

Sviluppo di un prototipo per la verifica della corretta disposizione di volumi della biblioteca di Ingegneria degli Studi di Firenze

Mandelli Lorenzo

E-mail address

lorenzo.mandelli@stud.unifi.it

Montaghi Alessandro

E-mail address

alessandro.montaghi@stud.unifi.it

February 21, 2021

1 Introduzione

La gestione manuale dei libri di una biblioteca, di una libreria o di un magazzino, può risultare dispendiosa in termini di tempo ed impegno organizzativo. Riguardo all'ambito del sistema bibliotecario, la creazione manuale dell'inventario dei volumi in dotazione, il loro periodico aggiornamento ed il giornaliero riordino dei volumi disposti sugli scaffali, risulta molto costoso in termini di manodopera e può facilmente essere soggetto ad errori.

A tale proposito, per facilitare la gestione all'interno delle biblioteche, sono stati proposti e sviluppati numerosi sistemi di gestione automatizzata per facilitare l'identificazione dei libri posizionati sugli scaffali. Tra questi sistemi, possiamo ricordare il metodo di contrassegnare fisicamente ogni libro con un identificatore di tipo RFID (Radio-frequency identification) o dotare ogni volume di etichette (tags) riportanti un codice a barre (barcode). Questi identificatori RFID e etichette barcode possono essere lette, tramite appositi lettori, dal personale della biblioteca per gestire l'inventario, stabilire la corretta collocazione dei libri ed estrarre le informazioni riguardanti i volumi presenti.

Comunque, i sistemi basati su barcode o RFID possono avere costi elevati, dovuti ad esempio alle etichette e all'acquisto degli appositi lettori, che li rendono accessibili solo alle grandi istituzioni. Inoltre, l'impiego del lettore limita al solo personale la possibilità di reperire le informazioni riguardo i volumi posizionati sugli scaffali.

Per tali motivi, sono stati proposti nuovi sistemi di tracciamento dei volumi più convenienti da un punto di vista di costi ed utilizzabili anche dai normali utenti della biblioteca o della libreria. In generale e per limitare i costi di realizzazione e di esercizio, i sistemi proposti si basano sull'impiego di apposite

applicazioni sviluppate per essere installate sugli smartphone moderni e sfruttarne le caratteristiche hardware che questi possiedono. Gli attuali smartphone sono dotati di fotocamere ad alta risoluzione con messa a fuoco automatica, di sensori di orientamento e localizzazione e permettono l'accesso a reti ad alta velocità come la rete WiFi, oramai presente in ogni biblioteca. In ambito bibliotecario, queste numerose funzionalità rendono interessante l'uso di queste apparecchiature per l'identificazione e l'inventariazione automatizzata delle risorse all'interno di una biblioteca. Tramite l'impiego di un normale smartphone, il personale o un utente interessato può fotografare il dorso (o costa) dei libri presenti su di un dato scaffale. Le immagini raccolte possono essere inviate a un server centrale che le elabora, ne estrae le informazioni dal database della biblioteca e identifica automaticamente i vari volumi. Le informazioni elaborate dal server centrale possono essere restituite sotto forma di metadati che possono riportare la corretta posizione del volume, la descrizione del libro, informazioni riguardo l'autore o gli autori, la casa editrice e così via. Inoltre, ulteriori ed aggiornate informazioni possono essere recuperate da Internet, o dai server di altre biblioteche, e trasferite allo smartphone e al database centrale, aggiornando in questo modo l'inventario.

L'impiego di sistema per la gestione dei volumi di una biblioteca basato sull'uso di apparecchiature smartphone richiede lo sviluppo di appositi metodi di text detection (rilevamento del testo) e text recognition (riconoscimento del testo). In particolare, il fulcro su cui sono basati i metodi di text detection è quello di sviluppare appositi algoritmi capaci di distinguere il testo dallo sfondo. Comunque, come riportato da diversi autori [20, 17, 5, 4], esistono diverse problematiche nell'applicare la text detection e recognition utilizzando immagini o filmati che ritraggono il dorso dei libri. Per esempio, gli algoritmi sviluppati per riconoscere scene naturali danno risultati non soddisfacenti quando impiegati nel riconoscere il testo riportato sul dorso dei libri [13]. Alcuni algoritmi di text detection e recognition assumono che i libri ripresi siano perfettamente allineati lungo l'asse verticale e che ogni immagine venga scattata di fronte allo scaffale [12]. Più in generale, i dorsi dei libri presentano diversi colori, le scritte hanno caratteri e dimensioni diverse, presenza di artefatti grafici oltre al testo (ad esempio, il logo della casa editrice), testi con diversi orientamenti, le biblioteche presentano illuminazioni diverse a seconda della sezione e del momento del giorno e così via. Tutti questi aspetti possono influenzare in modo negativo le normali tecniche di text detection e recognition.

In questo lavoro, si propone un sistema per l'identificazione e l'estrazione delle informazioni riguardanti i volumi presenti in una biblioteca universitaria.

Il sistema proposto (1) utilizza la fotocamera dello smartphone per raccogliere le immagini delle costole dei libri e (2) la rete WiFi del cellulare per trasferire le immagini ad un server centrale. Il server centrale, una volta ricevuta l'immagine, (3) applica delle apposite procedure per la text detection e recognition dell'immagine stessa e (4) estrae le informazioni riguardanti il libro. Queste informazioni (5) sono trasferite allo smartphone per una loro visualizzazione. La



Figure 1: Veduta della Biblioteca della Scuola di Ingegneria dell’Università degli Studi di Firenze.

parte centrale della procedura proposta riguarda la text detection e recognition. A tale proposito, il sistema proposto impiega una scene text detection pipeline, conosciuta con l’acronimo di EAST (Efficient and Accurate Scene Text Detector), ed implementata da Zhou et al. [23]. La pipeline proposta dagli autori, che utilizza un modello di tipo Fully Convolutional Network (FCN), individua direttamente le aree in cui sono presenti linee di testo nelle immagini in ingresso, escludendo i passaggi intermedi che sono sovente lenti e ridondanti.

2 Related work

In letteratura sono descritte diverse tecniche di text detection e recognition applicate al dorso dei libri. Alcuni dei lavori presentati in questa sezione descrivono metodi che migliorano alcuni aspetti della text detection e recognition dei testi presenti sulla costa dei volumi, mentre altri articoli riportano sistemi completi per la creazione di soluzioni per l'inventariazione automatica dei libri nelle biblioteche.

In Taira et al. [18] viene proposto una tecnica per l'identificazione dei bordi dei libri basata su l'impiego di modello di automi di stato finito, formulato a partire dalla disposizione dei libri su di uno scaffale. In una seconda fase, il titolo del libro è estratto da ogni immagine derivante dalla prima fase.

Castro et al. [6] hanno presentato un sistema di scaffali iterativi basati su una combinazione di camera-proiettore che sfrutta, per il riconoscimento dei libri, l'istogramma generato a partire dai colori dei coste dei volumi.

Nella ricerca presentata da Quoc e Choi [14] viene descritto un sistema per riconoscere i libri sugli scaffali impiegano dei robot con telecamere. Questi robot, individuano i dorsi dei singoli libri sugli scaffali tramite le linee rette prodotte a partire dalle immagini prelevata dalla fotocamera in dotazione. Sfruttando queste linee, viene rilevato il bordo del libro, individuato l'area che racchiude il dorso del libro (bounding box) e poi successivamente, viene letto il testo del libro tramite appositi algoritmi.

In Chen et al. [5, 4] viene impiegato l'algoritmo SURF (Speeded-up robust features) [2] per il riconoscimento dei libri dal loro dorso. Usando uno smartphone per scattare le immagini degli scaffali, da queste immagini l'algoritmo SURF estrae le informazioni necessarie alla sua identificazione. Queste informazioni sono poi utilizzate per interrogare la banca dati centrale della biblioteca. Inoltre, tramite le informazioni sulla posizione dal dispositivo mobile, utilizzando i suoi sensori, è possibile creare un inventario di libri in grado di riconoscere la loro posizione.

Matsushita et al. [13] hanno sviluppato uno scaffale interattivo tramite l'uso di videocamere posizionate sul soffitto. Il riconoscimento dei singoli libri avviene tramite la copertina dei volumi, quando questi sono rimossi dallo scaffale.

In Tsai et al. [20] è implementato un sistema per l'inventariazione dei libri tramite un pipeline ibrida di riconoscimento dei volumi dai loro dorsi. Il sistema proposto è composto da due fasi in sequenza. Nella prima fase, detta di Book Spine Extraction, viene identificato il bordo del dorso dei libri presenti sugli scaffali per ottenere l'immagine dei singoli dorsi. In una seconda fase, chiamata Book Spine Recognition, da ogni immagine precedentemente estratta viene rilevato e riconosciuto il testo riportato sul dorso. Infine, il testo estratto viene usato come parola chiave per cercare informazioni in un dato database.

Duan et al. [8] hanno proposto un sistema per il riconoscimento del dorso del libro basato sul rilevamento del numero dell'etichetta di riferimento presente sul dorso dei libri nelle biblioteche. Questi autori hanno usato la segmentazione sul colore per rilevare le etichette e una segmentazione sulle linee per identificare i bordi dei singoli dorsi dei libri. Infine, tramite un'operazione detta di contour

clustering ed utilizzando il Tesseract OCR Engine [16], il sistema proposto da questi autori rileva il numero di identificazione del libro.

Talker e Moses [19], hanno descritto una segmentazione dei dorsi dei libri da qualsiasi prospettiva. Il sistema descritto utilizza un modello detto di shape dependent active contour per individuare i bordi rettangolari dei libri.

Yang et al. [22] hanno sviluppato un sviluppo un sistema per la lettura dei testi dal dorso dei volumi, basato su modelli deep-neural networks, specificatamente costruito per la creazione e la gestione degli inventari dei libri delle biblioteche. Il sistema sviluppato da questi autori, prende in ingresso le immagini dei libri posizionati sugli scaffali, applica un algoritmo di (text/no-text) CNN (Convolutional Neural Network) per effettuare l'operazione segmentazione dei bordi dei dorsi di ogni libro, di localizzazione dei testi all'interno di ogni dorso segmentato e di individuazione delle singole parole nei testi localizzati [9]. In seguito, per la fase di text recognition, vengono applicati modelli basati su Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN).

Il prototipo di sistema di inventariazione libri, proposto da Cao et al. [3] e chiamato BSRBOT (Book Spine Recognition Based on OpenCV and Tesseract), è basato su tecniche che rientrano nell'ambito della image processing. Presa un'immagine corrispondente ai libri di uno scaffale, il sistema applica una prima fase detta di Book Spine Segmentation. Questa fase coinvolge una serie di algoritmi per ottenere come risultato finale l'identificazione del dorso di tutti i libri presenti nell'immagine. In successione, viene effettuata l'identificazione ed il riconoscimento del numero di identificazione riportato sull'etichetta apposta sul dorso di ogni libro. Tale numero è usato come query per estrarre informazioni dal database. Infine e solo in casi particolari, ovvero quando il numero di identificazione risulta non leggibile (es. etichetta rovinata e non leggibile), viene applicata una terza fase definita Spine Book image matching che associa l'immagine del dorso del libro alle immagini dei dorsi raccolti in un apposito database.

Recentemente, Anegawa e Arisugi [1] hanno studiato l'applicazione del metodo Rotational Region CNN (R^2CNN) [10], basato su di una CNN, per il rilevamento del testo riportato sui dorsi dei libri. Gli autori hanno addestrato la CNN con dati provenienti da un altro dominio, ossia il dataset ICDAR 2015 [11], ed hanno successivamente migliorato le prestazioni dell'algoritmo per la text detection dei volumi posizionati in uno scaffale. Inizialmente, l'algoritmo R^2CNN genera le regioni candidati usando una Region Proposal Network (RPN). I risultati ottenuti dall'applicazione diretta di R^2CNN hanno presentato due principali problemi, ossia un dorso poteva presentare diverse bounding box e una bounding box poteva non coprire correttamente il testo riportato. Per queste ragioni, sono state applicate operazioni di Grouping e Adjusting per migliorare la prestazione finale del sistema.

3 Materiali e Metodi

3.1 Obiettivi

L'obiettivo di questo progetto è quello di realizzare un'applicazione in grado di validare la corretta disposizione dei libri sugli scaffali della biblioteca della Scuola di Ingegneria dell'Università degli Studi di Firenze.

L'applicazione sviluppata, che prende in ingresso un'immagine contentente i dorsi dei libri disposti su uno scaffale, è strutturata nella seguente pipeline di lavoro:

- Le zone dell'immagine in cui è presente del testo sono rilevate utilizzando l'algoritmo di text detection EAST.
- Un'algoritmo di clustering, il DBSCAN, è applicato alle zone individuate. Ogni cluster rappresenta una scritta contigua, ossia il titolo e/o l'autore di un singolo volume.
- Ad ogni cluster è applicato l'algoritmo di text recognition *Google Tesseract* per estrarne il testo contenuto.
- I testi ottenuti sono utilizzati per determinare in quale sezione della biblioteca è stata scattata l'immagine attraverso un confronto con il database della biblioteca. Infine il sistema tenta di rilevare l'eventuale presenza di libri posizionati in modo errato nello scaffale.

Il sistema sopra descritto è schematizzato nella seguente figura 2.

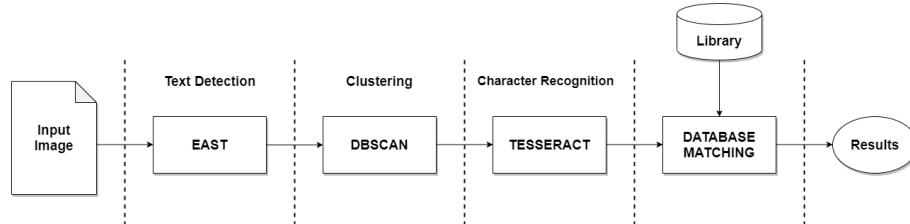


Figure 2: Pipeline di lavoro dell'applicazione.

3.2 Algoritmi

Nel corso degli anni sono stati proposti diversi algoritmi per la *text detection* e per la *text recognition*. Per la realizzazione di questo progetto, sono stati impiegati i seguenti algoritmi.

3.2.1 EAST

L'algoritmo di text detection EAST (*Efficient and Accurate Scene Text Detector*) [23] utilizza una singola rete neurale per individuare le parole o le righe

di testo presente in un’immagine. In combinazione a un qualsiasi metodo di riconoscimento del testo, EAST rappresenta la fase iniziale per un work-flow di gestione dei testi.

In dettaglio, questo algoritmo, è di tipo (fully) convolutional network con fase detta di *non-max suppression* (NMS) *merging state* per produrre l’output finale. La (fully) convolutional network viene utilizzata per individuare il testo contenuto nell’immagine e la NMS viene usata per unire le diverse caselle di testo, rilevate in modo impreciso, in un unico riquadro di delimitazione. L’idea alla base di questo algoritmo è quella di rilevare grandi regioni di testo che richiedono features dalla fase successiva della rete neurale, mentre vengono rilevate piccole regioni di parole che richiedono features di basso livello dalle fasi iniziali. Per creare questa rete, gli autori hanno utilizzato tre rami (*Feature Extractor Stem*, *Feature Merging Branch* e *Output Layer*) che si combinano in un’unica rete neurale finale.

Per quanto riguarda l’impiego di EAST nella realizzazione del sistema proposto, è stata utilizzata un’implementazione basata su Tensorflow disponibile gratuitamente al seguente repository github: "<https://github.com/argman/EAST>". Utilizzando il linguaggio di programmazione Python versione 3.6.9 (data di rilascio: Luglio 2, 2019), la libreria EAST è stata integrata all’interno del workflow del sistema. Tramite apposite funzioni in Python, che richiamano al loro interno l’algoritmo EAST, questo riceve in ingresso un’immagine, scattata mediante cellulare, che ritrae i volumi presenti su di un scaffale e produce in uscita la posizione delle aree, le *bounding box*, caratterizzate dal contenere del testo.

Al fine di ottimizzare le successive fasi del workflow, le zone rilevate con area sotto una certa dimensione vengono rimosse in quanto spesso riferite a numeri identificativi presenti sui dorsi dei libri della biblioteca, ma non presenti come informazione nei database. La soglia scelta è di tipo adattativo e cambia in funzione della dimensione massima delle zone rilevate, in questo modo essa risulta essere invariante rispetto alla distanza a cui la foto viene stata scattata.

3.2.2 DBSCAN

Date le caratteristiche implementative di EAST e le specificità delle immagini prese in considerazione nel progetto, questo algoritmo individua il testo riportato sui dorsi dei volumi sotto forma di singole parole, ossia ogni parola è all’interno di un *dato bounding box*. Al fine di raggruppare queste parole tra loro sulla base di appartenenza al singolo volume, si è resa necessaria una fase intermedia di clusterizzazione dei *bounding box*. Tra i vari algoritmi di clustering considerati [21], il DBSCAN è quello che ha ottenuto i migliori risultati sia in termini di precisione che in tempo di esecuzione.

Il DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) è un algoritmo di clustering di tipo density-based, proposto nel 1996 da Martin Ester, Hans-Peter Kriegel, Jörg Sander e Xiaowei Xu [15]. Esso permette



(a) Immagine

(b) Risultati

Figure 3: Data un’immagine in ingresso, raffigurante i libri presenti su di un dato scaffale della Biblioteca della Scuola di Ingegneria di Firenze, l’algoritmo EAST restituisce una serie di bounding box che rappresentano le aree delle immagini con presenza di testo. (a) Immagine originale e (b) Bounding box individuati tramite l’algoritmo EAST.

di connettere regioni di punti con densità sufficientemente alta e di isolare gli outlier presenti nell’insieme dei dati. L’approccio più comune del DBSCAN è il cosiddetto *center-based*, che prevede di stimare la densità di un particolare punto contando i punti all’interno di una circonferenza con un raggio stabilito r . In conseguenza a questo, è possibile dividere i punti all’interno di ciascuna circonferenza in *core points* (punti in cui all’interno della circonferenza cade un numero di punti maggiori o uguali ad una soglia MinP decisa dall’utente), *border points* (punti che non sono core points ma che cadono comunque all’interno di una circonferenza di un altro punto) e *noise points* (gli outlier).

Per lo sviluppo della pipeline del progetto, è stato utilizzato l’algoritmo DBSCAN implementato nella libreria di Python *scikit-learn Machine Learning* (versione 0.24.1).

L’algoritmo è stato applicato sui vertici dei bouding box ottenuti tramite la precedente elaborazione di EAST. Essendo la disposizione dei testi nell’immagine prevalentemente lungo l’asse principale del dorso dei volumi, sono state provate varie impostazioni di metrica da utilizzare durante la fase di clustering. La distanza euclidea pesata maggiormente lungo l’asse delle ascisse è quella che ha fornito i migliori risultati di clusterizzazione dei vertici. In questo modo, è stato possibile unire nello stesso cluster aree i cui vertici sono contenuti in elissi verticali centrate nei punti stessi.

Successivamente all’applicazione dell’algoritmo DBSCAN sono stati applicati alle zone di testo tre ulteriori criteri di raggruppamento:

- vertici appartenti alla stessa Bounding Box sono riuniti nello stesso cluster.

- le zone di testo le cui aree hanno intersezione non vuota sono riunite nello stesso cluster.
- se un punto interno a una zona di testo è a distanza minore di una soglia rispetto a un vertice di un'altra zona, esse sono riunite nello stesso cluster. La soglia scelta è la stessa distanza euclidea pesata utilizzata durante la clusterizzazione attraverso DBSCAN.



Figure 4: Risultato del clustering dell’algoritmo DBSCAN **(b)** applicato ai vertici delle bounding box individuate attraverso EAST **(a)**.

3.3 Google Tesseract

Il Tesseract è un sistema di riconoscimento di caratteri (*OCR, Optical Character Recognition*) [16]. Questo sistema permette di convertire il testo contenuto in un’immagine in caratteri comprensibili ad un elaboratore di testi. Il Tesseract OCR è stato originariamente sviluppato presso Hewlett-Packard Laboratories tra il 1985 e il 1994. Nel 2005 è stato reso open source da Hewlett-Packard. Dal 2006, l’ulteriore sviluppo di Tesseract è stato sponsorizzato da Google. Dalla versione 4, Tesseract ha aggiunto nuove funzionalità di deep-learning basate su reti di tipo LSTM (Long short-term memory).

Il framework Google tesseract è stato applicato alle zone di testo rilevate nell’immagine attraverso EAST e clusterizzate secondo le modalità descritte nelle sezioni precedente. Al fine di migliorare le prestazioni di Tesseract, gli sviluppatori di Google consigliano di applicare un pre-processamento alle immagini. A tale fine, tramite la libreria *OpenCV* [7], da ogni singola zona di testo sono state ottenute ulteriori due immagini derivante ognuna dall’applicazione di due filtri:

- **Gray Scale**, ossia ogni immagine è stata convertita in un scala di grigi.

- **Image Thresholding**, cioè utilizzando un valore globale di soglia (*threshold binary*) ogni immagine è stata convertita in un'immagine binaria.

L'immagine originale, l'immagine in gray scale e l'immagine binaria sono state usate come ingresso al Tesseract OCR.

Infine, dato che nella libreria della Scuola di Ingegneria sono presenti testi in lingua italiana ed inglese, nelle impostazioni finali di Tesseract sono state considerate queste due lingue.

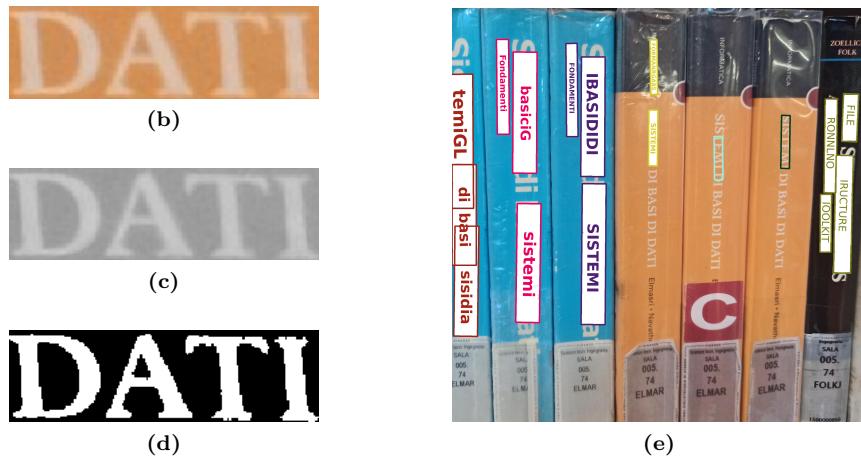


Figure 5: In (b), (c) e (d) sono mostrate le elaborazioni eseguite su una bounding box individuata dall'algoritmo EAST per l'applicazione del framework Google Tesseract, in (e) è mostrato il risultato complessivo ottenuto.

3.4 Database Matching

Il procedimento di Database Matching è suddivisibile in due sottosezioni: la ricerca della sezione più probabile e la verifica degli errori.

3.4.1 Ricerca della sezione più probabile

In questa parte si vuole stabilire in quale sezione degli scaffali della biblioteca l'immagine sia stata presa e quali libri essa contenga. Il procedimento sviluppato è il seguente:

- Per ogni cluster di stringhe, ottenute attraverso il framework Google Tesseract nelle modalità descritte nella sezione precedente, si vuole trovare quali, tra tutti i libri del database della biblioteca, siano i candidati più probabili. A questo fine per ogni stringa del cluster è eseguito l'algoritmo *Edit Distance* rispetto a ogni parola di ogni libro presente nel database. Ai libri che risultano avere la corrispondenza massima (ovvero con distanza

di *Edit Distance* minima) è associato un valore di confidenza espresso dalla formula:

$$confidence = \frac{size(word) - d}{size(word)} \quad (1)$$

dove *word* è la parola del cluster, *size(word)* è il suo numero di caratteri e *d* è la distanza minima ottenuta attraverso il confronto con *Edit Distance*. Una volta calcolati tutti i valori di confidenza e i rispettivi libri a distanza minima per ogni parola del cluster, i valori appartenenti agli stessi libri sono sommati tra loro. Infine i libri con valore finale di confidenza massimo sono selezionati per i successivi passi del procedimento.

- I valori di confidenza e i relativi libri di ogni cluster ottenuti sono riuniti in un solo istogramma che ha per ascissa il numero identificativo di ogni libro del database e per ordinata il relativo valore complessivo di confidenza ottenuto.

Nel caso preso in esame della biblioteca dell'università degli Studi di Firenze, il database dei libri non presenta l'informazione degli indici del libro iniziale e finale degli scaffali. Ciò non consente di tentare di stabilire univocamente quale fosse lo scaffale su cui è stata presa l'immagine e di conseguenza quanti libri essa contenesse.

Per ovviare a questa problematica l'applicazione cerca di stabilire direttamente quali siano gli indici del libro iniziale e finale presenti nella foto e considera il numero di libri presenti come un input proveniente dall'utente. Utilizzando quindi l'istogramma sopra calcolato, è eseguita la ricerca di quale sotto sequenza di indici contigui comporti la massima somma di valori di confidenza associati.

Definisce quindi gli indici iniziali e finali dei libri presenti nell'immagine come i valori di inizio e di fine della sottosequenza ottenuta.

3.5 verifica degli errori

Al fine di verificare la corretta disposizione dei libri dell'immagine è necessario considerare le due seguenti operazioni:

- controllare che non siano presenti libri provenienti da altre sezioni della biblioteca e quindi fuori posto.
- verificare che l'ordinamento dei libri presenti nell'immagine rispetti quello indicato dal database.

Il primo punto è ottenuto verificando che ogni cluster abbia una confidenza maggiore di appartenere alla sezione compresa tra indici iniziali e finali calcolati nella sezione precedente che rispetto all'esterno di essi. Viene verificato infatti che la somma dei valori di confidenza dei libri compresi nell'intervallo considerato e associati a un cluster sia maggiore di qualsiasi altra confidenza che il

cluster abbia con un libro presente all'esterno dell'intervallo.

I cluster che non rispettano questo criterio sono segnalati come possibili errori.

Per verificare invece se l'ordinamento dei libri presenti nell'immagine rispetta quello indicato dal database della biblioteca, è stato necessario cercare di associare ciascun libro del database nell'intervallo considerato ad almeno un cluster. L'algoritmo per l'assegnazione opera nel seguente modo: per ogni cluster è calcolato quale sia il libro più probabile da associare secondo le modalità viste nella sezione 3.4.1. Successivamente è verificato che il libro candidato non fornisca un valore di confidenza maggiore in prossimità di altri cluster: se ciò non avviene si associa il libro al cluster considerato, se invece ciò avviene si opera una nuova ricerca partendo dal cluster che ha dato risultato massimo in modo ricorsivo.

Questo procedimento permette di associare ogni libro candidato, preso dal database della biblioteca nell'intervallo considerato, a un cluster in modo che la confidenza di associazione tra essi risulti massimale.

Infine una volta completate le associazioni i cluster sono riordinati in base alle proprie ascisse e si verifica se la successione dei rispettivi libri candidati associati rispetta quella indicata dal database. I cluster che non rispettano questo criterio sono segnalati come possibili errori.

4 Risultati

L'applicazione software sviluppata è stata infine testata su un campione di 10 immagini provenienti dalla biblioteca di Ingegneria dell'Università degli Studi di Firenze e contenenti all'incirca 140 volumi in varie condizioni ambientali.

Il sistema è stato quindi testato in base alle sue capacità di:

- individuare la sezione della biblioteca in cui le foto siano state scattate.
- di associare correttamente i libri presenti nell'immagine ai volumi presenti nel database della biblioteca.
- rilevare eventuali volumi mal posizionati presenti nelle immagini.

4.1 Errori di sezione

In questa fase si vuole analizzare i risultati prodotti dal sistema durante la fase di *ricerca della sezione più probabile*, descritta precedentemente nel paragrafo 3.4.1.

Al fine di quantificare l'errore commesso nell'individuazione dell'intervallo dei libri presenti nell'immagine è stato scelto di utilizzare una metrica di errore data dalla formula:

$$E = 1 - \frac{|X \cap Y|}{|Y|} \quad (2)$$

con X l'insieme di volumi predetto dal sistema e Y l'insieme di volumi effettivamente presenti nella foto.

L'errore di sezione ottenuto sul campione di immagini ha fornito un valore medio pari a 0.869 e varianza 0.010.

Immagine	# volumi totali	# match corretti	# match errati	# volumi non rilevati
1	13	9	2	2
2	11	8	1	2
3	15	10	2	3
4	14	8	3	3
5	14	6	5	3
6	11	6	4	1
7	14	9	3	2
8	17	16	1	0
9	17	6	1	10
10	14	4	7	3

Table 1: Risultati del *matching cluster-volume* ottenuto su un campione di 10 immagini contenenti 140 libri.

4.2 Errori di matching

In questa sezione sono analizzati le capacità del sistema di riconoscere correttamente un volume presente in un’immagine attraverso il testo estratto con il framework *Google Tesseract*, nelle modalità descritte nella sezione 3.3, e utilizzando l’intervallo di libri predetto durante la fase di *ricerca della sezione più probabile*, descritto nel paragrafo 3.4.1.

Per ogni libro effettivamente presente nell’immagine il sistema può dare come risultato un’associazione corretta con il relativo volume nel database, un’associazione errata con un’altro volume presente nell’immagine o la mancata rilevazione attribuibile alle fasi precedenti di text detection, clusterizzazione e text recognition. I risultati ottenuti, mostrati in tabella 1, permettono di calcolare il valore medio e la varianza dei match corretti che assumono rispettivamente i valori di 0.614 e 0.054. Tale valore di varianza è attribuibile alle varie condizioni ambientali riscontrabili nelle immagini utilizzate come campione: in condizioni di scarsa illuminazione, di presenza di pellicole trasparenti o di testo rovinato il riconoscimento dei volumi da parte dell’applicazione può risultare fortemente penalizzato.



Figure 6: Esempio di applicazione del sistema proposto in cui l’immagine presenta un libro mal posizionato.

	# totali	# corretti	# falsi positivi
Errore collocazione	4	1	3
Errore ordinamento	0	0	0

Figure 7: Tabella che mostra il numero di errori di ordinamento e collocazione ottenuti.

4.3 Errori di collocazione e ordinamento

Infine in questa sezione è testata la capacità del sistema di rilevare volumi mal posizionati nelle immagini o provenienti da altre sezioni della biblioteca.

A questo scopo definiamo:

- **Errore di collocazione:** le segnalazioni di errore prodotte dall'applicazione durante la fase di verifica degli errori riguardante la presenza di libri provenienti da altre sezioni.
- **Errore di ordinamento:** le segnalazioni di errore prodotte durante la fase di verifica degli errori riguardante l'ordinamento dei libri nell'immagine.

La capacità del sistema di rilevare queste tipologie di errore è direttamente collegata alla precisione delle sue fasi precedenti di *ricerca della sezione più probabile*, sezione 3.4.1, e di *matching cluster-volume*, sezione 3.5.

Un esempio di rilevazione di errore di collocazione e di ordinamento ottenuto è riportato in figura 6. Come si può vedere il sistema riesce a rilevare correttamente il libro mal posizionato nell'immagine (il libro caratterizzato dal dorso di colore bianco), ma allo stesso tempo predice alcuni errori di collocazione che sono in realtà falsi positivi.

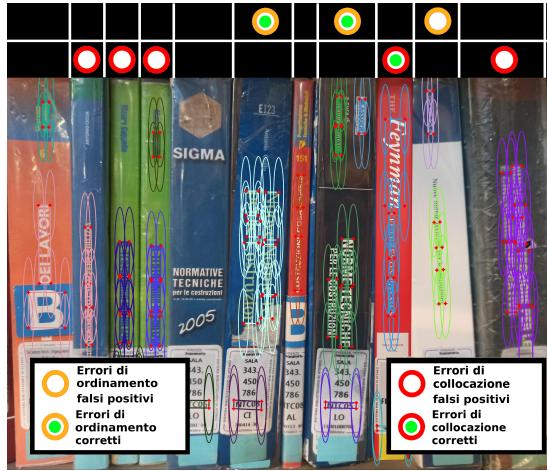


Figure 8: Esempio di applicazione del sistema proposto in cui l'immagine presenta un libro mal posizionato e due libri il cui ordine è stato scambiato.

	# totali	# corretti	# falsi positivi
Errore collocazione	5	1	4
Errore ordinamento	3	2	1

Figure 9: Tabella che mostra il numero di errori di ordinamento e collocazione.

L'applicazione del sistema proposto è fortemente dipendente dalla qualità dell'immagini e dalle condizioni in cui esse sono stata catturate come è possibile notare dall'esempio in figura 7, che mostrano i risultati in presenza di condizioni non ideali. Nell'immagine sono infatti presenti volumi ricoperti da pellicole trasparenti e libri di piccola dimensione. Nonostante il sistema riesca infatti a predirre correttamente sia l'errore di collocazione che i due errori di ordinamento presenti, restituisce un maggior numero di falsi positivi rispetto al caso precedente.