

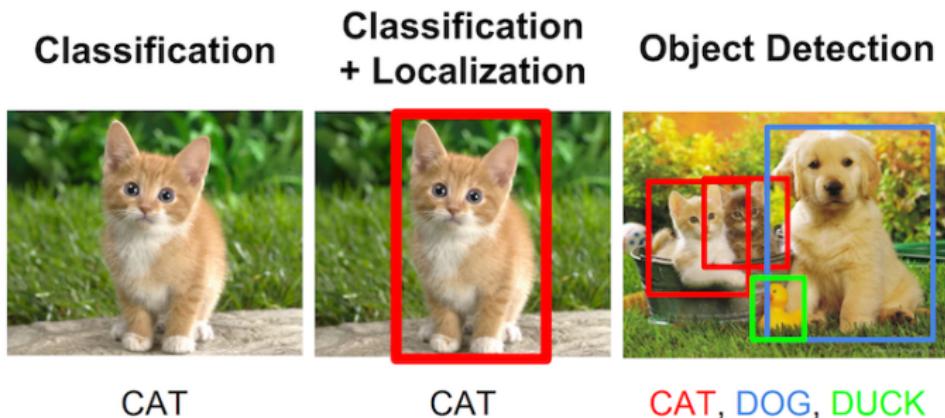
# Traffic Signal Detection for Mobile Devices

Milo Marchetti e Nicolò Bartelucci

Alma Mater Studiorum - Università di Bologna

# Object Detection

Object Detection e Image Classification sono tecnologie che consentono di analizzare un'immagine sfruttando processi di Computer Vision e Image Processing.



Nell'ambito stradale la Object Detection risulta essere la scelta migliore in quanto si ha la necessità di valutare contemporaneamente più oggetti a schermo.

# Strumenti Utilizzati



TensorFlow



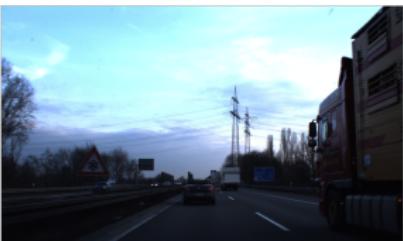
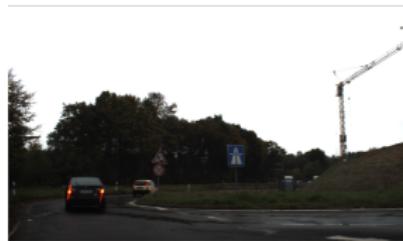
TensorFlow  
Lite



android studio

# The German Traffic Sign Detection Dataset

The German Traffic Sign Detection Dataset è una raccolta di immagini destinata ai campi di ricerca della Computer Vision al fine di valutare il rilevamento di segnali stradali in singole immagini. Il Dataset è composto da 900 immagini di 43 classi di segnali diversi, divise in 600 per il training e 300 per l'evaluation.



# Modelli Addestrati

## SSD Inception v2

Modello basato sull'unione di SSD e Inception v2 che è caratterizzato da una velocità di 42 ms e una precisione di 24 mAP

## SSD Mobilenet v2

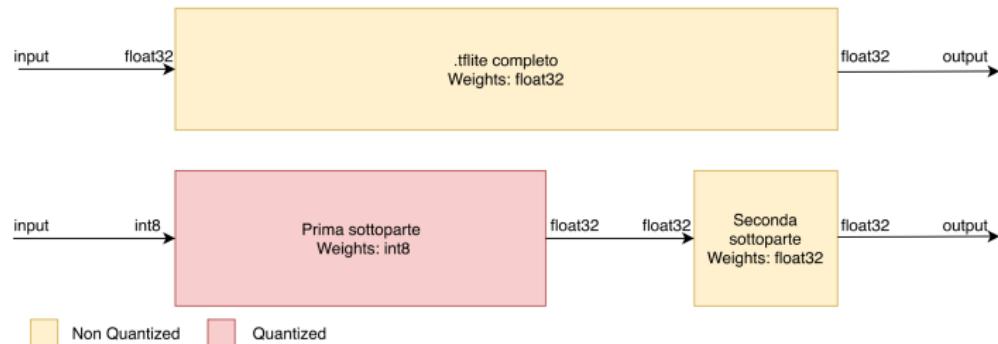
Modello basato sull'unione di SSD e Mobilenet v2 che è caratterizzato da una velocità di 31 ms e una precisione di 22 mAP

## SSD Mobilenet v3 Small

Modello basato sull'unione di SSD e Mobilenet v3 che è caratterizzato da una velocità di 43 ms (su un dispositivo Pixel 1) e una precisione di 15.6 mAP

# Quantizzazione

La quantizzazione di una rete neurale post-training è una tecnica utilizzata per ridurre il peso del modello “.tflite”. Un problema della quantizzazione di modelli tflite è il mancato supporto alle custom operations: delle operazioni non supportate ufficialmente da TensorFlow Lite.



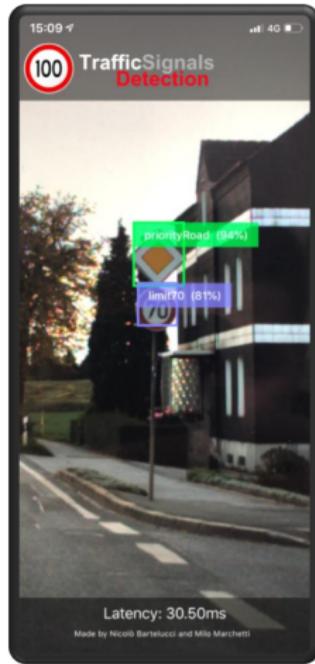
Per sopperire questa mancanza di supporto si è deciso di dividere il grafo completo in due sottografi al fine di quantizzare la maggior parte della rete.

# Applicazione Android/iOS

Il principio di funzionamento è molto semplice: sono effettuate continue chiamate alle api TensorFlow Lite fornendo in input un'immagine acquisita dalla fotocamera del dispositivo in esecuzione.

Gli output ottenuti sono:

- bounding-box
- classe di previsione
- precisione percentuale
- tempo di latenza



# Risultati: Modelli non Quantizzati

I test sono stati effettuati su iPhone X, Xiaomi Mi A1 e Huawei P9 Lite. I risultati relativi alle performance di esecuzione dei modelli per dispositivo sono riportati nella tabella.

Non-Quantized Model			
	SSD Inception v2	SSD Mobilenet v2	SSD Mobilenet v3
<b>iPhone X</b>	135 ~ 138 ms ~ 7.5 fps	~ 27 ms ~ 37 fps	~ 17 ms ~ 58 fps
<b>Xiaomi Mi A1</b>	650 ~ 720 ms ~ 1.4 fps	160 ~ 210 ms ~ 5 fps	80 ~ 110 ms ~ 10 fps
<b>Huawei P9 Lite</b>	730 ~ 790 ms ~ 1.3 fps	260 ~ 290 ms ~ 3.5 fps	100 ~ 120 ms ~ 9 fps

# Risultati: Modelli Quantizzati

Quantized Model			
	SSD Inception v2	SSD Mobilenet v2	SSD Mobilenet v3
Xiaomi Mi A1	620 ~ 680 ms ~ 1.5 fps	200 ~ 220 ms ~ 4.5 fps	100 ~ 140 ms ~ 8 fps
Huawei P9 Lite	640 ~ 690 ms ~ 1.5 fps	190 ~ 220 ms ~ 5 fps	140 ~ 150 ms ~ 7 fps

L'approccio proposto di quantizzazione, mirato a sorvolare il problema delle custom operation, non sempre ha portato a una riduzione dei tempi di latenza.

## Concludendo

- ① La rete ottenuta con il modello SSD Mobilenet v2 è quella che ha portato i migliori compromessi di velocità di esecuzione e precisione media, perciò la demo si baserà su tale modello.
- ② In dispositivi più datati la riduzione della dimensione del modello ha portato a un miglioramento delle performance, mentre in dispositivi più performanti invece, il guadagno dovuto alla quantizzazione non è stato sufficiente a superare la perdita dovuta all'overhead.

## Sviluppi Futuri Proposti

- ① Ricerca di un dataset più ricco ed esteso.
- ② Ri-quantizzazione dopo il supporto alle custom operation.

GitHub: <https://github.com/nicobargit/TrafficSignalDetection.git>