Spatially Adaptive Computation Time for Residual Networks

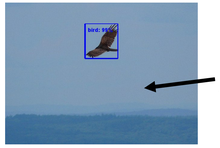
Michael Figurnov[1] Maxwell D. Collins[2] Yukun Zhu[2] Li Zhang[2] Jonathan Huang[2] Dmitry Vetrov[1,3] Ruslan Salakhutdinov[4]

[1]National Research University Higher School of Economics [2]Google Inc. [3]Yandex [4]Carnegie Mellon University

IEEE 2017 Conference on Computer Vision and Pattern Recognition

CVPR July 21-26 2017 HONOLULU

Motivation



Do we really need to put these blue pixels through 100 layers to detect the bird?

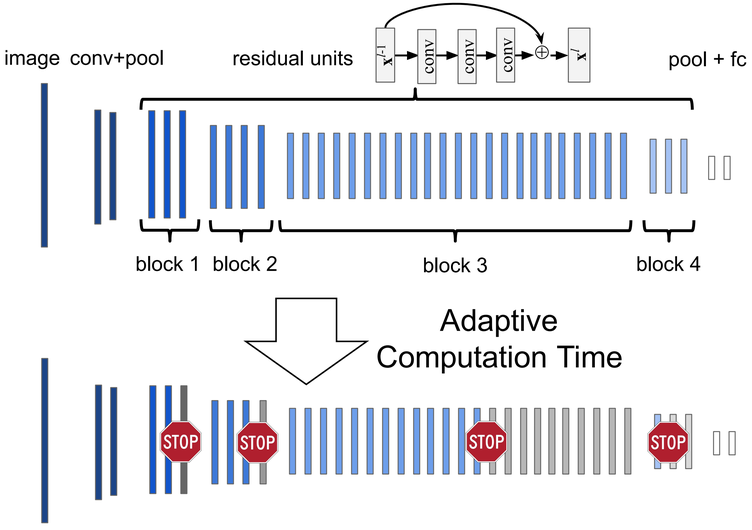
Can we make networks **faster** and **more interpretable** by adapting the amount of computation?

Contribution

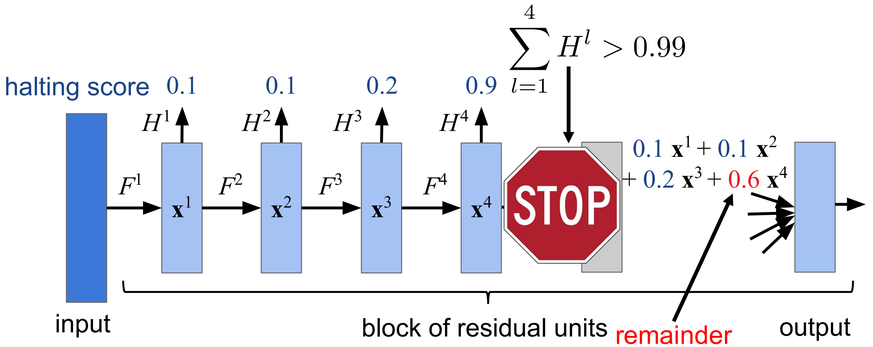
A novel mechanism that

- adapts the computation **spatially**
- provides introspection
- is problem-agnostic
- is end-to-end trainable
- scales to ImageNet and COCO

Residual Network (ResNet)



Adaptive Computation Time (ACT)

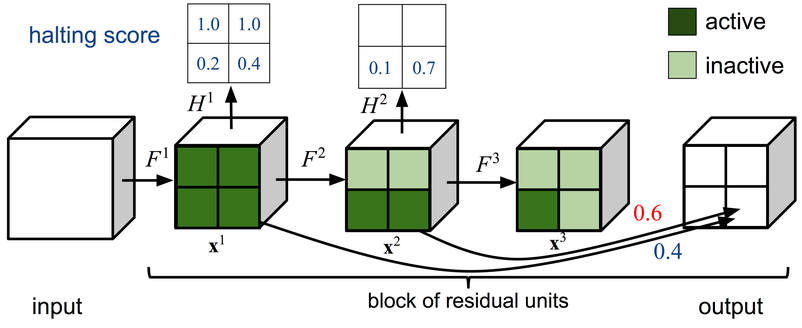


Loss function $\mathcal{L}' = \mathcal{L} + \tau(N + R)$, $R = 1 - \sum_{l=1}^{N-1} H^l$

original loss; number of residual units; ponder cost (close to N, but differentiable)

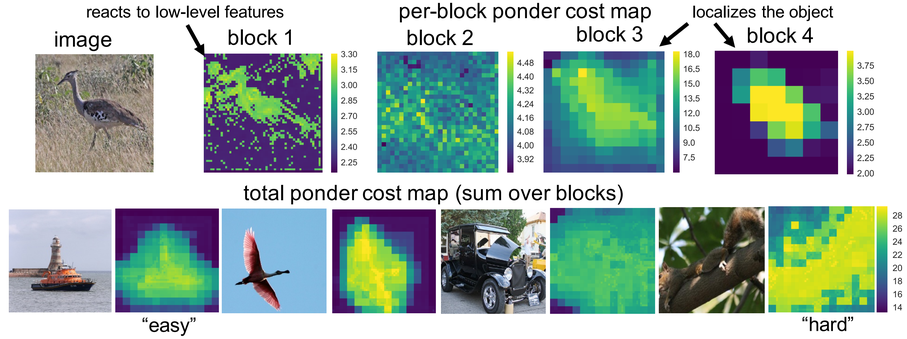
Spatially Adaptive Computation Time (SACT)

Apply ACT "convolutionally" to every spatial position of residual network's block

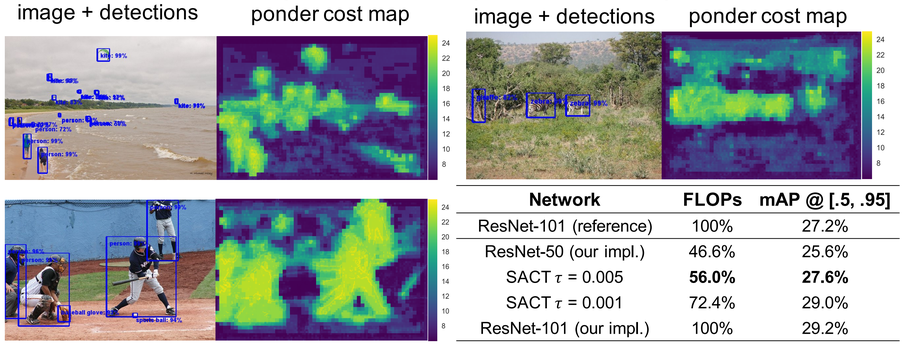


Residual units are evaluated only in the active positions!

Image classification (ImageNet dataset)



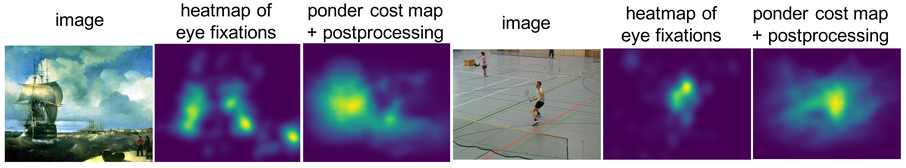
Object detection (COCO dataset)



Network	FLOPs	mAP @ [.5, .95]
ResNet-101 (reference)	100%	27.2%
ResNet-50 (our impl.)	46.6%	25.6%
SACT τ = 0.005	**56.0%**	**27.6%**
SACT τ = 0.001	72.4%	29.0%
ResNet-101 (our impl.)	100%	29.2%

SACT has better speed-mAP trade-off compared to reducing the depth of ResNet

Visual saliency (cat2000 dataset)



Ponder cost maps tend to highlight salient objects, even though the SACT model is not trained on cat2000!

Code: github.com/mfigurnov/sact