JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

DynamicDet: A Unified Dynamic Architecture for Object Detection

Zhihao Lin   Yongtao Wang   Jinhe Zhang   Xiaojie Chu

Wangxuan Institute of Computer Technology, Peking University

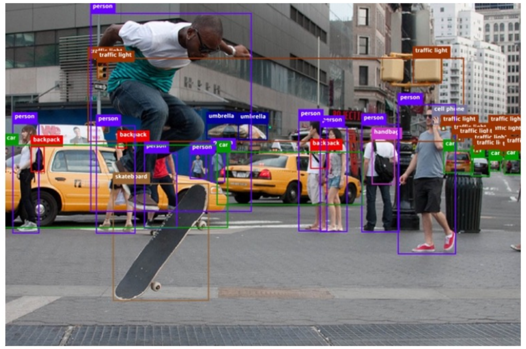
Poster tag: TUE-PM-206

Contact us: linzhihao@stu.pku.edu.cn, wyt@pku.edu.cn

Paper: https://arxiv.org/abs/2304.05552

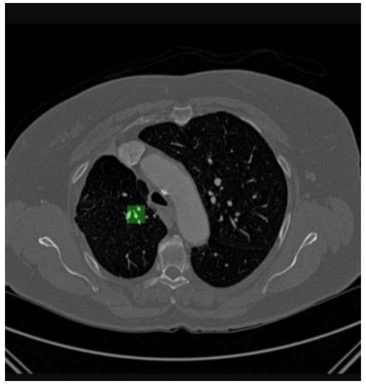
Dynamic neural network

“Easy” image

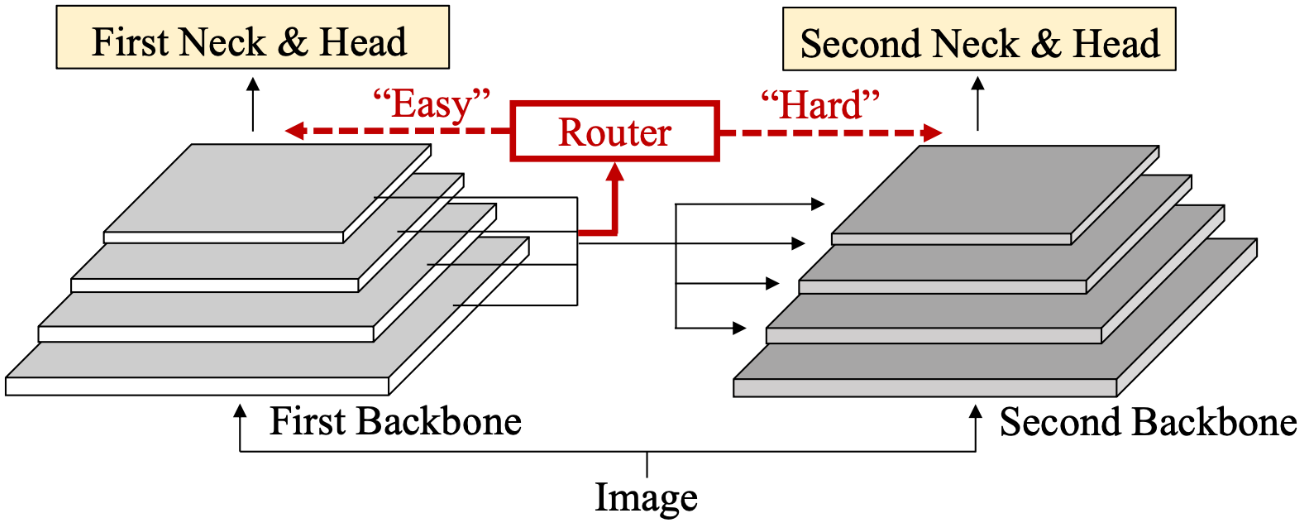


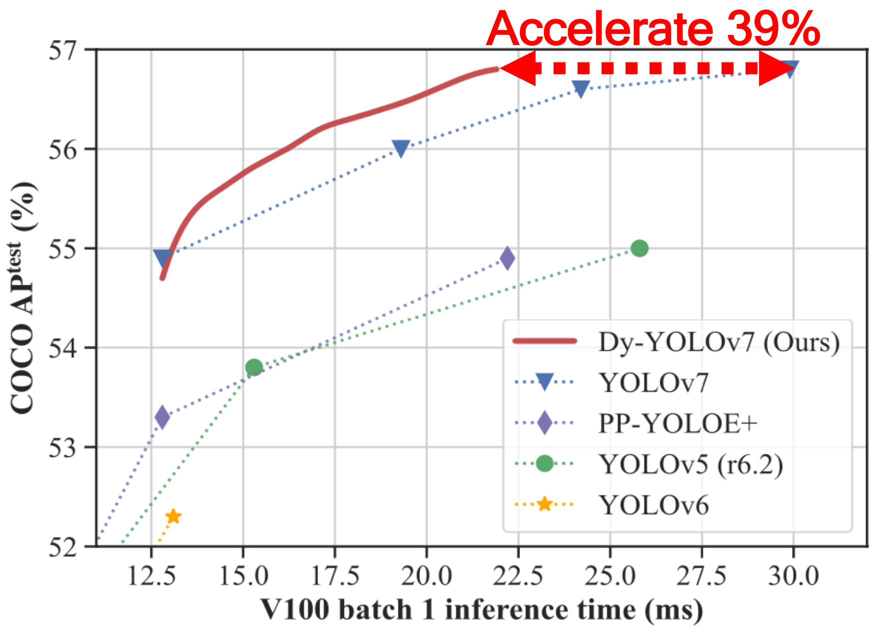
“Hard” image

Object detection



A unified dynamic architecture for object detection





Outline

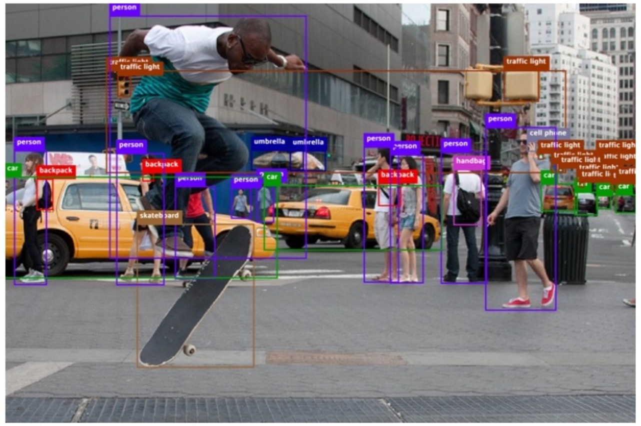
- Introduction
- Approach
  - Overall architecture
  - Adaptive router
  - Optimization strategy
  - Variable-speed inference
- Experiments
- Conclusion

Outline

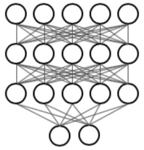
- Introduction
- Approach
  - Overall architecture
  - Adaptive router
  - Optimization strategy
  - Variable-speed inference
- Experiments
- Conclusion

Dynamic neural network

"Easy" image



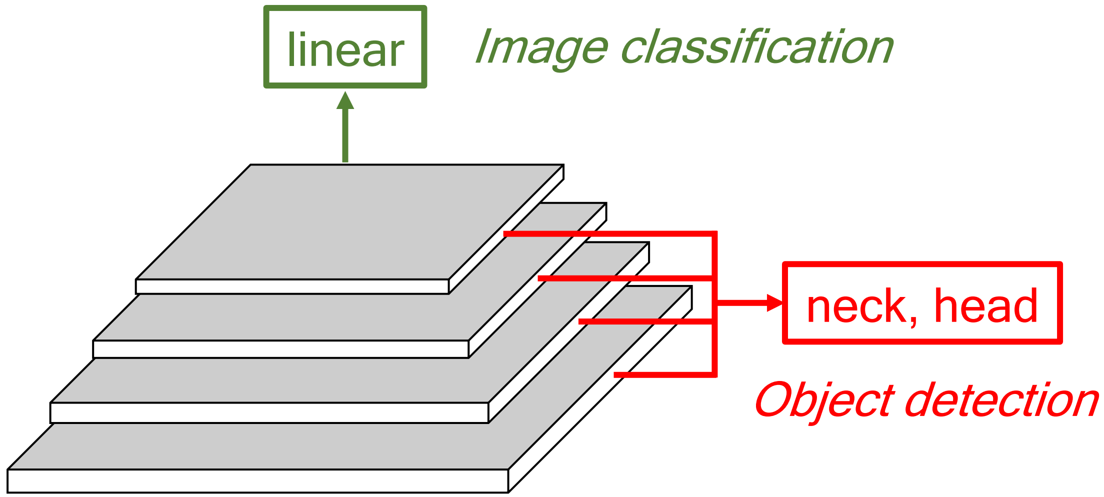
"Hard" image

Human brain  → Neural network 

Early exiting!

Dynamic object detection?

- Decoder for image classification task
  - Linear layer
    - light
    - single-scale feature
- Decoder for object detection task
  - Neck and head
    - heavy
    - multi-scale features

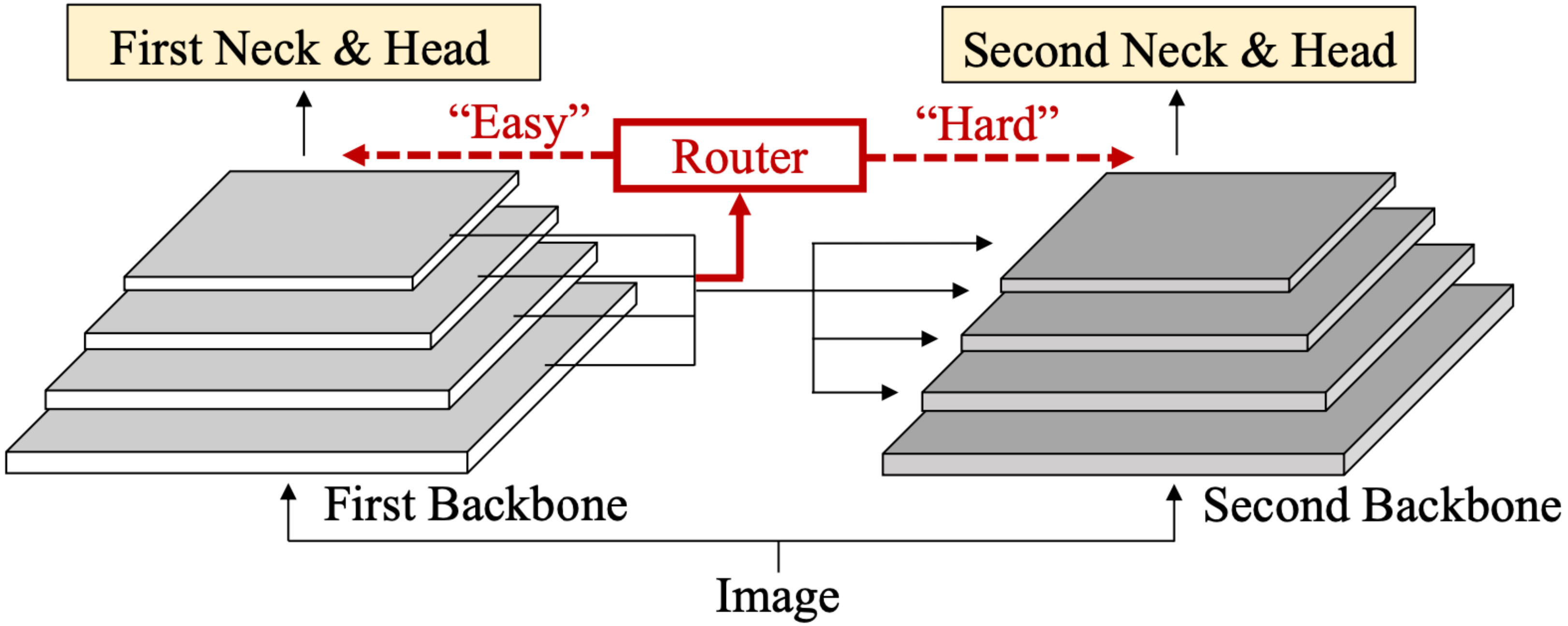


Outline

- Introduction
- Approach
  - Overall architecture
  - Adaptive router
  - Optimization strategy
  - Variable-speed inference
- Experiments
- Conclusion

Overall architecture

- Two detectors, one router
  - Evolved from CBNet*



* T. Liang et al. CBNet: A composite backbone network architecture for object detection. IEEE TIP, 2022.

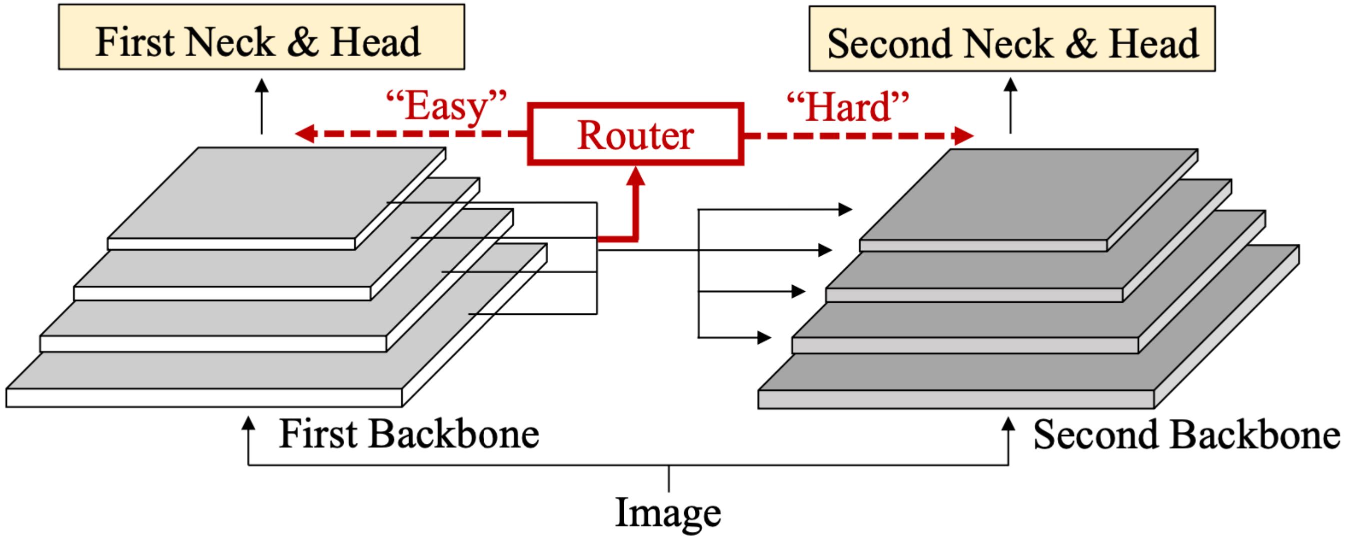
Adaptive router

- **Input:** multi-scale features $F_1 = \{f_1^{\{1\}}, f_1^{\{2\}}, \ldots, f_1^{\{L\}}\}$
- **Output:** predicted difficulty score $\phi \in \mathbb{R}^1$
  - Compress F_1 to $\tilde{F}_1$
    - $\tilde{F}_1 = \mathcal{C}\left(\mathcal{P}\left(f_1^{\{1\}}\right), \mathcal{P}\left(f_1^{\{2\}}\right), \ldots, \mathcal{P}\left(f_1^{\{L\}}\right)\right).$
  - Map $\tilde{F}_1$ to ϕ
    - $\phi = \sigma(W_2\left(\delta(W_1\tilde{F}_1 + b_1)\right) + b_2).$

Optimization strategy

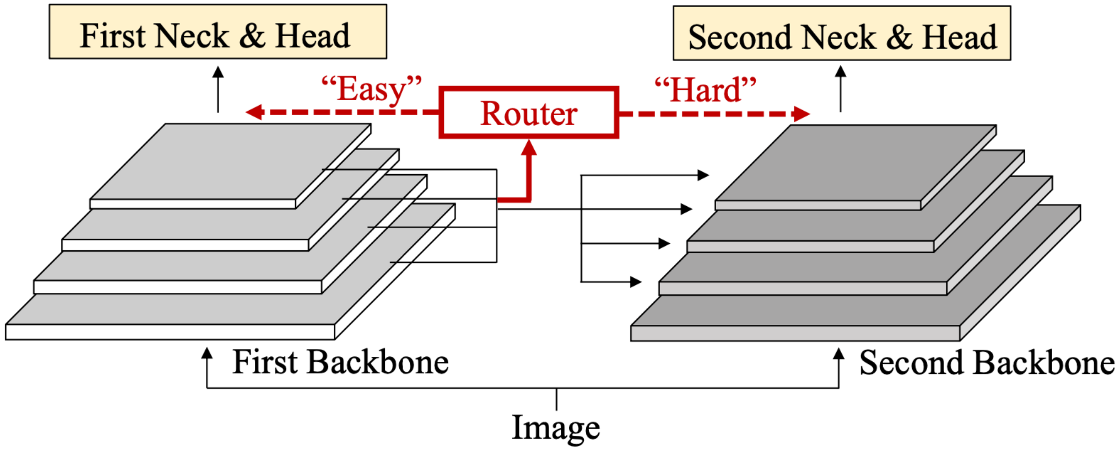
- Step 1, jointly train the cascaded detectors

$$\min_{\Theta_1,\Theta_2} \left(\mathcal{L}_{det}^{\{1\}}(\mathbf{x},\mathbf{y}|\Theta_1) + \mathcal{L}_{det}^{\{2\}}(\mathbf{x},\mathbf{y}|\Theta_2)\right)$$



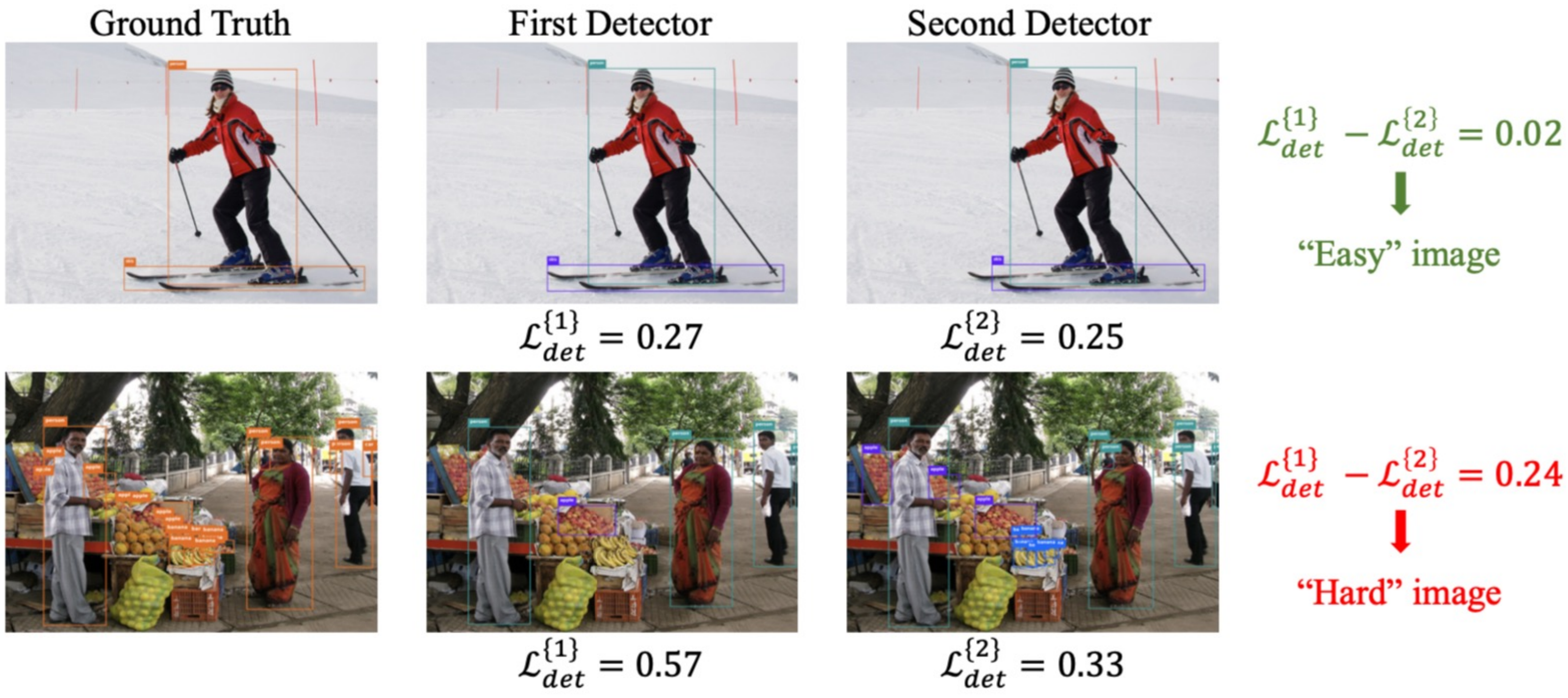
Optimization strategy

- Step 2, train the adaptive router
  - Freeze the parameters of two detectors
  - Naive methods
    - (1) $\min_{\Theta_{\mathcal{R}}} \left((1-\phi)\mathcal{L}_{det}^{\{1\}}(\mathbf{x},\mathbf{y}|\Theta_1) + \phi\mathcal{L}_{det}^{\{2\}}(\mathbf{x},\mathbf{y}|\Theta_2)\right),$
    - (2) $\min_{\Theta_{\mathcal{R}}} \left((1-\phi)\mathcal{L}_{det}^{\{1\}}(\mathbf{x},\mathbf{y}|\Theta_1) + \phi\mathcal{L}_{det}^{\{2\}}(\mathbf{x},\mathbf{y}|\Theta_2) + \boldsymbol{\lambda\phi}\right).$



Optimization strategy

- Training loss difference between two detectors

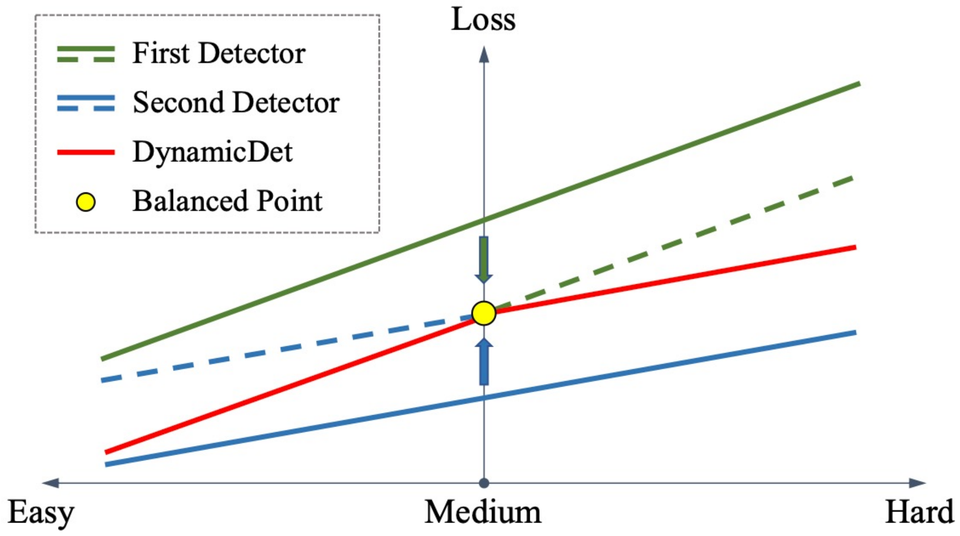


Optimization strategy

- Adaptive offset
  - To balance the losses of two detectors
  - the median of the loss difference on the training set Δ

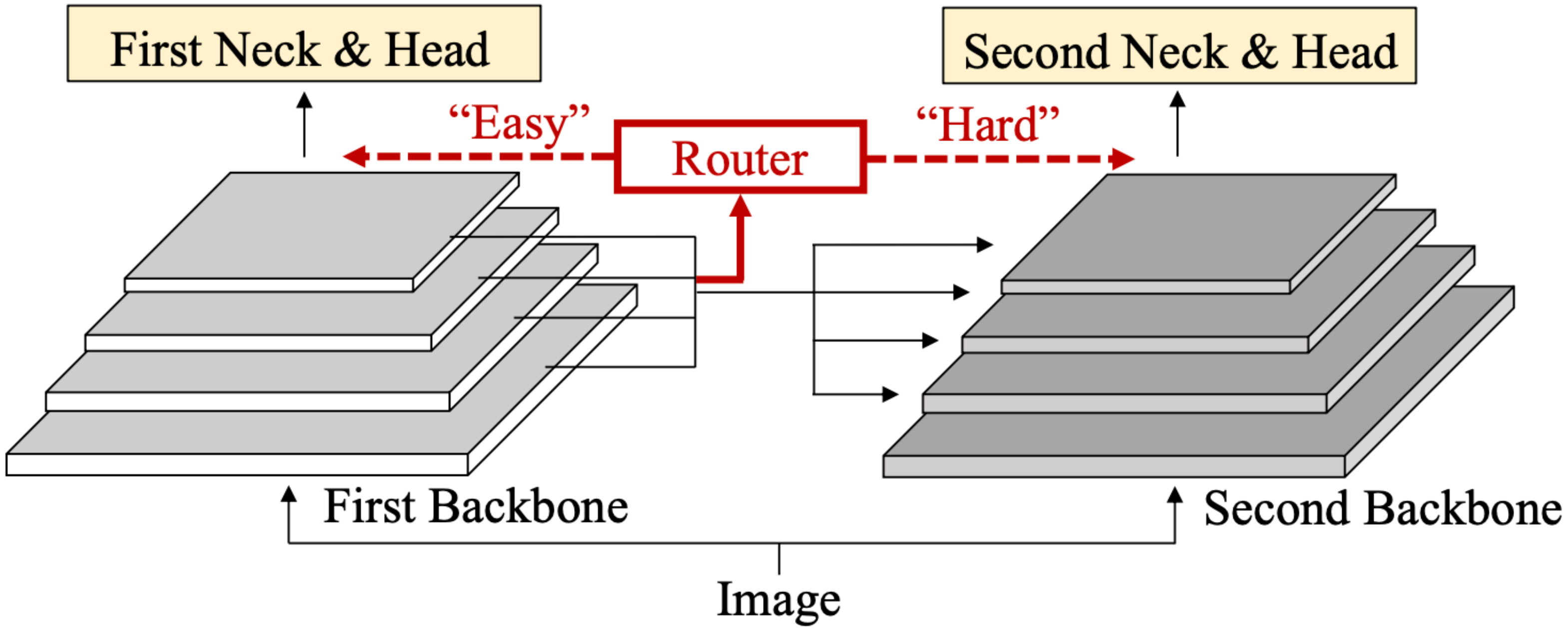
$$\min_{\Theta_{\mathcal{R}}} \left((1-\phi)(\mathcal{L}_{det}^{\{1\}}(\mathbf{x},\mathbf{y}|\Theta_1)-\boldsymbol{\Delta}/\mathbf{2}) + \phi(\mathcal{L}_{det}^{\{2\}}(\mathbf{x},\mathbf{y}|\Theta_2)+\boldsymbol{\Delta}/\mathbf{2})\right)$$

Hyperparameter-free!



Variable-speed inference

- One dynamic detector for a wide range of trade-offs



Variable-speed inference

- How to achieve the target latency by one dynamic detector?
  - Latency of the first detector lat_1
  - Latency of the cascaded two detectors lat_2
  - Target latency lat_t
  - The difficulty scores of the validation set $\mathcal{S}_{val}$

$$k = \frac{lat_t - lat_1}{lat_2 - lat_1}, \quad lat_1 \leq lat_t \leq lat_2,$$

$$\tau_{val} = \text{percentile}(\mathcal{S}_{val}, k),$$

$$\boxed{\tau_{test} = \tau_{val}.}$$ (these two sets are i.i.d.)

Outline

- Introduction
- Approach
  - Overall architecture
  - Adaptive router
  - Optimization strategy
  - Variable-speed inference
- Experiments
- Conclusion

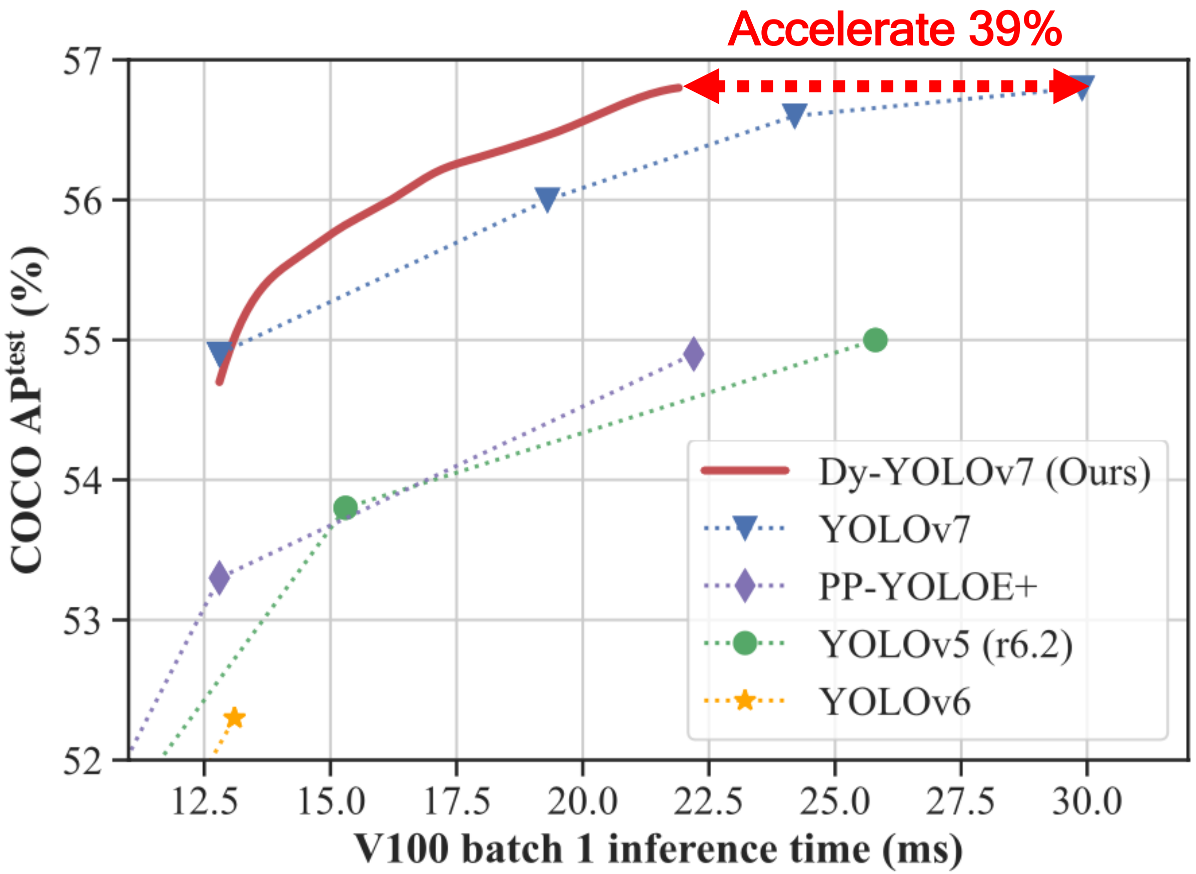
State-of-the-art trade-offs

Model	Size	FLOPs	FPS	AP
EAutoDet-X [48]	640	225.3G	41†	49.2
YOLOX-L [9]	640	155.6G	69†	50.1
YOLOX-X [9]	640	281.9G	58†	51.5
YOLOv5-L (r6.2) [11]	640	109.1G	114	49.0
YOLOv5-X (r6.2) [11]	640	205.7G	100	50.9
YOLOv6-M [21]	640	82.2G	109	49.6
YOLOv6-L [21]	640	144.0G	76	52.4
PP-YOLOE+-M [53]	640	49.9G	123†	50.0
PP-YOLOE+-L [53]	640	110.1G	78†	53.3
PP-YOLOE+-X [53]	640	206.6G	45†	54.9
YOLOv7 [45]	640	104.7G	114	51.4
Dy-YOLOv7 / 10	**640**	**112.4G**	**110**	**52.1**
Dy-YOLOv7 / 50	**640**	**143.2G**	**96**	**53.3**
Dy-YOLOv7 / 90	**640**	**174.0G**	**85**	**53.8**
Dy-YOLOv7 / 100	**640**	**181.7G**	**83**	**53.9**
YOLOv7-X [45]	640	189.9G	105	53.1
Dy-YOLOv7-X / 10	**640**	**201.7G**	**98**	**53.3**
Dy-YOLOv7-X / 50	**640**	**248.9G**	**78**	**54.4**
Dy-YOLOv7-X / 90	**640**	**296.1G**	**65**	**55.0**
Dy-YOLOv7-X / 100	**640**	**307.9G**	**64**	**55.0**

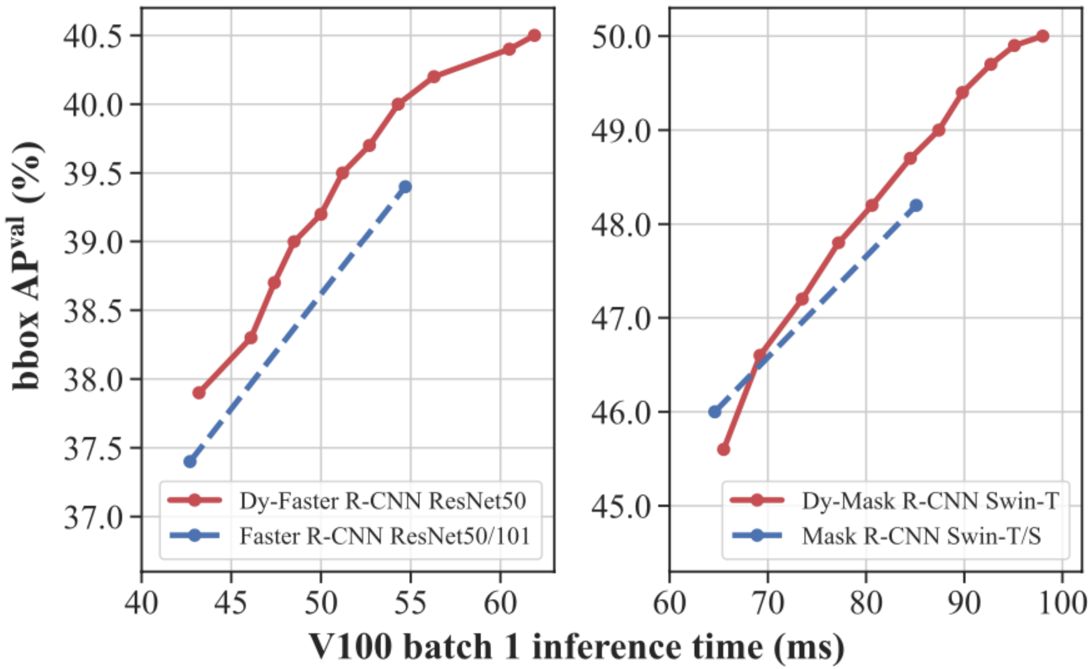
Model	Size	FLOPs	FPS	AP
YOLOv5-M6 (r6.2) [11]	1280	200.0G	96	51.4
YOLOv5-L6 (r6.2) [11]	1280	445.6G	65	53.8
YOLOv5-X6 (r6.2) [11]	1280	839.2G	39	55.0
YOLOv7-W6 [45]	1280	360.0G	78	54.9
YOLOv7-E6 [45]	1280	515.2G	52	56.0
YOLOv7-D6 [45]	1280	806.8G	41	56.6
YOLOv7-E6E [45]	1280	843.2G	33	56.8
Dy-YOLOv7-W6 / 10	**1280**	**384.2G**	**74**	**55.2**
Dy-YOLOv7-W6 / 50	**1280**	**480.8G**	**58**	**56.1**
Dy-YOLOv7-W6 / 90	**1280**	**577.4G**	**48**	**56.7**
Dy-YOLOv7-W6 / 100	**1280**	**601.6G**	**46**	**56.8**

[1] The FPS marked with † are from the corresponding papers, and others are measured on the same machine with 1 NVIDIA V100 GPU.

State-of-the-art trade-offs



Generality for two-stage detectors



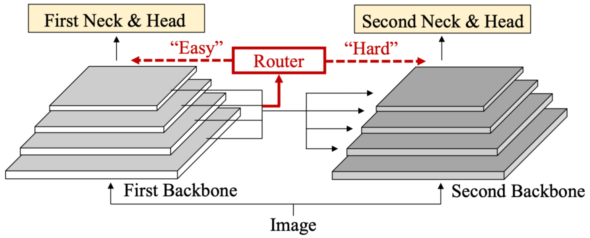
Model	FLOPs	FPS	AP_{box}	AP_{mask}
Faster R-CNN ResNet50 [15, 39]	207.1G	23	37.4	-
Faster R-CNN ResNet101 [15, 39]	283.1G	18	39.4	-
Dy-Faster R-CNN ResNet50 / 50	**245.4G**	**20**	**39.5**	-
Dy-Faster R-CNN ResNet50 / 90	**276.0G**	**17**	**40.4**	-
Mask R-CNN Swin-T [14, 31]	263.8G	15	46.0	41.6
Mask R-CNN Swin-S [14, 31]	353.8G	12	48.2	43.2
Dy-Mask R-CNN Swin-T / 50	**310.6G**	**12**	**48.7**	**43.6**
Dy-Mask R-CNN Swin-T / 90	**348.0G**	**11**	**49.9**	**44.2**

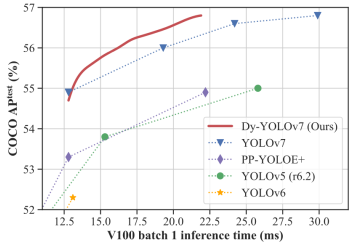
Visualization

- Easy: fewer objects, usual camera viewpoint, clean background
- Hard: more small objects, complex scenes, severe occlusion

Outline

- Introduction
- Approach
  - Overall architecture
  - Adaptive router
  - Optimization strategy
  - Variable-speed inference
- Experiments
- Conclusion





Conclusion

- A unified dynamic architecture for object detection, <u>DynamicDet</u>
  - **Dynamic architecture** to support dynamic inference on detectors
  - **Adaptive router** to predict the difficulty score of each image and determine the inference route
  - **Hyperparameter-free optimization strategy** with an adaptive offset to training the dynamic detectors
  - **Variable-speed inference strategy** for model deployment
  - Achieve <u>a wide range of state-of-the-art accuracy-speed trade-offs</u> with only one dynamic detector

Thanks!

Contact us: linzhihao@stu.pku.edu.cn, wyt@pku.edu.cn

Paper: https://arxiv.org/abs/2304.05552