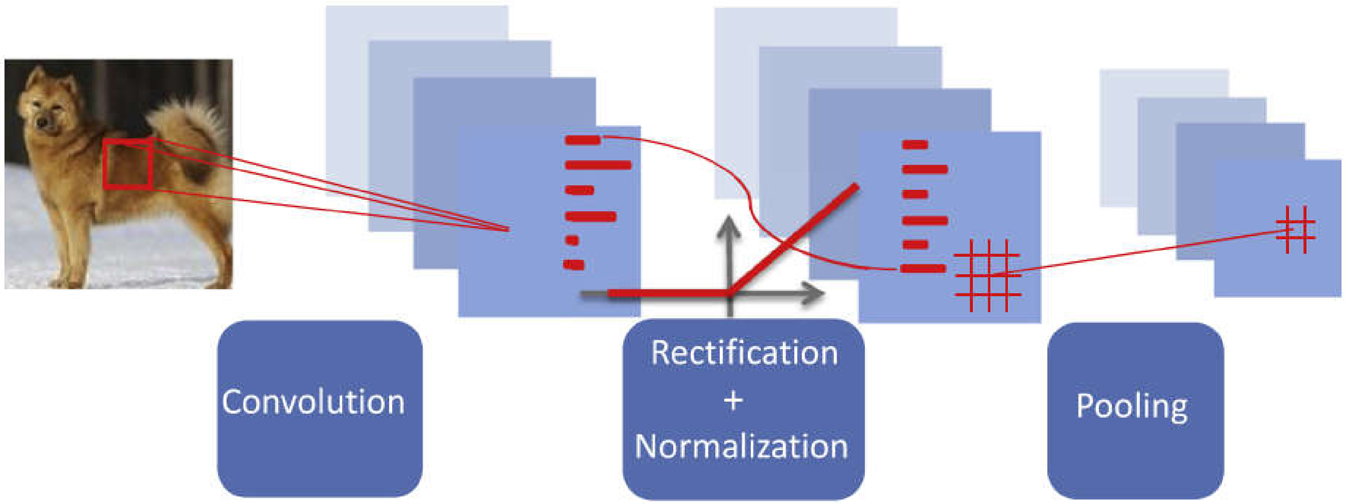
Detail-Preserving Pooling in Deep Networks

Zenglin Shi

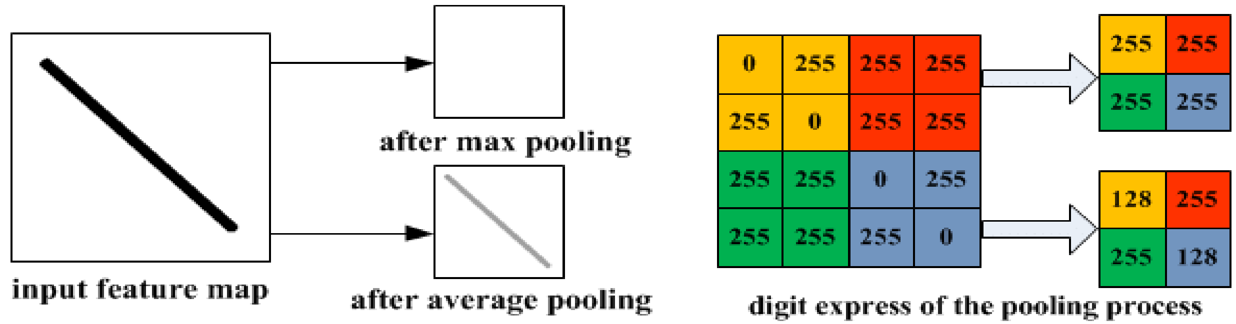
Dec. 10, 2018

Pooling Mechanism

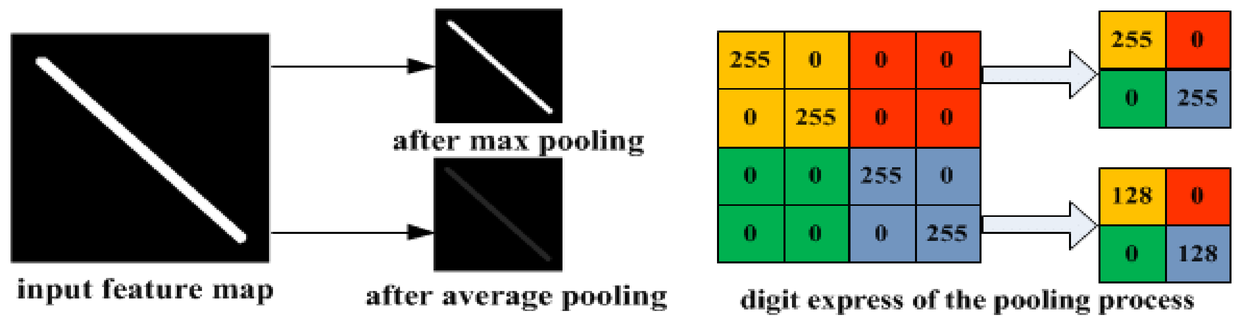


- Dimention Reduction
- Information Compression
- RF Enlarging
- Linear Transformation
- Translation Invariance

Max & AVG. Pooling

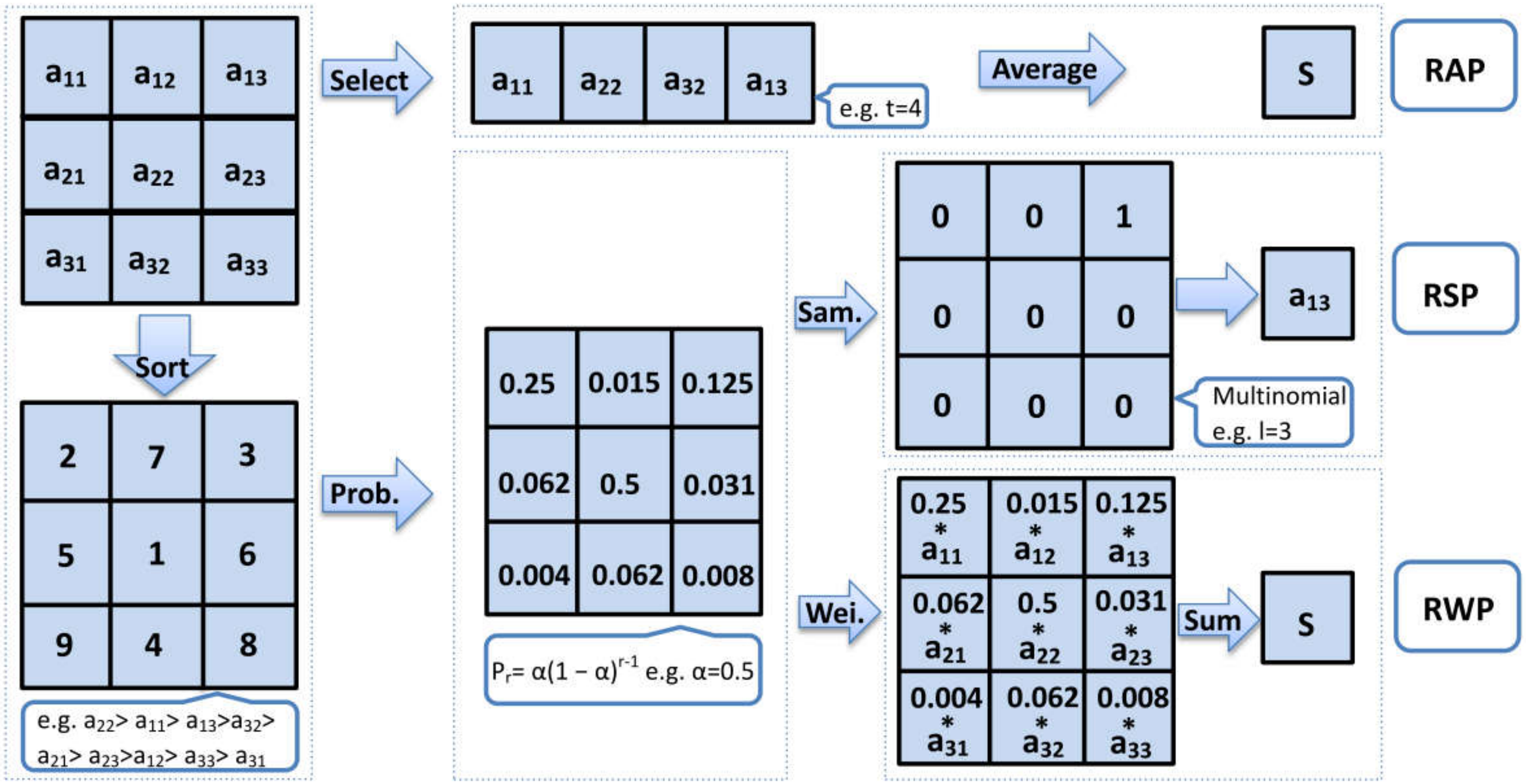


(a) Illustration of max pooling drawback

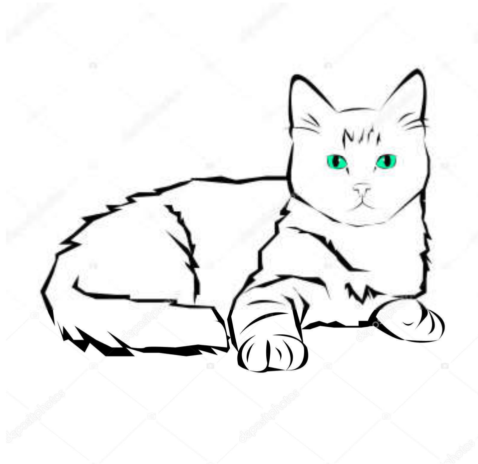


(b) Illustration of average pooling drawback

Weighted AVG. Pooling



Detail-Preserving Pooling-Motivation



Detail-Preserving Pooling-DPID

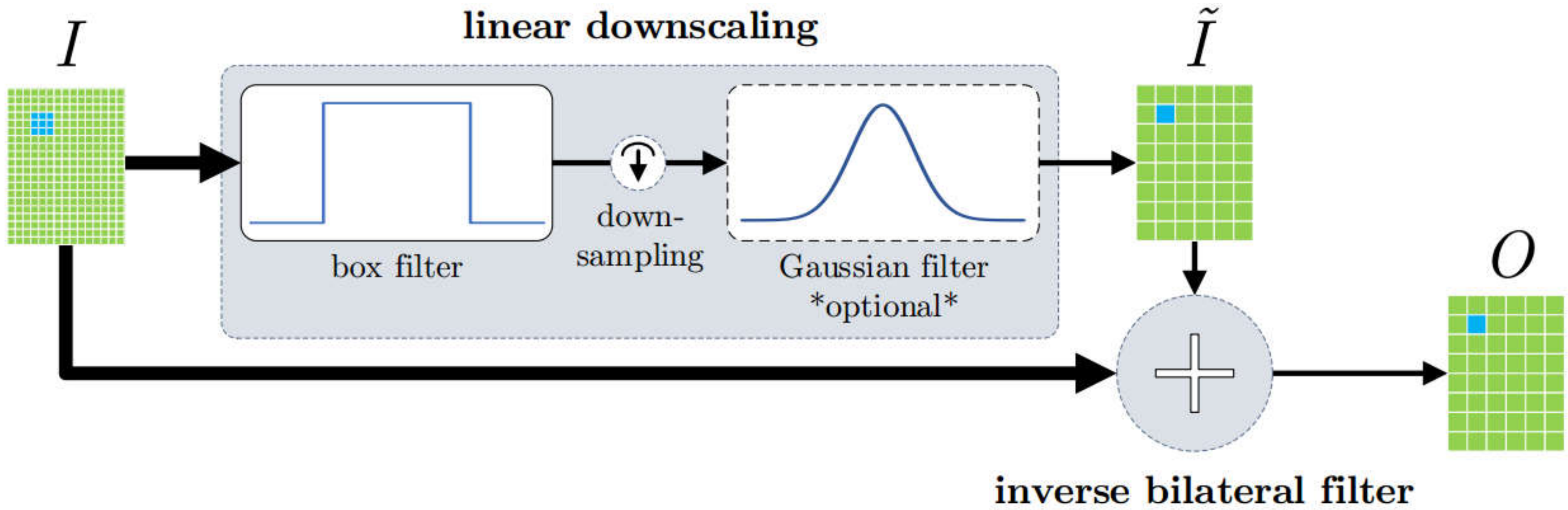


Figure 2. Diagram of detail-preserving downscaling (DPID) [31] and our detail-preserving pooling (DPP). DPP omits the Gaussian filter; Full-DPP replaces the box filter with a learned 2D filter.

Detail-Preserving Pooling-DPID

- Given an input image I[·], detail-preserving image downscaling (DPID) calculates the downscaled output at pixel p as

$$O[p] = \frac{1}{k_p} \sum_{q \in \Omega_p} I[q] \cdot \left\| I[q] - \tilde{I}[p] \right\|^{\lambda}, \qquad (1)$$

- in which the linearly downscaled image I˜ is given by

$$\tilde{I} = I_D * \tfrac{1}{16} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}^T \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}. \qquad (2)$$

- The weights are normalized with k_p

$$k_p = \sum_{q \in \Omega_p} \| I[q] - \tilde{I}[p] \|^{\lambda} \qquad (3)$$

Detail-Preserving Pooling-DPID

(a) Original Image

(b) DPID ($\lambda = 1$)

(c) Box filter (average pooling)

(d) Spatial maximum \ extremum

Detail-Preserving Pooling-DPP

- Detail-preserving pooling of an input activation map I at spatial output position p as

$$\mathcal{D}_{\alpha,\lambda}(I)[p] = \frac{1}{\sum_{q'\in\Omega_p} w_{\alpha,\lambda}[p,q']} \sum_{q\in\Omega_p} w_{\alpha,\lambda}[p,q] I[q]. \quad (4)$$

- Equation (4) computes a spatially weighted average of the input nodes in a neighborhood $I[q]_{q\in\Omega p}$ for which we define weights $w_{\alpha,\lambda}[p, q]$ as

$$w_{\alpha,\lambda}[p,q] = \alpha + \rho_\lambda \left(I[q] - \tilde{I}[p] \right). \quad (5)$$

Detail-Preserving Pooling-DPP

- For the symmetric variant of the reward function ρλ(·), employ the differentiable (generalized) Charbonnier penalty,

$$\rho_{\text{Sym}}(x) = \left(\sqrt{x^2 + \epsilon^2}\right)^{\lambda} \quad (6)$$

- The asymmetric variant of $\rho_{\lambda}(\cdot)$ only rewards positive arguments and is formulated as,

$$\rho_{\text{Asym}}(x) = \left(\sqrt{\max(0, x)^2 + \epsilon^2}\right)^{\lambda}, \quad (7)$$

Detail-Preserving Pooling-DPP

- For the sake of simplicity in notation, we reformulate this such that the weights are normalized as

$$\tilde{w}_{\alpha,\lambda}[p,q] = \frac{w_{\alpha,\lambda}[p,q]}{\sum_{q' \in \Omega_p} w_{\alpha,\lambda}[p,q']} \quad (8)$$

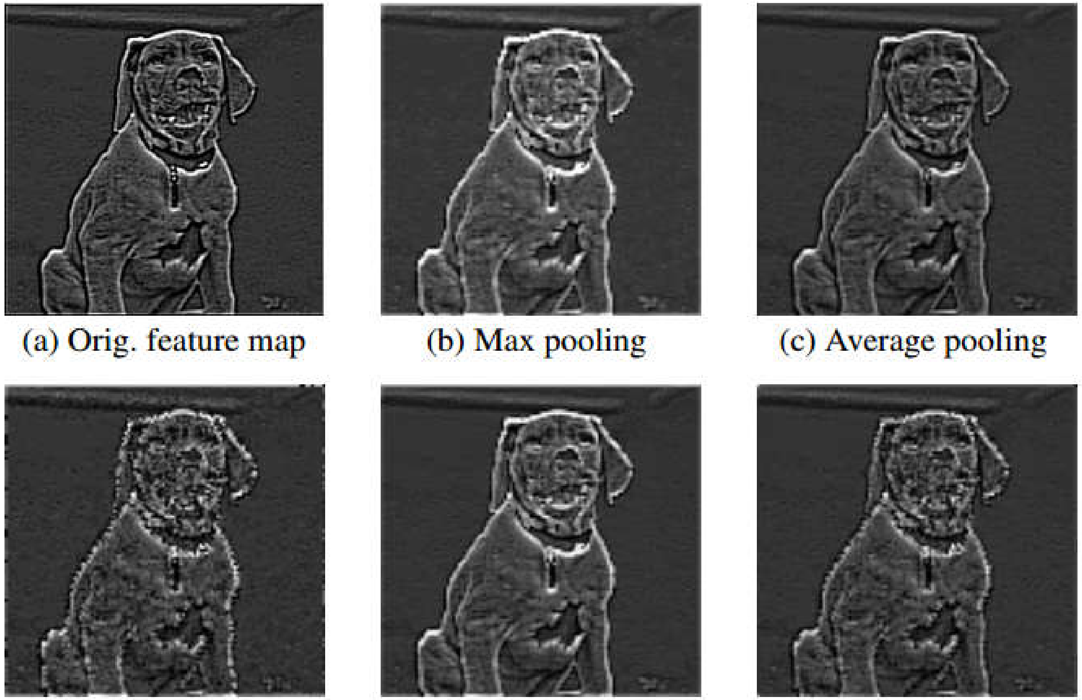
- which allows us to write DPP as

$$\mathcal{D}_{\alpha,\lambda}(I)[p] = \sum_{q \in \Omega_p} \tilde{w}_{\alpha,\lambda}[p,q] I[q] \quad (9)$$

- In Eq. (4), I˜ is the result of a linear downscaling, achieving full flexibility with

$$\tilde{I}_F[p] = \sum_{q \in \tilde{\Omega}_p} F[q] I[q] \qquad (10)$$

Detail-Preserving Pooling-DPP



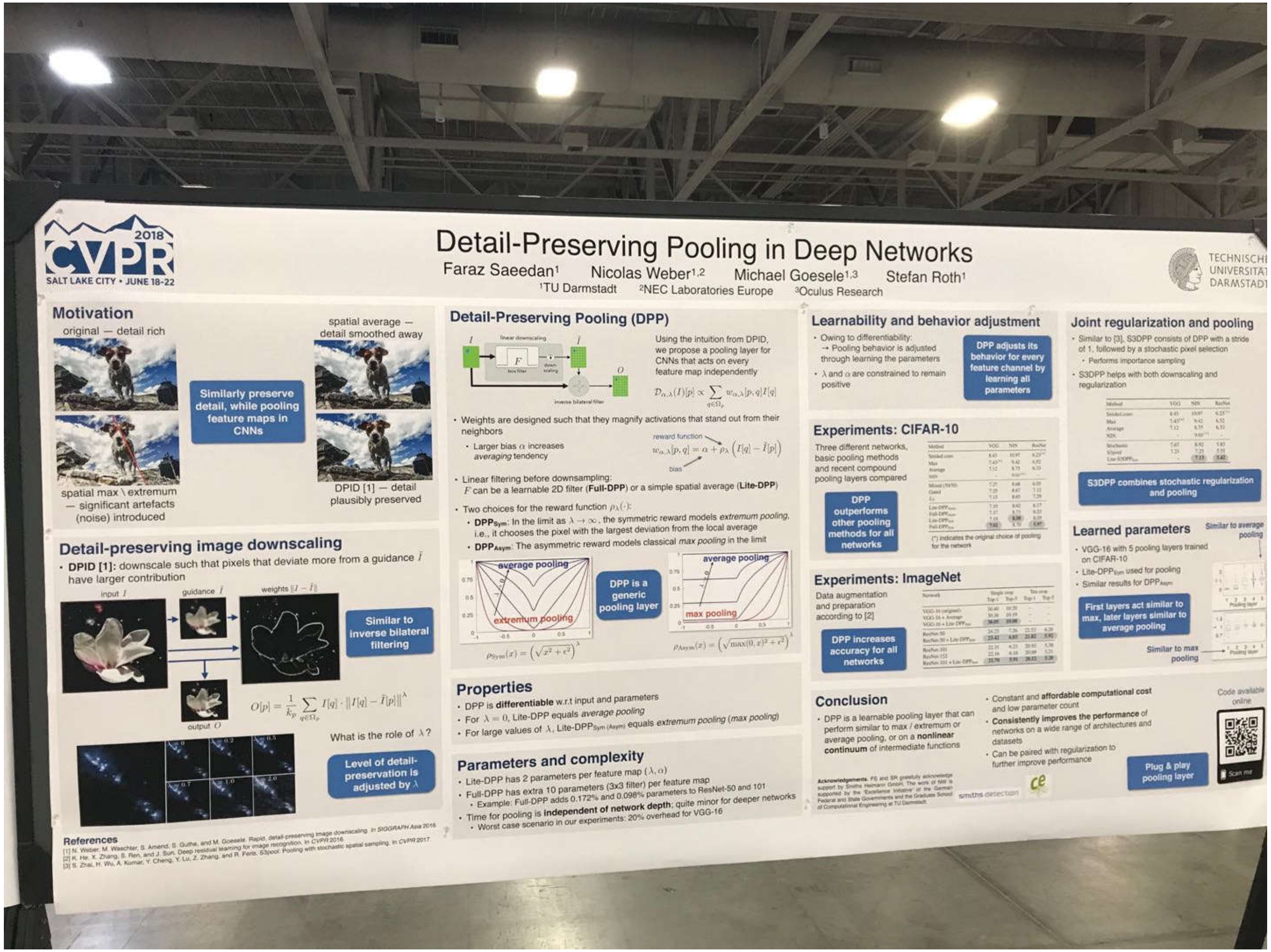
(a) Orig. feature map (b) Max pooling (c) Average pooling

(d) Extremum pooling (e) Asymmetric DPP ($\lambda = 1$) (f) Symmetric DPP ($\lambda = 2.7$)

Figure 4. Visualization of different pooling methods on an example feature map taken from the second layer of VGG-16. For both reward variants, the bias α is set to 0 to visually magnify the effect of the inverse bilateral weights. *Best viewed on screen.*

Detail-Preserving Pooling-DPP

	Method	VGG	NIN	ResNet
Deterministic methods	Strided conv.	8.43 ± 0.20	10.97 ± 0.10	$6.23^{(*)}$
	Max	$7.43 \pm 0.20^{(*)}$	9.42 ± 0.07	6.52
	Average	7.12 ± 0.18	8.75 ± 0.15	6.33
	NIN	–	$9.01 \pm 0.11^{(*)}$	–
	Mixed (50/50)	7.27 ± 0.20	8.68 ± 0.23	6.05
	Gated	7.25 ± 0.14	8.67 ± 0.22	7.12
	L_2	7.15 ± 0.18	8.65 ± 0.12	7.29
	Lite-$\mathrm{DPP}_{\mathrm{Asym}}$	$\underline{7.10} \pm 0.15$	8.62 ± 0.10	6.17
	Full-$\mathrm{DPP}_{\mathrm{Asym}}$	7.17 ± 0.18	8.73 ± 0.05	6.23
	Lite-$\mathrm{DPP}_{\mathrm{Sym}}$	7.19 ± 0.10	8.58 ± 0.11	6.05
	Full-$\mathrm{DPP}_{\mathrm{Sym}}$	$\mathbf{7.02} \pm 0.18$	8.70 ± 0.14	5.97
Stoch.	Stochastic	7.67 ± 0.10	8.92 ± 0.09	5.83
	S3pool	7.21 ± 0.14	$\underline{7.23} \pm 0.08$	$\underline{5.55}$
	Lite-$\mathrm{S3DPP}_{\mathrm{Sym}}$	–	$\mathbf{7.13} \pm 0.09$	**5.42**


CVPR 2018
SALT LAKE CITY • JUNE 18-22
Detail-Preserving Pooling in Deep Networks
Faraz Saeedan1 Nicolas Weber1,2 Michael Goesele1,3 Stefan Roth1
1TU Darmstadt 2NEC Laboratories Europe 3Oculus Research
TECHNISCHE UNIVERSITÄT DARMSTADT
Motivation
original — detail rich
spatial average — detail smoothed away
Similarly preserve detail, while pooling feature maps in CNNs
spatial max \ extremum — significant artefacts (noise) introduced
DPID [1] — detail plausibly preserved
Detail-preserving image downscaling
• DPID [1]: downscale such that pixels that deviate more from a guidance Ĩ have larger contribution
input I
guidance Ĩ
weights ‖I − Ĩ‖
output O
Similar to inverse bilateral filtering
What is the role of λ?
Level of detail-preservation is adjusted by λ
Detail-Preserving Pooling (DPP)
linear downscaling
box filter
down scaling
inverse bilateral filter
Using the intuition from DPID, we propose a pooling layer for CNNs that acts on every feature map independently
• Weights are designed such that they magnify activations that stand out from their neighbors
• Larger bias α increases averaging tendency
reward function
bias
• Linear filtering before downsampling: F can be a learnable 2D filter (Full-DPP) or a simple spatial average (Lite-DPP)
• Two choices for the reward function ρλ(·):
• DPPSym: In the limit as λ → ∞, the symmetric reward models extremum pooling, i.e., it chooses the pixel with the largest deviation from the local average
• DPPAsym: The asymmetric reward models classical max pooling in the limit
average pooling
extremum pooling
DPP is a generic pooling layer
average pooling
max pooling
Properties
• DPP is differentiable w.r.t input and parameters
• For λ = 0, Lite-DPP equals average pooling
• For large values of λ, Lite-DPPSym (Asym) equals extremum pooling (max pooling)
Parameters and complexity
• Lite-DPP has 2 parameters per feature map (λ, α)
• Full-DPP has extra 10 parameters (3x3 filter) per feature map
• Example: Full-DPP adds 0.172% and 0.098% parameters to ResNet-50 and 101
• Time for pooling is independent of network depth; quite minor for deeper networks
• Worst case scenario in our experiments: 20% overhead for VGG-16
Learnability and behavior adjustment
• Owing to differentiability:
→ Pooling behavior is adjusted through learning the parameters
• λ and α are constrained to remain positive
DPP adjusts its behavior for every feature channel by learning all parameters
Experiments: CIFAR-10
Three different networks, basic pooling methods and recent compound pooling layers compared
DPP outperforms other pooling methods for all networks
(*) indicates the original choice of pooling for the network
Experiments: ImageNet
Data augmentation and preparation according to [2]
DPP increases accuracy for all networks
Conclusion
• DPP is a learnable pooling layer that can perform similar to max / extremum or average pooling, or on a nonlinear continuum of intermediate functions
• Constant and affordable computational cost and low parameter count
• Consistently improves the performance of networks on a wide range of architectures and datasets
• Can be paired with regularization to further improve performance
Plug & play pooling layer
smiths detection
Joint regularization and pooling
• Similar to [3], S3DPP consists of DPP with a stride of 1, followed by a stochastic pixel selection
• Performs importance sampling
• S3DPP helps with both downscaling and regularization
S3DPP combines stochastic regularization and pooling
Learned parameters
Similar to average pooling
• VGG-16 with 5 pooling layers trained on CIFAR-10
• Lite-DPPSym used for pooling
• Similar results for DPPAsym
First layers act similar to max, later layers similar to average pooling
Similar to max pooling
Code available online
Scan me
References
[1] N. Weber, M. Waechter, S. Amend, S. Guthe, and M. Goesele. Rapid, detail-preserving image downscaling. In SIGGRAPH Asia 2016.
[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR 2016.
[3] S. Zhai, H. Wu, A. Kumar, Y. Cheng, Y. Lu, Z. Zhang, and R. Feris. S3pool: Pooling with stochastic spatial sampling. In CVPR 2017.