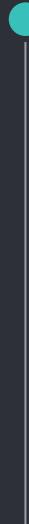


IPCV 2021 - Prof. Giuseppe Scarpa



# Cityscapes Instance Segmentation

Antimo Iannucci  
Fabio d'Andrea  
Antonio Spallone

M63001040  
M63000989  
P38000042

Mario Pace  
Guido Di Chiara

M63000988  
M63000986

- Task del Progetto

- **Instance Segmentation** di immagini rilevate in contesti urbani



Immagine originale



Immagine segmentata

- Semantic Segmentation vs Instance Segmentation



Semantic Segmentation

Istanze multiple della stessa classe sono trattate come singole entità



Instance Segmentation

Istanze multiple della stessa classe sono trattate come entità distinte

The logo for Google Colab, featuring the word "colab" in a bold, orange, sans-serif font inside a solid teal circle.

colab

## Google Colab

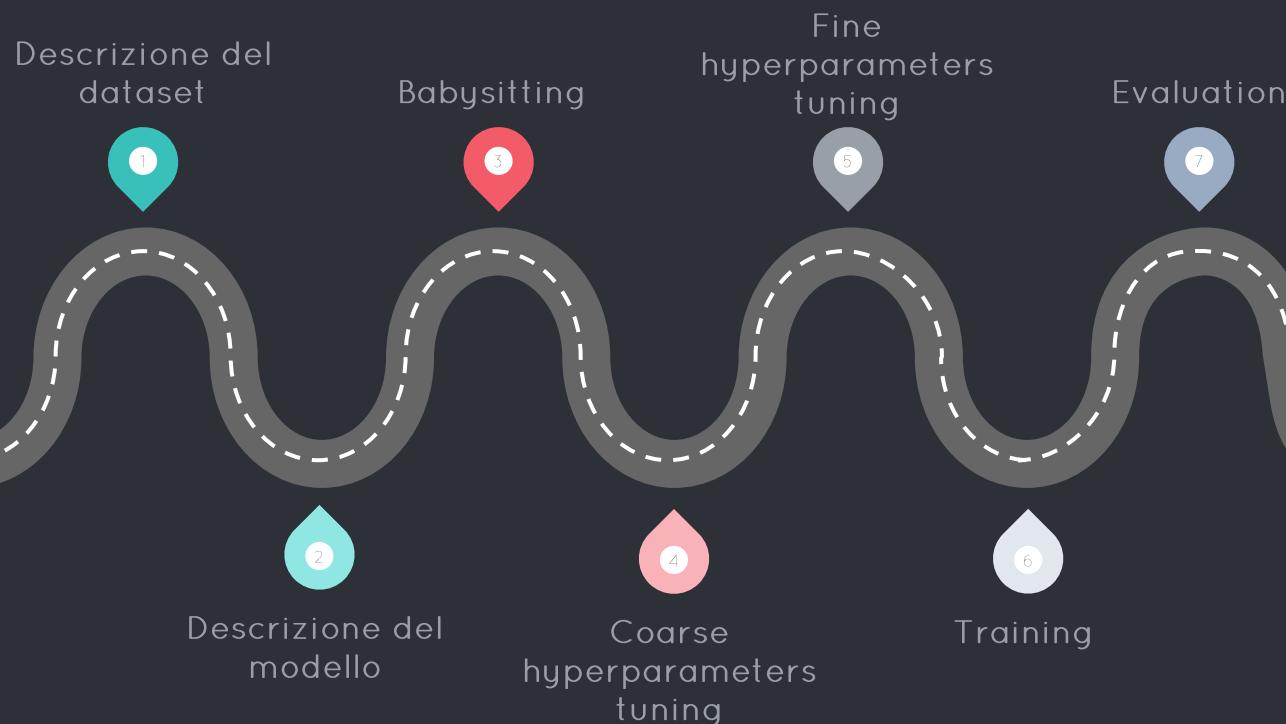
- Tempo di utilizzo GPU limitato
- RAM della GPU limitata a 16 GB
- Spazio su Drive limitato a 15 GB



# Matterport Mask R-CNN

Implementazione dell'architettura **Mask R-CNN**  
realizzata da **Matterport**, costruita su TensorFlow

- Overview del processo



1

# Descrizione del dataset

Analisi dei dati a disposizione e preprocessing

- Dataset originale



Immagine originale  
2048x1024



Label per semantic  
segmentation  
2048x1024



Label per instance  
segmentation  
File json contenente i  
vertici dei poligoni

- Suddivisione del dataset

- Training set

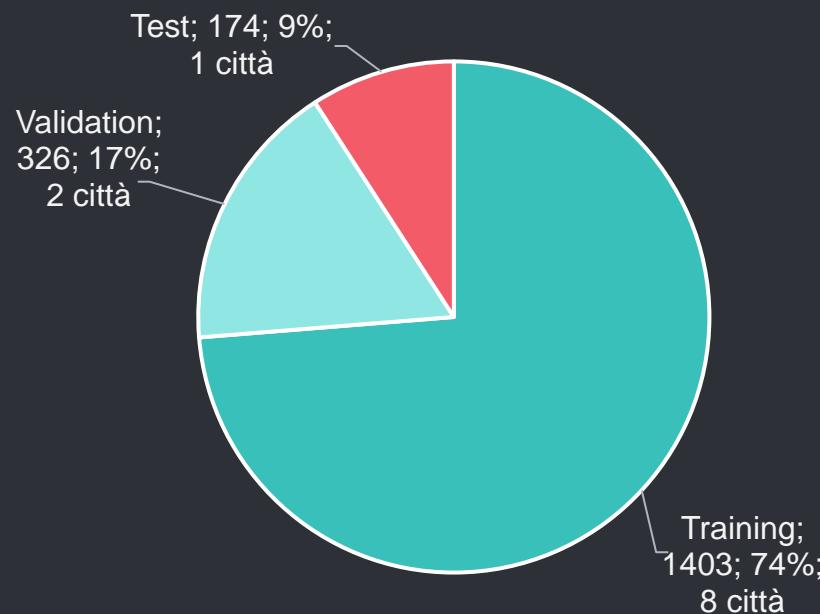
Addestramento del  
modello

- Validation set

Hyperparameters  
tuning

- Test set

Valutazione del  
modello

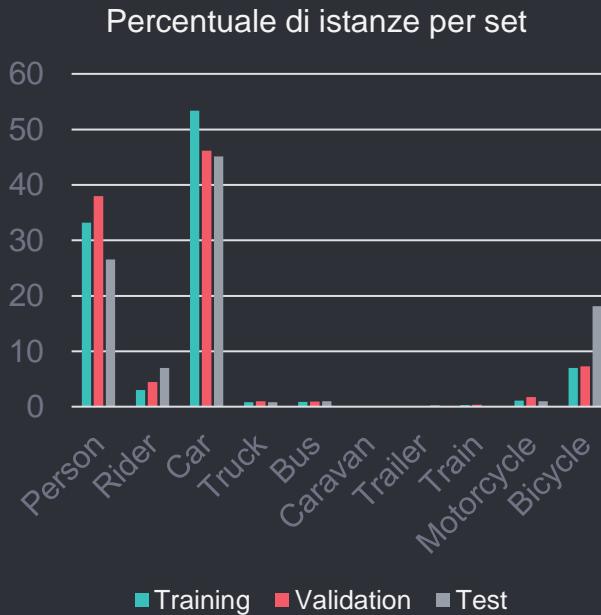
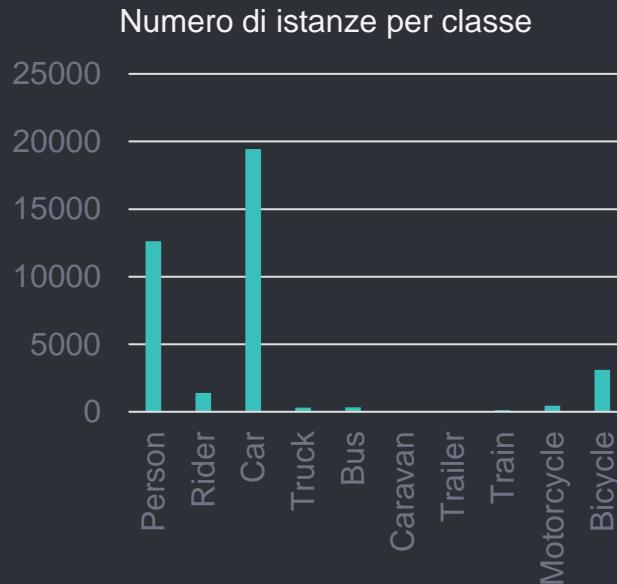






## Distribuzioni delle classi

- Dallo studio della distribuzione si evince come le classi siano ben bilanciate in tutti i set definiti



## • Statistiche

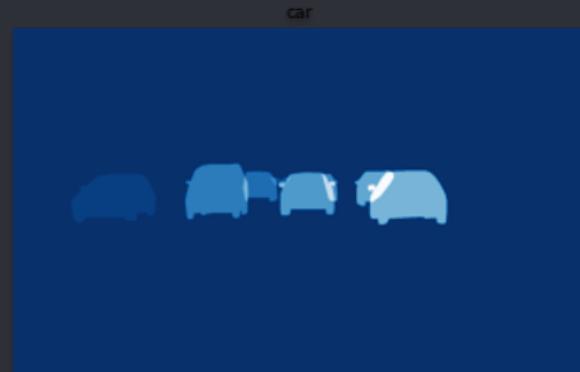
	Training	Validation	Test
Max istanze per immagine	120	66	46
Immagini senza istanze	3	8	0



- Dataset processato: masks
  - A partire dai file json si costruiscono le maschere delle diverse istanze delle classi



Immagine originale



Maschere

- Dataset processato: mini-masks



Mask

Maschere 2048x1024 usate  
per il test del modello



Mini-mask

Maschere 56x56 usate per  
training e validation del  
modello

- Dataset processato: mini-masks
  - A partire dalle mini-masks è possibile ricostruire delle maschere approssimate di 2048x1024 pixel



Maschere originali



Maschere ricostruite dalle  
mini-masks

- Dataset processato: bounding boxes
  - A partire dalle maschere delle istanze si determinano i bounding boxes



Bounding boxes

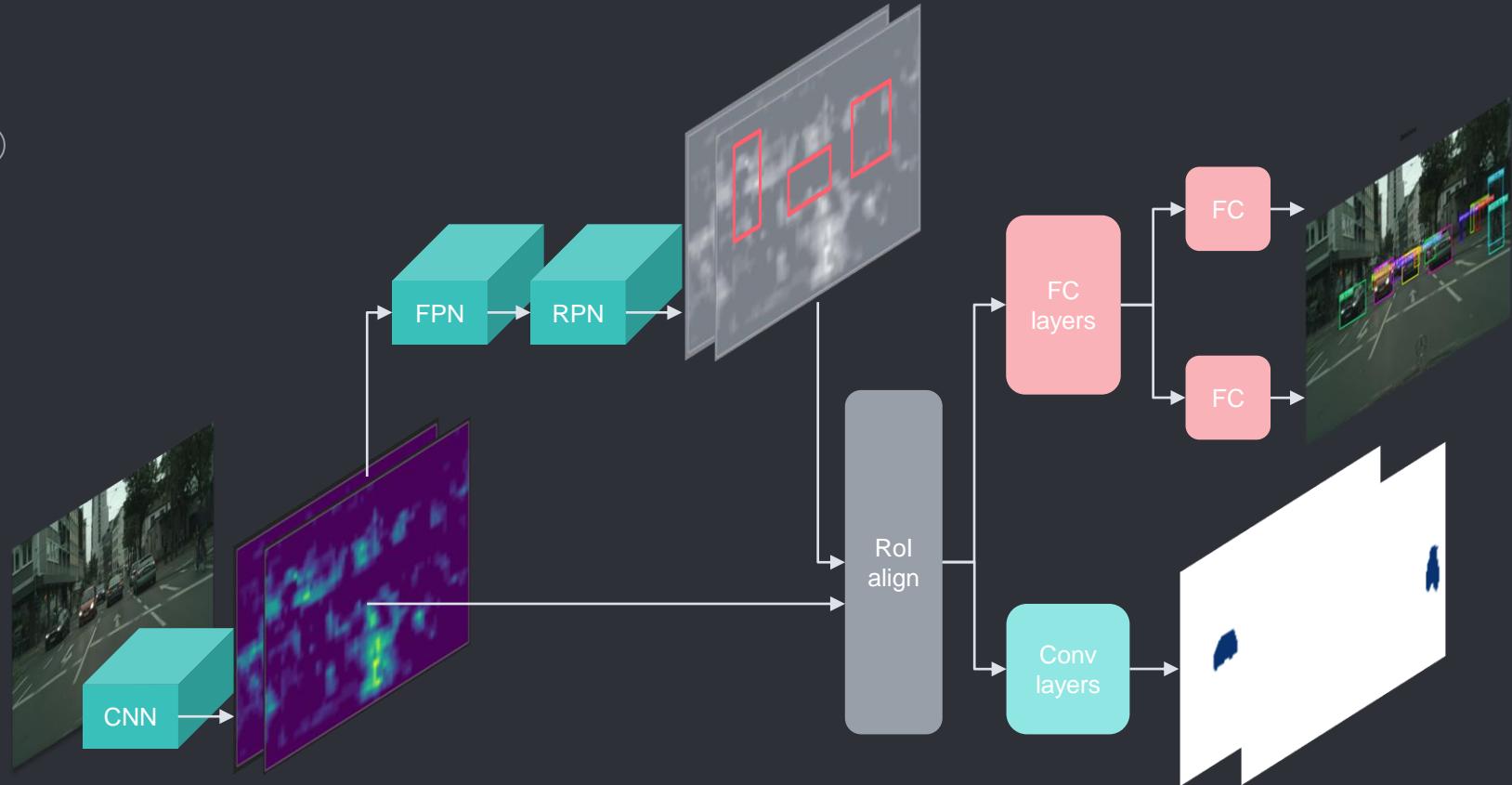
2

## Descrizione del modello

Descrizione della Mask R-CNN implementata da Matterport

- Matterport Mask R-CNN
  - Architettura Mask R-CNN proposta da **Facebook**
    - <https://arxiv.org/pdf/1703.06870.pdf>
  - Implementazione realizzata da **Matterport**
    - [https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN)

- Architettura Mask R-CNN







## Problema multi-task



- Smooth L1

$$L_{SL_1} = \frac{1}{N} \sum_{i=1}^N z_i \text{ dove } z_i = \begin{cases} \frac{0.5(x_i - y_i)^2}{\beta} & se |x_i - y_i| < \beta \\ |x_i - y_i| - 0.5\beta & altrimenti \end{cases}$$

- Categorical cross-entropy

$$L_{CE} = -\frac{1}{N} \sum_{i=1}^N \frac{1}{M} \sum_{m=1}^M \sum_{c=1}^C y_{im}^c \log(\widehat{y_{im}^c})$$

- Binary cross-entropy

$$L_{BCE} = -\frac{1}{N} \sum_{i=1}^N \sum_{m=1}^M y_{im} \log(\widehat{y_{im}}) + (1 - y_{im}) \log(1 - \widehat{y_{im}})$$

- Criticità di Matterport

- Criticità

- Implementazione basata su **TensorFlow 1**
  - Disponibile una versione della rete adattata al dataset **COCO**
    - ▣ 80 classi
    - ▣ Immagini di dimensioni variabili
  - Rappresentazione binaria delle maschere
  - Possibilità di utilizzare come ottimizzatore solo SGD

- Soluzioni

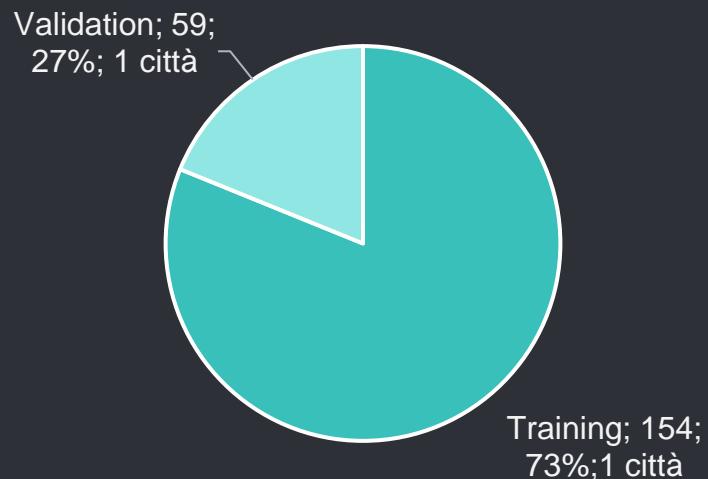
- Implementazione basata su **TensorFlow 2**
  - Adattamento della rete a **Cityscapes**
    - ▣ 10 classi
    - ▣ Immagini 2048x1024
  - Conversione dei file *json* in maschere binarie
  - Implementata la possibilità di utilizzare anche l'ottimizzatore Adam

3

## Babysitting

Verifica del corretto funzionamento del modello

- Creazione del dataset ridotto
  - Per diminuire la complessità computazionale in una prima fase si è scelto di ridurre le dimensioni del dataset



- Babysitting
- Dataset ridotto
  - Training set (154 immagini)
  - Validazione effettuata sugli stessi dati di training
- Obiettivo
  - Verificare l'andamento della loss

- Modello utilizzato

- In tabella sono riportati i principali iperparametri utilizzati per il modello

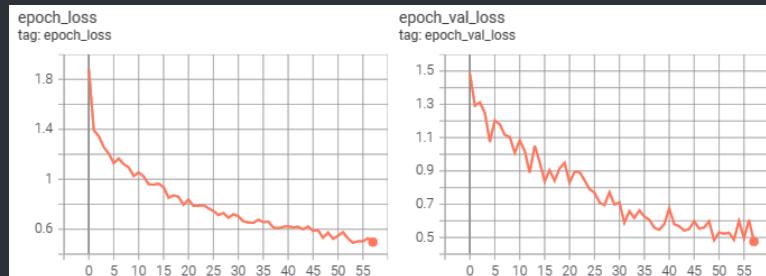
Iperparametro	Valore
Pesi	COCO
Backbone	ResNet101
Ottimizzatore	SGD
Learning rate	$10^{-3}$
Momentum	0.9
Weight decay	0
Training steps	154
Validation steps	30
Batch size	1



## Risultati



- Si addestra la rete per più di **50 epoch**
- Osservando le curve di loss si può concludere che il modello può essere addestrato correttamente



# 4

## Coarse hyperparameter tuning

Coarse tuning degli iperparametri

- Coarse hyperparameter tuning
  - Dataset ridotto
    - Training set (154 immagini)
    - Validation set (59 immagini)
  - Obiettivo
    - Valutare alcuni iperparametri di base:
      - Pesi
      - Backbone
      - Ottimizzatore

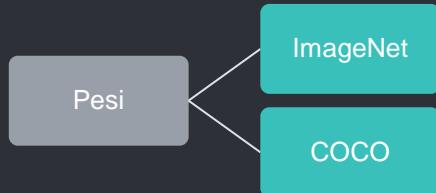
- Vincoli

- Ciascun modello è stato addestrato per **15 epochhe** con i seguenti iperparametri:

Iperparametro	Valore
Learning rate	$10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$
Momentum	0.9
Weight decay	$10^{-4}$
Training steps	154
Validation steps	30
Batch size	1

● Pesi

- Si è scelto di adottare il **transfer learning**, avendo a disposizione pesi di architetture pre-addestrate
  - ImageNet vs COCO



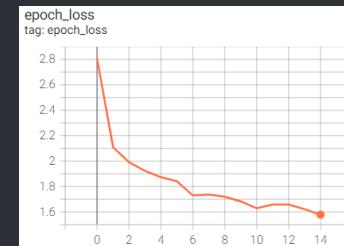


## ImageNet vs COCO

- Come prevedibile partendo dai pesi di COCO si ottengono prestazioni migliori
  - Il dataset COCO presenta 7 classi su 10 in comune con CityScapes
  - I pesi per ImageNet sono disponibili solo per ResNet50



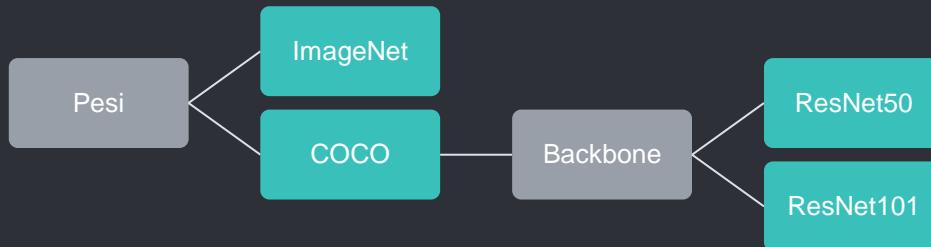
ImageNet ( $lr = 10^{-3}$ )



COCO ( $lr = 10^{-4}$ )

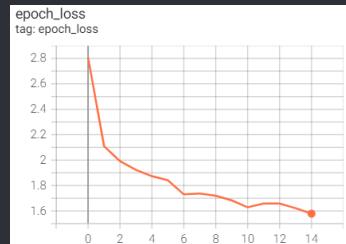
- Backbone

- L'implementazione di Matterport offre due alternative per la **backbone** della Mask R-CNN
  - ResNet50 vs ResNet101

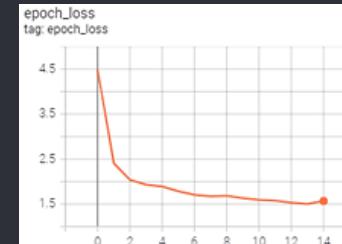


- ResNet50 vs ResNet101

- Come prevedibile ResNet101 raggiunge prestazioni migliori, avendo più livelli convoluzionali
  - ▣ Grazie alle skip connection aumentando il numero di livelli, in generale, si ha un valore di loss inferiore



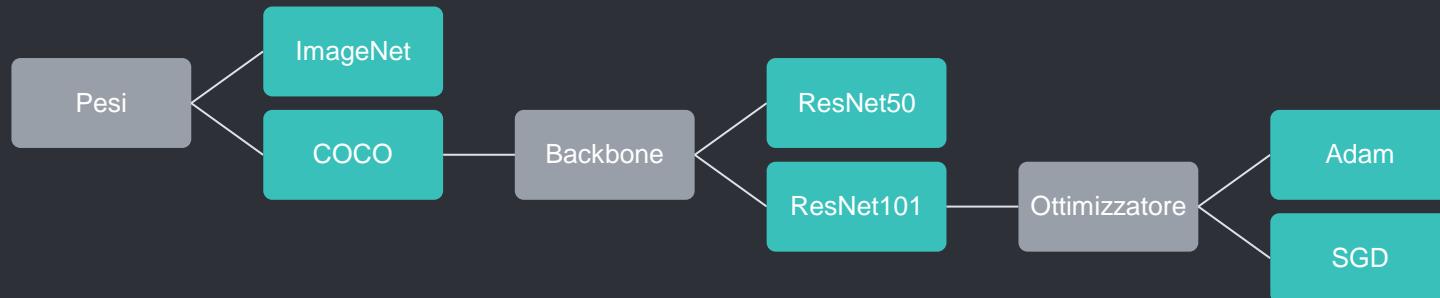
ResNet50 ( $lr = 10^{-4}$ )



ResNet101 ( $lr = 10^{-5}$ )

- Ottimizzatore

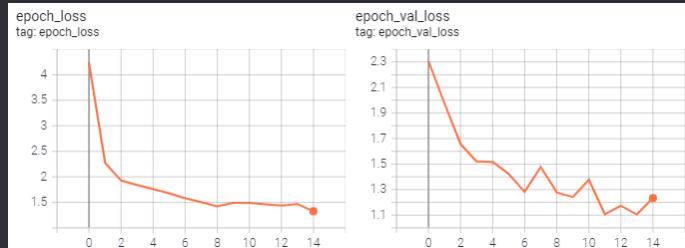
- Di default l'implementazione di Matterport prevede esclusivamente l'**ottimizzatore** SGD
- È stata estesa la libreria fornendo la possibilità di cambiare ottimizzatore
  - Adam vs SGD



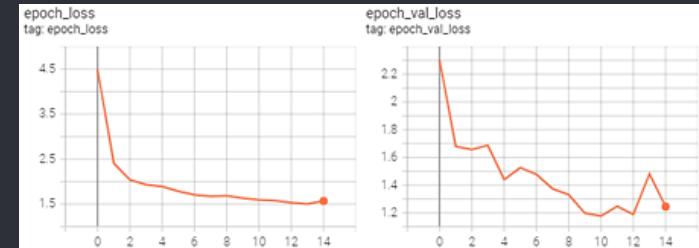


## Adam vs SGD

- Le prestazioni dei due ottimizzatori sono molto simili
  - ▣ Si procede alla fase di fine hyperparameters tuning per entrambe le configurazioni



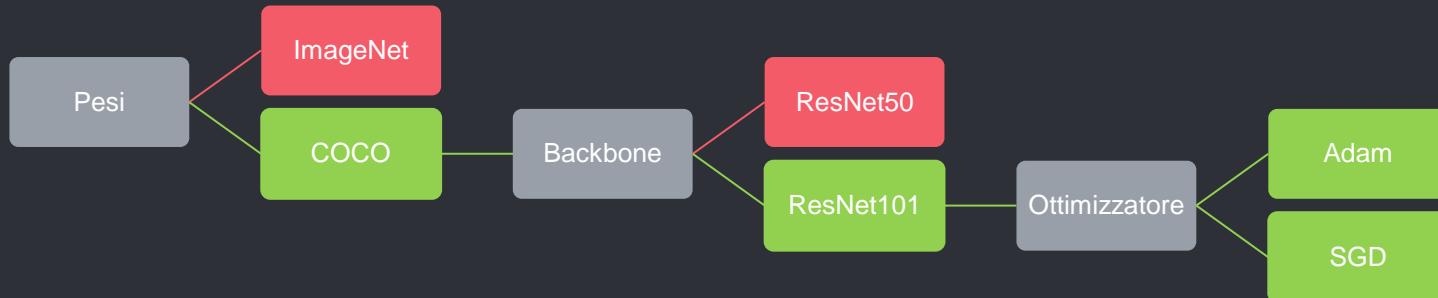
Adam ( $lr = 10^{-6}$ )



SGD ( $lr = 10^{-5}$ )

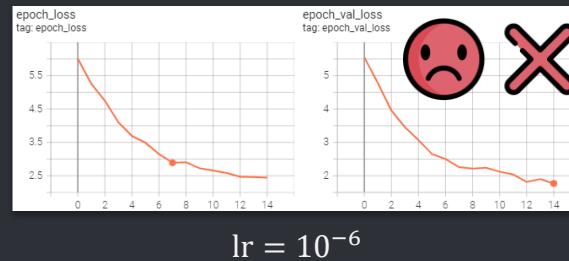
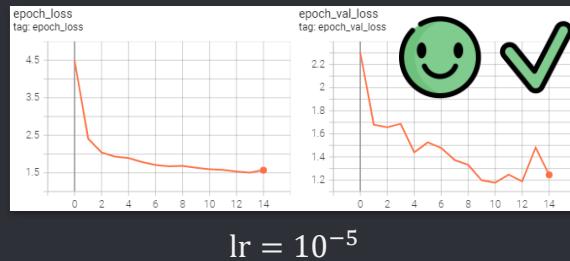
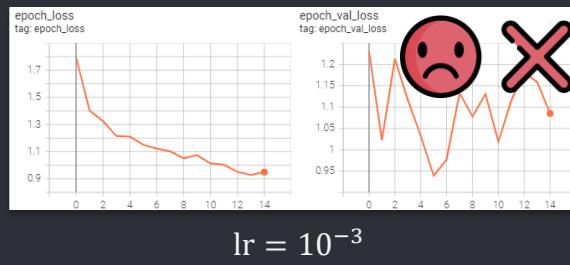
- Soluzione migliore

- Dall'analisi effettuata è emerso che la situazione migliore prevede i seguenti iperparametri:
  - Pesi: **COCO**
  - Backbone: **ResNet101**
  - Ottimizzatore: **SGD** e **Adam**



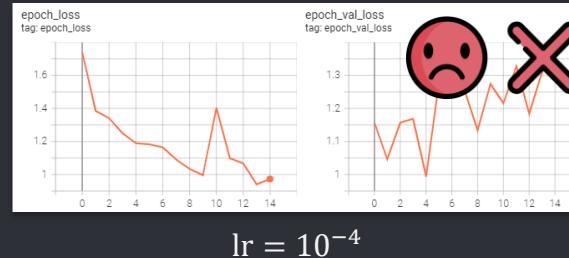
- Learning rate di SGD

- Si seleziona il migliore intervallo di learning rate per un successivo raffinamento
  - Per valori di learning rate superiori a  $10^{-3}$  la loss è NaN



- Learning rate di Adam

- Si seleziona il migliore intervallo di learning rate per un successivo raffinamento
  - Per valori di learning rate superiori a  $10^{-4}$  la loss è NaN



5

## Fine hyperparameter tuning

Fine tuning degli iperparametri

- Fine hyperparameter tuning
- Dataset ridotto
  - Training set (154 immagini)
  - Validation set (59 immagini)
- Obiettivo:
  - Ricercare i migliori iperparametri per la rete
  - Spazio di ricerca degli iperparametri:
    - Learning rate  $lr \in [10^{-5}, 10^{-4}] / [10^{-6}, 10^{-5}]$
    - Weight decay  $wd \in [10^{-4}, 10^{-2}]$
    - Momentum  $m \in [0.9, 0.99]$

- HyperBand

- Metodo di ricerca fornito da **Keras Tuner**
- Si fissa un budget di risorse (epoches)
  - Si allocano uniformemente le risorse tra tante configurazioni differenti
  - Si addestrano i modelli per poche epoches
  - Si scartano i modelli meno promettenti
  - Si allocano nuovamente le risorse tra i modelli più promettenti
- Il processo continua fino a quando i migliori modelli non sono addestrati per certo numero di epoches (20)

- Risultati di SGD

- Di seguito sono riportate le due configurazioni che hanno ottenuto lo score migliore (loss sul validation)
  - ▣ Si osserva che gli iperparametri individuati sono molto simili tra loro

Iperparametro	Valore
Learning rate	$5.93 \cdot 10^{-5}$
Momentum	0.98
Weight decay	$5.83 \cdot 10^{-3}$
Score	0.86

Iperparametro	Valore
Learning rate	$7.1 \cdot 10^{-5}$
Momentum	0.93
Weight decay	$5.83 \cdot 10^{-3}$
Score	0.91

- Risultati di Adam

- Di seguito sono riportate le due configurazioni che hanno ottenuto lo score migliore (loss sul validation)
  - ▣ Si osserva che lo score in termini di loss è superiore rispetto a quello calcolato in SGD

Iperparametro	Valore
Learning rate	$7.3 \cdot 10^{-6}$
Momentum	0.96
Weight decay	$5.67 \cdot 10^{-4}$
Score	0.93

Iperparametro	Valore
Learning rate	$5.73 \cdot 10^{-6}$
Momentum	0.95
Weight decay	$4.13 \cdot 10^{-3}$
Score	0.94

6

## Training

Addestramento del modello più promettente



## Training



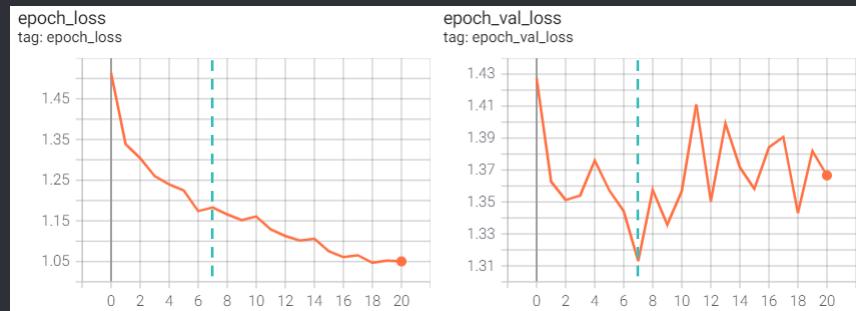
- Dataset completo
  - Training set (1403 immagini)
  - Validation set (326 immagini)
- Obiettivi
  - Addestrare il modello più promettente fissando i migliori iperparametri individuati nella fase precedente
  - Valutare le prestazioni dell'apprendimento con l'aggiunta di data augmentation ed ulteriori varianti



## Modello migliore



- Di seguito sono riportate le curve di apprendimento relative al modello addestrato con i migliori iperparametri individuati
- Seguendo la regola dell'**early-stopping**, si considera il modello addestrato fino all'epoca 8



COCO

$$lr = 6 \cdot 10^{-5}$$

ResNet101

$$m = 0.98$$

SGD

$$wd = 6 \cdot 10^{-3}$$



## Data augmentation



- Si definiscono le seguenti operazioni di data augmentation



Immagine originale



Flip

Flipping orizzontale  
con probabilità 0.5

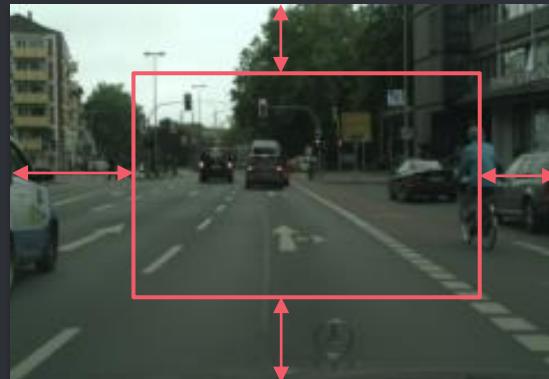


Crop

Cropping random di  
una patch con  
probabilità 0.5

- Criticità del crop

- I crop sono ottenuti scartando randomicamente una percentuale di pixel da 0 a 20 in ogni direzione
  - Il centro è sempre presente nella patch estratta
- A valle del crop è realizzato un reshaping alla dimensione originale dell'immagine
  - Si perde l'aspect ratio della patch





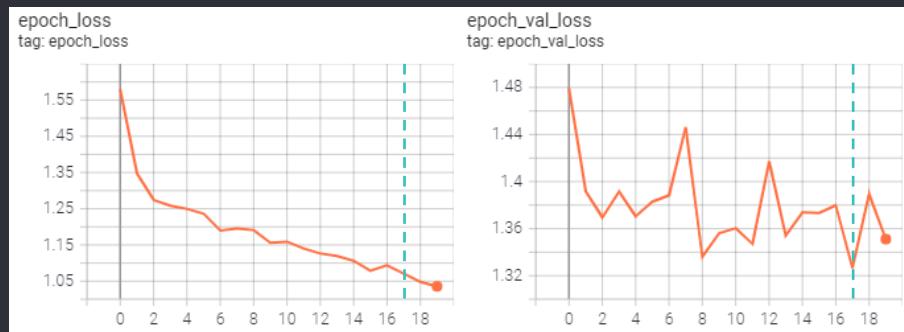
## Modello migliore + flip



- Di seguito sono riportati i risultati ottenuti applicando la sola trasformazione di **flip**



- Modello migliore + flip e crop
  - Di seguito sono riportati i risultati ottenuti applicando entrambe le trasformazioni di **flip** e **crop**

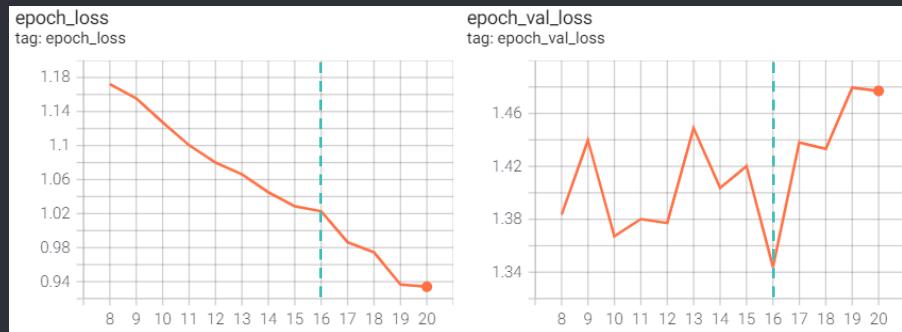




## Sbloccare la backbone



- Si addestrano anche i pesi degli ultimi livelli convoluzionali della backbone
  - ▣ L'addestramento continua dall'epoca 8 del modello migliore

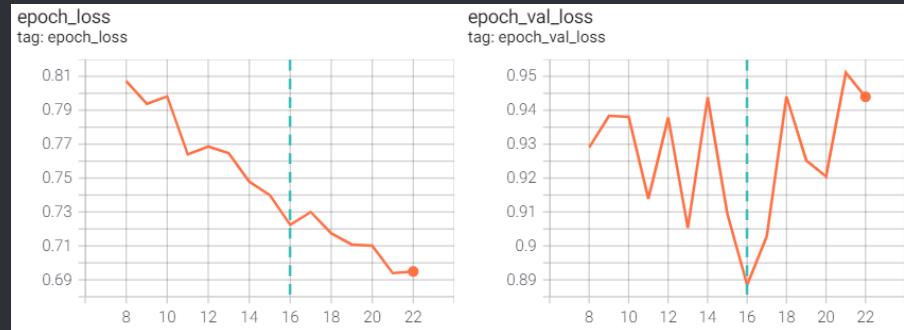




## Bloccare RPN e FPN



- Si bloccano i pesi di RPN ed FPN e si azzera il contributo delle relative loss
  - L'addestramento continua dall'epoca 8 del modello migliore
  - $L = L_{R-CNN_{box}} + L_{R-CNN_{class}} + L_{R-CNN_{mask}}$



# 7

## Model Evaluation

Metriche utilizzate ed evaluation del modello



## Evaluation

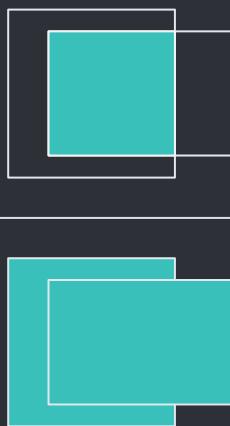


- Dataset completo
  - Test set
- Obiettivi
  - Valutare le prestazioni dei modelli addestrati in termini di metriche
  - Valutare visivamente i risultati dei modelli

- Metriche di valutazione
  - Le prestazioni della Mask R-CNN si misurano in termini di **Average Precision (AP)**, definita a partire da altre metriche:
    - Intersection over Union (IoU)
    - Precision e Recall
    - Precision – Recall Curve

- Intersection over Union (IoU)

- Rapporto tra intersezione e unione della *segmentation mask* predetta e della *segmentation mask* della ground truth

$$IoU = \frac{A \cap B}{A \cup B} = \frac{\text{Area of intersection}}{\text{Area of union}}$$




## Precision e Recall



### Precision

Istanze rilevanti restituite  
rispetto al totale delle istanze  
restituite

$$P = \frac{TP}{TP + FP}$$

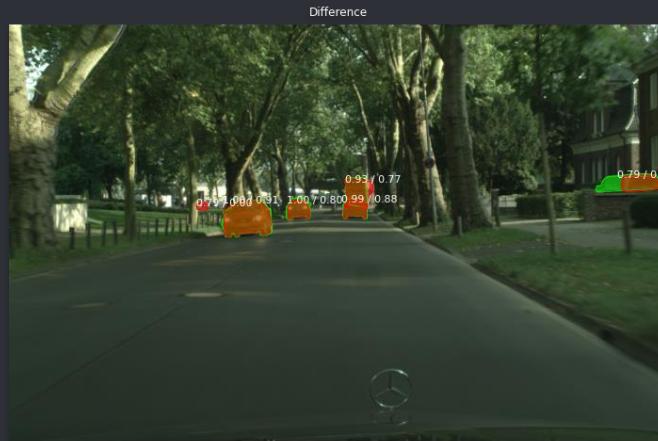
### Recall

Istanze rilevanti restituite  
rispetto al totale delle istanze  
rilevanti

$$R = \frac{TP}{TP + FN}$$

- Precision – Recall Curve (Esempio)

- Si evidenzia la differenza tra le istanze predette e le istanze della ground truth
  - Per ogni coppia di istanze (predizione – ground truth) si calcola la IoU





## Precision – Recall Curve (Esempio)

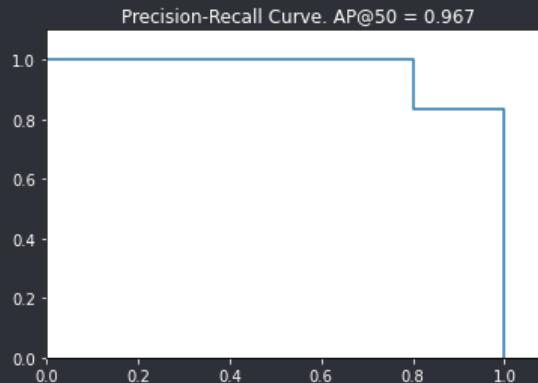
- Si verificano i match tra istanze predette e istanze della ground truth
- Si ha un match se:
  - Le classi delle istanze coincidono
  - La IoU delle maschere è superiore ad una **soglia**
- Si contano i match effettivi
  - $pred\_match = [4, 3, 2, 1, -1, 0]$
  - $sum\_match = [1, 2, 3, 4, 4, 5]$



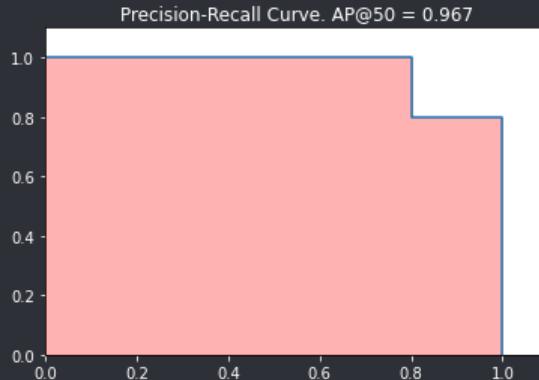


## Precision – Recall Curve (Esempio)

- Si calcolano precision e recall al variare del numero di istanze restituite a partire dal vettore  $sum\_match = [1, 2, 3, 4, 4, 5]$ 
  - $precisions = [1/1, 2/2, 3/3, 4/4, 4/5, 5/6] = [1, 1, 1, 1, 0.8, 0.83]$
  - $recalls = [1/5, 2/5, 3/5, 4/5, 4/5, 5/5] = [0.2, 0.4, 0.6, 0.8, 0.8, 1]$
- Si costruisce la curva con i valori di precision e recall ottenuti



- Average Precision (AP)
  - Per ogni immagine si calcola l'AP come area della Precision – Recall Curve
    - Si fissa come soglia:  $IoU = 0.5$



- Si calcola l'AP complessiva come media delle AP di tutte le immagini di test

- AP dei modelli addestrati



	Modello migliore	Flip	Flip e crop	Backbone sbloccata	RPN e FPN bloccate
$AP^{0.5}$	0.587	0.611	0.615	0.608	0.623

- Class Average Precision (cAP)
  - Per ogni immagine si calcola l'AP per classe
    - Si calcola l'AP considerando solo le istanze di una singola classe del dataset
    - Le AP sono calcolate fissando:  $IoU = 0.5$
  - Per ogni classe si calcola la cAP complessiva come media delle AP di tutte le immagini di test

- cAP dei modelli addestrati

	Modello migliore	Flip	Flip e crop	Backbone sbloccata	RPN e FPN bloccate
Car	0.713	0.722	0.736	0.719	0.745
Person	0.464	0.486	0.487	0.473	0.493
Rider	0.530	0.592	0.563	0.561	0.574
Motorcycle	0.190	0.223	0.194	0.208	0.181
Bicycle	0.402	0.443	0.436	0.432	0.444
Bus	0.527	0.575	0.542	0.521	0.604
Trailer	0	0	0	0	0
Truck	0.161	0.197	0.212	0.113	0.242
Caravan	0	0	0	0	0
Train	0	0	0	0	0

- Range Average Precision (rAP)
  - Si calcola l'AP complessiva su tutte le immagini di test per diversi valori di soglia di IoU
    - $\text{IoU} \in [0.5, 0.95]$  con passo **0.05**
  - Si calcola la **rAP** come media delle AP calcolate considerando le diverse soglie

- rAP dei modelli addestrati

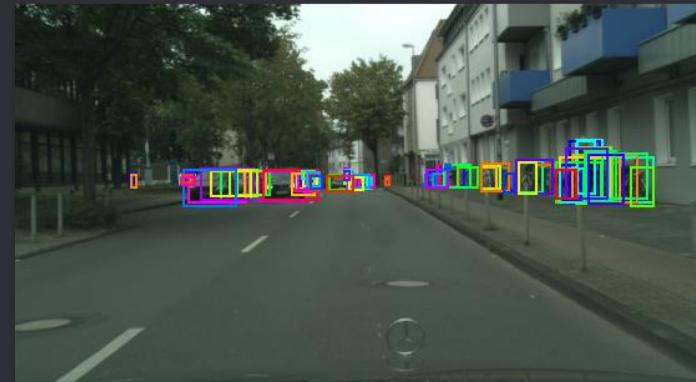
	Modello migliore	Flip	Flip e crop	Backbone sbloccata	RPN e FPN bloccate
<b>AP<sup>0.5</sup></b>	0.587	0.611	0.615	0.608	0.623
<b>AP<sup>0.55</sup></b>	0.566	0.586	0.590	0.581	0.602
<b>AP<sup>0.6</sup></b>	0.534	0.549	0.546	0.544	0.567
<b>AP<sup>0.65</sup></b>	0.487	0.500	0.488	0.496	0.508
<b>AP<sup>0.7</sup></b>	0.430	0.437	0.416	0.434	0.444
<b>AP<sup>0.75</sup></b>	0.352	0.360	0.338	0.360	0.362
<b>AP<sup>0.8</sup></b>	0.274	0.270	0.250	0.280	0.275
<b>AP<sup>0.85</sup></b>	0.187	0.185	0.171	0.190	0.188
<b>AP<sup>0.9</sup></b>	0.099	0.095	0.085	0.106	0.100
<b>AP<sup>0.95</sup></b>	0.001	0.000	0.000	0.003	0.001
<b>rAP</b>	0.352	0.360	0.350	0.360	0.367



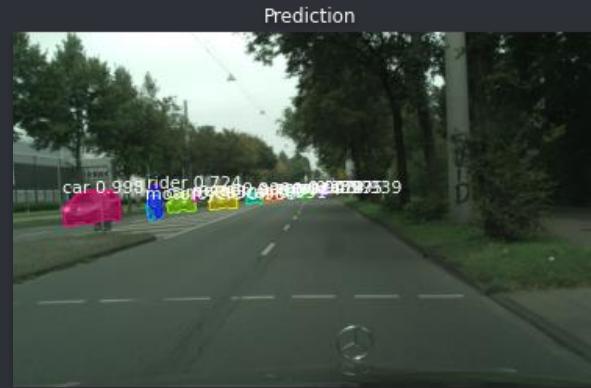
# Inferenza

Si visualizzano le predizioni della rete  
per alcune immagini di test

- Region of Interest (RoI)
  - La RPN restituisce delle Region of Interest (RoI), regioni in cui si trovano potenziali istanze

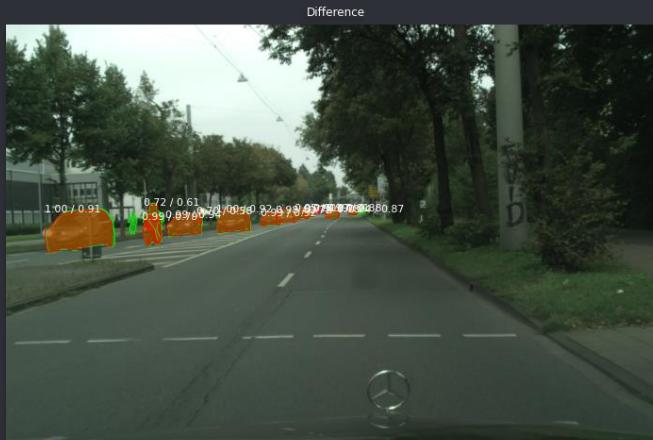


- Instance Segmentation



- Instance Segmentation

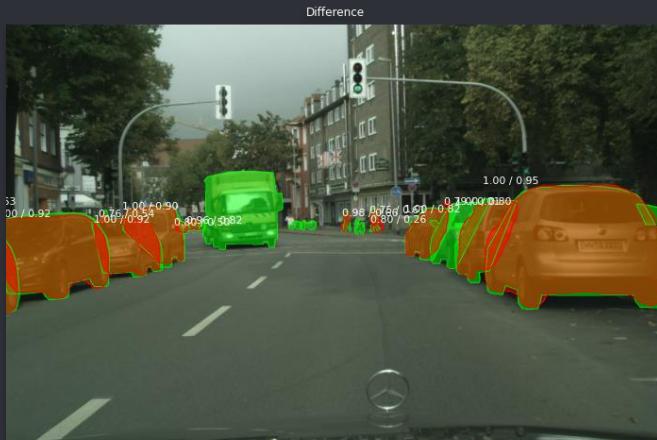
- Si visualizzano in un'unica immagine le differenze tra maschere effettive e maschere predette



- Object Detection



- Confronti tra i modelli



Modello migliore  
( $AP^{0.5} = 0.587$ )



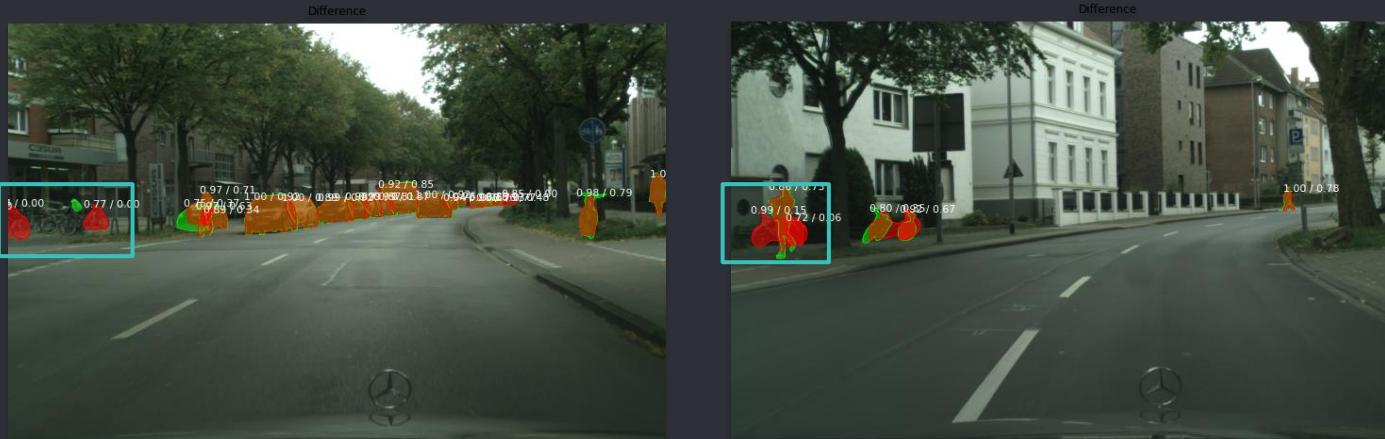
Modello migliore (E8) +  
RPN e FPN bloccate  
( $AP^{0.5} = 0.623$ )



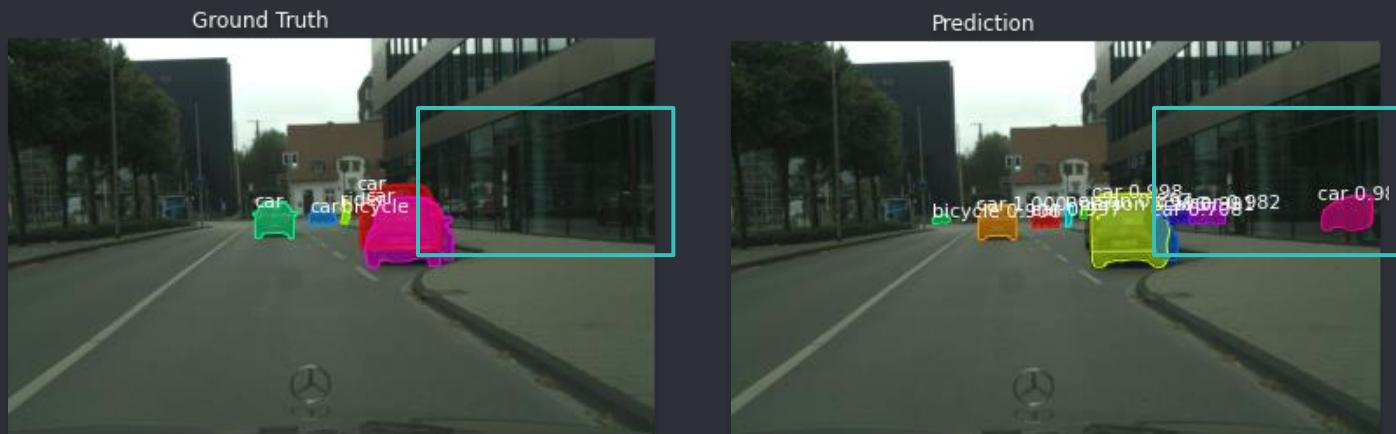
# Considerazioni

Si ripotano alcune considerazioni  
sulle predizioni realizzate dalla rete

- Imprecisioni della ground truth
  - In alcuni casi la ground truth presenta istanze per le quali non sono presenti le maschere



- Limiti del modello
  - Le immagini riflesse nei vetri confondono la rete



- Limiti del modello
  - Istanze sovrapposte possono confondere la rete
    - In alcuni paper recenti si cerca di ovviare a questo problema



- Classi simili
  - Alcune classi simili confondono la rete
    - Un **rider** viene classificato anche come **persona**



- Classi poco frequenti
  - Alcune classi poco frequenti sono difficili da classificare
    - Un **truck** viene classificato come **car**
    - Un **bus** viene classificato correttamente



Grazie per l'attenzione!

Domande?