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.

1.1 Note esplicative

Allo scopo di evitare ambiguità a lettori esterni al gruppo, si specifica che all'interno del documento verranno inseriti dei termini con un carattere 'G' come pedice, questo significa che il significato inteso in quella situazione è stato inserito nel Glossario

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 ci troviamo ad osservare 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.

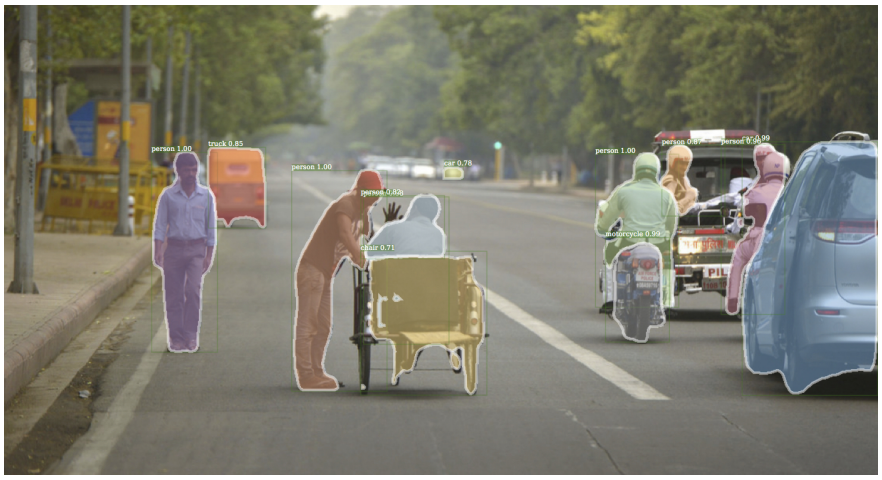

Figura 2: Esempio di un'immagine con box, categoria e probabilità per ogni elemento riconosciuto in essa

Il modo più semplice per andare incontro a questo problema è quello di utilizzare una R-CNN (Regional Convolutional Neural Network) la quale tramite un algoritmo greedy chiamato Selective Search estrae delle regioni di interesse nelle quali è probabile che vi 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 per 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). Essa prende come input un'immagine di qualsiasi dimensione e restituisce come output un insieme di rettangoli associati ad un probabilità che esse contengano un oggetto o meno. Tramite una CNN viene prima costruita la mappa delle caratteristiche più significative dell'immagine, in seguito viene utilizzata una sliding window per scorrere la mappa delle caratteristiche e darle in input a due fully-connected layers dei quali uno serve per individuare le coordinate del box dell'oggetto[1] mentre l'altro serve per ritornare la probabilità che nel box vi sia effettivamente un oggetto[2]. Infine queste regioni vengono passate ad una Fast 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, tuttavia 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 600 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 da un video ad alta risoluzione. In aggiunta, i video in 4K o addirittura 8K sono al giorno d'oggi sempre più diffusi 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 nel quale, diminuendone le dimensioni e perdendo quindi qualità, non sarebbe stato possibile riconoscere alcune delle persone individuate nel frame.

[1] Box-regression layer
[2] Box-classification layer

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 per 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 tanto che molti dei modelli attuali effettuano un ridimensionamento dell'immagine per adattarla al meglio al loro input. L'idea per aggirare il problema 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 o addirittura essere riconosciuto ogni volta come se fosse un oggetto appartenente ad una categoria diversa.
Un primo approccio per risolvere questo problema è quello di porre maggiore attenzione alle labels degli oggetti posizionati in prossimità dei confini delle regioni in quanto con molta probabilità è possibile che l'oggetto continui invece nella regione adiacente piuttosto che essere interamente contenuto nella regione esaminata. Se è quindi presente una label anche in una regione adiacente si passa allora alla verifica che le due labels possano effettivamente appartenere allo stesso elemento. Per fare ciò bisogna assicurarsi che le due labels siano compatibili sia in termini di categoria che di posizione entro una certa soglia ed in caso affermativo fonderle in una sola label che contenente l'oggetto intero.
Un' altra soluzione esaminata è quella di suddividere l'immagine intera in regioni con sovrap-

posizione. In questo modo un elemento giacente a ridosso tra una o più regioni genererebbe due o più labels sovrapposte. Tramite un algoritmo di Non-Maximum Suppression, per ogni insieme di labels parzialmente sovrapposte, si possono eliminare le labels con probabilità minore tenendo valida solo quella con probabilità massima.

4 Progettazione

4.1 Algoritmo per la frammentazione dell'immagine

Per quanto riguarda il problema della frammentazione dei frames in 4K è stato individuato un apposito algoritmo in grado di effettuare il riconoscimento degli oggetti in nei frames presenti senza doverli ridimensionare ma utilizzando invece una tecnica conosciuta come frammentazione dell'immagine.

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$_{\mathbf{G}}$ in modo tale che ogni regione sia efficacemente analizzabile da una Faster R-CNN. Per facilitare l'operazione di 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 stride$_{\mathbf{G}}$ che indica quanti pixels della regione tralasciare, sia in verticale che in orizzontale, prima che cominci quella 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$_{\mathbf{G}}$ 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$_{\mathbf{G}}$ di appartenenza dell'elemento individuato;
- **Score$_{\mathbf{G}}$** : 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.

4.1.2 Rimozione degli elementi individuati più volte all'interno delle aree di sovrapposizione

A causa della presenza delle aree di sovrapposizione dovute alla struttura delle regioni, gli elementi giacenti in queste particolari zone del frame verranno individuati tante volte quante sono le regioni che si sovrappongono in quella determinata zona. Per eliminare le copie duplicate e tenerne solo una viene utilizzato un algoritmo chiamato Average Non-Max Suppression (ANMS) che è una variante del Non-Max Suppression tipicamente utilizzato dalle reti di tipo RCNN e le sue varianti. Invece che tenere la label con lo score maggiore ed eliminare tutte le altre, come box viene usato il box che deriva dalla media dei box di tutte labels e lo score viene calcolato come la media delle loro scores. Questo metodo è fondato sul ragionamento che non bisognerebbe buttare via delle informazioni già possedute ma piuttosto riutilizzarle per scoprire qualcosa di nuovo. Ad uno stesso elemento visualizzato dentro due sezioni differenti di un' immagine potrebbero venirgli assegnati due score diversi. Mentre NMS conserverebbe solo il valore più alto tra i due, ANMS li utilizzerebbe entrambi per ottenere un valore ancora più affidabile.

4.1.3 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 numero varia in base al numero di regioni sulle quali giace l'oggetto come riportato in figX. Il secondo problema è che due elementi vicini [3], anche se classificati nella stessa categoria, 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, la 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 sovrapposizione 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 e sono considerate come *libere*;

[3] Due label sono considerate vicine se appartengono a due regioni confinanti e sono situate dentro o in prossimità di un'area di sovrapposizione

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 la categoria a loro associata tramite la classificazione insieme allo score assegnato. Per definire il risultato della condizione è inoltre necessario stabilire una $\text{soglia}_{\mathbf{G}}$ 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 labels hanno la stessa categoria ed entrambe con score uguale o maggiore della soglia;

- **True**: Le due labels hanno la stessa categoria ma almeno una delle due ha score minore della soglia;

- **True**: Le due labels hanno categoria diversa ma almeno una delle due ha score minore della soglia;

- **False**: Le due labels hanno categoria diversa ed entrambe con score uguale o maggiore della soglia;

E' da notare che labels intersecanti ma nella stessa regione non sono motivo di interesse in quanto l'algoritmo di Non-Maximum Suppression utilizzato dalla rete in fase di post-processing ci assicura che labels intersecanti individuino elementi diversi. In seguito bisogna trasformare ogni raggruppamento in una nuova label che racchiuda tutte le labels che lo compongono. 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.

4.1.4 Raggruppamenti di labels utilizzati come region proposal

Un ulteriore miglioramento dell'algoritmo viene ottenuto utilizzando le labels ottenute dal procedimento descritto in precedenza come nuove regioni sulle quali applicare nuovamente Faster R-CNN per identificare nuovamente gli elementi contenuti nella regione ma con maggiore precisione in quanto questa volta l'area non verrà affetta da problemi di frammentazione dando quindi la possibilità alla rete di esaminare l'oggetto per intero. La regione viene prima inizializzata rimuovendo la sua label e poi ripopolata dalle nuove labels identificate dalla rete. Il primo problema che salta fuori è che durante questo procedimento la rete identificherà nuovamente anche quegli elementi che casualmente si trovavano dentro la regione coinvolta ma che erano già stati trovati anche in precedenza. Tuttavia questo problema viene tranquillamente risolto applicando un algoritmo di Average Non-Max Suppression, utilizzato già in precedenza, per eliminare oggetti quasi completamente sovrapposti. Il secondo problema riguarda ancora gli oggetti che stanno a cavallo tra la regione interessata e l'immagine originale, questa volta però, avendoli già individuati nella loro interezza durante la prima fase è quindi solamente necessario integrare la nuova label con quella già trovata in precedenza. Un caso particolare lo si ha quando la la regione in esame risulti essere così estesa da vanificare i vantaggi ottenuti dalla frammentazione. Per far fronte a questo problema basta ridimensionare l'area coinvolta fino a portarla ad avere dimensioni gestibili da una rete. In questo caso la perdita di risoluzione e quindi di dettagli non comporterebbe un grave problema in quanto gli elementi visibili solo grazie all'alta definizione sono già stati individuati nella fase precedente. Nel caso in cui dovessero venire nuovamente identificati verrebbero gestiti dall'ANMS per ottenerne una migliore approssimazione. Questa funzionalità permette di migliorare l'accuratezza quando si vogliono identificare oggetti che si estendono su due o più regioni o per migliorare la detection di gruppi di elementi molto vicini tra loro ed in prossimità di un confine.

5 Tecnologie

5.1 Python

5.2 Tensorflow

5.3 OpenCV

5.4 Numpy

6 Sviluppo

7 Risultati ottenuti

8 Glossario

B

Box E' un rettangolo che identifica il perimetro entro quale l'oggetto riconosciuto si trova.

C

Categoria L'obiettivo della classificazione è quello di assegnare all'oggetto individuato la sua categoria di appartenenza.

L

Label Etichetta che viene data ad ogni elemento riconosciuto e ne indica la categoria di appartenenza, una detection box ed uno score.

R

Raggruppamento Insieme di labels correlate tra loro tale che da una qualsiasi label del gruppo sia possibile raggiungere qualsiasi altra label dello stesso insieme passando solo per label vicine [4].

Regione Una sezione di un'immagine con un'area di dimensione minore dell'area dell'immagine originale.

S

Score La probabilità che la categoria associata all'elemento riconosciuto sia quella corretta.

Soglia E' il valore che lo score di una label deve eguagliare o superare per essere considerata affidabile.

[4] Due label sono considerate vicine se appartengono a due regioni confinanti e sono situate dentro o in prossimità di un'area di sovrapposizione

Stride

T

Tolleranza La distanza in pixels per la quale una label può discostare da una zona di interesse per cui sia ancora considerata essere intersecata con la zona di interesse.

9 Bibliografia

Riferimenti bibliografici

[1] Sito web dell'azienda Studiomapp.
`https://www.studiomapp.com/en/`

[2] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*. University of Toronto.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation*. UC Berkeley.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. Microsoft Research.

[5] Vıt Ruzicka, Franz Franchetti. *Fast and accurate object detection in high resolution 4K and 8K video using GPUs*. Carnegie Mellon University.

[6] Imagenet database.
`http://www.image-net.org/`

10 Appendice