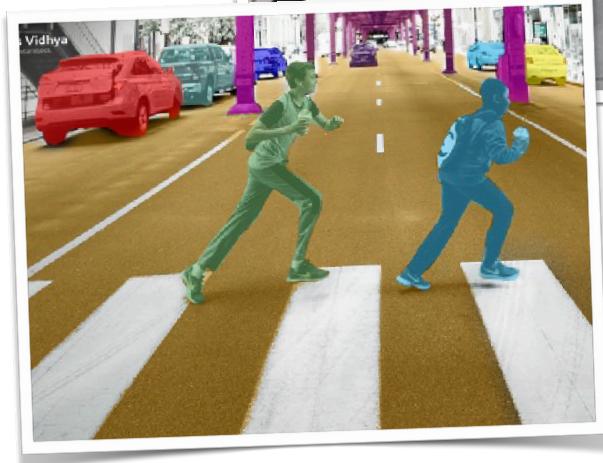


La Segmentazione



Cos'è la segmentazione delle immagini ?

La segmentazione delle immagini è una tecnica della computer vision che suddivide un'immagine digitale in sezioni o gruppi di pixel per semplificare il rilevamento degli oggetti e altre attività correlate. Questa metodologia facilita l'analisi dei dati visivi complessi suddividendo un'immagine in segmenti strutturati, consentendo un'elaborazione più efficiente e accurata.

I metodi tradizionali di segmentazione utilizzano algoritmi che analizzano caratteristiche visive di base, come colore e luminosità, per individuare confini tra oggetti e regioni di sfondo. Con l'avvento del machine learning e del deep learning, tecniche più avanzate si basano su modelli addestrati con dataset annotati per distinguere con precisione specifici tipi di oggetti e regioni presenti in un'immagine.

Essendo una tecnica estremamente flessibile, la segmentazione delle immagini trova applicazioni in molti settori, tra cui l'imaging medico per supportare la diagnosi, la robotica e le auto autonome per navigare l'ambiente, e l'analisi di immagini satellitari per identificare elementi di interesse.

Segmentazione Semantica

La segmentazione semantica è il tipo più semplice di segmentazione delle immagini. Un modello di segmentazione semantica assegna una classe semantica a ogni pixel, ma non genera altri contesti o informazioni (come gli oggetti).

La segmentazione semantica tratta tutti i pixel come "elementi" ; non fa differenza tra "elementi" e "cose".

Ad esempio, un modello di segmentazione semantica addestrato per identificare determinate classi in una strada cittadina produrrebbe maschere di segmentazione che indicano i confini e i contorni per ciascuna classe rilevante di "cose" (come veicoli o pali della luce) ed "elementi" (come strade e marciapiedi), ma non creerebbe alcuna distinzione tra più istanze della stessa classe. Ad esempio, le auto parcheggiate una di fronte all'altra potrebbero essere trattate semplicemente come un unico, lungo segmento di "auto".



(a) Source Image



(b) Semantic Segmentation

Segmentazione delle istanze

La segmentazione delle istanze inverte le priorità della segmentazione semantica: mentre gli algoritmi di segmentazione semantica prevedono solo la classificazione semantica di ciascun pixel (senza considerare le singole istanze), la segmentazione delle istanze delinea la forma esatta di ogni istanza di oggetto separata.

La segmentazione delle istanze isola le "cose" dagli "elementi", che ignora, produce una maschera di segmentazione precisa invece di un riquadro di delimitazione approssimativo.

È un compito più difficile della segmentazione semantica: anche quando elementi della stessa classe si toccano o addirittura si sovrappongono, i modelli di segmentazione delle istanze devono essere in grado di separare e determinare la forma di ciascuno di essi, mentre i modelli di segmentazione semantica possono semplicemente raggrupparli insieme. Consideriamo, ad esempio, come i due diversi modelli trattano le auto parcheggiate in questa immagine di una strada di città.



(a) Source Image



(b) Semantic Segmentation

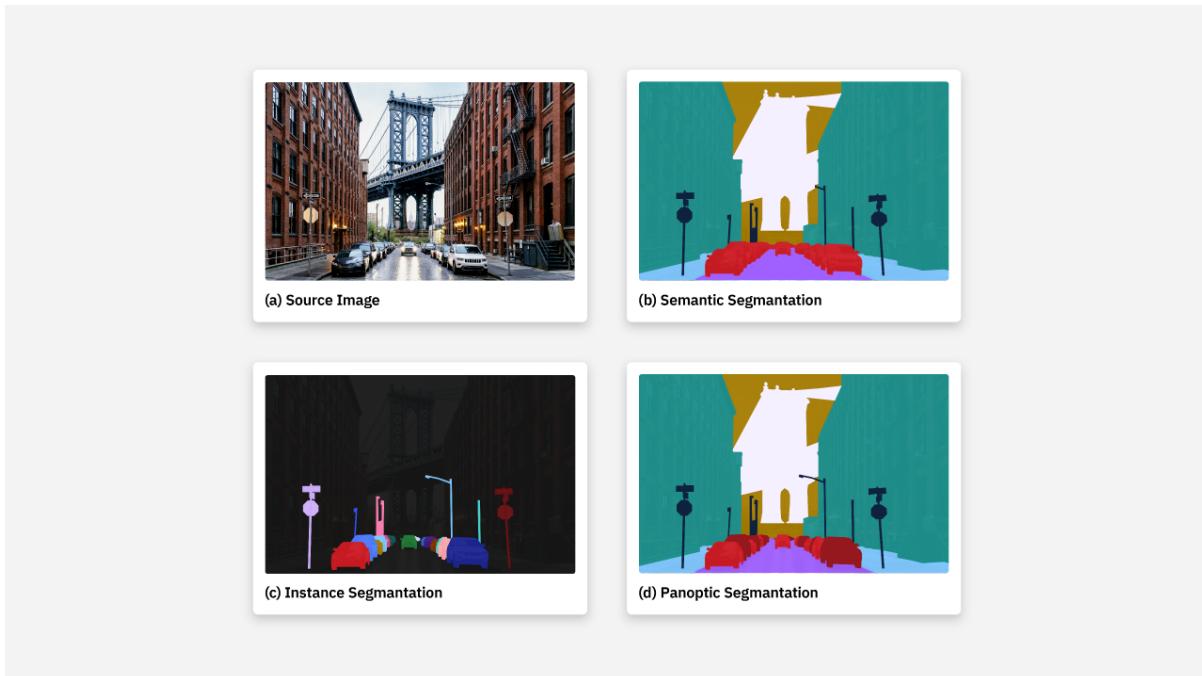


(c) Instance Segmentation

Segmentazione panottica

I modelli di segmentazione panottica determinano la classificazione semantica di tutti i pixel e differenziano ogni istanza di oggetto in un'immagine, combinando i vantaggi della segmentazione semantica e di istanza.

In un'attività di segmentazione panottica, ogni pixel deve essere annotato sia con un'etichetta semantica che con un "ID istanza". I pixel che condividono la stessa etichetta e lo stesso ID appartengono allo stesso oggetto; per i pixel determinati come "elementi", l'ID istanza viene ignorato.



I tentativi iniziali di modelli di segmentazione panottica combinavano semplicemente i due modelli, eseguendo ogni compito separatamente e quindi combinando i risultati in una fase di post-elaborazione. Questo approccio presenta due principali inconvenienti: richiede un notevole sovraccarico di calcolo e presenta discrepanze tra i punti dati ottenuti dalla

rete di segmentazione semantica e i punti dati ottenuti dalla rete di segmentazione delle istanze.

Le nuove architetture di segmentazione panottica mirano a evitare questi inconvenienti con un approccio più unificato al deep learning. La maggior parte di esse si basa su una rete "dorsale", come una rete di piramidi di caratteristiche (FPN - Featured Pyramid Network).

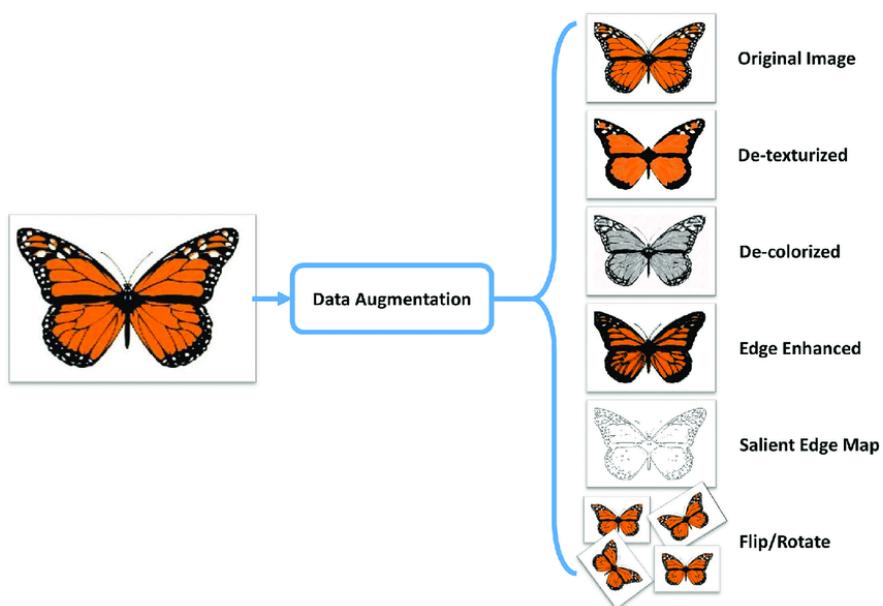
Dataset annotation

La *data annotation* (annotazione dei dati) si riferisce alla classificazione e etichettatura di dati o immagini per convalidare gli algoritmi di segmentazione o altre soluzioni basate sull'intelligenza artificiale. La segmentazione panottica è anche utilizzata per l'annotazione dei dataset. Ad esempio, la segmentazione panottica viene utilizzata in combinazione con un annotatore umano e un assistente automatizzato, che collaborano per annotare i dati: l'azione dell'annotatore umano funge da segnale contestuale per cui l'assistente intelligente reagisce e annota altre parti dell'immagine. Inoltre, viene proposto un modello di segmentazione panottica debolmente supervisionato, che esegue segmentazione semantica e delle istanze, e viene utilizzato per annotare i dataset. Questo modello non rileva istanze sovrapposte, ma ha raggiunto un'accuratezza del 95% sui dati del Pascal VOC 2012 (Visual Object Classes Challenge).

Un'applicazione della segmentazione panottica per l'annotazione dei dataset è per esempio in ambito industriale dove viene utilizzato un modello 3D per generare modelli di edifici industriali. Ciò può migliorare le operazioni di inventario eseguite a distanza, permettendo una stima precisa degli oggetti. Ad esempio, in un impianto nucleare, l'analisi della posizione delle attrezzature attraverso immagini panoramiche annotate con segmentazione panottica consente di ridurre significativamente i costi e i tempi di manutenzione.

Data augmentation

Un'altra promettente applicazione della segmentazione panottica è per l'augmented data, ovvero l'augmentazione dei dati. Utilizzando la segmentazione panottica, è possibile progettare schemi di augmentazione dei dati che operano esclusivamente nello spazio dei pixel, senza richiedere dati aggiuntivi o formazione supplementare, e che sono computazionalmente poco costosi da implementare. Ad esempio, viene proposto un metodo di augmentazione dei dati basato sulla segmentazione panottica, chiamato PanDA. In particolare, il ri-addestramento di modelli esistenti su dataset augmentati da PanDA (generati con un singolo set di parametri congelati) ha portato a guadagni significativi in termini di segmentazione delle istanze e segmentazione panottica, oltre a miglioramenti nella rilevazione attraverso diversi modelli, domini di dataset e scale. Inoltre, grazie all'efficienza dei dataset di immagini di addestramento poco realistici (sintetizzati da PanDA), emerge una nuova riflessione sulla necessità di realismo nelle immagini per garantire un'augmentazione dei dati potente e robusta.



La segmentazione a livello implementato

I risultato di una tecnica di segmentazione deve rispettare alcune proprietà:

- la segmentazione deve essere completa, ovvero tutti i pixel dell'immagine devono appartenere ad almeno una regione della partizione;
- l'insieme di pixel di una regione dell'immagine deve essere connesso;
- tutte le regioni di un'immagine devono essere separate tra di loro;
- deve soddisfare il criterio di omogeneità di features derivanti da componenti spettrali definiti in base all'intensità, dal colore o la texture.

Il suo obiettivo è fornire un'unica mappa segmentata che identifichi sia le classi semantiche sia gli oggetti individuali nelle immagini. Vediamo il concetto matematico alla base, suddiviso per punti:

1. Definizione della Segmentazione Panottica

Segmentazione Semantică: Ogni pixel p è etichettato con una classe c , dove $c \in C$ (insieme delle classi semantiche).

Segmentazione Instance-Level: Ogni pixel p appartenente a oggetti contabili (come persone o automobili) è associato a una classe c e a un identificativo dell'istanza i , con $i \in I$ (insieme delle istanze).

La segmentazione panottica combina queste due informazioni creando una mappa che rappresenta sia le classi semantiche globali c sia le istanze i degli oggetti.

2. Loss Function per l'Ottimizzazione

L'algoritmo utilizza una funzione di perdita che combina obiettivi distinti per le due segmentazioni:

$$\mathcal{L} = \mathcal{L}_{\text{semantica}} + \mathcal{L}_{\text{instance}}$$

Perdita Semantica:

Soltamente una cross-entropy loss sui pixel, definita come:

$$\mathcal{L}_{\text{semantica}} = -\frac{1}{N} \sum p = 1^N \sum_{c \in C} y_{p,c} \log(\hat{y}_{p,c}),$$

Misura che quantifica quanto un modello di intelligenza artificiale sbaglia nel classificare i pixel di un'immagine in categorie predefinite (es. "cielo", "strada", "albero").

- N : Numero totale di pixel nell'immagine.
- p : Indice che scorre i pixel dell'immagine.
- c : Classe a cui un pixel può appartenere.
- $y_{p,c}$: Etichetta vera per il pixel p e classe c (valore 1 se il pixel appartiene a c , 0 altrimenti).
- $\hat{y}_{p,c}$: Probabilità predetta dal modello che il pixel p appartenga alla classe c .

Perdita Instance-Level:

assicura che il modello riesca a capire quali oggetti sono separati e ad assegnare correttamente un'identità a ciascun oggetto (ad esempio, distinguere una persona da un'altra, o una macchina da un'altra).

$$\mathcal{L}_{\text{instance}} = \mathcal{L}_{\text{box}} + \mathcal{L}_{\text{mask}}$$

Incluse una combinazione di localizzazione (bounding box), classificazione e maschere pixel-wise per le istanze. Ad esempio:

- Bounding Box Loss (\mathcal{L}_{box}): Solitamente una perdita di tipo smooth IoU (intersection over union, un parametro che si occupa di valutare il risultato prodotto dalla fase di detection):

$$\mathcal{L}_{\text{box}} = \frac{1}{M} \sum_{i=1}^M \text{SmoothL1}(b_i, \hat{b}_i)$$

dove b_i e \hat{b}_i sono rispettivamente il bounding box vero e predetto per l'istanza i .

- Maschera Instance-Level ($\mathcal{L}_{\text{mask}}$): Usa la cross-entropy sui pixel di ciascuna maschera:

$$\mathcal{L}_{\text{mask}} = -\frac{1}{|P|} \sum_{p \in P} \left[y_{p,i} \log(\hat{y}_{p,i}) + (1 - y_{p,i}) \log(1 - \hat{y}_{p,i}) \right].$$

- $|P|$: Numero totale di pixel in una maschera per un'istanza specifica.
- p : Un singolo pixel della maschera.
- $y_{p,i}$: Etichetta vera per il pixel p appartenente all'istanza i (è 1 se il pixel appartiene all'oggetto, 0 se non ci appartiene).
- $\hat{y}_{p,i}$: Probabilità predetta dal modello che il pixel p appartenga all'istanza i (è la probabilità che il pixel appartenga all'oggetto).

3. Integrazione dei Risultati

Dopo l'addestramento, i risultati di semantica e instance-level vengono fusi:

- Unione delle Maschere: Ogni pixel è classificato in:
 - Oggetti contabili (tramite l'identificatore di istanza).
 - Classi semantiche per oggetti non contabili (es. cielo, strada).

Matematicamente, la mappa panottica è definita come:

$$\text{Panoptic Map}(p) = \begin{cases} (c, i) & \text{se } p \text{ appartiene a un'istanza contabile} \\ c & \text{se } p \text{ appartiene a una classe non contabile} \end{cases}$$

4. Metriche di Valutazione

La qualità della segmentazione panottica è valutata tramite Panoptic Quality (PQ), che combina precisione e completezza:

$$PQ = \frac{\sum_{(c,i) \in TP} \text{IoU}((c, i), (\hat{c}, \hat{i}))}{|TP| + \frac{1}{2}(|FP| + |FN|)},$$

- TP : Coppie corrette (true positives) di classi e istanze.
- FP : Falsi positivi.
- FN : Falsi negativi.
- IoU: Intersezione su unione tra maschere predette e reali.

5. Architettura Tipica

Gli algoritmi moderni (es. Panoptic FPN) combinano:

- Reti Convoluzionali: Per estrarre caratteristiche $\phi(x)$ da un'immagine x .
- Branch Semantico: Per predire la mappa semantica.
- Branch Instance: Per identificare e segmentare oggetti contabili.

Il modello ottimizza entrambe le branche simultaneamente utilizzando funzioni di perdita combinate.

Feature Pyramid Networks (FPN)

Le Feature Pyramid Networks (FPN) sono un'architettura di rete utilizzata principalmente nella visione artificiale, in particolare per compiti come la rilevazione di oggetti e la segmentazione semantica. L'idea delle FPN è quella di migliorare la capacità di una rete neurale di rilevare oggetti su scale diverse, sfruttando una rappresentazione multi-risoluzione delle caratteristiche.

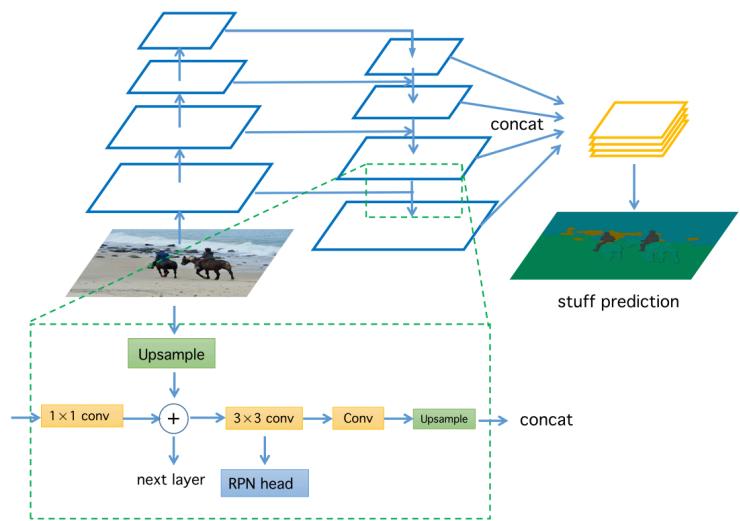
Contesto e Problema

Nel rilevamento di oggetti, gli oggetti possono variare enormemente in termini di dimensioni all'interno di un'immagine:

Oggetti piccoli possono essere difficili da rilevare in strati più profondi della rete (che tendono a catturare caratteristiche più astratte e meno dettagliate).

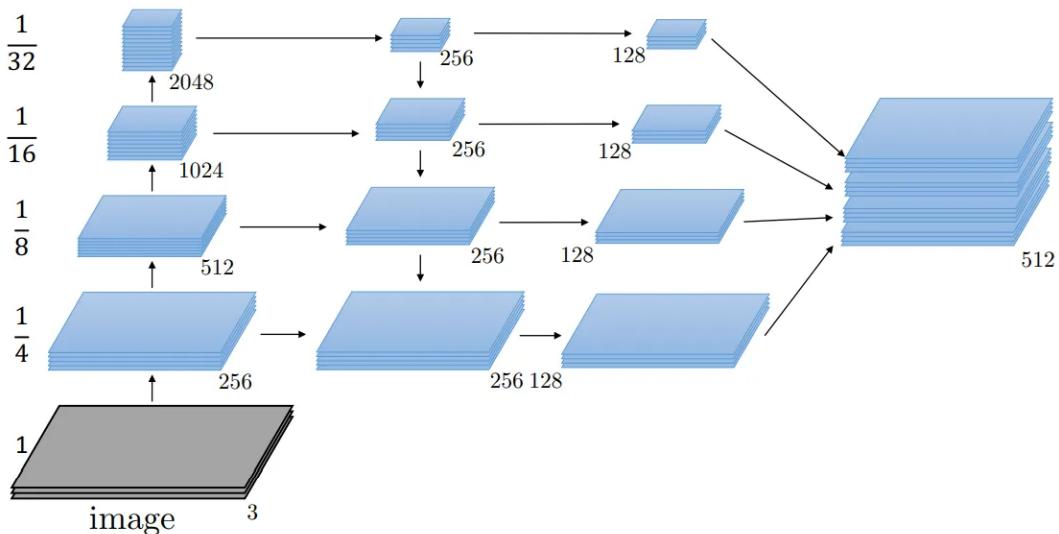
Oggetti grandi possono richiedere caratteristiche più globali e meno dettagliate.

Tradizionalmente, una rete convoluzionale (CNN) genera una gerarchia di caratteristiche a diverse risoluzioni, ma utilizzare queste caratteristiche in modo efficiente è una sfida. Le FPN sono state progettate per risolvere questo problema.



Architettura FPN

Le FPN costruiscono una piramide di caratteristiche sfruttando due principali flussi:



1. Bottom-up Pathway:

- Questo flusso corrisponde all'estrazione delle caratteristiche tramite una classica rete convoluzionale profonda (come ResNet).
- Man mano che si procede nei livelli della rete, le caratteristiche diventano più astratte, con una risoluzione spaziale ridotta.

2. Top-down Pathway:

- In questo flusso, le caratteristiche astratte e a bassa risoluzione vengono propagate verso l'alto.
- Ogni livello della piramide utilizza la upsampling (riscalatura verso l'alto) per combinarsi con le caratteristiche di risoluzione più alta provenienti dal livello inferiore della rete.

3. Lateral Connections:

- Per combinare le informazioni dei due flussi, ogni livello del flusso top-down viene fuso con le caratteristiche corrispondenti dal flusso bottom-up tramite connessioni laterali (ad esempio, tramite convoluzioni 1x1).

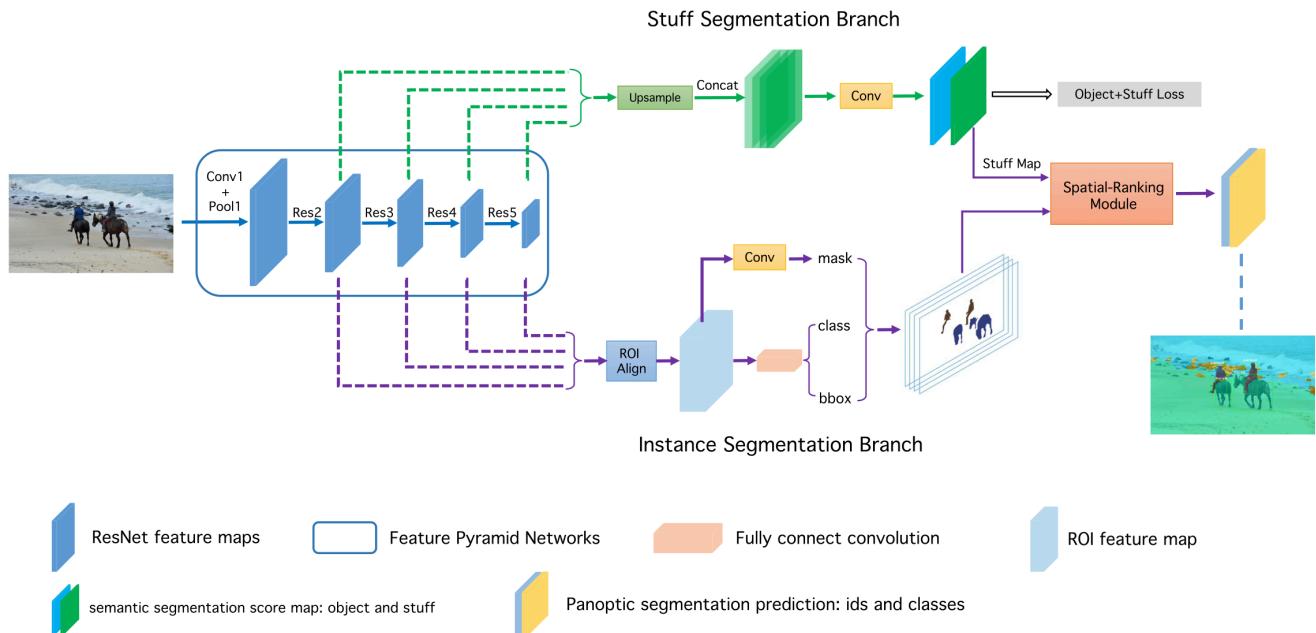
Output

Il risultato è una piramide di caratteristiche che include rappresentazioni a diverse risoluzioni spaziali. Ogni livello della piramide è utile per rilevare oggetti di dimensioni specifiche:

- Risoluzioni alte → oggetti piccoli.
- Risoluzioni basse → oggetti grandi.

Ogni livello della piramide può essere usato come input per un rilevatore o classificatore di oggetti.

End-to-end Network Architecture



Nella segmentazione, si distinguono due principali categorie di elementi in un'immagine:

“Stuff”: si tratta di regioni di un'immagine che rappresentano materiali o sfondi continui e privi di contorni distinti. Esempi comuni includono cielo, prato, acqua, sabbia, pavimento, muro. Non hanno contorni precisi e non si possono contare come entità discrete.

“Things”: sono oggetti ben definiti con contorni chiari che si possono contare come entità distinte. Esempi includono persone, automobili, animali, ecc.

La figura mostra una pipeline end-to-end per la segmentazione panottica, che combina i due rami principali: Stuff Segmentation Branch (per materiali continui) e Instance Segmentation Branch (per oggetti distinti). Ecco come funzionano i due rami e come si integrano nello Spatial-Ranking Module:

Stuff Segmentation Branch

Questo ramo è responsabile della segmentazione delle aree continue, come cielo, terreno o acqua.

Le mappe di caratteristiche derivate dalla backbone (ResNet) e dalla Feature Pyramid Network (FPN) vengono upsampleate (riscalate) per recuperare dettagli a risoluzioni più alte.

Le caratteristiche multi-scala sono concatenate e processate da convoluzioni (es. 3×3) per produrre una Stuff Map, che rappresenta le previsioni delle classi di "stuff" per ogni pixel.

La loss funzione per lo stuff è calcolata simultaneamente alla loss per le istanze, contribuendo all'addestramento globale.

Instance Segmentation Branch

Questo ramo si occupa di segmentare oggetti distinti (es. cavalli, persone).

Le mappe di caratteristiche vengono inviate al layer ROIAlign, che estrae le caratteristiche specifiche delle regioni proposte dall'RPN (Region Proposal Network).

Queste caratteristiche vengono utilizzate per:

- Predire le classi degli oggetti (class).
- Calcolare le coordinate dei bounding box (bbox).
- Generare le maschere di istanza (mask) per ogni oggetto.

Integrazione nello Spatial-Ranking Module

Lo Spatial-Ranking Module (SRM) unisce le informazioni provenienti dai due rami (Stuff Map e maschere di istanza):

- La Stuff Map contiene le previsioni per i materiali continui.
- Le maschere di istanza, i bounding box e le classificazioni provengono dall'Instance Branch.
- Il SRM utilizza queste informazioni per risolvere sovrapposizioni e conflitti tra lo "stuff" e le "things", generando una mappa finale coerente e senza ambiguità.
- Ad esempio, in caso di sovrapposizione, il modulo può dare priorità a oggetti ("things") rispetto a materiali continui ("stuff").

Output Finale

Dopo il passaggio attraverso lo Spatial-Ranking Module, la rete genera una previsione panottica: un'unica mappa in cui ogni pixel è assegnato a una classe di "stuff" o a un'istanza di "thing". Questo permette una comprensione globale e dettagliata dell'immagine.

Minimizzare la Loss Function

Allenare la rete mostrata nel paragrafo precedente implica minimizzare una loss function, che è progettata per guidare l'ottimizzazione dell'intera pipeline. In questo caso, la loss function si compone di diverse componenti che corrispondono ai diversi obiettivi del modello, ossia la segmentazione delle istanze (things) e la segmentazione dei materiali continui (stuff) che già abbiamo approfondito nelle sezioni precedenti di questo articolo.

1. Instance Segmentation Loss:

Comprende le perdite relative al ramo di segmentazione delle istanze:

Classification Loss ($\mathcal{L}_{\text{instance}}$): Calcola l'errore nel predire la classe corretta di un oggetto (ad esempio, cavallo, persona, ecc.).

Bounding Box Regression Loss (\mathcal{L}_{box}): Penalizza la discrepanza tra le coordinate predette dei riquadri di delimitazione e quelle reali.

Mask Loss ($\mathcal{L}_{\text{mask}}$): Valuta l'accuratezza delle maschere binarie predette per ciascuna istanza.

2. Stuff Segmentation Loss:

È una perdita pixel-wise calcolata tra la mappa di previsione dello "stuff" (ad esempio, sabbia, cielo) e la mappa di segmentazione reale. Solitamente utilizza una metrica come la cross-entropy per ogni pixel.

3. Object + Stuff Joint Loss:

Combina le perdite dei due rami (Instance e Stuff), spesso attraverso una somma ponderata:

$$\mathcal{L} = \lambda_{\text{instance}} \cdot \mathcal{L}_{\text{instance}} + \lambda_{\text{stuff}} \cdot \mathcal{L}_{\text{stuff}}$$

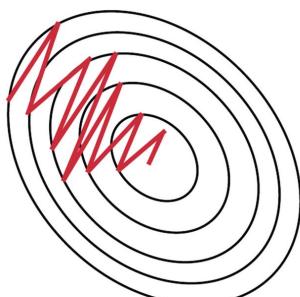
Dove $\lambda_{\text{instance}}$ ed λ_{stuff} sono pesi che bilanciano l'importanza dei due rami.

4. Spatial-Ranking Module Loss (se presente):

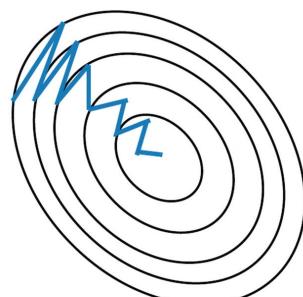
Questa perdita aggiuntiva valuta come le predizioni dei due rami vengono integrate nello Spatial-Ranking Module, garantendo coerenza tra oggetti e materiali.

5. Processo di Ottimizzazione

Durante l'allenamento, la rete utilizza la loss function totale per aggiornare i parametri tramite un metodo di ottimizzazione, come SGD (Stochastic Gradient Descent) o Adam. L'obiettivo è minimizzare la loss totale, migliorando così la qualità sia della segmentazione delle istanze che di quella dei materiali continui.



Stochastic Gradient
Descent **without**
Momentum



Stochastic Gradient
Descent **with**
Momentum

Detectron2

Detectron2 è una libreria di Deep Learning sviluppata da Facebook AI Research (FAIR) per la realizzazione di modelli avanzati di visione artificiale. È una versione aggiornata di Detectron, che fornisce implementazioni di algoritmi per il rilevamento degli oggetti, segmentazione delle immagini, riconoscimento delle pose, e altre attività di visione computazionale. Detectron2 è costruito su PyTorch, una delle librerie di deep learning più popolari, ed è progettato per essere altamente modulare, facilmente estendibile e adatto a ricerche avanzate.

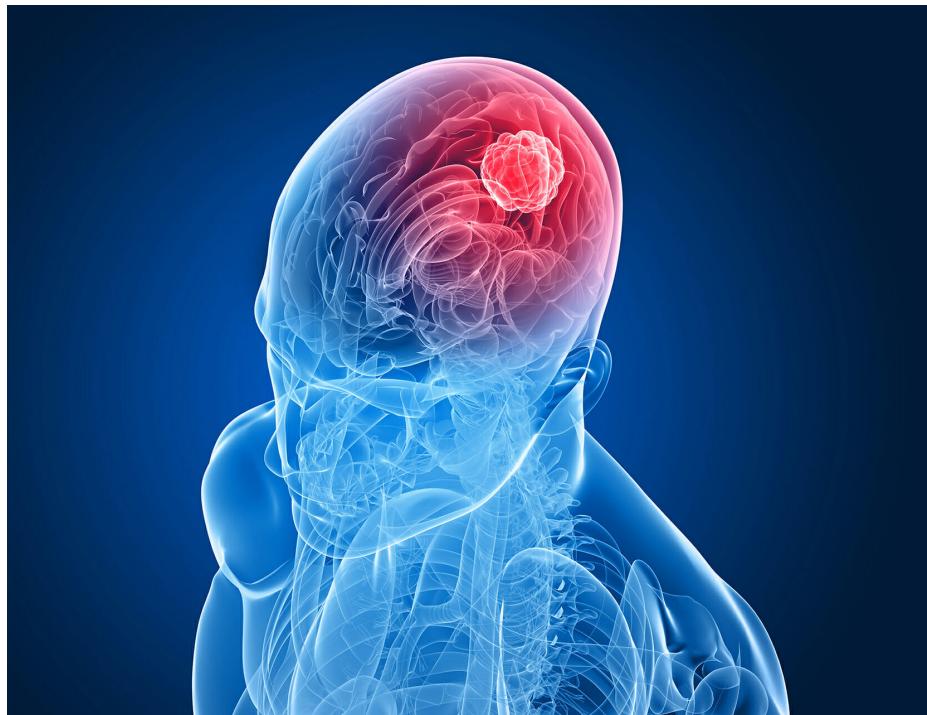


Detectron2 è progettato per essere utilizzato sia in scenari di ricerca che in applicazioni pratiche. Grazie alla sua flessibilità, è ampiamente usato per addestrare modelli su grandi set di dati, permettendo di sfruttare l'architettura a più GPU per accelerare il processo di addestramento. Inoltre, offre una facile interfaccia per personalizzare e adattare i modelli a nuovi task.

Segmentazione Panottica con Detectron2

Esempio d'uso su analisi mediche

Utilizzeremo il modello di Detectron2 fornito per processare immagini mediche, in particolare analisi di tumori nel cervello.



Per fare questo necessitiamo di un dataset con una mole consistente di immagini etichettate, a questo scopo esistono diversi tool online per reperire dati sicuri e già etichettati come Roboflow.

Object Detection

Overview

9.9k images

brain tumor Computer Vision Project

Roboflow 100 Updated 2 years ago

26 stars

Download Project

12k views 811 downloads

TAGS

Object Detection Model yolov5

CLASSES (3)

label0 label1 label2

mAP 79.7% Precision 88.5% Recall 76.1%

Try This Model

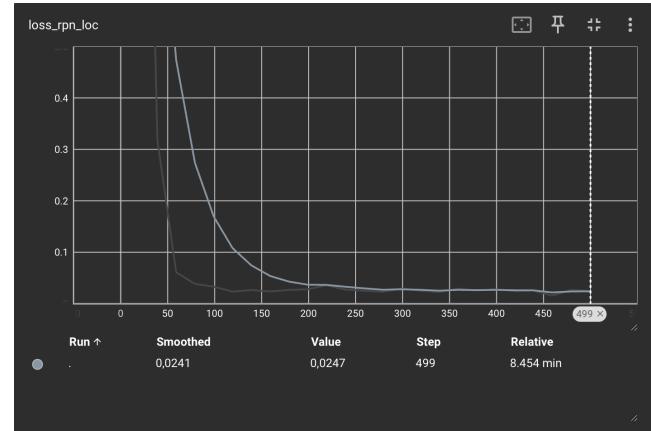
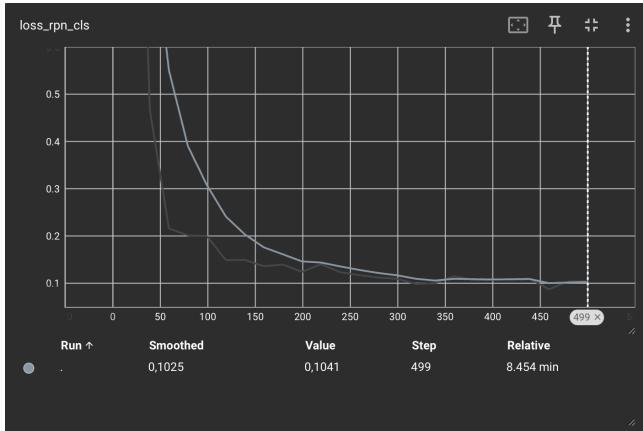
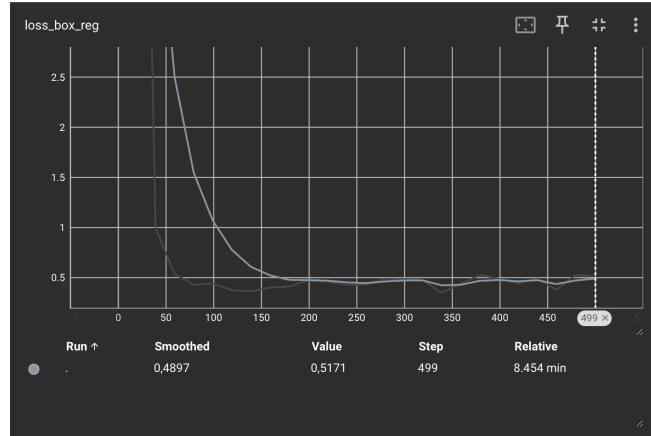
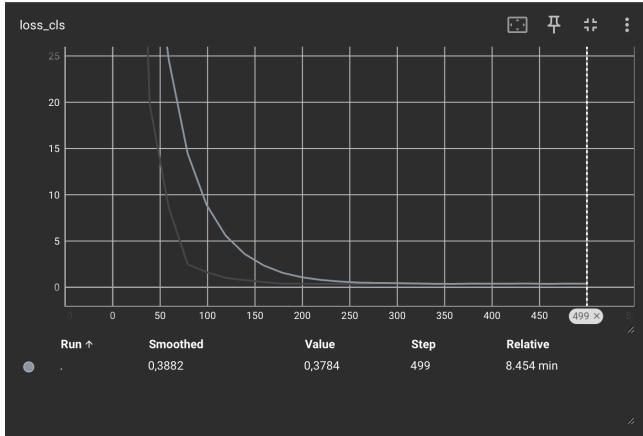
Drop an image or browse your device

Una volta scaricato il dataset possiamo passare alla configurazione dei parametri del nostro modello. Il modello iniziale pre-addestrato usato è una versione di Faster R-CNN con una ResNet-50 come backbone e una Feature Pyramid Network per migliorare le prestazioni su oggetti di diverse dimensioni, addestrato sul dataset COCO

Il Learning Rate viene alzato gradualmente nelle prime 100 iterazioni fino a valore, per poi scendere verso le iterazioni finali per ottimizzare la ricerca del minimo globale.

Una volta addestrato il modello restituirà la perdita d'errore aggiornata sull'addestramento:

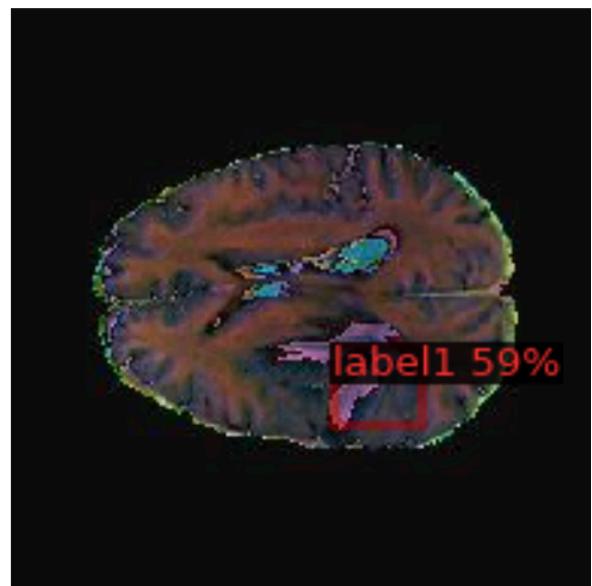
1. **Total_loss**: È la somma totale di tutte le componenti della perdita. Questo valore rappresenta la perdita complessiva che il modello sta cercando di minimizzare durante l'addestramento.
2. **Loss_cls**: La perdita di classificazione. Misura quanto il modello ha sbagliato nella classificazione delle regioni proposte come oggetti. Un valore più basso indica che il modello sta facendo bene nel classificare gli oggetti nelle immagini.
3. **Loss_box_reg**: La perdita di regressione della box. Indica quanto il modello ha sbagliato nel predire le coordinate delle bounding box degli oggetti. Questa perdita è più alta quando le bounding box predette si discostano significativamente dalla posizione corretta degli oggetti nell'immagine.
4. **Loss_rpn_cls**: La perdita di classificazione della RPN (Region Proposal Network). Questo valore indica l'errore nella classificazione delle proposte di regioni che il modello crea, cioè se una proposta di regione è un oggetto o uno sfondo. Un valore più basso è desiderabile.
5. **Loss_rpn_loc**: La perdita di localizzazione della RPN. Misura quanto il modello ha sbagliato nel posizionare correttamente le proposte di regioni nelle immagini. È il contributo alla perdita legato alla precisione della posizione delle proposte generate dalla RPN.



Validazione dei dati

Una volta finito l'addestramento della rete verrà fatta la validazione dei dati sul validation_set, questo passaggio permette di ottimizzare i parametri e scegliere il miglior modello (senza influenzare l'addestramento diretto).

Il risultato ottenuto presenta risultati di questo tipo:



Documentazione su:

openaccess doc

tesi politecnico torino