed del 4/4/2008 sulla città di Cabimas (Lago di Maracaibo, Venezuela). Prodotto Stripmap avente 5m di risoluzione spaziale. Nonostante una risoluzione media l'algoritmo di MM è ancora capace di individuare la maggior parte delle strade.

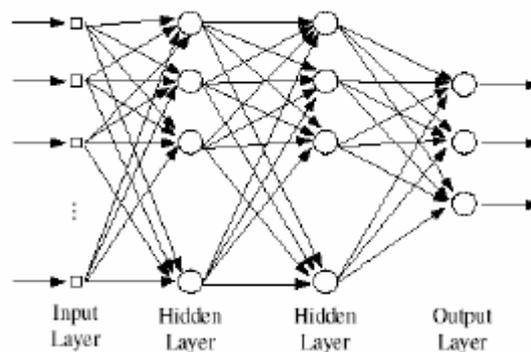
## Capitolo 6

# Le Reti Neurali

La disponibilità di un insieme di algoritmi non lineari per *l'estrazione delle features* (mediante l'utilizzo del livello nascosto o hidden layer) e per la *classificazione dei pattern*, rende le *reti neurali* particolarmente adatte al problema del *pattern recognition*. Infatti, la loro capacità di addestrarsi (mediante procedure iterative), e gestire relazioni non lineari fra l'input e l'output, adattandosi ai pattern di esempio, le ha rese uno strumento indispensabile nel campo dell'individuazione automatica degli oggetti, giustificando il crescente utilizzo che ne sta derivando.

Le *Reti Neurali* possono essere considerate come un sistema computazionale fortemente parallelo, costituito da un grande numero di processori estremamente semplici ma intensamente interconnessi fra loro.

I modelli basati su *reti neurali* utilizzano i principi di organizzazione dei dati, come: *addestramento*, *generalizzazione*, *tolleranza ai guasti* e *distribuzione dei dati*, in una **rete di grafi pesati**. I *nodi* sono rappresentati da neuroni artificiali e gli *archi diretti* (pesati) rappresentano le connessioni tra l'output di un livello di neuroni e l'input del successivo livello, come mostrato nella figura 6.1.

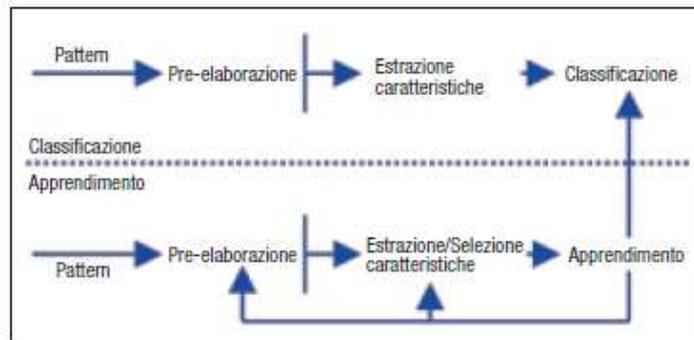


**Figura 6.1.** Architettura generale di una Rete Neurale (tipo MLP) in cui vengono evidenziati gli strati di input, intermedi (nascosti) e di output fortemente interconnessi mediante i nodi e gli archi della stessa.

L'approccio al problema del pattern recognition basato sulle *reti neurali*, diversamente da quello che sembra, è implicitamente equivalente o

molto simile ai metodi di pattern recognition basati sui *modelli statistici* come si evince dalla figura 6.2. Ma indipendentemente dalle similitudini, le reti neurali offrono molti vantaggi ai progettisti di sistemi di *pattern recognition*, rispetto ad altri approcci quali:

- un approccio unificato alla selezione ed estrazione delle features;
- una strategia generale e flessibile per trovare soluzioni ai problemi di classificazione in cui le regioni di decisione siano non lineari;



**Figura 6.2.** Schema generale di un sistema di pattern recognition basato su classificazione statistica (sopra). Le fasi rappresentate in figura, descrivono anche in generale i sistemi basati su reti neurali (fig. 6.1).

Nei seguenti paragrafi verrà presentata un'introduzione alle *reti neurali* e ai principali tipi usati nel campo del *pattern recognition*. Infine, in conclusione del capitolo saranno presentati i riconoscitori basati sulle reti neurali che sono stati sviluppati ed implementati per risolvere i problemi dell'individuazione automatica delle macchie di petrolio in mare.

## 6.1 Il Neurone (modello di McCulloch-Pitts)

Un *neurone* è l'unità di calcolo fondamentale della rete neurale ed è formato da 3 elementi di base nel modello neurale:

1. un insieme di *sinapsi* o *connessioni* ciascuna delle quali è caratterizzata da un peso (efficacia sinaptica); a differenza del modello umano, il modello artificiale può avere pesi sia negativi che positivi;
2. un *sommatore* che somma i segnali in input pesati dalle rispettive sinapsi, producendo in output una combinazione lineare degli input;
3. una *funzione di attivazione* per limitare l'ampiezza dell'output di un neurone. Tipicamente per comodità l'ampiezza degli output appartengono all'intervallo  $[0,1]$  oppure  $[-1,1]$ .

Il modello neuronale include anche un *valore soglia* che ha l'effetto, a seconda della sua positività o negatività, di aumentare o diminuire l'input netto alla funzione di attivazione.

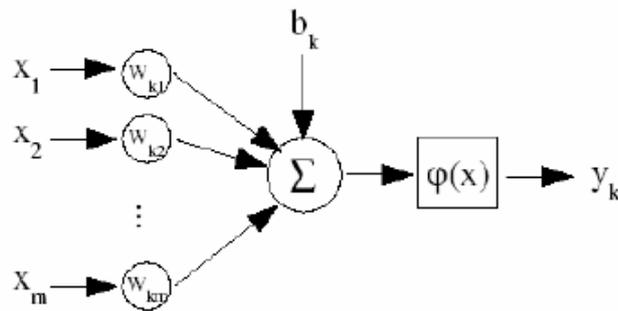


Figura 6.3. Modello non lineare di un neurone.

In termini matematici, possiamo descrivere un neurone k con le seguenti equazioni:

$$u_k = \sum_{j=1}^m w_{kj} \cdot x_j$$

$$y_k = \varphi(u_k + b_k)$$
(6.1)

dove:

- $w_{ki}$  sono i pesi sinaptici del neurone k;
- $x_j$  sono le grandezze in ingresso alla rete
- $u_k$  è la combinazione lineare degli input nel neurone k;
- $b_k$  è il valore soglia del neurone k;
- $\varphi(x)$  è la funzione di attivazione;
- $y_k$  è l'output generato dal neurone k.

Possiamo riformulare le 2 equazioni precedenti inglobando il valore soglia in  $u_k$ , ottenendo così un modello equivalente al precedente:

$$v_k = \sum_{j=0}^m w_{kj} \cdot x_j$$

$$y_k = \varphi(v_k)$$
(6.2)

dove l'input  $x_0=1$  è  $w_{k0}=b_k$ . Definiamo  $v_k$  *potenziale di attivazione*.

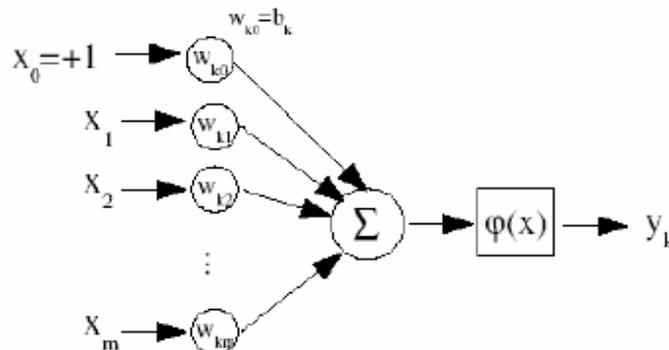


Figura 6.4. Altro modello non lineare di un neurone.

## 6.2 Funzioni di attivazione

Identifichiamo 3 tipi base di funzione di attivazione:

### 6.2.1 Funzione threshold (o funzione Heaviside)

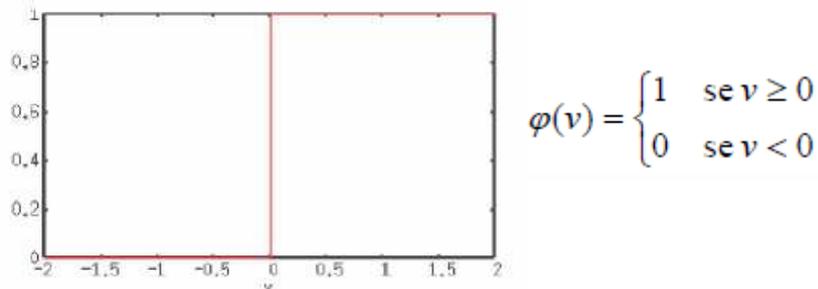


Figura 6.5. Funzione threshold (soglia).

Questa funzione è usata nel modello di McCulloch-Pitts.

### 6.2.2 Funzione piecewise-linear

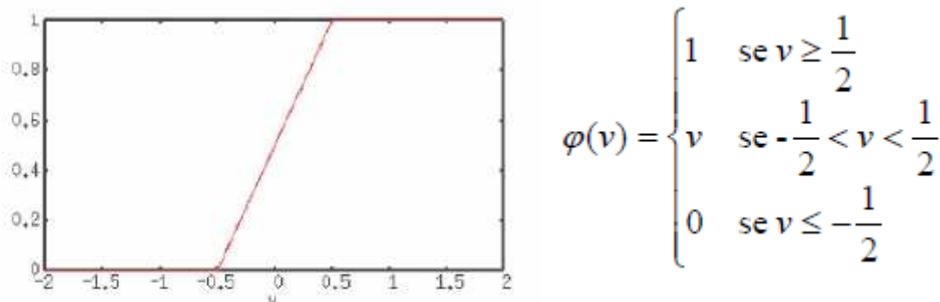


Figura 6.6. Funzione piecewise-linear

### 6.2.3 Funzione sigmoide

E' la funzione più usata nella costruzione di reti neurali artificiali. E' una funzione strettamente crescente, che esibisce un bilanciamento tra un comportamento lineare e non lineare.

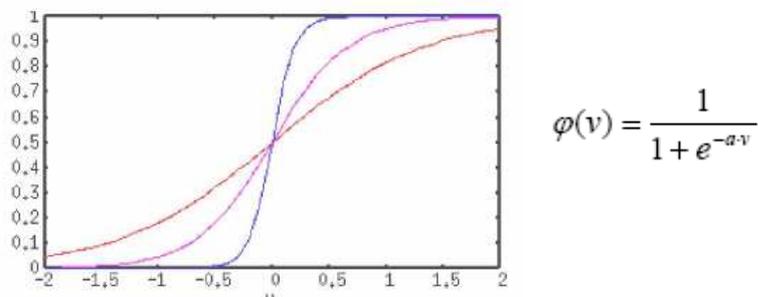


Figura 6.7. Funzione logistica. In rosso  $a=1.5$ ; in viola  $a=3$ ; in blu  $a=10$ .

Questa funzione approssima la threshold function se  $a \rightarrow +\infty$ .

A volte è desiderabile avere una funzione di attivazione nell'intervallo  $[-1,1]$ . In questo caso possiamo utilizzare la tangente iperbolica.

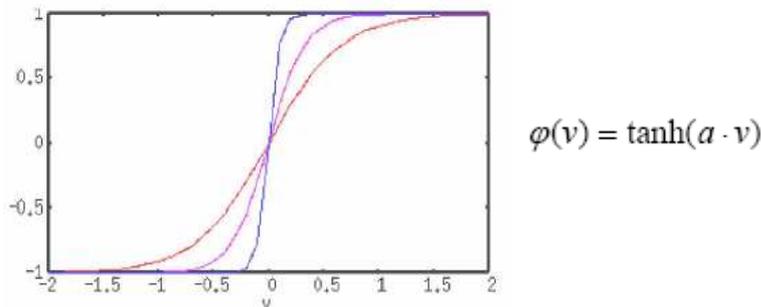


Figura 6.8. Tangente iperbolica. In rosso  $a=1.5$ ; in viola  $a=3$ ; in blu  $a=10$ .

### 6.3 Architettura delle Reti Neurali

Il modo con cui è strutturata la rete dipende dall'algoritmo di apprendimento che si ha intenzione di usare. In generale identifichiamo 3 classi di reti.

#### 6.3.1 Rete feedforward ad uno strato

In questa forma semplice di rete a strati, abbiamo i nodi di input (**input layer**) e uno strato di neuroni (**output layer**). Il segnale nella rete si propaga in avanti in modo aciclico, partendo dal layer di input e terminando in quello di output. Non ci sono connessioni che tornano indietro e nemmeno connessioni trasversali nel layer di output.

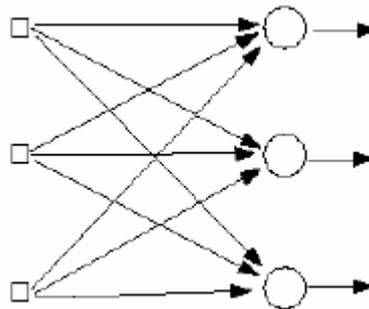


Figura 6.9. Rete forward ad uno strato.

#### 6.3.2 Rete feedforward a più strati

Questa classe di reti feedforward si distingue dalla precedente, dal fatto che tra lo strato di input e quello di output abbiamo uno o più strati di neuroni nascosti (**hidden layers**). Ogni strato ha connessioni entranti dal precedente strato e uscenti in quello successivo, quindi la propagazione del segnale avviene in avanti senza cicli e senza connessioni trasversali.

Questo tipo di architettura fornisce alla rete maggiore complessità e una prospettiva globale in quanto aumentano le interazioni tra i neuroni.

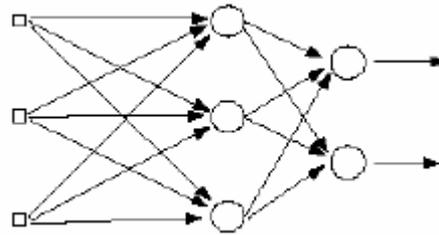


Figura 6.10. Rete feedforward con 2 strati.

### 6.3.3 Rete ricorrente o feedback

Una rete ricorrente si distingue dalle precedenti nel fatto che è ciclica. La presenza di cicli ha un impatto profondo sulle capacità di apprendimento della rete e sulle sue performance, poiché rendono il sistema dinamico.

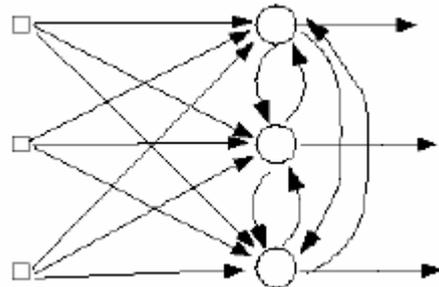


Figura 6.11. Rete ricorrente o feedback.

## 6.4 Processi di apprendimento

L'apprendimento è un processo col quale parametri liberi di una rete neurale sono adattati, attraverso un processo di stimolazione, all'ambiente in cui essa è inserita. Il tipo di apprendimento è determinato dal modo in cui questi adattamenti avvengono.

Un **algoritmo di apprendimento** è un insieme di regole ben definite che risolvono un problema di apprendimento. Questi algoritmi possono essere di due tipi:

1. **con supervisione:** c'è un insegnante che conosce l'ambiente che fornisce mappature corrette di input/output. La rete dovrà aggiustare i propri parametri liberi in modo da emulare l'insegnante in modo statisticamente ottimale.
2. **senza supervisione:** la rete apprende autonomamente.

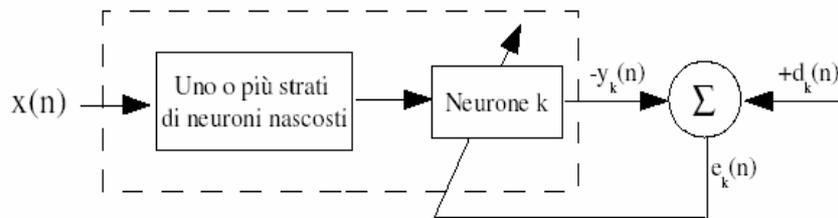
Vediamo alcune tipologie di apprendimento.

### 6.4.1 Apprendimento con correzione di errore

Ogni neurone  $k$  riceve in ingresso un segnale di stimolo  $x(n)$  e genera una risposta  $y_k(n)$ , con  $n$  tempo discreto. Indichiamo inoltre con  $d_k(n)$  la risposta desiderata. Di conseguenza si genera un segnale di errore  $e_k(n)$ .

$$e_k(n) = d_k(n) - y_k(n) \quad (6.3)$$

Il segnale di errore  $e_k(n)$ , attua un meccanismo di controllo con l'obiettivo di applicare una sequenza di aggiustamenti ai pesi sinaptici del neurone  $k$ , al fine di avvicinare la risposta ottenuta a quella desiderata.



**Figura 6.12.** Esempio di apprendimento con correzione di errore con rete multistrato feed-forward.

Questo processo avviene un passo alla volta minimizzando una **funzione costo**:

$$E(n) = \frac{1}{2} \cdot e_k^2(n) \quad (6.4)$$

Per fare ciò si ricorre al **metodo del gradiente** o metodo di Widrow-Hoff. Siano  $w_{kj}(n)$  i pesi sinaptici del neurone  $k$  eccitati dall'elemento  $x_j(n)$  del segnale di stimolo, allora l'aggiustamento applicato a  $w_{kj}(n)$  è:

$$\Delta w_{kj} = \eta \cdot e_k(n) \cdot x_j(n) \quad (6.5)$$

dove  $\eta$  è una costante positiva detta **tasso di apprendimento**. I nuovi pesi saranno allora:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (6.6)$$

L'apprendimento con correzione di errore è un esempio di sistema ricorrente la cui stabilità dipende dai parametri che costituiscono il ciclo e in particolare da  $\eta$ . La scelta di questo parametro è importante perché influenza la convergenza e la stabilità del processo di apprendimento.

### 6.4.2 Apprendimento basato sulla memoria

Nell'apprendimento basato sulla memoria, tutti (o molti) delle passate esperienze vengono archiviate in una larga memoria di coppie input-output correttamente classificate  $\{(\mathbf{x}_i, d_i)\}_{i=1}^n$ , dove  $\mathbf{x}_i$  è il vettore di input e  $d_i$  è la risposta desiderata, che senza perdita di generalità abbiamo ridotto ad uno scalare.

Quando verrà richiesta la classificazione di un esempio mai incontrato prima  $\mathbf{x}_{\text{test}}$  il sistema risponde trovando ed analizzando gli esempi memorizzati in un intorno di  $\mathbf{x}_{\text{test}}$ .



Tutti i metodi di apprendimento basati su memoria comportano 2 ingredienti base:

- criterio usato per definire l'intorno di un vettore test  $\mathbf{x}_{test}$  ;
- metodo di apprendimento applicato sugli esempi nell'intorno di  $\mathbf{x}_{test}$  .

Un esempio di questo metodo è il **metodo nearest-neighbor** in cui l'esempio più vicino all'esempio test  $\mathbf{x}'_N$  è quello che ha distanza euclidea minima.

$$\mathbf{x}'_N \in \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \quad (6.7)$$

$$\min_i [d(\mathbf{x}_i, \mathbf{x}_{test})] = d(\mathbf{x}'_N, \mathbf{x}_{test})$$

dove  $d$  è la distanza euclidea. La classe che viene assegnata ad  $\mathbf{x}_{test}$  è la stessa di  $\mathbf{x}'_N$ .

Una variante di questo metodo è il **k-nearest neighbor** in cui:

- l'intorno dell'esempio test non è più uno solo, ma l'insieme dei  $k$  esempi memorizzati più vicini;
- la classe assegnata è quella con la frequenza maggiore nell'intorno dell'esempio test.

## 6.5 Apprendimento di *Hebbian*

Postulato di Hebb sull'apprendimento: Quando un assone di un neurone A è abbastanza vicino da eccitare un neurone B e questo, in modo ripetitivo e persistente, gli invia un potenziale di azione, inizia un processo di crescita in uno o entrambi i neuroni tali da incrementare l'efficienza di A.

Da ciò possiamo ricavare due regole:

1. se 2 neuroni connessi da una sinapsi si attivano simultaneamente, allora il peso della sinapsi viene progressivamente incrementato;
2. se 2 neuroni connessi si attivano in modo asincrono, allora il peso della sinapsi viene progressivamente diminuito o eliminato.

Una sinapsi di questo tipo è detta **sinapsi di Hebbian**. Se la correlazione dei segnali porta ad un incremento dell'efficacia sinaptica chiameremo questa modifica **hebbiana**, se porta ad una riduzione la chiameremo **anti-hebbiana**.

## 6.6 Apprendimento *competitivo*

Con l'apprendimento competitivo i neuroni di uscita di una rete neurale competono tra di loro per divenire attivi. Solo un neurone può essere attivo in un certo momento  $n$ .

Ci sono 3 elementi base per un metodo di apprendimento competitivo:

1. un insieme di neuroni uguali a meno di pesi sinaptici generati a caso che rispondono in maniera differente ad un dato insieme di input;
2. un limite alla “forza” di ciascun neurone;
3. un meccanismo che permette ai neuroni di competere per il diritto a rispondere ad un dato sottoinsieme di input cosicché un solo neurone o uno solo per gruppo è attivo in un certo momento. Il neurone che vince è chiamato **winner-takes-all**.

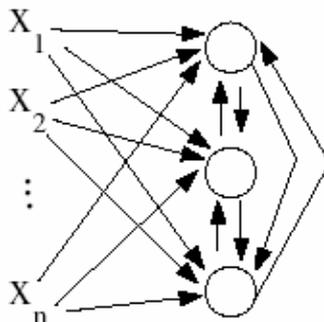
In questo modo i neuroni tendono a specializzarsi su un insieme di input simili divenendo riconoscitori di caratteristiche per differenti classi di input.

Nella forma più semplice la rete neurale ha un solo strato di neuroni di uscita, completamente connessi ai nodi di input (connessioni forward eccitatorie). La rete può includere connessioni tra i neuroni che portano ad inibizioni laterali (connessioni feedback inibitorie).

Il neurone  $k$  che vince la competizione è quello con input netto  $v_k$  più alto per un dato input  $\mathbf{x}$ . Il segnale di output  $y_k$  del neurone vittorioso è settato a 1, mentre quello degli altri viene settato a 0.

$$y_k = \begin{cases} 1 & \text{se } v_k > v_j \quad \forall j, j \neq k \\ 0 & \text{altrimenti} \end{cases} \quad (6.8)$$

dove  $v_k$  è la combinazione lineare di tutti gli input forward e feedback.



**Figura 6.13.** Struttura di una semplice rete di apprendimento competitivo.

Siano  $w_{kj} \geq 0$  i pesi sinaptici tra input  $j$  e neurone  $k$ . Supponiamo che ogni neurone abbia un ammontare fisso di peso sinaptico distribuito tra i nodi di input:

$$\sum_j w_{kj} = 1 \quad \forall k$$



Il neurone che vince la competizione apprende spostando i pesi sinaptici dagli input inattivi agli input attivi. Per fare ciò si utilizza la regola standard di apprendimento competitivo:

$$\Delta w_{kj} = \begin{cases} \eta \cdot (x_j - w_{kj}) & \text{se il neurone } k \text{ vince} \\ 0 & \text{altrimenti} \end{cases} \quad (6.9)$$

## 6.7 Attività di apprendimento

### 6.7.1 Associazione di pattern

Una **memoria associativa** è una memoria ispirata al cervello che impara per associazioni.

Le associazioni possono essere di due forme:

1. **auto-associazioni**: una rete neurale memorizza un insieme di pattern (vettori) presentandoli ripetutamente alla rete. Successivamente viene presentato alla rete un pattern (tra quelli memorizzati nella rete) parziale o distorto e l'obiettivo è che la rete richiami la versione del pattern completa/originali che è stata precedentemente memorizzata. L'apprendimento in questo caso è generalmente non supervisionato;
2. **etero-associazioni**: ogni pattern è associato ad un altro pattern. Sia  $\mathbf{x}_k$  il pattern chiave a cui è associato  $\mathbf{y}_k$  ovvero il pattern memorizzato. Il pattern chiave  $\mathbf{x}_k$  opera come uno stimolo per richiamare il pattern  $\mathbf{y}_k$ . Nelle autoassociazioni avevamo che  $\mathbf{x}_k = \mathbf{y}_k$  mentre in questo caso abbiamo che  $\mathbf{x}_k \neq \mathbf{y}_k$ .

La memoria associativa si articola in 2 fasi:

1. **fase di memorizzazione**: in cui la rete viene addestrata per fare in modo che i pattern vengano associati;
2. **fase di richiamo**: in cui si richiama dalla rete un pattern memorizzato a seguito della presentazione di una versione parziale o distorta di un pattern chiave.

### 6.7.2 Riconoscimento di pattern

Il **riconoscimento di pattern** è il processo in cui un pattern/segnale viene assegnato ad una classe (categoria). Una rete neurale riconosce pattern a seguito di una sessione di addestramento, nella quale alla rete vengono presentati ripetutamente un insieme di pattern di addestramento con specificato per ognuno la categoria a cui appartengono. Quando verrà presentato un pattern mai visto prima ma appartenete ad una stesso categoria di pattern che ha appreso, la rete sarà in grado di classificarla grazie alle informazioni estratte dai dati di addestramento.

Ogni pattern rappresenta nello **spazio decisionale** multidimensionale un punto. Questo spazio è suddiviso in regioni, ognuna delle quali associata ad una classe. I confini di queste regioni sono determinate dalla rete attraverso il processo di addestramento.

Il riconoscimento di pattern con reti neurali può assumere 2 forme:

1. il sistema viene splittato in 2 parti: una rete non supervisionata per l'estrazione delle caratteristiche (**features**) e una rete supervisionata per la classificazione. Un pattern  $\mathbf{x}$  è rappresentato da un punto dimensionale dello **spazio dei dati**. L'estrazione di caratteristiche è una trasformazione che mappa  $\mathbf{x}$  in un punto  $\mathbf{y}$  dello spazio  $q$ -dimensionale intermedio detto **spazio delle caratteristiche** con  $q < m$ . La classificazione è una trasformazione che mappa dal punto  $\mathbf{y}$  in una classe dello spazio  $r$ -dimensionale detto **spazio decisionale** con  $r$  il numero totale di classi;
2. il sistema consiste di una rete feedforward a più strati singola addestrata con un algoritmo di apprendimento supervisionato. L'attività di estrazione delle caratteristiche è un processo interno alla rete svolta dagli strati nascosti.

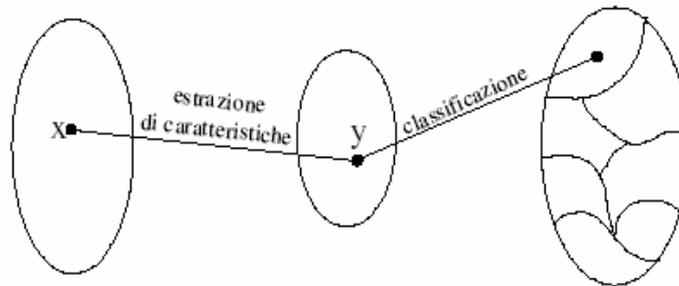


Figura 6.14. Approccio classico alla classificazione di pattern.

### 6.7.3 Approssimazione di funzioni

Si consideri una mappatura input-output non lineare descritta da:

$$\mathbf{d} = f(\mathbf{x})$$

dove  $\mathbf{x}$  è l'input e  $\mathbf{d}$  è l'output e sia la funzione  $f$  la nostra incognita. Ci vengono però forniti un insieme di esempi:

$$\{(\mathbf{x}_i, \mathbf{d}_i)\}_{i=1}^n$$

Vogliamo creare una rete neurale che approssimi la funzione  $f$  in modo che la funzione  $F$ , che descrive la mappa input-output realizzata dalla rete stessa, sia vicina ad  $f$  in senso euclideo per tutti gli input:

$$\|F(\mathbf{x}) - f(\mathbf{x})\| < \varepsilon, \quad \forall \mathbf{x} \quad (6.10)$$



dove  $\varepsilon$  è un numero positivo piccolo. Maggiori sono le dimensioni e i parametri liberi della rete minore sarà  $\varepsilon$ .

Il problema dell'approssimazione di funzioni è un candidato perfetto per l'apprendimento supervisionato con  $\mathbf{x}_i$  il vettore di input e  $\mathbf{d}_i$  la risposta desiderata.

## 6.8 Single Layer Perceptron (SLP)

Il *perceptrone* è la forma più semplice di rete neurale usata per classificare pattern *linearmente separabili*, ovvero pattern che stanno ai lati opposti di un iperpiano. Esso consiste di un singolo neurone con pesi sinaptici e soglia modificabili.

Di seguito viene proposto l'algoritmo di apprendimento conosciuto come algoritmo di convergenza del perceptrone. Rosenblatt dimostrò che se i patterns scelti per addestrare la rete appartengono a 2 classi linearmente separate, allora l'algoritmo di apprendimento converge e lo spazio decisionale viene diviso in 2 da un iperpiano. Questo teorema è chiamato **teorema di convergenza del perceptrone**.

### 6.8.1 Teorema di convergenza del perceptrone

Affinché il perceptrone funzioni correttamente è necessario che lo spazio decisionale sia formato da 2 classi  $C_1$  e  $C_2$ , separate linearmente da un iperpiano.

Siano  $X_1$  e  $X_2$  gli insiemi formati dagli input di addestramento che appartengono rispettivamente alle classi  $C_1$  e  $C_2$ , e la cui unione forma l'intero insieme di addestramento  $X$ . Assumiamo senza perdita di generalità che la funzione di attivazione  $\varphi(x)=\text{sgn}$  (funzione segno), in modo che:

$$\begin{aligned} y(n) &> 0 && \text{se } \mathbf{x}(n) \in X_1 \\ y(n) &\leq 0 && \text{se } \mathbf{x}(n) \in X_2 \end{aligned} \quad (6.11)$$

ma dal momento che  $y(n)$  e  $v(n)$  hanno lo stesso segno possiamo mettere  $v(n)$  al posto di  $y(n)$ . Quindi posto  $\mathbf{w}$  un vettore di pesi che classifica correttamente gli  $\mathbf{x} \in X$ , sarà corretta la seguente relazione:

$$\begin{aligned} v(n) &= \mathbf{w}^T \cdot \mathbf{x} > 0 && \forall \mathbf{x} \in X_1 \\ v(n) &= \mathbf{w}^T \cdot \mathbf{x} \leq 0 && \forall \mathbf{x} \in X_2 \end{aligned} \quad (6.12)$$

L'algoritmo di apprendimento può essere formulato nel seguente modo:

1. Se l' $n$ -esimo elemento di  $X$ ,  $\mathbf{x}(n)$ , è classificato correttamente dal vettore dei pesi  $\mathbf{w}(n)$ , allora non vengono fatte correzioni al vettore  $\mathbf{w}(n)$  quindi  $\mathbf{w}(n+1) = \mathbf{w}(n)$ ;

2. Altrimenti il vettore  $\mathbf{w}(n)$  viene aggiornato nel seguente modo:

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) - \eta(n) \cdot \mathbf{x}(n) \quad \forall \mathbf{x} \in X_2 \\ \mathbf{w}(n+1) &= \mathbf{w}(n) + \eta(n) \cdot \mathbf{x}(n) \quad \forall \mathbf{x} \in X_1\end{aligned}\quad (6.13)$$

dove il fattore di apprendimento  $\eta(n)$  controlla le modifiche applicate ai pesi all'n-esima iterazione.

Se  $\eta(n)=\eta>0$  con  $\eta$  una costante indipendente dal numero di iterazioni  $n$ , allora abbiamo il metodo **fixed increment adaptation** per il perceptrone. Nel seguito verrà dimostrata la convergenza dell'apprendimento con metodo fixed increment adaptation rule con  $\eta=1$ .

Sia  $\mathbf{w}(0)=\mathbf{0}$ . Supponiamo  $\mathbf{w}^T(n) \cdot \mathbf{x}(n) < 0$  per  $n=1, 2, \dots$  e  $\mathbf{x}(n) \in X_1$ , ovvero il perceptrone classifica in modo errato i vettori di input  $\mathbf{x}(1), \mathbf{x}(2), \dots$

La regola di aggiornamento dei pesi per questo caso è la seguente:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{x}(n) \quad \forall \mathbf{x} \in X_1 \quad (6.14)$$

Dal momento che  $\mathbf{w}(0)=\mathbf{0}$ , allora abbiamo che:

$$\mathbf{w}(n+1) = \sum_{i=1}^n \mathbf{x}(i) \quad (6.15)$$

Sia  $\mathbf{w}_0$  una soluzione che classifica correttamente gli  $\mathbf{x}(i)$  e definiamo come:

$$\alpha = \min_{\mathbf{x}(n) \in X_1} \mathbf{w}_0^T \cdot \mathbf{x}(n) \quad (6.16)$$

dove  $\alpha > 0$  dal momento che la classificazione è corretta con  $\mathbf{w}_0$  vettore dei pesi e  $\mathbf{x}(n) \in X_1$ . Possiamo dedurre che:

$$\mathbf{w}_0^T \cdot \mathbf{w}(n+1) = \mathbf{w}_0^T \cdot \sum_{i=1}^n \mathbf{x}(i) \geq \alpha \cdot n$$

Utilizziamo a questo punto la disuguaglianza di Cauchy-Schwarz e scriviamo:

$$\|\mathbf{w}_0\|^2 \cdot \|\mathbf{w}(n+1)\|^2 \geq [\mathbf{w}_0^T \cdot \mathbf{w}(n+1)]^2 \geq n^2 \cdot \alpha^2 \quad (6.17)$$

quindi:

$$\|\mathbf{w}(n+1)\|^2 \geq \frac{n^2 \cdot \alpha^2}{\|\mathbf{w}_0\|^2}$$

Facciamo ora queste altre considerazioni. Riscriviamo la regola di aggiornamento applicando la norma quadrata euclidea ad entrambi i lati ottenendo:



$$\|\mathbf{w}(n+1)\|^2 = \|\mathbf{w}(n)\|^2 + \|\mathbf{x}(n)\|^2 + 2 \cdot \mathbf{w}^T(n) \cdot \mathbf{x}(n) \quad (6.18)$$

dove  $\beta > 0$  è definito come:

$$\beta = \max_{\mathbf{x}(n) \in X_1} \|\mathbf{x}(n)\|^2 \quad (6.19)$$

Quindi il peso aggiornato  $\mathbf{w}(n+1)$  è limitato superiormente ed inferiormente e di conseguenza esiste un numero massimo di iterazioni  $n_{\max}$  tale per cui:

$$\frac{n_{\max}^2 \cdot \alpha^2}{\|\mathbf{w}_0\|^2} = \|\mathbf{w}(n+1)\|^2 = n_{\max} \cdot \beta \quad (6.20)$$

da cui ricaviamo  $n_{\max}$ :

$$n_{\max} = \frac{\beta \cdot \|\mathbf{w}_0\|^2}{\alpha^2} \quad (6.21)$$

Quindi abbiamo dimostrato che se  $X_1$  e  $X_2$  sono due insiemi linearmente separabili, il percettrone converge con al massimo  $n_{\max}$  iterazioni utilizzando il metodo fixed increment adaptation.

Nel caso di  $\eta(n)$  variabile, consideriamo  $\eta(n)$  come il più piccolo intero per cui:

$$\eta(n) \cdot \mathbf{x}^T(n) \cdot \mathbf{x}(n) > \left| \mathbf{w}^T(n) \cdot \mathbf{x}(n) \right| \quad (6.22)$$

avremo che se all'iterazione  $n$   $\mathbf{w}^T(n) \cdot \mathbf{x}(n)$  ha un segno errato, se ripresentiamo lo stesso input all'iterazione successiva avremo che il segno diventa corretto infatti:

1. se  $\mathbf{w}^T(n) \cdot \mathbf{x}(n) \leq 0$  e  $\mathbf{x}(n) \in X_1$  abbiamo:  
 $\mathbf{w}^T(n+1) \cdot \mathbf{x}(n) = \mathbf{w}^T(n) \cdot \mathbf{x}(n) + \eta(n) \cdot \mathbf{x}^T(n) \cdot \mathbf{x}(n) > 0$
2. se  $\mathbf{w}^T(n) \cdot \mathbf{x}(n) > 0$  e  $\mathbf{x}(n) \in X_2$  abbiamo:  
 $\mathbf{w}^T(n+1) \cdot \mathbf{x}(n) = \mathbf{w}^T(n) \cdot \mathbf{x}(n) - \eta(n) \cdot \mathbf{x}^T(n) \cdot \mathbf{x}(n) < 0$

*In altre parole ogni pattern viene presentato più volte di seguito finché non viene classificato correttamente.*

## 6.8.2 Algoritmo di apprendimento del percettrone

Possiamo sintetizzare quando descritto precedentemente, in una serie di passi che costituiscono l'algoritmo di apprendimento del percettrone:

1. **Inizializzazione:** si setta  $\mathbf{w}(0)=\mathbf{0}$  ed  $n=0$ ;
2. **Attivazione:** al tempo  $n$  si attiva il perceptrone applicando l'input  $\mathbf{x}(n)$  e la risposta desiderata  $d(n)$ ;
3. **Calcolo dell'output:** si calcola l'output secondo la seguente formula:  

$$\mathbf{y}(n) = \text{sgn}(\mathbf{w}^T(n) \cdot \mathbf{x}(n))$$

dove  $\text{sgn}$  è la funzione segno definita come: 
$$\text{sgn}(x) = \begin{cases} +1 & \text{se } x > 0 \\ -1 & \text{se } x \leq 0 \end{cases}$$

4. **Aggiornamento dei pesi:** si aggiornano i pesi secondo la seguente formula:  

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta \cdot [d(n) - y(n)] \cdot \mathbf{x}(n)$$

dove  $d(n) = \begin{cases} +1 & \text{se } x(n) \in X_1 \\ -1 & \text{se } x(n) \in X_2 \end{cases}$

5. **Continuazione:** si incrementa  $n$  e si torna a 2.

Il processo termina quando si è giunti ad uno step  $n_0$  tale per cui  $\mathbf{w}(n_0) = \mathbf{w}(n_0+1) = \mathbf{w}(n_0+2) = \dots$  ovvero finché tutti i pattern sono classificati correttamente senza modifiche ai pesi.

Il parametro di apprendimento è una costante reale positiva limitata all'intervallo  $(0,1]$ , tenendo conto di queste considerazioni:

- più piccola è  $\eta$  più si tiene in considerazione il peso passato, garantendo stime stabili, a scapito della velocità di convergenza;
- più è grande e maggiore è la velocità di convergenza.

## 6.9. Esempio di applicazione: Operatori booleani

Abbiamo visto che il perceptrone è in grado di classificare input che sono linearmente separabili. Possiamo costruire perceptron che risolvono l'AND, l'OR, il NOT, ma non possiamo costruire un perceptrone che risolve lo XOR.

### 6.9.1 NOT

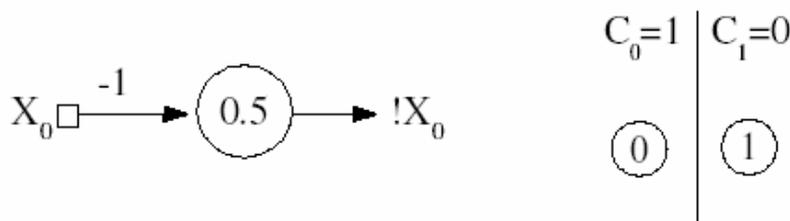


Figura 6.15. Perceptrone per operatore NOT.

## 6.9.2 AND

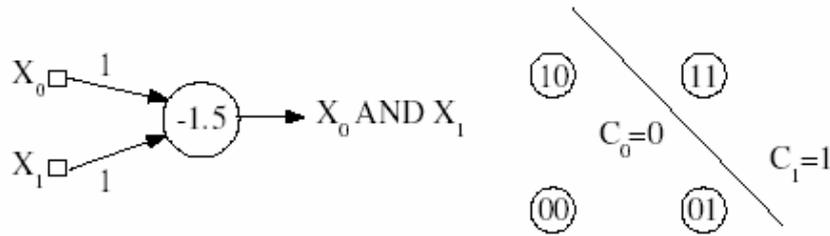


Figura 6.16. Percettrone per operatore AND.

## 6.9.3 OR

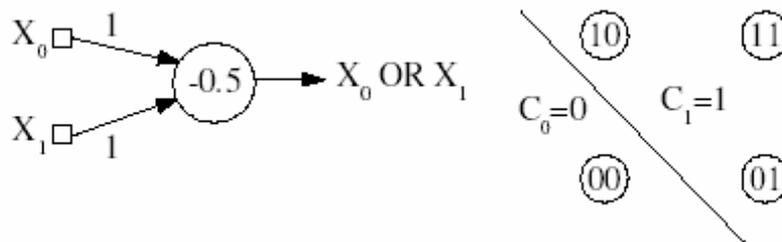


Figura 6.17. Percettrone per operatore OR.

## 6.9.4 XOR

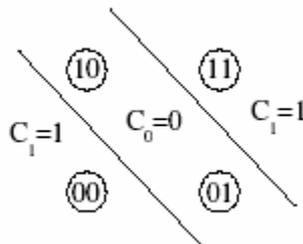


Figura 6.18. Non separabilità lineare dell'operatore XOR

## 6.10 Multilayers Perceptrons

La rete consiste in un insieme di ingressi (input layer), uno o più strati nascosti di neuroni (hidden layers) e un insieme di neuroni di uscita (output layer). Il segnale di input si propaga attraverso la rete in avanti da layer a layer.

Una rete di questo tipo ha 3 caratteristiche distintive:

- ogni neurone include una **funzione di attivazione non lineare differenziabile** (ad es: funzione sigmoideale);
- la rete contiene una o più strati nascosti (**hidden layers**) che non fanno parte né dell'input né dell'output della rete;
- la rete ha un **alta connettività**.

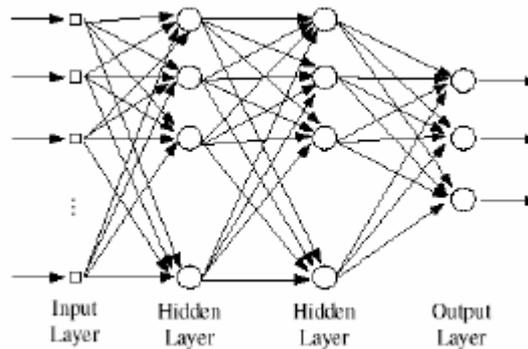
Un metodo efficiente per l'addestramento della rete è quello del **back-propagation**.

In questa rete troviamo 2 tipi di segnali:

1. **segnali di funzione**: un segnale di funzione è un segnale di input o stimolo che entra nel layer di input, si propaga in avanti attraverso i layer nascosti per emergere al layer di uscita come segnale di output;
2. **segnale di errore**: un segnale di errore ha origine nel layer di uscita e si propaga all'indietro attraverso la rete.

Ogni neurone nascosto o d'uscita di un multilayer perceptron esegue 2 computazioni:

1. la **computazione del segnale di funzione** espressa come funzione non lineare continua di un segnale di input e dei pesi sinaptici associati al neurone;
2. la **computazione di un vettore gradiente** necessario per aggiornamento dei pesi sinaptici.



*Figura 6.19. Multilayer perceptrons*

### 6.10.1 Algoritmo di back-propagation

Distinguiamo 2 casi:

1. neurone del layer di output;
2. neurone di un hidden layer;

### 6.10.2 Neurone j di output

Il **segnale d'errore** del neurone di output j all'iterazione n (presentazione dell'n-esimo esempio del training set) è definito da:

$$e_j(n) = d_j(n) - y_j(n) \quad (6.23)$$

dove:

- $d_j(n)$  è l'output atteso del neurone j;
- $y_j(n)$  è l'output del neurone j.



L'**errore totale** dell'output layer presentando l'n-esimo esempio del training set è definito come:

$$E(n) = \frac{1}{2} \cdot \sum_{j \in C} e_j^2(n) \quad (6.24)$$

dove C è l'insieme dei neuroni che formano l'output layer.

Sia N il numero totale di pattern contenuti nel training set. L'errore quadrato medio rappresenta la funzione costo ed è dato da

$$E_{av} = \frac{1}{N} \cdot \sum_{i=1}^N E(n) \quad (6.25)$$

L'obiettivo dell'addestramento è quello di minimizzare  $E_{av}$  aggiustando opportunamente i parametri liberi della rete neurale.

Considereremo un metodo semplice che aggiorna i pesi di pattern in pattern fino al raggiungimento di un'epoca. Un'**epoca** è una presentazione completa dell'intero training set. L'aggiustamento dei pesi viene fatto in base all'errore calcolato per ogni pattern presentato alla rete.

La media aritmetica di queste variazioni di peso su pattern singoli del training set sono una stima della variazione reale che risulterebbe da modifiche di peso che minimizzino la funzione costo  $E_{av}$  sull'intero training set.

Per minimizzare la funzione costo si adotta il metodo della discesa del gradiente, e gli aggiornamenti di peso avvengono in verso opposto al gradiente.

$$\Delta w_{ji}(n) = -\eta \cdot \frac{\partial E(n)}{\partial w_{ji}(n)} \quad (6.26)$$

dove  $\eta$  è il fattore di apprendimento.

Calcoliamo il gradiente:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

applicando la regola della catena abbiamo che il gradiente è pari al prodotto di 4 derivate parziali. Calcoliamo ciascuna derivata come,

$$\frac{\partial E(n)}{\partial e_j(n)} = \frac{1}{2} \cdot \sum_{k \in C} \frac{\partial e_k^2(n)}{\partial e_j(n)} = e_j(n)$$

e dal momento che consideriamo un neurone di output abbiamo che:

$$\begin{aligned}\frac{\partial e_j(n)}{\partial y_j(n)} &= \frac{\partial (d_j(n) - y_j(n))}{\partial y_j(n)} = -1 \\ \frac{\partial y_j(n)}{\partial v_j(n)} &= \frac{\partial \varphi[v_j(n)]}{\partial v_j(n)} = \varphi'[v_j(n)] \\ \frac{\partial v_j(n)}{\partial w_{ji}(n)} &= \sum_{k=0}^m y_k(n) \cdot \frac{\partial w_{jk}(n)}{\partial w_{ji}(n)} = y_i(n)\end{aligned}$$

da cui ricaviamo che,

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -\delta_j(n) \cdot y_i(n)$$

dove  $\delta_j(n)$  viene definito **gradiente locale**, ed equivale a:

$$\delta_j(n) = -\frac{\partial E(n)}{\partial v_j(n)} = -\frac{\partial E(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n) \cdot \varphi'[v_j(n)]$$

Quindi la variazione di peso sinaptico tra il neurone  $i$  e  $j$  relativo all' $n$ -esimo esempio del training set è la seguente:

$$\Delta w_{ji}(n) = \eta \cdot \delta_j(n) \cdot y_i(n) \quad (6.27)$$

### 6.10.3 Neurone $j$ nascosto

Quando il neurone appartiene ad uno strato nascosto, non abbiamo una specifica risposta desiderata. Il segnale di errore deve essere determinato ricorsivamente dal segnale di errore di tutti i neuroni ai quali questo neurone nascosto è connesso.

Eseguiamo i calcoli considerando il nodo appartenente all'ultimo strato nascosto. Ridefiniamo a questo proposito il gradiente locale come,

$$\delta_j(n) = -\frac{\partial E(n)}{\partial v_j(n)} = -\frac{\partial E(n)}{\partial e_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} = -\frac{\partial E(n)}{\partial y_j(n)} \cdot \varphi'[v_j(n)]$$

Calcoliamo la derivata:

$$\frac{\partial E(n)}{\partial y_j(n)} = \frac{1}{2} \cdot \sum_{k \in C} \frac{\partial e_k^2(n)}{\partial e_k(n)} \cdot \frac{\partial e_k(n)}{\partial y_k(n)} \cdot \frac{\partial y_k(n)}{\partial v_k(n)} \cdot \frac{\partial v_k(n)}{\partial y_j(n)} = -\sum_{k \in C} e_k(n) \cdot \varphi'[v_k(n)] \cdot w_{kj}(n)$$

dove trattandosi di un neurone nascosto  $j \notin C$ .

A questo punto abbiamo che  $\delta_k(n) = e_k(n) \cdot \varphi'[v_k(n)]$  e quindi:

$$\frac{\partial E(n)}{\partial y_j(n)} = -\sum_{k \in C} \delta_k(n) \cdot w_{kj}(n)$$

Abbiamo supposto che  $j$  fosse collegato a neuroni di output ma in generale questa formula vale anche se  $j$  fosse collegato ad un altro strato nascosto. In questo caso però la sommatoria non è estesa ai neuroni di uscita ma ai neuroni dello strato successivo. Per fare questo possiamo



considerare la sommatoria estesa a tutti i neuroni perché  $w_{kj} = 0$  se  $j$  non è adiacente a  $k$ ; quindi in generale:

$$\frac{\partial E(n)}{\partial y_j(n)} = -\sum_k \delta_k(n) \cdot w_{kj}(n)$$

A questo punto il gradiente locale del neurone nascosto  $j$  relativo all' $n$ -esimo esempio del training set è:

$$\delta_j(n) = \varphi'[v_j(n)] \cdot \sum_k \delta_k(n) \cdot w_{kj}(n) \quad (6.28)$$

## 6.11 In Generale

In generale la variazione di peso sinaptico tra il neurone  $i$  e  $j$  relativo all' $n$ -esimo esempio del training set è dato da:

$$\Delta w_{ji}(n) = \eta \cdot \delta_j(n) \cdot y_i(n) \quad (6.29)$$

dove  $\delta_j(n) = \begin{cases} e_j(n) - \varphi[v_j(n)] & \text{se } j \text{ è un neurone di output} \\ \varphi'[v_j(n)] \cdot \sum_k \delta_k(n) \cdot w_{kj}(n) & \text{altrimenti} \end{cases}$

Questo modello prende il nome di apprendimento **on-line** in quanto la rete apprende un esempio alla volta. Esiste però un altro modello detto **off-line** o **batch** che considera tutti gli esempi del training set alla volta, anche se questo è meno coerente al modello biologico. Per l'apprendimento off-line abbiamo la seguente funzione costo:

$$E = \frac{1}{2} \cdot \sum_n \sum_{j \in C} e_j^2(n) \quad (6.30)$$

da cui deriva la seguente regola di apprendimento:

$$\Delta w_{ji}(n) = \eta \cdot \sum_n \delta_j(n) \cdot y_i(n) \quad (6.31)$$

## 6.12 Fattore di Apprendimento

La scelta del fattore di apprendimento influenza molto il comportamento dell'algoritmo, infatti se scegliamo valori troppo piccoli, la convergenza sarà lenta, mentre se scegliamo valori troppo grandi si rischia di avere una rete instabile con comportamento oscillatorio.

Un metodo semplice per incrementare il fattore di apprendimento senza il rischio di rendere la rete instabile è quello di modificare la regola di aggiornamento inserendo il **momento**.

$$\Delta w_{ji}(n) = \alpha \cdot \Delta w_{ji}(n-1) + \eta \cdot \delta_j(n) \cdot y_i(n) \quad (6.33)$$

dove  $\alpha$  è un numero positivo.



Se espandiamo ricorsivamente la formula otteniamo

$$\Delta w_{\mu}(n) = \eta \cdot \sum_{t=0}^n \alpha^{n-t} \cdot \delta_j(t) \cdot y_i(t)$$

ricordando che  $\frac{\partial E(t)}{\partial w_{\mu}(t)} = -\delta_j(t) \cdot y_i(t)$  abbiamo:

$$\Delta w_{\mu}(n) = -\eta \cdot \sum_{t=0}^n \alpha^{n-t} \cdot \frac{\partial E(t)}{\partial w_{\mu}(t)} \quad (6.34)$$

che converge per valori di  $\alpha$  di  $[0,1)$ . Inoltre se la derivata parziale tende a mantenere lo stesso segno su iterazioni consecutive, grazie alla sommatoria, l'aggiornamento sarà per valori più ampi e quindi tende ad **accelerare nelle discese**. Se però la derivata ha segni opposti ad iterazioni consecutive la sommatoria tende a diminuire l'ampiezza dell'aggiornamento facendo in modo di avere un **effetto stabilizzante** quando abbiamo oscillazioni.

Oltre a questo l'uso del momento ha il vantaggio di **prevenire** che il processo di apprendimento incappi in **minimi locali** della funzione d'errore.

### 6.13 Criteri d'arresto

In generale non esistono criteri d'arresto ben definiti; comunque ne citiamo 3.

1. L'algoritmo di back-propagation può essere interrotto quando è nelle vicinanze di un minimo locale, ovvero quando la norma euclidea del vettore gradiente è inferiore di una soglia sufficientemente piccola.
2. L'algoritmo di back-propagation può essere interrotto quando la percentuale di variazione dell'errore quadrato medio tra due epoche consecutive è sufficientemente piccola.
3. L'algoritmo di back-propagation può essere interrotto quando la capacità di generalizzazione è adeguata.

Per il terzo punto vediamo più avanti nel dettaglio come si stima la capacità di generalizzazione di una rete.

### 6.14 Regole euristiche per migliorare l'algoritmo di back-propagation.

Vediamo alcune regole che consentono di migliorare le prestazioni dell'algoritmo di back-propagation:

- Se abbiamo un training set grande e altamente ridondante è consigliato l'uso del metodo di back-propagation online.



- Ogni pattern scelto da presentare alla rete dovrebbe massimizzare il contenuto informativo ovvero essere radicalmente differente dagli esempi usati precedentemente e in generale portare la rete a sbagliare. In questo modo presentiamo alla rete un training set che cerca di coprire in modo completo lo spazio degli input.
- Se la rete viene addestrata con una funzione di attivazione antisimmetrica, questa apprende più velocemente. Una funzione è antisimmetrica se  $\varphi(-x) = -\varphi(x)$ .
- E' importante che la risposta desiderata venga scelta all'interno del codominio della funzione di attivazione perché altrimenti l'algoritmo tende a portare i parametri liberi della rete a infinito.
- Ogni vettore di input dovrebbe venire pre-processato in modo che la media calcolata sull'intero training set tenda a 0 o altrimenti sia piccola se comparato con la sua deviazione standard. Se supponiamo vettori di input con componenti positive avremo che i pesi del primo layer nascosto si incrementano/decrementano insieme. Se il vettore dei pesi di un neurone deve cambiare direzione lo può fare solo zigzagando attraverso la superficie dell'errore, cosa che tipicamente è lenta e quindi va evitata. Questo non può accadere se gli input vengono normalizzati come precedentemente detto. Ulteriori normalizzazioni che possono essere fatte sono la scorrelazione delle variabili di input seguita da una scalatura in modo che le loro covarianze siano approssimativamente uguali. In questo modo ci si assicura che i diversi pesi sinaptici della rete apprendano approssimativamente alla stessa velocità.
- Se inizializziamo i pesi sinaptici con valori grandi avremo che la rete giunge alla saturazione; se ciò accade il gradiente locale assumerebbe valori piccoli che rallenterebbero il processo di apprendimento. Se viceversa i pesi fossero troppo piccoli, l'algoritmo si troverebbe a dover operare su una area molto piatta attorno all'origine della superficie d'errore e sfortunatamente l'origine è un punto stazionario né di minimo né di massimo.

Vediamo allora come inizializzare la rete facendo le seguenti considerazioni. Supponiamo di avere una rete con funzione di attivazione la tangente iperbolica, e con valore soglia settata a 0, quindi  $w_{j0} = 0, \forall j$ .

Esprimiamo l'input netto di un neurone come:

$$v_j = \sum_{i=1}^m w_{ji} \cdot y_i \quad (6.35)$$

dove  $m$  è il numero di neuroni della rete.

Supponiamo inoltre che :

- la media degli  $i$ -esimi componenti dell'input  $y_i$  di ciascun neurone sia nulla e la loro varianza sia unitaria:

$$\begin{aligned}\mu_{y_i} &= E[y_i] = 0, \quad \forall i \\ \sigma_{y_i}^2 &= E[(y_i - \mu_{y_i})^2] = E[y_i^2] = 1, \quad \forall i\end{aligned}$$

- gli input siano non correlati con gli altri input e con i pesi sinaptici e poniamo:

$$E[y_i \cdot y_k] = \begin{cases} 1 & \text{se } k = i \\ 0 & \text{se } k \neq i \end{cases}$$

- i pesi siano scelti da un insieme uniformemente distribuito con media zero:

$$\mu_{w_{ij}} = E[w_{ij}] = 0, \quad \forall (i, j)$$

e varianza

$$\sigma_w^2 = E[(w_{ij} - \mu_w)^2] = E[w_{ij}^2], \quad \forall (i, j)$$

Definiamo a questo punto la media e la varianza dell'input netto  $v_j$

$$\mu_{v_j} = E[v_j] = \sum_{i=1}^m E[w_{ji} \cdot y_i] = \sum_{i=1}^m E[w_{ji}] \cdot E[y_i]$$

dove  $E[w_{ji} \cdot y_i] = E[w_{ji}] \cdot E[y_i]$  in quanto scorrelati.

$$\begin{aligned}\sigma_v^2 &= E[(v_j - \mu_{v_j})^2] = E[v_j^2] = \sum_{i=1}^m \sum_{k=1}^m E[w_{ji} \cdot w_{jk} \cdot y_i \cdot y_k] \\ &= \sum_{i=1}^m \sum_{k=1}^m E[w_{ji} \cdot w_{jk}] \cdot E[y_i \cdot y_k] = \sum_{i=1}^m E[w_{ji}^2] = m \cdot \sigma_w^2\end{aligned} \quad (6.36)$$

dove  $E[w_{ji} \cdot w_{jk} \cdot y_i \cdot y_k] = E[w_{ji} \cdot w_{jk}] \cdot E[y_i \cdot y_k]$  in quanto i pesi e gli input sono scorrelati.

Ora vogliamo che la deviazione standard dell'input netto stia nella zona di transazione, tra l'area lineare e quella satura della funzione di attivazione. Poniamo quindi  $\sigma_v = 1$  quindi abbiamo che:

$$\begin{aligned}m \cdot \sigma_w^2 &= 1 \\ \sigma_w &= \frac{1}{\sqrt{m}}\end{aligned} \quad (6.37)$$

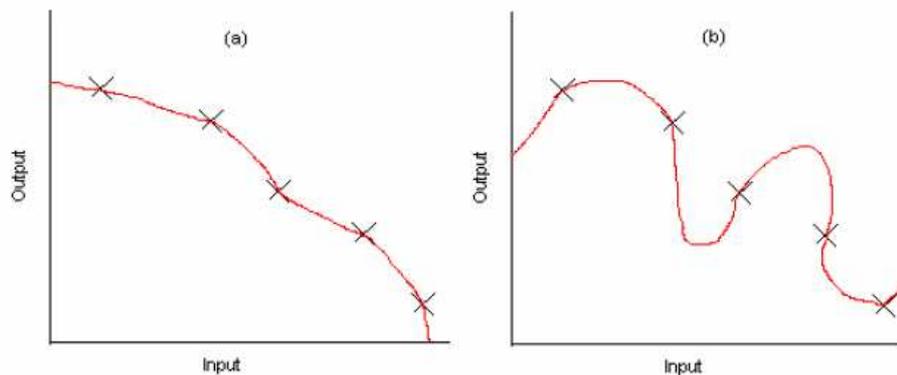
Anziché vedere  $m$  come il numero di neuroni consideriamolo come numero di connessioni sinaptiche della rete. Quindi è desiderabile che i pesi sinaptici siano scelti da una distribuzione uniforme che abbia media zero e varianza pari al reciproco del numero di connessioni sinaptiche della rete neurale.

## 6.15 Generalizzazione

Una rete neurale generalizza bene se la mappa input/output che genera la rete è corretta (o quasi) per esempi di test mai presentati alla rete in fase di training. Si assume che gli esempi di test siano tratti dalla stessa popolazione usata per generare il training set.

Il problema di apprendimento può essere visto come un problema di “approssimazione di una curva”. La rete di per sé può essere considerata semplicemente una mappa input/output non lineare, quindi una buona generalizzazione può essere vista come una buona interpolazione dei dati di input.

Una rete progettata per generalizzare bene produce mappature input/output corrette anche se l’input è lievemente differente dagli esempi usati in fase di training; se però la rete viene addestrata con troppi esempi, la rete rischia di memorizzare il training set. Questo fenomeno è detto **overfitting** o **overtraining**. Una rete “sovra-addestrata” perde la capacità di generalizzazione.



**Figura 6.20.** Esempio di buona generalizzazione (a), e di una cattiva generalizzazione (overfitted) (b)

La generalizzazione è influenzata da 3 fattori:

1. le dimensioni del training set;
2. l’architettura della rete neurale;
3. la complessità del problema.

Alla luce del fatto che non abbiamo controllo sul punto (3), possiamo vedere il problema della generalizzazione da 2 punti di vista:

1. l’architettura della rete è prefissata e lo scopo è determinare una dimensione del training set ottimale per una buona generalizzazione;
2. la dimensione del training set è prefissata e lo scopo è di determinare la migliore architettura di rete per una buona generalizzazione.

Vediamo un'euristica per il punto (1).

Nella pratica si è visto che si ha una buona generalizzazione, se la dimensione del training set  $N$  soddisfa la condizione

$$N = O\left(\frac{W}{\varepsilon}\right) \quad (6.38)$$

dove  $W$  è il numero totale di parametri liberi nella rete e  $\varepsilon$  denota la frazione di errori di classificazione permessi sui dati di test.

## 6.16 Approssimazioni di Funzioni

*Un perceptrone multilayer addestrato con back-propagation può essere visto come un approssimatore di funzioni universale.*

**Teorema dell'approssimazione universale:** sia  $\varphi$  una funzione continua, non costante, limitata e monotona crescente. Sia un ipercubo  $I_{m_0}$  unitario  $m_0$ -dimensionale  $[0,1]^{m_0}$ . Lo spazio delle funzioni continue su  $I_{m_0}$  lo denotiamo con  $C(I_{m_0})$ . Allora dati una qualunque funzione  $f \in C(I_{m_0})$  e  $\varepsilon > 0$ , esiste un intero  $m_1$  e insiemi di costanti reali  $\alpha_i$ ,  $\beta_i$  e  $w_{ij}$ , dove  $i = 1, \dots, m_1$  e  $j = 1, \dots, m_0$  tale che possiamo definire,

$$F(x_1, \dots, x_{m_0}) = \sum_{i=1}^{m_1} \alpha_i \cdot \varphi\left(\sum_{j=1}^{m_0} w_{ij} \cdot x_j + \beta_i\right) \quad (6.39)$$

come una realizzazione approssimata della funzione  $f$ ; ovvero:

$$|F(x_1, \dots, x_{m_0}) - f(x_1, \dots, x_{m_0})| < \varepsilon \quad (6.40)$$

per ogni  $x_1, \dots, x_m$  che giace nello spazio di input.

Il teorema è di sola esistenza perché non ci dice come calcolare l'insieme di costanti necessari a creare  $F$ . Il teorema dice quindi che uno strato nascosto è sufficiente ad un perceptrone multilayer per una  $\varepsilon$ -approssimazione di un generico training set rappresentato dall'insieme di input  $x_1, \dots, x_m$  e output desiderati  $f(x_1, \dots, x_m)$ .

## 6.17 Cross-validation

Vediamo ora uno strumento per risolvere il problema della selezione dell'architettura della rete neurale. Più precisamente questo strumento consente, dato un insieme di possibili modelli, di scegliere quello migliore secondo certi criteri.



Inizialmente il set di dati viene ripartito in modo random in un training set e un test set. Il training set è ulteriormente ripartito in 2 insiemi disgiunti:

- **Estimation subset:** usato per il training del modello;
- **Validation subset:** usato per validare il modello.

La motivazione è quella di validare il modello su un set di dati diverso da quello usato per l'addestramento. In questo modo possiamo usare il training set per stimare le prestazioni dei vari modelli candidati e scegliere quindi il migliore. C'è comunque la possibilità che il modello più performante in realtà è incappato in un overfitting del validation set. Per evitare questo si ricorre al test set per verificare la capacità di generalizzazione della rete.

### 6.17.1 Selezione del Modello

Il problema della selezione del modello è essenzialmente quello di scegliere il perceptrone con il miglior valore di  $W$ , il numero di parametri liberi. Più precisamente posto che la risposta desiderata relativa ad un input  $\mathbf{x}$  sia  $d \in \{0,1\}$ , definiamo l'**errore di generalizzazione** come:

$$\varepsilon_g(F) = P(F(\mathbf{x}) \neq d), \quad \mathbf{x} \in X \quad (6.41)$$

Supponiamo di avere a disposizione il seguente training set

$$Ts = \{(\mathbf{x}_i, d_i)\}_{i=1}^N$$

L'obiettivo è di selezionare la particolare funzione  $F(\mathbf{x}, \mathbf{w})$ , che minimizza l'errore di generalizzazione, che risulta presentando alla rete input dal test set.

Per ogni dimensione del test set  $N$ , possiamo sempre trovare un perceptrone multilayer con un numero sufficiente di parametri liberi  $W_{\max}(N)$  tale per cui il training set  $Ts$  è interpolato adeguatamente. Chiamiamo  $W_{\max}(N)$  **fitting number**. Una procedura ragionevole di selezione del modello sceglierebbe una funzione  $F(\mathbf{x}, \mathbf{w})$  che richiede  $W \leq W_{\max}(N)$ .

Sia  $r \in [0,1]$  un parametro che determina il partizionamento di  $Ts$  nell'estimation subset e validation subset. L'estimation subset denotato con  $TS'$  è usato per addestrare una sequenza di perceptron multilayer di complessità crescente con un numero di parametri liberi  $W \leq W_{\max}[(1-r) \cdot N]$ .

L'uso del cross-validation porta alla scelta di

$$F_{cv} = \min_{k=1,2,\dots,v} \{e_t''(F_k)\} \quad (6.42)$$



dove  $v$  corrisponde a  $Wv \leq W_{\max} [(1-r) \cdot N]$  e  $e_t(F_k)$  è l'errore di classificazione prodotto da  $F_k$  se testato sul validation subset  $T_s''$ , consistente in  $r \cdot N$  esempi.

Il problema che ci poniamo ora è come determinare il parametro  $r$ . Identifichiamo diverse proprietà qualitative del valore ottimo di  $r$ :

- se la complessità della funzione target è piccola comparata con il numero di esempi  $N$ , la prestazione del cross-validation non è influenzata dalla scelta di  $r$ ;
- se la complessità della funzione target è maggiore comparata col numero di esempi  $N$ , la scelta di  $r$  ha un effetto pronunciato sulle prestazioni del cross-validation;
- un valore di  $r$  fisso opera quasi in maniera ottimale per un largo numero di funzioni target complesse.

Un valore fisso di  $r$  che risulta essere buono è 0.2, ovvero 80% estimation subset, 20% validation subset.

## 6.18 Reti di Hopfield

Le reti di Hopfield sono:

- **reti ricorrenti** ad uno strato in cui ogni neurone è connesso a tutti gli altri (sono assenti connessioni con sé stesso);
- **simmetriche**: perché hanno la matrice dei pesi sinaptici simmetrica, quindi  $w_{ji} = w_{ij}$ ,  $\forall i, j$ ;
- **non lineari**: ogni neurone ha una funzione di attivazione non lineare invertibile.

Possiamo distinguere diverse tecniche di aggiornamento dello stato dei neuroni:

- **aggiornamento asincrono**: in cui si aggiorna un neurone alla volta;
- **aggiornamento sincrono**: tutti i neuroni vengono aggiornati allo stesso istante;
- **aggiornamento continuo**: in cui tutti i neuroni si aggiornano continuamente.

Vedremo le reti di Hopfield nel caso discreto con aggiornamento asincrono e nel caso continuo con aggiornamento continuo.

### 6.18.1 Caso Discreto

Nel caso discreto si usano neuroni di McCulloch-Pitts, in cui però l'input netto del generico neurone  $i$  è dato da

$$v_i = \sum_{j \neq i} w_{ij} \cdot x_j + I_i \quad (6.43)$$

dove  $x_j$  è lo stato del neurone  $j$  e  $I_i$  è una corrente esterna (input esterno) applicata al neurone  $i$ . Se  $I_i = 0$  abbiamo il tradizionale neurone di McCulloch-Pitts.

Ogni neurone aggiorna il proprio stato in modo asincrono nel seguente modo:

$$x_i(t) = \begin{cases} -1 & \text{se } v_i(t) < 0 \\ x_i(t-1) & \text{se } v_i(t) = 0 \\ +1 & \text{se } v_i(t) > 0 \end{cases}$$

Abbiamo detto la matrice dei pesi deve essere simmetrica perché questa è una condizione sufficiente alla convergenza, in quel caso infatti il sistema ha una funzione di energia di Lyapunov che viene minimizzata con l'evolvere del processo. Inoltre abbiamo detto che non ci devono essere autoloop, quindi settiamo  $w_{ii} = 0, \forall i$

La funzione quadratica di energia è la seguente:

$$E(t) = -\frac{1}{2} \cdot \mathbf{x}(t)^T \cdot \mathbf{W} \cdot \mathbf{x}(t) - \mathbf{x}(t)^T \cdot \mathbf{I} \quad (6.44)$$

Abbiamo detto che i neuroni si aggiornano in modo asincrono, quindi supponiamo sia  $h$  il neurone che ha cambiato stato, allora:

$$\Delta E(t) = -\frac{1}{2} \cdot \sum_i \sum_j w_{ij} \cdot x_i(t+1) \cdot x_j(t+1) - \sum_i I_i \cdot x_i(t+1) + \frac{1}{2} \cdot \sum_i \sum_j w_{ij} \cdot x_i(t) \cdot x_j(t) + \sum_i I_i \cdot x_i(t)$$

ora dal momento che  $h$  è l'unico stato che è cambiato abbiamo che,

$$x_i(t+1) = x_i(t), \quad \forall i \neq h$$

quindi

$$\begin{aligned} \Delta E(t) = & -\frac{1}{2} \cdot \left\{ \sum_j w_{hj} \cdot [x_h(t+1) \cdot x_j(t+1) - x_h(t) \cdot x_j(t)] + \sum_j w_{jh} \cdot [x_h(t+1) \cdot x_j(t+1) - x_h(t) \cdot x_j(t)] \right\} - \\ & + I_h \cdot \Delta x_h(t) = -\sum_j w_{hj} \cdot x_j(t) \cdot \Delta x_h(t) - I_h \cdot \Delta x_h(t) = -\Delta x_h(t) \cdot \left[ \sum_j w_{hj} \cdot x_j(t) + I_h \right] = -\Delta x_h(t) \cdot v_h(t+1) < 0 \end{aligned}$$

dove sfruttando la simmetria della matrice dei pesi e il fatto che ha gli elementi diagonali nulli, abbiamo che le sommatorie del primo passaggio sono uguali; nel terzo passaggio abbiamo sfruttato il fatto che,

$$v_h(t+1) = \sum_{j \neq h} w_{hj} \cdot x_j(t) + I_h \quad \text{e} \quad \Delta x_h(t) = x_h(t+1) - x_h(t)$$

e nell'ultimo passaggio  $-\Delta x_h(t) \cdot v_h(t+1)$  è sempre negativo in quanto per la regola di aggiornamento del neurone,

$$\text{sgn}[x_h(t+1)] = \text{sgn}[v_h(t+1)]$$

e un cambio di stato nel neurone h implica che

$$x_h(t+1) = -x_h(t)$$

e quindi:

$$\begin{aligned} \text{sgn}(\Delta E) &= -\text{sgn}[\Delta x_h(t)] \cdot \text{sgn}[v_k(t+1)] = -\text{sgn}\{x_h(t+1) - x_h(t)\} \cdot \text{sgn}[v_k(t+1)] = \\ &= -\text{sgn}\{x_h(t+1) + x_h(t+1)\} \cdot \text{sgn}[x_h(t+1)] = -\text{sgn}[x_h(t+1)]^2 = -1 < 0 \end{aligned}$$

### 6.18.2 Caso Continuo

La regola di aggiornamento dello stato di un generico neurone i è nel caso continuo la seguente:

$$x_i + C_i \cdot \frac{dx_i}{dt} = \varphi_i[v_i] = \varphi_i \left[ \sum_j w_{ij} \cdot x_j + I_i \right] \quad (6.45)$$

da cui segue

$$C_i \cdot \frac{dx_i}{dt} = -x_i + \varphi_i \left[ \sum_j w_{ij} \cdot x_j + I_i \right] \quad (6.46)$$

dove  $\varphi_i$  è una funzione di attivazione continua, non lineare e crescente associata al neurone i e  $C_i$  è una costante.

Possiamo vedere come il sistema raggiunge la stabilità solo quando il vettore velocità del sistema si azzerava ovvero quando

$$\frac{dx_i}{dt} \Big|_{=} = 0, \quad \forall i$$

In maniera del tutto equivalente possiamo esprimere l'equazione di stato non incentrando l'attenzione sulla variazione dello stato nel tempo, quanto sulla variazione dell'input netto nel tempo, pervenendo alla seguente equazione:

$$v_i + C_i \cdot \frac{dv_i}{dt} = \sum_j w_{ij} \cdot x_j + I_i = \sum_j w_{ij} \cdot \varphi_j(v_j) + I_i$$

da cui segue

$$C_i \cdot \frac{dv_i}{dt} = -v_i + \sum_j w_{ij} \cdot \varphi_j(v_j) + I_i$$

La *funzione di energia del sistema* nel caso continuo è la seguente:

$$E = -\frac{1}{2} \cdot \sum_i \sum_j w_{ij} \cdot x_i \cdot x_j + \sum_i \int_0^{x_i} \varphi_i^{-1}(x) \cdot dx - \sum_i I_i \cdot x_i \quad (6.47)$$



Così come nel caso discreto se  $W=W^T$  (matrice dei pesi simmetrica) allora la funzione è Lyapunoviana e quindi  $dE/dt \leq 0$ , dove in caso di uguaglianza ci troviamo in un punto stazionario.

Facciamo notare che non è possibile che un sistema con funzione di Lyapunov entri in ciclo, infatti trattandosi di una funzione monotona decrescente ad ogni istante temporale  $t$  abbiamo che  $E(t+1) \leq E(t)$ , dove con  $E(t)$  indichiamo il valore della funzione di energia relativa allo stato del sistema al tempo  $t$ .

Supponiamo ora di avere un ciclo nella sequenza degli stati cui corrisponde la seguente relazione della funzione di energia  $E(k) \leq E(k+1) \leq \dots \leq E(k+m) = E(k)$ , dove lo stato  $x(k+m) = x(k)$ . Questa condizione non soddisfa la condizione di monotonia a meno che non valga l'uguaglianza per tutto il ciclo infatti risulterebbe . Ma se vale l'uguaglianza non avremo un ciclo ma un punto stazionario, quindi non possiamo avere cicli.

### 6.18.3 Problemi

I problemi che si hanno utilizzando una rete di Hopfield come strumento di ottimizzazione sono:

- in molti casi abbiamo un numero di connessione dell'ordine di  $n^4$ ;
- abbiamo un numero di unità dell'ordine di  $n^2$ ;
- le soluzioni ottenute sono raramente ammissibili;
- difficoltà nel determinare i parametri per risolvere il problema;
- difficoltà di evitare minimi locali.

## 6.19 Individuazione di Versamenti di Petrolio in Mare su Immagini SAR.

I versamenti di petrolio (*oil spill*) in mare rappresentano un serio problema ambientale mondiale. Ogni giorno si verificano scarichi illegali e accidentali da navi, che inevitabilmente finiscono in mare. Il database ITOPF contiene informazioni su circa 10.000 fuoriuscite di petrolio nel periodo 1970-2001 dalle petroliere e mercantili.

L'inquinamento da petrolio è un problema enorme e complesso, che ha spesso impatti significativi e di lungo-termine sull'ambiente, sull'ecologia e sulle condizioni socio-economiche della zona inquinata. Fortunatamente le fuoriuscite di petrolio più importanti sono di interesse pubblico, e di solito sono sotto il controllo delle autorità di ambientali. D'altra parte, le piccole fuoriuscite di petrolio che si verificano in mare aperto, a causa di operazioni tra navi, sono di solito non controllate e non contabilizzate ma hanno un minore interesse pubblico.

Al fine di attuare efficaci contromisure nei versamenti di petrolio, è necessario sviluppare sistemi per il monitoraggio di questi eventi. Il telerilevamento, con la sua capacità di rilevare le caratteristiche spettrali dei film d'olio, è stato sempre uno strumento importante in questo tipo di monitoraggio.

Le immagini SAR (Radar ad Apertura Sintetica), sono ampiamente utilizzate per la rilevazione delle fuoriuscite di petrolio nell'ambiente marino, poiché sono indipendenti dalla luce del sole e non sono, in genere, influenzate dalla nuvolosità presente nell'area del disastro. I valori del segnale RADAR (*backscattering*) generati dal petrolio presente in mare [48], sono molto simili ai valori del segnale forniti dai tratti di mare molto calmo e altri fenomeni oceanici denominati "*look-alike*" (come ad esempio, le correnti, mulinelli, zone frontali, celle di pioggia, ecc). Quest'ultime circostanze, aumentano la complessità insita nell'individuazione dei potenziali versamenti di petrolio in mare mediante l'uso delle immagini SAR, poiché esistono molti fattori che concorrono a generare falsi allarmi che portano alla mancata individuazione di quanto ricercato. Creare un sistema di *pattern recognition automatico*, robusto e affidabile, nell'*individuazione degli oil spill* rappresenta la sfida principale della ricerca in questo campo.

Le caratteristiche geometriche, contestuali e i livelli di backscattering estratti (caratteristiche delle macchie di olio) dalle immagini SAR in fase di elaborazione (figura 6.21), sono quelle necessarie per affrontare il problema della **classificazione** delle macchie rilevate (oil spill o falso allarme). E a partire da queste informazioni, esistono diversi metodi di classificazione che variano per il livello di automatizzazione, complessità e numero d'input necessari per raggiungere un certo livello di confidenza rispetto a quanto classificato. E tanto maggiore sarà la capacità discriminativa delle caratteristiche considerate, tanto più preciso e semplice sarà il processo di classificazione [36].



**Figura 6.21.** Catena di processing per l'individuazione degli oil spill sulle immagini SAR.

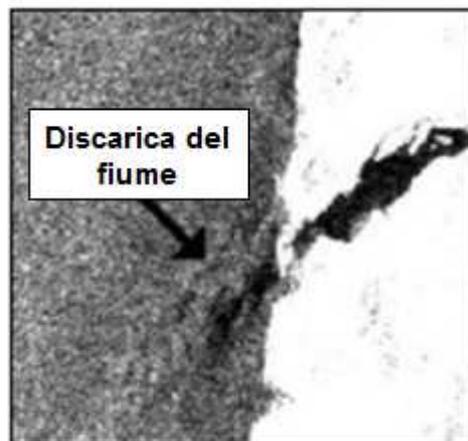
Esistono due approcci nell'implementazione della metodologia utilizzata per rilevare i versamenti di petrolio: manuale e automatico [48], [49].

Il riconoscimento manuale è un metodo che dipende completamente dalla capacità dell'analista e della sua esperienza nel settore. Tutte le macchie suscettibili di analisi (potenziali versamenti) vengono interpretate visivamente dall'operatore, il quale si serve di informazioni

contestuali come la velocità dei venti, le infrastrutture, e altre fonti d'informazione per interpretare ciò che è stato rilevato. Inoltre, anche le informazioni di contrasto tra l'area interessata e il suo intorno, l'omogeneità delle macchie e la loro forme. La determinazione dei livelli di fiducia assegnati alle macchie rilevate non sono una scienza esatta, e sono sempre soggetti a un certo grado di incertezza [48].

Fiscella [50] afferma che un interprete umano di immagini ha capacità simili rispetto agli algoritmi di classificazione, nel discriminare versamenti da falsi allarmi (fig. 6.22). Sebbene questo approccio sia generalmente efficace, questa metodologia è molto dispendiosa dal punto di vista del tempo di processing poiché richiede la combinazione degli algoritmi di riconoscimento automatico con le conoscenze specifiche dell'esperto.

Oggi sono presenti sul mercato vari servizi dedicati all'identificazione e al monitoraggio di questo tipo di evento. Esistono metodologie completamente automatizzate (ad esempio quello fornito dalla *NR, Norsk Regnesentral - Norwegian Computer Center*), semi-automatiche con algoritmi controllati da esperti (fornito ad esempio dalla *QinetiQ Group*), o manuali mediante ispezioni visive effettuate da operatori esperti (*KSAT Kongsberg Satellite Services*).



**Figura 6.22.** Immagine ASAR del Fiume Miño (confine Spagna- Portogallo).

*La macchia oscura evidenziata può generare un falso allarme poiché ha una risposta simile a un versamento di petrolio.*

*Fonte: Gonzalez Vilas 2007 [54].*

Nei metodi manuali, le informazioni di contesto sono un fattore di grande importanza per la discriminazione a *posteriori* di quanto rilevato, permettendo la distinzione tra veri versamenti di petrolio e falsi allarmi. *La sfida da affrontare in questo contesto, è dunque: come tradurre le competenze degli esperti in algoritmi automatici e semiautomatici di oil spill detection.*

La tabella 6.1 riporta alcuni dei diversi fenomeni che possono provocare falsi allarmi nell'individuazione di versamenti di petrolio, rappresentando quindi un prezioso appoggio nella corretta discriminazione degli stessi.

Fenomeno geofisico	Espressione nelle immagini SAR	Frequenza geografica	Limitazioni climatiche	Valori di backscattering (dB)	Gradiente (dB/100m)
<b>Strato naturale</b>	Simili alle interazioni delle correnti	Regioni costiere e sorgenti	Scompaiono con venti superiori ai 7m/s	-24 a -15	1,5 - 3,5
<b>Iceberg</b>	Grandi aree oscure	Principalmente lungo i bordi degli Iceberg. Anche in mare aperto	In inverno o notti fredde vicino area di iceberg	-24 a -14	1 - 2
<b>Velocità del vento</b>	Grandi aree oscure	Dovunque	Velocità dei venti inferiore ai 3m/s	-26 a -18	0,3 – 0,5
<b>Celle di pioggia</b>	Celle brillanti con centri oscuri	Regioni subtropicali	Pioggia densa e venti forti	-24 a -8	1,4 - 5
<b>Ombre costiere</b>	Aree oscure vicino alle coste	In zone vicino alle coste e fiordi	Alte velocità dei venti (15m/s)	-24 a -8	1,5 - 3
<b>Interazione onde – correnti</b>	Curve fini, brillanti o oscure	Aree di correnti forti	Velocità dei venti minori di 10-12 m/s	-24 a -8	0,2 - 0,4
<b>Onde interne</b>	Serie parallele di linee brillanti e oscure	In acque poco profonde	Velocità del vento inferiore a 8m/s	-24 a -8	0,4 - 1

*Tabella 6.1. Riassunto dei fenomeni e delle caratteristiche che possono generare falsi allarmi nella oil spill detection su immagini SAR. ( Hovland 1994).*

La creazione di un insieme di regole (tabella 6.1) da utilizzare nella fase iniziale per aggiustare le probabilità dei potenziali oil spill/falsi allarmi può consentire una notevole riduzione del set di dati da analizzare. Infatti, un primo passo nell'individuazione delle aree potenziali da trattare è quello di eliminare le aree a basso regime di vento o inesistente, poiché rende impossibile la detection degli oil spill in queste condizioni. Come mostra la figura 6.23, il regime predominante di venti deboli, riduce notevolmente l'utilizzo di questo dato SAR nella nostra applicazione [52].



**Figura 6.23.** Immagine Radarsat del Lago di Maracaibo (Venezuela).

Nei metodi semi-automatici, come quelli utilizzati dal gruppo *QinetiQ*, gli algoritmi sono progettati per rilevare le zone oscure presenti nelle immagini, e il lavoro di classificazione finale viene lasciato nelle mani di un operatore esperto [48].

I diversi approcci seguiti nell'individuazione dei versamenti di petrolio, indipendentemente dal grado di automatizzazione presente negli stessi, possono condurre a divergenze nei risultati come dimostra la letteratura scientifica sull'argomento [49], a causa della notevole difficoltà a tenere sotto controllo tutti i parametri che intervengono nel processo. In tale articolo infatti si mostra appunto come, su una sequenza di 32 immagini SAR (satellite Radarsat) con 17 macchie di petrolio verificate, la metodologia del *KSAT* è riuscita a rilevare 15 macchie, l'algoritmo implementato dall'*NR* ne ha individuato 14, mentre quello della *QinetiQ* solo 12. Questi risultati evidenziano chiaramente, le difficoltà oggettive cui vanno incontro gli operatori quando assegnano un certo grado di confidenza alla macchia rilevata. Lo stesso lavoro, ha evidenziato come i tempi di processing variano mediamente da 3 a 25 minuti, lasciando intuire come questo parametro possa aumentare notevolmente all'aumentare del numero delle scene analizzate. Inoltre, un punto molto importante da tenere in considerazione nei metodi automatici, sono le differenze nelle risposte dei vari sensori SAR, poiché questi condizionano le caratteristiche delle scene riprese e quindi i parametri che intervengono nei vari algoritmi per il riconoscimento automatico degli oil spill.

Diversi sono gli autori e gli algoritmi utilizzati per affrontare questo problema. Si va dal *Support Vector Machine* (Zhang Gonzalez Hua), alle *Reti Neurali* (Del Frate, Topouzelis, Hang), alla *Logica Fuzzy* (Keramitsoglou, Karathanassi), alla *Regressione Multipla* o *Analisi Discriminante di Fisher* (Nirchio), agli *Approcci Probabilistici* (Solberg,

Fiscella, Nirchio, Topouzelis). Il grado di accuratezza di questi metodi varia dal 80 al 98%, nel riconoscimento di versamento di petrolio e di falsi allarmi, e sono stati testati su un notevole numero d'immagini SAR riprese dai satelliti ERS 1, ERS 2, Envisat e ALOS.

	Oil Spill				Falso Allarme			
	minimo	massimo	media	varianza	minimo	massimo	media	varianza
Area	382	346550	30471,09	55930,64	401	907090	47242,99	116988,16
Asimmetria	0,15	1	0,86	0,19	0,05	1	0,72	0,26
Media dello sfondo	78,94	172,21	115,94	18,56	78,12	168,74	113,97	17,52
Perimetro macchia	138	25290	2812,06	3823,01	192	32326	4233,16	5038,28
Varianza/Media dello sfondo (1)	0,24	0,54	0,39	0,08	0,25	0,54	0,42	0,07
Varianza dello sfondo	26,83	59,87	44,28	8,52	26,57	57,9	46,8	7,81
Media oggetto /Media zona	0,15	0,76	0,39	0,13	0,18	0,75	0,45	0,14
Differenza Media sfondo e minimo dell'oggetto	77,6	157,21	113,08	18,02	71,54	163,53	112,83	17,56
Media gradiente	23,36	121,29	72,36	20,67	23,84	116,05	67,58	21,27
Media oggetto/ media dello sfondo	0,15	0,8	0,38	0,14	0,17	0,75	0,41	0,14
Varianza oggetto /Varianza sfondo	0,39	0,91	0,63	0,1	0,47	0,87	0,65	0,1
Massimo del gradiente	141	255	226,62	34,11	140	255	234,92	28,72
Media del gradiente	54,19	149,36	79,85	14,38	42,83	110,51	79,11	11,66
Diffusione	1,05	22,9	5,72	4,8	1,02	22,29	4,06	4,51
Differenza media rispetto contorno	10,56	65,98	35,86	12,44	6,29	63,74	31,84	13,36
Media Intensità oggetto	17,37	130,33	43,57	18,07	17,44	91,45	46,4	15,39
Varianza oggetto /media oggetto (2)	0,3	1,31	0,72	0,29	0,27	1,6	0,72	0,26
Rapporto (2)/(1)	1,04	3,61	1,84	0,62	1,06	3,08	1,73	0,54
Varianza Intensità dell'oggetto	12,84	43,35	27,92	7,19	15,9	47,09	30,32	6,51
Perimetro/Area	0,02	0,75	0,21	0,14	0,03	0,95	0,25	0,19
Complessità	1,51	11,77	4,33	2,4	1,58	16,28	6,02	2,56
Tessitura di forma	0,17	0,24	0,22	0,01	0,21	0,29	0,23	0,01
Tessitura spettrale	14,79	51,47	33,84	8,44	18,14	58,06	36,37	7,58

**Tabella 6.2.** Risultati della classificazione (estrazione delle caratteristiche) usando 24 immagini ERS 2. Caratterizzazione di Topouzelis (2009) relativa agli oil spill e i falsi allarmi studiati.

Nel lavoro di Del Frate [53], la discriminazione è stata fatta in funzione dell'esperienza dell'analista. I risultati ottenuti portano alla conclusione che, i versamenti di petrolio (*oil spill*) hanno forme meno complesse e più sottili, presentano valori del gradiente più elevati rispetto a quelli presentati dai falsi allarmi, i quali hanno anche generalmente dimensioni maggiori.

	Oil Spill				Falso Allarme			
	minimo	massimo	media	varianza	minimo	massimo	media	varianza
Area (km <sup>2</sup> )	0,4	40,6	0,4	7,5	1,1	115,6	13,3	17,0
Perimetro (km)	4,2	117,6	28,1	22,1	7,1	396,4	52,4	57,2
Complessità	1,1	6,8	3,2	1,1	1,1	10,4	3,9	1,7
Diffusione	0,1	40,8	4,2	7,4	0,1	45,2	11,8	11,4
Varianza dell'oggetto	0,8	3,8	1,7	0,6	0,9	3,2	2,0	0,6
Varianza dello sfondo	0,8	3,0	1,1	0,4	0,9	2,3	1,5	0,4
Differenza Media dello sfondo e minimo dell'oggetto	3,2	15,7	9,2	2,6	2,6	14,9	10,8	2,2
Differenza Media dello sfondo e media dell'oggetto	0,4	10,9	4,8	1,9	0,4	9,3	5,3	1,7
Massimo gradiente	2,8	15,5	7,2	2,2	3,6	16,8	8,5	2,6
Media del gradiente	0	6,5	3,0	1,1	0	5,2	2,7	1,0
Varianza gradiente	0,8	2,7	1,4	0,4	0,6	2,6	1,5	0,5

*Tabella 6.3. Risultati della classificazione usando 600 immagini ERS 1. Caratterizzazione di Del Frate (2000) relativa agli oil spill e falsi allarmi studiati.*

Classificazione	Immagine	Area	Perimetro	Complessità	Shape factor
Oil Spill	1	25.095.000	111.800	6,30	36,62
	3	58.694	6.144	7,16	12,57
	5	25.492	4.726	8,35	41,43
Falsa allarme	2	119.965.000	357.400	9,21	1,36
	4	7.310	986	3,25	1,23

*Tabella 6.4. Risultati della classificazione su immagini ALOS/Palsar. Caratterizzazione di Hang (2009) relativa agli oil spill e falsi allarmi studiati.*

Gonzalez Vilas [54] ha realizzato un monitoraggio del disastro occorso al largo delle coste della Galicia (Spagna) nel 2002, quando la petroliera *Prestige* versò accidentalmente in mare più di 76.000 m<sup>3</sup> di combustibile. Tale studio è stato realizzato usando come riferimento le immagini ENVISAT/ASAR (più di 27), per seguire nello spazio-tempo il comportamento di una così grande fuoriuscita di greggio. In funzione di quanto analizzato, l'autore conclude che i parametri relativi alla grandezza (*area, perimetro, ecc*) non riescono, in questo caso, a separare il versamento d'idrocarburo dai falsi allarmi, servendo unicamente allo scopo descrittivo della macchia. Inoltre, ha trovato una grande quantità di falsi allarmi nei pressi delle baie, dove comunque possono essere usate le caratteristiche radiometriche, per discriminare oil spill da falsi allarmi, poiché queste evidenziano sostanziali differenze (il valor medio del segnale di backscattering della macchia rispetto alla media del suo contorno, ha avuto in questo caso un buon potere discriminatorio).



Sebbene tutti gli autori studiati (con diverse tecniche e algoritmi) raggiungono livelli accettabili di classificazioni corrette, **non tutti** ricorrono allo stesso gruppo di caratteristiche per meglio discriminare un versamento di petrolio da un falso allarme. La tabella 6.5 riporta una sintesi delle caratteristiche, più importanti e usate dalla maggioranza degli autori, nell'individuazione degli *oil spill*.

	Gonzalez 2008	Del Frate 2000	Topouzelis 2009	Keramitsoglou 2006	Karathanassi 2006	Nirchio 2004	Solberg 1999	Fiscella 2000	Brekke 2005	Hang 2009	Montali 2006
<b>Spaziali</b>											
Area		X	X	X	X	X	X	X	X	X	
Perimetro		X	X			X		X		X	
Bounding Box							X				
Centroide Oggetto										X	
Complessità	X	X	X		X		X		X	X	
Eccentricità											
Asse Maggiore e Minore										X	
Rapporto Lunghezza/Larghezza			X		X						
Estensione											
Rapporto Perimetro/Area			X			X		X			
Diametri Equivalenti											
1° Momento			X				X		X		
Fattore Forma			X								
Orientazione	X										
Asimmetria			X		X						
Diffusione		X	X			X		X			
Larghezza							X		X		
Dimensione Frattale			X								
<b>Tessiturali</b>											
Media (GLCM)			X		X						
Varianza (GLCM)											
Omogeneità (GLCM)							X				
Contrasto (GDCML)							X		X		X
Dissimilarità (GLCM)											X
Entropia (GLCM)											X
2° Momento (GLCM)											
Correlazione (GLMC)											X
Massimo Gradiente Bordo Oggetto		X	X								
Media Gradiente Bordo Oggetto		X	X				X	X	X		
Varianza Gradiente Bordo Oggetto	X	X	X								
Mediana Gradiente Borde Oggetto	X										
Media Contrasto		X	X								
Massimo Contrasto		X	X								
Rapporto Media			X			X		X			X



Oggetto/Media Sfondo											
Rapporto Varianza Oggetto/Varianza Sfondo			X			X		X			X
Rapporto Media/Varianza Intensità Oggetto (1)						X		X			
Rapporto Media/Varianza Intensità Sfondo (2)						X		X			X
Rapporto tra (1)/(2)								X			
<b>Contestuali</b>											
Venti									X		
Distanza da Grandi Aree					X				X		
Distanza da Terra	X			X	X						
Distanza a Imbarcazione e/o Infrastruttura							X		X		
Distanza da Oggetti Frattali						X					
Numero di Oggetti nell'Immagine					X			X			
Numero di Oggetti Vicini					X			X		X	
<b>Spettrali</b>											
Media Intensità			X		X	X		X			X
Minimo Intensità											
Massimo Intensità											
Varianza Intensità		X	X	X			X		X		X
Rapporto Varianza/Media				X			X				X
Mediana	X										X
Media dello Sfondo			X			X		X			X
Minimo Sfondo											
Massimo Sfondo											
Varianza Sfondo	X	X	X			X		X			X
Rapporto Varianza/Media Sfondo			X								
Rapporto tra Varianza/Media Oggetto e Varianza/Media Sfondo			X								
Differenza Media rispetto Contorno			X		X						

**Tabella 6.5.** Sintesi delle più importanti caratteristiche considerate nei processi per l'individuazione di versamenti di petrolio.

Diversi autori hanno cercato di stabilire una gerarchia in ordine di importanza, delle caratteristiche con maggiore potere discriminatorio [57], [58], [59]. Ma questo non significa necessariamente che, la combinazione di quelle con maggiore potere discriminatorio dia risultati più efficaci di quelli forniti da una combinazione di caratteristiche avente minore capacità. La spiegazione risiede nel fatto che la capacità

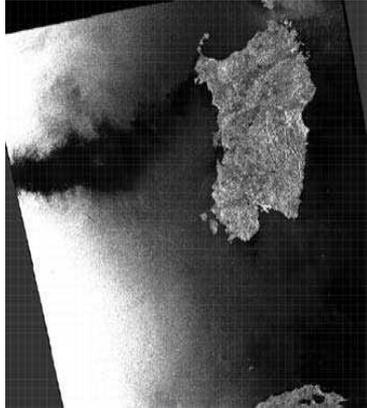
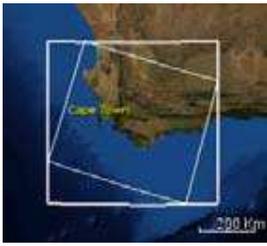
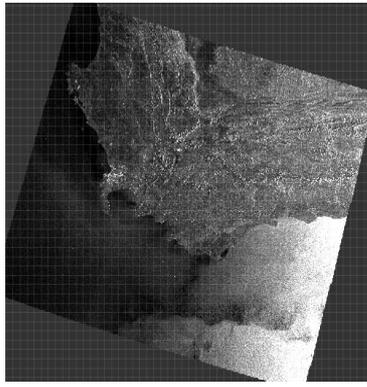
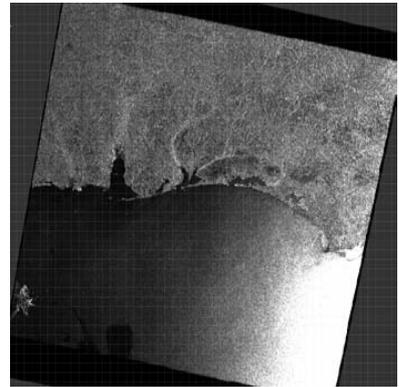
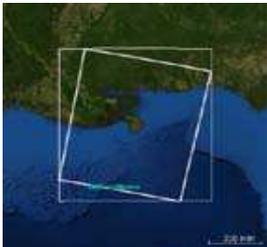
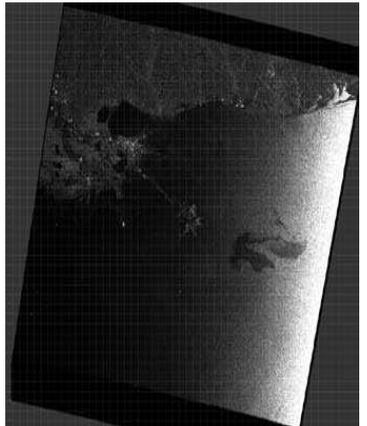
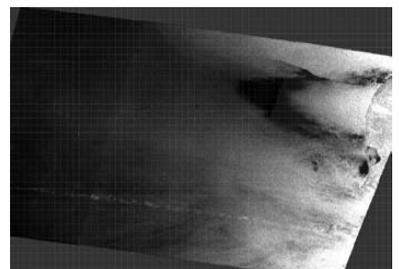


discriminatoria di una sola caratteristica può essere misurata esplicitamente, mentre nel caso di una combinazione di caratteristiche, la capacità di discriminazione non è uguale alla somma delle singole capacità [59]. E questo argomento merita certamente di essere approfondito in ulteriori ricerche, in modo da fare chiarezza sui meccanismi che stanno alla base del potere discriminatorio delle caratteristiche combinate.

Area	Orientazione	Energia (GLCM) dell'Oggetto	Omogeneità (GLMC) dello Sfondo	Rapporto Varianza/Media Intensità Zona	Rapporto Media Oggetto/Media Sfondo	Distanza Media a Barche e Piattaforme
Perimetro	Diametro Equivalente	Contrasto (GLCM) Oggetto	Energia (GLMC) Sfondo	Massima Intensità Gradiente	Rapporto Varianza Oggetto/Varianza Sfondo	Mediana Distanza da Barche e/o Piattaforme
Complessità (Del Frate)	Diffusione	Correlazione (GLMC) Oggetto	Contrasto (GLMC) Sfondo	Minimo Intensità Gradiente	Rapporto Media Oggetto/Media Zona	Varianza Distanza da Barche e/o Piattaforme
Complessità (Solberg)	Intensità Massima Oggetto	Entropia Oggetto	Correlazione (GLMC) Sfondo	Media Intensità Gradiente	Massima Distanza da Terra	Massima Velocità del Vento
Complessità (Karathanassi)	Intensità Minima Oggetto	Massima Intensità Sfondo	Entropia Sfondo	Varianza Intensità Gradiente	Minima Distanza da Terra	Minima Velocità del Vento
Fattore di Forma	Media Intensità Oggetto	Minima Intensità Sfondo	Massima Intensità Area	Mediana Intensità Gradiente	Distanza Media da Terra	Media Velocità del Vento
Asimmetria	Varianza Intensità Oggetto	Media Intensità Sfondo	Minima Intensità Area	Rapporto Varianza/Media Intensità Gradiente	Mediana Distanza da Terra	Varianza Velocità del Vento
Asse Maggiore	Mediana Intensità Oggetto	Varianza Intensità Sfondo	Media Intensità Zona	Tassa tra: Rapporto Varianza/Media Oggetto e Varianza/Media Sfondo	Varianza Distanza da Terra	Mediana Velocità del Vento
Asse Minore	Rapporto Varianza/Media Intensità Oggetto	Mediana Intensità Sfondo	Varianza Intensità Zona	Differenza Media Sfondo e Media Oggetto	Massima Distanza da Barche e/o Piattaforme	
Eccentricità	Omogeneità (GLMC) Oggetto	Rapporto Varianza/Media Intensità Sfondo	Mediana Intensità Zona	Differenza Media Sfondo e Minimo Oggetto	Minima Distanza da Barche e/o Piattaforme	

**Tabella 6.6.** Sintesi delle caratteristiche usate in questo studio per l'individuazione di versamenti di petrolio mediante reti neurali.



Immagine	Quicklook	Immagine	Quicklook
<p>Missione Envisat Tipo ASA WSM 1P Acquisizione: Passaggio discendente Track: Orbita:</p> 		<p>Missione Envisat Tipo ASA WSM 1P Acquisizione: 19/01/11 Passaggio discendente Track: 338 Orbita: 46473</p> 	
<p>Missione Envisat Tipo ASA WSM 1P Acquisizione: 31/03/05 Passaggio discendente Track: 35 Orbita: 16121</p> 		<p>Missione Envisat Tipo ASA WSM 1P Acquisizione: 09/05/10 Passaggio discendente Track: 183 Orbita: 42822</p> 	
<p>Missione Envisat Tipo ASA WSM 1P Acquisizione: 26/04/10 Passaggio discendente Track: 498 Orbita: 42636</p> 		<p>Missione Envisat Tipo ASA WSM 1P Acquisizione: 30/03/11 Passaggio discendente Track: 0 Orbita: 47479</p> 	

*Tabella 6.7. Immagini Envisat/ASAR elaborate nell'ambito di questo lavoro.*

Il *Multilayer Perceptron* (MLP) è un esempio di rete neurale artificiale che viene utilizzato per risolvere una serie di problemi, tra cui pattern recognition e interpolazioni. Questo tipo di rete neurale, ha mostrato la sua efficacia in altri lavori precedenti [53], ed è stato utilizzato nel presente lavoro, per classificare i punti rilevati (potenziali macchie di petrolio).

Come anticipato nei paragrafi precedenti, gli algoritmi per il riconoscimento automatico devono essere costruiti in funzione di alcuni parametri che dipendono dal tipo di sensore utilizzato (lo stesso oggetto non fornisce la stessa risposta variando il tipo di sensore). Per questo motivo, e vista la maggiore disponibilità di immagini Envisat/ASAR, questo studio prende in considerazione 30 scene estratte dalle 6 immagini Envisat (tabella 6.7) disponibili, per costruire il riconoscitore di *oil spill* basato sulle reti neurali.

Il pacchetto di scene disponibili, dallo stesso sensore, ha un certo numero di zone oscure nelle quali compaiono alcune fuoriuscite di petrolio verificate. Per ridurre il volume dei dati trattati, abbiamo estratto le zone oscure contenenti le regioni d'interesse.

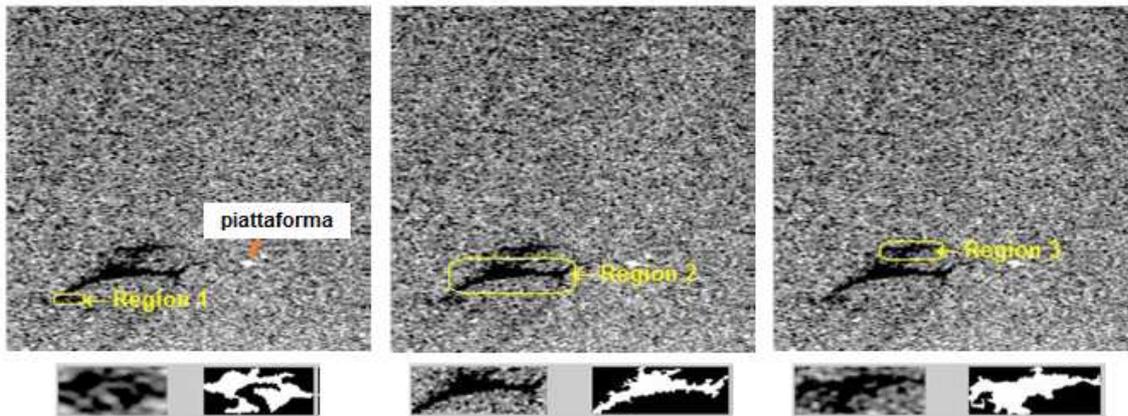
Tra le varie tipologie di falsi allarmi individuati in questo gruppo di scene, i più importanti sono quelli dovuti a ombre di venti e le zone con velocità superficiale dei venti inferiore alla soglia minima necessaria a permettere l'individuazione dei versamenti di petrolio sulle immagini SAR. Comunque, per futuri studi suggeriamo l'aggiunta al set di una più ampia varietà di falsi allarmi in modo da dare più forza al database.

Nella sua prima fase, il processo di classificazione genera (figura 6.24) le informazioni (caratteristiche descritte nei paragrafi precedenti) necessarie all'addestramento della rete neurale, consentendo successivamente di ottenere la classificazione desiderata.



**Figura 6.24.** Schema per la generazione delle Caratteristiche che addestrano la rete neurale.

Il processo così descritto ha generato un totale di 228 macchie da analizzare (potenziali versamenti di petrolio), delle quali 16 rappresentano effettivamente degli *oil spill* verificati mentre le rimanenti 212 macchie costituiscono dei falsi allarmi. Le matrici d'informazione così generate, sono state usate per la costruzione di un'opportuna *rete neurale MLP*, la quale rappresenta il sistema vero e proprio per il riconoscimento automatico dei versamenti di petrolio in mare. Il 70% dei casi considerati (160 campioni) hanno contribuito all'addestramento della rete, il 15% (34 campioni) per la validazione della stessa e il restante 15% (34 campioni) sono stati usati come casi test.



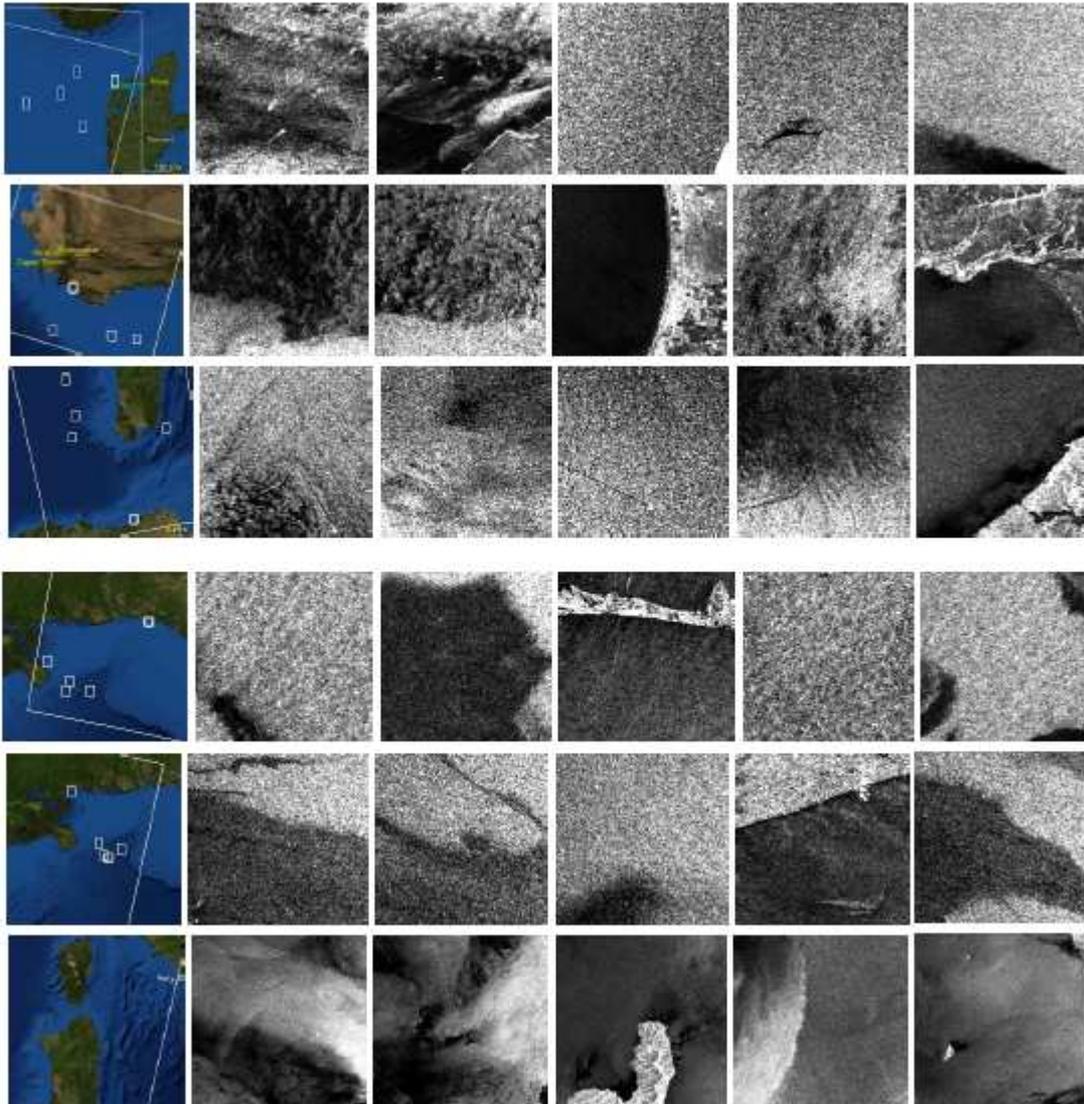
**Figura 6.25.** Esempio di zona d'interesse (con potenziale versamento di petrolio) estratta da una delle immagini *Envisat* studiate (avente 3 regioni di analisi).

Le caratteristiche usate per l'addestramento della rete derivano dalle 30 aree campioni estratte dalle immagini considerate (figura. 6.27). Il processo così descritto ha generato un totale di 228 macchie da analizzare (potenziali versamenti di petrolio), delle quali 16 rappresentano effettivamente degli *oil spill* verificati mentre le rimanenti 212 macchie costituiscono dei falsi allarmi. Le matrici d'informazioni



**Figura 6.26.** Passi nell'implementazione del sistema di riconoscimento automatico basato sulle reti neurali.

così generate, sono state usate per la costruzione di un'opportuna *rete neurale MLP*, la quale rappresenta il sistema vero e proprio per il riconoscimento automatico (figura 6.26) dei versamenti di petrolio in mare. Il 70% dei casi considerati (160 campioni) hanno contribuito all'addestramento [60].



**Figura 6.27.** 30 zone d'interesse usate per l'estrazione delle caratteristiche che vengono usate per addestrare la rete neurale MLP e quindi il sistema.

Come dimostrato nei paragrafi precedenti, la rete si aggiusterà (sistemando i propri parametri) in funzione dell'errore dovuto ai casi usati durante la fase di addestramento. I casi di validazione sono utilizzati per misurare il grado di generalizzazione della rete e fermare il processo di addestramento quando questo non migliora più il comportamento della stessa. Infine, i casi rimanenti utilizzati come test servono solo a fornire misure indipendenti delle performance della rete

durante e dopo l'addestramento. Il numero di neuroni nascosti, utilizzati per la progettazione della rete è stato fissato in 20 (figura 6.28).

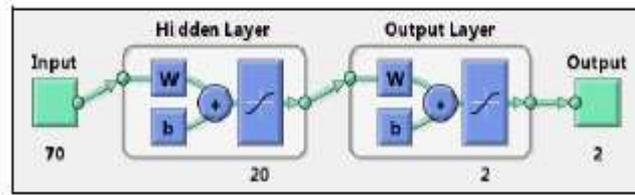


Figura 6.28. Schema generale della Rete Neurale utilizzata.

Le migliori prestazioni della rete sono state raggiunte dopo un numero d'iterazioni pari a 12. La figura 6.29 sintetizza i vari grafici che forniscono le prestazioni della rete MLP usata in questo modello di pattern recognition.

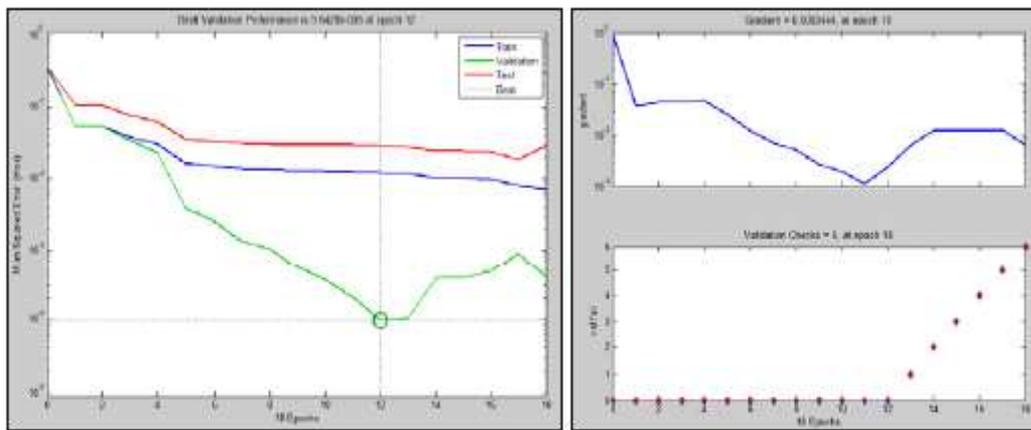
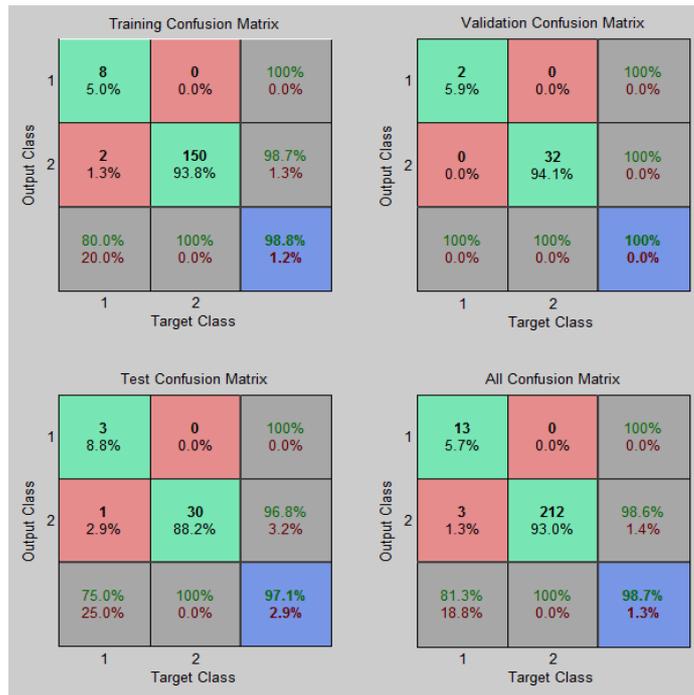


Figura 6.29. Performance della rete neurale MLP.

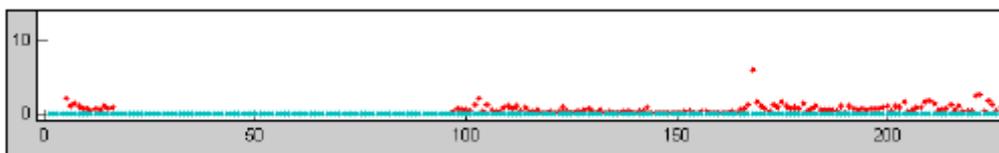
La figura 6.30 rappresenta la matrice di confusione del sistema, dove possono essere apprezzate le assegnazioni compiute dalla rete per ogni gruppo di dati. Lungo la diagonale troviamo il numero di casi *correttamente classificati*, mentre che fuori dalla diagonale troviamo i casi *non classificati correttamente*. La cella blu fornisce la percentuale di casi classificati correttamente (e non).

Nel gruppo dei 228 casi trattati, i 212 esempi di falsi allarmi sono stati classificati correttamente (neanche un errore), mentre per i 16 casi considerati di oil spill accertati la corretta classificazione è stata pari al 81,3% dei casi, classificando in modo sbagliato 3 macchie come falsi allarmi anziché versamenti di petrolio. La figura 31 fornisce l'errore quadratico medio del sistema durante la classificazione di 225 casi (i 3 casi sbagliati non sono riportati), e si può notare come tale errore si mantenga inferiore al 10% costantemente.



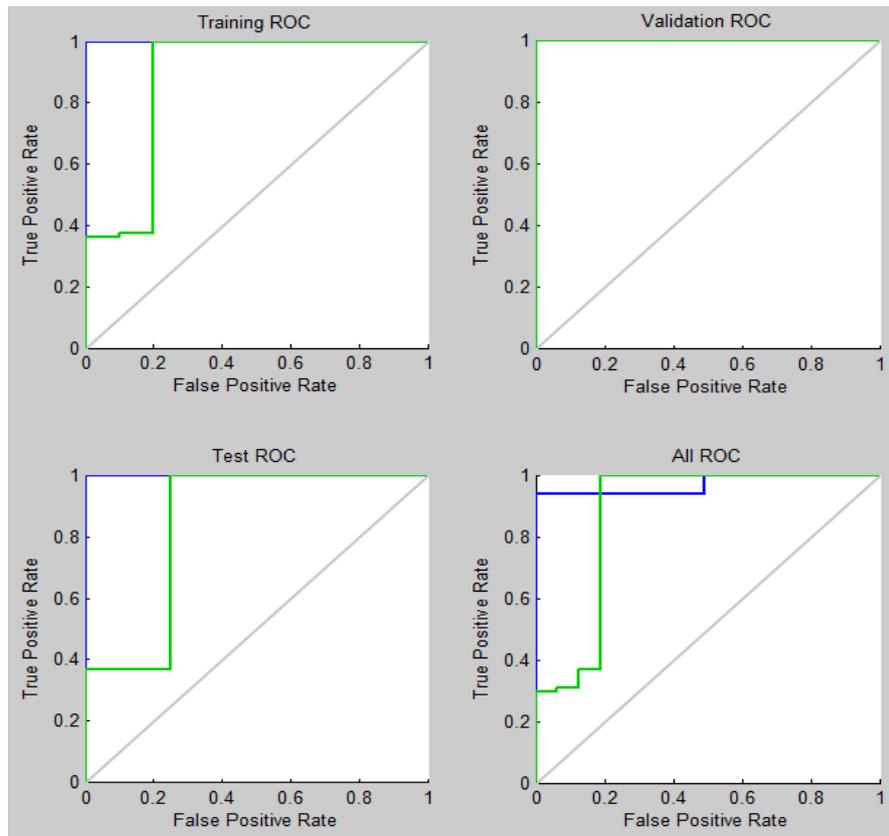
**Figura 6.30.** Matrice di confusione relativa alla classificazione della rete neurale.

Infine, può anche essere vista l'efficacia del classificatore neurale attraverso la curva ROC (figura 6.32). In questi grafici le linee colorate rappresentano le caratteristiche di funzionamento del sistema per le categorie *oil spill* e *falso allarme*. La curva esprime la percentuale di veri positivi (sensibilità) rispetto ai falsi positivi (specificità). La risposta ideale (ottima) dovrebbe mostrare i punti nell'angolo superiore sinistro, con 100% di sensibilità e 100% di specificità.



**Figura 6.31.** Errore quadratico medio (tranne i 3 casi di classificazione non corretta) generato dal sistema durante la classificazione di 225 casi.

Nonostante la percentuale totale di classificazioni corrette realizzata dal sistema raggiunga il livello del 98,7%, è auspicabile introdurre ulteriori casi validati (qualora siano disponibili) durante l'addestramento della rete, in modo da consentire una maggiore robustezza del riconoscitore qui implementato.



**Figura 6.32.** Curve ROC del processo di classificazione.

Infine, ricordiamo che il nostro scopo principale in questo lavoro è quello di trovare e implementare sistemi capaci di individuare automaticamente gli oggetti sulle immagini satellitari. E' evidente che anche in questo caso l'obiettivo è stato raggiunto, poiché con l'attuazione di quanto descritto nei paragrafi precedenti esiste la possibilità di creare una catena automatizzata per l'individuazione di macchie di petrolio sulle immagini SAR. Infatti, avendo a disposizione la rete neurale precedente, addestrata e consistente, è possibile, di fronte ad una potenziale macchia di petrolio, inserire le caratteristiche della macchia individuata per stabilire in modo automatico, l'appartenenza o meno alle classi "spill" o "falso allarme".

Questa classificazione, insieme ad una tabella di attributi geografici e generali, può consentire la generazione e proiezione automatica dell'oggetto analizzato su una mappa di riferimento (come riportato in figure 6.33 e 6.34 per esempio).

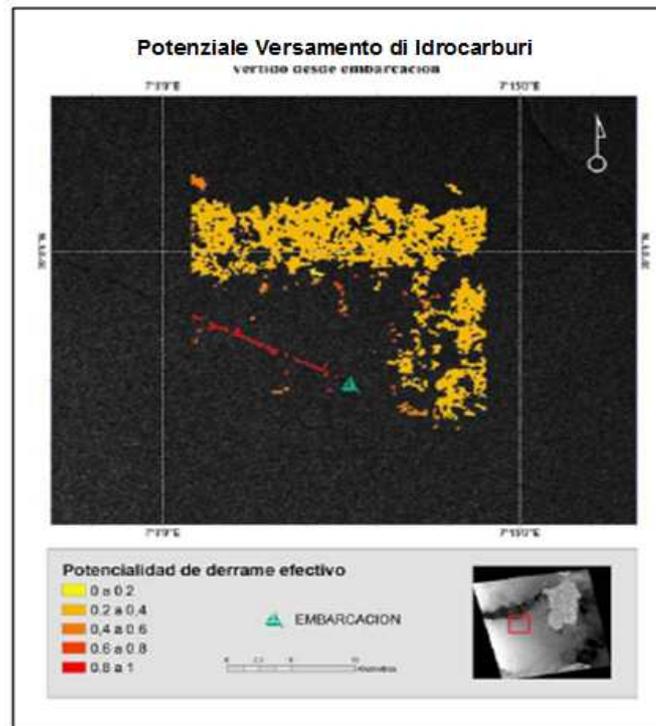


Figura 6.33. Mappa di output con la proiezione della classificazione ottenuta.

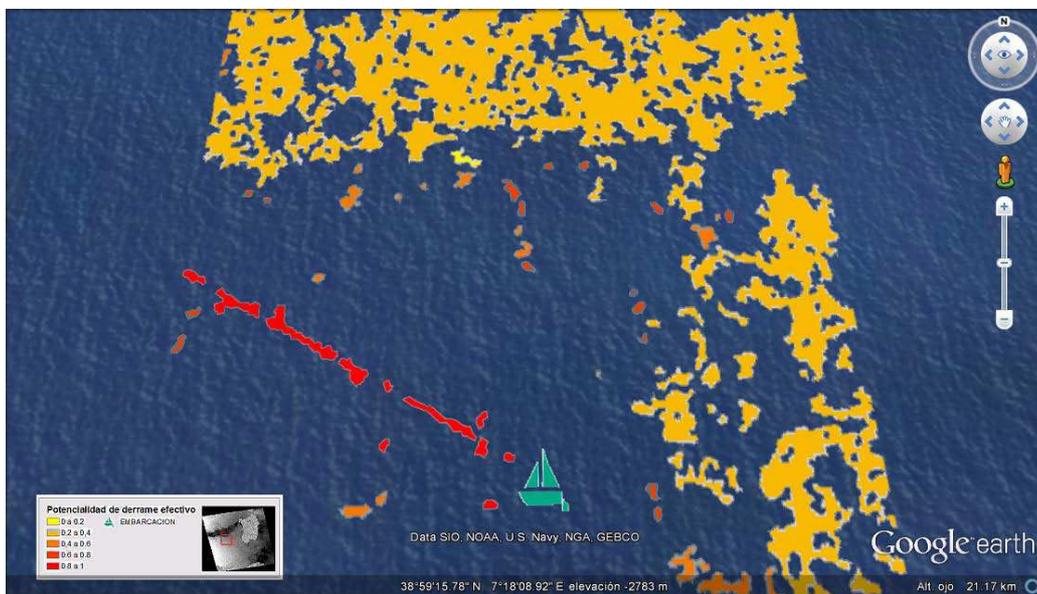


Figura 6.34. Mappa di output con la proiezione dell'oggetto classificato (in ambiente Google Earth).



## CONCLUSIONI E SVILUPPI FUTURI

Nella prefazione del libro sul *Pattern Recognition* del 1974, Vapnik e Chervonenkis scrissero: “...Per costruire la teoria della *Pattern Recognition*, prima di tutto è necessario formalizzare completamente il problema del riconoscimento automatico degli oggetti (...), e questo risultato sembra essere difficile da raggiungere.” Come abbiamo visto nei paragrafi precedenti le tecniche di *pattern recognition* sono molto dipendenti dal dominio di applicazione, ed anche le reti neurali che sembrano un approccio più generale in realtà dipendono dall’insieme di apprendimento (*training set*) e dai parametri degli algoritmi di *back-propagation* usati nell’addestramento (per esempio: il *learning rate*).

Essenzialmente, i differenti punti di vista sulla formulazione del problema della *pattern recognition* sono rappresentati dalla risposta ad una domanda: esistono principi generali adeguati per descrivere classi di pattern di diversa natura, oppure lo sviluppo di un linguaggio di descrizione dei pattern, è un problema che va affrontato dai singoli esperti per ogni dominio in ogni singolo caso?. Se la prima parte della domanda fosse vera, allora la scoperta di questi principi dovrebbe essere la principale direzione della ricerca nella *pattern recognition*.. Altrimenti la *pattern recognition* si ridurrebbe a minimizzare l’errore nell’assegnare il pattern alla classe sbagliata, o *risk minimization*, e questo implicherebbe la definizione di regioni di decisione specifiche per ogni classe di pattern e diventerebbe un problema di statistica applicata.

A distanza di quasi 40 anni le risposte alla domanda fondamentale non sono state trovate, e questo è il motivo per cui formulare un modello di *pattern recognition* è ancora un problema molto arduo. La maggior parte dei ricercatori (anche noi nel lavoro descritto in questa tesi di dottorato) ha comunque adottato il secondo punto di vista, e la teoria della *pattern recognition* è attualmente legata strettamente alla teoria della *risk minimization* con una classe di regioni di decisioni dipendenti dal dominio.

Alla fine degli anni cinquanta, i primi ricercatori sulla *pattern recognition* cercavano di formulare nuovi modelli e non di ridurre i problemi a schemi matematici o statistici noti. Infatti in quest’ottica quello che attualmente accade, e cioè di ridurre i problemi di *pattern recognition* a quelli di *risk minimization*, ha sollevato non poco disappunto nelle attese dei ricercatori. A quell’epoca però molti condividevano la sensazione che si potesse trovare qualcosa di completamente nuovo per modellare i problemi di *pattern recognition*.



Attualmente, dopo più di quarant'anni, è chiaramente naturale chiedersi se l'importanza della modellazione formale nella rappresentazione dei pattern è stata pienamente compresa. Evidentemente la risposta è no, poiché l'atteggiamento di riutilizzare schemi matematici noti continua a prevalere. Naturalmente, va detto che non c'è stato un grandissimo sforzo in questo senso per arrivare ad un formalismo unificante. Negli anni '70 ed '80, per esempio, i nuovi approcci sintattici e strutturali alla *pattern recognition* diedero un ruolo centrale alla rappresentazione non numerica dei pattern, e ci si aspettava che il futuro del campo fosse connesso all'integrazione degli approcci strutturali con quelli classici, basati sui *vector spaces* (spazi di vettori di *feature*). Ma questo non è ancora avvenuto, o meglio non sembra sufficientemente unificante.

Sembra in realtà di essere arrivati ad una svolta cruciale nel campo della *pattern recognition* in quest'ultimo periodo. Infatti, c'è la necessità di produrre inquadramenti formali che generalizzino l'approccio in modo da evidenziare la natura intrinsecamente "intelligente" dell'attività di *pattern recognition*. Basandoci su questo aspetto, vanno definiti i concetti chiavi su cui poggiare queste infrastrutture formali o framework della teoria. Come evidenziato in diversi workshop e conferenze sull'argomento, deve essere compreso che l'investimento di ingenti risorse nello sviluppo di nuove tecniche basate sul vecchio approccio (riduzione a statistica applicata) è inefficiente se comparato a quello necessario per costruire un impianto formale adeguato. Infatti, è da sottolineare il fatto che la scienza è orientata al futuro e quindi alla scoperta e costruzione degli scenari generali o "big picture" e non alla risoluzione indipendente di problemi, per quanto essi siano complicati.

Nonostante queste premesse, riconosciamo che questa tesi di dottorato ha privilegiato, per questioni di comodità, di necessità e di risorse comunque limitate, l'approccio statistico (o della *risk minimization*) nell'affrontare il problema della *pattern recognition*. Tuttavia, come evidenziato nella descrizione di ogni singolo capitolo di questo lavoro, lo sforzo nello sviluppo di numerosi algoritmi dedicati (sfruttando di volta in volta la tecnica più adatta tra quelle disponibili), ci ha condotto alla creazione di un sistema di algoritmi che può essere impiegato nei contesti più eterogenei possibili del *pattern recognition* (e non solo in quelli in cui abbiamo sviluppato i nostri algoritmi), in modo affidabile e automatico.

Ulteriori sforzi possono essere ancora compiuti nella direzione che porta alla realizzazione, per esempio, di un'interfaccia grafica (GUI) più *user friendly*, che consenta a chiunque (anche non particolarmente esperto), di usare queste tecniche già implementate per gli scopi ed le esigenze più svariate.



Se guardiamo alla necessità di risolvere il problema dell'estrazione delle informazioni di interesse (oggetti) dalle immagini satellitari, minimizzando l'intervento dell'operatore senza necessariamente cercare una soluzione buona per tutte le applicazioni, estremamente varie nel caso dell'osservazione da satellite, allora possiamo dire che questa tesi ha dimostrato la possibilità di ottenere l'obiettivo ricorrendo a varie tecniche caratterizzate, soprattutto, dalla loro innovatività rispetto alle tecniche storicamente applicate nell'elaborazione delle immagini telerilevate.

Di esse potrà giovare lo sviluppo di catene di elaborazione automatica in corso di realizzazione presso il dipartimento dove l'attività di dottorato è stata condotta. Le elaborazioni automatiche che si governeranno dei metodi qui descritti comprendono:

- l'individuazione di oggetti assegnati;
- l'individuazione di strade, aeroporti e l'aggiornamento delle mappe stradali soprattutto in aree remote;
- analisi di change detection (monitoraggio di strutture sensibili);
- co-registrazione di immagini per applicazioni di monitoraggio;
- applicazioni per la sicurezza (monitoraggio dei confini nazionali);
- etc.

Per completezza dobbiamo anche dire che gli sviluppi futuri si stanno orientando verso la *inductive class representation*, che descritta in poche parole, è un approccio sia generativo che induttivo alla *pattern recognition*. Il fatto di essere un approccio *generativo* implica che ha la capacità di “generare” oggetti dalle classi, basandosi sulla rappresentazione formale delle stesse. L'*induzione* nell'approccio, è intesa come la possibilità che la rappresentazione delle classi possa essere dedotta da un piccolo insieme di esempi (*training set*), e che tale rappresentazione sia stabile rispetto ai vari tipi di “rumore” ed “errori” presenti a seconda della rappresentazione scelta.

La ricerca nel settore propone attualmente un approccio chiamato ETS, *Evolving Transformation System*. L'ETS, diversamente dai classici modelli computazionali (per esempio, linguaggi e grammatiche formali), evidenzia la natura evolutiva delle classi di oggetti e quindi della loro rappresentazione. Queste strutture matematiche permettono di rappresentare dinamicamente, durante il processo di apprendimento, la struttura composta degli oggetti all'interno di un ambiente induttivo o evolutivo. In altre parole, il modello è basato su un nuovo tipo di struttura per la rappresentazione (induttiva) degli oggetti, chiamata *struct*, che è definita come un elemento di uno spazio generalizzato di misure, su cui la misura di similarità (tra oggetti) è rappresentata da un insieme (ottimo) finito di trasformazioni strutturali pesate (per esempio, come le



operazioni di inserimento di oggetti sensibili al contesto), costruite durante il processo di apprendimento.



## REFERENZE BIBLIOGRAFICHE

- [1] Jain A.K., Duin R.P.W. & Mao J. (2000), “*Statistical Pattern Recognition: A review*”. IEEE transaction on pattern analysis and machine intelligence, 22.
- [2] Watanabe S. (1985), “*Pattern Recognition: Human and Mechanical*”. New York, Wiley.
- [3] Pavlidis T. (1980), “*Structural Pattern Recognition*”. Springer-Verlag, Berlino.
- [4] Esposito M., Mazzariello C., Oliviero F., Romano S.P. & Sansone C. (2005), “*Evaluating Pattern Recognition Techniques in Intrusion Detection Systems*”. Proceeding of 5th Workshop on Pattern Recognition on Information Systems (PRIS 05), 144-153.
- [5] Ullman J.R. (1973), “*Pattern Recognition Techniques*“. Butterworths.
- [6] Fukunaga K. (1990), “*Introduction Statistical Pattern Recognition*”. Academic Press, Second edition.
- [7] Schalkoff R.J. (1992), “*Pattern Recognition: Statistical, Structural and Neural Approaches*”. Wiley, New York.
- [8] Devroye L., Györfi L. & Lugosi G. (1996), “*A Probabilistic Theory of Pattern Recognition*”. New York, Springer-Verlag.
- [9] Oommen B.J. & Kashyap R.L. (1997), “*A Formal Theory for Optimal and Information Theoretic Syntactic Pattern Recognition*”. Ottawa, Canada.
- [10] Fu K.S. (1982), “*Syntactic Pattern Recognition and Applications*”. Prentice Hall.
- [11] Fu K.S. (1974), “*Syntactic Methods in Pattern Recognition*”. Academic Press.
- [12] Cordella L.P., Foggia P., Sansone C. & Vento M. (2002), “*Learning Structural Shape Description from Examples*”. Pattern Recognition Letters, 23, 1427-1437.
- [13] Pavlidis T., Sakoda W.J. & Shi H. (1995), “*Matching Graph Embeddings for Shape Analysis*”. Third International Conference on Document Analysis and Recognition (ICDAR'95), 2, 729.



- [14] Foggia P. (1999), “*Classificazione e Prototipazione di Descrizioni Strutturali Rappresentate Attraverso Grafi Relazionali con Attributi*”. Tesi di dottorato presso l'Università Federico II di Napoli.
- [15] Shao P.K., Soltani S. & Wong A.K.C. (1988), “*A Survey of Thresholding Techniques*”. Computer vision, graphics, image processing, 41, 233-260.
- [16] Arcelli C., & di Baja G.S (1985), “*A Width Independent Fast Thinning Algorithm*”. IEEE Transaction, Pattern analysis and machine intelligence, 7, 463-474.
- [17] Arcelli C., Cordella L.P. & Levialdi S. (1981), “*From Local Maxima to Connected Skeletons*”. IEEE Transaction, Pattern analysis and machine intelligence, 2, 134-143.
- [18] Duda R.O., Hart P.E., Stork D.G. (2001). “*Pattern Classification*”. John Wiley & Sons, 2nd edition.
- [19] Ramer U. (1972), “*An Interactive Procedure for the Polygonal Approximation of Planar Curves*”. Computer graphics and image processing, 1, 244-256.
- [20] Kong T., Rosenfield A. (1989), “*Digital Topology: Introduction and Survey*”. Computer Vision, Graphics and Image Processing, 48, 357-393.
- [21] Slansky J., Chazin R.L., Hansen B.J. (1972), “*Minimum-perimeter Polygons of Digitalized Silhouettes*”. IEEE transaction on computers, 21, 260-268.
- [22] Pavlidis T., Horowitz S.L. (1974), “*Segmentation of Planar Curves*”. IEEE transaction on computers, 23, 860-870.
- [23] Wu W.Y., Wang M.J.J. (1993), “*Detecting the Dominant Point by the Curvature-based Polygonal Approximation*”. CVGIP: Graphical Models and Image Processing, 55, 79-88.
- [24] Grenander U. (1993), “*General Pattern Theory*”. Oxford University Press.
- [25] Ngu A.H., Sheng Q., Huynh D., Lei R., (2001), “*Combining Multi-Visual Features for Efficient Indexing in a Large Image Database*”. The VLDB Journal, Vol. 9, n. 4, p. 279-293.
- [26] Jain A.K., “*Fundamentals of Digital Image Processing*”. Prentice-



Hall Int.

[27] Ullman J.D. (1973), "*Pattern Recognition Techniques*". Butterworths, London.

[28] Teh C. and Chin R.T. (1988), "*On Image Analysis by the Methods of Moments*". IEEE Trans. on Pattern Analysis and machine Intellig., 10:496-512.

[29] Hu M.K. (1961), "*Pattern Recognition by Moment Invariants*". In Proc. IRE, number 49.

[30] Durani S.A., Breeding K.J., McGhee R.B. (1977), "*Aircraft Identification by Moment Invariants*". IEEE Trans. Computer, C26.

[31] Wong R.Y., Hall E.L. (1978), "*Scene Matching with Invariant Moment*". Computer Graphics and Image Processing, 8:16-24.

[32] Reiss T.H. (1993), "*Recognition Planar Objects Using Invariant Image Features*". Springer-Verlag.

[33] Hu M.K. (1977), "*Visual Pattern Recognition by Moment Invariants*". IEEE Comp. Society.

[34] Teague M. (1980), "*Image Analysis via the General Theory of Moments*". J. Opt. Soc. Am.

[35] Khotanzad A., Hong Y.H. (1990), "*Rotation Invariant Image Recognition Using Features Selected via a Systematic Method*". Pattern Recognition, 23:1089-1101.

[36] Zernike F. (1934), "*Phisica I*".

[37] Courant R., Hilbert D. (1953), "*Methods of Mathematical Phisics*". Interscience, New York.

[38] Pavlidis T. (1977), "*Structural Pattern Recognition*". New York, Springer-Verlag.

[39] R. Gonzalez and R. Woods (2002), "*Digital Image Processing*". 2nd ed., Prentice Hall, Upper Saddle River, NJ.

[40] E. R. Dougherty and L. A. Lotufo (2003), "*Hands-on Morphological Image Processing*". SPIE - International Society for Optical Engin.



- [41] R. Gonzalez, R. Woods, and S. Eddins (2004), “*Digital Image Processing using Matlab*”. Pearson-Prentice Hall, Upper Saddle River, NJ.
- [42] S. Giada, T. De Groeve, D. Ehrlich, and P. Soille (2002), “*Information Extraction from Very High Resolution Satellite Imagery over Lukole Refugee Camp, Tanzania*”. *Int. J. Remote Sensing*, vol. 24, N° 22, pp. 4251-4266.
- [43] Ehrlich D., Lang S., Laneve G., Mubareka S., Schneiderbauer S., Tiede D. (2005), “*Can Earth Observation Help to Improve the Need for Information on Population ?*”. Jasani B., Pesaresi M. & Schneiderbauer S. (eds.) *GMOSS – Global Monitoring for Security and Stability – volume 1*. Springer.
- [44] UNHCR, 2005. “*Statistical Overview of Populations of Refugees, Asylum-Seekers, Internally Displaced Persons, Stateless Persons, and Other Persons of Concern to UNHCR*”. Available online under [www.unhcr.org](http://www.unhcr.org) (accessed 08/2006).
- [45] “*Image Processing Toolbox, User Guide*”, Version 3,4 (2001-2003). The MathWorks, Inc., Natick, MA.
- [46] J. Holly Reed (2002), “*Demographic Assessment Techniques in Complex Humanitarian Emergencies: Summary of a Workshop*,” Roundtable on the Demography of Forced Migration, National Research Council, The National Academies Press.
- [47] “*eCognition User Guide*” (2004), Definiens Imaging.
- [48] A. Solberg, P. Clayton, M. Indregard (2005), “*Delivery D2–Report on Benchmarking Oil Spill Recognition Approaches and Best Practices*”. Ver.2.1. 18.05. Kongsberg Satellite Services.
- [49] C. Brekke, A. Solberg (2005), “*Oil Spill Detection by Satellite Remote Sensing*”. *Remote Sensing of Environment* 95, 1 –13.
- [50] Fiscella B., Giancaspro A., Nirchio F., Pavese P. and Trivero P., (2000), “*Oil Spill Detection Using Marine SAR Images*”, *International Journal of Remote Sensing*, 21: 18, 3561 – 3566.
- [51] Santilli G., Laneve G. (2010), “*Oil spill Detection on Radar Images by Using Mathematical Morphology*”. 61<sup>st</sup> International Astronautical Congress (IAC), Prague, CZ.



- [52] Laneve G., Santilli G., Marzialetti P., Fusilli L. (2011), "Monitoring Oil Spill in Lake Maracaibo: a Challenge for EO Satellite Systems".
- [53] Del Frate F., Petrocchi A., Lichtenegger J. y Calabresi G. (2000), "Neural Networks for Oil Spill Detection Using ERS-SAR Data". IEEE Transactions on Geoscience and Remote Sensing, Vol.38, N.5.
- [54] González-Vilas L., Torres J.M., Martínez-Iglesias G., Mosquera A. y Corchado J.M. (2007). "Utilización de Datos de ENVISAT para la Detección de Vertidos de Hidrocarburos". Revista de Teledetección, Número Especial: 55-59.
- [55] Montali A., Giacinto G. Migliaccio, M. Gambardella A. (2006), "Supervised Pattern Classification Techniques for Oil Spill Classification in SAR Images: Preliminary Results". SEASAR2006 Workshop, ESA-ESRIN, Frascati, Italy, January 23-26.
- [56] Montali A., Giacinto G. Migliaccio, M. Gambardella A. (2006), "Supervised Pattern Classification Techniques for Oil Spill Classification in SAR Images: Preliminary Results". SEASAR2006 Workshop, ESA-ESRIN, Frascati, Italy, January 23-26.
- [57] Karathanassi V., Topouzelis K., Pavlakis P., Rokos D. (2006), "An Object-oriented Methodology to Detect Oil Spills". International Journal of Remote Sensing, 27, pp. 5235–5251.
- [58] Topouzelis K., Karathanassi V., Pavlakis P., Dokos D. (2007), "Detection and Discrimination Between Oil Spills and Look-Alike Phenomena Through Neural Networks". ISPRS Journal of Photogrammetry & Remote Sensing 62, 264–270.
- [59] Topouzelis K., Stathakis D. and Karathanassi V. (2009), "Investigation of Genetic Algorithms Contribution to Feature Selection for Oil Spill Detection". International Journal of Remote Sensing Vol. 30, No. 3, 611–625.
- [60] Marzialetti. P. (2012), "Monitoreo de Derrames de Hidrocarburos en Cuerpos de Agua Mediante Técnicas de Sensado Remoto". Tesis de Maestría en Aplicaciones Espaciales de Alerta y Respuesta Temprana a Emergencias, Comisión Nacional de Actividades Espaciales (CoNAE). Argentina.