Early Exit Yolo in video object detection

Paul Gay

GreenAI U.P.P.A.

03-28-2022

How much do we need to compute ?

Related work

Different research axis for light AI

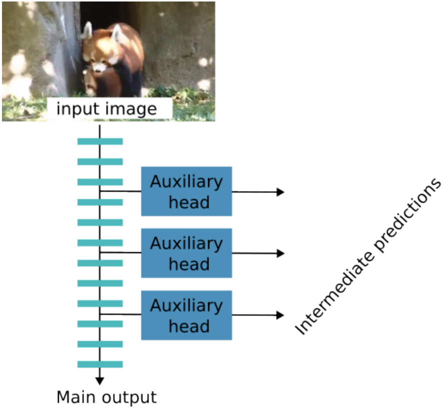
- Architecture search
- Quantization
- Pruning
- Distillation
- Dynamic inference and **Early Exit**

Today work is the application of Early Exit for Object detection in videos

Early exit Intuition

Not all images are equal

Early exit principle



- Reduce computation by predicting with early good enough features
- Adding additional **Auxiliary heads**

Motivations

Some substantial advantages

- Obviously, reduce the computation
- One flexible model rather than multiple ones
- Share computation across multiple devices [Laskaridis et al., 2020, Leontiadis et al., 2021]
- A venue to study otherthinking [Kaya et al., 2019]
- Leverage ensemble techniques to build confidence measure [Wang et al., 2020, Hu et al., 2020, Qendro et al., 2021]

Dynamic inference related work

Some questions

- Which loss function?
  - Often, simple sum of losses
  - Accuracy and Computation cost are heterogeneous and non differentiable terms
- Where to place the heads ?
  - compromise between computational cost and accuracy [Lin et al., 2022, Huang et al., 2017, Bakhtiarnia et al., 2021]
- When to exit?
  - simple strategy to check the score entropy
  - Gating mechanism with Gumble Softmax trick [Veit and Belongie, 2018]
  - Learning halting scores [Figurnov et al., 2017]
  - Reinforcement learning [Bolukbasi et al., 2017, Wang et al., 2018, Guan et al., 2017]

Application to video

Smooth videos seems a natural application for Early Exit mechanism

- A central topic in video models :

  Re-use previous features to avoid redundant computation

Related work on video processing

Remove redundancy on videos along the resolution, temporal and network depth dimensions

- Propagate features
  - LSTM [Liu and Zhu, 2018], Optical Flow [Zhu et al., 2018, Wang et al., 2021], Attention [Guo et al., 2019, Jiang et al., 2020], and tracking [Lu et al., 2020, Feichtenhofer et al., 2017]
- Different branches in parallel [Wu et al., 2019, Feichtenhofer et al., 2019, Sabet et al., 2021] or predefined check points [Wu et al., 2020]

In general, handmade architecture tradeoff.

Our proposal

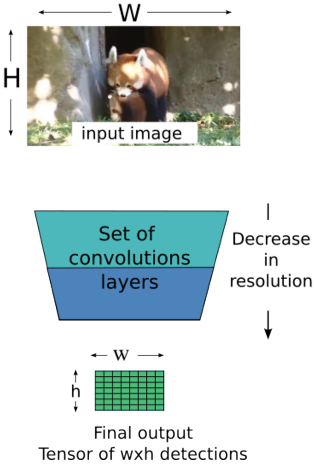
Our motivation is that Early Exit is a flexible an automatic mechanism to explore the tradeoff between accuracy and cost in smooth videos.

Our proposal

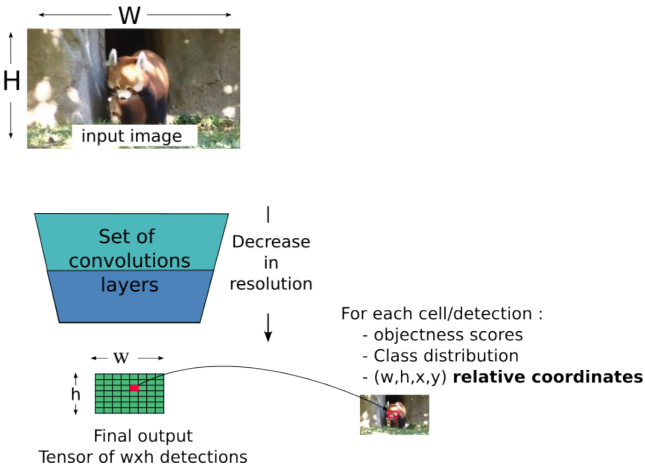
Our motivation is that Early Exit is a flexible an automatic mechanism to explore the tradeoff between accuracy and cost in smooth videos.

I'll try to show it with the yolov5 model

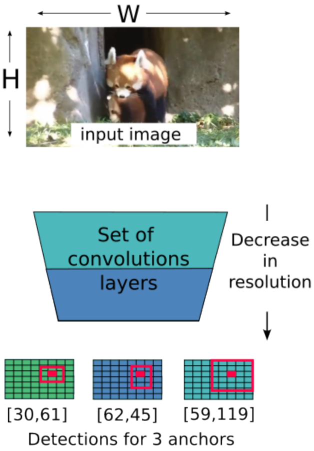
Yolo Principle



Yolo Principle

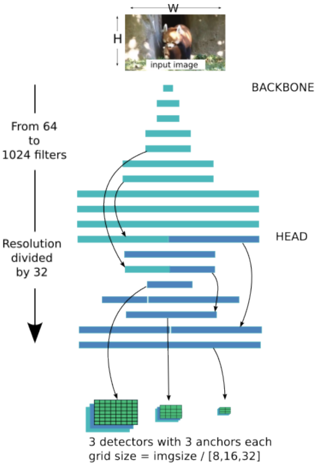


Yolo Principle



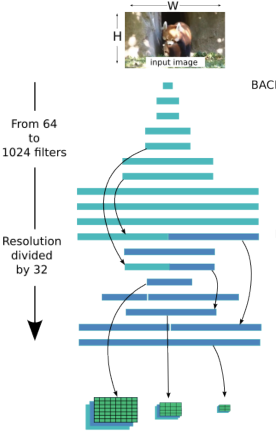
For each cell/detection :
- (x,y) offsets w.r.t the cell center
- (w,h) offsets w.r.t a given anchor box

Yolo in more details



Plus scale coumpound strategy (yolov5n/s/m/l/x)

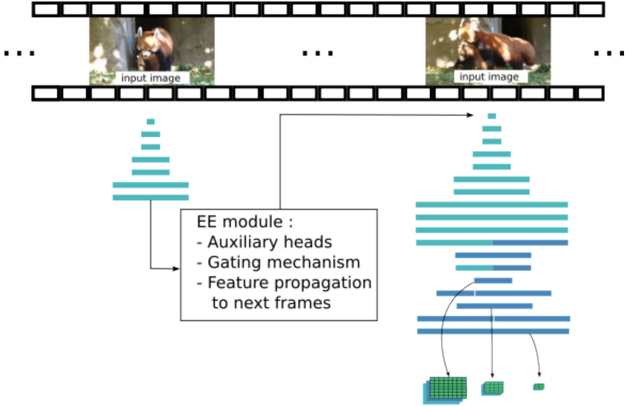
Yolo in more details



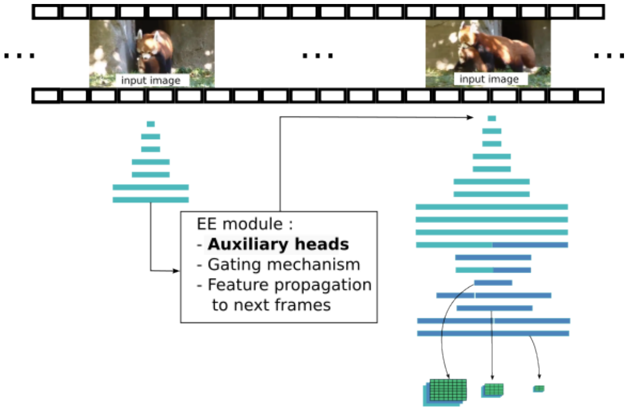
time(ms)	GFLOPs	#params (10^3)
7.83	0.72	3
6.50	0.95	18
15.20	0.96	18
4.15	0.95	73
10.30	1.48	115
3.00	0.95	295
8.51	2.00	625
2.79	0.94	1180
4.01	0.95	1182
3.75	0.53	656
0.45	0.11	131
5.81	1.16	361
0.56	0.11	33
9.59	1.16	90
1.54	0.47	147
4.75	0.95	296
1.29	0.47	590
3.95	0.95	1182
4.03	0.74	229
Total :		
98	16.5	7235

Ok, so how do we apply it to a video?

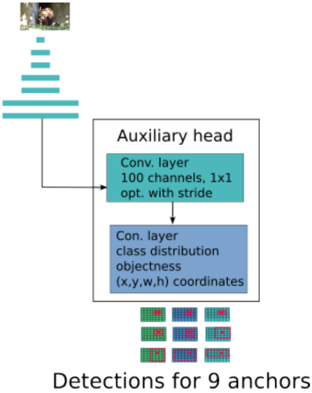
Early Exit for efficient video processing



Early Exit for efficient video processing



Auxiliary head implementation



- Simplify the head to keep the computation in the backbone
  - Backbone computation benefit to the next heads
  - Limit the modification : easy to adapt to new models
- 100 channel layer to extract an uniform representation

Experiment on Coco dataset

- Adding auxiliary heads to all Yolov5 layers
- Test with 2 different training
  - Train everything from scratch
  - Start from a pre-trained yolo and fine tune only the head
- Recording the mean AP for each head and each epoch
- No test time augmentation / no data augmentation

Results

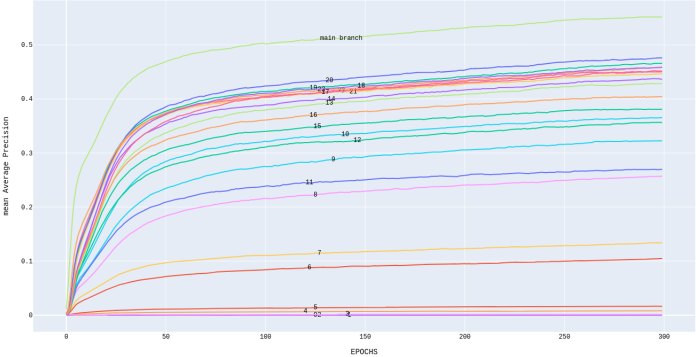


Figure 2: Traning everthing from scratch

- Accuracy increases with the depth
- Note that adding auxiliary heads did not harm initial accuracy

Results

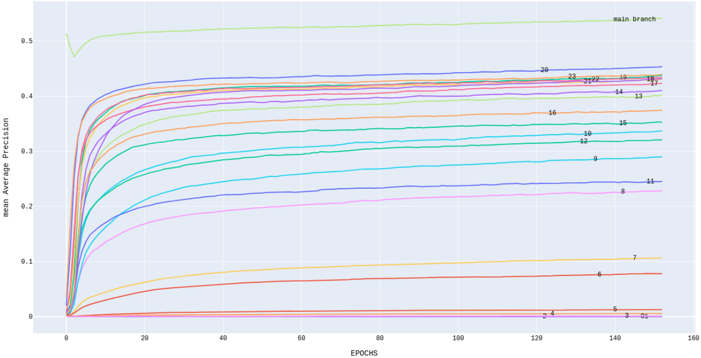


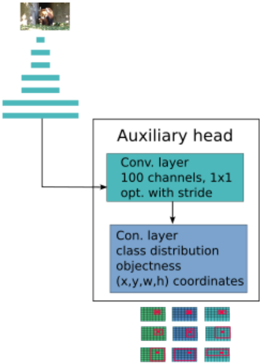
Figure 3: Freezed pre-trained model and fine-tuning on Aux. heads

- Results do no vary massively with the change of strategy
- Usecase: benefit from an expensive pretrained model

Cost of this method

What is the cost of this simple and small Auxiliary head?

Auxiliary head implementation



Detections for 9 anchors

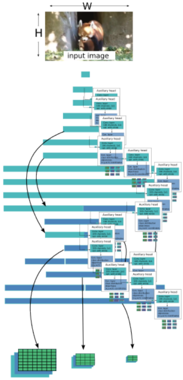
Number of parameters :

$$c\_in \times 100 + 100 \times n\_out \times n\_anch$$

Gflops :

$$c\_in \times Hout \times Wout \times 100$$
$$+$$
$$100 \times Hout \times Wout \times n\_out \times n\_anch$$

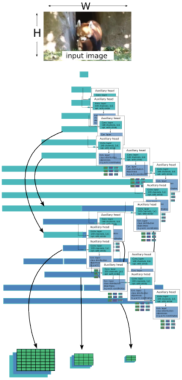
Auxiliary head implementation



Main branch time(ms)	GFLOPs	#params (10^3)	Aux. heads time(ms)	GFLOPs	#params ( )
7.83	0.72	3	9.79	1.03	80
6.50	0.95	18	8.96	1.07	83
15.20	0.96	18	8.76	1.07	83
4.15	0.95	73	8.65	1.16	90
10.30	1.48	115	8.57	1.16	90
3.00	0.95	295	2.05	0.33	103
8.51	2.00	625	2.50	0.33	103
2.79	0.94	1180	0.62	0.10	128
4.01	0.95	1182	0.68	0.10	128
3.75	0.53	656	0.71	0.10	128
0.45	0.11	131	0.58	0.08	103
5.81	1.16	361	2.57	0.33	103
0.56	0.11	33	1.98	0.29	90
9.59	1.16	90	8.71	1.16	90
1.54	0.47	147	2.09	0.29	90
4.75	0.95	296	2.29	0.33	103
1.29	0.47	590	0.57	0.08	103
3.95	0.95	1182	0.69	0.10	128
4.03	0.74	229			
Total :			Total Aux. Heads		
98	16.5	7235	97	12.8	2490

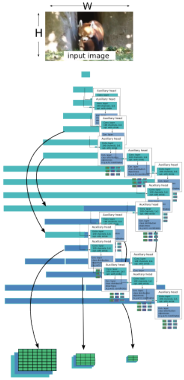
- Size and complexity augmented by 2 for yolov5s
- Gating mechanism or supsampling of the output is required to be efficient

Auxiliary head implementation



Main branch time(ms)	GFLOPs	#params (10^3)	Aux. heads time(ms)	GFLOPs	#params ( )
7.83	0.72	3	9.79	1.03	80
6.50	0.95	18	8.96	1.07	83
15.20	0.96	18	8.76	1.07	83
4.15	0.95	73	8.65	1.16	90
10.30	1.48	115	8.57	1.16	90
3.00	0.95	295	2.05	0.33	103
8.51	2.00	625	2.50	0.33	103
2.79	0.94	1180	0.62	0.10	128
4.01	0.95	1182	0.68	0.10	128
3.75	0.53	656	0.71	0.10	128
0.45	0.11	131	0.58	0.08	103
5.81	1.16	361	2.57	0.33	103
0.56	0.11	33	1.98	0.29	90
9.59	1.16	90	8.71	1.16	90
1.54	0.47	147	2.09	0.29	90
4.75	0.95	296	2.29	0.33	103
1.29	0.47	590	0.57	0.08	103
3.95	0.95	1182	0.69	0.10	128
4.03	0.74	229			
Total :			Total Aux. Heads		
98	16.5	7235	97	12.8	2490

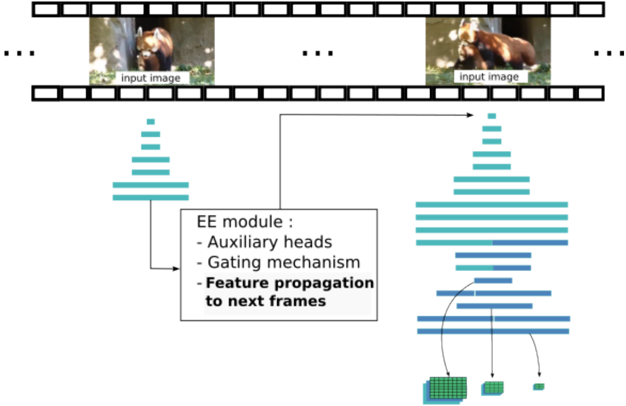
- Size and complexity of the full model is rougly augmented by 2
- Gating mechanism or supsampling of the output is required to be efficient



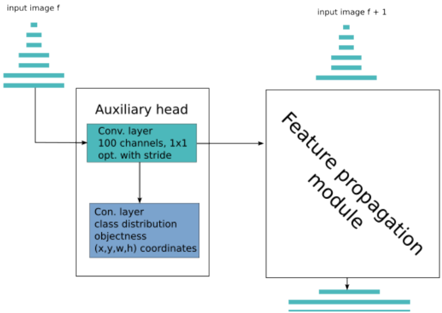
Main branch			Aux. heads		
time(ms)	GFLOPs	#params (10^3)	time(ms)	GFLOPs	#params (10^3)
7.83	0.72	3	9.79	1.03	80
6.50	0.95	18	8.96	1.07	83
15.20	0.96	18	8.76	1.07	83
4.15	0.95	73	8.65	1.16	90
10.30	1.48	115	8.57	1.16	90
3.00	0.95	295	2.05	0.33	103
8.51	2.00	625	2.50	0.33	103
2.79	0.94	1180	0.62	0.10	128
4.01	0.95	1182	0.68	0.10	128
3.75	0.53	656	0.71	0.10	128
0.45	0.11	131	0.58	0.08	103
5.81	1.16	361	2.57	0.33	103
0.56	0.11	33	1.98	0.29	90
9.59	1.16	90	8.71	1.16	90
1.54	0.47	147	2.09	0.29	90
4.75	0.95	296	2.29	0.33	103
1.29	0.47	590	0.57	0.08	103
3.95	0.95	1182	0.69	0.10	128
4.03	0.74	229			
Total :			Total Aux. Heads		
98	16.5	7235	97	12.8	2490

- Size and complexity of the full model is rougly augmented by 2
- Gating mechanism or supsampling of the output is required to be efficient

Early Exit for efficient video processing

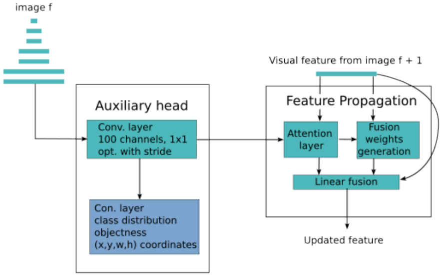


Feature propagation



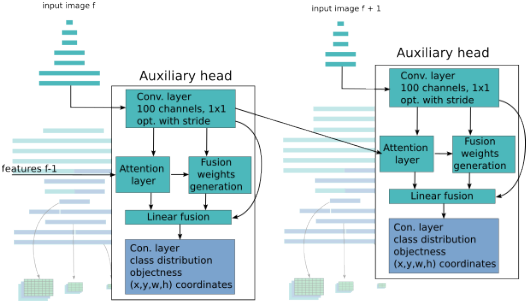
- Goal : include visual features computed from previous image
- Deal with common tracking issue : alignment, appearance variation,...

Feature propagation



- Align previous feature with attention mechanism
- Learned weights to linearly combine aligned and current features

Feature Propagation



- Inclusion of the feature propagation into the Auxiliary head
- Recursive feature update

Let f^{t-1}, f^t, 100 dim feature set for frames $t-1$ and t

$$f_{align}^{t-1} = attention(f^{t-1}, f^t)$$

f_{align}^{t-1} are aligned features computed with attention where current features f^t are the queries and f^{t-1} the values.

$$w = sigmoid(conv([f_{align}^{t-1}, f^t]))$$

The final features are computed as :

$$f_{final}^t = w \times f_{align}^{t-1} + (1-w) \times f^t$$

Note: Some resizing are also required.

Cost of the Feature propagation

- Number of parameters due to modelling choice
  - Fixed 100 dimension and resizing to 20 x 20
  - $\approx$ 90K additional parameters
- Memory cost depends on the grid size
  - Attention matrix is $(20 \times 20) \times (w \times h)$
  - where w,h $\in [20, 40, 80]$
- Around 40% of flops added for each head.

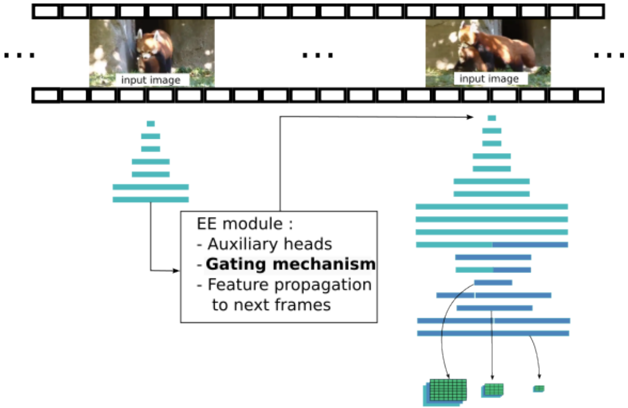
How powerfull is this module ?

- The previous features are selected by attention mechanism
  - i.e. a dot product, in other words, linear correlation (+ some layers)
  - Thus, it will probably learn to reinforce similar features between the two frames, or disminish different ones.
- Some positional and time delay embeddings could be added
  - In order to trust a nearby pixel in a nearby frame.

- The training of the feature propagation module might interfere with other parts of the model
- It is likely that you will start from a pre-trained model
- Train the feature propagation while freezing the rest of the network
- Then train the whole network.

GPUs are currently running to evaluate the method on imagenetvid...

When to exit ?


input image
input image
EE module :
- Auxiliary heads
- **Gating mechanism**
- Feature propagation
to next frames

method

- Non homogenous and non differentiable cost function
  - Classification loss + computing cost loss
- Related work with reinforcement learning and **Learning halting scores**
- Our proposal : learning the difficulty of a task from the behavior of the feature along the network
- A training dataset can be build from the output of our model
- No implementation done yet.

How much gain can we hope?

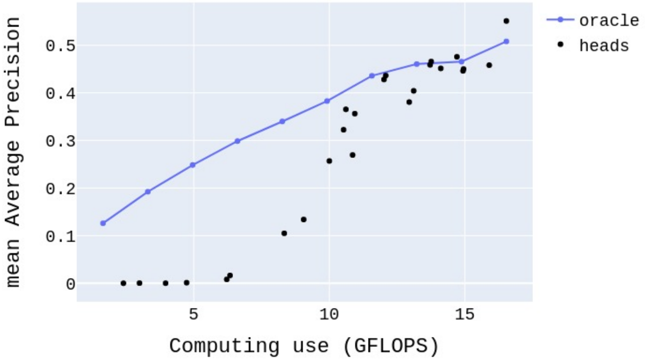
Construction of an oracle : best accuracy for a given computation budget

- Not trivial for the mAP metric
  - Need to recompute the ROC curve for each prediction change
- Simplifying assumption :

  the sum of the mAP for each image $\approx$ the overall mAP
- Linar programming problem :

  Maximize the sum of the mAP given a computation budget

Oracle Results



- Gains can be due to the choice of NO DETECTION or selecting a suitable auxiliary head.

Perspectives

- Evaluation the feature fusion module
- Experiment of the Gating mechanism
- Expressivity of the feature propagation module
  - Time gap and positional embedding
- Better training of the Auxiliary heads with Distillation

Conclusion

Around 172.032 Kw/h have been used in this work so far

This is around 8.6 Kgs of CO_2 equivalents

Thank you for your attention

References I

[Bakhtiarnia et al., 2021] Bakhtiarnia, A., Zhang, Q., and Iosifidis, A. (2021).
Multi-exit vision transformer for dynamic inference.
arXiv preprint arXiv:2106.15183.

[Bolukbasi et al., 2017] Bolukbasi, T., Wang, J., Dekel, O., and Saligrama, V. (2017).
Adaptive neural networks for fast test-time prediction.
arXiv preprint arXiv:1702.07811, 1(3).

[Feichtenhofer et al., 2019] Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019).
Slowfast networks for video recognition.
In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211.

References II

[Feichtenhofer et al., 2017] Feichtenhofer, C., Pinz, A., and Zisserman, A. (2017).
Detect to track and track to detect.
In *Proceedings of the IEEE international conference on computer vision*, pages 3038–3046.

[Figurnov et al., 2017] Figurnov, M., Collins, M. D., Zhu, Y., Zhang, L., Huang, J., Vetrov, D., and Salakhutdinov, R. (2017).
Spatially adaptive computation time for residual networks.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1039–1048.

References III

[Guan et al., 2017] Guan, J., Liu, Y., Liu, Q., and Peng, J. (2017).
Energy-efficient amortized inference with cascaded deep classifiers.
arXiv preprint arXiv:1710.03368.

[Guo et al., 2019] Guo, C., Fan, B., Gu, J., Zhang, Q., Xiang, S., Prinet, V., and Pan, C. (2019).
Progressive sparse local attention for video object detection.
In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3909–3918.

References IV

[Hu et al., 2020] Hu, T.-K., Chen, T., Wang, H., and Wang, Z. (2020).
Triple wins: Boosting accuracy, robustness and efficiency together by enabling input-adaptive inference.
arXiv preprint arXiv:2002.10025.

[Huang et al., 2017] Huang, G., Chen, D., Li, T., Wu, F., Van Der Maaten, L., and Weinberger, K. Q. (2017).
Multi-scale dense networks for resource efficient image classification.
arXiv preprint arXiv:1703.09844.

[Jiang et al., 2020] Jiang, Z., Liu, Y., Yang, C., Liu, J., Gao, P., Zhang, Q., Xiang, S., and Pan, C. (2020).
Learning where to focus for efficient video object detection.
In *European conference on computer vision*, pages 18–34. Springer.

[Kaya et al., 2019] Kaya, Y., Hong, S., and Dumitras, T. (2019).
Shallow-deep networks: Understanding and mitigating network overthinking.
In *International conference on machine learning*, pages 3301–3310. PMLR.

[Laskaridis et al., 2020] Laskaridis, S., Venieris, S. I., Almeida, M., Leontiadis, I., and Lane, N. D. (2020).
Spinn: synergistic progressive inference of neural networks over device and cloud.
In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–15.

[Leontiadis et al., 2021] Leontiadis, I., Laskaridis, S., Venieris, S. I., and Lane, N. D. (2021).
It's always personal: Using early exits for efficient on-device cnn personalisation.
In *Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications*, pages 15–21.

References VII

[Lin et al., 2022] Lin, S., Ji, B., Ji, R., and Yao, A. (2022).
A closer look at branch classifiers of multi-exit architectures.
arXiv preprint arXiv:2204.13347.

[Liu and Zhu, 2018] Liu, M. and Zhu, M. (2018).
Mobile video object detection with temporally-aware feature maps.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5686–5695.

[Lu et al., 2020] Lu, Z., Rathod, V., Votel, R., and Huang, J. (2020).
Retinatrack: Online single stage joint detection and tracking.
In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14668–14678.

[Qendro et al., 2021] Qendro, L., Campbell, A., Lio, P., and Mascolo, C. (2021).
Early exit ensembles for uncertainty quantification.
In *Machine Learning for Health*, pages 181–195. PMLR.

[Sabet et al., 2021] Sabet, A., Hare, J., Al-Hashimi, B., and Merrett, G. V. (2021).
Temporal early exits for efficient video object detection.
arXiv preprint arXiv:2106.11208.

[Veit and Belongie, 2018] Veit, A. and Belongie, S. (2018).
Convolutional networks with adaptive inference graphs.
In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–18.

[Wang et al., 2021] Wang, X., Huang, Z., Liao, B., Huang, L., Gong, Y., and Huang, C. (2021).
Real-time and accurate object detection in compressed video by long short-term feature aggregation.
Computer Vision and Image Understanding, 206:103188.

[Wang et al., 2020] Wang, X., Kondratyuk, D., Christiansen, E., Kitani, K. M., Alon, Y., and Eban, E. (2020).
Wisdom of committees: An overlooked approach to faster and more accurate models.
arXiv preprint arXiv:2012.01988.

[Wang et al., 2018] Wang, X., Yu, F., Dou, Z.-Y., Darrell, T., and Gonzalez, J. E. (2018).
Skipnet: Learning dynamic routing in convolutional networks.
In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 409–424.

[Wu et al., 2020] Wu, W., He, D., Tan, X., Chen, S., Yang, Y., and Wen, S. (2020).
Dynamic inference: A new approach toward efficient video action recognition.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 676–677.

[Wu et al., 2019] Wu, Z., Xiong, C., Jiang, Y.-G., and Davis, L. S. (2019).
Liteeval: A coarse-to-fine framework for resource efficient video recognition.
Advances in neural information processing systems, 32.

[Zhu et al., 2018] Zhu, X., Dai, J., Yuan, L., and Wei, Y. (2018).
Towards high performance video object detection.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7210–7218.