

84-19
2002

Elaborazione di testi antichi a stampa: un sistema integrato per l'analisi e il riconoscimento di caratteri

La fusione di tecniche computazionalmente veloci e di approcci analiticamente complessi, quali i modelli a campi di Markov per immagini, con sistemi di classificazione basati su reti neurali, permette di costruire una procedura per la segmentazione adattiva di linee di testo e per il successivo riconoscimento dei caratteri

Stefano Vezzosi, Luigi Bedini, Anna Tonazzini

*Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"
Area della Ricerca CNR di Pisa
Via G. Moruzzi, 1
56124 Pisa*



Abstract

I documenti a stampa antichi presentano spesso degradazioni tipiche che determinano una difficile comprensione del testo a causa di caratteri mal definiti o addirittura completamente rovinati. L'inchiostro, col passare degli anni, è sottoposto ad un processo di diffusione locale non uniforme, causa di caratteri spezzati o della presenza di macchie che rendono complessa una suddivisione dei singoli elementi del testo finalizzata all'uso di riconoscitori automatici (Optical Character Recognition – OCR). Tutti questi inconvenienti possono essere, almeno in parte, corretti con tecniche elettroniche di elaborazione e restauro delle immagini. In questo lavoro verrà proposto l'uso di tecniche di restauro basate su wavelet, deconvoluzione cieca e modelli a campi di Markov per immagini, integrate con un classificatore intelligente, basato su reti neurali, per il riconoscimento dei caratteri di testo. In particolare, verrà proposto un sistema che, a partire da una singola immagine del documento, acquisita in luce visibile, riduce il rumore presente nell'immagine con una tecnica wavelet, allinea le righe di testo, le identifica e le estrae dalla pagina, quindi le segmenta inizialmente con un semplice e veloce approccio di binarizzazione a soglia, e analizza i vari blob individuati usando una rete neurale opportunamente addestrata. Qualora la probabilità di riconoscimento di un blob sia bassa, il processo di segmentazione viene affinato mediante un processo di deconvoluzione cieca e binarizzazione congiunte, e il risultato viene riproposto alla rete neurale. Questo permette di ottenere una segmentazione dei singoli caratteri molto precisa e un riconoscimento con una alta affidabilità, rendendo possibile e relativamente veloce l'elaborazione e l'analisi dei testi, e la loro archiviazione in formato elettronico.

1. Introduzione

Pacchetti software per l'Optical Character Recognition (OCR), attualmente disponibili per effettuare il riconoscimento automatico di caratteri a stampa, non possono essere applicati con successo a testi antichi, dove l'invecchiamento della carta, la diffusione dell'inchiostro e altri processi hanno degradato fortemente la qualità dei documenti. La causa principale è che la preliminare fase di segmentazione dei caratteri è resa particolarmente difficile dalla riduzione non uniforme del contrasto nell'immagine, dalla spaziatura non uniforme dei caratteri, dalla presenza di caratteri spezzati, attaccati e/o fusi assieme, e dalla presenza di un

forte rumore nello sfondo. Anche sistemi intelligenti per il riconoscimento di caratteri (ICR), basati sull'uso di reti neurali [14,25], sebbene più robusti nel riconoscere caratteri degradati, richiedono una loro corretta segmentazione. Recentemente sono stati proposti approcci globali al riconoscimento, basati su modelli del tipo Hidden Markov Models (HMMs) [24], in grado di riconoscere una sequenza di caratteri senza la necessità di una loro preliminare segmentazione. Inoltre, questi approcci permettono di incorporare nel processo di riconoscimento conoscenze contestuali, espresse in forma probabilistica. Nonostante i risultati promettenti riportati in letteratura, quando vengono applicati a documenti antichi degradati questi metodi non sembrano fornire prestazioni soddisfacenti, in quanto producono ancora parecchi errori, specialmente quando i caratteri si toccano o sono fusi assieme.

Quindi, una preelaborazione dei documenti mediante metodi di enhancement/restauro di immagini diventa necessaria. Nella maggior parte delle situazioni, è ragionevole modellare il processo di degradazione come un filtro di blur spazio-variante più rumore additivo. Tuttavia, poiché il filtro di blur è non noto, devono essere utilizzate tecniche cieche di deblurring e denoising, e opportuni modelli per l'immagine devono essere incorporati nel problema al fine di regolarizzarlo. Modelli a campi di Markov (MRF) si sono rivelati efficienti e flessibili sia nel restauro cieco che nella segmentazione di immagini [18,19]. In particolare, il loro uso consente di eseguire il restauro cieco e la segmentazione dei caratteri congiuntamente. In lavori precedenti [20,21], è stato mostrato che il restauro cieco combinato con una binarizzazione (segmentazione) basata su modelli MRF dà risultati soddisfacenti per la segmentazione di caratteri in documenti antichi degradati. Il problema è formulato come la minimizzazione di una funzione costo, o funzione energia, che esprime la consistenza della soluzione con i dati, e sia vincoli radiometrici sui livelli di grigio dell'immagine, che vincoli geometrici sui bordi dei caratteri. La strategia adottata per il calcolo della soluzione alterna passi di segmentazione con passi di stima dell'operatore di degradazione. I documenti elaborati sono adatti ad essere analizzati da sistemi ICR, basati su reti neurali. Tuttavia, la complessità computazionale di queste sofisticate tecniche di elaborazione di immagini è molto alta, e ne impedisce il loro uso su documenti molto estesi. Per superare questa difficoltà, il passo di restauro/segmentazione e il passo di riconoscimento potrebbero essere integrati fra loro, in modo da usare le tecniche cieche per affinare la segmentazione solo in quelle zone del documento dove il processo di riconoscimento fallisce.

In questo lavoro verrà definito un sistema integrato che inizialmente applica un classificatore basato su una rete neurale a una pagina del documento, preelaborata mediante semplici e veloci tecniche di enhancement di immagini e di segmentazione dei caratteri, e quindi, sulla base della probabilità di riconoscimento, rielabora automaticamente quelle piccole porzioni dell'immagine dove caratteri spezzati o congiunti non sono stati riconosciuti correttamente. Quando viene applicato a piccole aree, il restauro cieco accoppiato con la binarizzazione a campi di Markov è relativamente poco costoso e molto efficace nel produrre caratteri ben segmentati e quindi riconoscibili correttamente.

2. Overview del sistema

Il sistema integrato per l'elaborazione, l'analisi e il riconoscimento di caratteri in testi a stampa antichi proposto in questo lavoro è formato da vari blocchi di elaborazione eseguiti in sequenza sul documento digitale, come mostrato in Figura 1. Il primo è un blocco di filtraggio per la rimozione del rumore, che analizza il documento originale usando tecniche di decomposizione basate sulla trasformata discreta wavelet. La funzione base utilizzata per la decomposizione è la funzione *symlet*, con cui vengono calcolati i filtri passa basso e passa alto

per ottenere i coefficienti relativi ad un filtraggio del quarto ordine. I coefficienti associati alle basse frequenze dell'immagine vengono lasciati inalterati, mentre quelli associati alle alte frequenze vengono sottoposti ad una operazione di soglia in modo da eliminare quelli direttamente legati al processo di rumore. La ricostruzione viene poi calcolata usando filtri speculari ai precedenti, e rappresenta un filtraggio passa basso di tipo adattivo [1,2,3,4,5].

Successivamente, in un secondo blocco elaborativo, le righe del testo vengono identificate ed estratte usando un approccio nel dominio di Fourier, che calcola la trasformata della proiezione dell'immagine lungo la direzione verticale ed identifica la frequenza associata al segnale periodico che rappresenta la ripetizione delle righe di testo.

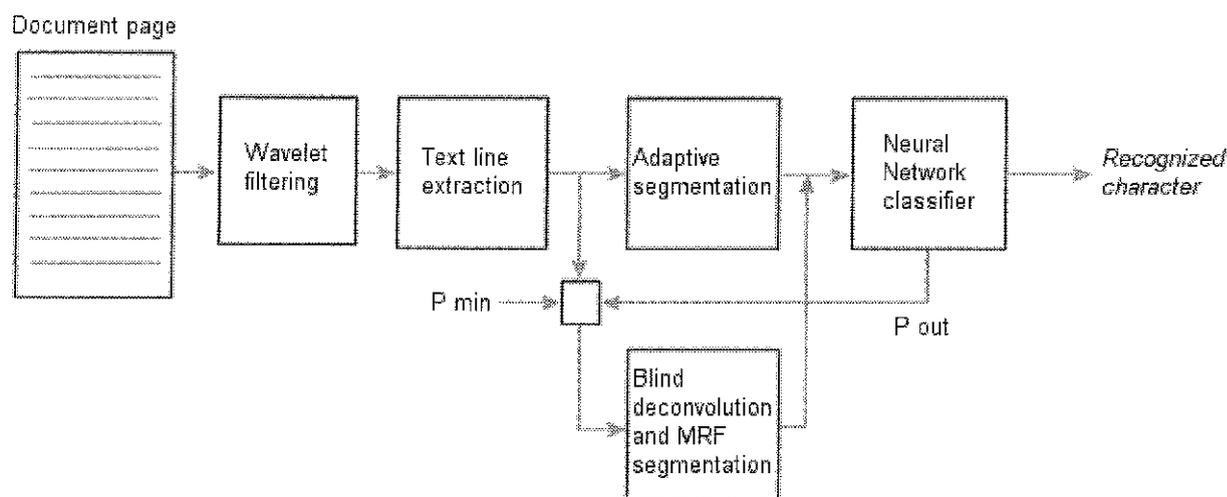


Fig. 1 –Schema a blocchi del sistema integrato

Il terzo blocco procede ad una binarizzazione della riga, basata essenzialmente su una semplice e veloce operazione di sogliatura locale adattiva, allo scopo di segmentare la riga in blob corrispondenti ai singoli caratteri che la costituiscono.

Nel quarto blocco, i vari blob identificati vengono elaborati individualmente da un'opportuna rete neurale, la quale produce in uscita un risultato che identifica il carattere riconosciuto e restituisce un valore di probabilità legato all'affidabilità del riconoscimento effettuato. La rete usata è una feed-forward addestrata con un algoritmo di back-propagation [7,8,9], usando un training set costruito appositamente per lo scopo con caratteri selezionati da una pagina di testo del documento trattato, a loro volta binarizzati e segmentati con la procedura descritta sopra.

La parte più critica di tutta la procedura è costituita dal passo di segmentazione, in quanto i più recenti modelli di reti neurali proposti in letteratura consentono il riconoscimento efficiente anche di caratteri degradati, purché questi risultino correttamente segmentati. L'efficienza della segmentazione è invece fortemente condizionata dalla qualità del testo stesso. La tecnica di binarizzazione e sogliatura locale utilizzata nella procedura costituisce un primo approccio, veloce ma non particolarmente accurato, al problema della segmentazione dei caratteri in testi degradati. Occorre però affermare che questa tecnica si è rivelata sufficientemente accurata nella maggior parte dei casi, quando cioè il testo non presenta gravi danni, rendendo così efficiente l'algoritmo di riconoscimento, e di immediata attuazione l'intera procedura proposta. In presenza di testo fortemente corrotto, è invece facile che la probabilità associata al singolo riconoscimento scenda sotto un valore prefissato, principalmente a causa di una cattiva segmentazione. In questi casi, è necessario affinare la

segmentazione, utilizzando tecniche più complesse, realizzate da un quinto blocco elaborativo, che interviene retroattivamente nella porzione di riga di testo i cui blob hanno dato luogo ad un riconoscimento errato. Per esempio, qualora si fosse in presenza di due blob contigui riconosciuti entrambi con probabilità bassa e poco estesi orizzontalmente, l'algoritmo procede al loro accorpamento, e il nuovo blob viene di nuovo dato in input alla rete neurale. La nuova probabilità di riconoscimento viene quindi confrontata con quelle precedenti. Se tale elaborazione ha portato ad un risultato più affidabile, allora viene scelta la nuova segmentazione come la più appropriata al caso in esame. Nell'ipotesi in cui la probabilità di riconoscimento sia bassa ed il blob in esame sia particolarmente esteso orizzontalmente, si procede invece ad una nuova segmentazione locale della zona individuata, usando tecniche basate su algoritmi di deconvoluzione cieca [16,17] e modelli a campi di Markov (MRF) [18,19]. Il risultato dell'elaborazione viene di nuovo passato in input alla rete neurale, e se il riconoscimento è valido viene aggiornato il vettore di segmentazione.

Il processo di binarizzazione che utilizza tecniche di deconvoluzione cieca e modelli a campi di Markov è un processo iterativo che alterna passi di stima dell'operatore di degradazione, modellato qualitativamente come un processo di blur più rumore additivo, a passi di stima dell'immagine binaria che meglio aderisce ai dati e che soddisfa informazioni a priori eventualmente disponibili. Questo si realizza mediante la minimizzazione di un'opportuna funzione energia, composta da due termini. Il primo misura la consistenza fra i dati originali e l'immagine binarizzata corrente, convoluta con la maschera di blur corrente. Il secondo, legato al modello MRF scelto, è dato dalla somma di un insieme di funzioni potenziale che esprimono vincoli geometrici locali sulle configurazioni dei pixel dell'immagine binarizzata desiderata [20,21].

L'uso di tale metodo permette di affinare il risultato della segmentazione e quindi l'efficienza del riconoscimento, a scapito però del tempo di calcolo. Tuttavia, il ricorso a queste tecniche complesse è limitato a piccole aree del documento particolarmente danneggiate, e che, dagli esperimenti effettuati, sono risultate relativamente poco frequenti. L'anello di reazione implementato viene inoltre pilotato automaticamente da semplici parametri, quali la probabilità minima accettata per il riconoscimento dei singoli caratteri, e la massima estensione orizzontale, in pixel, associata ad un singolo blob, rendendo il sistema ancora di semplice e veloce utilizzo.

Nel seguito verranno descritti in dettaglio i vari blocchi di elaborazione che compongono il sistema proposto, e verranno illustrati alcuni dei risultati sperimentali ottenuti.

3. Denoising mediante la trasformata discreta wavelet

Nella maggior parte dei casi le pagine del documento risultano danneggiate o corrotte da macchie e diffusioni dell'inchiostro. Considerato che un buon risultato per la segmentazione dipende essenzialmente dalla qualità dell'immagine da analizzare, è di fondamentale importanza migliorare tale qualità, in modo da recuperare la maggior parte dell'informazione contenuta nel testo originale. Ipotizzando ragionevolmente che la degradazione sia qualitativamente modellabile come una sfocatura (convoluzione dell'immagine ideale con una maschera di blur) e sovrapposizione di rumore additivo, il tipo di preelaborazione necessaria fa uso di tecniche di filtraggio per la rimozione del rumore e di tecniche di deconvoluzione per la rimozione della sfocatura. Nel caso di interesse, si ha l'ulteriore complicazione che il processo di degradazione è incognito, cioè la maschera di blur associata al processo di sfocatura non è nota.

La tecnica principe per il restauro di immagini sfocate e rumorose, in assenza di informazioni sulla degradazione, è la deconvoluzione cieca bidimensionale, che permette di ricostruire sia l'immagine originale che la maschera di blur che ha degradato il testo, come risultato di un processo iterativo numerico con infiniti passi [11,12]. Il principale inconveniente di questa tecnica è l'estrema complessità del calcolo, che rende impensabile il suo utilizzo su un intero documento. Occorre inoltre notare che la stima della degradazione dipende anche dalla zona di testo analizzata: in generale è infatti possibile che la maschera di blur sia invariante solo in piccole porzioni di testo, mentre è soggetta a grandi variazioni all'interno di una intera pagina. Per questo motivo, una deconvoluzione globale sull'intero documento non riesce sempre a produrre i risultati sperati, e alcuni caratteri connessi fra loro possono essere separati con successo soltanto stimando localmente la maschera di blur. È necessario allora pensare a una tecnica di deconvoluzione locale, che sia legata essenzialmente alla porzione di testo analizzata. In pratica, occorre suddividere la pagina in tante sottoimmagini ed elaborarle indipendentemente le une dalle altre. Questo comporta tuttavia delle distorsioni in corrispondenza dei bordi delle singole parti, che potrebbero creare difficoltà nel successivo processo di binarizzazione/segmentazione. Per questi motivi, un approccio combinato deconvoluzione cieca-campi di Markov verrà utilizzato solamente per piccole aree dell'immagine e congiuntamente al processo di binarizzazione.

Una preliminare rimozione del rumore risulta tuttavia necessaria. A questo scopo, un approccio ben più veloce, che permette di analizzare ed elaborare l'intero documento in una singola passata, è quello del filtraggio basato sulla decomposizione mediante trasformata wavelet discreta (DWT). Questo strumento permette un'analisi in frequenza a multirisoluzione e un efficiente filtraggio del rumore che si manifesta alle frequenze più alte.

L'idea che è alla base della trasformata wavelet è la decomposizione multipla di un segnale usando un banco di filtri passa alto e passa basso, collegati in cascata tra loro, intervallati da stadi di decimazione [1,2,3,5]. Questo processo permette di ottenere un insieme di valori, detti coefficienti della decomposizione wavelet, che descrivono il segnale analizzato rispetto alla funzione madre scelta. La ricostruzione del segnale a partire dai coefficienti, eventualmente modificati in maniera opportuna a seconda degli scopi, utilizza filtri speculari ai precedenti. Il procedimento di analisi e sintesi è illustrato in Figura 2.

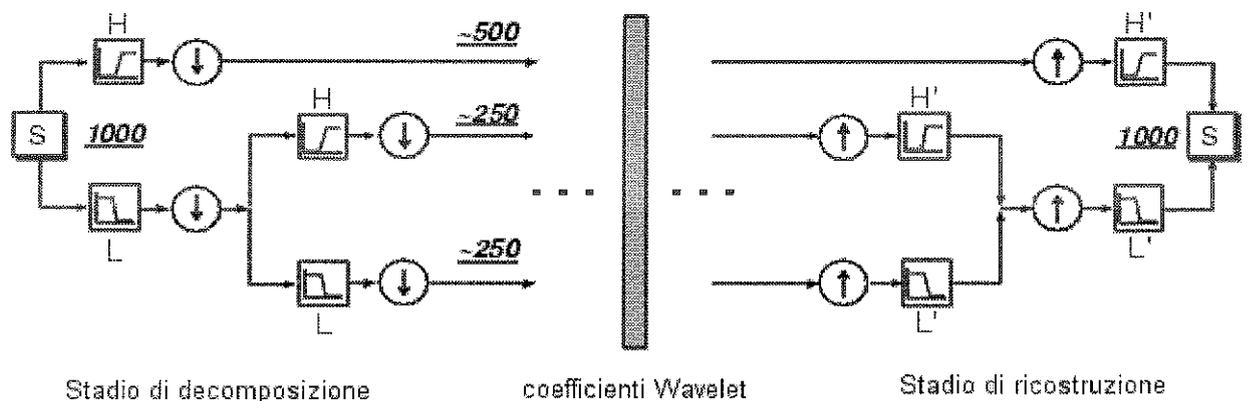


Fig. 2 – Schema generale del processo di analisi e di sintesi di un segnale usando la tecnica wavelet. Il segnale di esempio S è stato campionato inizialmente su 1000 valori.

In questa rappresentazione appare evidente la struttura a cascata del banco di filtraggio che porta alla derivazione dei coefficienti wavelet, necessari per la consecutiva operazione di sintesi del segnale. E' evidente anche l'uso dei blocchi sottocampionatori che permettono di sviluppare l'analisi a risoluzione variabile.

Nel caso di interesse in questo lavoro, trascurando per il momento l'effetto del blur, l'immagine degradata g , di dimensione $M \times N$, può essere espressa come:

$$g = f + n \quad (3.1)$$

dove f è l'immagine ideale ed n è un processo di rumore assunto bianco Gaussiano, di deviazione standard σ . La decomposizione wavelet discreta di g risulta essere:

$$C(s, p) = \sum_{i=1}^M \sum_{j=1}^N g_{i,j} \cdot \Psi_{i,j}(s, p) \quad (3.2)$$

dove s and p sono i parametri discreti di scala e posizione, rispettivamente, C sono i coefficienti della decomposizione, e Ψ è la funzione wavelet madre. Poiché il rumore si addensa alle frequenze più alte, è ragionevole aspettarsi che i coefficienti dello sviluppo estratti dalle uscite dei filtri passa alto siano pesantemente influenzati dal rumore presente nell'immagine: se si esegue una operazione di soglia sui coefficienti e si ricostruisce il segnale usando il nuovo insieme di coefficienti, si ottiene una ricostruzione dei dati in cui il processo di rumore è stato opportunamente filtrato. Un aspetto critico dell'intera procedura, è la scelta della funzione madre per calcolare la decomposizione, e il tipo di soglia per ottenere il filtraggio. Per quanto riguarda la scelta della funzione wavelet, non esistono criteri analitici da seguire; tuttavia, poiché la risposta in frequenza dei singoli filtri dipende da tale funzione, è evidente che funzioni diverse mettono in risalto features diverse dell'immagine. Per la presente analisi è stata scelta come funzione madre una wavelet di ordine 4, appartenente alla famiglia *symlet*, illustrata in Figura 3.

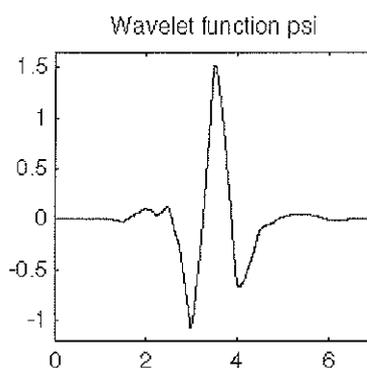


Fig. 3 – Andamento della funzione wavelet *symlet* di ordine 4, scelta per il filtraggio del documento.

Una volta scelta la funzione per l'analisi, è immediata la derivazione delle funzioni di trasferimento associate ai singoli filtri del banco di decomposizione. Per quanto riguarda le operazioni di soglia sui coefficienti, è stato usato un approccio di *soft thresholding* [4], in modo da considerare solo i coefficienti maggiori della densità spettrale di potenza associata al rumore, stimata con tecniche numeriche, e tralasciare gli altri. Questo ha lo scopo di

preservare i dettagli dell'immagine associati ai bordi dei caratteri, anch'essi rappresentati da coefficienti ad alta frequenza. I coefficienti ottenuti dopo l'operazione di sogliaatura sono dati da:

$$\hat{C}(s, p) = \begin{cases} \text{sign}(C(s, p)) \cdot (|C(s, p)| - \sigma^2), & |C(s, p)| \geq \sigma^2; \\ 0 & \text{otherwise;} \end{cases} \quad (3.3)$$

A questo punto, usando il banco di filtri inverso e i blocchi di sovracampionamento, è possibile ricostruire l'immagine \hat{f} , che rappresenta una stima dell'immagine ideale f . In Figura 4 è mostrata una porzione di una pagina di testo filtrata con l'approccio wavelet usando la funzione *symlet* di ordine 4 come wavelet madre:

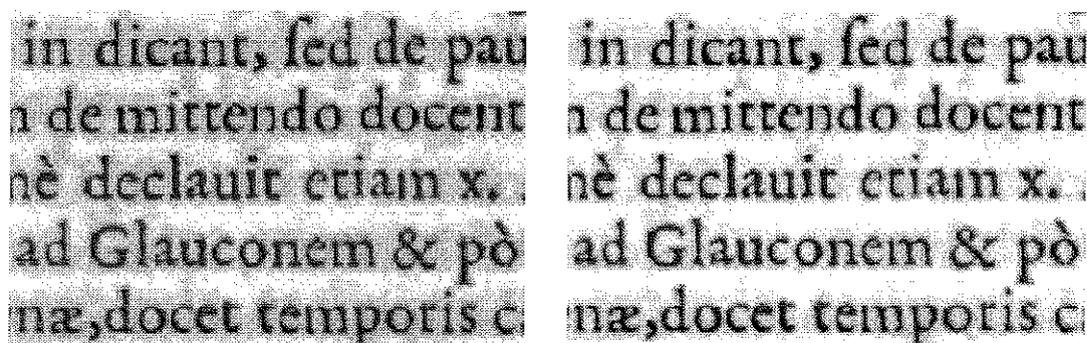


Fig. 4- Confronto tra l'immagine originale, a sinistra, e quella filtrata con la funzione wavelet *symlet*, a destra.

E' evidente come le variazioni associate al segnale di sfondo siano molto attenuate dopo il filtraggio, cosa che permette una binarizzazione più accurata e, di conseguenza, un migliore riconoscimento dei blob individuati.

4. Estrazione delle righe di testo contenute nel documento

Per identificare ed estrarre le singole righe di testo contenute nel documento è stato usato un approccio nel dominio della frequenza. Poiché le righe si ripetono nella pagina con una distanza fissa verticalmente, si può analizzare la trasformata di Fourier del segnale dato dalla proiezione lungo l'asse verticale dei valori associati a tutti i pixel. Lo spettro di ampiezza di questo segnale presenta tre picchi: uno a frequenza nulla, legato al valore medio dell'immagine, e due, ovviamente simmetrici rispetto all'origine, legati alla frequenza con cui le singole righe si ripetono. La stima di questa frequenza permette di ricavare il periodo del segnale costituito dalle righe di testo, e quindi, nota la posizione della prima riga, consente di estrarle tutte. Per identificare l'ordinata iniziale della prima riga, è possibile analizzare l'andamento del valore medio del segnale estratto dalle singole righe di pixel che compongono l'immagine: in questo modo si ottiene un segnale che presenta due massimi in corrispondenza delle linee di contenimento di una riga di testo, il margine inferiore e quello superiore di scrittura. La media di questi due valori rappresenta appunto l'ordinata iniziale di estrazione delle righe. Estratta così la prima riga di testo, poiché è nota la frequenza di ripetizione di questo segnale, si può agevolmente estrarre le rimanenti. In questo modo si

ottiene un vettore che contiene tutte le righe di testo che formano la pagina, pronte per essere analizzate dai blocchi successivi del sistema.

5. Binarizzazione e segmentazione adattiva delle righe

Il successivo passo della procedura esamina individualmente e in sequenza le righe estratte, e per ciascuna riga segmenta ed estrae i caratteri che la costituiscono. A tale scopo è utile effettuare una preliminare discriminazione fra pixel di sfondo e pixel corrispondenti ai caratteri, riportando i livelli di grigio dei pixel della riga dagli usuali 256 a due soli livelli, a e b . Tale processo verrà di seguito indicato come “binarizzazione”, sebbene i due livelli scelti, a e b , non siano 0 e 1, ma corrispondano ai valori medi dello sfondo e dei caratteri, calcolati empiricamente esaminando alcune porzioni della riga stessa, o esaminandone l’istogramma, che risulta bimodale. Assumendo caratteri scuri su sfondo chiaro, a tutti i pixel con valore al di sotto di una opportuna soglia th verrà assegnato un livello $a < th$, e a tutti i pixel con valore al di sopra di th verrà assegnato un livello $b > th$. I caratteri saranno quindi segmentati estraendo le varie componenti connesse a valore a . Il problema consiste quindi essenzialmente nell’individuare la soglia ottimale th . Un criterio intuitivo è quello di calcolare la soglia th a cui corrisponde la minima distanza fra l’immagine a due livelli e quella originale a più livelli.

Poiché, come già detto, la qualità dell’immagine varia molto da zona a zona, è ragionevole pensare che anche la soglia ottima possa variare da zona a zona, e per zone anche molto piccole. Per affinare ulteriormente il processo di binarizzazione, si può quindi pensare di suddividere la linea di testo da binarizzare in L sottoblocchi, individuabili per esempio separando anche grossolanamente i vari caratteri, e applicare iterativamente il metodo proposto alle singole sottoimmagini. In questo modo, la soglia scelta è ottima e adattiva, perché dipende dalla zona di testo studiata. In Figura 5 è mostrata una riga di testo estratta dal documento e binarizzata prima con un semplice approccio a soglia costante per tutta la riga, poi con la tecnica adattiva illustrata sopra.

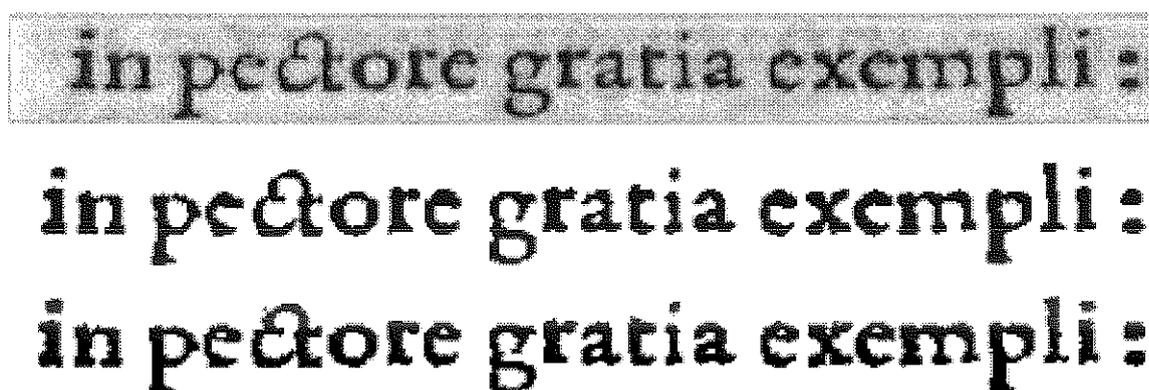


Fig. 5 – Dall’alto verso il basso: parziale riga di testo estratta dal documento originale in scala di grigi; binarizzazione della riga usando una soglia fissa, risultato ottenuto con la tecnica adattiva proposta: si noti come le “asole” delle ‘e’ siano ora ben evidenziate.

Come si può osservare, il risultato ottenibile con il metodo proposto è molto più accurato di quello derivante da una semplice operazione di soglia costante, consentendo una maggior discriminazione fra caratteri simili, come la “e” e la “c”.

Questo approccio permette di binarizzare l’immagine e segmentare i caratteri molto velocemente, ma non riesce ad eliminare completamente i problemi connessi ai caratteri che

appaiono attaccati o spezzati; tuttavia, data l'estrema semplicità e il buon funzionamento nella maggior parte dei casi, tale tecnica verrà utilizzata come primo input al successivo processo riconoscimento dei caratteri. Come già evidenziato in precedenza, i casi critici, individuabili attraverso un test sulla probabilità di riconoscimento, verranno trattati a parte con tecniche più sofisticate di blind deconvolution basate su modelli a campi di Markov.

6. Il cuore del sistema: la rete neurale per il riconoscimento dei caratteri

Una volta separati i singoli blob il sistema passa allo stadio successivo, quello di riconoscimento dei caratteri, realizzato mediante una rete neurale opportunamente progettata ed addestrata. La rete utilizzata è una feed-forward multistrato [9], addestrata usando un algoritmo di back-propagation [7,8,9] su un training set costruito usando caratteri estratti casualmente da una pagina del documento trattato. Nel caso della sperimentazione illustrata in questo lavoro sono stati utilizzati 1200 caratteri di una pagina dell'*Opera Omnia* di Girolamo Cardano, risalente al XVI secolo. Il training set costruito presenta come unico inconveniente un forte sbilanciamento nel numero di campioni per alcune lettere dell'alfabeto che hanno una bassa occorrenza nel documento (ad esempio, molte lettere maiuscole sono presenti in pochi campioni, come anche i numeri da 0 a 9 e i segni di punteggiatura). Tutte le lettere individuate sono state binarizzate ed estratte utilizzando il metodo di minimizzazione dell'energia del segnale residuo.

La topologia della rete è stata scelta rielaborando modelli già trattati in letteratura, e presenta uno stato iniziale formato da 180 neuroni, uno strato nascosto formato da 250 neuroni e uno strato di uscita con un numero di neuroni scelto in base al numero di caratteri da riconoscere (in questo caso è stato fissato questo numero considerando le comuni lettere dell'alfabeto, sia minuscole che maiuscole, più un set aggiuntivo di estensione che comprendesse sia i numeri da 0 a 9 che caratteri peculiari del testo analizzato quali la ae, la s allungata, e tutti i caratteri originariamente legati fra loro). Tutti i neuroni sono stati scelti con una funzione di attivazione di tipo logaritmico sigmoidale.

Per quanto riguarda l'addestramento, passo fondamentale affinché la rete fornisca risultati consistenti al problema trattato, è stato usato un algoritmo di back-propagation. L'algoritmo back-propagation di base permette l'aggiornamento dei pesi associati ad ogni strato della rete neurale usando le informazioni estratte dal calcolo della funzione di performance della rete. Poiché la rete prevede uno strato nascosto e non accessibile dall'esterno, l'unico modo per modificare i pesi associati a tale strato è quello di calcolare il gradiente della funzione di performance e quindi far propagare all'indietro le modifiche ai coefficienti dei vari strati a partire dall'ultimo, che è quello accessibile. Per misurare il grado di prossimità al risultato ottimale, con l'insieme di addestramento viene fornita alla rete una matrice che contiene, per ogni ingresso, l'uscita desiderata (matrice di 'target'). L'errore associato al riconoscimento viene così calcolato grazie al confronto tra l'uscita reale della rete e l'uscita desiderata. Il parametro che permette di valutare quantitativamente il comportamento della rete neurale è l'errore quadratico medio (mse). Analiticamente, l'algoritmo di back-propagation classico può essere riassunto dalla seguente relazione:

$$x_{k+1} = x_k - a_k g_k \quad (6.1)$$

dove x_k è il vettore corrente dei pesi associati alla rete, g_k il gradiente corrente e a_k la velocità di apprendimento della rete; la variabile k identifica il passo corrente dell'apprendimento ('epoca'). Esistono due modi in cui l'algoritmo può essere applicato, il modo incrementale, in

cui il gradiente viene calcolato dopo ogni ingresso presentato alla rete e i pesi vengono aggiornati di conseguenza, ed il modo batch, in cui prima di calcolare il gradiente della funzione di performance vengono proposti tutti gli ingressi alla rete e soltanto alla fine vengono aggiornati i vari pesi. Gli algoritmi classici di addestramento delle reti appartengono a queste due classi e hanno l'inconveniente di essere molto lenti nel convergere ad una soluzione accettabile in termini di mse. Una tecnica che permette un addestramento molto più efficiente (tempo di convergenza minore di un fattore compreso tra 10 e 100 rispetto agli altri algoritmi) è rappresentata dalla classe di algoritmi adattivi che modificano il valore della velocità di apprendimento durante l'addestramento della rete [6,10]: la tecnica usata per addestrare la rete neurale usata nel sistema di riconoscimento appartiene a questa classe di algoritmi.

Per approfondire alcuni aspetti sia teorici che realizzativi della tecnica usata, è necessario innanzi tutto considerare che l'algoritmo di addestramento è molto sensibile alla scelta del parametro a_k , la velocità di apprendimento. Infatti, se tale parametro è troppo grande, l'algoritmo può oscillare e diventare instabile; se al contrario è troppo piccolo, il tempo richiesto per convergere è molto lungo. E' evidente allora che una scelta ottimale di a_k ci permette di ottimizzare le prestazioni dell'algoritmo usato. L'idea di rendere la tecnica adattiva permette di mantenere le prestazioni del calcolo prossime a quelle ottimali per ogni passo computazionale. Questo può essere realizzato nel modo descritto di seguito.

Innanzitutto devono essere calcolate le uscite iniziali della rete ed i relativi errori; viene poi fissato un numero massimo di passi di addestramento in cui ci si aspetta che il valore dell'errore quadratico medio associato all'uscita converga ad un valore prestabilito, e viene calcolata la prima iterazione. Il nuovo valore da associare ai pesi di ciascuno strato viene valutato per ogni epoca usando il valore corrente per la velocità di apprendimento. A questo punto, se il nuovo errore supera l'errore associato al passo precedente di un certo valore prefissato, i nuovi pesi vengono tralasciati e la velocità di apprendimento decrementata linearmente usando un parametro fisso. Altrimenti, se il nuovo errore è minore di quello precedente, vengono aggiornati i pesi con i nuovi valori calcolati, e la velocità di apprendimento incrementata tramite un secondo parametro fisso.

Per diminuire la probabilità che l'addestramento della rete si arresti a causa di un minimo locale della funzione errore, viene comunemente applicata la tecnica dei momenti: in questo modo l'algoritmo diventa sensibile non solo al gradiente locale della funzione errore, ma anche all'andamento di questa valutato nei passi precedenti, agendo così come un filtro passa basso per ignorare i cambiamenti locali nella funzione errore.

Grazie all'implementazione di questa tecnica, che prende il nome di *batch gradient descent with momentum and variable learning rate* [6], l'addestramento della rete usata nel sistema integrato per il riconoscimento di caratteri ha impiegato un tempo pari a circa due ore su un PC dotato di processore Pentium III @ 700 MHz, 128 MB Ram, su sistema operativo Windows Me e ambiente di sviluppo Matlab Release 6. Il target per l'errore quadratico medio è stato fissato empiricamente a 0.001, come giusto compromesso tra affidabilità nel riconoscimento e tempo di addestramento.

Uno dei problemi essenziali che sono stati affrontati nel progetto della rete neurale è quello della standardizzazione dei vettori in ingresso. Data la grande disomogeneità delle dimensioni orizzontale e verticale dei blob individuabili nel testo da analizzare, si è ritenuto opportuno estrarre i singoli elementi di testo usando le informazioni fornite dall'algoritmo di segmentazione, e rimappare questi su griglie di dimensioni prefissate pari a 30 pixel sia in orizzontale che in verticale. Ovviamente, tale trattamento è stato applicato sia ai caratteri usati per l'addestramento della rete, che ai blob estratti nell'attività generale di riconoscimento.

Questa scelta permette sia di eliminare problemi legati a traslazioni spaziali dell'ingresso entro la zona individuata (le reti neurali, in generale, non sono invariati né a variazioni di scala, né a rototraslazioni dell'ingresso), sia di minimizzare il numero di neuroni nello strato iniziale della rete neurale. L'unico inconveniente legato a questa scelta è quello della perdita di informazioni legate alla posizione spaziale del carattere originale, cosa che in taluni casi può rendere difficoltoso il riconoscimento.

Come alternativa a questo, potrebbe essere utilizzata una rete neurale più complessa, per esempio una rete convoluzionale [13,14,15], che, calcolando N convoluzioni dell'ingresso con un kernel variabile arbitrariamente, permette di classificare il vettore di ingresso in base ad un certo numero di features invarianti sia alla scala che alle rototraslazioni. Ovviamente la complessità implementativa di una rete di questo tipo è molto maggiore, ed il conseguente addestramento è molto più lungo.

7. Trattamento di caratteri spezzati

Una volta che la rete è stata addestrata nel modo illustrato nel paragrafo precedente, la capacità di riconoscimento sui caratteri correttamente segmentati è piuttosto alta, almeno nei limiti dell'errore quadratico medio imposto come target per l'addestramento. Da numerosi esperimenti effettuati è stato riscontrato che, quando la probabilità associata al riconoscimento supera il 90%, la corrispondente segmentazione è abbastanza accurata e il riconoscimento è corretto. Quando invece la probabilità è bassa, il riconoscimento è sbagliato e la corrispondente segmentazione è usualmente inaccurata. In questo caso, è necessario ricorrere a tecniche alternative che ci permettano di ricalcolare il vettore di segmentazione, per poi procedere ad un nuovo riconoscimento.

Il caso più semplice che si può presentare è quello di due blob consecutivi molto vicini tra loro (entro 2-3 pixel), con probabilità di riconoscimento bassa e con estensioni orizzontali relativamente limitate. In questo caso vengono accorpati i due blob e la rete neurale viene forzata a riconoscere la nuova macrostruttura individuata. Qualora la probabilità associata al nuovo riconoscimento sia maggiore della probabilità media calcolata dai due precedenti riconoscimenti, il vettore di segmentazione viene aggiornato. Questa situazione è molto frequente nei testi analizzati, e corrisponde essenzialmente al caso della "n", della "u" o della "m" spezzate, come si vede nelle Figure 6 e 7. Infatti, a causa di dispersioni dell'inchiostro o a difetti di stampa, il legame delle due gambe della "n" viene spesso a mancare e la segmentazione preliminare individua in questo caso due singoli blob adiacenti di dimensioni orizzontali minime. Il riconoscimento di questi elementi, spesso, avviene con una bassa probabilità: mentre è facile che la rete riconosca il primo blob come una r, è praticamente impossibile associare al restante carattere una uscita valida (cioè ad alta probabilità).

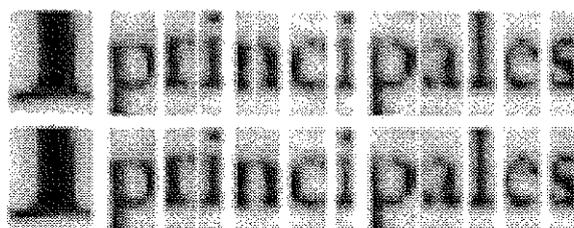


Fig. 6– In alto: risultato della segmentazione adattiva, sovrapposto all'immagine originale a livelli di grigio; in basso: il vettore di segmentazione è stato aggiornato in base all'analisi dei risultati forniti dalla rete neurale.

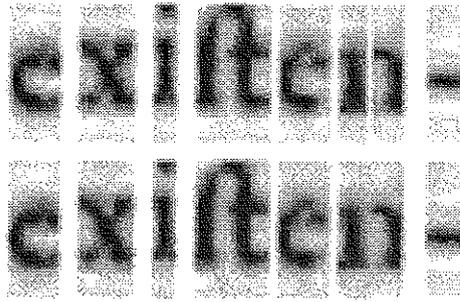


Fig. 7 – Anche in questo caso, l'analisi del risultato prodotto dalla rete neurale permette di ricostruire la corretta segmentazione per la n spezzata.

Il problema inverso, quello della separazione di due caratteri connessi per effetto della diffusione dell'inchiostro, è di più difficile soluzione e necessita di strumenti molto più potenti ed avanzati.

8. Uso di modelli a campi di Markov e deconvoluzione cieca per la separazione di due o più caratteri congiunti

Quando due o più caratteri consecutivi risultano connessi tra di loro, una semplice operazione di soglia, anche se locale, non permette di separare i singoli caratteri in blob disgiunti. Questo fatto è estremamente dannoso per la rete neurale, che si trova a dover associare una uscita ad un pattern di ingresso chiaramente non convenzionale. In questi casi si è riscontrato che miglioramenti significativi della binarizzazione, e quindi della segmentazione, possono essere ottenuti rielaborando la zona critica con tecniche di deconvoluzione cieca [16,17] abbinata a modelli a campi di Markov [18,19] per immagini a due livelli di grigio.

Questo tipo di tecniche si basano sulla definizione di una opportuna funzione energia avente come variabili sia l'immagine da binarizzare che i coefficienti della maschera di blur. La funzione energia contiene un termine di consistenza fra i dati e l'immagine stimata, più vari altri termini, detti potenziali, che descrivono i vincoli desiderati per l'immagine. In accordo al formalismo dei campi di Markov, tali potenziali sono funzioni locali di piccole configurazioni di pixel (clique), e sono particolarmente flessibili nel descrivere vincoli locali sulla geometria della scena.

Chiamando h la maschera di blur incognita, il modello di degradazione completo è ora dato da:

$$\mathbf{g} = \mathbf{H}(h)\mathbf{f} + \mathbf{n} \quad (8.1)$$

dove \mathbf{g} , \mathbf{f} ed \mathbf{n} sono le rappresentazioni lessicografiche vettoriali di dimensione MN per g , f ed n , rispettivamente, e $\mathbf{H}(h)$ è la matrice che realizza la convoluzione discreta bidimensionale come prodotto matrice-vettore. I coefficienti di $\mathbf{H}(h)$ dipendono dai coefficienti di h secondo una legge nota.

Il modello a campi di Markov usato nella definizione dei potenziali è un'estensione del modello di Ising, che considera un sistema di intorno di ordine massimo pari a 3 [18]. Un campo di Markov è un processo stocastico caratterizzato dalla seguente distribuzione di probabilità, detta *distribuzione di Gibbs*:

$$P(f) = \frac{1}{Z} e^{-\lambda U(f)} \quad (8.2)$$

dove f è l'immagine da stimare, Z è la funzione di partizione, ossia una costante di normalizzazione, e λ un parametro positivo. Il termine energia $U(f)$ definisce il contributo integrale dei potenziali associati alle possibili configurazioni degli intorno di pixel nell'immagine, ed è così definito:

$$U(f) = \sum_{c \in C} V_c(f) \quad (8.3)$$

dove il termine $V_c(f)$ indica il potenziale associato alla particolare clique c esaminata, cioè a un determinato insieme di pixel adiacenti, all'interno dell'intorno considerato. Il contributo di tutti i potenziali delle clique viene poi sommato per avere il termine totale associato alla particolare configurazione assunta dall'intorno di pixel esaminato. Nel modello di Ising, i potenziali sono funzioni tabellari delle configurazioni di clique, e, nel nostro caso, sono caratterizzati da 14 parametri liberi, scelti empiricamente. In particolare, tali potenziali tendono a favorire l'uguaglianza di due pixel adiacenti nelle varie direzioni, ed esprimono la frequenza di occorrenza aspettata per particolari configurazioni di pixel di testo [20,21].

La funzione energia da minimizzare è data da:

$$E(f, h) = \|\mathbf{g} - \mathbf{H}(h)\mathbf{f}\|^2 + \lambda U(f) \quad (8.4)$$

dove \mathbf{g} rappresenta l'immagine originale, in scala di grigio su 8 bit per pixel, e cioè su 256 livelli di grigio, mentre f è su due livelli, scelti nell'intervallo $[0,255]$ usando le informazioni estratte dall'istogramma bimodale di \mathbf{g} . Il parametro positivo λ , già introdotto, rappresenta il peso relativo dei potenziali, cioè delle informazioni a priori, rispetto al termine di consistenza con i dati, espresso dal primo termine della (8.4).

Tale energia è non convessa e a variabili miste, continue e discrete, e per la sua minimizzazione occorrono quindi algoritmi di ottimizzazione non convessa, basati in tutto o in parte su tecniche stocastiche, tipo annealing simulato [22]. Per trattare funzioni a variabili miste, usualmente si adotta uno schema iterativo in cui un passo di minimizzazione rispetto all'immagine, data la stima corrente della maschera di blur, si alterna ad un passo di minimizzazione rispetto alla maschera di blur, data la stima corrente dell'immagine. Quindi, all'interno di uno schema di annealing simulato, la minimizzazione rispetto all'immagine, a due valori, si realizza mediante un algoritmo di Metropolis o un campionatore di Gibbs, mentre la minimizzazione rispetto alla maschera di blur h , a valori continui, si realizza con algoritmi di gradiente discendente. Schemi di calcolo di questo tipo, sebbene efficienti nel calcolare l'ottimo globale del problema, risultano usualmente computazionalmente molto costosi, perché a lenta convergenza. Per ovviare in parte a questi inconvenienti, l'approccio proposto si basa sull'assunzione che la maschera di blur possa essere stimata con buona approssimazione già nei primi passi dell'algoritmo, e possa quindi essere mantenuta costante per i passi successivi, nei quali verrà calcolata solo l'immagine. Grazie a questo modo di procedere, il processo risulta molto velocizzato. In Figura 8 è mostrata, come esempio, la maschera di blur calcolata usando 5 iterazioni del processo di deconvoluzione cieca completo, per una sottoimmagine contenente un carattere "u". Come valore iniziale per l'immagine è stata usata l'immagine iniziale degradata, e come stima iniziale per il blur un impulso bidimensionale [11,12].

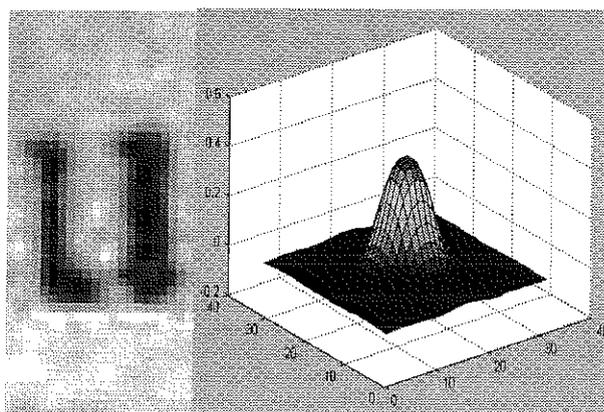


Fig. 8 – Lettera ‘u’ e stima locale della maschera di blur, usando 5 iterazioni del processo di deconvoluzione cieca.

Una volta stimata la maschera di blur, l’energia (8.4) ha come unica variabile l’immagine f a valori discreti (2 nel nostro caso). Poiché l’energia risulta comunque ancora non convessa, si adotta un algoritmo di annealing simulato (SA), caratterizzato da un parametro T , detto temperatura, che determina la varianza, attorno al massimo, della probabilità associata all’energia. Tale temperatura viene fatta decrescere lentamente da un valore iniziale sufficientemente grande (distribuzione quasi uniforme) fino a zero (distribuzione pari a una delta di Dirac concentrata sul massimo dell’energia). Nel SA, ad ogni valore della temperatura, vengono proposte variazioni (scambi) casuali per i vari pixel di f , esaminati individualmente e più volte secondo un ordine prestabilito. Il criterio di accettazione della variazione è probabilistico, in accordo a un algoritmo di Metropolis o a un campionario di Gibbs [22,23], in modo tale che per valori alti della temperatura possano essere accettate anche variazioni del valore del pixel che incrementano l’energia, così da uscire da eventuali minimi locali. Al tendere a zero della temperatura, solo le variazioni che diminuiscono l’energia vengono accettate, garantendo così la convergenza dell’algoritmo.

Nella specifica implementazione del SA, il parametro λ è stato fatto variare con una legge lineare in modo opposto alla variazione della temperatura di processo, in modo da rendere l’algoritmo adattivo, per pesare inizialmente allo stesso modo i due termini dell’energia (consistenza con i dati e informazioni a priori), e di aumentare l’effetto associato al modello per l’immagine al crescere del numero di iterazioni. Con questo procedimento sono stati ottenuti risultati soddisfacenti per la separazione dei blob connessi già dopo un numero di iterazioni compreso tra 10 e 15. In Figura 9 è mostrato un esempio dell’applicazione della tecnica descritta a due caratteri congiunti.

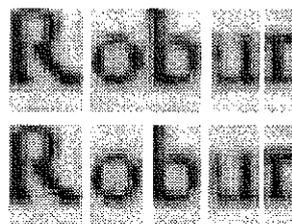


Fig. 9– In alto: risultato della segmentazione adattiva; in basso: segmentazione del blob ‘ob’ mediante l’uso di modelli a campi di Markov.

9. Risultati sperimentali dell'analisi di una riga di testo

Il sistema descritto è stato sperimentato su una pagina estratta dall'*Opera Omnia* di Girolamo Cardano. Nel seguito vengono illustrati i risultati ottenuti dall'analisi di una riga di testo. In Figura 10 è mostrata la riga estratta dalla pagina originale.

Fig. 10 – Riga di testo individuata ed estratta per la segmentazione ed il riconoscimento.

L'immagine estratta è stata sottoposta al filtraggio wavelet per la riduzione del rumore, e ad una successiva operazione di binarizzazione adattiva, come illustrato nel paragrafo 5. Il risultato della segmentazione preliminare, sovrapposto all'immagine a livelli di grigio, è illustrato in Figura 11.

Fig. 11 – Segmentazione preliminare della riga di testo di Figura 10, sovrapposta all'immagine a livelli di grigio.

Dall'esame della figura sono evidenti alcuni difetti di segmentazione, imputabili a errori nel processo di binarizzazione: per esempio, le 'n' di 'principales' e di 'existen' sono risultate entrambe spezzate, e quindi identificate come due blob consecutivi, mentre il gruppo di caratteri 'ob', a sua volta, non è stato correttamente segmentato, e i due caratteri sono rimasti congiunti.

Limitatamente a queste situazioni, il sistema procede ad un affinamento della segmentazione, secondo quanto descritto in precedenza: la segmentazione viene rieffettuata per quei blob per cui la rete neurale non ha prodotto un risultato affidabile, evidenziandolo con una bassa probabilità di riconoscimento (nel nostro caso, minore del 90%). Il primo passo è la verifica di blob adiacenti relativamente poco estesi orizzontalmente, con bassa probabilità di riconoscimento, e poco distanti nel vettore di segmentazione: per questi elementi viene tentato l'approccio di aggancio forzato ed il successivo riconoscimento della macrostruttura individuata. Il nuovo vettore di segmentazione, adesso, separa correttamente le due 'n', ma lascia ancora agganciati i due caratteri che formano il blob 'ob'. In questa particolare situazione, il sistema automaticamente procede all'affinamento della binarizzazione, mediante l'applicazione della tecnica basata su deconvoluzione cieca e modelli a campi di Markov. Questo processo viene applicato solo alla sottoimmagine contenente il blob da analizzare, e il vettore di segmentazione ottenuto viene fornito nuovamente in input alla rete neurale. In Figura 12 è mostrata la segmentazione finale.

Fig. 12 – Segmentazione finale della riga di testo di Figura 10, sovrapposta all'immagine a livelli di grigio.

La stringa riconosciuta dalla rete neurale in questo caso è risultata essere “principales (t)obur virium(l) (si)orbi existen -“, dove solo 3 blob su un totale di 34 sono riconosciuti con una probabilità inferiore al 90%. La difficoltà nel riconoscere i caratteri R, M e la virgola è essenzialmente dovuta al fatto che nel training set fornito alla rete durante l'apprendimento erano presenti solo pochi esemplari di tali caratteri.

Conclusioni

In questo lavoro è stato presentato un sistema integrato per l'elaborazione e l'analisi di documenti a stampa antichi anche fortemente degradati, finalizzato al riconoscimento dei caratteri di testo. Il sistema, costituito da più blocchi eseguiti in cascata, a partire da una singola immagine del documento riduce il rumore presente nell'immagine, utilizzando una tecnica di decomposizione e filtraggio basata sulla trasformata wavelet, allinea le righe di testo, le identifica e le estrae dalla pagina, quindi le segmenta con un semplice e veloce approccio a soglia adattiva, e analizza i vari blob individuati usando una rete neurale feed-forward multistrato, addestrata con un algoritmo di back-propagation. La probabilità di riconoscimento in uscita dalla rete viene utilizzata come un parametro discriminante per l'attivazione automatica di un processo di retroazione, che riporta il sistema ad un blocco di raffinamento della segmentazione. Questo blocco agisce solo sulla piccola porzione di testo dove si è riscontrato un riconoscimento non attendibile, e utilizza tecniche di decodifica cieca e di segmentazione a campi di Markov, la cui originale complessità risulta ridotta quando applicate su poche sottoimmagini di piccola dimensione. I risultati ottenuti dalla sperimentazione su pagine dell'*Opera Omnia* di Girolamo Cardano hanno evidenziato che il sistema, nel suo insieme, permette di ottenere una segmentazione dei singoli caratteri molto precisa e un riconoscimento con una alta efficienza, rendendo possibile e relativamente veloce l'elaborazione e l'analisi di testi anche fortemente deteriorati.

Riferimenti bibliografici

- [1] Coehn, A.; (1995), 'Wavelets and multiscale signal processing', Chapman and Hall.
- [2] Coehn, A.; I. Daubechies; B.Jaerth; P. Vial; (1993), 'Multiresolution analysis, wavelet and fast wavelet transform on an interval', CRAS Paris ser. A, t.316, pp.417-421.
- [3] Coifman, R.R.; Y. Meier; M. V. Wickerhauser; (1992), 'Wavelet analysis and signal processing', in Wavelets and their applications, M. B. Ruskai et al. (Eds.), pp.153-178, Jones and Bartlett.
- [4] Donoho, D. L.; (1995), 'Denoising by soft-tresholding', IEEE Transaction on information theory, 41(3), pp.613-627.
- [5] Strang, G.; T. Nguyen; (1996), *Wavelets and filter banks*, Wellesley – Cambridge press.
- [6] Vogl, T. P.; J. K. Mangis; A. K. Rigler; W. T. Zink and D. L. Alkon; (1988), 'Accelerating the convergence of the back propagation method', Biological Cybernetics, vol. 59, pp.256-264.
- [7] Rumelhart, D. E.; G. E. Hinton and R. J. Williams; (1986), 'Learning representation by back propagating errors', Nature, vol.323, pp.533-536.
- [8] Rumelhart, D. E.; G. E. Hinton and R. J. Williams; (1998), 'Learning representation by error propagation', in D. E. Rumelhart and J. L. Mc Clelland, eds. Parallel Data processing, vol.1, Cambridge, The MIT press, pp.318-362.
- [9] Rosenblatt, F.; (1961), 'Principles of Neurodynamics', Washington D.C.; Spartan-Press.

- [10] Nguyen, D. and B. Widrow; (1990), 'Improving the learning speed of two layer neural networks by choosing initial values of the adaptive weights', Proceedings of the international joint Conference on Neural Networks, vol.3, pp.21-26.
- [11] Kundur, D. and D. Hatzinakos; (1996), 'Blind image deconvolution', IEEE Signal processing magazine, pp.43-62.
- [12] Yang Yongyi; Henry Stark; Nikolas P. Galatsamos; (1993), 'A projection based approach to the blind deconvolution problem', in SPIE vol.2029, Digital Image Recovery and Synthesis.
- [13] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, 'Handwritten digit recognition with a back propagation network', AT&T Bell Laboratories, Holmdel, N. J. 07733.
- [14] Denker, J. S.; Gardner, W. R.; Graf, H. P.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, L. D.; Baird, H. S.; and Guyon, I.; (1989), 'Neural network recognizer for hand written zip code digits', in Touretzky, D.; editor, Advances in neural information processing systems, vol.1, pp.323-331, Denver. Morgan Kaufmann.
- [15] Fukushima, K. and Miyake, S.; (1982), 'Neocognitron: a new algorithm for pattern recognition tolerant to deformations and shifts in position', Pattern Recognition, 15, pp. 455-469.
- [16] G. R. Ayers and J. G. Dainty, (1988), "Iterative blind deconvolution method and its applications", Opt. Lett., 13, pp. 547-549.
- [17] Y. You and M. Kaveh, (1996), "A regularization approach to joint blur identification and image restoration", IEEE Trans. Image Processing, 5, pp. 416-428.
- [18] S. Z. Li, (1995), *Markov Random Field Modeling in Computer Vision*, Springer-Verlag, Tokyo.
- [19] F. C. Jeng and J. W. Woods, (1991), "Compound Gauss-Markov random fields for image estimation", IEEE Trans. Signal Processing, 39, pp. 683-697.
- [20] A. Tonazzini and L. Bedini, (1999), "Character segmentation in highly blurred ancient printed documents", Proc. 10th Int. Conf. on Image Analysis and Processing (Venice, September 27-29, 1999), pp. 836-841.
- [21] A. Tonazzini and L. Bedini, (2000), "Segmentation of printed characters in ancient degraded documents through blind image restoration and labeling", Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications", vol. 10, n. 2, pp. 273-287.
- [22] E. Aarts and J. Korst, (1989), *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons.
- [23] S. Geman and D. Geman, (1984), "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", IEEE Trans. Pattern Anal. Machine Intell., 6, pp. 721-740.
- [24] K. Aas and L. Eikvil, "Text page recognition using grey-level features and hidden Markov models", *Pattern Recognition*, 29(6), 1996, pp. 977-985.
- [25] H. I. Avi-Itzhak, T. A. Diep and H. Garland, "High Accuracy Optical Character Recognition Using Neural Networks with Centroid Dithering", *IEEE Trans. PAMI*, 17(2), 1995, 218-224.