Riconoscimento automatico di libri di letteratura fantastica

di Giacomo Fantazzini - giacomo.fantazzini2@studio.unibo.it

Obiettivo

I romanzi e i racconti brevi di letteratura fantastica vengono spesso ripubblicati in Italia con titoli, traduzioni e contenuti aggiuntivi differenti. Per questo motivo, l'identificazione del reale contenuto di un certo volume da parte di lettori e collezionisti è sempre una sfida.

Lo scopo di questo progetto è sviluppare un'applicazione per smartphone basata su reti neurali convoluzionali in grado di aiutare l'utilizzatore a identificare l'edizione e il contenuto di un certo volume semplicemente scattandone una foto alla copertina.

Il codice dell'intero progetto e il dataset prodotto per la sua realizzazione sono consultabili su *GitHub*[0]. Tutti i link alle risorse sono riportati in fondo al documento.

Costruzione del dataset

Vista la difficoltà nel reperire i campioni con cui addestrare la rete, ho deciso di limitare il riconoscimento dei libri al solo sottogenere "Fantascienza Cyberpunk" e agli autori in qualche modo collegati a questo movimento, mantenendo comunque un approccio generalizzabile all'intera letteratura fantastica.

La principale fonte di informazioni a riguardo è certamente il "*Catalogo Vegetti della letteratura fantastica pubblicata in Italia*"[1], disponibile online.

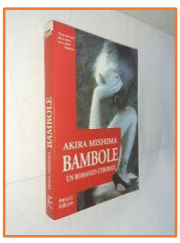
Come prima cosa, ho realizzato uno script che, data la lista degli autori da coprire, visita le relative pagine di catalogo e memorizza i collegamenti alle pagine di ciascuna opera prodotta. Successivamente, per ciascuna opera, visita tutte le pagine di catalogo dei volumi che la includono (edizioni nel caso dei romanzi e raccolte nel caso dei racconti brevi) e ne scarica:

- titolo;
- autore (o curatore nel caso delle raccolte);
- foto di copertina;
- codice *NILF* (identificatore univoco del libro all'interno del catalogo).

Questo script è realizzato in *Python* e fa uso delle librerie *BeautifulSoup* (parser HTML) e *MechanicalSoup* (suo wrapper con funzionalità da Web browser).

Per l'effettiva creazione del dataset, ho realizzato un secondo script che effettua per ciascun libro una ricerca su *eBay*[2] tramite titolo e autore e scarica tutte le foto dai relativi annunci di libri in vendita. Al termine di questa fase, è stata necessaria una pulizia manuale degli errori dovuti a edizioni differenti con stessa coppia titolo-autore e a quei titoli troppo corti e generici per poter escludere oggettistica di altro tipo dalla ricerca.

Per aumentare la copertura dei volumi senza sufficienti campioni, ho impostato un'ulteriore ricerca automatizzata per scaricare le foto dagli annunci pubblicati sul sito "*Compro Vendo Libri*"[3]. Al termine di questa fase, il dataset consiste di 4839 fotografie afferenti a 656 volumi, ciascun volume avente tra le 4 e le 20 foto.



Preprocessing e data-augmentation

Ho optato per una suddivisione del dataset in train, validation e test set con una proporzione 70-20-10 (assicurandomi che ciascun set avesse almeno un'immagine per ciascun libro) e ho preparato una funzione che applica uno scaling e un padding a ciascuna immagine in modo da renderle tutte quadrate e della stessa dimensione (impostabile in modo parametrico).

Per migliorare la generalità del test set, ho deciso di sottoporlo a data-augmentation fino a portare il numero di campioni a 60 immagini per ciascun volume. Avendo in alcuni casi pochi campioni di partenza, ho preferito scrivere a mano le funzioni di augmentation in modo da introdurre il maggior numero possibile di variazioni significative, applicando cambiamenti casuali (entro limiti ragionevoli) di:

- luminosità e contrasto;
- sfocatura e nitidezza;
- traslazione, rotazione e scala;
- shear (molto leggero);
- ricolorazione (molto leggera);
- rumore gaussiano aggiuntivo.

Ho infine preparato due funzioni che organizzano i set a livello di filesystem nelle strutture richieste rispettivamente dai framework *TensorFlow* e *PyTorch*. Come etichette di classe ho utilizzato il codice identificativo *NILF* dei volumi.

Creazione e addestramento della rete

Ho effettuato i primi esperimenti su *TensorFlow v1* addestrando da zero una rete ispirata a *LeNet-5* e altre due reti simili proposte dai paper "*Judging a Book by its Cover*"[4] e "*Deep Learning Approaches towards Book Covers Classification*"[5], che trattano il problema della previsione del genere letterario di un libro a partire dalla sua copertina. Questo approccio non ha prodotto risultati soddisfacenti, raggiungendo un'accuracy sul test set inferiore al 30% anche nei casi migliori.

Nel tentativo di ottenere risultati migliori, ho successivamente fatto delle prove di transfer-learning e fine-tuning in *PyTorch* a partire da una rete *AlexNet* preaddestrata su *ImageNet*. In questo modo ho ottenuto delle accuracy attorno all'85%.

Visto il potenziale di questa tecnica, ho iniziato a sperimentare reti via via più complesse, sia facendo fine-tuning su tutti i parametri della rete (*full*), sia riaddestrando unicamente i parametri dell'ultimo strato fully-connected (*fast*). Ho ottenuto in questo modo i seguenti risultati:

Nome del modello	Dimensione del campione	Accuracy (fast)	Accuracy (full)	Note
AlexNet	227 × 227 × 3	84,4 %	86,4 %	storage richiesto > 200 MB
ResNet-18	224 × 224 × 3	96,0 %	98,9 %	
ResNet-50	224 × 224 × 3	96,7 %	-	
Inception-v3	299 × 299 × 3	89,7 %	-	

PyTorch supporta unicamente la quantizzazione da FP32 a INT8. Le tecniche proposte per fare ciò sono tre: dinamica, statica PTQ (Post Training Quantization) e statica QAT (Quantization Aware Training). La quantizzazione dinamica si applica al modello normalmente addestrato, tuttavia gli strati convoluzionali non sono supportati e produce risultati poco soddisfacenti in questo dominio applicativo. Le due tecniche statiche producono risultati migliori, ma necessitano di applicare modifiche alla struttura della rete prima dell'addestramento, rendendo difficile il confronto con la rete base al fine di valutarne costi e benefici. In particolare, la tecnica QAT dovrebbe produrre risultati superiori rispetto al PTQ, a scapito di un tempo di addestramento significativamente più lungo.

I risultati ottenuti sulle reti quantizzate con tecnica QAT e ottimizzate per architetture ARM si sono dimostrati abbastanza coerenti con quelli relativi alle reti originali:

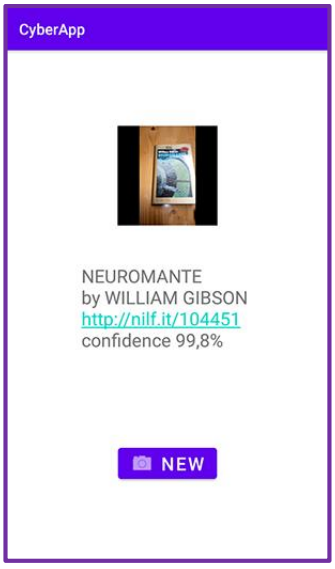
Modello quantizzato	**Dimensione del campione**	**Accuracy (fast)**	**Accuracy (full)**	**Note**
AlexNet	227 × 227 × 3	85,6 %	86,8 %	PTQ (QAT non supportata)
ResNet-18	224 × 224 × 3	89,1 %	98,0 %	
ResNet-50	224 × 224 × 3	91,6 %	-	tempo di inferenza > 10 s
Inception-v3	299 × 299 × 3	-	-	TorchScript non supportato

La rete selezionata per la distribuzione è la *ResNet-18* riaddestrata con fine-tuning di tutti i parametri, quantizzata a 8 bit con tecnica QAT, ottimizzata per dispositivi mobili ARM e convertita in formato *PyTorch Lite*. Questa rete pesa 11 MB, non presenta rallentamenti in fase di inferenza e ha dimostrato un'accuracy sul test set del 98%.

Deployment su smartphone Android

Ho realizzato un'applicazione Android scritta in *Kotlin* che incapsula la rete e la controlla tramite le API *PyTorch Mobile*. Accanto alla rete, l'applicazione contiene le coppie titolo-autore da associare a ciascuna etichetta (il link alla relativa pagina di catalogo online può essere derivato dall'etichetta stessa).

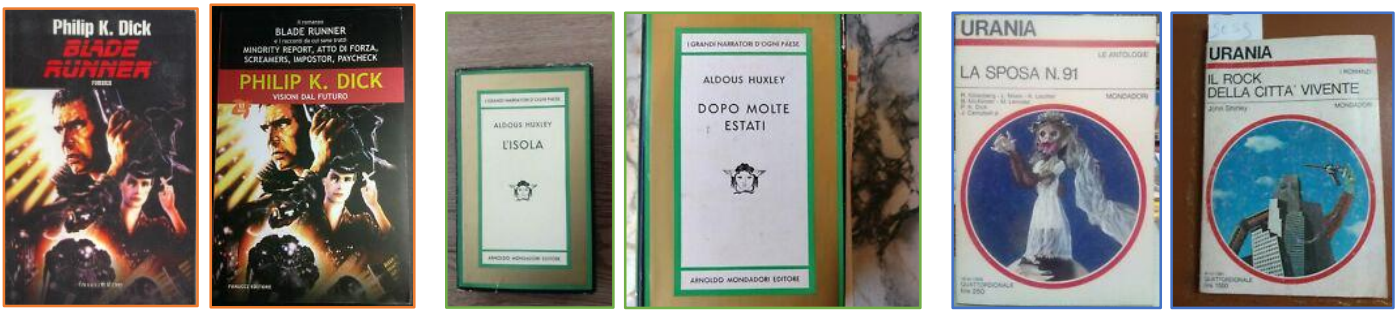
L'app presenta un pulsante che, se premuto, avvia la fotocamera e consente all'utente di scattare una fotografia. Quando l'utilizzatore conferma la fotografia, l'applicazione la sottopone a padding e scaling, la fa analizzare alla rete e lancia un browser Web sulla pagina online del *Catalogo Vegetti* relativa al volume riconosciuto. Quando si ritorna all'applicazione, è possibile visualizzare una miniatura della fotografia elaborata, i dati offline e la percentuale di confidenza del risultato ottenuto, oltre al link per tornare al catalogo.



Conclusioni e sviluppi futuri

L'applicazione funziona correttamente e dimostra una precisione soddisfacente. Tutti i test aggiuntivi effettuati fotografando libri veri hanno avuto esito positivo con discreti margini di confidenza. L'accuratezza si è rivelata piuttosto robusta anche in caso di foto di scarsa qualità e di occlusioni parziali del soggetto.

Gli errori di classificazione più frequenti si sono rivelati, come prevedibile, quelli relativi a volumi appartenenti alla stessa collana e aventi allo stesso tempo illustrazioni simili.



La principale limitazione di questa applicazione è certamente la ristrettezza del catalogo, dovuta alla scarsità di immagini su cui addestrare la rete e alla necessità di una fase di rimozione manuale degli errori tra le immagini scaricate in automatico.

Effettuando una nuova ricerca di immagini a poche settimane dall'inizio del progetto, lo script è riuscito a trovare almeno una nuova fotografia per quasi tutti i libri su cui è stato eseguito. È dunque presumibile che, ripetendo questa operazione periodicamente nel corso di un paio d'anni, si riesca a migliorare significativamente la qualità del dataset.

Per quanto riguarda la fase di pulizia manuale, sarebbe possibile semplificarla introducendo una nuova rete addestrata a distinguere i libri dagli altri tipi di oggetto. Si potrebbe anche pensare di introdurre un ulteriore classificatore che suddivida le immagini di diverse edizioni di uno stesso libro per somiglianza e che le pre-etichetti confrontandole con le copertine provenienti dal catalogo.

Si noti inoltre che diversi errori di classificazione della rete hanno in realtà evidenziato errori di etichettatura nel dataset (successivamente corretti) dovuti a riedizioni con copertine molto simili tra loro. È dunque pensabile che la rete stessa possa essere usata a posteriori per migliorare la qualità del dataset in vista di eventuali applicazioni future.

Desidero sottolineare, infine, che il fine-tuning di reti preaddestrate si è confermato come una tecnica fondamentale per superare il problema del dataset di piccole dimensioni con molte classi da riconoscere.

Link, fonti e citazioni

[0] https://github.com/JackFantaz/CyberApp

[1] https://www.fantascienza.com/catalogo/

[2] https://www.ebay.it/

[3] https://www.comprovendolibri.it/home.asp

[4] https://arxiv.org/pdf/1610.09204.pdf

[5] https://pdfs.semanticscholar.org/dc97/f5e9176ba278085cd2c56db56b8851fa864c.pdf