Google Landmark Recognition

Gianmarco Baiocchi
Politecnico di Torino
Torino, Italy
s289686@studenti.polito.it

Abstract

In questo paper viene presentata una soluzione alla challenge Google Landmark Recognition. La challenge pone come obiettivo, a partire da un'immagine RGB, il riconoscimento di uno o più landmarks, ovvero punti di riferimento. La soluzione proposta si basa su tre passaggi. Un primo step di discrimanizione della presenza o meno di uno o più landmarks nella foto di input. Un successivo passaggio di recupero delle immagini, presenti in un dataset, più simili a quella analizzata. Infine, grazie alla similarità predetta e l'etichetta di ciascuna immagine ottenuta dal passo precedente, vengono definiti i landmarks presenti.

1. Introduzione

Durante gli ultimi anni, Google ha organizzato diverse challenge a cui hanno partecipato numerose persone, una di queste è la competizione *Google Landmark Recognition* [2].
Con la challenge *Google Landmark Recognition*, proposta per diversi anni consecutivi, si è cercato di realizzare un sistema che a partire da immagini RGB sia capace di riconoscere i *landmarks*, che tradotto significa "punti di riferimento". Attraverso il seguente esempio, può essere compreso meglio ciò che si vuole realizzare. Immaginiamo di essere turisti in vacanza e scattiamo una foto ad un edificio in particolare, di cui però non conosciamo il nome; in questa competizione si va ad implementare una Rete Neurale che, prendendo in input un'immagine RGB, predica uno o più punti di riferimento, come ad esempio: la Tour Eiffel, il Colosseo, il Big Ben e molti altri. I landmarks possono essere di diverse tipologie: elementi naturali, come laghi o parchi, o artificiali, come edifici o monumenti.
La soluzione definita in seguito, si basa su alcuni passaggi. Il primo prevede una Rete Neurale Convoluzionale, che permette di classificare le immagini in due categorie: landmarks presente e landmarks non presente. Il passo successivo, che prevede l'analisi delle sole immagini in cui sono presenti uno o più punti di riferimento, prevede l'esecuzione di un metodo di retrieval, ovvero il recupero delle *N* immagini più simili a quella analizzata. Nell'implementazione di questa operazione, è stata utilizzata una Rete Neurale Siamese, che viene sfruttata per l'estrapolazione di un vettore di features da ogni singola immagine e, attraverso un metodo di confronto, un'elaborazione delle *N* immagini più simili a quella analizzata. Infine, grazie al risultato ottenuto dalla fase di retrieval, viene eseguita una classificazione multi-label all'immagine passata in input.

2. Dati

Per lo svolgimento della challenge, è stato definito uno specifico dataset da utilizzare, ovvero *Google Landmarks Dataset v2* [1].
Google Landmarks Dataset v2 è un dataset pubblico di Google, contenente quasi 5 milioni di immagini suddivise in più di 100 mila classi. Il dataset è suddiviso in tre parti: training dataset, index dataset e test dataset.

2.1. Training set

Viste le risorse computazionali e di archiviazione a disposizione, è stato deciso di lavorare solamente su una parte ristretta del dataset. Il sottoinsieme del training set preso in considerazione comprende 66 173 immagini, suddivise in 32 770 classi, le quali sono contenute nei primi 21 file compressi scaricabili attraverso uno script, che può essere trovato sulla pagina ufficiale di *Google Landmarks Dataset v2* [1].
Vista la necessità di utilizzare due reti neurali, a partire dal dataset appena descritto sono state definite due varianti per ciascun modello.

2.1.1 Dataset per Siamese Neural Network

Per quanto riguarda la Siamese Neural Network, è stata ulteriormente diminuita la dimensione del training set, questo poiché per allenare la rete a riconoscere i vari landmark è necessario che per ciascuna classe sia presente almeno una coppia d'immagini. Questo ha portato ad un numero

di classi pari a 13 417 e d'immagini a 46 820. Tuttavia, il numero di occorrenze per ciascuna classe filtrata è molto variabile: sono presenti casi in cui ci sono solamente due immagini a rappresentare la classe o altri casi in cui ce ne sono fino ad una decina. Per cercare di generalizzare il più possibile l'allenamento della Rete Neurale, si sono generate coppie *immagine riferimento - immagine positiva* e *immagine riferimento - immagine negativa* per ciascuna classe ad ogni epoca dell'allenamento. Quindi prendendo una batch di classi, per ciascuna di esse vengono scelte: un'immagine di riferimento casuale, un'immagine positiva casuale (diversa da quella di riferimento) e un'immagine negativa randomica che sia di una classe diversa.

Figure 1. Esempio di coppie di immagini del dataset per la Siamese Neural Network.

Nella fase di allenamento, è stato utilizzato un validation dataset per controllare l'andamento dell'apprendimento. Per generare l'insieme degli elementi di validazione, è stato selezionato il 20% delle classi presenti nel dataset.

2.2. Dataset per CNN per classificazione binaria

Per ciò che riguarda la Rete Neurale utilizzata per discriminare la presenza o l'assenza di landmarks, è stato utilizzato l'intero dataset definito precedentemente; tuttavia, le immagini sono state etichettate, con label 1 o 0, rispetto alla presenza o assenza del landmark nelle training set studiato per la Rete Neurale Siamese.

Figure 2. Esempio di immagini dal dataset per la Rete Neurale per la discriminazione della presenza o assenza di landmarks.

Come nel caso del dataset per la Rete Neurale Siamese, è stato suddiviso il training set in due sottoinsiemi, utilizzando un rapporto 80-20, così da utilizzare il gruppo d'immagini minore per validare il modello della CNN durante l'allenamento.

2.3. Data Augmentation

In entrambi gli specifici dataset, è stata utilizzata la data augmentation, permettendo di ottenere da un'unica immagine più copie differenti tra loro attraverso trasformazioni e modifiche, così da aumentare la cardinalità del training set. Per fare data augmentation sono state utilizzate più specificatamente le seguenti operazioni: regolazione di luminosità, regolazione di contrasto, specchiatura orizzontale, zoom-in.

Figure 3. Esempio di data augmentation.

2.4. Test set

Per quanto riguarda il dataset per il processo di testing, è stato utilizzato quello presente in *Google Landmarks Dataset v2*. Il test set è composto da 117 577 immagini, delle quali però sono state utilizzate solo 29 930 casuali, in modo tale di diminuire il tempo di esecuzione della fase di testing e avere una distribuzione di classi uniformi.

3. Metodo

Come descritto precedentemente, per eseguire il riconoscimento dei punti di riferimento sono state utilizzate due reti neurali: CNN per classificazione binaria e Rete Neurale Siamese.
Come punto di riferimento per l'implementazione della soluzione, è stato scelto il paper del secondo classificato alla competizione *Google Landmark Recognition 2019* [5]. Quello che si è cercato di fare è stato semplificare l'approccio utilizzato, restando in linea con le risorse disponibili, ma allo stesso tempo cercando di seguire il flusso di operazioni da loro definito.
La prima scelta in assoluto da prendere è stata la dimensione dell'immagine di input. Una risoluzione troppo piccola avrebbe fatto perdere molte informazioni, poiché in molti casi i monumenti si differenziano per qualche piccolo dettaglio, ma allo stesso tempo una larghezza ed

un'altezza eccessiva non avrebbero portato benefici considerevoli all'aumento del tempo di allenamento. È stato quindi scelto di mantenere come input un'immagine a tre canali con risoluzione 128x128.

3.1. Convolutional Neural Network for binary classification

L'introduzione di una Rete Neurale Convoluzionale è stata ideata per una iniziale discriminazione delle immagini. Il problema del riconoscimento dei landmarks ammette la possibilità che in un'immagine analizzata ci siano: nessuno, uno o più punti di riferimento. Con questa CNN si va quindi a predirre se un'immagine può contenere almeno un landmark, in modo tale che, in caso non sia presente, non sia necessario eseguire l'operazione di confronto con l'intero training set, portando così ad una diminuzione considerevole del tempo di esecuzione del sistema.
La Neural Network si basa su ResNet50, già allenata su ImageNet. Di ResNet50 sono stati presi in considerazione solo i livelli iniziali; quindi, prendendo in input un'immagine RGB, restituisce un vettore di dimensione 2048, ovvero il max pooling della quinta operazione di convoluzione. L'output restituito dalla Residual Neural Network è passato ad un livello finale di classificazione, composto da una singola unità.
Dell'architettura finale sono allenati solo alcuni livelli; infatti, i livelli iniziali della ResNet50, comprendente le prime tre operazioni convoluzionali, sono stati bloccati. Questo ha portato all'allenamento di 22 086 657 parametri a fronte dei 23 589 761 totali.
Per allenare la rete si è cercato di trovare un compromesso tra velocità di convergenza e generalizzazione, sono quindi stati impostati i seguenti iperparametri: learning rate pari a 10^{-3}, Adam optimizer e dimensione di batch uguale a 32.

3.2. Siamese Neural Network

Solo dopo essere stata etichettata come "landmark presente", un'immagine viene passata allo step successivo, il quale pone come obiettivo il recupero delle 100 immagini del training set più simili a quella di ricerca.
Per svolgere questo task è stata utilizzata una Siamese Neural Network e, come per la rete precedente, ci si è basati su una rete Convoluzionale già addestrata su ImageNet, più nello specifico ResNet152. Esattamente come nel caso della rete Convoluzionale per la classificazione binaria, è stata "troncata" la rete e ottenuto da essa un vettore di output di dimensione pari a 2048. Ciò che viene prodotto da ResNet152 viene successivamente passato ad un fully-connected layer che permette di passare da una dimensione 2048 a 128; questo vettore viene infine utilizzato come spazio di features per la comparazione e la predizione di similirità tra più immagini.
Per il confronto delle due immagini sono stati introdotti due layer. Il primo serve a generare un unico vettore di dimensione 128, ciò che fa è calcolare la differenza di ogni i-esimo valore dei due vettori ed elevarla al quadrato, in modo tale da avere un valore positivo che aumenti con l'aumentare della distanza. Il secondo livello, composto da una singola unità, ha invece il compito di imparare la similarità delle due immagini in base al vettore distanza, fornendo quindi un risultato tendente ad 1 in caso di similarità alta e 0 altrimenti.

Per allenare la rete Siamese è stato scelto come ottimizzatore Adam e come learning rate un valore pari a 10^{-6}. Inoltre, è stata utilizzata la Contrastive Loss [4] come funzione di costo, poiché ideata per modelli basati sulla distanza vettoriale.

3.3. Query expansion

Una volta ottenute le 100 immagini più simili, viene utilizzato un approccio di query expansion. Con l'operazione di query expansion, utilizzando i risultati ottenuti da una prima ricerca, si va a recuperare, per una seconda volta, gli elementi più simili alla query image, ovvero l'immagine di ricerca. È stato dimostrato che attraverso un approccio di query expansion si possono ottenere prestazioni migliori, come riportato in un paper analizzato [3].
Esistono numerosi metodi per fare query expansion e quello più basilare, utilizzato anche in questo contesto, è quello dello sfruttamento della media. Ciò che viene svolto è il recupero iniziale delle 100 immagini del training set più simili alla query image, da queste immagini viene successivamente ottenuto un nuovo vettore di features, avente sempre dimensione 128, che è calcolato come la media aritmetica dei features vectores delle 100 immagini restituite dalla precedente ricerca. Infine, questo nuovo vettore calcolato viene utilizzato per fare un ulteriore recupero, che definirà ulteriori 100 immagini.
Il risultato ottenuto dalla retrieval viene infine utilizzato per la classificazione finale. Per l'assegnazione delle label, vengono tenute in considerazione solo le classi delle immagini con confidenza di similirità maggiori o uguali a 0,9 e attribuite all'etichetta di predizione.

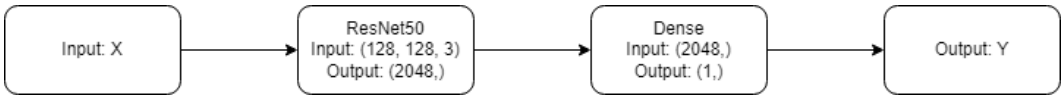

Figure 4. Architettura della Rete Neurale Convoluzionale.

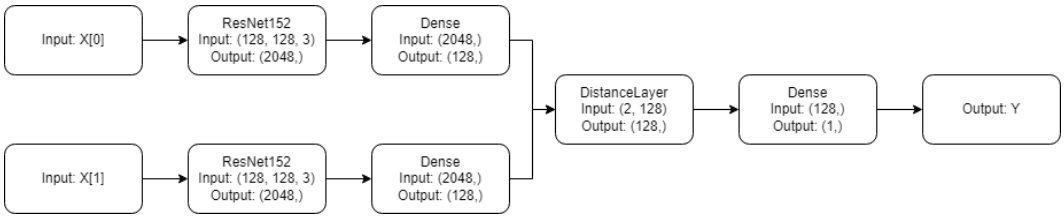

Figure 5. Architettura della Rete Neurale Siamese.

4. Esperimenti

Nella valutazione della soluzione proposta, ci si è inizialmente focalizzati sull'apprendimento di ciascuna rete neurale, cercando per ciascuna i giusti iperparametri.

4.1. Allenamento della binary classification CNN

Nell'allenamento della Rete Neurale Convoluzionale per il riconoscimento della presenza di landmarks, si è cercato fin da subito di trovare iperparametri che ammettessero una valore di bias equilibrato. Seppur si sia riusciti ad ottenere l'obiettivo appena descritto, si è arrivati ad il miglior risultato, tra quelli ottenuti, che presenta un evidente problema. Com'é possibile analizzare nella figura 6, ci si trova in un caso di overfitting. Infatti, dopo la decima epoca, si può notare un rapido incremento di accuratezza e decremento di loss per il training set, quando però il validation set mostra un andamento costante dell'accuratezza, con valore poco più alto di 0,5, e funzione di costo in deciso peggioramento. Seppure i grafici appena riportati ci potessero bastare per comprendere il fatto che la rete non abbia generalizzato sufficientemente, è stato comunque deciso di vedere i risultati sul test set. Sull'insieme di dati di testing si è ottenuto precisione pari a 0,015 73 e recall uguale a 0,662 39. Lo sbilanciamento tra queste due metriche, così evidente, è dovuto al fatto che il numero di elementi del test set in cui sia presente almeno un landmark copre solamente l'1,592% dell'intero insieme.

4.2. Allenamento della Siamese Neural Network

Per ciò che concerne l'allenamento della Rete Neurale Siamese, si è andati a riscontrare, sotto qualche aspetto, un caso simile alla CNN precedente. Com'é possibile analizzare dalla figura 8, l'andamento dell'acuratezza durante l'allenamento è divergente per i due insiemi di dati, ovvero il training set e validation set. Ci si trova anche in questo caso in una situazione di overfitting. Allo stesso tempo, però, è possibile notare un valore di accuratezza abbastanza basso, nell'intorno di 0,7 per quanto riguarda il training set. Ciò può essere dovuto da due possibili problemi: alto bias o learning rate troppo basso. A seconda del caso, si può risolvere ciascun problema con un approccio specifico. Ad esempio, in caso di apprendimento lento, può essere aumentato il tempo di allenamento, quindi incrementando il numero di epoche, o può essere aumentato il learning rate. Come per la CNN, è stato deciso di valutare le metriche di precisione e recall singolarmente sulla Siamese Neural Network basandosi sul test set. Le immagini del set di testing, nel caso fossero senza landmarks, sono state comparate ad un elemento casuale del training set, mentre, nel caso contrario, sono state confrontate ad un immagine del training set con lo stesso punto d'interesse. I risultati ottenuti per precisione e recall sono stati rispettivamente 0,008 19 e 0,279 62.

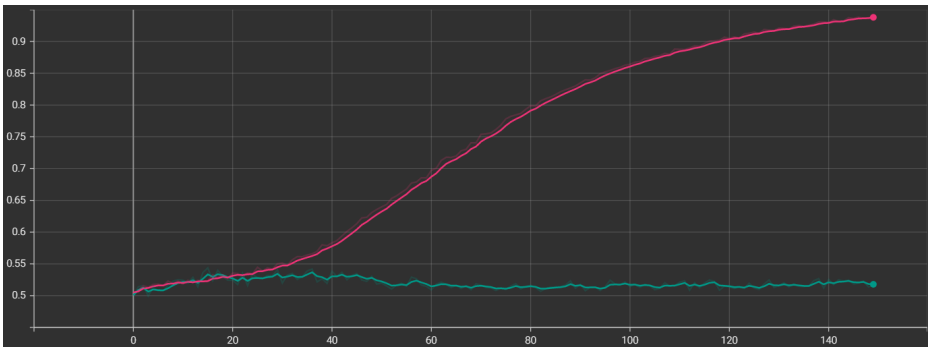

Figure 6. Andamento dell'accuratezza durante l'allenamento della CNN per il riconoscimento della presenza di landmarks. In rosso il training set e in verde il validation set.

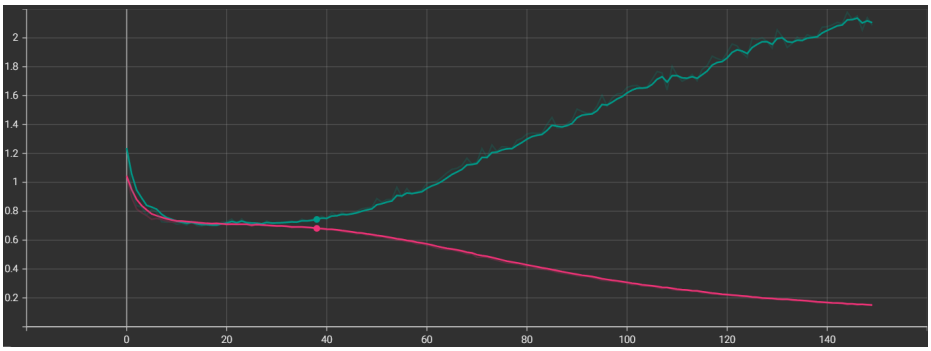

Figure 7. Andamento della loss durante l'allenamento della CNN per il riconoscimento della presenza di landmarks. In rosso il training set e in verde il validation set.

4.3. Test globale

Infine, è stato valutato il modello proposto nel complessivo.
Un aspetto valutato inizialmente è stato l'inserimento della Rete Neurale per il riconoscimento della presenza di landmarks, poiché, per definire il fatto che non sia presente alcun punto d'interesse, basterebbe il fatto che l'operazione di recupero non ammetta alcuna immagine del training set con similarità sopra alla soglia definita. È stato comparato il sistema con e senza l'ausilio della Rete Neurale per la classificazione, valutando così: tempo di esecuzione e precisione delle predizioni. Il modello con la CNN ha impiegato 332,75 secondi per la classificazione del test set in "landmarks presente" e "landmarks assente" e 46,375 secondi per il recupero delle 100 immagini più simili nel training set, per un totale di 379,125 secondi. Mentre, senza l'utilizzo della CNN iniziale, è stato ottenuto un tempo di esecuzione complessivo di 7 975,232 secondi. Si è quindi ottenuto un frazionamento del tempo di 21 volte, grazie all'utilizzo della binary classification CNN. Tuttavia, è necessario considerare anche le perfomance del modello con e senza la Rete Neurale per il riconoscimento della presenza di landmarks.
In entrambi i casi, sono state calcolate precision, recall e Global Average Precision (GAP), ma per tutti i dati si è ottenuto un valore pari a 0. Ciò è dovuto alla bassa precisione della Rete Neurale Siamese, che definisce una confidenza di similarità molto alta per immagini con landmarks diversi. Infatti, ordinando i risultati ottenuti dal confronto della query image con tutti gli elementi del training set in base alla confidenza di similarità, è stato ottenuto un posizionamento medio in classifica, prima e dopo query expansion, rispettivamente di 15 687 e 16 994.

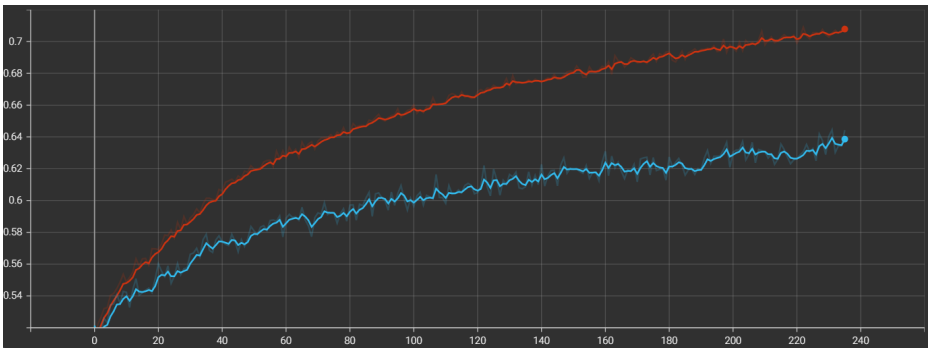

Figure 8. Andamento dell'accuratezza durante l'allenamento della Siamese Neural Network. In rosso il training set e in azzurro il validation set.

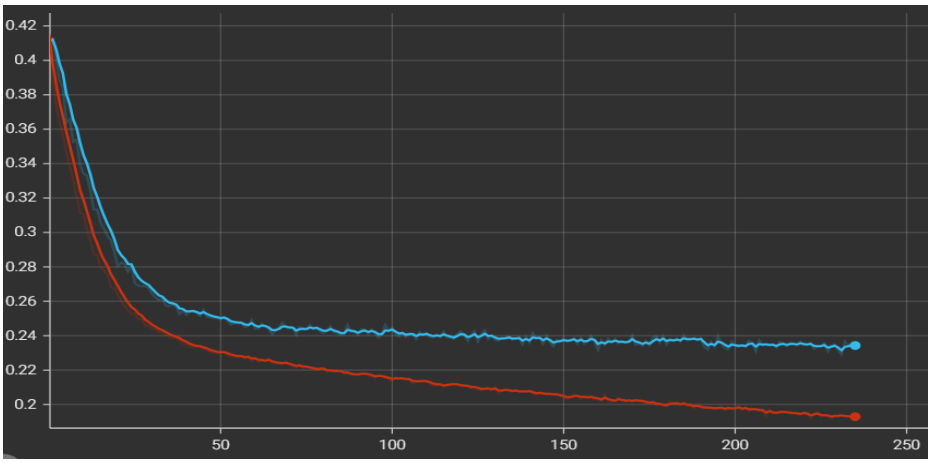

Figure 9. Andamento della loss function durante l'allenamento della Siamese Neural Network. In rosso il training set e in azzurro il validation set.

5. Conclusioni

Com'é stato possibile analizzare nel capitolo 4, la soluzione proposta non è in grado di ritornare risultati consistenti. Il motivo di ciò può essere dovuto ad alcuni fattori. Primo fra tutti: l'overfitting; è necessario allenare le reti neurali sull'intero dataset, ma, per fare ciò, è sicuramente necessario un sistema di elaborazione sufficientemente potente.
Un secondo fattore è la complessità di *Google Landmarks Dataset v2*, poiché presenta un numero considerevole di classi con poche immagini per ciascuna di esse; infatti, molte soluzioni proposte dai partecipanti alla challenge definiscono un modello di Rete Neurale molto complesso. Prendendo come esempio il modello del secondo classificato dell'edizione del 2019 [5], si potrebbe inizialmente realizzare una Neural Network che ritorni delle regioni d'interesse classificate sulla tipologia di landmark trovato (ad esempio edificio, lago, chiesa, ecc.). Successivamente, per ciascuna di queste RoIs, applicare un metodo di retrieval ad hoc per il landmark classificato, così da recuperare le N immagini più simili.
Un ultimo aspetto a cui possono essere apportate migliorie, è la fase di query expansion, che, come visto dagli esperimenti, ha peggiorato le performance del sistema. Infatti, molto spesso la query expansion, legata alla retrieval image-based, viene implementata sfruttando metriche di image processing, come istogrammi di tonalità o luminosità. Inoltre, il calcolo della nuova query, viene spesso elaborato come media ponderata delle immagini ottenute, calcolando i pesi in base alla confidenzialità e al posizionamento in classifica.

References

[1] Github - google landmarks dataset v2. URL: `https://github.com/cvdfoundation/google-landmark`.

[2] Kaggle - google landmark recognition 2021. URL: `https://www.kaggle.com/c/landmark-recognition-2021`.

[3] Query expansion based on top-ranked images for content-based medical image retrieval. URL: https://ieeexplore.ieee.org/abstract/document/9239268.

[4] Maksym Bekuzarov. Losses explained: Contrastive loss. URL: https://medium.com/@maksym.bekuzarov/losses-explained-contrastive-loss-f8f57fe32246.

[5] Yuning Du Xianglong Meng Hui Ren Kaibing Chen, Cheng Cui. 2nd place and 2nd place solution to kaggle landmark recognition and retrieval competition 2019. URL: https://arxiv.org/pdf/1906.03990v2.pdf.