Neural Compression and Filtering for Edge-assisted Real-time Object Detection in Challenged Networks

Yoshitomo Matsubara **Marco Levorato**

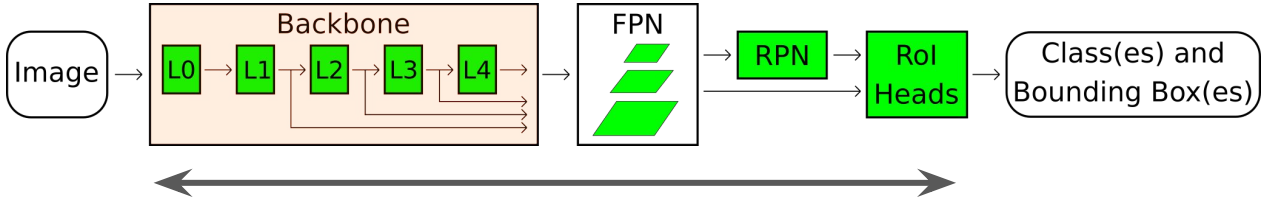
University of California, Irvine

January 12-15, 2021 @ Milan, Italy (Virtual)
25th International Conference on Pattern Recognition

Problem Setting

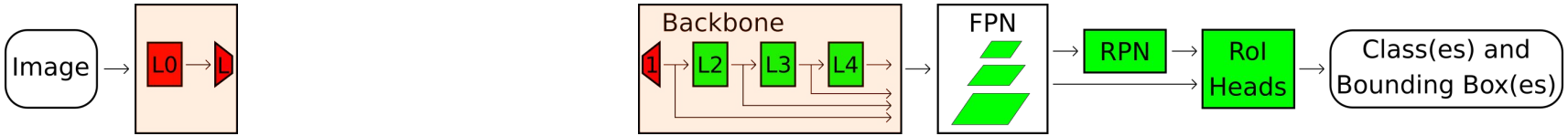
R-CNN model

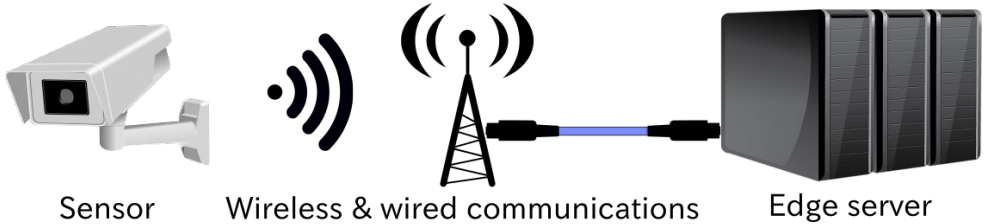
Powerful & Complex



Minimize [total inference time](#) T
providing as high accuracy as possible

Split Computing





Resource-constrained edge computing system

$$T = \tau_{\text{head}} + \tau_{\text{comm}} + \tau_{\text{tail}}$$

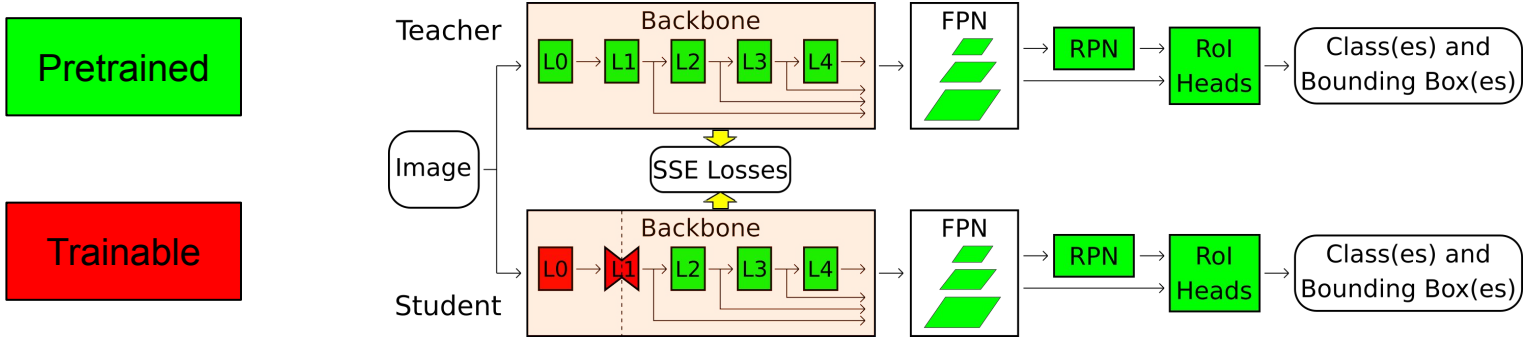
Bottleneck Injection & Generalized Head Network Distillation

- Introduce "*Bottleneck*" to pretrained R-CNN object detectors

  ➡ Reduce size of data to be transferred to edge server

- Generalize head network distillation (*Matsubara et al. 2019*)

  ➡ - Learn compressed bottleneck representations while preserving accuracy
  - Train head portion only, thus save training time



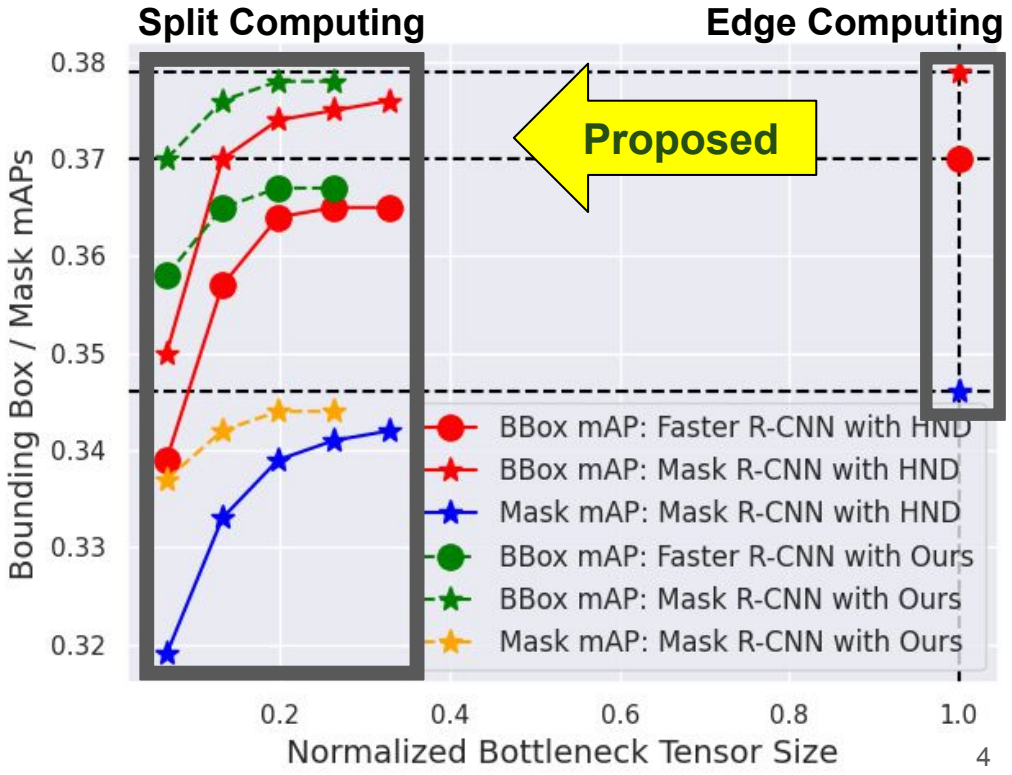
Small Bottleneck Representations vs. Detection Performance

Dataset: COCO 2017

Metrics: mAP (IoU: 0.5-0.95)
(mean Average Precision)

Compared to input tensor/JPEG file,
- 93% tensor size reduction
- 36% file size reduction

at a cost of ~1pt mAP loss



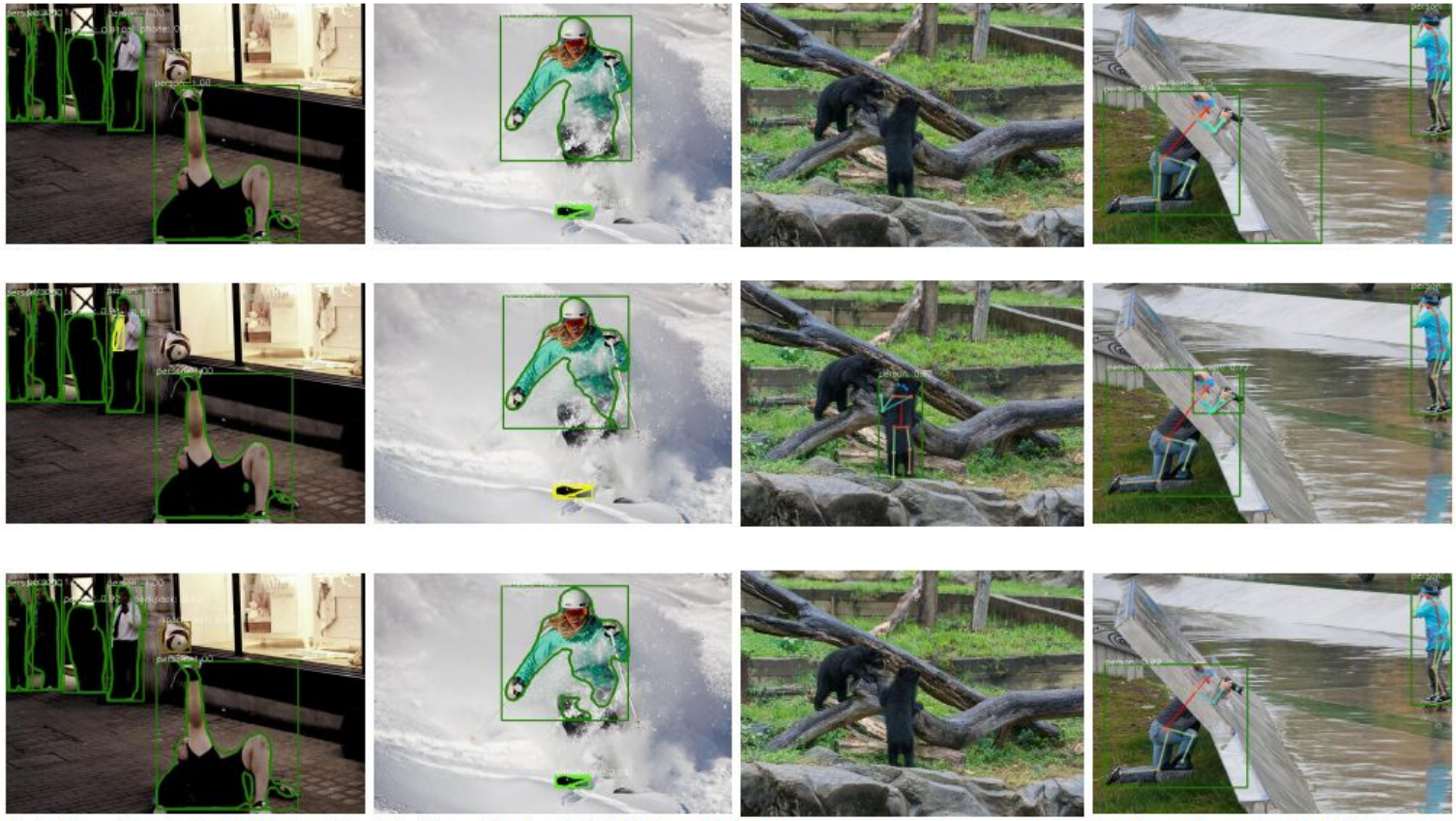
Qualitative Analysis

Comparable

Pretrained
Original models

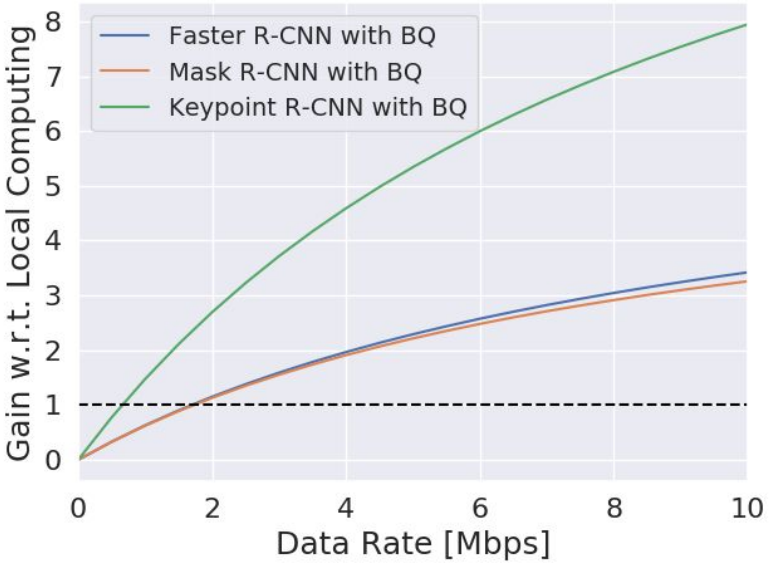
HND
Models w/ Bottleneck

GHND
Models w/ Bottleneck



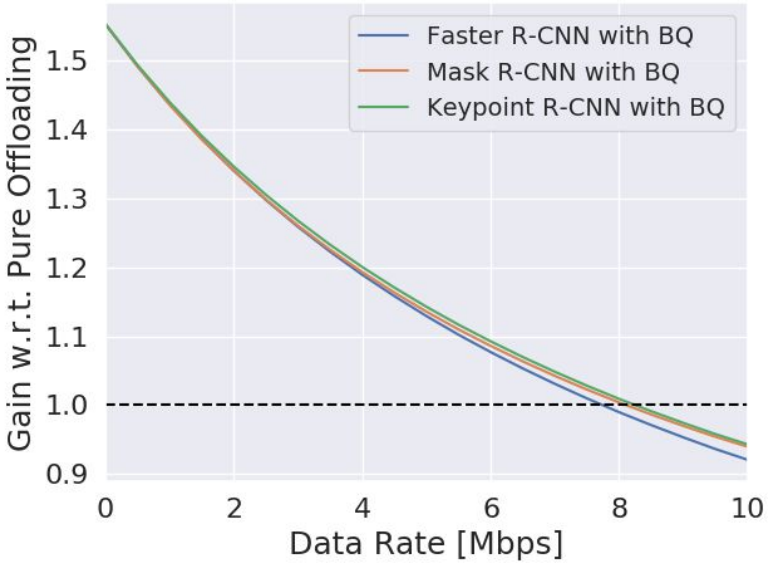
Inference Time Evaluation (Local: Jetson TX2, Edge: Desktop w/ one GPU)

vs. Local Computing



Gain = T_{SC} / T_{LC}

vs. Pure Offloading (w/ JPEG compression)



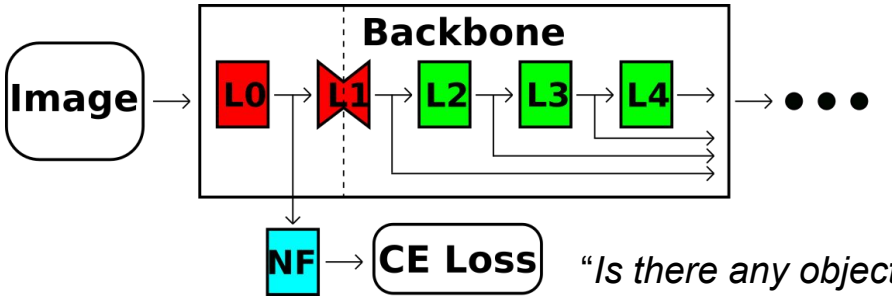
Gain = T_{SC} / T_{PO}

Proposed approach is beneficial at where Gain > 1

Filtering out Images with No Objects of Interest

- Introduce a lightweight "*Neural Filter*" (NF) to the head-distilled R-CNN model
- Filter out "*empty*" images

➡ The downstream pipeline can be skipped (i.e., no offloading) if filtered out



"*Is there any objects of interest?*"

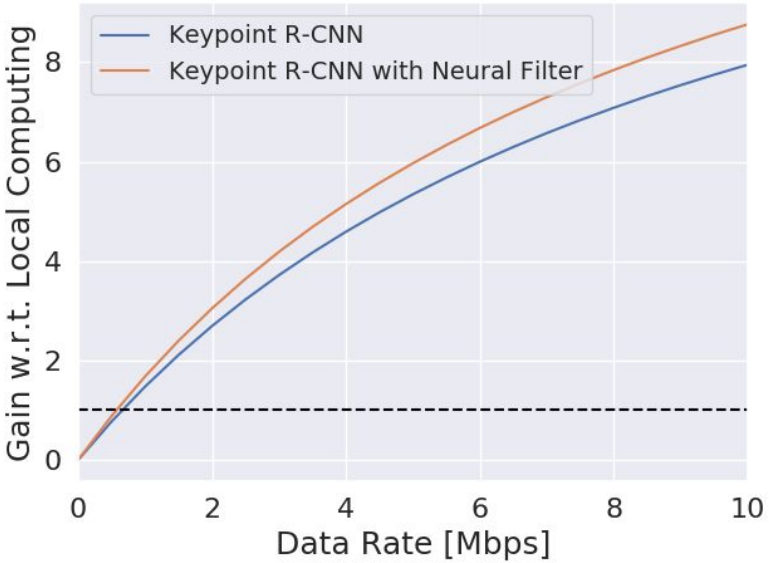
No persons vs. Two persons

- Train a NF as a binary classifier on COCO 2017, freezing all the other modules

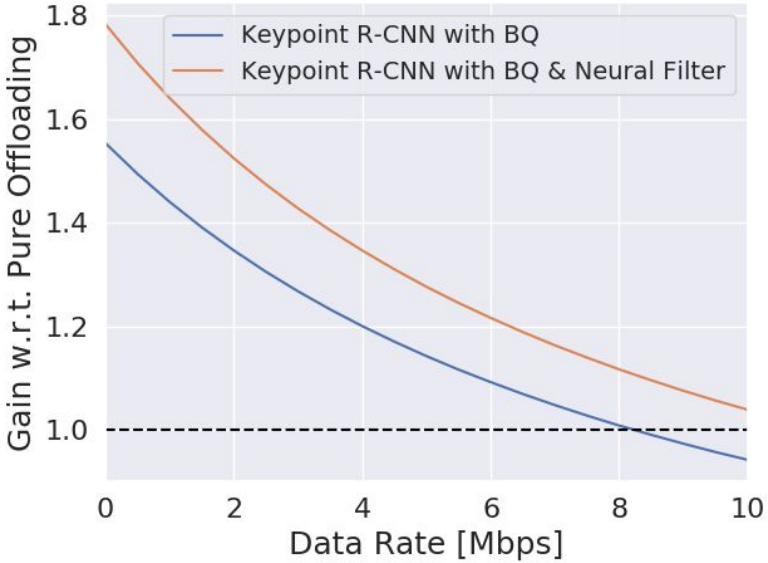
➡ ROC-AUC: 0.919 for COCO 2017 validation split

Inference Time Evaluation with Neural Filter

vs. Local Computing



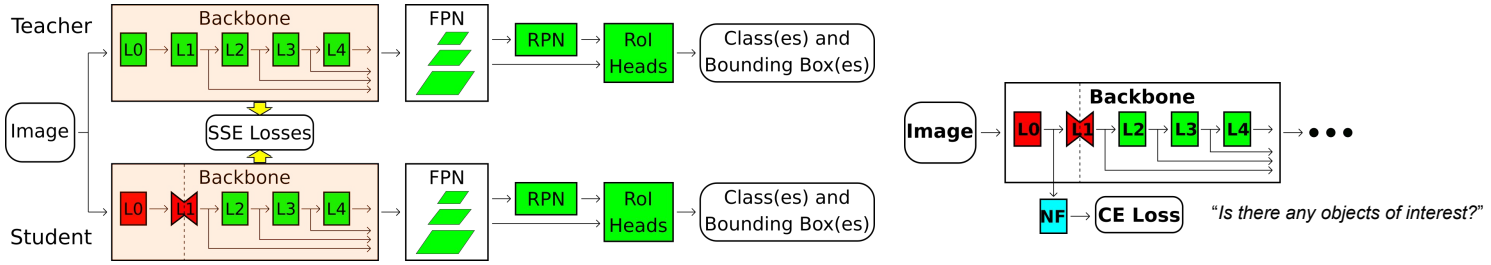
vs. Pure Offloading (w/ JPEG compression)



NF widened range of data rate that split computing is more beneficial

Summary

In this work,



- Refined our head network distillation technique for object detection tasks
- Introduced a neural filter to filter out "*empty*" images for efficient inference

Three benchmark models: Faster, Mask and Keypoint R-CNNs on COCO 2017

We are the first to

- Successfully introduce small bottlenecks to the models w/ ~1pt mAP loss
- Discuss split computing w/ bottleneck-injected object detectors

  that offers improved latency in challenged network configurations

See you in poster session T3.2!

Yoshitomo Matsubara **Marco Levorato**

yoshitom@uci.edu @yoshitomo_cs

Code & Trained models are available at
https://github.com/yoshitomo-matsubara/