

RISOLUTORE DI SUDOKU IN PYTHON CON OPENCV E KERAS

Simone Fidanza

22/07/2022

Sommario

Per estrarre la griglia di un sudoku da un'immagine dello stesso è stata utilizzata la libreria OpenCV, in seguito le singole cifre sono state analizzate dalla Convolutional Neural Network ispirata al modello LeNet5. Dopo aver predetto le cifre e aver costruito una griglia artificiale di numeri, il sudoku viene risolto da un'algoritmo.

1 Introduzione

L'obiettivo ultimo dell'applicazione è quello di risolvere un sudoku da un'immagine. Per poter fare ciò sono necessari diversi passaggi:

1. Fornire un'immagine in input contenente un sudoku;
2. Localizzare la griglia del sudoku e estrarla;
3. Data la griglia, individuare ogni singola cella della stessa;
4. Determinare se la cella contiene una cifra e se sì effettuare il riconoscimento ottico dei caratteri (per brevità OCR);
5. Risolvere il sudoku mediante un algoritmo;
6. Mostrare il sudoku risolto all'utente.

L'applicazione fa ampio utilizzo delle librerie OpenCV¹ [1], TensorFlow [6], Keras [2] e NumPy [4].

2 Localizzazione della griglia

Come prima cosa, dopo aver ricevuto l'immagine di un sudoku in input, è necessario individuare la griglia del sudoku e per farlo è stata utilizzata la libreria OpenCV. L'immagine iniziale subisce una serie di passaggi, ovvero:

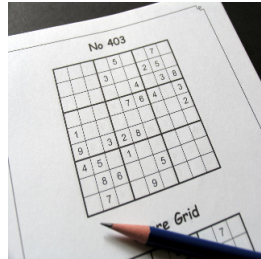
- viene convertita in scala di grigi con `cv2.cvtColor()`;
- viene applicata una sfocatura Gaussiana con `cv2.GaussianBlur()` per renderla più uniforme;
- viene segmentata con `cv2.adaptiveThreshold()`² per trasformarla in

¹acronimo in lingua inglese di *Open Source Computer Vision Library*

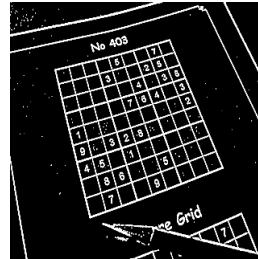
²è stata preferita una soglia adattiva per conservare quanti dettagli possibile dell'immagine

un'immagine binaria;

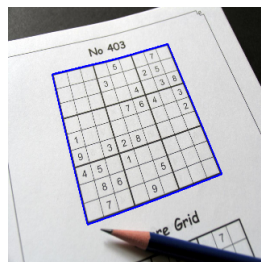
- viene invertita con `cv2.bitwise_not()`;
- viene infine erosa con `cv2.erode()` e successivamente dilatata con `cv2.dilate()` per rimuovere il rumore presente nell'immagine.



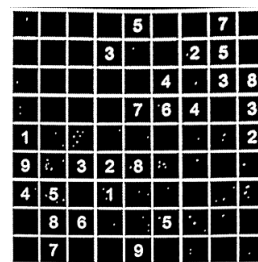
(a) Originale



(b) Pre-elaborazione



(c) Griglia



(d) Deformazione

Figura 2.1: Processo di estrazione della griglia di un sudoku

Il risultato di questi passaggi è illustrato in figura 2.1b. Dopo che l'immagine è stata pre-elaborata, si può procedere all'estrazione della griglia.

Innanzitutto è necessario individuare il bordo della griglia. È stato supposto che questo sia il più grande tra i bordi che la funzione `cv2.findContours()` restituirà, come illustrato in figura 2.1c. Approssimando il bordo con `cv2.approxPolyDP()` è possibile ottenere gli angoli della griglia. Usando sia quest'ultimi che `cv2.getPerspectiveTransform()` che `cv2.WarpPerspective()` l'immagine viene deformata e resa piana, come illustrato in figura 2.1d.

3 Analisi delle celle

Per determinare se una cella contiene un numero o meno è necessario innanzitutto estrarre le singole celle dalla griglia piana. Per fare ciò, sono state divise per 9 le dimensioni della griglia e sono state estratte 81 celle di tali dimensioni. Attraverso un semplice algoritmo viene verificato se la cella contiene un numero o meno. Se la cella contiene un numero, questo viene estratto e analizzato dalla rete neurale, la predizione viene inserita nella lista rappresentante la griglia del sudoku. Se la cella non contiene un numero, nella lista viene inserito il numero 0.

4 Modello

Per poter effettuare l'OCR è necessario un modello di Machine Learning. Il modello proposto è un miglioramento della rete neurale LeNet5 [5]. Il modello non è altro che una semplice Convolutional Neural Network (CNN per brevità) che è stata allenata sul dataset MNIST [3] che contiene 60,000 immagini per l'allenamento e 10,000 per l'analisi. Il modello è illustrato in figura 4.1.

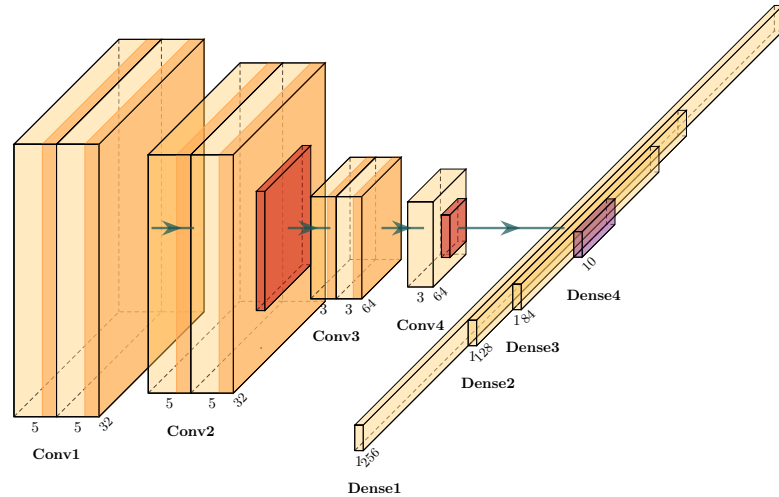


Figura 4.1: Architettura della rete neurale LeNet5+. Ogni piano rappresenta un layer della Rete Neurale.

Per allenare LeNet5+ è stato utilizzato un tasso di apprendimento variabile. Nel momento in cui il modello registra che l'apprendimento è “stagnante”, il tasso dello stesso viene ridotto di 0.2 attraverso la classe `ReduceLR0nPlateau()` del modulo `callbacks` di `keras`. Utilizzando l'accorgimento precedente e allenando il modello per 30 epoche, con una dimensione del lotto di 86, esso ha raggiunto un'accuratezza del $\approx 99.67\%$.

5 Risoluzione del sudoku

Dopo aver estratto le cifre dalla griglia e aver formato una griglia “digitale” del sudoku, quest'ultima viene passata alla funzione `SudokuSolver.solve()` che risolve il sudoku. Infine viene stampata schermo la soluzione.

6 Limitazioni conosciute

Poiché l'algoritmo di estrazione della griglia non è perfetto, seguono le limitazioni conosciute:

- nel deformare e appiattire un'immagine già piana, questa viene ruotata di 90° in senso orario. Per ovviare a questo inconveniente, è stata aggiunta una rotazione in senso antiorario;

- nel determinare se una cella è priva o meno di numero, poiché l'algoritmo si basa sul numero di pixel bianchi, a volte l'algoritmo può ignorare celle che contengono dei numeri. Per questo motivo la funzione `SudokuExtractor.construct_board()` accetta un parametro opzionale che è proprio il valore minimo di pixel bianchi;
- nel predire le cifre, la rete neurale confonde il numero 1 col numero 7 e raramente il numero 6 col numero 5 o 8³

Riferimenti bibliografici

- [1] G. Bradski. «The OpenCV Library». In: *Dr. Dobb's Journal of Software Tools* (2000).
- [2] François Chollet et al. *Keras*. <https://keras.io>. 2015.
- [3] Li Deng. «The mnist database of handwritten digit images for machine learning research». In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.
- [4] Charles R. Harris et al. «Array programming with NumPy». In: *Nature* 585.7825 (set. 2020), pp. 357–362. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [5] Yann LeCun et al. «Gradient-based learning applied to document recognition». In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.
- [6] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: <https://www.tensorflow.org/>.

³questo accade quando la parte superiore del numero 6 è troppo incurvata verso il basso