Università degli studi di Padova

Dipartimento di matematica

Corso di laurea in Informatica

Riconoscimento e tracciamento di elementi su video ad alta risoluzione

Laureando:
Davide Liu

Relatore:
Prof. Lamberto Ballan

Tutor aziendale:
Leonardo Dal Zovo

Anno Accademico 2018/2019

Indice

Elenco delle tabelle

Elenco delle figure

1 Introduzione

La computer vision è un ambito dell'intelligenza artificiale il cui scopo è quello di insegnare alle macchine non solo a vedere un' immagine, ma anche a riconoscere gli elementi che la compongono in modo da poter interpretare il suo contenuto come farebbe il cervello di un qualsiasi essere vivente. Nonostante le attuali tecniche di deep learning rendano possibile questo compito, è comunque necessaria una grande quantità di immagini e di tempo per poter allenare una rete neurale a sufficienza in modo da riuscire a riconoscere correttamente degli oggetti in un' immagine non incontrata durante il processo di allenamento.

Lo scopo del progetto di stage è stato quello di progettare e realizzare un sistema di riconoscimento e tracciamento di specifici elementi all' interno di un video ad alta risoluzione. Questo progetto ha comportato sfide e complessità aggiuntive rispetto all' analisi degli elementi presenti in una singola foto sia per il fatto che un video è composto da una sequenza di frames anzichè da una singola immagine, sia per il fatto che i frames trattati erano in formato Ultra High Definition (4K) e quindi elaborare l'intero frame in una solo volta sarebbe stato troppo oneroso dal punto di vista computazionale.

Riassumendo i tre problemi principali che sono stati affrontati, in ordine sequenziale, sono i seguenti:

- Frammentazione dell'immagine;
- Tracciamento degli elementi;
- Stabilizzazione delle label degli elementi tracciati.

Ognuno dei sotto-problemi è stato analizzato a fondo, ne è stata trovata un' adeguata soluzione ed è stata applicata ai fini di realizzare un prodotto il più performante possibile.

2 Ambiente Aziendale

Figura 1: Logo di Studiomapp

STUDIOMAPP, fondata a fine 2015, è una startup innovativa con sedi a Ravenna e Roma, in Italia. Sviluppa algoritmi di intelligenza artificiale specifici per Geo-calcolo e dati geo-spaziali in modo da fornire soluzioni innovative per smart cities, mobilità, trasporti e logistica, turismo e beni culturali, immobiliare e real estate, agricoltura, territorio e gestione delle risorse naturali, adattamento ai cambiamenti climatici.

E' la prima startup dell'Emilia Romagna selezionata dall'ESA BIC Lazio, l'incubatore dell'Agenzia Spaziale Europea (ESA) ed è supportata da importanti istituzioni, acceleratori e grandi attori. Membro fondatore dal 2016 della rete Copernicus Academy, si impegna a diffondere i benefici dell'utilizzo dei dati di Osservazione della Terra per la qualità della vita dei cittadini e la competitività delle PMI.

Ha acquisito una solida esperienza nella promozione di Copernicus, il programma europeo di osservazione della Terra, nell'ecosistema Startup condividendo conoscenza e formazione. Ad oggi la società ha organizzato più di 40 eventi che hanno raggiunto più di 1000 persone che vanno dal pubblico generale, agli studenti, alle startup, ai funzionari pubblici, alle autorità locali, alle PMI.

3 Analisi dei problemi

3.1 Tecniche di computer vision

Per riconoscere un singolo elemento in un'immagine viene tipicamente utilizzata una Convolutional Neural Networks (CNN) allenata con grandi quantità di immagini che possono essere tranquillamente reperite in rete già raggruppate in datasets come ad esempio ImageNet. La vera sfida salta fuori non appena ci troviamo a dover identificare nella stessa immagine diversi oggetti appartenenti a categorie diverse, di differenti dimensioni e posizioni e talvolta anche sovrapposti. Questa situazione è molto comune quando osserviamo qualsiasi foto rappresentante il mondo reale. Il risultato ottimale sarebbe quindi di avere una label di dimensioni corrette per ogni oggetto identificato mostrando anche la categoria di appartenenza dell'elemento e la probabilità che la classificazione sia effettivamente quella corretta.

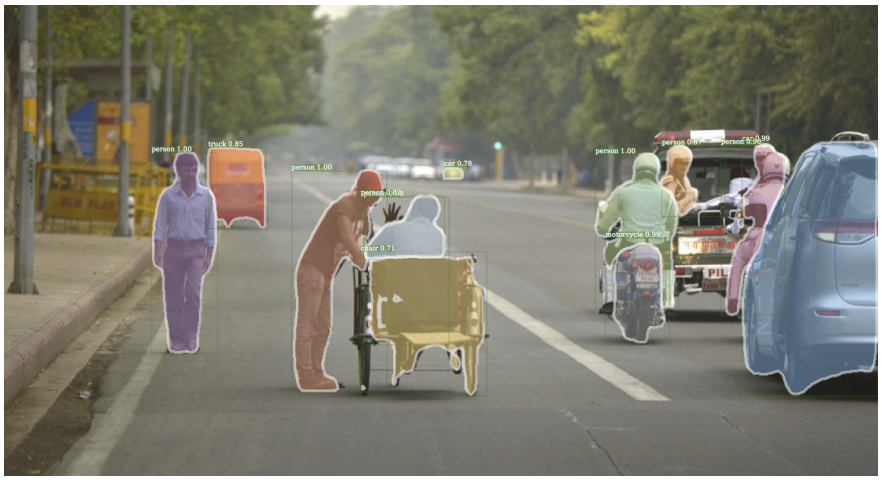

Figura 2: Esempio di un'immagine con label, categoria e probabilità per ogni elemento riconosciuto in essa

Il modo più semplice per andare incontro a questo problema è quello di utilizzare una R-CNN (Regional Convolutional Neural Network) che tramite un algoritmo greedy chiamato Selective Search estrae delle regioni di interesse nelle quali è probabile che sia presente un oggetto. Queste regioni vengono poi date singolarmente in input ad una normale CNN la quale ha il compito di estrarne le caratteristiche principali e poi utilizzare una Support Vector Machine (SVM) presente nell'ultimo strato della CNN per rilevare la presenza di un oggetto ed eventualmente classificarlo. Questo tipo di soluzione presenta il difetto di

richiedere molto tempo durante l'operazione di ricerca delle regioni di interesse. La versione più avanzata della R-CNN ed attualmente in uso nei sistemi di computer vision attuali è chiamata Faster R-CNN e risolve il bottleneck della sua antecedente sostituendo la Selective Research con una Region Proposal Network (RPN). Con una CNN viene prima costruita la mappa delle caratteristiche più significative dell'immagine, in seguito vengono passate queste caratteristiche ad una RPN la quale ritorna delle regioni associate ad un probabilità che esse contengano un oggetto. Infine queste regioni vengono passate ad una R-CNN che avrà come al solito il compito di riconoscere e classificare l'oggetto.

3.2 Computer vision applicata su immagini ad alta risoluzione

Sebbene sul web sia abbastanza facile reperire grandi quantità di immagini con cui allenare i propri modelli, però la maggior parte di questi datasets contiene solamente immagini a bassa risoluzione ed è difficile trovare in rete grandi datasets di immagini o video in 4K. Inoltre, la maggior parte dei modelli sono stati progettati per lavorare su immagini a bassa risoluzione (tra 200 e 300 pixels) sia per il fatto che una bassa risoluzione è comunque sufficiente per riconoscere e classificare un elemento, sia perchè è più efficiente lavorare su immagini di bassa qualità che su immagini in 4K.
Lo svantaggio che questo comporta è che nelle immagini in bassa risoluzione si perdono molti dei dettagli che invece potrebbero essere catturati da un immagine o video ad alta risoluzione. In aggiunta, i video in 4K o addirittura 8K sono sempre più diffusi al giorno d'oggi perciò anche gli attuali modelli dovranno prima o poi adattarsi per trattare efficacemente immagini di tali risoluzioni. In figura si può vedere un esempio di un frame in alta risoluzione, diminuendone le dimensioni, perdendo quindi qualità, non sarebbe stato possibile riconoscere alcune delle persone individuate nel frame.

Figura 3: Esempio di un frame di un video in 4K

3.3 Frammentazione dell'immagine

Un'immagine in 4K ha una risoluzione di 3840 x 2160 pixels con un totale di 8294400 di pixels. Una rete neurale in grado di ricevere un input così corposo dovrebbe allenare un numero molto elevato di parametri risultando così in un processo molto lungo e dispendioso. L'idea è quella di frammentare l'immagine in diverse sotto-immagini di dimensioni minori e quindi gestibili efficacemente da una singola rete neurale. (FIGURA) Tuttavia questo procedimento non è esente da difficoltà, nel caso in cui un oggetto dovesse trovarsi su più regioni diverse esso potrebbe venire identificato più volte ed essere riconosciuto ogni volta come se fosse un oggetto diverso.
Un primo approccio per risolvere questo problema è stato quello di porre maggiore attenzione alle labels degli oggetti posizionate in prossimità dei confini delle regioni in quanto con molta probabilità è possibile che l'oggetto continui invece nella regione adiacente. Se quindi presente una label anche nella regione adiacente allora si è passati al controllo che le labels possano effettivamente appartenere allo stesso elemento. Per fare ciò è stato sufficiente assicurarsi che le due labels combaciassero entro una certa soglia ed in caso affermativo fonderle in una sola label che contenesse l'oggetto intero.
Un' altra soluzione esaminata è stata quella di suddividere l'immagine intera in regioni con sovrapposizione. In questo modo un elemento giacente a ridosso tra una o più regioni avrebbe generato due o più labels sovrapposte. Tramite un algoritmo di Non-Maximum Suppression, per ogni insieme di labels parzialmente sovrapposte, sono state eliminate le labels con probabilità minore tenendo valida solo quella con probabilità massima.

4 Progettazione

4.1 Algoritmo di frammentazione dell'immagine

Per quanto riguarda il problema della frammentazione dei frames in 4K è stato individuato un apposito algoritmo consistente di diversi steps.

4.1.1 Scomposizione del frame originale in regioni

Il frame in 4K viene scomposto in una matrice di n x m sotto-immagini chiamate regioni in modo tale che ogni regione sia efficacemente analizzabile da una Faster R-CNN. Per facilitare la detection da parte della rete ogni regione si sovrappone leggermente con le sue regioni adiacenti. Per definire la quantità di pixels da coinvolgere nella sovrapposizione viene definito uno $\textbf{stride}_{\textbf{G}}$ che indica quanti pixels della regione tralasciare, sia in verticale che in orizzontale, prima che cominci la regione successiva, ovviamente lo stride deve essere minore della larghezza di una regione. Una Faster R-CNN dopo aver elaborato singolarmente ogni regione darà in output una lista di labels con le seguenti caratteristiche:

- **(min-x, min-y)**: coordinate del vertice in basso a destra del rettangolo rappresentante la label dell'oggetto riconosciuto;
- **(max-x, max-y)**: coordinate del vertice in alto a sinistra del rettangolo rappresentante la label dell'oggetto riconosciuto;
- **Classe**: è un numero naturale che indica la categoria di appartenenza dell'elemento individuato;
- $\textbf{Score}_{\textbf{G}}$: rappresenta la misura di probabilità che la classificazione sia effettivamente quella corretta.

Successivamente viene aggiustata la posizione delle labels individuate in modo da traslarle nella loro posizione corretta all'interno dell'immagine originale non frammentata. Questo viene fatto aggiungendo un adeguato offset alle coordinate dei vertici dei box delle labels sulla base della loro regione di appartenenza. (IMMAGINE)

4.1.2 Creazione di raggruppamenti di labels correlate

A questo punto tutti gli elementi sono stati individuati e classificati ma rimane comunque il problema che a causa della precedente scomposizione gli oggetti situati in prossimità o all'interno delle aree di scomposizione risulterebbero individuati due o più volte, questo varia

in base al numero di regione sulle quali giace l'oggetto figX. Il secondo problema è che due elementi "vicini" , anche se classificati allo stesso modo, non è detto che necessariamente debbano rappresentare lo stesso elemento. Un esempio di questo caso lo si può notare in figX. Un caso ancora peggiore è quello mostrato in figX dove non solo l'elemento è situato su più regioni differenti ma sussiste anche il problema che ogni parte dell'elemento verrebbe classificata in modo diverso a causa della loro ambiguità. Infine, l'ultima figura (figX) mostra un oggetto che si distribuisce su molte regioni ed ogni sua label presenta dimensioni diverse. La soluzione individuata consiste nel raggruppare label correlate tra loro in insiemi di labels dette raggruppamenti$_{\mathbf{G}}$. Inizialmente vengono individuate tutte le labels situate in prossimità dei confini di regioni ovvero quelle labels che intersecano le aree di sovrapposizioni o che non distino più di un fissato numero di pixels, detto $_{\mathbf{G}}$ tolleranza, da esse. Per creare i raggruppamenti di labels è stato ideato il seguente algoritmo:

1. Vengono tenute solo le labels vicino ad un confine di regione;
2. Seleziona una label libera e la fa diventare controllata;
3. Per ogni label controllata controlla se ci sono altre labels che la intersecano o che siano distanti entro la tolleranza fissata e che rispettino una **condizione di verità**;
4. Le labels così trovate diventano a loro volta controllate;
5. Si ripetono i punti 3 e 4 fino a che non sia più possibile trovare ulteriori labels;
6. Tutte le label controllate vengono ora classificate come raggruppate e viene assegnato un numero progressivo ad ogni label controllata in modo da identificarne il gruppo di appartenenza;
7. Si ripetono i punti da 2 a 6 fino a che tutte le labels non vengano raggruppate.

Condizione di verità: Per effettuare un corretto raggruppamento delle labels viene anche tenuta in considerazione l'etichetta a loro associata tramite la classificazione e lo score assegnato. Per definire il risultato della condizione è inoltre necessario stabilire una soglia$_{\mathbf{G}}$ di probabilità per considerare un'etichetta come affidabile o meno. Di seguito vengono riportati i vari casi per decidere se la condizione è vera o falsa.

- **True**: Le due label hanno la stessa etichetta ed entrambe con score uguale o maggiore della soglia;
- **True**: Le due label hanno la stessa etichetta ma almeno una delle due ha score minore della soglia;

- **True**: Le due label hanno la etichetta diversa ma almeno una delle due ha score minore della soglia;
- **False**: Le due label hanno la etichetta diversa ed entrambe con score uguale o maggiore della soglia;

In seguito bisogna trasformare ogni raggruppamento in una nuova label che ne racchiuda tutte le labels. Per fare questo vengono esaminate le coordinate di ogni vertice di tutte le labels di un raggruppamento in modo tale da trovare quattro nuovi vertici di un rettangolo che soddisfi i requisiti sopra discussi. Il nuovo box così creato andrà a sostituire le labels del rispettivo raggruppamento e come etichetta verrà tenuta l'etichetta posseduta dalla label con score maggiore. A questo punto l'algoritmo può dirsi concluso ed è in grado di riconoscere gli elementi in un'immagine in 4K con un'accuratezza accettabile e buona velocità. Tuttavia in casi particolari come quello mostrato in figura figX l'algoritmo commetterebbe un errore in quanto individuerebbe due individui con una sola label.

4.1.3 Raggruppamento di labels come region proposal

5 Tecnologie

6 Sviluppo

7 Risultati ottenuti

8 Glossario

9 Bibliografia

Fonti: Sito aziendale https://www.studiomapp.com/en/ Tecniche di computer vision, frammentation https://users.ece.cmu.edu/~franzf/papers/hpec_2018_vr.pdf https://www.mpi-inf.mpg.de/fileadmin/inf/d2/HLCV/cv-ss13-0605-sliding-window.pdf

10 Appendice