UNIVERSITÀ DELLA CALABRIA
DIPARTIMENTO DI **INGEGNERIA INFORMATICA, MODELLISTICA, ELETTRONICA E SISTEMISTICA**

22 Gennaio 2026

Corso di Laurea Magistrale in Ingegneria Informatica
Corso di Computer Vision - A.A. 2025/2026

Rilevamento e Segmentazione di Cavi Elettrici su Dataset TTPLA

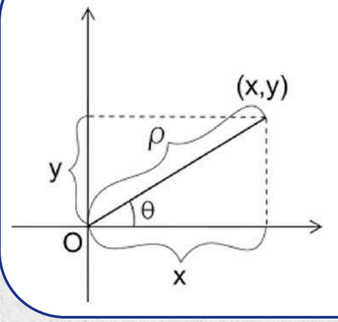
Implementazione basata su Detectron2 e PointRend con Loss Ibrida

Studenti:
Martucci Anastasia, matr. 271316
Zappia Giuseppe, matr. 268784

Docenti:
Prof. Manco Giuseppe
Prof. Pisani Francesco Sergio

Obiettivi del Progetto e Definizione del Task

- **Il Task**: sviluppo di un modello di Instance Segmentation per individuare cavi elettrici e i relativi pixel in scenari complessi.
- **Il Dataset**: utilizzo del dataset TTPLA, composto da 842 immagini di training e 400 immagini di test con risoluzione 700×700.
- **L'output richiesto**: il modello deve restituire per ogni oggetto: lo score, la bounding box, la maschera di segmentazione e le coordinate polari (ρ,θ) della retta coincidente col cavo.



$$\begin{cases} \rho = \sqrt{x^2 + y^2} \\ \theta = arctg\frac{y}{x} \end{cases} \quad \text{con } 0 \leq \theta \leq 2\pi$$

Conversione della maschera in coordinate polari
(ρ = distanza dall'origine, θ = angolo)

Metrica di Valutazione: il successo è misurato tramite il Line Detection Score (LDS), che pondera la qualità della segmentazione con la precisione geometrica dell'angolo:

$$LDS = mAP + mAR + 2 \cdot e^{-0.12 \cdot \Delta\theta}$$

Panoramica Dataset (Adattamento TTPLA)

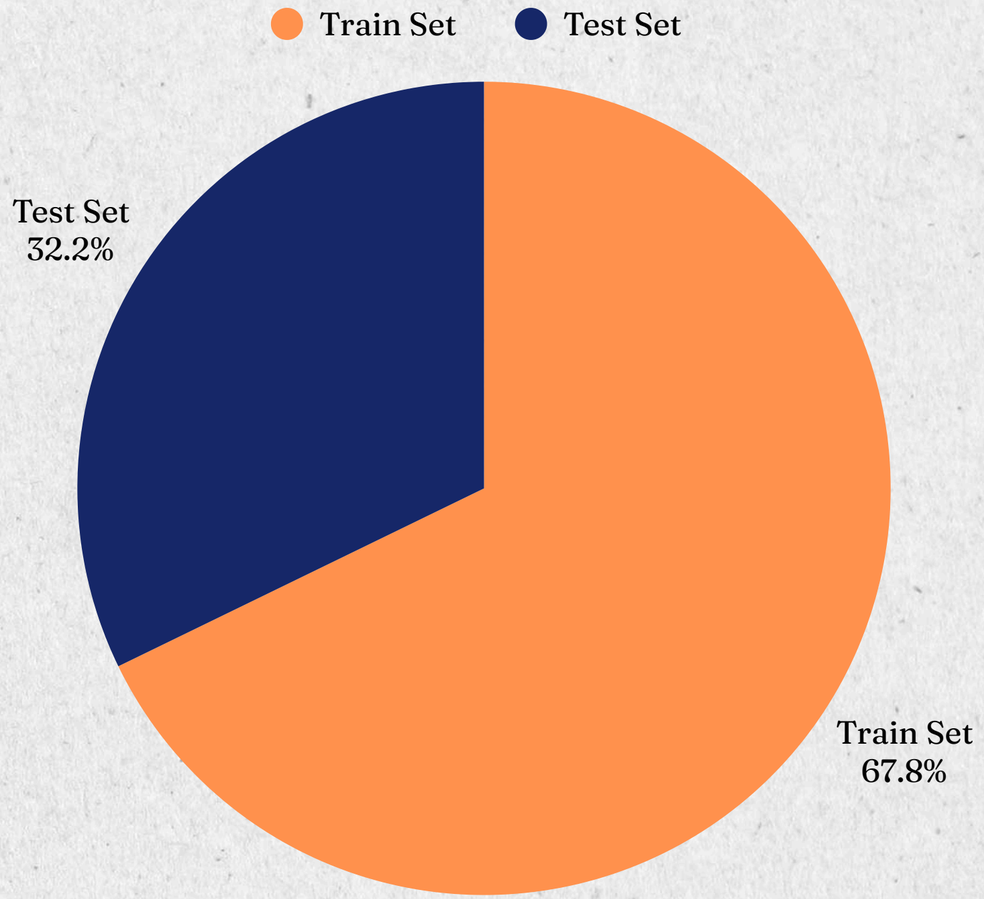
Dati tecnici

- **Fonte**: derivato dal dataset open-source TTPLA (Transmission Tower and Power Line Aerial Images).
- **Preprocessing**: immagini ridimensionate uniformemente a 700x700 pixel.
- **Volumetria totale**: 1.242 immagini aeree catturate tramite UAV (droni) in vari scenari (urbani, rurali, montani).
- **Data split**:

1. Train set: 842 immagini (~68%).
2. Test set: 400 immagini (~32%).

- **Struttura delle annotazioni** (Ground Truth): file train.json e test.json che contengono
  - Maschere di segmentazione (pixel-level).
  - Coordinate geometriche delle rette.

Distribuzione del Dataset



Studio della letteratura

Problematiche emerse dal task:

Il problema del downsampling

Le architetture classiche di segmentazione semantica utilizzano operazioni di pooling che riducono la risoluzione spaziale. La letteratura conferma che questo processo fa "scomparire" i cavi sottili, rendendo impossibile una ricostruzione continua nelle maschere finali

Complessità dello sfondo

Le immagini aeree (come quelle del dataset TTPLA) presentano sfondi "cluttered" (caotici) con vegetazione, strade ed edifici che creano rumore e possono essere confusi con i cavi stessi

Sbilanciamento delle classi

Un problema critico evidenziato in tutti gli studi è che i pixel appartenenti ai cavi rappresentano una frazione minuscola dell'immagine (spesso meno dell'1%) rispetto allo sfondo. Questo porta le reti standard a ignorare i cavi se la loss function non è calibrata correttamente

Discontinuità

Un difetto comune nei modelli è la segmentazione frammentata, dove il cavo viene predetto come una serie di trattini disgiunti invece che come una linea continua

Caratteristiche del target

La letteratura definisce i cavi elettrici come oggetti "estremamente sottili", che spesso occupano una larghezza di soli 1-3 pixel nelle immagini aeree

Analisi Comparativa e Selezione del Modello

Esperimenti Preliminari

Sono state valutate diverse architetture allo Stato dell'Arte, equipaggiate con backbone profonde (es. ResNet101, ViT-L) per massimizzare l'estrazione di feature e massimizzare la capacità di apprendimento.

Alcune delle architetture che hanno prodotto risultati i **risultati migliori** sono:

One-Stage Detectors:
-YOLO11x-seg
-YOLO26x-seg

Semantic Segmentation:
UNet++ con backbone ResNet101

Foundation Models:
Pipeline ibrida Bbox YOLO + SAM (Segment Anything Model) con backbone ViT-L (Vision Transformer Large)

Limitazioni operative riscontrate nelle architetture testate

YOLO Series

Nonostante l'alta velocità, i modelli hanno mostrato una scarsissima precisione sui bordi. L'approccio basato su Bounding Box include eccessivo rumore di fondo per oggetti diagonali e sottili, abbassando la Precision

UNet++

Ha sofferto di bassa Recall e frammentazione delle maschere, dovuta alla perdita di informazioni spaziali durante il downsampling e alla difficoltà nel gestire il forte sbilanciamento delle classi (background vs cavi)

SAM (ViT-L)

Sebbene potente su oggetti generici, ha fallito nel catturare la struttura fine (1-3 px) dei cavi. Anche guidato dai BBox ottenuti tramite Yolo11x, il modello tende a perdere la connettività del cavo o a includere artefatti dello sfondo

Inefficacia del Post-Processing

I tentativi di raffinamento (Scheletrizzazione, Hough Transform) non sono riusciti a ricostruire le geometrie perse a causa di maschere di segmentazione iniziali troppo discontinue o rumorose.

Scelta del Modello Finale: Mask R-CNN + PointRend

I tentativi precedenti hanno fallito a causa del downsampling aggressivo che cancella gli oggetti sottili (1-3 pixel) e della complicata distinzione tra istanze sovrapposte. Da qui nasce una **necessità**: Avere un'architettura che garantisca Instance Segmentation (per separare i cavi) mantenendo però la risoluzione pixel-perfect sui bordi.

L'APPROCCIO TEORICO: POINT-BASED RENDERING

1

Superare la "Griglia Fissa"

Le reti standard (come Mask R-CNN vanilla) usano un'operazione chiamata RoIAlign che proietta l'oggetto rilevato su una griglia di dimensioni fisse, solitamente 28×28 pixel. Per oggetti grandi, perdere dettagli riducendo non è grave. Per un cavo spesso 1-2 pixel, la compressione in una griglia 28x28, è un fenomeno di distruttivo. Matematicamente, le feature del cavo vengono "mediate" con quelle del cielo circostante, facendo scomparire il segnale che diventa "invisibile" o spezzato.

Scelta del Modello Finale: Mask R-CNN + PointRend

L'APPROCCIO TEORICO: POINT-BASED RENDERING

2

La Strategia di "Point Selection"

PointRend non calcola la maschera su tutti i pixel (che sarebbe lento), ma sceglie dove guardare. La rete fa una previsione grossolana, il modello calcola quali punti della maschera sono "incerti" (con probabilità vicina a 0.5). Per i cavi, questi punti corrispondono ai bordi del filo. Durante il training, il modello seleziona un certo numero di punti, concentrandosi proprio su questi bordi difficili, ignorando le vaste aree di cielo "facili".

3

Estrazione di Feature

Una volta identificati i punti "difficili" (i bordi del cavo), PointRend deve decidere se sono cavo o sfondo, per farlo, non usa le feature compresse della griglia 28x28 ma va prelevare il vettore delle caratteristiche corrispondente a quelle esatte coordinate x,y dalle mappe ad alta risoluzione della backbone.

Architettura nel Dettaglio

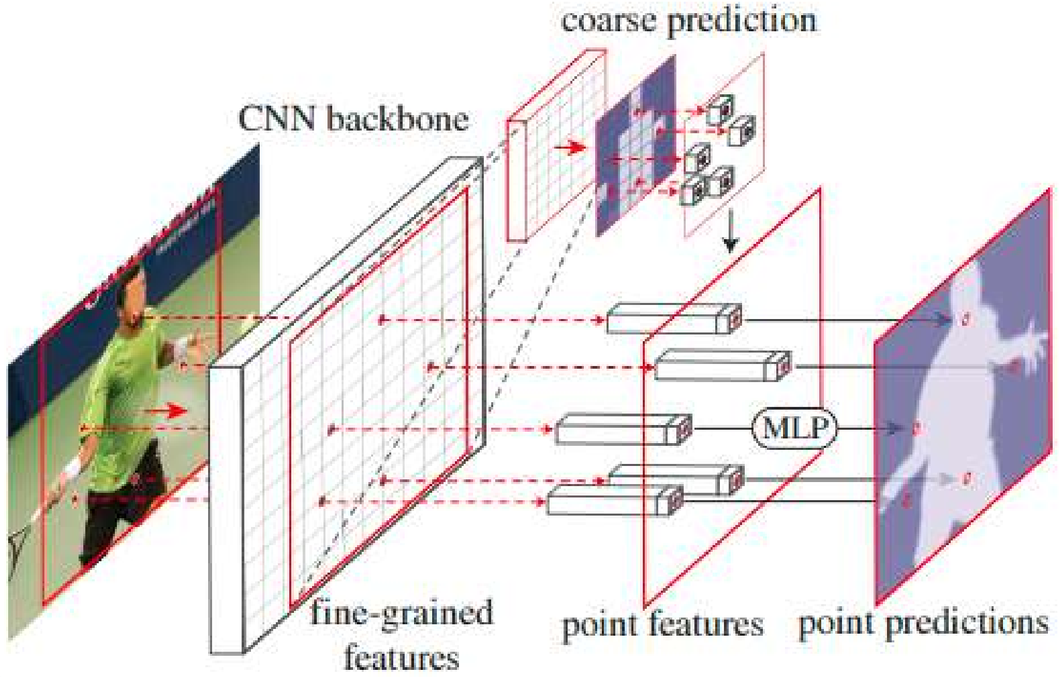
Viene utilizzato **Detectron2**, sviluppato da Facebook AI Research (FAIR) e basato su PyTorch. È una libreria open-source riconosciuta come uno degli standard industriali più avanzati per l'implementazione rapida e flessibile di algoritmi di object detection e instance segmentation.

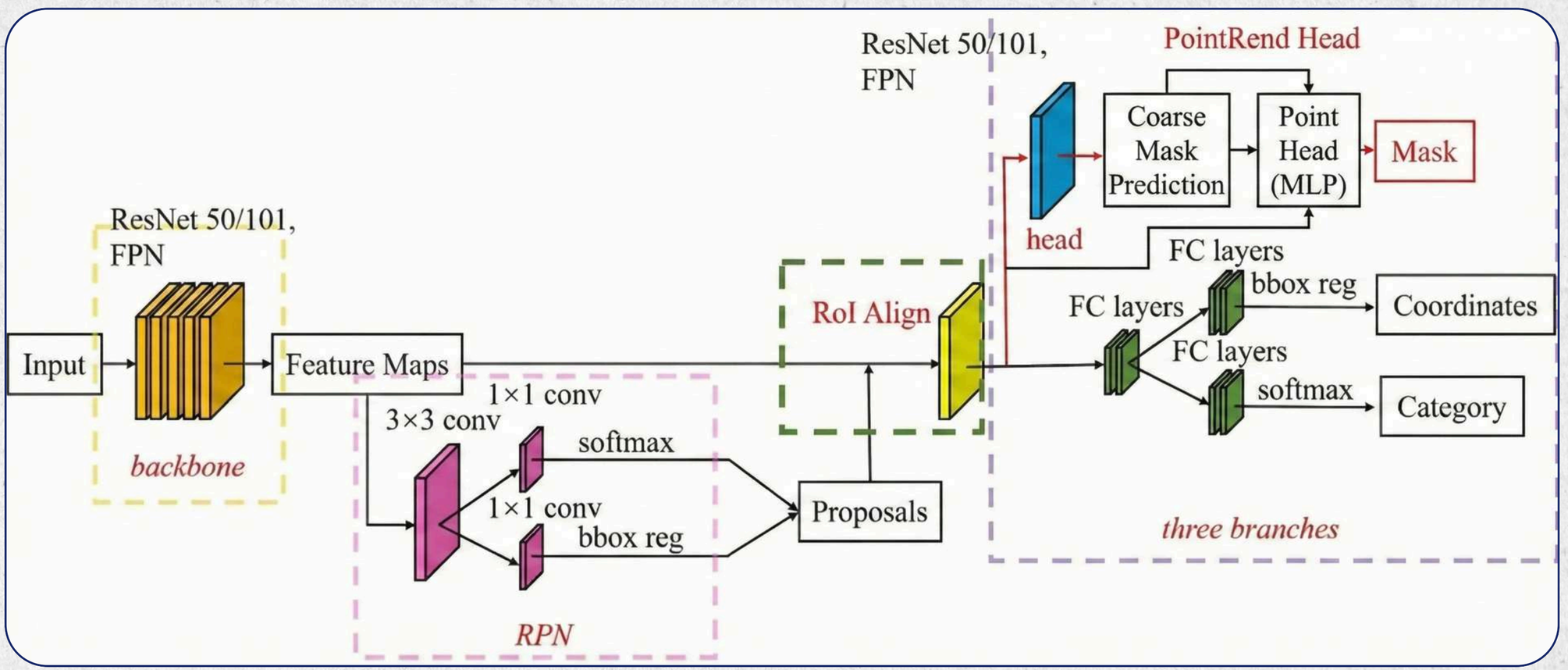
Il modello adottato è basato sulla configurazione avanzata PointRend (**pointrend_rcnn_R_50_FPN_3x_coco.yaml**).
3x schedule indica che la strategia di training prevede una durata tripla rispetto alle impostazioni base, garantendo una convergenza ottimale dei pesi e una maggiore robustezza del modello finale nel generalizzare su immagini complesse.

Nello specifico, l'architettura si fonda su una backbone ResNet-50 arricchita da una **Feature Pyramid Network** (**FPN**), che permette l'estrazione multiscala delle feature. A questa base è stata integrata una Custom Head che utilizza una funzione di perdita ibrida per massimizzare la precisione sui dettagli fini rispetto allo sfondo.

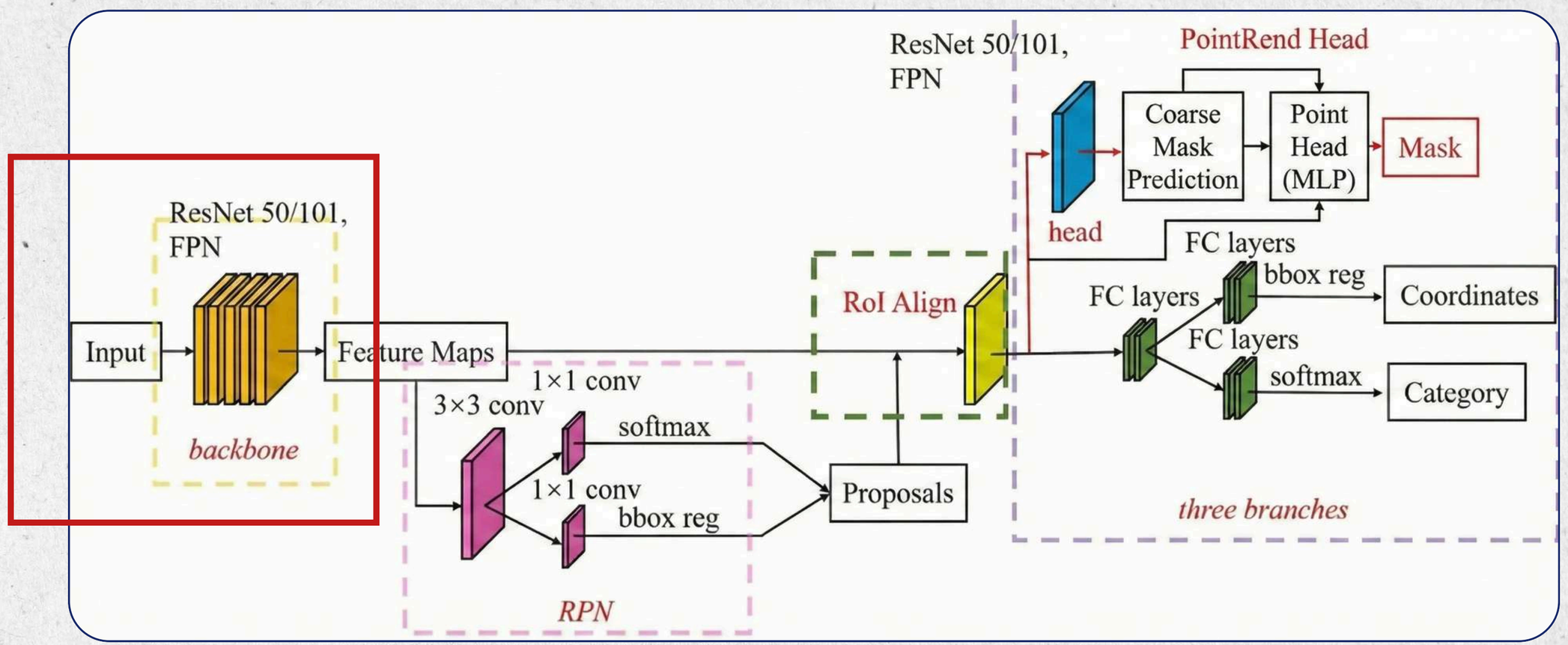
Mask R-CNN

Mask R-CNN rappresenta l'evoluzione naturale di Faster R-CNN, estendendo le capacità del modello originale per includere la **Instance Segmentation**. A differenza dei modelli che si limitano al rilevamento tramite bounding box, Mask R-CNN aggiunge una "testa" (detta anche *branch*, oltre a quelle già presenti di classificazione e bounding box) dedicata alla predizione di una **maschera binaria** per ogni oggetto, permettendo di identificare i pixel appartenenti a ciascuna istanza rilevata (instance segmentation).





Schema basato sull'architettura **Mask R-CNN**: il diagramma illustra i componenti fondamentali del framework, partendo dal Backbone (**ResNet/FPN**) per l'estrazione delle feature map, passando per la RPN (Region Proposal Network), fino al cruciale livello RoI Align. Infine PointRend per la generazione della maschera. Vediamo nel dettaglio come funziona ogni componente utilizzata



Backbone: ResNet 50 con FPN

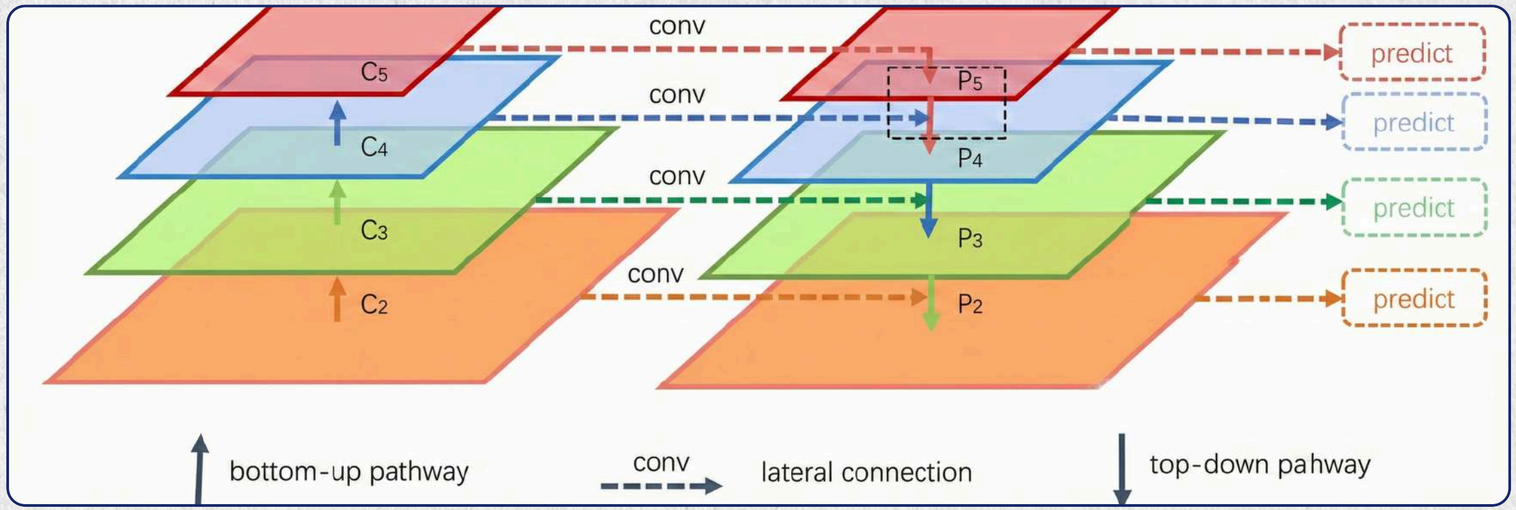
- **Input**: l'immagine passa attraverso una CNN (ResNet-50 o ResNet-101) che agisce come un estrattore di caratteristiche fondamentale: attraverso una serie di blocchi convoluzionali residui, trasforma i dati grezzi dei pixel in rappresentazioni matematiche complesse, catturando pattern visivi che vanno dai semplici bordi a strutture semantiche più articolate.
- **FPN**: permette di costruire una gerarchia di feature maps a diverse risoluzioni.
- **Feature Maps**: in output vengono prodotte una serie di feature maps contenenti informazioni dell'immagine catturate a differenti livelli.

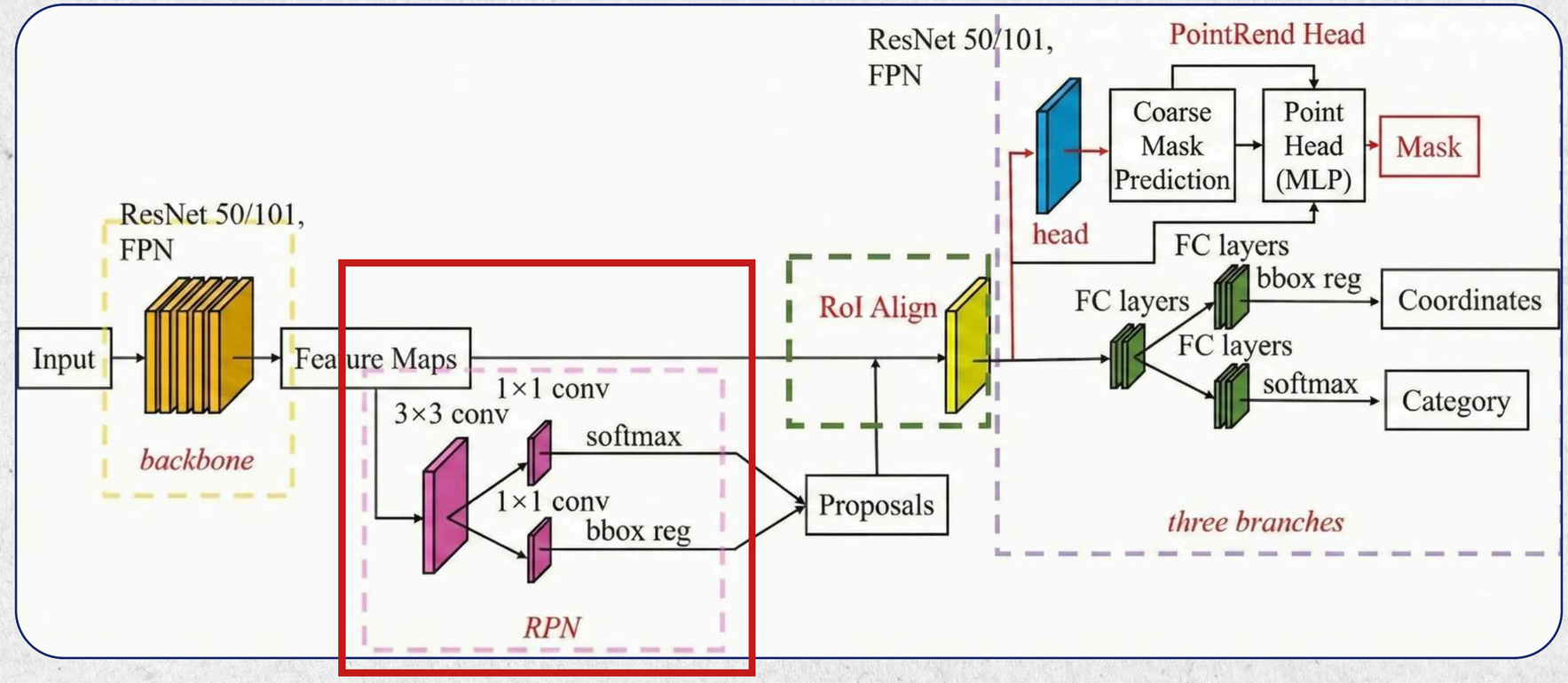
La **Feature Pyramid Network** (**FPN**) è un'architettura a piramide progettata per permettere al modello di riconoscere oggetti di dimensioni estremamente variabili, sfruttando mappe di caratteristiche estratte a scale differenti. Il suo funzionamento si basa sull'integrazione di due percorsi fondamentali:

- Percorso **Bottom-up**: in questa fase, la FPN utilizza la backbone (ResNet-50) per estrarre feature a profondità crescenti. Mentre i livelli iniziali preservano dettagli ad alta risoluzione spaziale (essenziali per individuare cavi sottili o lontani), i livelli più profondi distillano informazioni ad alta astrazione semantica, necessarie per identificare la struttura e il contesto globale degli oggetti più grandi.
- Percorso **Top-down** e Connessioni Laterali: attraverso un processo di "feature fusion", il percorso top-down campiona verso l'alto (upsampling) le mappe semantiche più profonde e le combina con quelle a risoluzione maggiore provenienti dai livelli precedenti tramite connessioni laterali.

Questa sinergia permette di generare in output una serie di feature maps multiscala in cui ogni livello della piramide possiede sia una forte componente semantica che un'elevata precisione geometrica. Tale struttura è il prerequisito fondamentale per le fasi successive di segmentazione, garantendo che il modello mantenga una sensibilità costante sia sui dettagli fini che sulle forme generali dei cavi presenti nell'immagine.

FPN



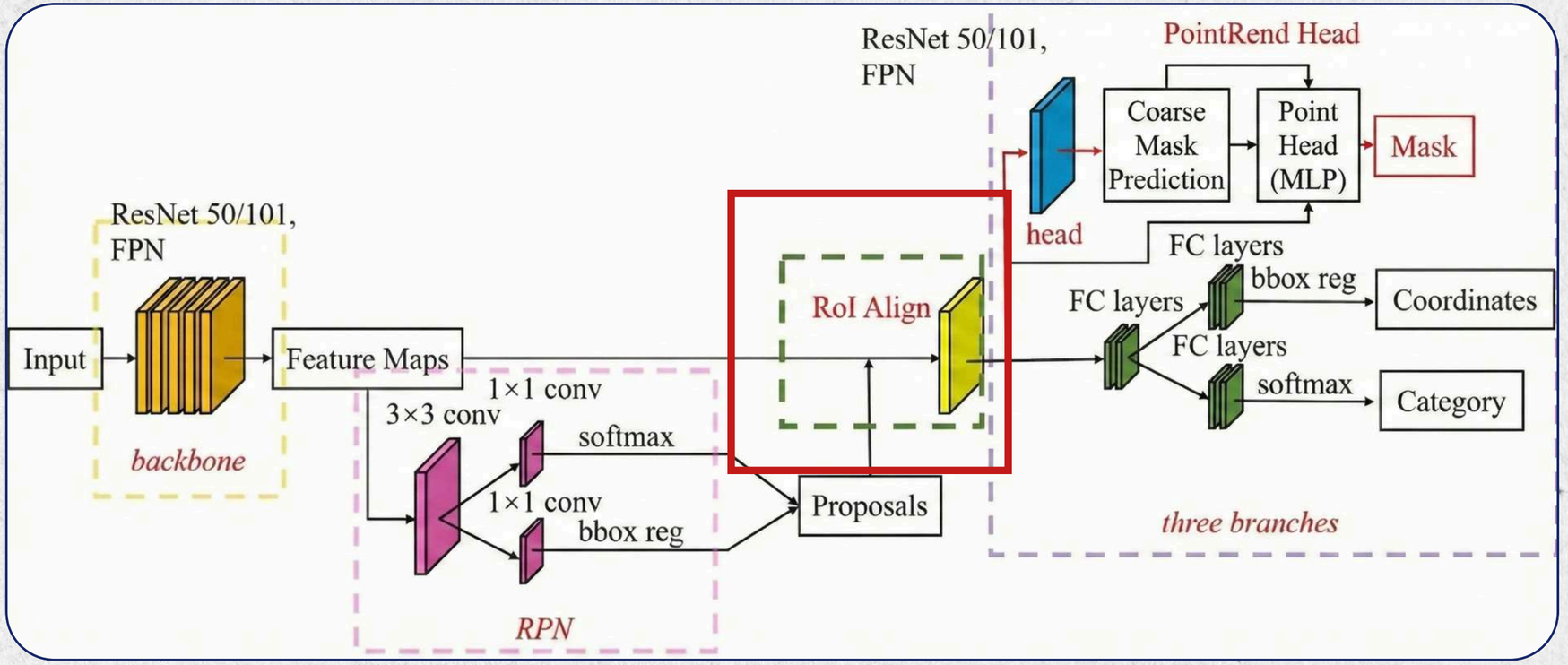


Region Proposal Network (RPN)

- Sfrutta le feature maps ottenutte dal blocco precedente per generare proposte di potenziali regioni che contengono oggetti. Per questo task utilizza:
- Convoluzioni

1. •3x3: Layer applicato per elaborare ulteriormente le feature maps e produrre rappresentazioni più compatte delle regioni.
2. •1x1: Due convoluzioni 1x1 che prevedono due differenti calcoli: softmax e bbox reg. La prima calcola la probabilità della presenza di un oggetto in una data regione (classificando le regioni come oggetti o sfondo), mentre la seconda calcola le coordinate della bounding box ottimizzando la posizione proposta dalla regione.

L'output della RPN è un insieme di **regione proposte** (proposals) che sono candidate a contenere oggetti.



Il modulo **RoI Align** rappresenta un'evoluzione critica rispetto al tradizionale RoI Pooling (usato in Fast e Faster-RCNN). Viene introdotto per risolvere il problema del disallineamento spaziale nelle reti di segmentazione. Mentre il RoI Pooling introduce errori di approssimazione dovuti alla quantizzazione (ovvero l'arrotondamento delle coordinate sulle mappe di feature), il RoI Align preserva l'integrità geometrica dei dati attraverso l'uso dell'interpolazione bilineare.

Nello specifico, questa operazione permette di campionare i valori delle feature in posizioni sub-pixel precise, garantendo che le regioni proposte (proposals) siano perfettamente allineate con le caratteristiche estratte dalla backbone. Tale accuratezza è determinante per il successo della segmentazione dei cavi: data la natura estremamente sottile di questi oggetti, anche un minimo errore di allineamento di pochi pixel comporterebbe una perdita di dettaglio fatale, compromettendo la capacità del modello di delineare contorni esatti e continui.

PointRend

PointRend (**Point-based Rendering**) è un modulo di rete neurale che tratta la segmentazione delle immagini non come una classificazione di pixel su una griglia fissa, ma come un problema di rendering grafico: invece di calcolare il valore di ogni singolo pixel su una griglia ad alta risoluzione (che è computazionalmente costoso), calcola le etichette solo in posizioni selezionate adattivamente dove è necessario **maggior dettaglio**

Selezione dei Punti (Point Selection)

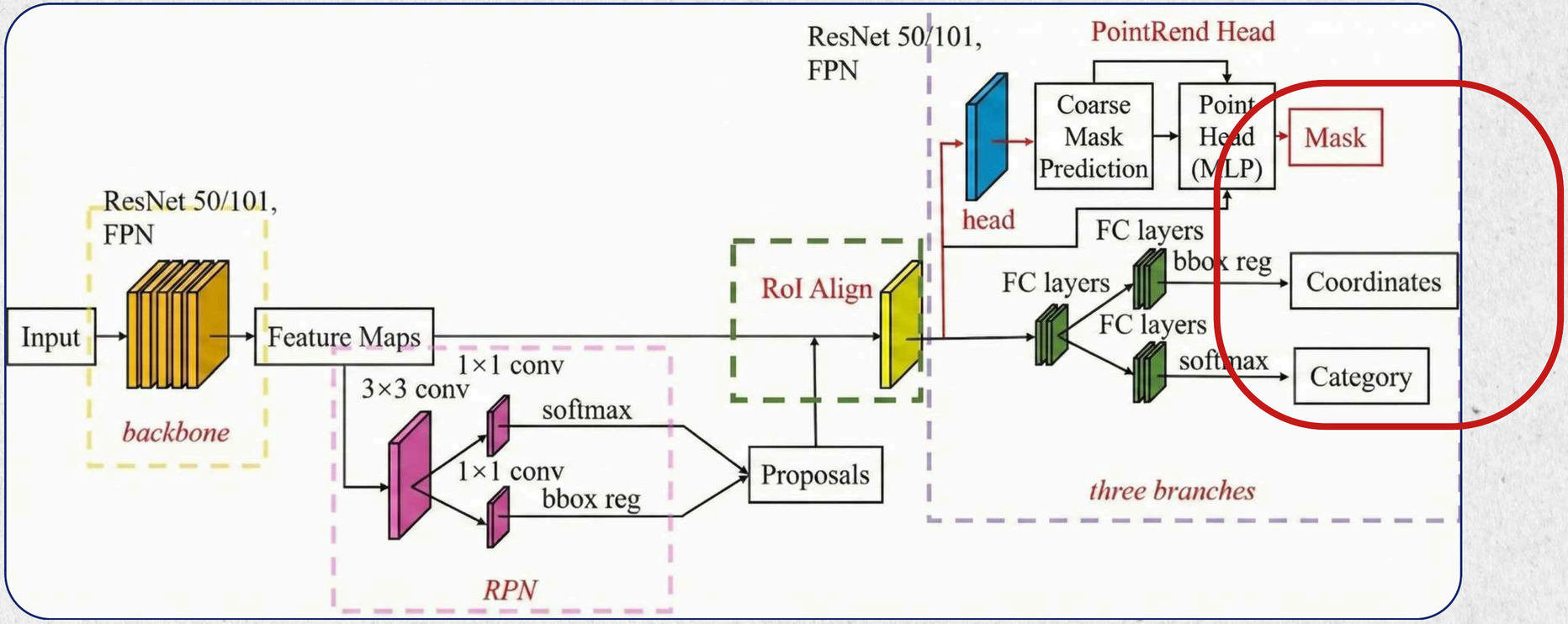
Il sistema sceglie un numero ridotto di punti su cui effettuare la predizione. Durante l'inferenza, usa una strategia di "suddivisione adattiva" per concentrarsi sulle aree **incerte** (spesso i bordi degli oggetti)

Rappresentazione Puntuale (Point-wise Representation)

Per ogni punto selezionato, estrae un vettore di feature usando l'interpolazione bilineare dalle feature map della CNN (che sono su griglia regolare). Queste feature combinano dettagli fini (che vengono dalle feature maps della CNN, come il livello P2) e predizioni grossolane (che vengono dalla Coarse Mask predetta inizialmente dal modello)

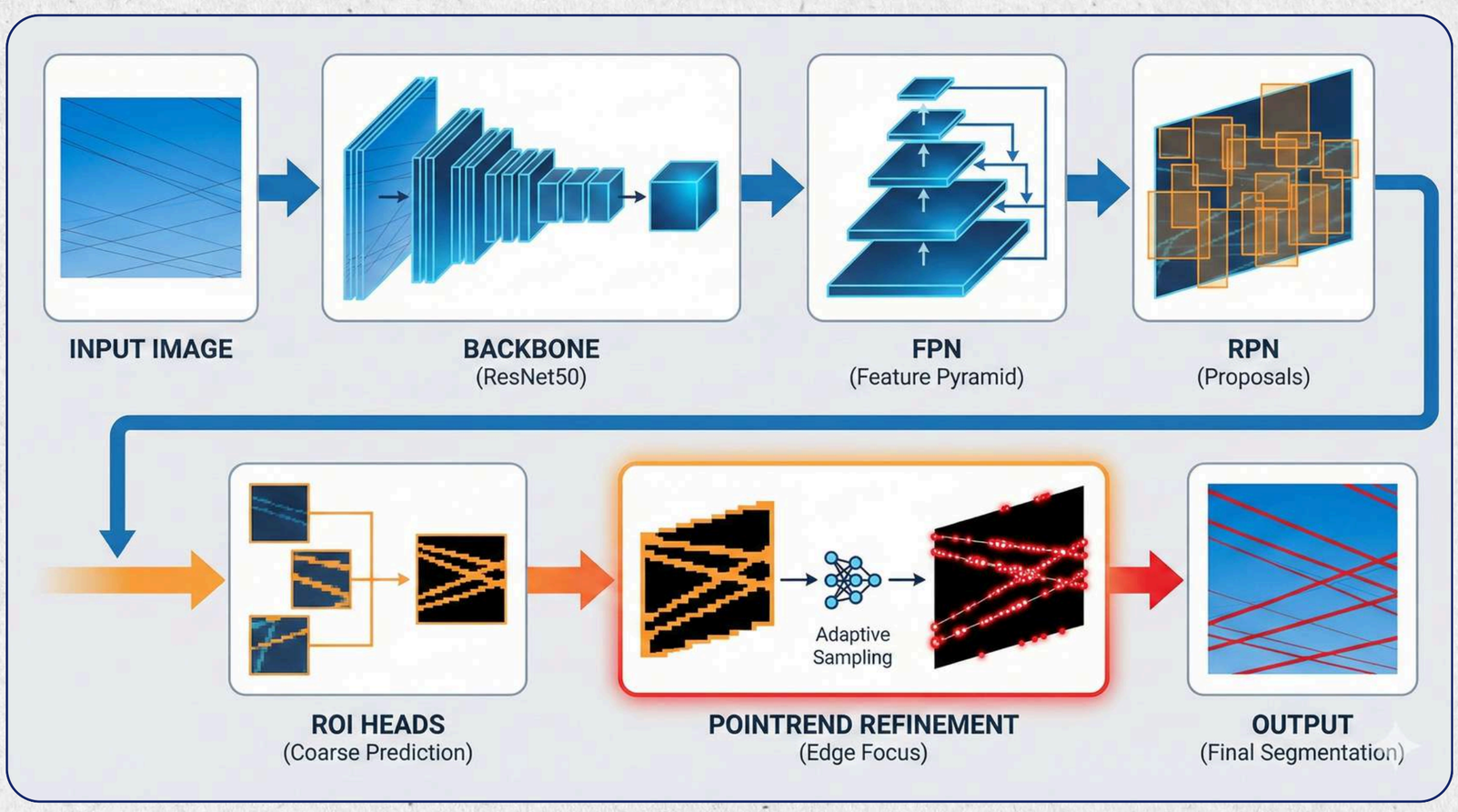
Point Head

Una piccola **rete neurale** (MLP - Multi-Layer Perceptron) predice l'etichetta di segmentazione per ciascun punto indipendentemente, partendo dalle feature estratte



- **Mask Head**: Testa per la generazione della maschera binaria tramite Pointrend.
- **Bounding Box Branch**: una serie di strati convoluzionali per la regressione delle coordinate (posizione e dimensione della box).
- **Category Branch**: una serie di strati convoluzionali e una funzione softmax che classificano l'oggetto

Confronto qualitativo: a sinistra la **segmentazione standard** con Mask R-CNN, a destra quella ottenuta con **PointRend**. Mantenendo la stessa base (ResNet-50 + FPN), l'approccio PointRend garantisce una definizione nettamente superiore sui bordi e sui dettagli fini dell'oggetto.



Pipeline di segmentazione basata su PointRend con backbone ResNet50 e FPN. Il diagramma illustra il processo di raffinamento iterativo delle maschere: dalla predizione grossolana (Coarse Prediction) al recupero dei dettagli fini sui bordi dei cavi tramite campionamento adattivo (Adaptive Sampling)

Architettura: Configurazione Custom PointRend

Architettura Base & BackBone

Modello: PointRend R-CNN
Backbone: ResNet-50-FPN (Feature Pyramid Network)
Pesi Iniziali: Pre-trained COCO

Anchor Generator

Aspect Ratios: [0.2, 0.5, 1.0, 2.0, 5.0] (Inclusi rapporti 1:5 e 5:1)

Custom Heads

Mask Head:
CustomCableMaskHead (classe derivata da PointRendMaskHead)
Point Head:
CustomCablePointHead (classe derivata da StandardPointHead)
Point Head Input Features:
Livello p2 della FPN
Point Head Structure: 3 Fully Connected Layers, dimensione 256
Sampling Strategy (Train): 2048 punti, Oversample Ratio 3.0, Importance Sample Ratio 0.75

Loss Function

Tipologia: Ibrida (Custom Loss)
Composizione: Dice Loss + Weighted Sigmoid Focal Loss
Formula: Loss= LossDice +(5.0×LossFocal)
Parametri Focal: alpha = 0.95, gamma = 2.0

Strategie di Data Augmentation

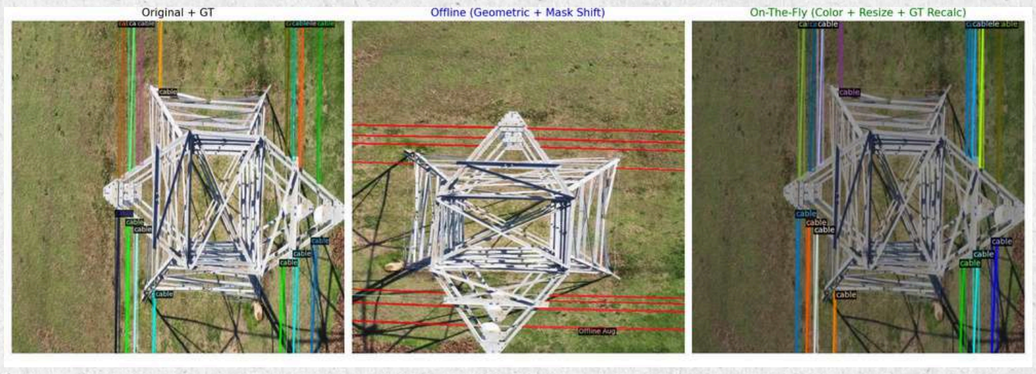
Al fine di irrobustire il modello a fronte di un numero ridotto di campioni (842), il training set è stato arricchito tramite una pipeline che integra la generazione sintetica preliminare (Offline) con trasformazioni in tempo reale durante l'addestramento (Online).

Data Augmentation Offline (Statica)

- **Obiettivo**: Moltiplicazione sicura del dataset (5x) (da 842 a 4.210 immagini) preservando la geometria dei "Thin Objects".
- **Tecnica**: Trasformazioni discrete e non distruttive:
  1. Rotazioni fisse a 90°, 180°, 270° (Nessuna interpolazione/aliasing sui cavi).
  2. Flip Orizzontale e Verticale.
- **Vantaggio**: Il modello impara l'invarianza alla rotazione senza perdere la definizione del pixel (fondamentale per linee di 1-3px).

Data Augmentation On-the-Fly (Dinamica)

- **Obiettivo**: Robustezza ambientale e generalizzazione.
- **Tecnica**: Augmentation applicata in tempo reale ad ogni epoca:
  1. Photometric Distortions (Luminosità, Contrasto, Saturazione) per simulare diverse condizioni meteo/luce.
  2. Ridimensionamento casuale del lato corto (ResizeShortestEdge) per implementare il Multi-scale Training.
- **Vantaggio**: Il modello non vede mai due volte la stessa identica immagine ("Infinite Dataset variation").



Data Augmentation: esempi di trasformazioni sintetiche (rotazioni, variazioni cromatiche, flip) applicate al training set. Questa strategia espande la varietà dei dati, rendendo il modello robusto a diverse condizioni di luce e prospettiva e prevenendo l'overfitting.

Strategia di Training & Ottimizzazione

Scelta degli iper-parametri

Batch Size: 4

Fissata pari al massimo supportato dalla GPU prima di andare in Out-Of-Memory

Scheduler lr: 32 e 39sima epoca

Gli step di decadimento del Learning Rate (a 34k e 41k) corrispondono matematicamente alla 32esima e 39esima epoca, seguendo la regola aurea di ridurre il learning rate per l'ultimo 10-25% del training per raffinare i minimi locali.

Learning Rate: 0.0025

Abbiamo calcolato un punto di partenza teorico ottimale usando la Linear Scaling Rule cioè lr_new=lr_old*(BS_new/BS_old). Partendo dal LR canonico di Detectron2 (0.02 su batch 16), scalando linearmente a batch 4 avremmo dovuto usare 0.005. Tuttavia, trattandosi di un task di Fine-Tuning e non di training from scratch, abbiamo dimezzato per evitare oscillazioni distruttive sui pesi pre-addestrati della ResNet

Numero Iterazioni: 45000

Il calcolo è stato derivato dalla dimensione del dataset aumentato e dalla Batch Size. Questo produce circa 1.000 iterazioni per epoca. Impostando il limite a 45.000, abbiamo targettizzato esattamente 43 epoche di addestramento.
In letteratura (specialmente in Detectron2), questo corrisponde alla cosiddetta '3x Schedule' (che tipicamente si aggira sulle 36-40 epoche), considerata lo standard per massimizzare la mAP su task difficili.

Strategia di Loss Ibrida

La sfida: Class Imbalance nei cavi sottili

È stata adottata una strategia di Model Selection per selezionare la loss function e gli iperparametri ottimizzando le risorse (GPU time).

- **Setup**: Subset casuale del 20% (~170 immagini), senza data augmentation.
- **Durata**: 2.000 iterazioni (~30 min/esperimento).
- **Obiettivo:** Valutare la migliore configurazione la funzione di loss.

Configurazione	Osservazioni	Esito
P1: Cross Entropy	Convergenza lenta, difficoltà evidente sui cavi sottili.	Scartato
P2: Dice Loss Pure	Gradiente instabile con forti oscillazioni.	Scartato
P3: Focal + Dice (1:1)	Componente Focal numericamente trascurabile (<1e-3).	Ricalibrato
P4: Focal (x5) + Dice	Convergenza stabile. Il peso x5 bilancia il recupero dei dettagli.	**Selezionato**

Strategia di Loss Ibrida

La sfida: Class Imbalance nei cavi sottili

In un task di cable detection, i cavi sono strutture filiformi che occupano una porzione minima dell'immagine (<1%) rispetto allo sfondo. Una loss standard fallirebbe, classificando tutto come "sfondo" per minimizzare l'errore medio.

Focal Loss

- **A cosa serve**: Risolve lo sbilanciamento tra classi (background vs cavo).
- **Perché sceglierla**: "Abbassa il volume" sugli esempi facili (lo sfondo uniforme) e costringe la rete a concentrarsi sui pixel difficili (i cavi), migliorando la precisione dei dettagli.

Dice Loss

- **A cosa serve**: Ottimizza la sovrapposizione globale (IoU) tra predizione e realtà.
- **Perché sceglierla**: Lavora sull'intera forma dell'oggetto, non sui singoli pixel. È fondamentale per mantenere la continuità del cavo ed evitare segmentazioni spezzettate.

La funzione di costo globale sarà quindi definita come una combinazione lineare pesata delle due componenti:

$$L_{total} = 5 \cdot L_{focal} + L_{dice}$$

Analisi del training e risultati

Andamento della **loss** durante il training; i pesi sono stati salvati ogni **5000** iterazioni per prevenire eventuali interruzioni dovurte alla saturazione delle risorse GPU disponibili.

Post-Processing: Configurazione dell'Inferenza e Stima Geometrica

Parametri Fissi di Inferenza:

- Manteniamo coerenti gli Aspect Ratios [0.2, 0.5, 1.0, 2.0, 5.0] usati in training per gestire la natura allungata dei cavi.
- Risoluzione di test fissata a 700x700 senza cropping, per garantire che il modello veda l'immagine intera come durante il training.
- Impostiamo DETECTIONS_PER_IMAGE = 500 (molto alto) per evitare di perdere cavi in immagini dense di tralicci.

Coordinate Polari:

- L'output primario è una maschera binaria.
- Per ottenere le rette (ρ,θ) che individuano i cavi, usiamo una regressione sui pixel attivi della maschera:

1. Algoritmo: cv2.fitLine con distanza L2 (Euclidea).
2. Conversione: dal vettore direttore (vx,vy) e punto (x0,y0) ai parametri polari normalizzati.

Filosofia del Post-Processing:

- L'obiettivo non è solo la detection, ma massimizzare la metrica LDS:

$$AP + AR + 2 \cdot AngleDiff$$

- Il post-processing serve a raffinare l'output grezzo per allinearlo geometricamente alla Ground Truth e ottimizzare il trade-off Precision/Recall.

Scelta degli Iperparametri

L’obiettivo è cercare di **massimizzare LDS** tramite una giusta scelta di **Score Threshold** (confidenza) e **NMS Threshold** (sovrapposizione).

Dall'analisi visiva preliminare sul training set, abbiamo notato che il modello, pur individuando correttamente la posizione dei cavi, tendeva ad assegnare loro confidence score bassi a causa della loro natura estremamente **sottile** e della **scarsa visibilità rispetto allo sfondo**.

Inoltre, per come è stato implementato il calcolo dell’angolo è necessario che l'oggetto venga rilevato. Un False Negative comporta un punteggio nullo in questo termine dominante

Abbiamo quindi scelto deliberatamente una soglia molto permissiva **(0.01)** per massimizzare la **Recall**. La logica è che per il calcolo dell'LDS è più penalizzante perdere un cavo esistente che rilevarne uno con incertezza.

Al contrario, una rilevazione con confidenza bassa, pur rischiando di introdurre rumore nella maschera, permette comunque di estrapolare l'equazione della retta. Dato che i cavi sono oggetti rigidi lineari, l'**angolo** Theta estratto è spesso **accurato** anche con maschere sub-ottimali. Pertanto, abbassare la soglia a 0.01 è una strategia per massimizzare la Recall e garantire che ogni cavo contribuisca al calcolo dell'errore angolare, accettando un lieve degrado della precisione pura che è meno impattante sul punteggio finale totale

Scelta degli Iperparametri

L’obiettivo è cercare di **massimizzare LDS** tramite una giusta scelta di **Score Threshold** (confidenza) e **NMS Threshold** (sovrapposizione).

La scelta di un NMS **alto (0.7)** deriva dalla natura geometrica specifica del problema dei cavi elettrici.
I cavi nelle immagini presentano spesso due caratteristiche: Corrono paralleli e molto vicini tra loro. Si incrociano prospetticamente.
Una soglia NMS standard (es. 0.5 o 0.3) è troppo aggressiva e rischierebbe di sopprimere un cavo reale solo perché si sovrappone parzialmente a un altro o gli passa molto vicino. Alzando la soglia a 0.7, permettiamo al modello di mantenere bounding box che si sovrappongono significativamente, preservando così i **cavi adiacenti o incrociati** che altrimenti verrebbero cancellati.

Post-Processing Geometrico: Correzione del Bias Sistematico

Per verificare la presenza di errori sistematici (non casuali), abbiamo eseguito l'inferenza sul Training Set. Confrontando le maschere predette con le GT di training, abbiamo cercato un pattern di disallineamento costante.
L'idea alla base è infatti che se il modello introduce uno shift geometrico sui dati di training, è altamente probabile che lo stesso bias si ripresenti sui dati di test.

Fase 1: Raccolta delle Coppie

Per ogni immagine, accoppiamo la maschera predetta M_pred con la maschera reale M_gt che ha l'IoU più alto, creando un dataset di "coppie di validazione".

Fase 2: Grid Search "Brute Force" sui Pixel

- È stato definito uno spazio di ricerca per lo shift (dx,dy) nel range [−3,+3] pixel.
- Per ogni combinazione di shift (i,j):

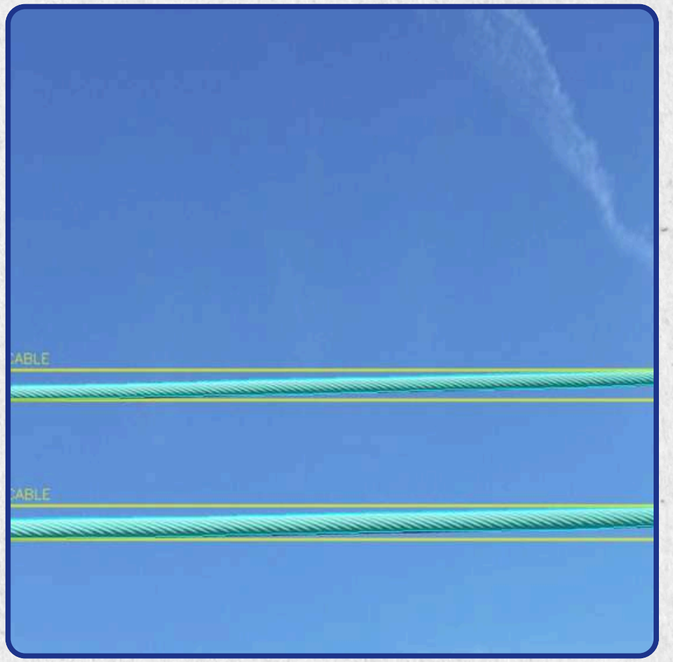
1. Applichiamo la traslazione a tutte le Mpred raccolte: $Mpred' = Shift(Mpred, i, j)$.
2. Ricalcoliamo l'IoU medio globale tra le Mpred′ (spostate) e le Mgt (fisse).

Fase 3: Selezione e Applicazione

La Grid Search posizionale sul training set ha evidenziato che le predizioni erano costantemente traslate rispetto alla realtà.
Lo shift ottimale calcolato per massimizzare l'IoU è risultato essere (+1 px, +1 px)

Come è stato applicato sul test set:

- Si esegue lo Slicing Numpy per spostare i bit della maschera: mask[1:, 1:].
- Viene Rigenerata la Bounding Box: Ricalcola [xmin, ymin, w, h] avvolgendo i nuovi pixel attivi per garantire la coerenza nel formato COCO.
- Viene ricalcolata la Linea: Estrae le coordinate polari dalla maschera post-shift



Esempi di inferenza qualitativa su scenari eterogenei Le immagini mostrano l'output del modello su sfondi a diversa complessità (vegetazione, terreno e cielo). Sono visibili in giallo le **Bounding Box**, che identificano l'area di presenza dell'oggetto, e in ciano le **maschere** di segmentazione, che delineano con precisione la geometria "pixel-perfect" dei cavi, confermando l'efficacia di PointRend nel catturare oggetti sottili.

Line Detection Score (LDS)

```
DONE (t=0.04s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.134
 Average Precision  (AP) @[ IoU=0.50       | area=   all | maxDets=100 ] = 0.355
 Average Precision  (AP) @[ IoU=0.75       | area=   all | maxDets=100 ] = 0.080
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.098
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.362
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.024
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.048
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.189
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.215
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.113
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.402
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.808
100%|██████████| 400/400 [00:45<00:00,  8.86it/s]
Total GT lines: 3328
Total predicted lines: 8742
Total matches: 3130
Lines with coordinate differences computed: 3130
avg_p50=np.float64(0.3549817988313255), avg_r50=np.float64(0.18852163461538457), angle_diff=np.float64(0.9990674934474352)
LDS = 2.5416384203415805
```

Il punteggio riflette la **precisione geometrica** del modello sui parametri polari (ρ e θ).
La retta predetta non è solo vicina: è geometricamente fedele al cavo fisico.
Inoltre, l'uso di una treshold fissata a 0.01 e della NMS a 0.7, garantiscono comunque un buon livello della Precision e della Recall richieste sul task

UNIVERSITÀ DELLA CALABRIA
DIPARTIMENTO DI **INGEGNERIA INFORMATICA, MODELLISTICA, ELETTRONICA E SISTEMISTICA**

22 Gennaio 2026

Corso di Laurea Magistrale in Ingegneria Informatica
Corso di Computer Vision - A.A. 2025/2026

Grazie per l'attenzione!

Studenti:
Martucci Anastasia, matr. 271316
Zappia Giuseppe, matr. 268784

Docenti:
Prof. Manco Giuseppe
Prof. Pisani Francesco Sergio