Spatially Adaptive Computation Time for Residual Networks

Michael Figurnov, Maxwell D. Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, Ruslan Salakhutdinov

Denis Belyakov, 152

Neural Nets

- Object detection
- Image segmentation
- Image-to-text
- Image generation
- NLP

- RL (yeah, Go and etc.)
- any other field you can think about…

Neural Nets

Problem?

Computation cost!

- Object detection
- Image segmentation
- Image-to-text
- Image generation
- NLP

- RL (yeah, Go and etc.)
- any other field you can think about…

Solution

Glimpse-based attention models:

- Processing small number of rectangular subregions

RNN, RL *and Blackjack*

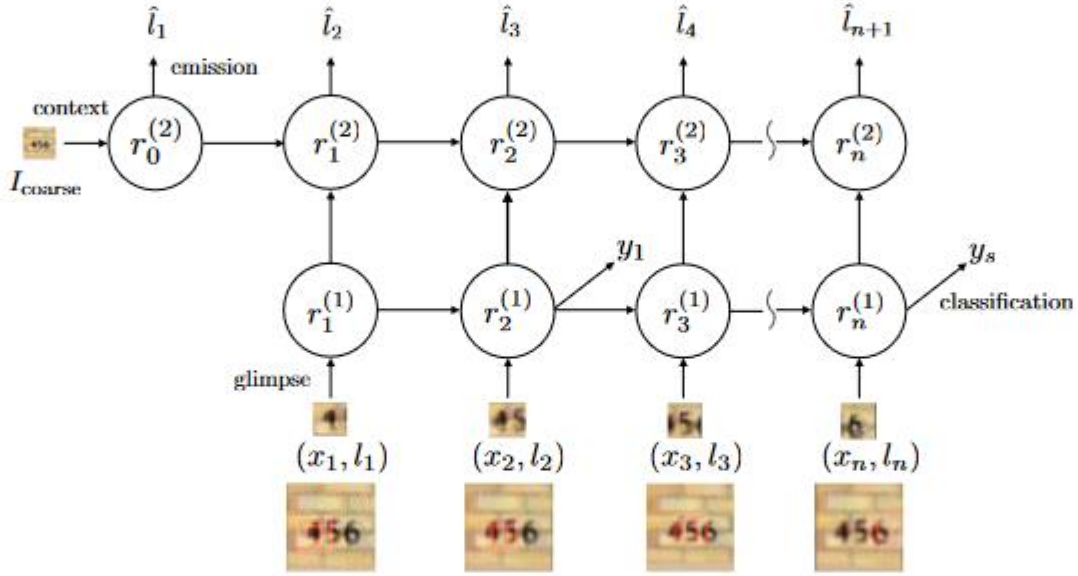


Figure 1: The deep recurrent attention model.

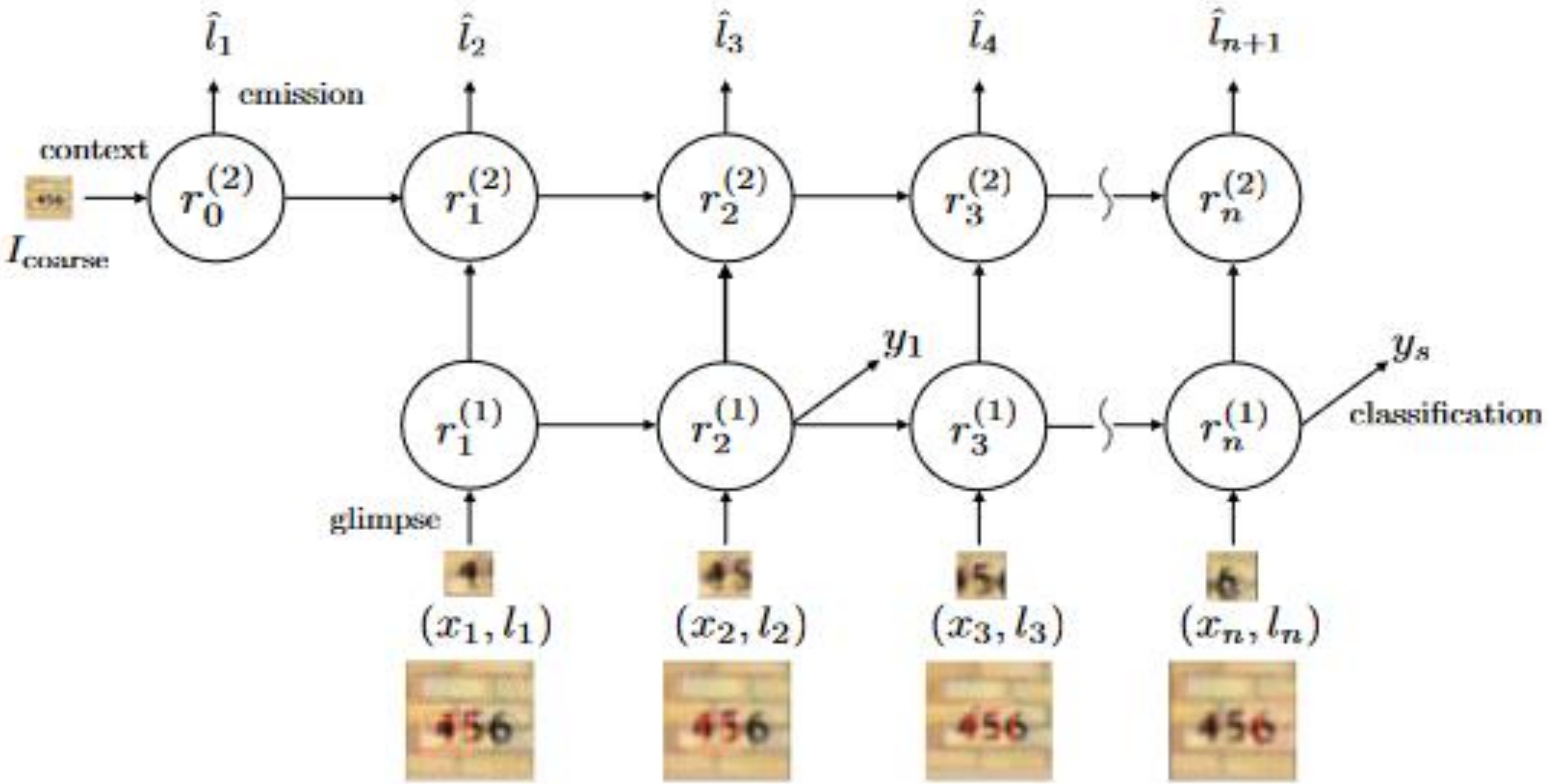


Figure 1: The deep recurrent attention model.

Solution

Glimpse-based attention models:

- Processing small number of rectangular subregions

Q: Are we happy?

A: Not yet…

- Not suitable for segmentation, image generation
- Requires heuristics or separate nn

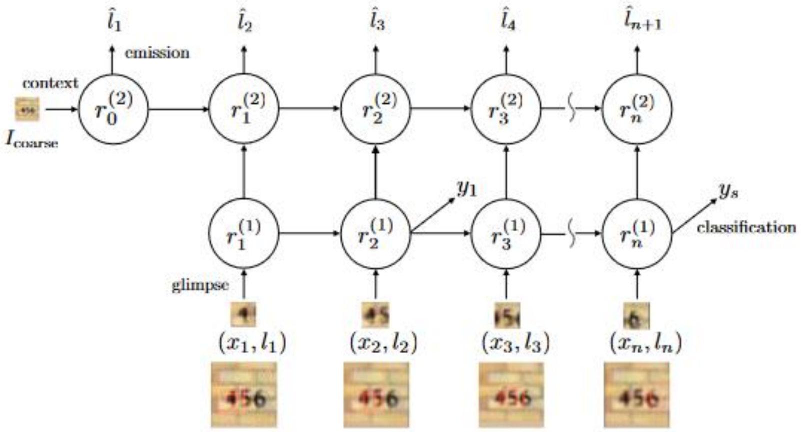
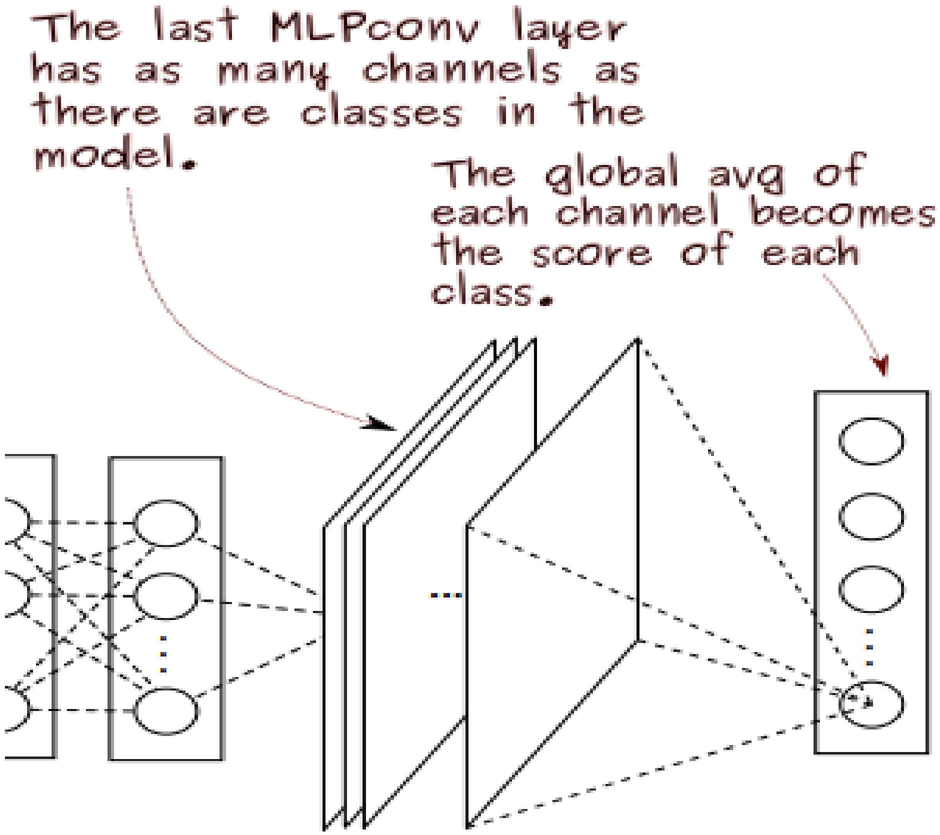


Figure 1: The deep recurrent attention model.

Global average pooling



From Network in Network, 2013

Global average pooling sums out the spatial information, thus it is more robust to spatial translations of the input.

SACT (Spatially Adaptive Computation Time)

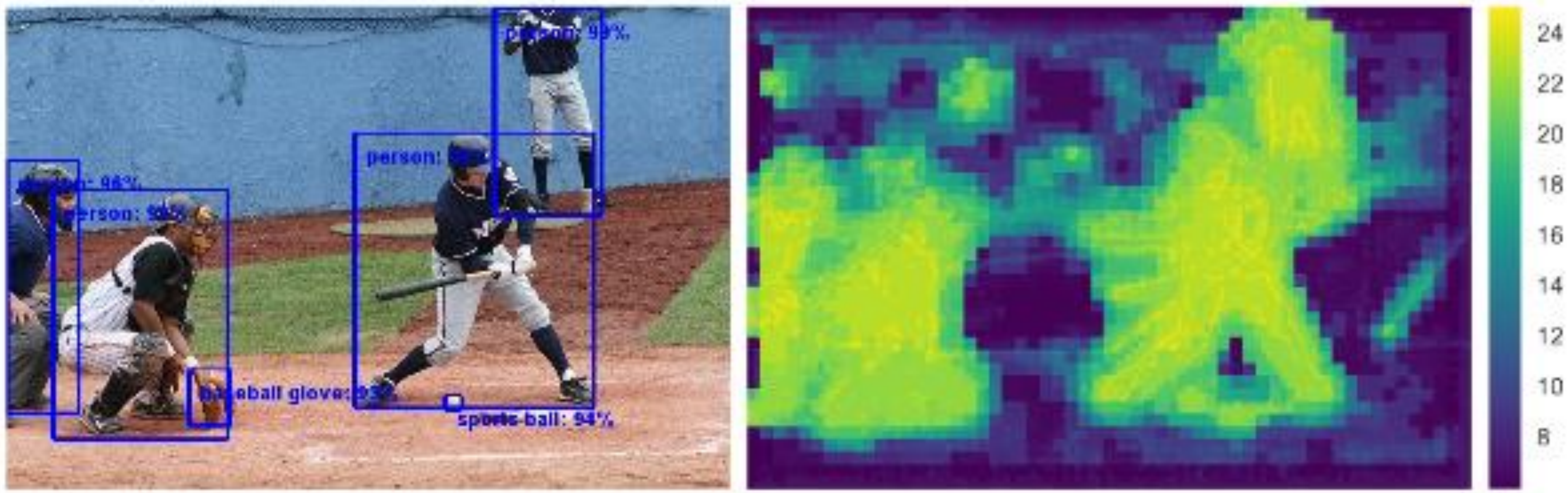
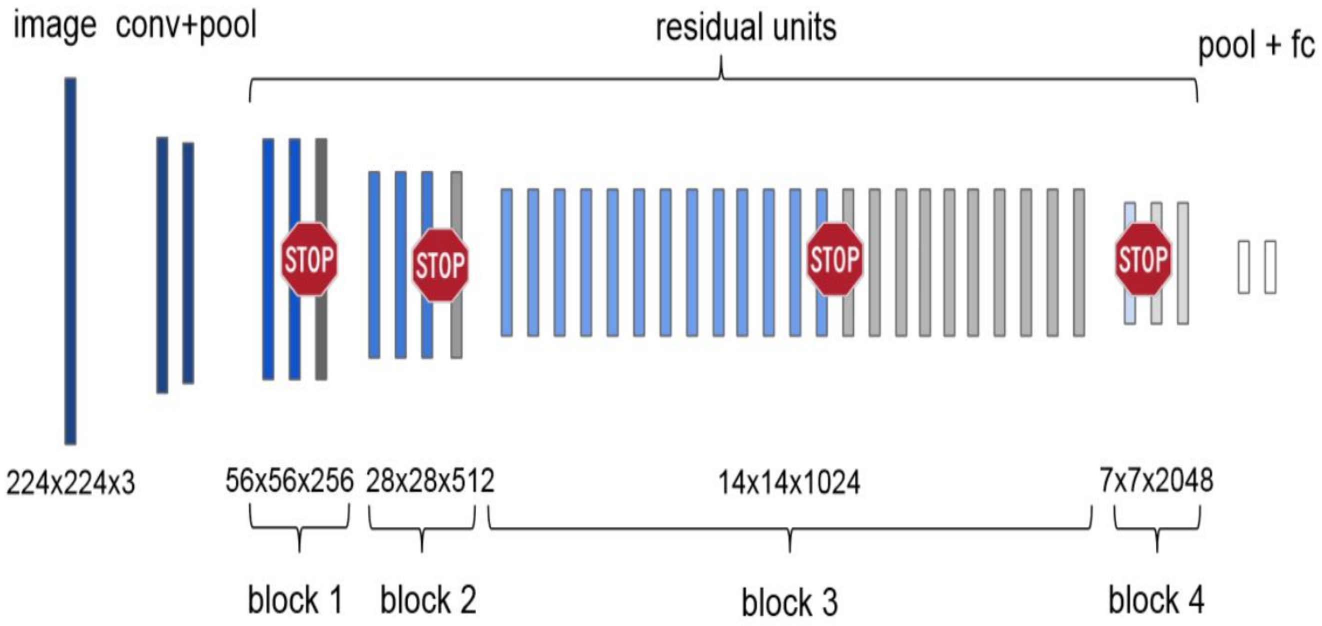


Figure 1: Left: object detections. Right: feature extractor SACT ponder cost (computation time) map for a COCO validation image. The proposed method learns to allocate more computation for the object-like regions of the image.

Stage one: ResNet

- ResNet101
- 1st : Conv + maxpool
  - Stride 4
- 4 residual blocks
  - 3, 4, 23, 3 residual units
- Conv with stride at each block's start
  - Doubles number of channels
- Global AvgPool at the end
  - Followed by dense layers



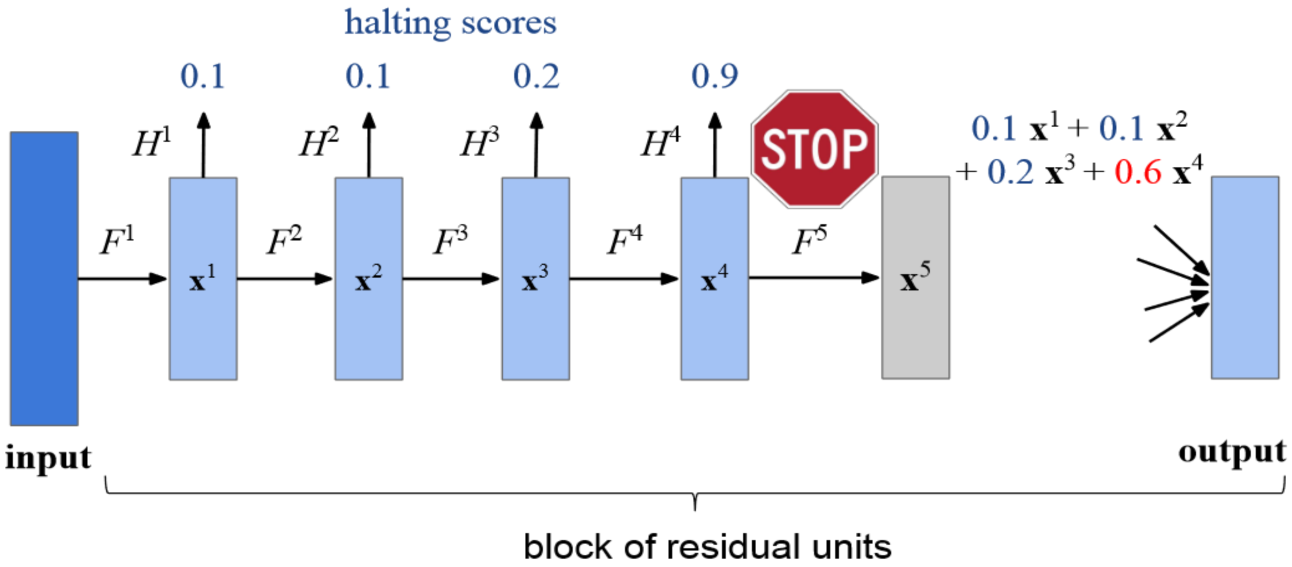
ACT two: informally

Q: What is the purpose Adaptive Computation Time?

A: Improve the computational efficiency

Key concepts:

- Halting score
- Remainder
- Ponder cost = $\text{Num}_{\text{evaluated}}$ + remainder



Minimizing the ponder cost increases halting scores of non-last units

ACT going deeper

- Block of L residual units (**tensors** H x W x Channels)

$$\mathbf{x}^0 = \mathbf{input},$$
$$\mathbf{x}^l = F^l(\mathbf{x}^{l-1}) = \mathbf{x}^{l-1} + f^l(\mathbf{x}^{l-1}),\ l = 1 \dots L,$$
$$\mathbf{output} = \mathbf{x}^L.$$

- Halting score for each residual unit

$$h^l = H^l(\mathbf{x}^l),\ l = 1 \dots (L-1),$$
$$h^L = 1.$$

$$h^l = H^l(\mathbf{x}^l) = \sigma(W^l \operatorname{pool}(\mathbf{x}^l) + b^l)$$

- N – number of residual units to evaluate

$$N = \min\left\{ n \in \{1 \dots L\} : \sum_{l=1}^{n} h^l \geq 1 - \varepsilon \right\}$$

- R – remainder, p^l – halting dist.

$$R = 1 - \sum_{l=1}^{N-1} h^l$$

$$p^l = \begin{cases} h^l & \text{if } l < N, \\ R & \text{if } l = N, \\ 0 & \text{if } l > N. \end{cases}$$

ACT going deeper

$$\textbf{output} = \sum_{l=1}^{L} p^l \mathbf{x}^l = \sum_{l=1}^{N} p^l \mathbf{x}^l .$$

$$\rho = N + R.$$

- As we cannot optimize N directly, we introduce the ***ponder cost*** and we ignore the gradient of N.

$$\frac{\partial \rho}{\partial h^l} = \begin{cases} -1 & \text{if } l < N, \\ 0 & \text{if } l \geq N. \end{cases}$$

- We apply ACT to each block independently and stack them.

- Loss function with added ***ponder cost*** opt.

$$\mathcal{L}' = \mathcal{L} + \tau \sum_{k=1}^{K} \rho_k .$$

ACT advantages

- Calculate blocks' outputs *"on the fly"*
- Adds very few params to base model
- ACT is a generalization of ResNet

Algorithm 1 Adaptive Computation Time for one block of residual units. ACT does not require storing the intermediate residual units outputs.

Input: 3D tensor **input**
Input: number of residual units in the block L
Input: $0 < \varepsilon < 1$
Output: 3D tensor **output**
Output: ponder cost ρ

$\mathbf{x} = \mathbf{input}$
$c = 0$ ▷ Cumulative halting score
$R = 1$ ▷ Remainder value
$\mathbf{output} = 0$ ▷ Output of the block
$\rho = 0$
for $l = 1 \ldots L$ **do**
    $\mathbf{x} = F^l(\mathbf{x})$
    if $l < L$ **then** $h = H^l(\mathbf{x})$
    else $h = 1$
    end if
    $c \mathrel{+}= h$
    $\rho \mathrel{+}= 1$
    if $c < 1 - \varepsilon$ **then**
        $\mathbf{output} \mathrel{+}= h \cdot \mathbf{x}$
        $R \mathrel{-}= h$
    else
        $\mathbf{output} \mathrel{+}= R \cdot \mathbf{x}$
        $\rho \mathrel{+}= R$
        break
    end if
end for
return output, ρ

We can ACT better : SACT

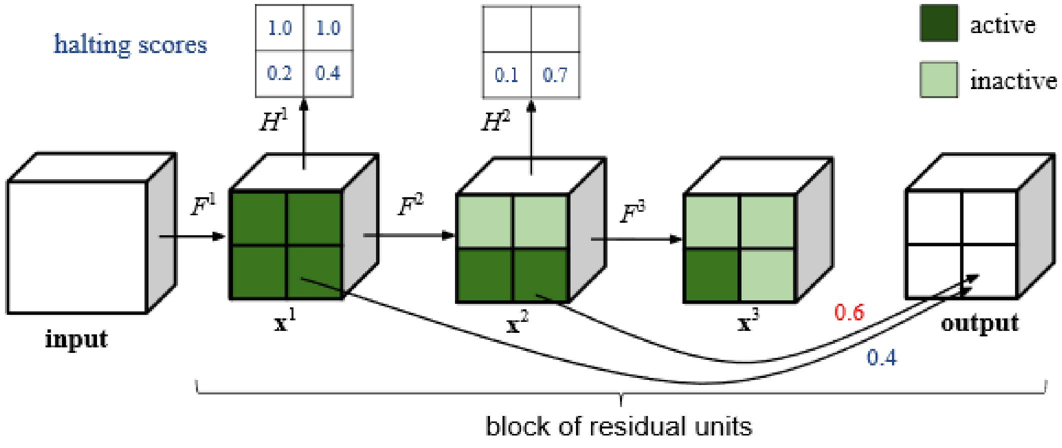
- Per position computation by applying ACT to each spatial position of the block.

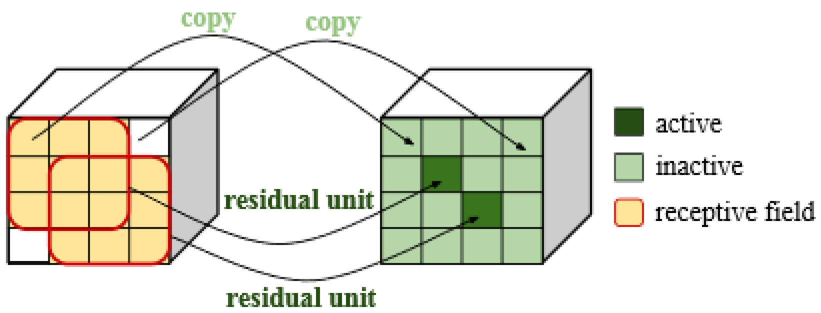
- ***Active positions*** – spatial locations where cum. halt score is less than 1.

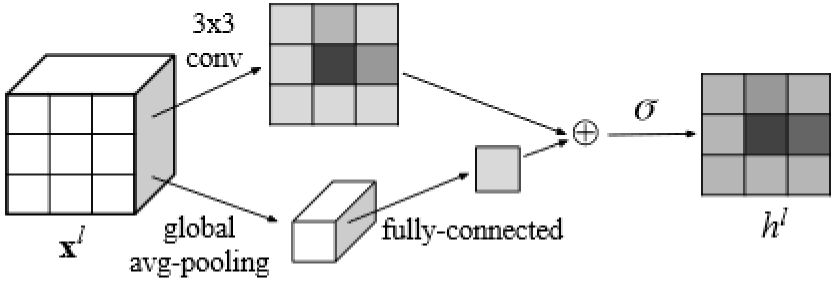
$$H^l(\mathbf{x}) = \sigma(\widetilde{W}^l * \mathbf{x} + W^l \operatorname{pool}(\mathbf{x}) + b^l)$$

- ***** is 3 x 3 convolution

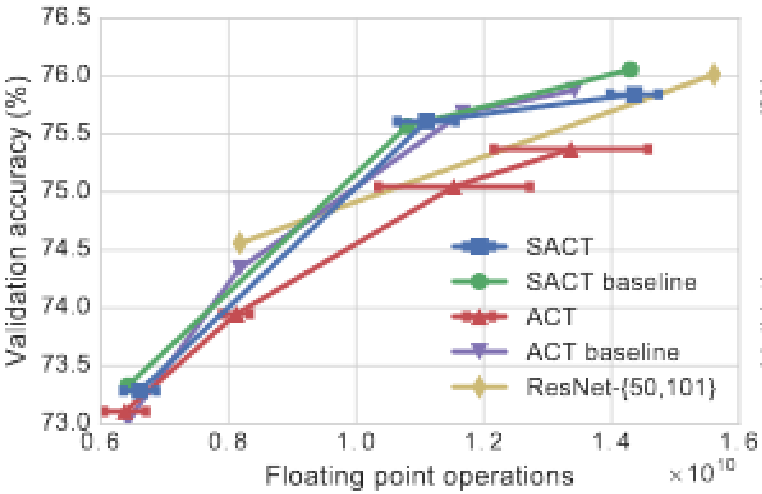
  Perforated conv. Layer:
  - Zeros instead of neighbors for skipped values
  - Tile the halting scores map



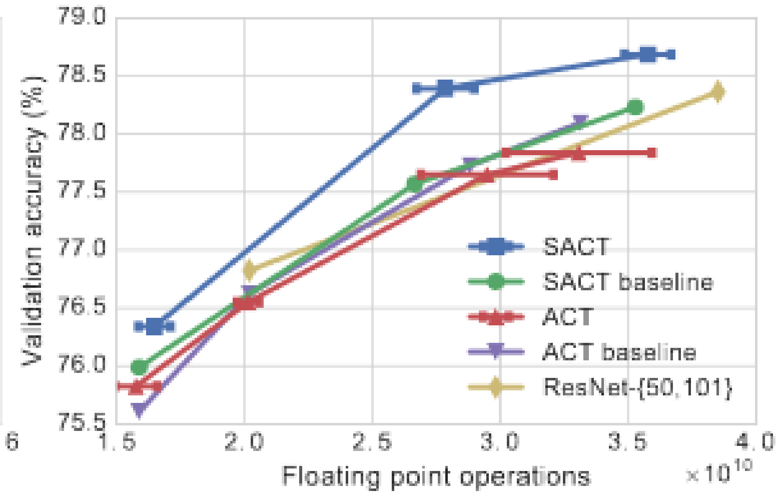




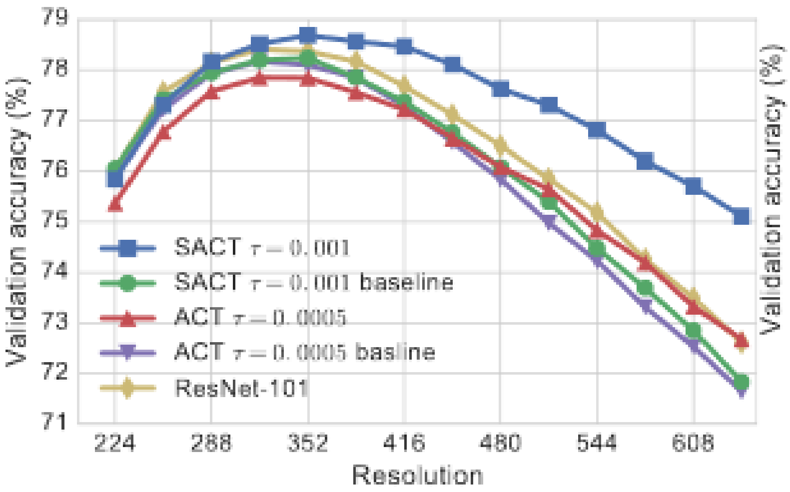
Results on ImageNet



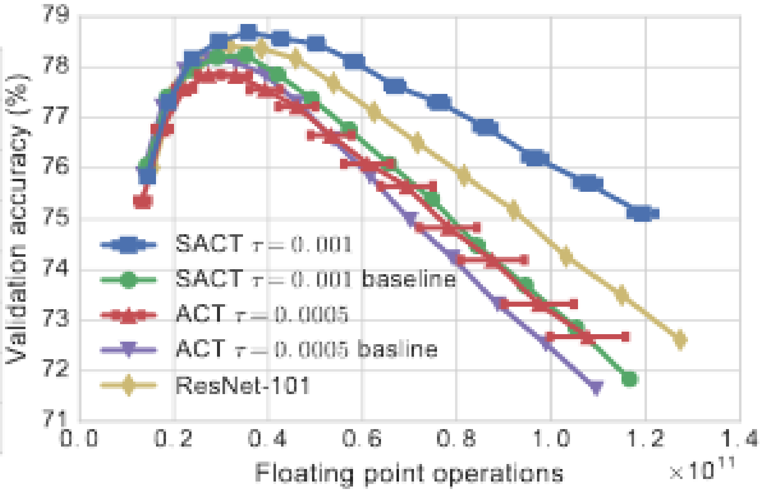
(a) Test resolution 224 × 224



(b) Test resolution 352 × 352



(c) Resolution *vs.* accuracy



(d) FLOPs *vs.* accuracy for varying resolution

Ponder costs visualization

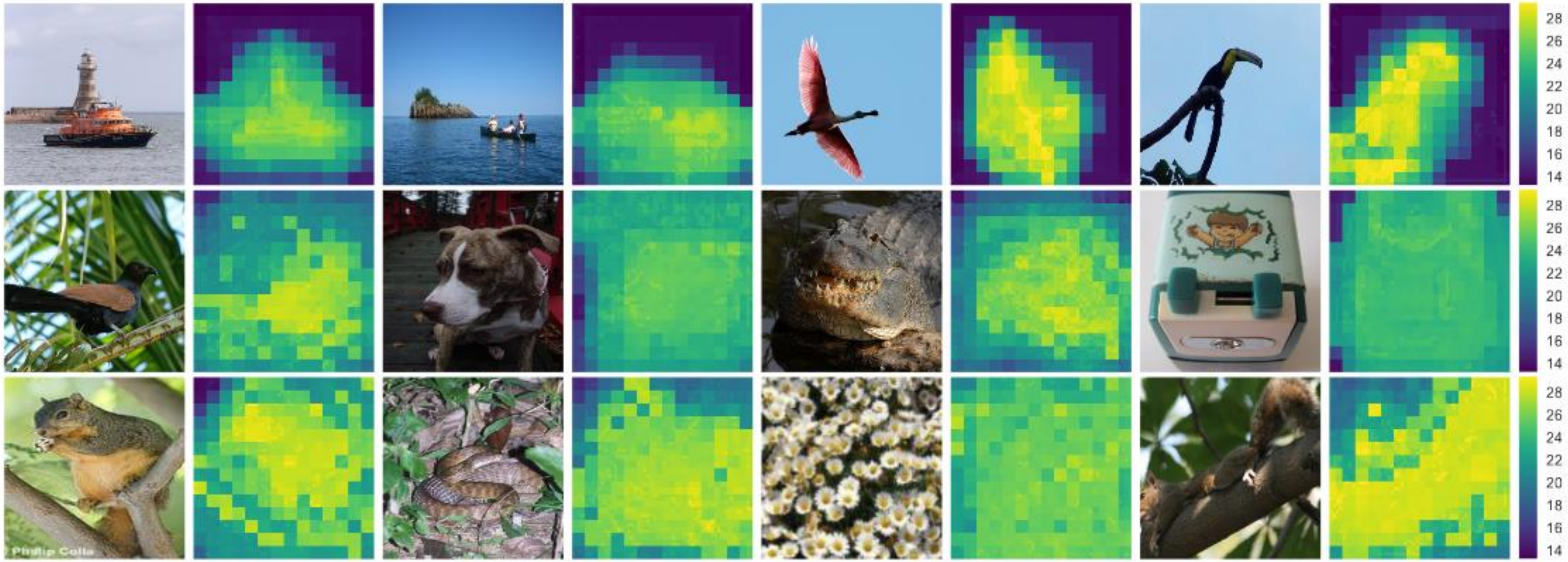


Figure 9: ImageNet validation set. SACT ($\tau = 0.005$) ponder cost maps. Top: low ponder cost (19.8-20.55), middle: average ponder cost (23.4-23.6), bottom: high ponder cost (24.9-26.0). SACT typically focuses the computation on the region of interest.

Object detection (COCO)

- Faster R-CNN pipeline:
  - Feature extractor
  - Region Proposal Net predicts rect. proposals
  - Box classifier
- Use 1-3 ResNet blocks as extractor, 4 – box clf
- Reuse pretrained models

Feature extractor	FLOPs (%)	mAP @ [.5, .95] (%)
ResNet-101 [16]	100	27.2
ResNet-50 (our impl.)	46.6	25.56
SACT $\tau = 0.005$	$\mathbf{56.0 \pm 8.5}$	**27.61**
SACT $\tau = 0.001$	72.4 ± 8.4	29.04
ResNet-101 (our impl.)	100	29.24

Table 1: COCO val set. Faster R-CNN with SACT results. FLOPs are average ($\pm$ one standard deviation) feature extractor floating point operations relative to ResNet-101 (that does 1.42E+11 operations). SACT improves the FLOPs-mAP trade-off compared to using ResNet without adaptive computation.

Like humans

Cat2000 dataset – human eye fixations
Reuse previous SACT models
Resize 1920 x 1080 to 320 x 180 and 640 x 360 respectively

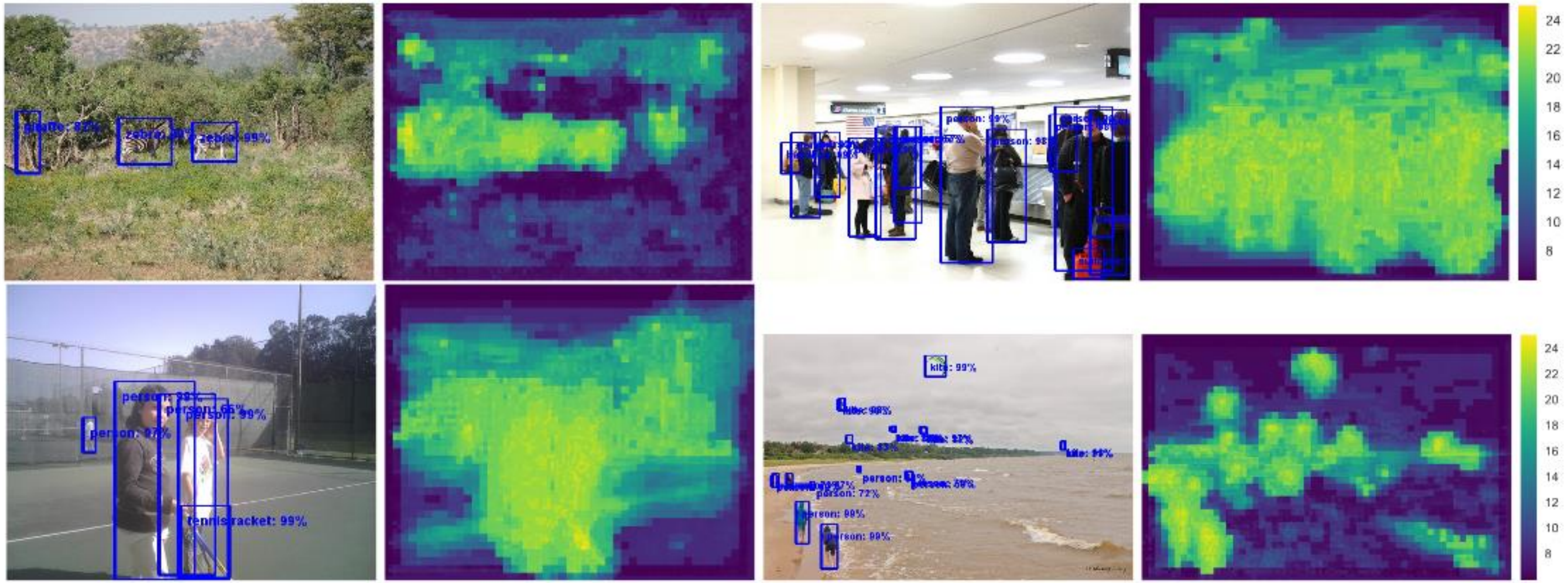


Figure 10: COCO testdev set. Detections and feature extractor ponder cost maps ($\tau = 0.005$). SACT allocates much more computation to the object-like regions of the image.

Like humans

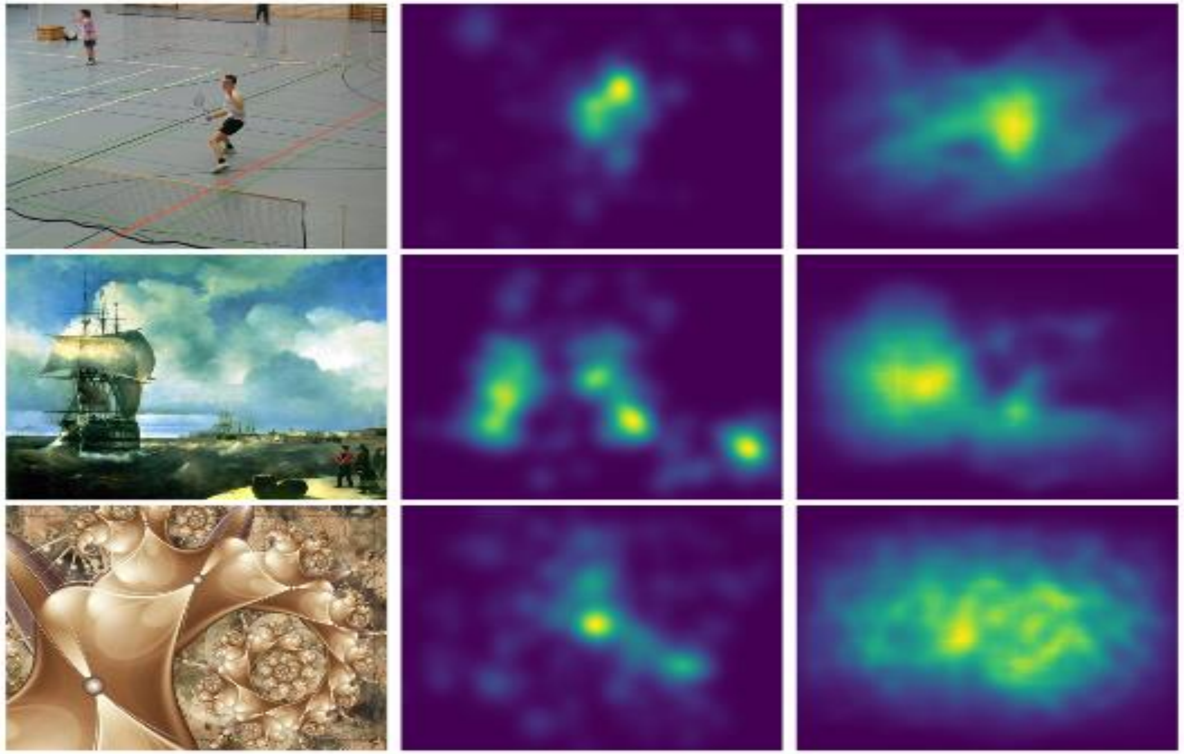


Figure 11: cat2000 saliency dataset. Left to right: image, human saliency, SACT ponder cost map (COCO model, $\tau = 0.005$) with postprocessing (see text) and softmax with temperature $1/5$. Note the center bias of the dataset. SACT model performs surprisingly well on out-of-domain images such as art and fractals.