YOLO-ReT: Towards High Accuracy Real-time Object Detection on Edge GPUs

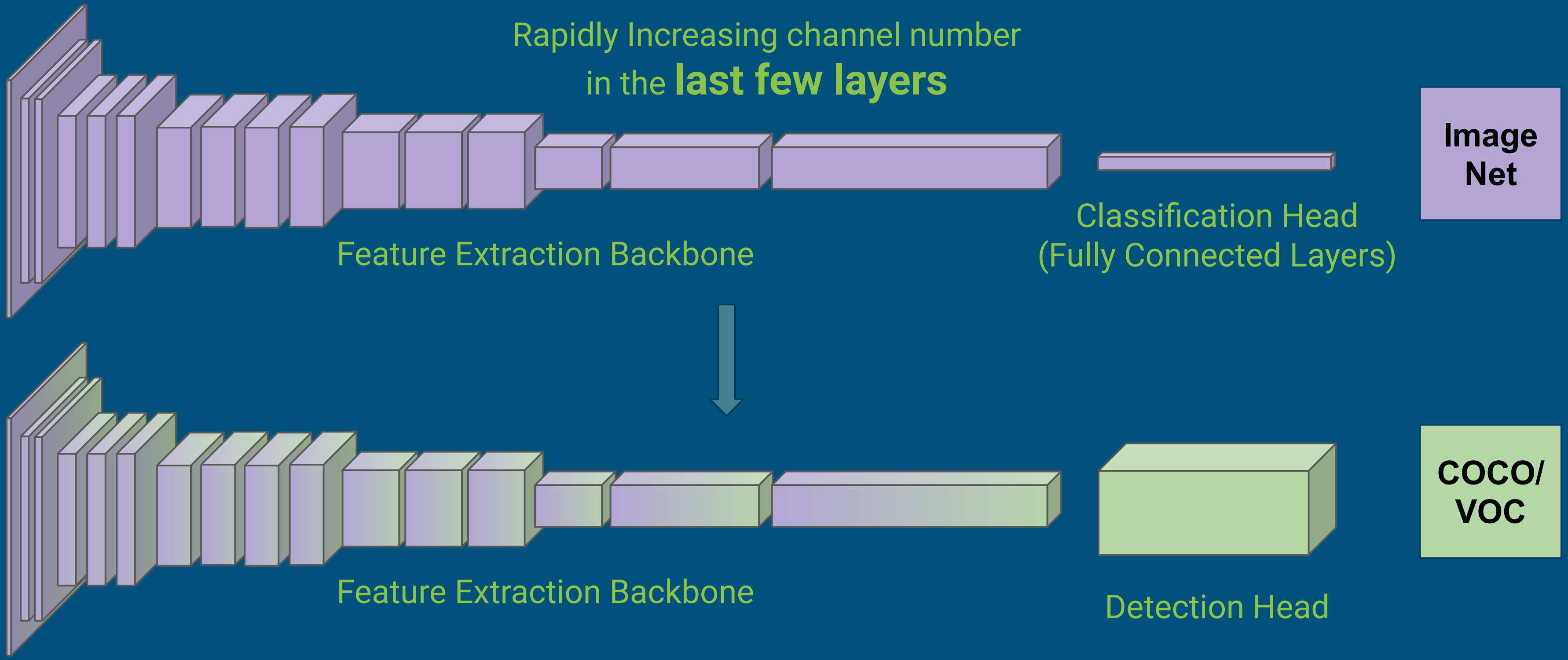
Prakhar Ganesh[1*], Yao Chen[1*], Yin Yang[2], Deming Chen[1,3], Marianne Winslett[1,3]

[1] Advanced Digital Sciences Center, Illinois at Singapore
[2] College of Science and Engineering, Hamad Bin Khalifa University, Qatar
[3] University of Illinois at Urbana-Champaign, USA

Background & Motivation

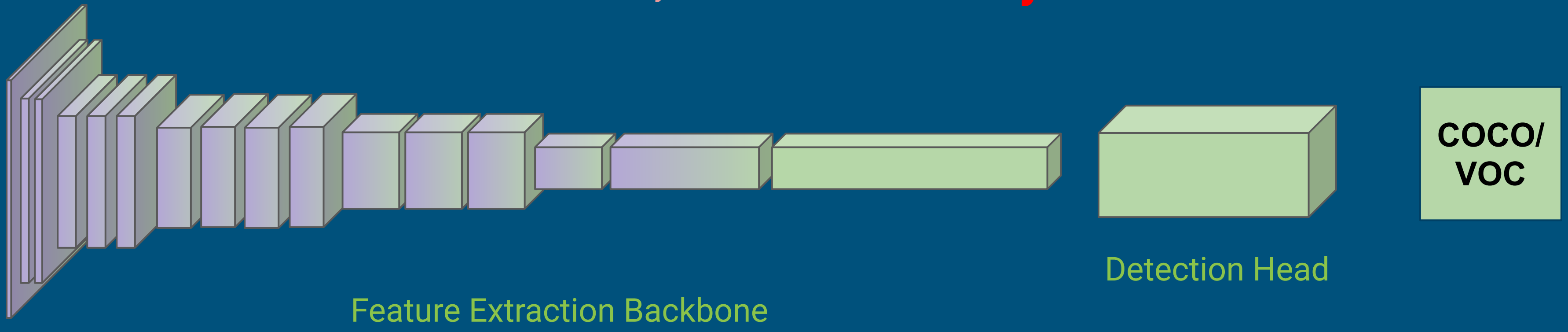
- **Transfer Learning in Object Detection**
- **Multi-Scale Feature Interaction**

Transfer Learning in Object Detection



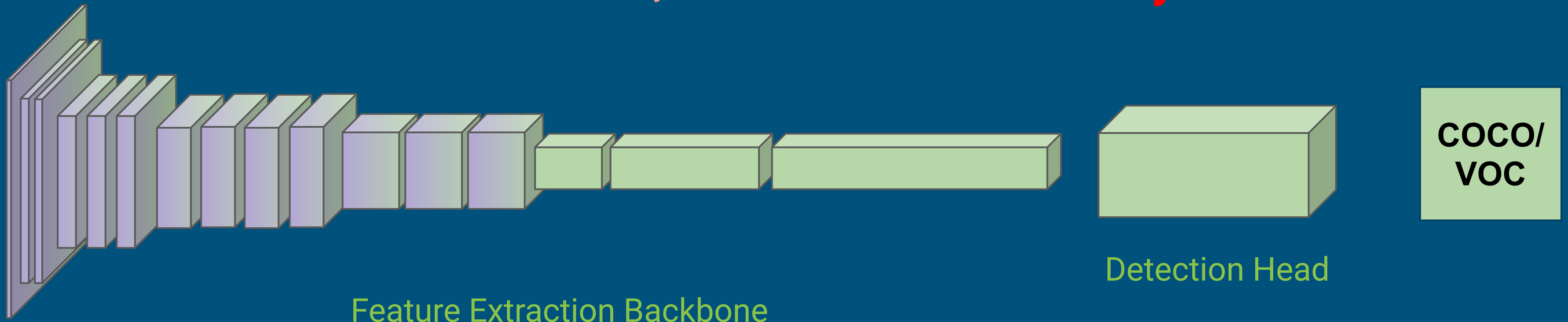
Transfer Learning in Object Detection

What if we had randomly initialised the **last layer?**



Transfer Learning in Object Detection

What if we had randomly initialised the **last three layers?**

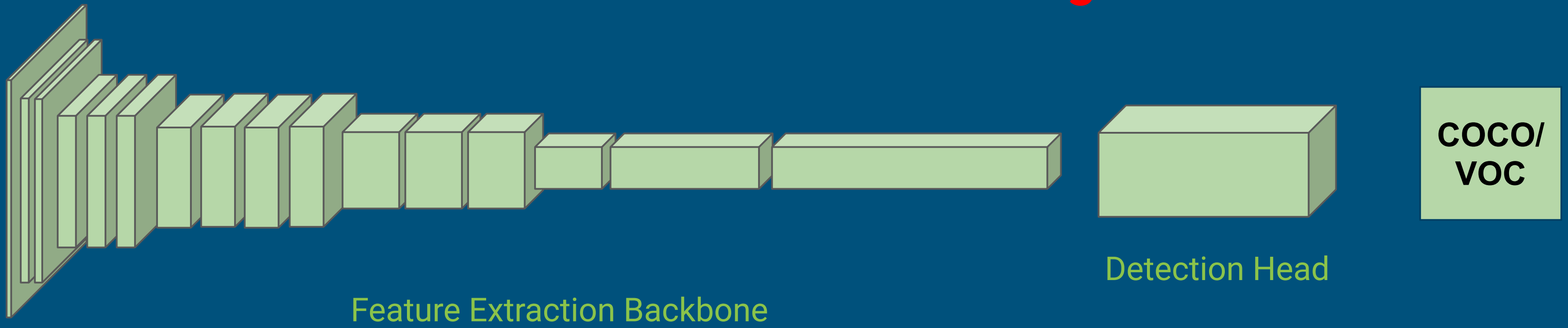


Detection Head

Feature Extraction Backbone

Transfer Learning in Object Detection

What if we had used **no transfer learning at all?**

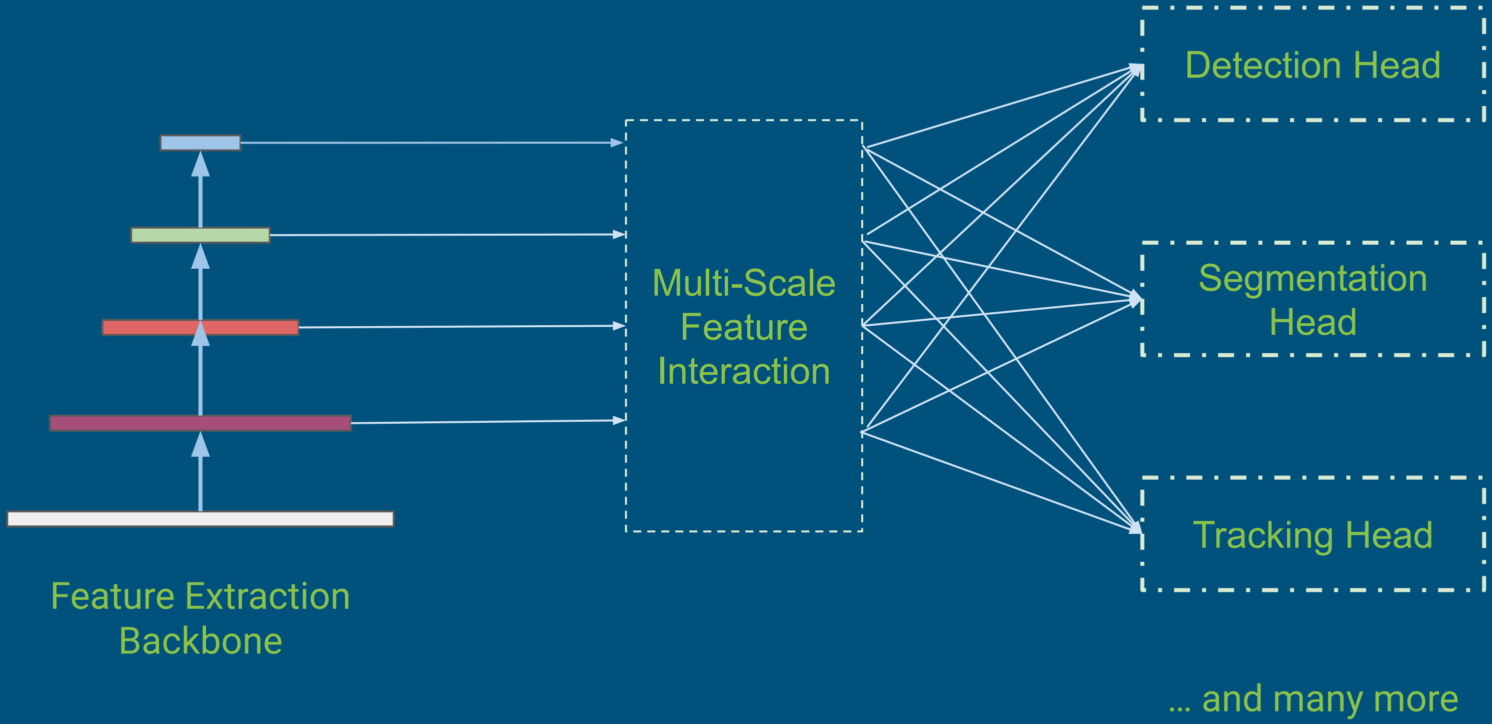


Transfer Learning in Object Detection

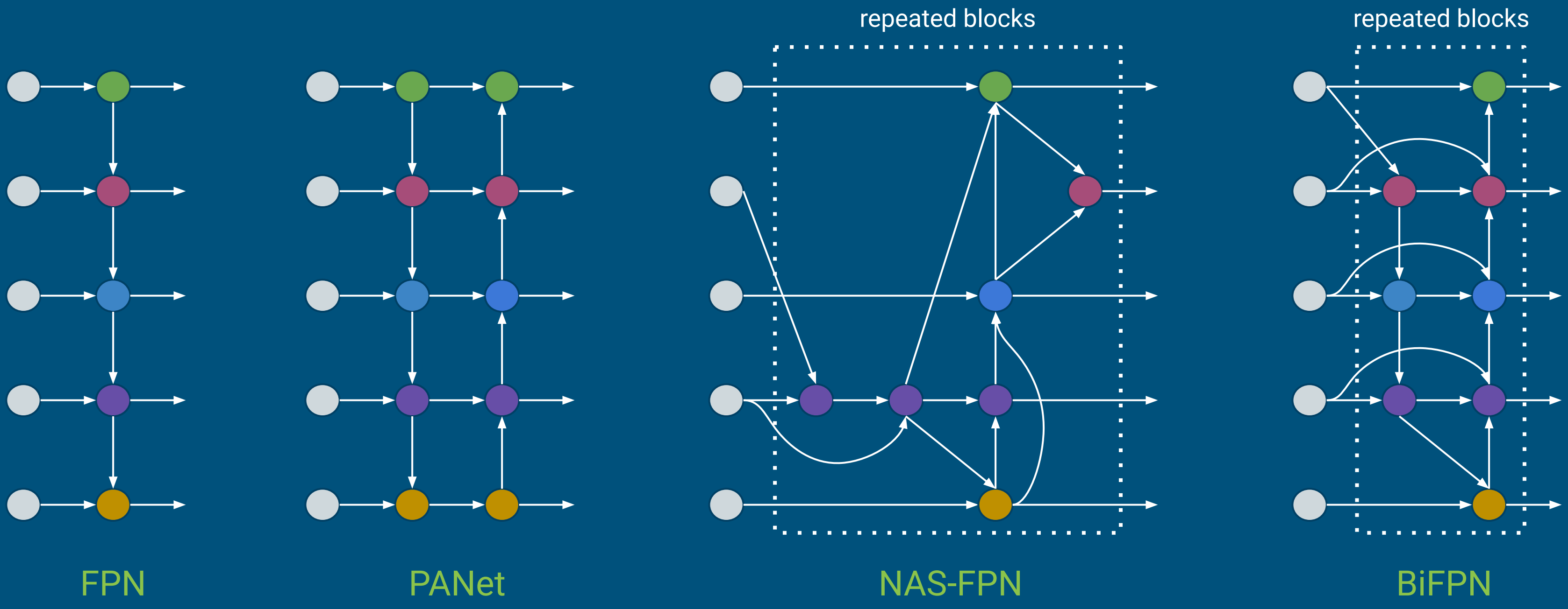
- Transfer learning plays an important role in model training, specially in a low data setting.
- Not every layer of a pre-trained model is equally useful. Initial layers are known to be task-agnostic, and last layers can be task-specific.
- Despite the existence of active research in transfer learning, most SOTA models in vision have not adapted to this behavior.

Neyshabur, Behnam, Hanie Sedghi, and Chiyuan Zhang. "What is being transferred in transfer learning?." NeurIPS. 2020.

Multi-Scale Feature Interaction



Multi-Scale Feature Interaction



Tan, Mingxing, Ruoming Pang, and Quoc V. Le. "EfficientDet: Scalable and efficient object detection." CVPR. 2020.

Multi-Scale Feature Interaction

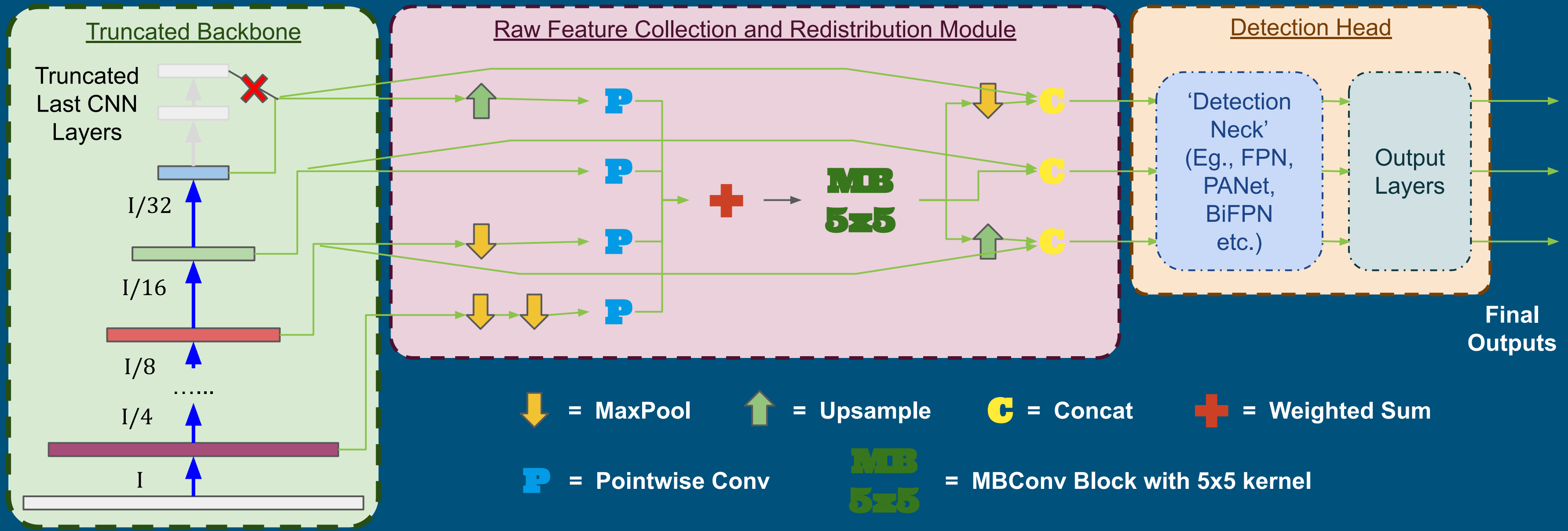
- Existing work focuses on some combination of top-down and/or bottom-up approaches.
- With the increasing complexity of these modules, the tradeoff between accuracy and efficiency has started saturating.
- NAS-based architectures have revealed the importance of direct connections between non-adjacent feature scales.

Ghiasi, Golnaz, Tsung-Yi Lin, and Quoc V. Le. "NAS-FPN: Learning scalable feature pyramid architecture for object detection." CVPR. 2019.

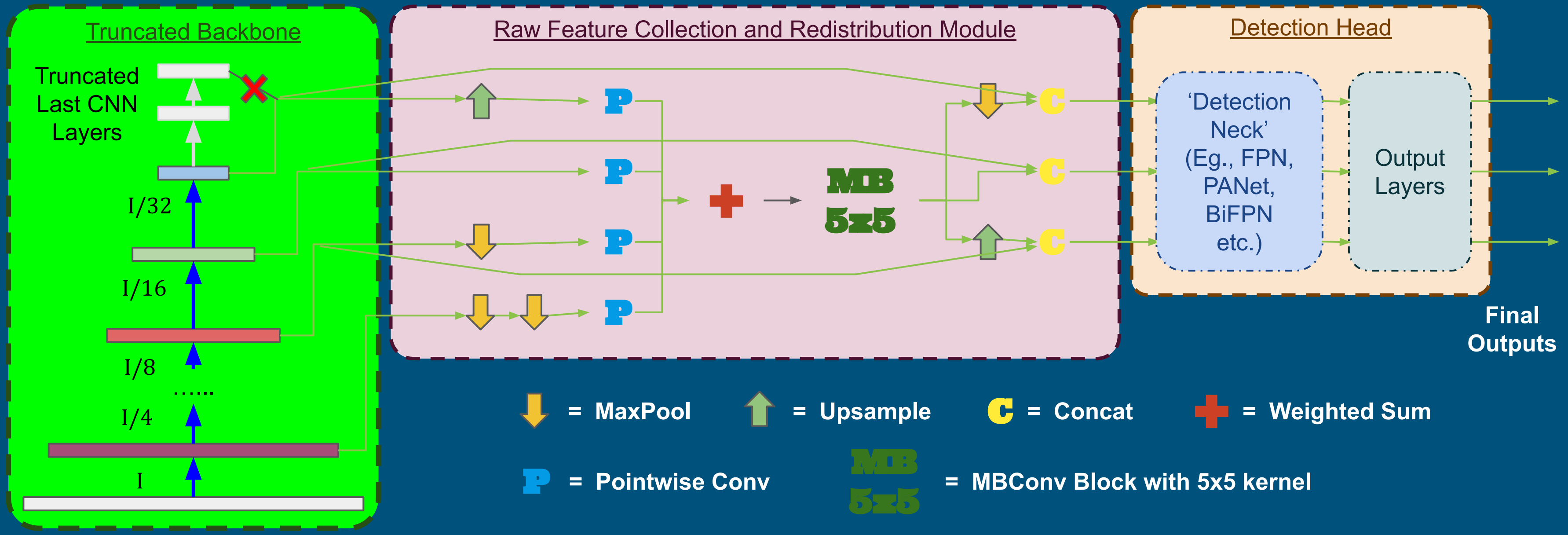
YOLO-ReT

- Importance of Individual Layers
- Backbone Truncation
- Raw Feature Collection and Redistribution

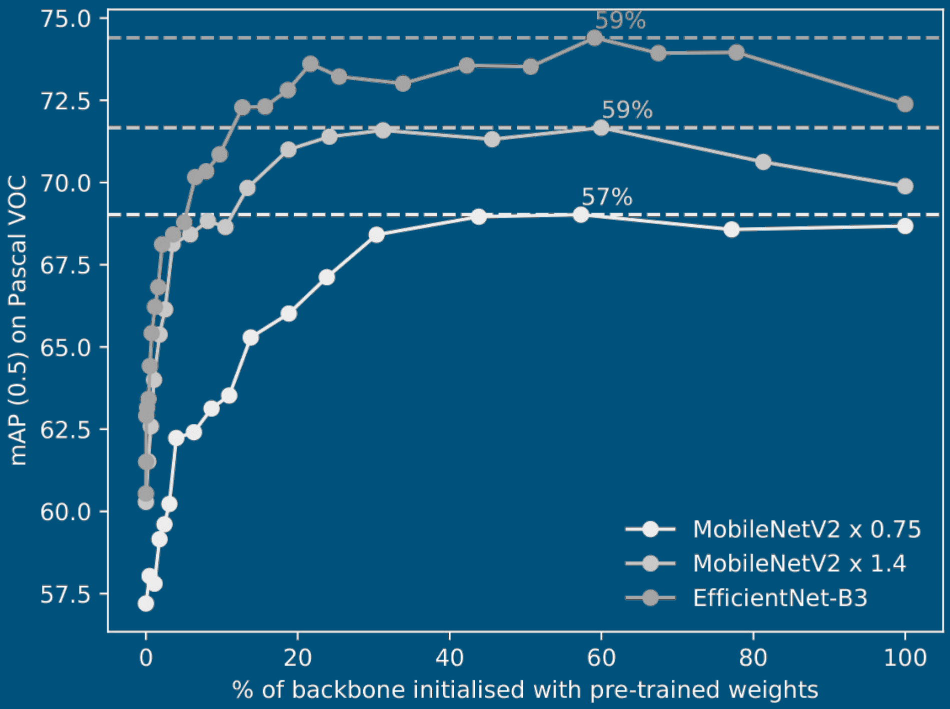
Complete Model



Complete Model



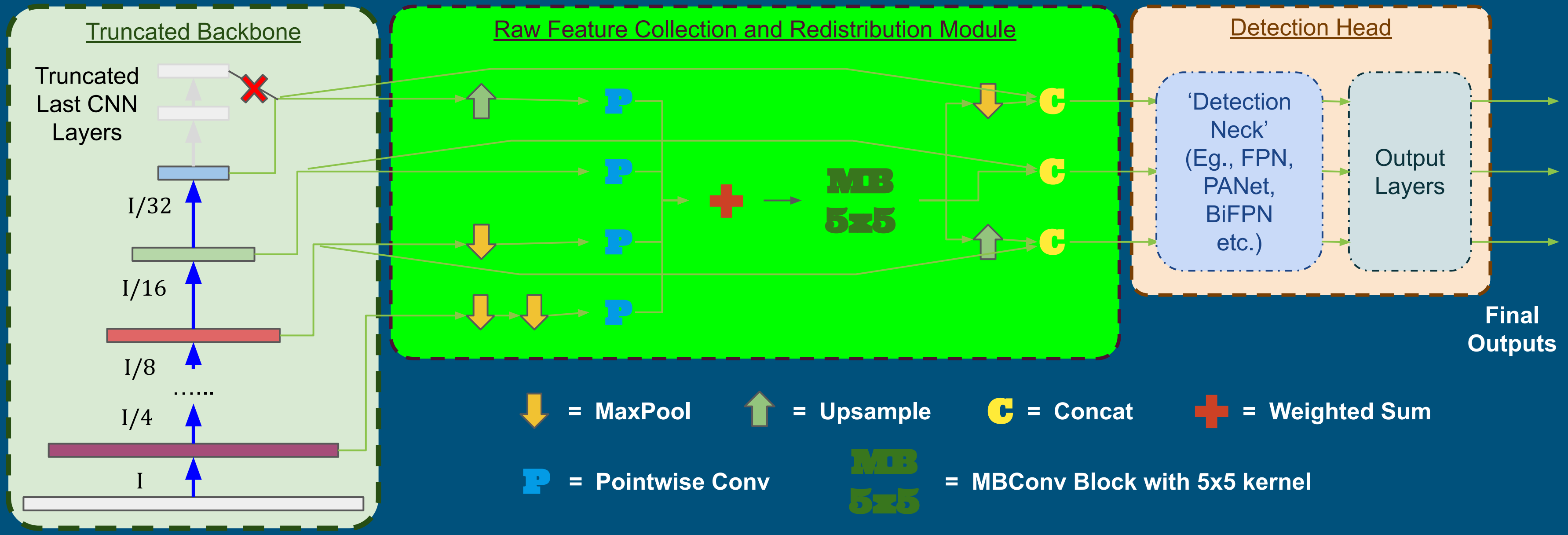
Importance of Individual Layers



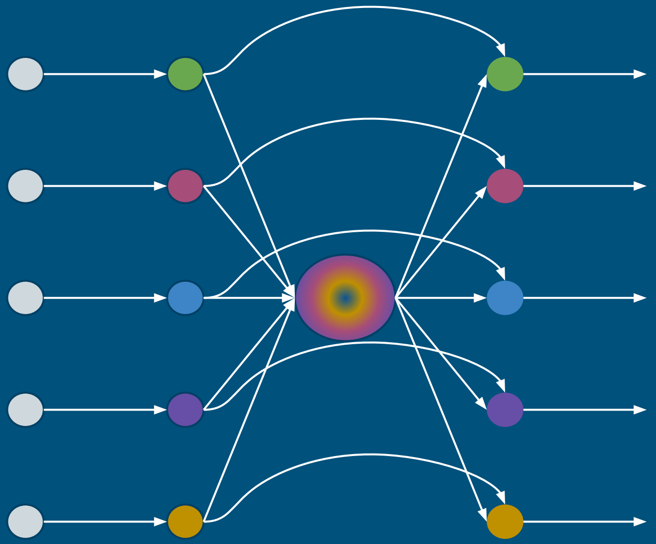
Backbone Truncation

- Initializing the last layers of the feature extraction backbone with transfer learning weights actually ‘hurts’ the performance.
- Since these last layers hold no transfer learning importance, they can be analysed purely from an architecture viewpoint.
- We propose that a truncated version of the feature extraction backbone is a better alternative to width reduction.

Complete Model

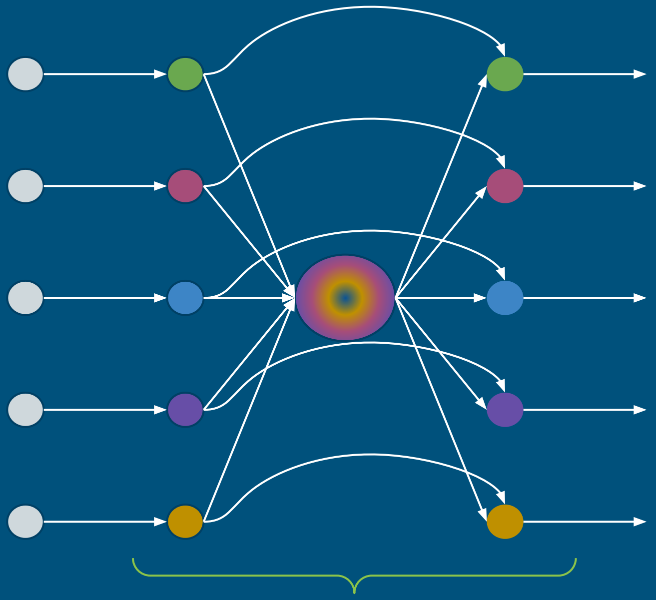


Raw Feature Collection and Redistribution



Collection Redistribution

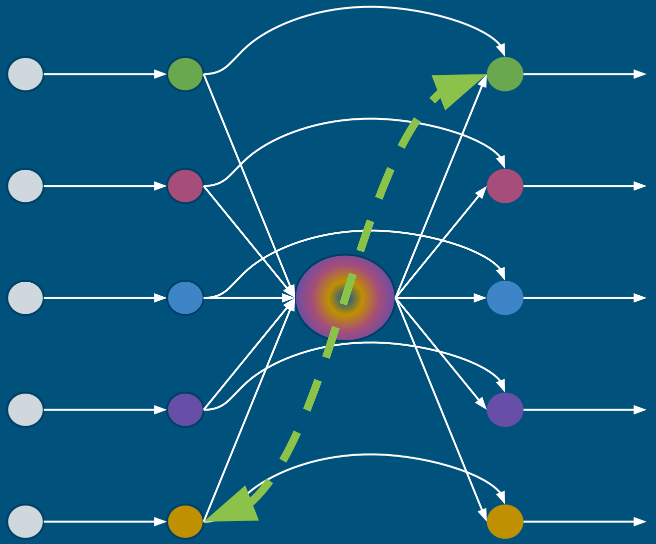
Raw Feature Collection and Redistribution



Simplistic Design

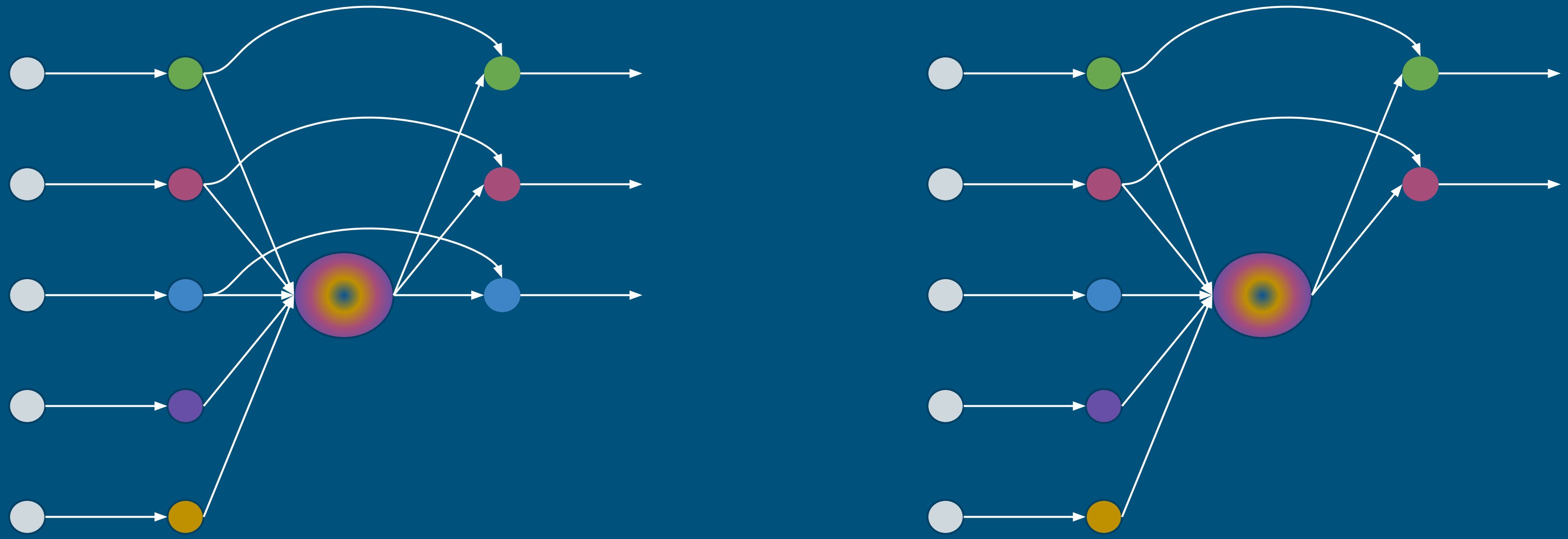
Minimal Network Fragmentation
(Each Collection and Redistribution Path Can be Executed in Parallel)

Raw Feature Collection and Redistribution



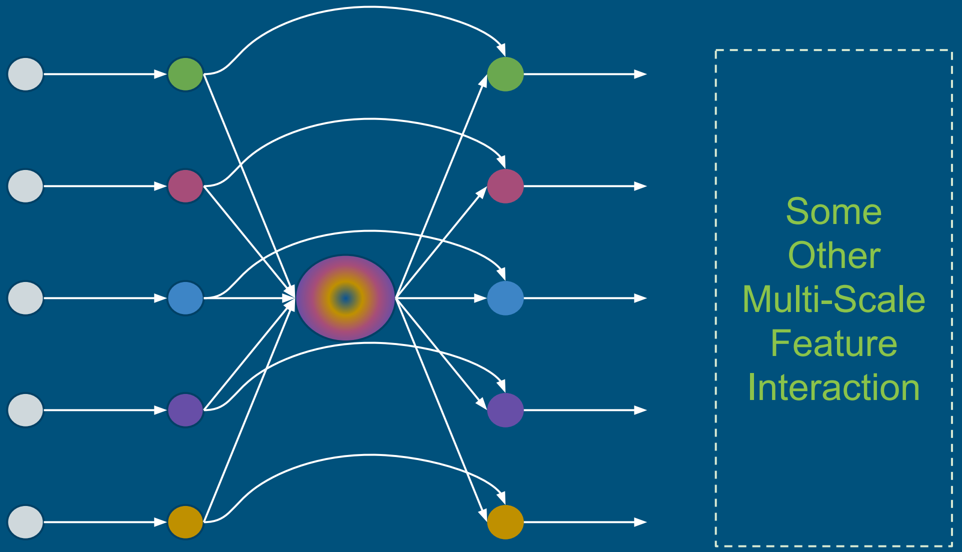
Direct Connection Paths Even Between Non-Adjacent Scales

Raw Feature Collection and Redistribution



Independent of the number of Output Scales

Raw Feature Collection and Redistribution



Cannot replace the meticulousness provided by other Multi-Scale Feature Interaction methods

But can be easily integrated as an additional feature processing

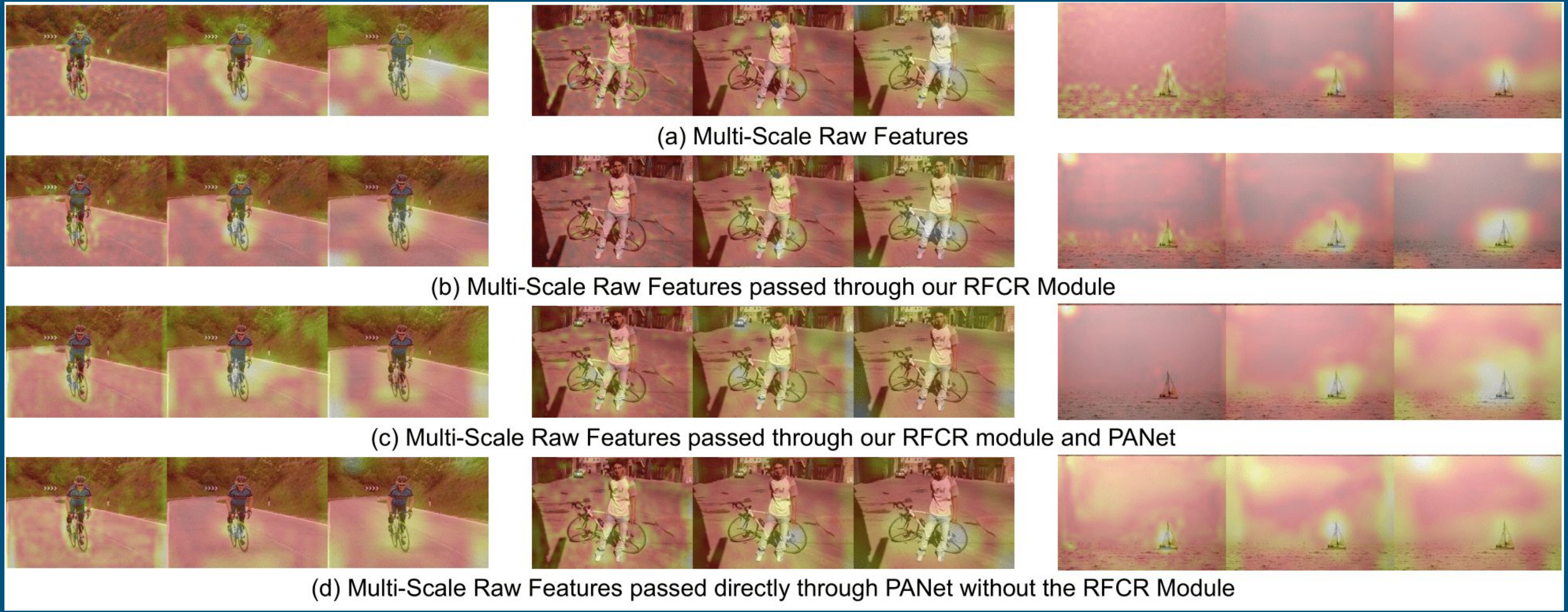
Evaluation

- Experiment Setup
- Component Ablation
- State of the art Models

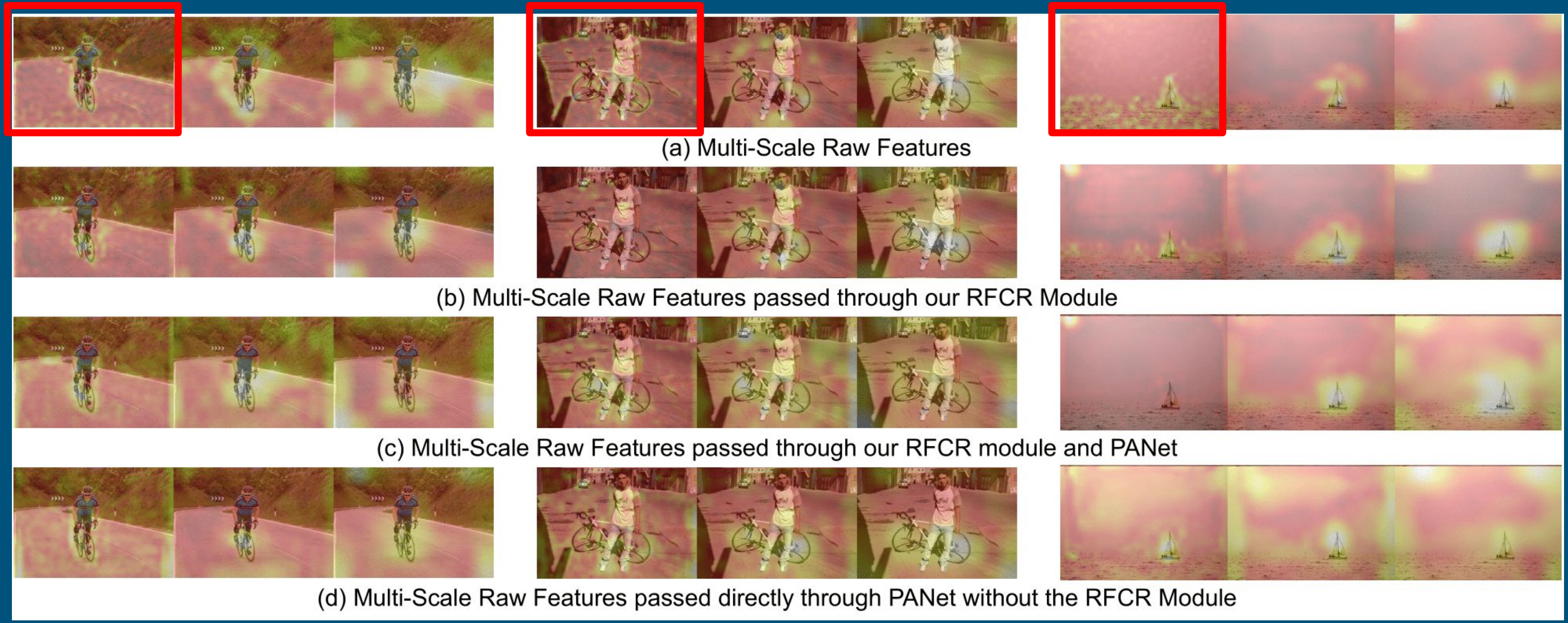
Experiment Setup

- We tested with 3 lightweight feature extraction backbones (MobileNetV2x0.75, x1.4, and EfficientNet-B3) and various feature interaction methods (FPN, PANet and BiFPN).
- We evaluated our methods on Pascal VOC and COCO datasets.
- We tested our models with on-device performance latencies, on Jetson Nano, Jetson Xavier NX and Jetson AGX Xavier.

Qualitative Heatmap Study



Qualitative Heatmap Study



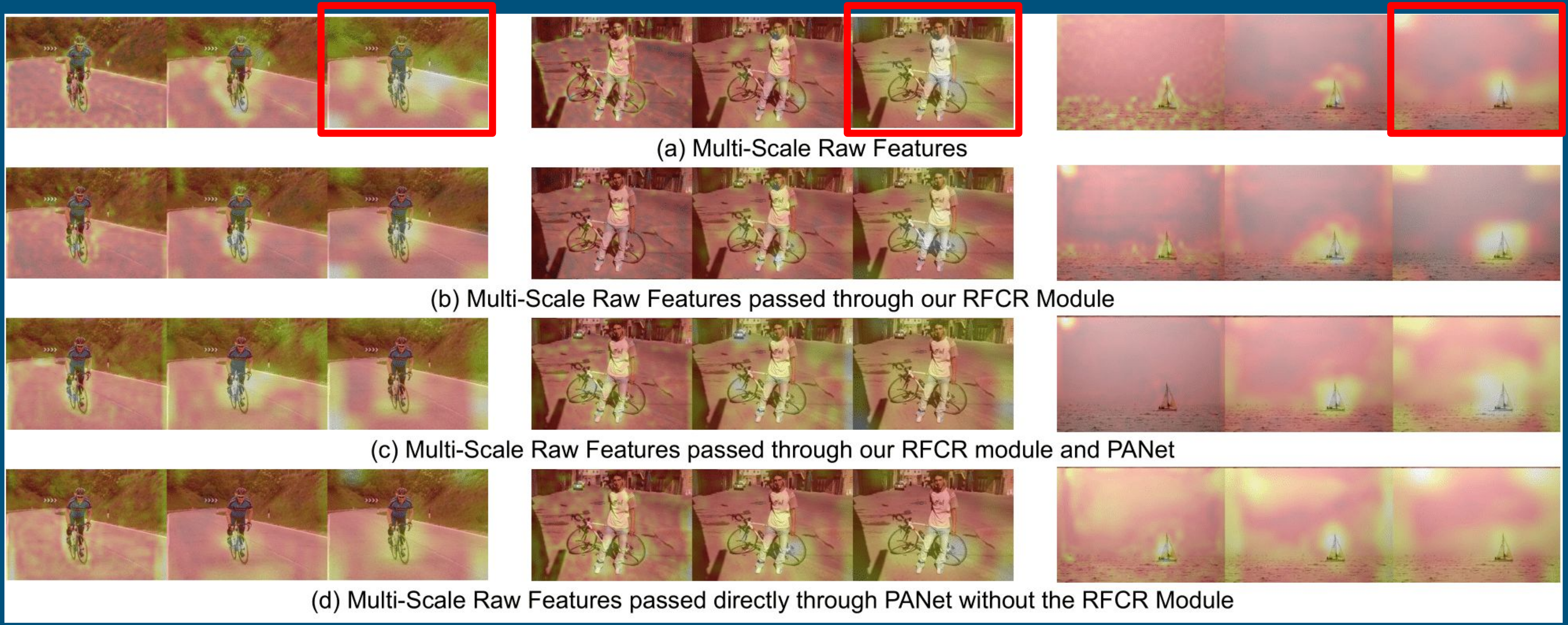
(a) Multi-Scale Raw Features

(b) Multi-Scale Raw Features passed through our RFCR Module

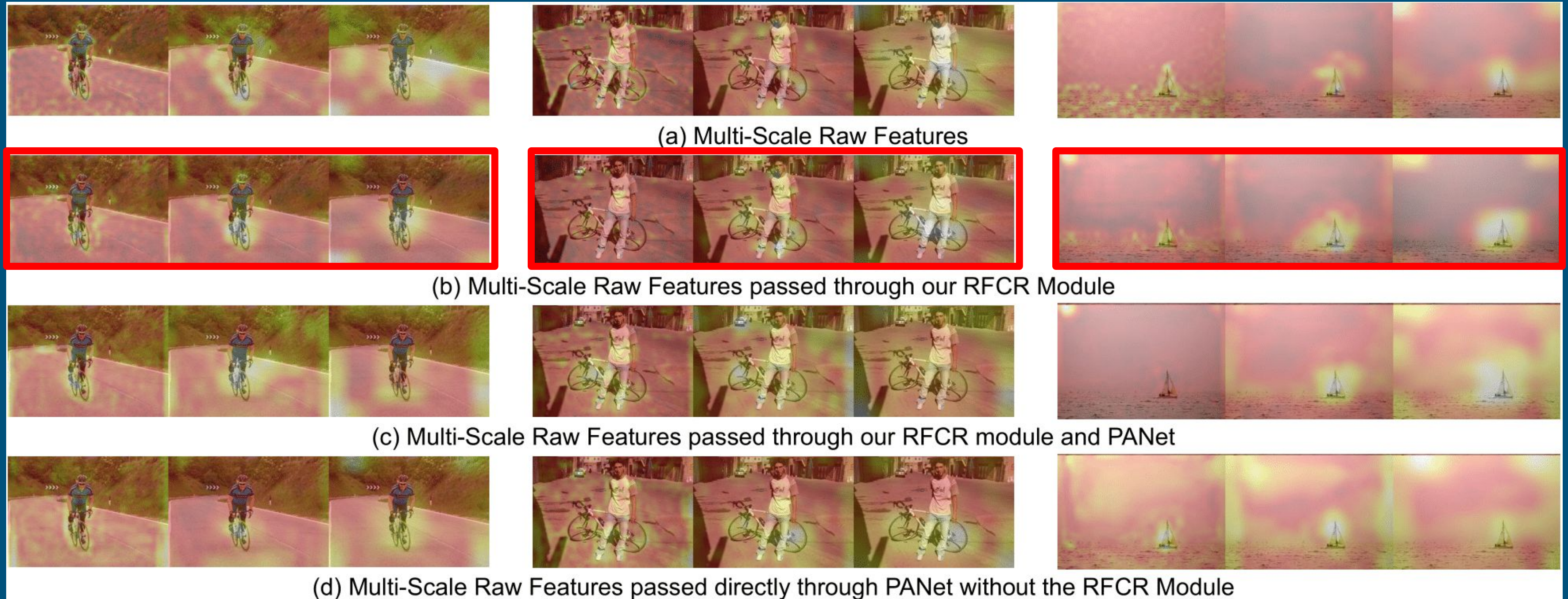
(c) Multi-Scale Raw Features passed through our RFCR module and PANet

(d) Multi-Scale Raw Features passed directly through PANet without the RFCR Module

Qualitative Heatmap Study



Qualitative Heatmap Study


(a) Multi-Scale Raw Features
(b) Multi-Scale Raw Features passed through our RFCR Module
(c) Multi-Scale Raw Features passed through our RFCR module and PANet
(d) Multi-Scale Raw Features passed directly through PANet without the RFCR Module

State of the art Models

Model	Input Resolution	FPS			AP50 (Detailed Results in Paper)	
		Nano	NX	AGX	VOC	COCO
Tiny-YOLOv3	416	27.36	66.55	91.71	61.30	33.10
Tinier-YOLO	416	30.14	68.73	92.09	65.70	34.00
YOLO-ReT-MobileNetV2 x 0.75	320	33.19	71.64	95.97	68.75	34.91
YOLO Nano	416	13.62	54.03	85.81	69.10	--
YOLO-ReT-MobileNetV2 x 1.4	320	23.01	65.37	93.49	70.35	35.77
YOLO Fastest	320	42.41	76.13	126.82	61.02	--
YOLO-ReT-MobileNetV2 x 1.4	224	43.16	84.32	113.94	62.91	31.63

State of the art Models

Model	Input Resolution	FPS			AP^{50} (Detailed Results in Paper)	
		Nano	NX	AGX	VOC	COCO
Tiny-YOLOv3	416	27.36	66.55	91.71	61.30	33.10
Tinier-YOLO	416	30.14	68.73	92.09	65.70	34.00
YOLO-ReT-MobileNetV2 x 0.75	320	**33.19**	**71.64**	**95.97**	**68.75**	**34.91**
YOLO Nano	416	13.62	54.03	85.81	69.10	--
YOLO-ReT-MobileNetV2 x 1.4	320	**23.01**	**65.37**	**93.49**	**70.35**	**35.77**
YOLO Fastest	320	42.41	76.13	**126.82**	61.02	--
YOLO-ReT-MobileNetV2 x 1.4	224	**43.16**	**84.32**	113.94	**62.91**	**31.63**

Code available at : github.com/prakharg24/yoloret

Thank You