Compressing the Input for CNN with the First Order Scattering Transform

Edouard Oyallon[1,2], *Eugene Belilovsky*[3], *Sergey Zagoruyko*[2], *Michal Valko*[2]

(1) CentraleSupélec (2) INRIA (3) Mila

Reducing the input size

- CNNs for images are typically fed with large images that have some redundant structures. Can we exploit this for reducing the input size?

$N{\times}N \longrightarrow$ *CNN*

Input

$n{\times}n{\times}k \longrightarrow$ *CNN*

Input can be "small"?

$N{\times}N \xrightarrow{?} n{\times}n{\times}k$

- We propose to introduce a representation which:
  - Reduces the spatial resolution **and** dimensionality
  - Preserves the input **and** is predefined, for natural images

Gabor wavelets and modulus

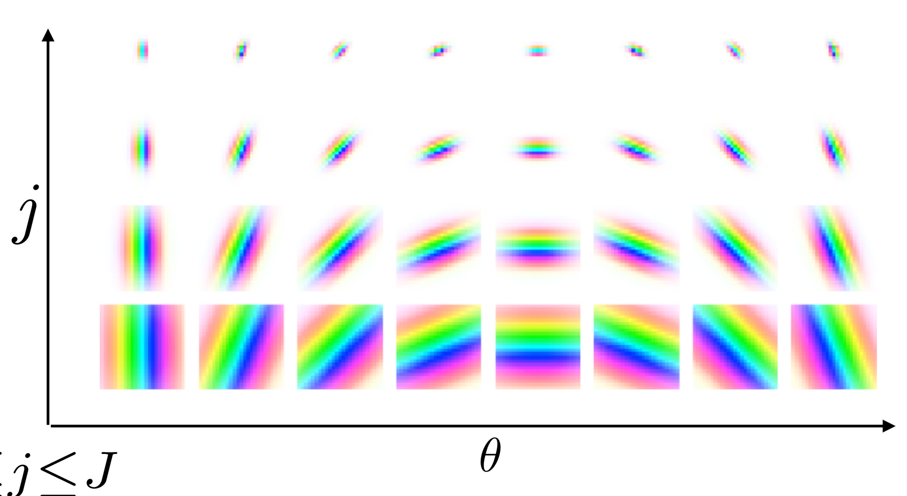
- We consider the Gabor wavelets, that have a good trade-off between space and frequency localization.

$$\psi_{j,\theta}(u) = \frac{1}{2^{2J}}\psi(r_{-\theta}\frac{u}{2^J})$$

$$\phi_J(u) = \frac{1}{2^{2J}}\phi(\frac{u}{2^J})$$

$$A_J x = x \star \phi_J$$

$$Wx = \{x \star \psi_{j,\theta}\}_{\theta\in\Theta, 0\leq j\leq J}$$



- We observe that a translation x_a of x by a leads to a phase multiplication:

$$x_a \star \psi(u) \approx e^{i\omega_0^T a} x \star \psi(u)$$

- The enveloppe is more invariant to translations: ideal for A_J.

First order Scattering Transform

$$x \longrightarrow W \longrightarrow |.| \longrightarrow A_J \longrightarrow$$

- The first order scattering is the succession of a wavelet transform, a point modulus and a spatial averaging.

$$Sx = \{|x \star \psi_{j,\theta}| \star \phi_J, x \star \phi_J\}_{\theta,j}$$

- It is similar to a SIFT with appropriate wavelets.

- It compresses the input image:

$$\frac{\#Sx}{\#x} = \frac{(1+\#\Theta J)}{2^J}$$

J	1	2	3	4
Compression ratio	2,2	1,1	0,39	0,13

Information preservation

- We propose a simple algorithm for reconstructing order via MSE minimization:

$$x = \arg\inf_{\tilde{x}} \|S\tilde{x} - Sy\|$$

- We observe that the first order Scattering does not lead to a significant loss when **reconstructing**:

Original | $J = 3$ | $J = 4$

- We empirically observe that the loss of image details is due to the windowed averaging.

Classification performances

Resnet

$x \to$ *Block 1* $\to$ *Block 2* $\to$ *Block 3* $\to$ *Block 4*

ScatResnet

$x \to$ *S* $\to$ *Block (a)* $\to$ *Block (b)* $\to$ *Block (c)*

- We replace the initial block of a ResNet by the order-1 Scattering:

	#params	Top 5	Top 1
Order 1,2 + ScatResNet-10	12,8M	88,6	68,7
Order 1 + ScatResnet-10	11,4M	87,7	67,7
Order 1 + WideScatResNet-50	107,2M	92,8	76,2
Order 1 + ScatResNet-50	27,8M	92,0	74,5
ResNet-50 (pytorch)	25,6M	92,9	76,1
ResNet-101 (pytorch)	45,4M	93,6	77,4
WideResNet-50	68,9M	94,0	77,9

Detection performances

Resnet

$x \to$ *Block 1* $\to$ *Block 2* $\to$ *Block 3* $\to$ *Faster RCNN modules*

ScatResnet

$x \to$ *S* $\to$ *Block (a)* $\to$ *Block (b)* $\to$ *Faster RCNN modules*

- We ablated the initial layers of a pre-trained ScatResNet.
- Our detection experiments demonstrate the spatial localisation of image details is preserved.

Pascal VOC7

	mAP
Faster-RCNN Order 1 + ScatResNet-50	73.3
Faster—RCNN ResNet-50 (ours)	70,5
Faster—RCNN ResNet-101 (ours)	72,5
Faster-RCNN VGG-16	70,2

COCO

	mAP
Faster-RCNN Order 1 + ScatResNet-50	32,2
Faster—RCNN ResNet-50 (ours)	31,0
Faster—RCNN ResNet-101 (ours)	34,5
Faster-RCNN VGG-16	29,2
Detectron	41,8

Speed performances

- Implemented via pytorch, we observe several savings:

Architecture	Speed (64 images)	Max Im Single gpu	Speed (4 images)	Max Im (Coco)
Order 1 + ScatResNet-50	0,072	175	0,073	9
ResNet-50	0,095	120	0,104	7
ResNet-101	0,158	70	0,182	2

CuPy

Conclusion

- Compress inputs and obtain a limited loss for supervised tasks
- Allows several memory and computation savings without learning.
- We applied no learning as the signals are natural images: can we learn better filters than wavelets for reducing a signal?

Related works

- *Towards Image Understanding from Deep Compression Without Decoding,* Torfason et al., ICLR 2018
- *Faster Neural Networks Straight from JPEG,* Gueguen et al., ICLR workshop 2018