Object Detection

Object Detection

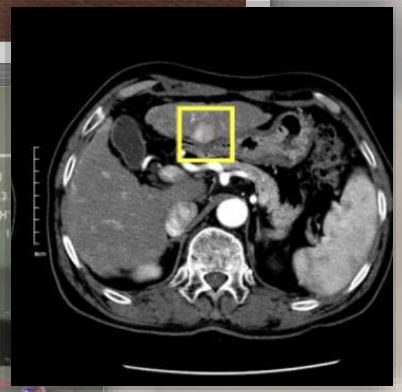
Premessa

Nell'ambito dell'apprendimento automatico, ML e DL, l'**object detection** è una branca della visione artificiale con due particolari scopi:

1. *Individuare e localizzare una serie di oggetti all'interno dell'input fornito.*
2. *Assegnare ad ogni istanza individuata un'etichetta che la rappresenti.*

Object Detection

Casi applicativi



Esistono una miriade di casi d'uso e applicazioni che, nella vita di tutti i giorni, sfruttano sistemi di detection automatica per semplificare situazioni, risolvere problemi…

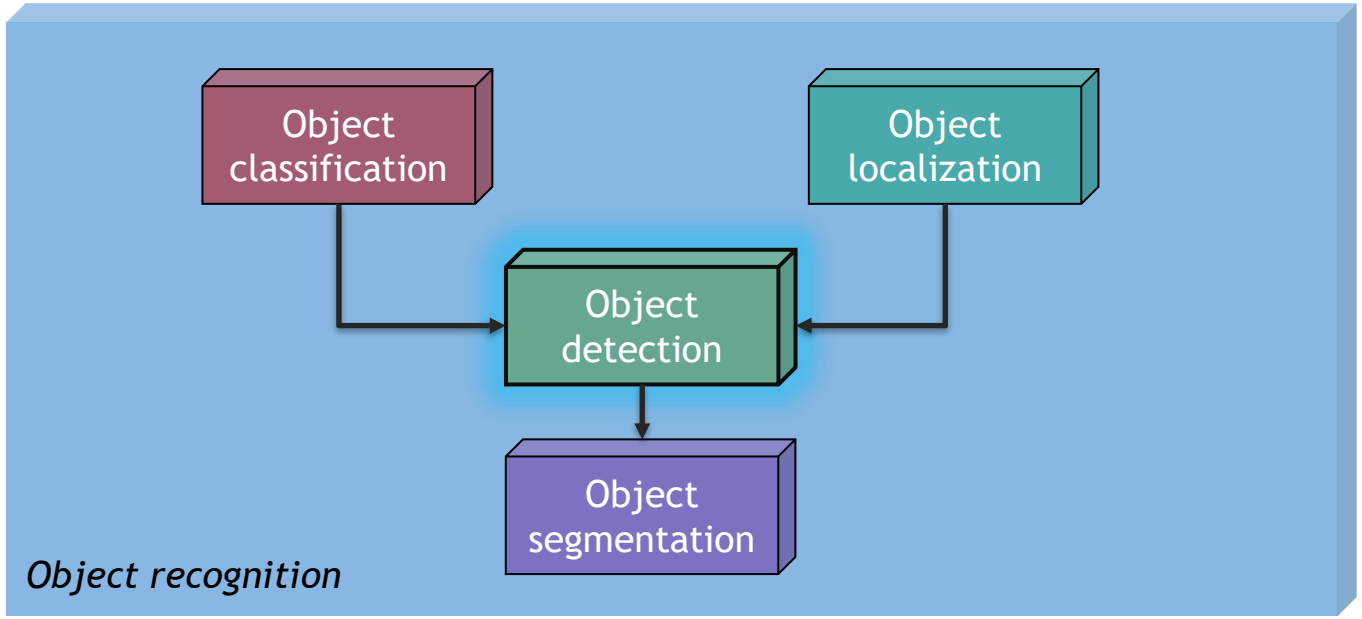
- Riconoscimento facciale.
- Controllo qualità.
- Guida autonoma.
- Diagnosi mediche.

Object Detection

Terminologia

Object detection è in realtà parte di un grande calderone di altre terminologie, casi d'uso e tecniche spesso fra loro confuse o nominate in maniera intercambiabile.

Uno dei modi più utilizzati per dare ordine alle varie terminologie è quello che vede l'**object recognition** a monte di tutto:



Object Detection

Terminologia

- **Object recognition** identifica in termini generali le tecniche che mirano a comprendere il contenuto di un immagine: sia inteso questo come l'assegnazione di etichette, la localizzazione di oggetti o altro...

- **Object localization** individua le posizioni di uno specifico oggetto di interesse all'interno di un'immagine.

- **Object classification** assegna una etichetta, in genere, all'intera immagine oppure a sotto-regioni specifiche.

- **Object detection** combina localizzazione e classificazione mirando a trovare istanze di oggetti all'interno di immagini e assegnarvi una etichetta.

- **Object segmentation** localizza gli oggetti cercando di seguire al meglio i contorni ed effettuando, nello specifico, una classificazione pixel a pixel.

Object Detection

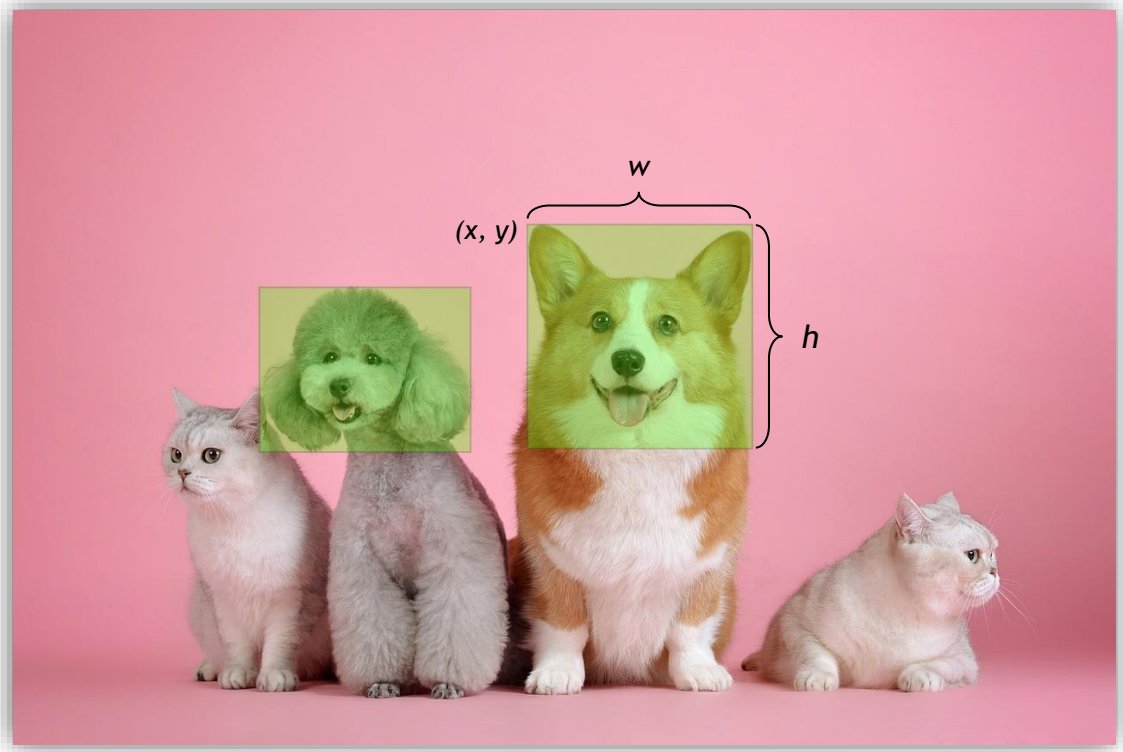
Terminologia: classification

- **Object classification** assegna una etichetta, in genere, all'intera immagine oppure a sotto-regioni specifiche.

Object Detection

Terminologia: localization

- **Object** **localization** individua le posizioni di uno specifico oggetto di interesse all'interno di un'immagine.



Dog face detector

Il modello è specializzato nel localizzare un oggetto.

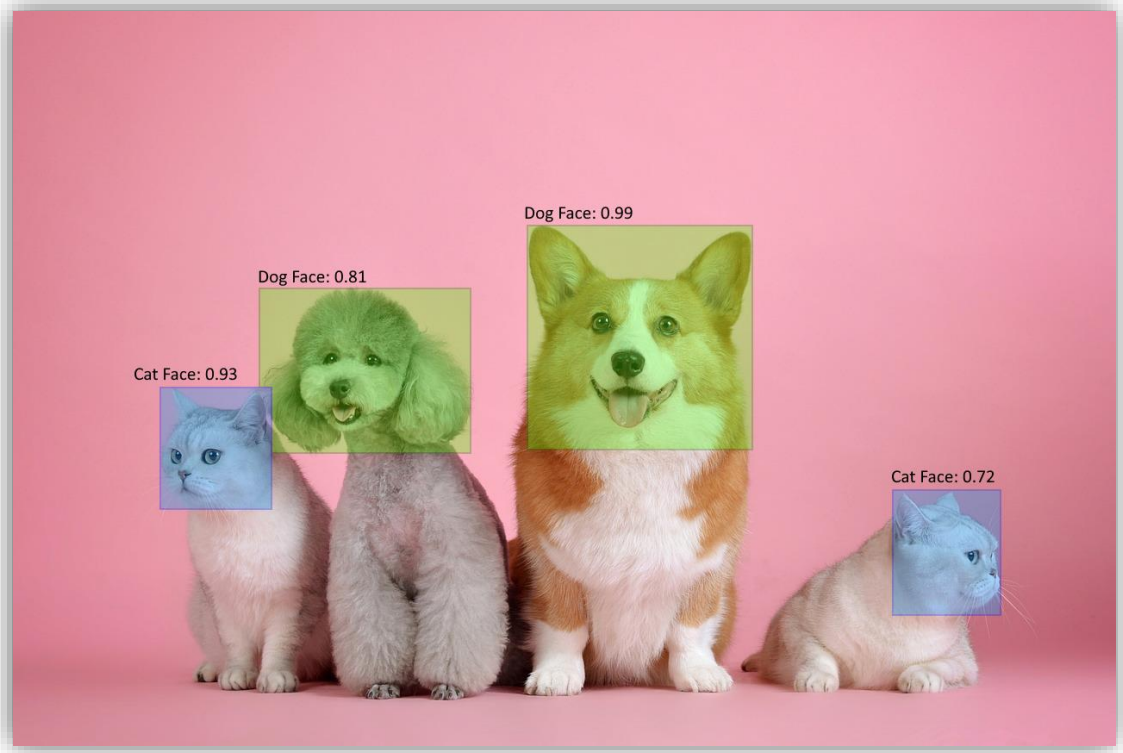
La localizzazione si traduce nel ricercare delle **Bounding Box**.

Le bounding box sono regioni di interesse descritte in genere da 4 valori:
[x, y, w, h]

Object Detection

Terminologia: detection

- **Object detection** combina localizzazione e classificazione mirando a trovare istanze di oggetti all'interno di immagini e assegnarvi una etichetta.



Il modello localizza più oggetti nell'immagine.

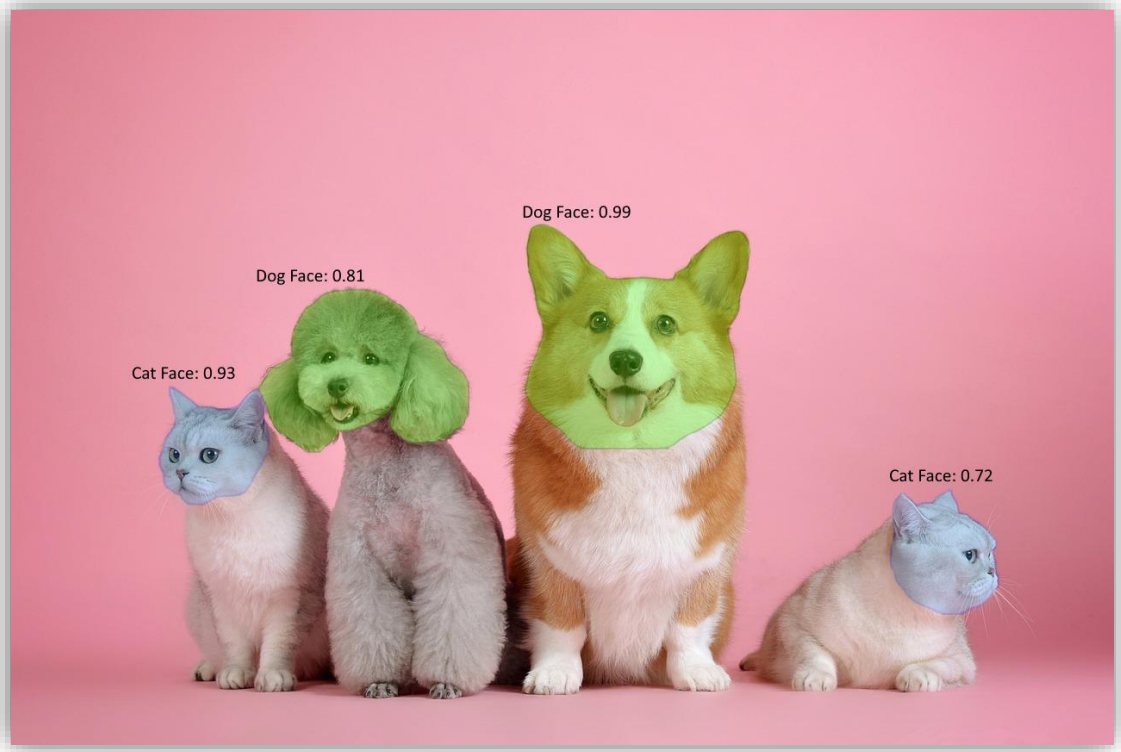
La localizzazione si traduce nel ricercare delle **Bounding Box**. Ad ognuna si assegna, con una certa confidenza, una classe.

Ci sono diversi modi di descrivere queste box, un esempio:
[x, y, w, h, $conf_{classe\ 1}$, …]

Object Detection

Terminologia: segmentation

- **Object segmentation** localizza gli oggetti cercando di seguire al meglio i contorni ed effettuando, nello specifico, una classificazione pixel a pixel.



Animal faces detector

Il modello localizza più oggetti nell'immagine.

La localizzazione si traduce nel ricercare gli esatti contorni degli oggetti.

A descriverli ci sarà la box ed anche una codifica dei singoli pixel che nella box appartengono o meno all'oggetto.

Object Detection

Architetture

Esistono diverse architetture ‘standard’ di object detection utilizzate nelle attività e nei problemi di tutti i giorni. Per distinguere l’una dall’altra i criteri principali sono i seguenti:

- *Numero di fasi di esecuzione*: ad una fase o due fasi.
- *Sistema di localizzazione*: region-proposal o anchor-boxes.
- *Tipo di estrattore di caratteristiche, o **backbone***.
- *Finalità*: esecuzioni real-time, massima accuratezza…

Object Detection

Una fase – Due fasi

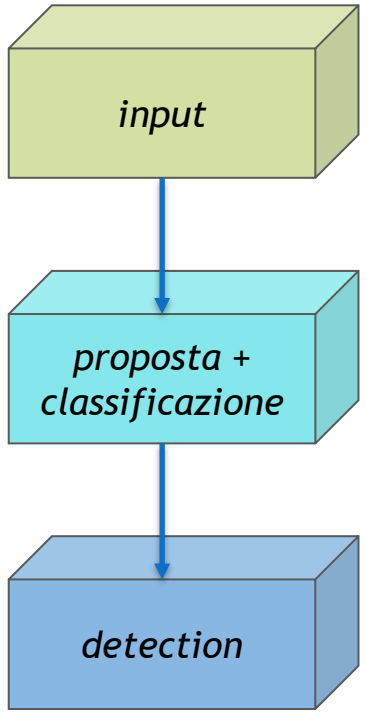
Il numero di **fasi**, *stage*, sta ad indicare il numero di ‘passate’ di elaborazione fatte nell’input prima di fornire in output le rilevazioni.

Una fase

Eseguono in un singolo momento:

1. Ricerca/proposta di zone in cui è probabile trovare oggetti
2. Classificazione di queste ultime.

Sono generalmente **più rapide** delle architetture a due fasi ma, al contrario, **meno accurate**.



Object Detection

Una fase – **Due fasi**

Il numero di **fasi**, *stage*, sta ad indicare il numero di ‘passate’ di elaborazione fatte nell’input prima di fornire in output le rilevazioni.

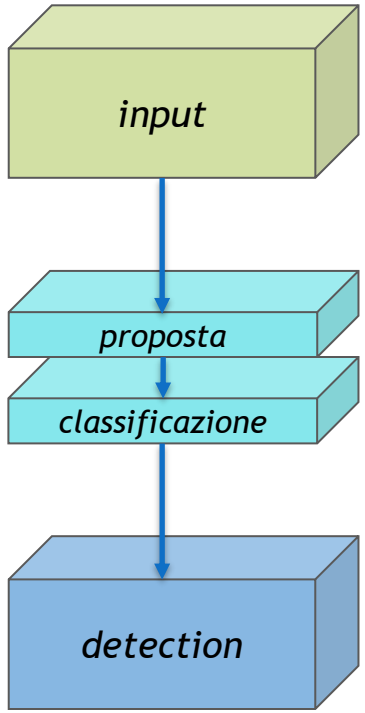
Due fasi

Fase 1: ricerca/proposta regioni in cui è possibile trovare oggetti.

Fase 2: le regioni vengono analizzate e gli oggetti in esse vengono classificati sulla base delle caratteristiche estratte.

A fronte di un maggior peso computazionale:

- **I tempi di inferenza crescono.**
- **le accuratezze ne risentono positivamente.**



Object Detection

Regioni - Ancore

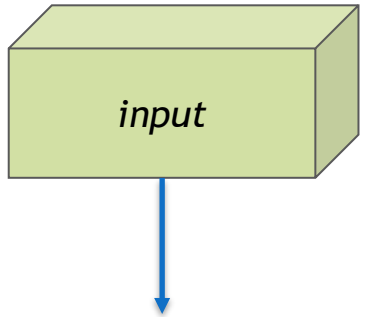
Regioni ed **ancore** hanno entrambe lo scopo di fornire una indicazione delle papabili zone di interesse per la rilevazione degli oggetti.

Regioni

Le regioni ‘candidate’ per la ricerca di oggetti sono proposte.

Le proposte vengono fatte tramite reti ed algoritmi specializzati nel far questo.

Le immagini sono analizzate al fine di scoprire quali zone sono potenzialmente portatrici di maggior contenuto informativo.



Prop.	X	Y	W	H
1	312.2	51.7	100	67
2	24.0	668.2	52	30
…	…	…	…	…

Object Detection

Regioni - **Ancore**

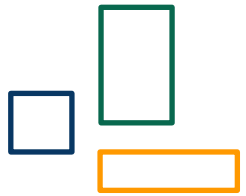
Regioni ed ancore hanno entrambe lo scopo di fornire una indicazione delle papabili zone di interesse per la rilevazione degli oggetti.

Ancore

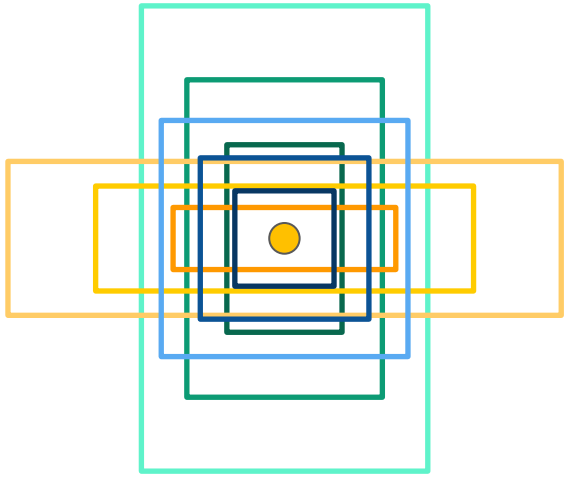
Le ancore sono intese come rettangoli posti nell'immagine campione e nei quali si ottimizza la ricerca degli oggetti.

Ogni rettangolo ha dimensione e forma specifica. Queste caratteristiche specializzano la ricerca degli oggetti.

Alcune ancore potranno specializzarsi nel carcare oggetti piccoli, grandi, larghi, alti…



…3 proporzioni…
…3 dimensioni…



…per ogni punto ancora della griglia immagine…

Object Detection

Cercare oggetti: sliding windows

Ricercare oggetti in maniera efficiente in un immagine è un punto fondamentale per ottenere modelli che poi possano definirsi utilizzabili in applicazioni industriali o di tutti i giorni.

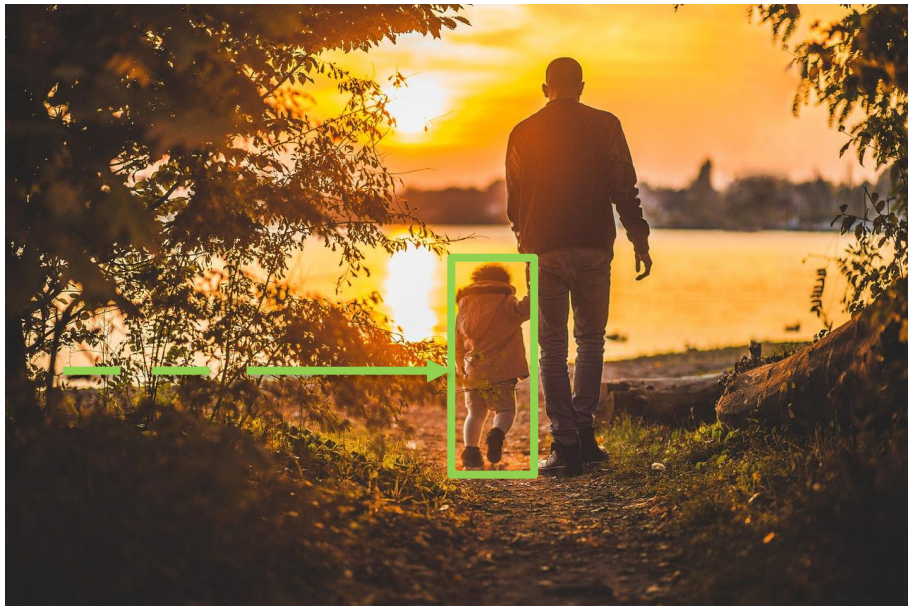
Si immagini ad esempio di voler rilevare delle persone in un'immagine:

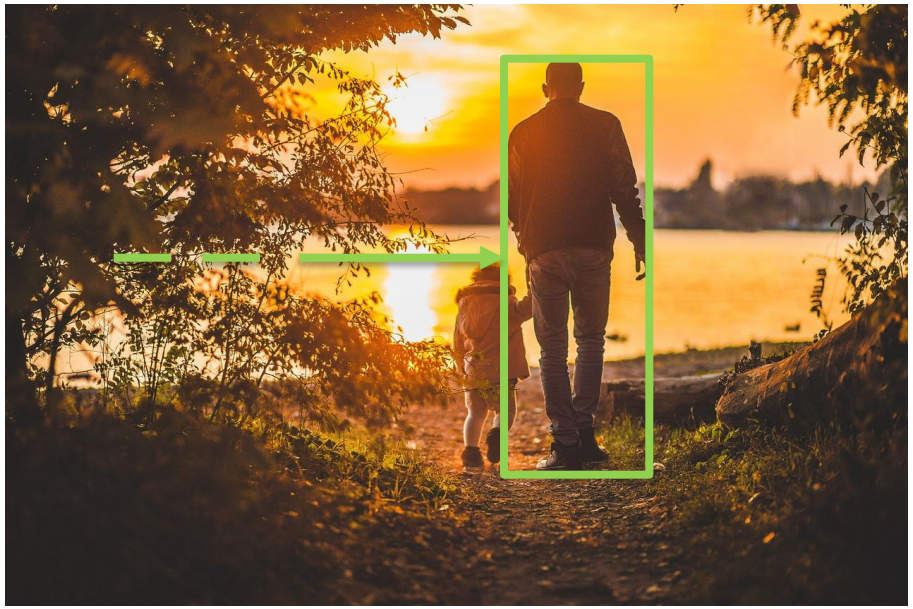
Ogni persona è diversa…più o meno alta, adulta, bambina.

Se cercassimo in maniera esaustiva ogni possibile condizione nell'immagine, sarebbe molto inefficiente.

Object Detection

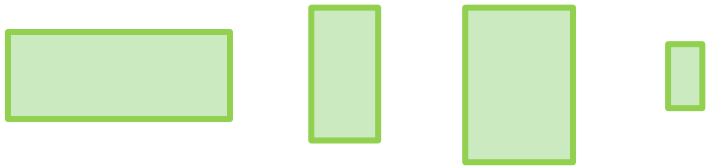
Cercare oggetti: sliding windows





Per cercare ‘persone’ di tipo *bambino*, faremmo scorrere una finestra di ricerca più piccola di quella di un adulto alla stessa distanza…

Potremmo voler cercare persone in distanza, vicine…o anche sdraiate…



Object Detection

Cercare oggetti: Selective Search

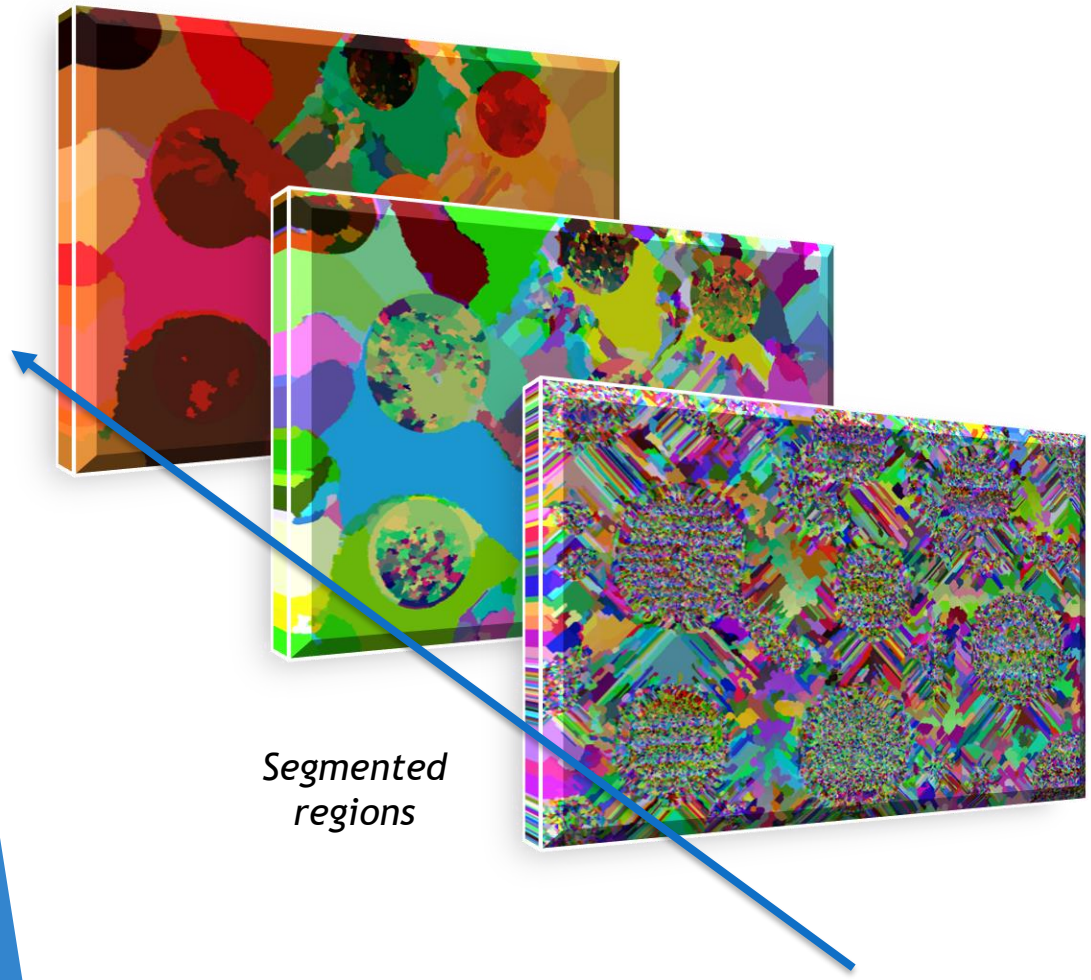
In risposta ad un sistema inefficiente di ricerca di zone di interesse, sono nati algoritmi e reti dedicate alla proposta di regioni.

Selective Search è uno fra gli algoritmi principali:

- *Analizza l'immagine al fine di raggruppare pixel con intensità simili.*
- *Lo fa in maniera gerarchica, accorpando e raggruppando le informazioni.*
- *Ogni regione che ne deriva, identifica una box di ricerca proposta.*
- *All'interno di queste zone poi, si farà la detection degli oggetti.*

Object Detection

Cercare oggetti: Selective Search



Segmented regions

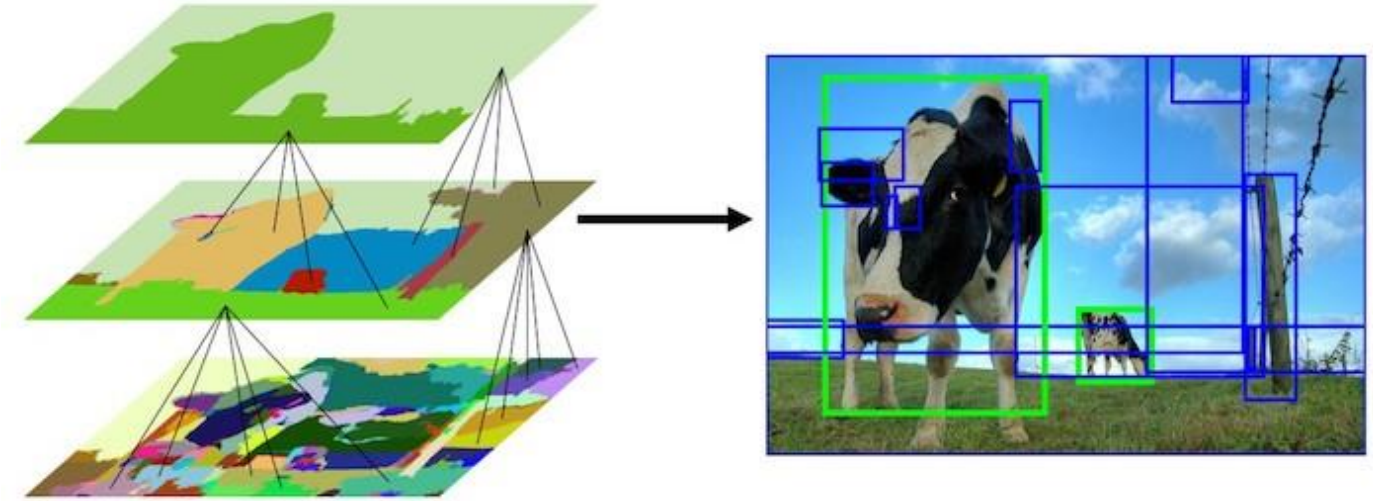
Input Image

Object Detection

Cercare oggetti: Selective Search

Il raggruppamento delle informazioni viene fatto sulla base di similarità, siano questa a livello di:

- *Dimensione*.
- *Colore*.
- *Forma*.
- *Texture*.
- …



Object Detection

Cercare oggetti: Edge Boxes

Questo metodo propone regioni sulla base di contorni e/o gruppi di contorni rilevati nell'immagine.

- Ricerca bordi e contorni con metodi classici.
- Scorre sull'immagine finestre diverse in dimensione e scala.
- Analizza la densità di contorni e bordi contenuti nelle finestre ed esterni alle finestra.
- Assegna un punteggio sulla base delle densità trovate.
- Applica un algoritmo di merge che combina più finestre e riduce le opzioni.

Object Detection

Backbone

Per backbone si intende l'architettura di rete scelta come componente principale di estrazione feature nel modello di object detection.

Trattando principalmente con immagini, le reti di **object detection** sfruttano come backbone delle **CNN** o simili:

- *Una raccolta di layer convoluzionali di estrazione feature.*
- *Layer di pooling per sintetizzare le informazioni.*
- *Normalizzazioni e attivazioni per stabilire la convergenza dell'apprendimento.*

I principali backbone si distinguono fra le classiche reti **CNN**:

- VGG.
- ResNet.
- Inception
- …

Object Detection

Architetture classiche: R-CNN

R-CNN è un'architettura di object detection a due fasi che combina la proposta di regioni rettangolari con la forza di estrazione features delle **CNN**:

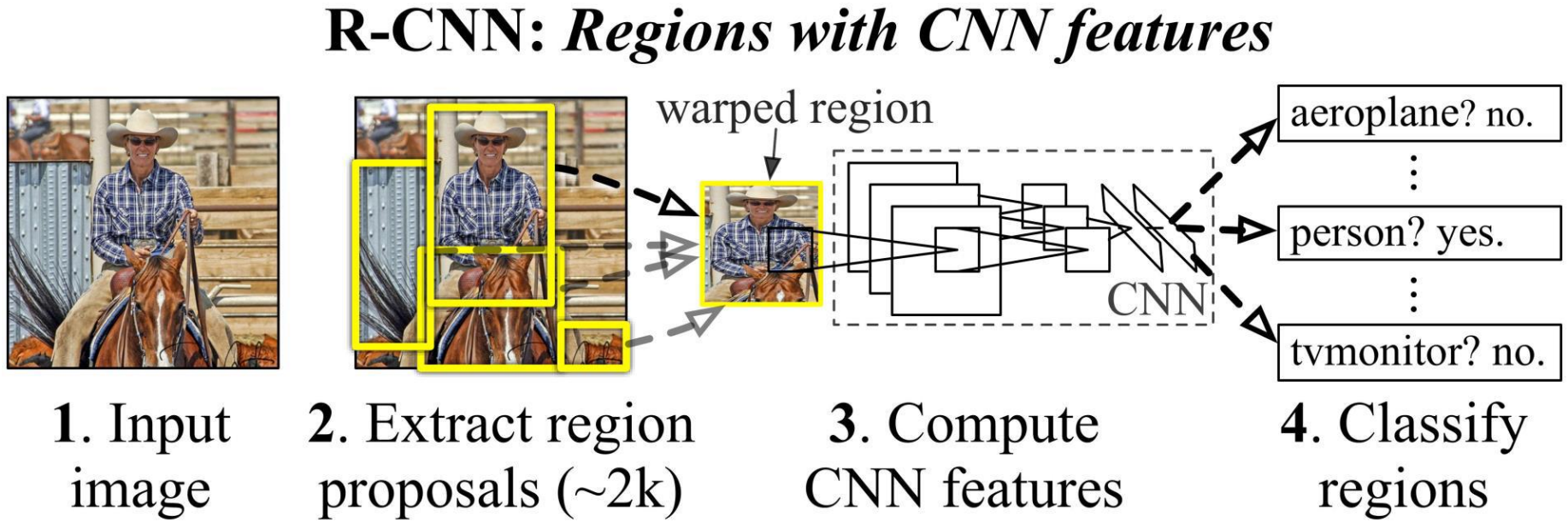
- Fase 1: *identifica, fra le proposte, le regioni che potrebbero contenere oggetti di interesse*.
- Fase 2: *classifica ogni oggetto trovato nelle regioni*.

Per step:

- Proposta regioni con un algoritmo dedicato, ad esempio **Selective Search**.
- Ridimensionamento regioni ad una dimensione fissa ed estrazione feature con **CNN**.
- Classificazione feature e regressione di box più raffinate.
- Algoritmo **NMS** (Non-Maximum suppression) sceglie fra le molteplici box le più confidenti e meno sovrapposte alle altre.

Object Detection

Architetture classiche: R-CNN



Object Detection

Architetture classiche: Fast R-CNN

Fast R-CNN è un detector a due fasi: parte da un **R-CNN** ed ha il presupposto di migliorarlo.

Per step:

- Proposta regioni con algoritmi quali **Selective Search**.
- Estrazione feature dall'intera immagine con una **CNN** pre-addestrata.
- Operazione di **ROI Pooling**:
  - L'input dell'operazione sono la feature map e le regioni proposte.
  - Le regioni sono riproiettare sulla feature map.
  - Ogni regione viene suddivisa in sotto-regioni di dimensione specifica.
  - Viene eseguito il pooling delle sotto-regioni.
  - Si concatenano i risultati in un vettore di dimensione nota e fissa.

Rif: *Region of interest pooling*

Object Detection

Architetture classiche: Fast R-CNN

Fast R-CNN è un detector a due fasi: parte da un **R-CNN** ed ha il presupposto di migliorarlo.

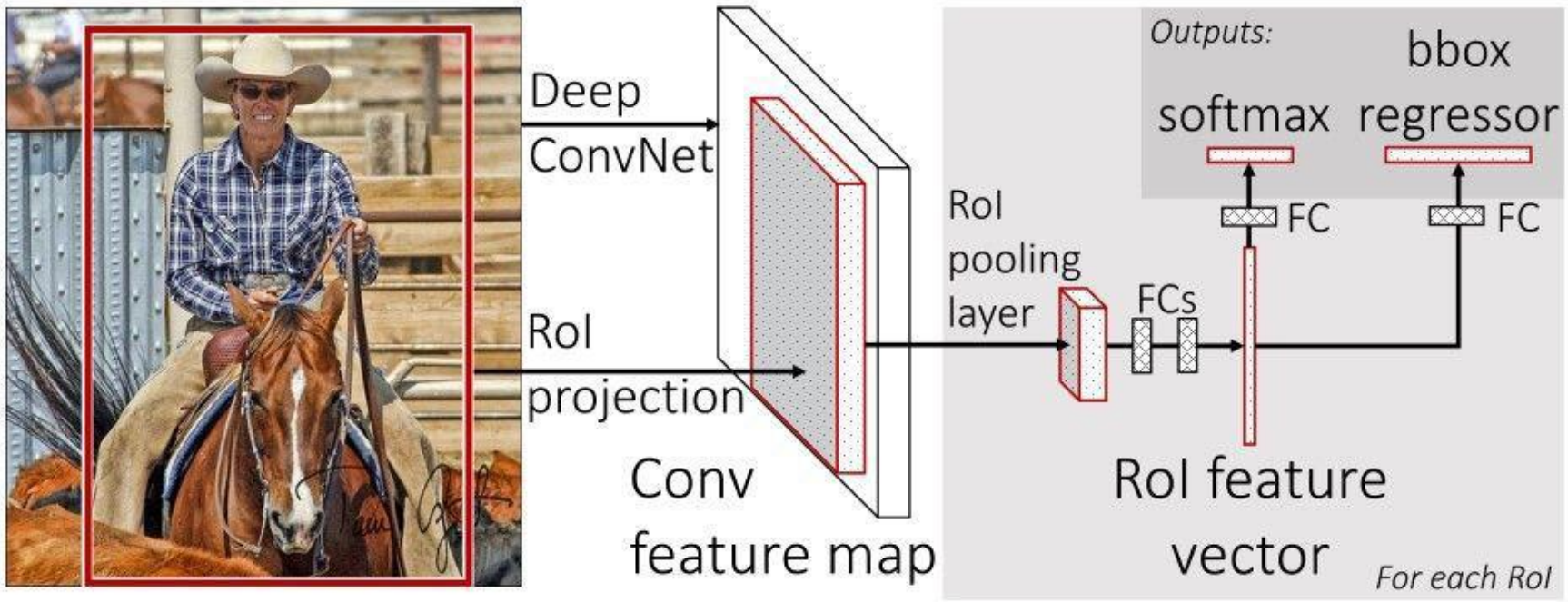
Per step:

- Le caratteristiche estratte dal pooling attraversano un classificatore ed un regressore.
- Il classificatore identifica la presenza o meno di un particolare oggetto.
- Il regressore adatta e migliora la proposta di box.

Rif: *Region of interest pooling*

Object Detection

Architetture classiche: Fast R-CNN



Fast R-CNN migliora **R-CNN** estraendo in un unico momento le feature dell'intera immagine e non regione per regione. L'accuratezza viene preservata ottenendo però efficienza e velocità in addestramento e inferenza.

Object Detection

Architetture classiche: Faster R-CNN

Faster R-CNN è un detector a due fasi: si sviluppa da **R-CNN** e **Fast R-CNN** ed ha il presupposto di migliorarli.

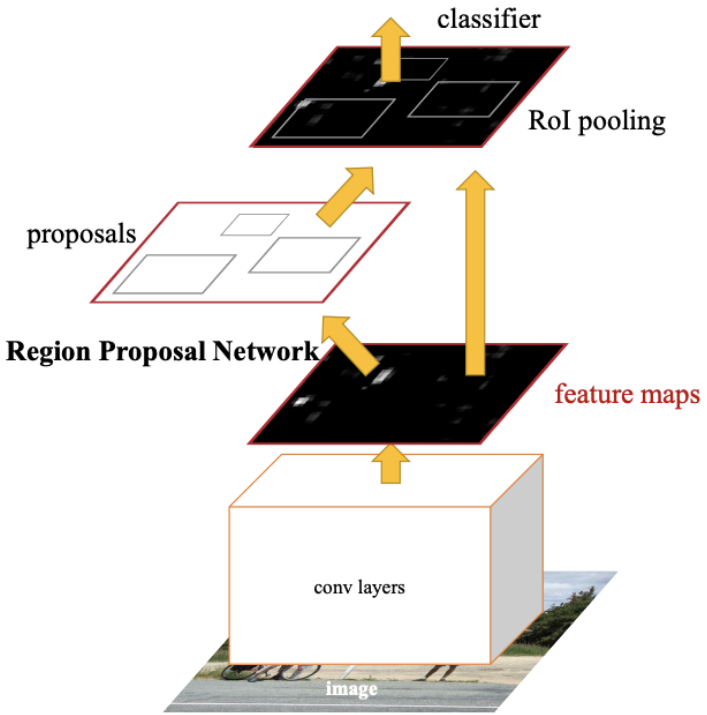
Per step:

- L'input attraversa una rete backbone di estrazione feature ottenendo una feature map.
- La feature map attraversa una rete dedicata alla proposta di regioni: la **RPN**.
- **RPN** parte da una serie predefinita di una bounding box di diversa proporzione e scala posizionate nell'immagine.
- Le ancore vengono raffinate e per ognuna si identifica la confidenza con la quale possiedono o meno oggetti di interesse.
- Sono proposte le regioni.
- Le regioni proposte da **RPN** attraversano il *Roi Pooling* come per **Fast R-CNN**.
- Le caratteristiche estratte dal *Roi Pooling* attraversano classificatore e regressore.

Object Detection

Architetture classiche: Faster R-CNN

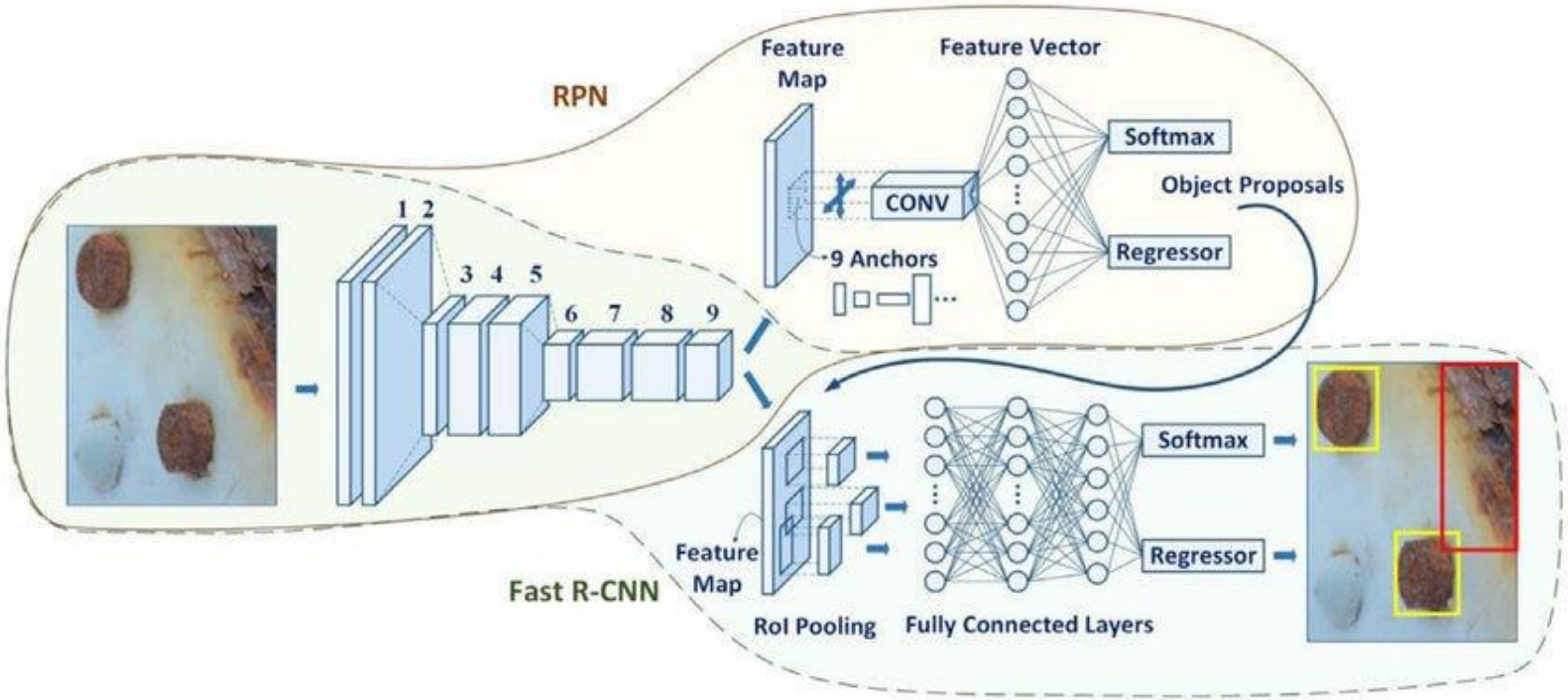
Come per **Fast R-CNN**, **Faster R-CNN** ottimizza l'intero sistema per un addestramento completo, sostituendo i classici algoritmi con **RPN**. A giovarne è nuovamente l'efficienza di addestramento e inferenza.



Object Detection

Architetture classiche: Faster R-CNN

Come per **Fast R-CNN**, **Faster R-CNN** ottimizza l'intero sistema per un addestramento completo, sostituendo i classici algoritmi con **RPN**. A giovarne è nuovamente l'efficienza di addestramento e inferenza.



Object Detection

Architetture classiche: SSD

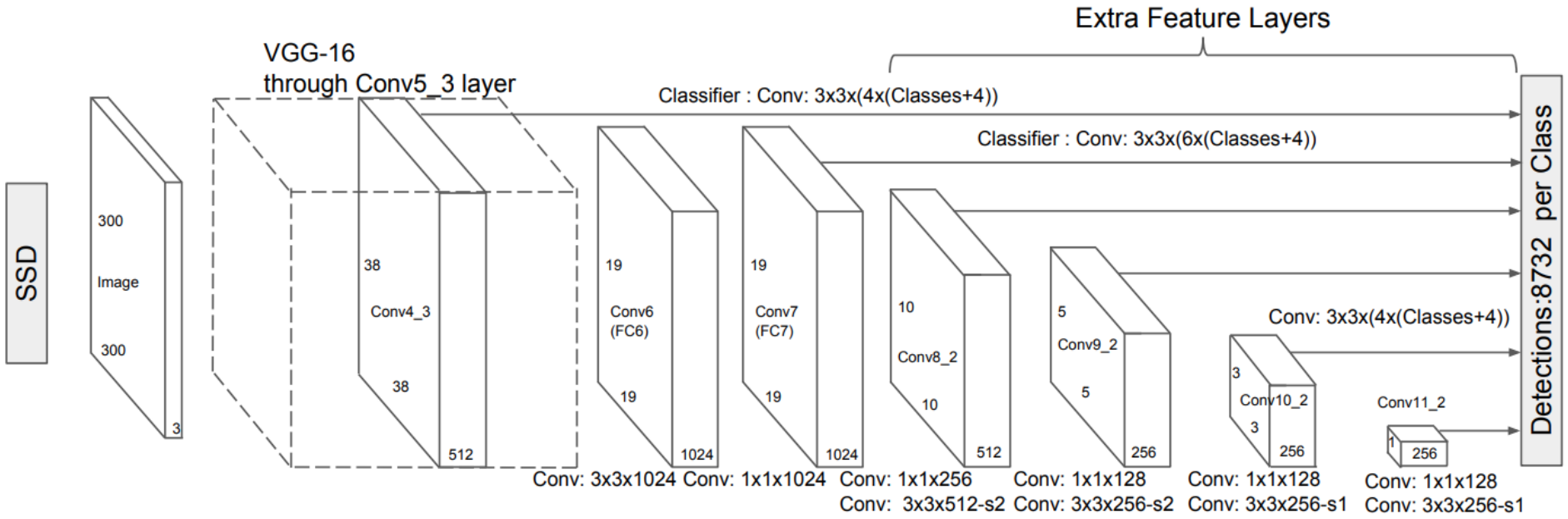
SSD è un detector a singola fase: l'input attraversa in un solo passaggio una rete neurale profonda estraendo oggetti, classificandoli e indicandone la box.

Per step:

- L'input attraversa una **CNN** backbone per estrarre caratteristiche di alto livello.
- La feature map diventa input di diversi layer convoluzionali al fine di estrarre caratteristiche in scale diverse. Ne conseguono diversi livelli di dettaglio e possibilità di localizzare oggetti di dimensioni diverse.
- Dalle feature map ottenute, per ogni posizione, genera un set ancore predefinite con aspetto e scala differente.
- Per ogni ancora viene predetta la probabilità di essere un particolare oggetto e l'offset da applicare all'ancora per meglio adattarsi all'oggetto stesso.
- Come fatto per altre architetture, un algoritmo di **NMS** riduce la ridondanza fra le proposte fatte mantenendo quelle più confidenti e meno sovrapposte.

Object Detection

Architetture classiche: SSD



Object Detection

Architetture classiche: YOLO

Questo detector a singola fase che sfrutta la potenzialità delle reti neurali profonde e sfrutta due sistemi distinti: un feature extractor ed un detector.

Per step:

- L'input attraversa una **CNN** backbone estraendo feature map di diversa scala.
- È applicata poi una suddivisione delle feature map in una griglia SxS. Ogni griglia, cella, farà riferimento ad una regione spaziale dell'immagine originale.
- Per ogni cella è predetto un numero fisso di box definite da: centro (x, y), larghezza, altezza e confidenza che la box contenga un oggetto.
- Per ogni cella è calcolata la probabilità di contenere le diverse istanze di oggetti possibili.
- Delle celle potenzialmente contenenti oggetti, si analizzano le box proposte sfruttando una soglia impostata di Intersection over Union, **IoU**.
- Proposte con **IoU** sotto soglia sono scartate, proposte valide sono considerate.
- Le molteplici proposte sono poi gestite da un algoritmo di **NMS**.

Object Detection

Architetture classiche: YOLO

Quanto descritto per YOLO va comunque preso come idea di architettura generale in quanto, al giorno d'oggi esistono molteplici varianti che ne conservano il nome.

Di variante in variante gli obbiettivi sono comunque condivisi:

- Accrescere l'accuratezza.
- Ridurre il tempo di inferenza.

Per raggiungere questi obbiettivi, miglioramenti incrementali vengono pian piano fatti. Di seguito un riferimento alle diverse architetture:

A Guide to the YOLO Family of Computer Vision Models

Proviamo?