Scale-free Texture Segmentation: Expert Feature-based versus Deep Learning strategies

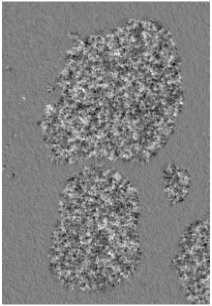
B. Pascal[1], V. Mauduit[1], N. Pustelnik[1], P. Abry[1]

28th European Signal Processing Conference (EUSIPCO 2020)
18 - 22 January 2021

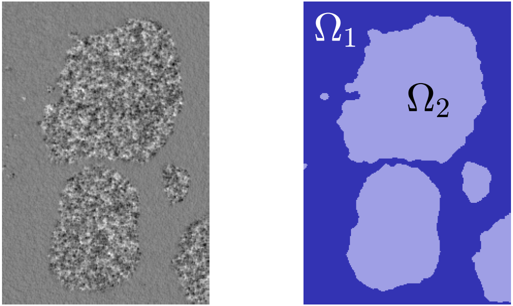
[1] Univ Lyon, ENS de Lyon, Univ Claude Bernard Lyon 1, CNRS, Laboratoire de Physique, F-69342 Lyon, France, firstname.lastname@ens-lyon.fr

Work supported by CBP (Blaise Pascal Center) with the use of SIDUS (Single Instance Distributing Universal System) implemented by E. Quemener.

Texture segmentation



Texture segmentation

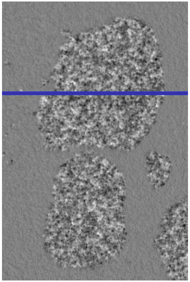


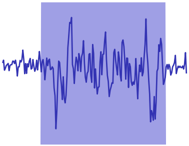
Purpose: obtain a partition of the image into Q **homogeneous** regions

$$\Omega = \Omega_1 \bigsqcup \ldots \bigsqcup \Omega_Q$$

Ω_q: pixels corresponding to the q^{th} texture

Piecewise monofractal model

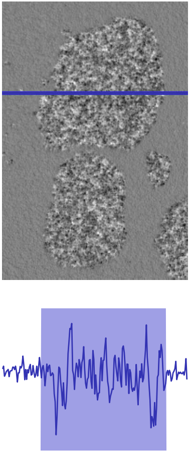




Piecewise monofractal model

Fractals attributes

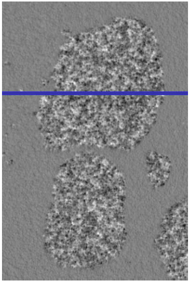
- variance σ^2 — *amplitude of variations*

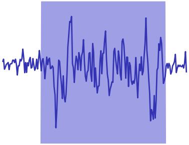


Piecewise monofractal model

Fractals attributes

- variance σ^2 — *amplitude of variations*
- local regularity h — *scale invariance*



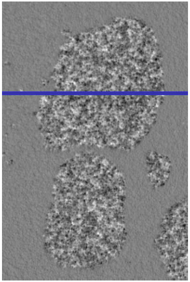


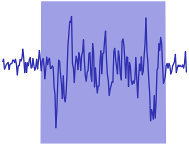
Piecewise monofractal model

Fractals attributes

- variance σ^2 — *amplitude of variations*
- local regularity h — *scale invariance*

$$|f(x) - f(y)| \leq \sigma(x)|x - y|^{h(x)}$$



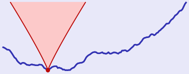


Piecewise monofractal model

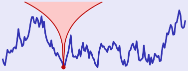
Fractals attributes

- variance σ^2 — *amplitude of variations*
- local regularity h — *scale invariance*

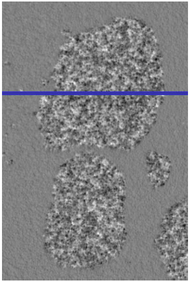
$$|f(x) - f(y)| \leq \sigma(x)|x - y|^{h(x)}$$

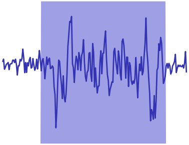


$h(x) \equiv h_1 = 0.9$



$h(x) \equiv h_2 = 0.3$





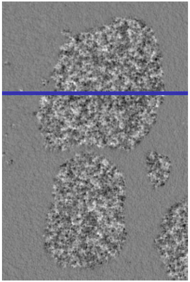
Piecewise monofractal model

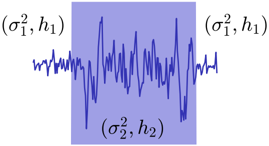
Fractals attributes

- variance σ^2 — *amplitude of variations*
- local regularity h — *scale invariance*

$$|f(x) - f(y)| \leq \sigma(x)|x - y|^{h(x)}$$

$h(x) \equiv h_1 = 0.9$ $\quad$ $h(x) \equiv h_2 = 0.3$





Segmentation

- σ^2 and h piecewise constant

Piecewise monofractal model

Fractals attributes

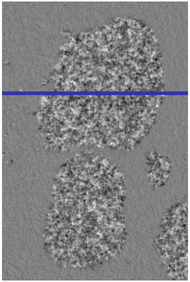
- <u>variance σ^2</u> *amplitude of variations*
- <u>local regularity h</u> *scale invariance*

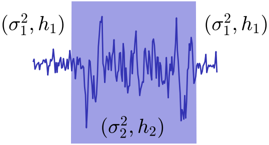
$$|f(x) - f(y)| \leq \sigma(x)|x - y|^{h(x)}$$

$h(x) \equiv h_1 = 0.9$ $\qquad$ $h(x) \equiv h_2 = 0.3$

Segmentation

- σ^2 and h piecewise constant
- region Ω_q characterized by (σ_q^2, h_q)





Unsupervised texture segmentation

Joint attributes estimation and regularization

Logarithm of wavelet *leaders*: $\ell_j(\boldsymbol{X})$

Unsupervised texture segmentation

Joint attributes estimation and regularization

Logarithm of wavelet *leaders*: $\ell_j(\boldsymbol{X}) \underset{2^j \to 0}{\simeq} \underset{\propto \log(\boldsymbol{\sigma}^2)}{\boldsymbol{v}} + j \underset{\text{regularity}}{\boldsymbol{h}}$

$$\sum_j \underbrace{\|\boldsymbol{v} + j\boldsymbol{h} - \ell_j(\boldsymbol{X})\|^2}_{\substack{\textbf{Least-Squares} \\ \to \text{ estimate fractal attributes}}}$$

Unsupervised texture segmentation

Joint attributes estimation and regularization

Logarithm of wavelet *leaders*: $\ell_j(\boldsymbol{X}) \underset{2^j \to 0}{\simeq} \underbrace{\boldsymbol{v}}_{\propto \log(\boldsymbol{\sigma}^2)} + j \underbrace{\boldsymbol{h}}_{\text{regularity}}$

Total Variation (TV) penalization

$$\mathcal{P}_\alpha(\boldsymbol{v}, \boldsymbol{h}) = \mathrm{TV}(\boldsymbol{v}) + \alpha \mathrm{TV}(\boldsymbol{h})$$

with $\mathrm{TV}(\boldsymbol{x}) = \|\mathbf{D}\boldsymbol{x}\|_{2,1}$ the isotropic Total Variation involving

- Horizontal and vertical differences $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2]$,
- Mixed norm: $\boldsymbol{z} = [\boldsymbol{z}_1; \dots, ; \boldsymbol{z}_I]$, $\quad \|\boldsymbol{z}\|_{2,1} = \sum_{\underline{n} \in \Omega} \|\boldsymbol{z}(\underline{n})\|_2$

$$\sum_j \underbrace{\|\boldsymbol{v} + j\boldsymbol{h} - \ell_j(\boldsymbol{X})\|^2}_{\substack{\textbf{Least-Squares} \\ \to \text{ estimate fractal attributes}}} + \lambda \underbrace{\mathcal{P}_\alpha(\boldsymbol{v}, \boldsymbol{h})}_{\substack{\textbf{Total Variation} \\ \to \text{ favors piecewise constancy}}}$$

Unsupervised texture segmentation

Joint attributes estimation and regularization

Logarithm of wavelet *leaders*: $\ell_j(\boldsymbol{X}) \underset{2^j \to 0}{\simeq} \underset{\propto \log(\boldsymbol{\sigma}^2)}{\boldsymbol{v}} + j \underset{\text{regularity}}{\boldsymbol{h}}$

Total Variation (TV) penalization

$$\mathcal{P}_\alpha(\boldsymbol{v}, \boldsymbol{h}) = \mathrm{TV}(\boldsymbol{v}) + \alpha \mathrm{TV}(\boldsymbol{h})$$

with $\mathrm{TV}(\boldsymbol{x}) = \|\mathbf{D}\boldsymbol{x}\|_{2,1}$ the isotropic Total Variation involving

- Horizontal and vertical differences $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2]$,
- Mixed norm: $\boldsymbol{z} = [\boldsymbol{z}_1; \ldots, ; \boldsymbol{z}_I]$, $\quad \|\boldsymbol{z}\|_{2,1} = \sum_{\underline{n} \in \Omega} \|\boldsymbol{z}(\underline{n})\|_2$

$$\left(\widehat{\boldsymbol{v}}, \widehat{\boldsymbol{h}}\right) = \underset{\boldsymbol{v}, \boldsymbol{h}}{\operatorname{argmin}} \sum_j \underbrace{\|\boldsymbol{v} + j\boldsymbol{h} - \ell_j(\boldsymbol{X})\|^2}_{\substack{\textbf{Least-Squares} \\ \to \text{ estimate fractal attributes}}} + \underbrace{\lambda\, \mathcal{P}_\alpha(\boldsymbol{v}, \boldsymbol{h})}_{\substack{\textbf{Total Variation} \\ \to \text{ favors piecewise constancy}}}$$

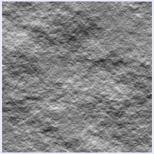
Unsupervised texture segmentation

Joint attributes estimation and regularization for TV-based segmentation

$$\left(\widehat{\boldsymbol{v}}, \widehat{\boldsymbol{h}}\right) = \underset{\boldsymbol{v},\boldsymbol{h}}{\operatorname{argmin}} \sum_j \underbrace{\|\boldsymbol{v} + j\boldsymbol{h} - \ell_j(\boldsymbol{X})\|^2}_{\substack{\textbf{Least-Squares} \\ \to \text{ estimate fractal attributes}}} + \lambda \underbrace{\mathcal{P}_\alpha(\boldsymbol{v}, \boldsymbol{h})}_{\substack{\textbf{Total Variation} \\ \to \text{ favors piecewise constancy}}}$$

Example of $Q = 2$ class segmentation

Textured image $\boldsymbol{X}$



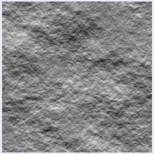
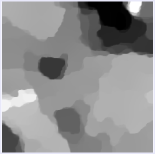
True segmentation

Unsupervised texture segmentation

Joint attributes estimation and regularization for TV-based segmentation

$$\left(\widehat{\boldsymbol{v}}, \widehat{\boldsymbol{h}}\right) = \underset{\boldsymbol{v},\boldsymbol{h}}{\operatorname{argmin}} \sum_j \underbrace{\|\boldsymbol{v} + j\boldsymbol{h} - \ell_j(\boldsymbol{X})\|^2}_{\substack{\textbf{Least-Squares} \\ \to \text{ estimate fractal attributes}}} + \lambda \underbrace{\mathcal{P}_\alpha(\boldsymbol{v}, \boldsymbol{h})}_{\substack{\textbf{Total Variation} \\ \to \text{ favors piecewise constancy}}}$$

Example of $Q = 2$ class segmentation

Textured image $\boldsymbol{X}$	True segmentation	*Joint* TV estimate $\widehat{\boldsymbol{h}}$
		

Unsupervised texture segmentation

Joint attributes estimation and regularization for TV-based segmentation

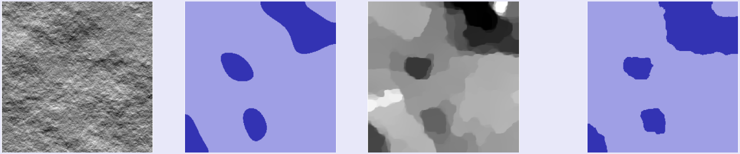
$$\left(\widehat{\boldsymbol{v}}, \widehat{\boldsymbol{h}}\right) = \underset{\boldsymbol{v},\boldsymbol{h}}{\operatorname{argmin}} \sum_j \underbrace{\|\boldsymbol{v} + j\boldsymbol{h} - \ell_j(\boldsymbol{X})\|^2}_{\text{Least-Squares}} + \lambda \underbrace{\mathcal{P}_\alpha(\boldsymbol{v}, \boldsymbol{h})}_{\text{Total Variation}}$$

Least-Squares → estimate fractal attributes

Total Variation → favors piecewise constancy

Example of $Q = 2$ class segmentation

Textured image $\boldsymbol{X}$	True segmentation	*Joint* TV estimate $\widehat{\boldsymbol{h}}$	Threshold estimate[†] $\widehat{\boldsymbol{M}}_{Joint} = \mathcal{T}_Q(\widehat{\boldsymbol{h}})$



[†] *(Cai, 2013)*

Supervised texture segmentation

Deep learning of Convolutional Neural Networks (CNN)

Supervised texture segmentation

Deep learning of Convolutional Neural Networks (CNN)

`input`
image $\boldsymbol{x}$ 

Supervised texture segmentation

Deep learning of Convolutional Neural Networks (CNN)

`input`		`Neural Network`
image $\boldsymbol{X}$	$\longrightarrow$	successive layers denoted $\mathcal{R}_\theta$
		where $\theta \in \mathbb{R}^{\mathcal{W}}$ are the network parameters

Supervised texture segmentation

Deep learning of Convolutional Neural Networks (CNN)

input		Neural Network		output
image $\boldsymbol{X}$	$\longrightarrow$	successive layers denoted $\mathcal{R}_\theta$	$\longrightarrow$	estimated label map $\widehat{\boldsymbol{M}}_{\mathrm{FCN}}$
		where $\theta \in \mathbb{R}^{\mathcal{W}}$ are the network parameters		

Supervised texture segmentation

Deep learning of Convolutional Neural Networks (CNN)

`input`		`Neural Network`		`output`
image $\boldsymbol{X}$	$\longrightarrow$	successive layers denoted $\mathcal{R}_\theta$	$\longrightarrow$	estimated label map $\widehat{\boldsymbol{M}}_{\text{FCN}}$
		where $\theta \in \mathbb{R}^{\mathcal{W}}$ are the network parameters		

Supervised learning:

$\longrightarrow$ minimization of the loss d summed over a learning dataset

$$\left\{ \left(\boldsymbol{X}^{(s)}, \boldsymbol{M}^{(s)} \right), \, s = 1, \ldots, \mathcal{S} \right\}$$

with $\boldsymbol{X}^{(s)}$: piecewise homogeneous texture, $\boldsymbol{M}^{(s)}$ true label map

Supervised texture segmentation

Deep learning of Convolutional Neural Networks (CNN)

input		Neural Network		output
image $\boldsymbol{X}$	$\longrightarrow$	successive layers denoted $\mathcal{R}_\theta$	$\longrightarrow$	estimated label map $\widehat{\boldsymbol{M}}_{\mathrm{FCN}}$
		where $\theta \in \mathbb{R}^{\mathcal{W}}$ are the network parameters		

Supervised learning:

$\longrightarrow$ minimization of the loss d summed over a learning dataset

$$\left\{ \left(\boldsymbol{X}^{(s)}, \boldsymbol{M}^{(s)} \right), \, s = 1, \ldots, \mathcal{S} \right\}$$

with $\boldsymbol{X}^{(s)}$: piecewise homogeneous texture, $\boldsymbol{M}^{(s)}$ true label map

$$\widehat{\theta} = \underset{\theta \in \mathbb{R}^{\mathcal{W}}}{\arg\min} \sum_{s=1}^{\mathcal{S}} d\left(\mathcal{R}_\theta(\boldsymbol{X}^{(s)}), \boldsymbol{M}^{(s)} \right)$$

Neural network architecture

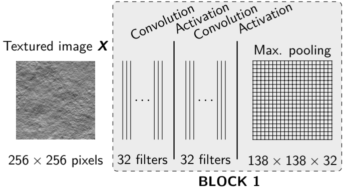
Schematic representation of the "simplest" net with $4 \cdot 10^5$ weights

Textured image $\boldsymbol{X}$

256×256 pixels

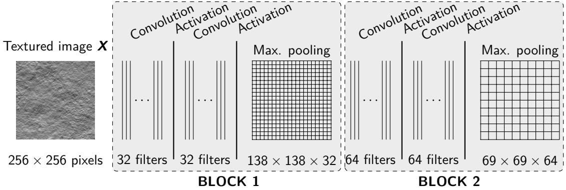
Neural network architecture

Schematic representation of the “simplest” net with $4 \cdot 10^5$ weights



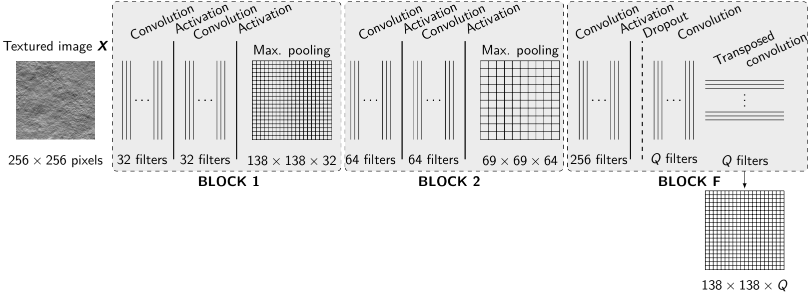
Neural network architecture

Schematic representation of the “simplest” net with $4 \cdot 10^5$ weights



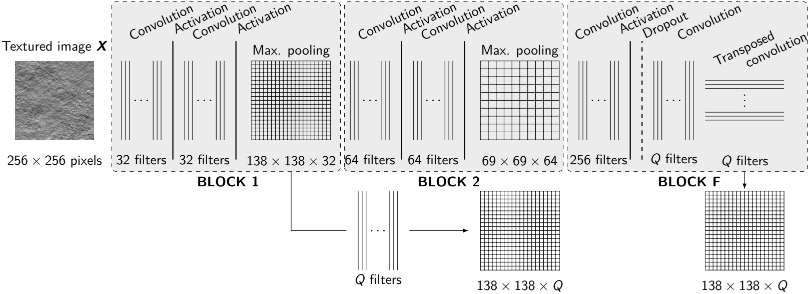
Neural network architecture

Schematic representation of the "simplest" net with $4 \cdot 10^5$ weights



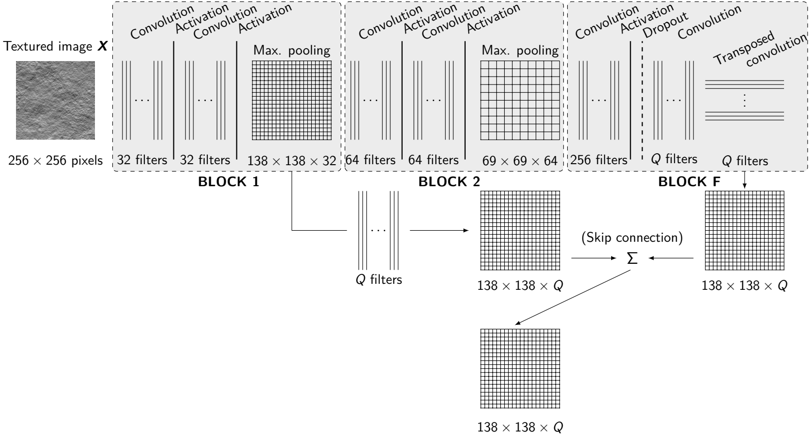
Neural network architecture

Schematic representation of the "simplest" net with $4 \cdot 10^5$ weights



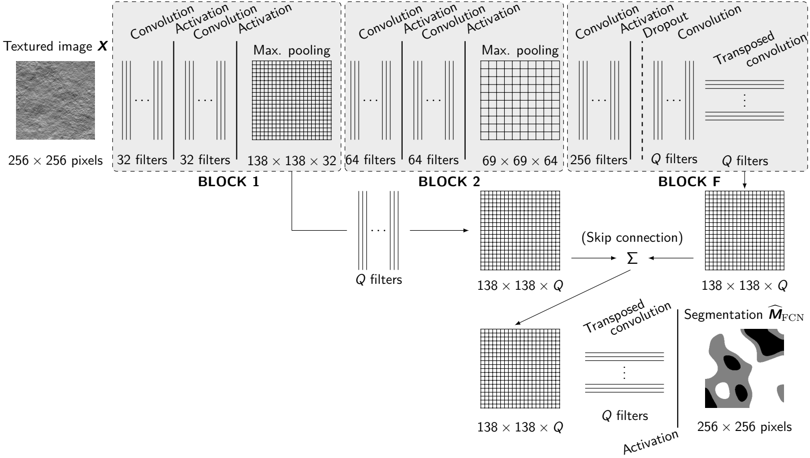
Neural network architecture

Schematic representation of the "simplest" net with $4 \cdot 10^5$ weights



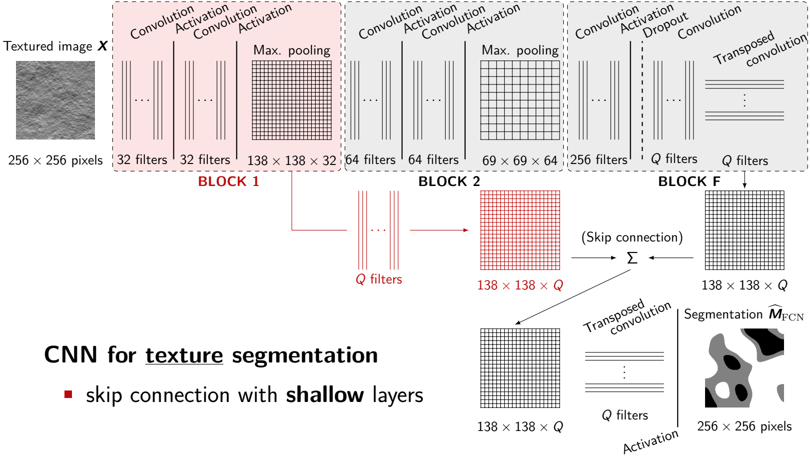
Neural network architecture

Schematic representation of the "simplest" net with $4 \cdot 10^5$ weights



Neural network architecture

Schematic representation of the “simplest” net with $4 \cdot 10^5$ weights

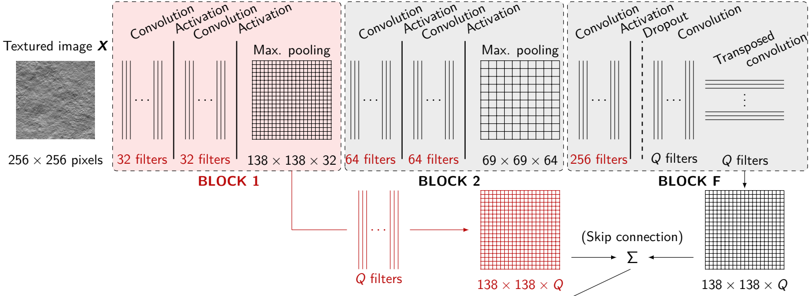


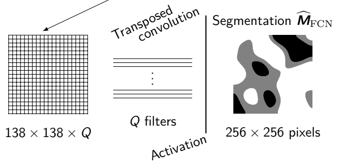
CNN for texture segmentation

- skip connection with **shallow** layers

Neural network architecture

Schematic representation of the "simplest" net with $4 \cdot 10^5$ weights





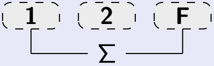
CNN for <u>texture</u> segmentation

- skip connection with **shallow** layers
- large number of filters

Neural network architecture

Three proposed networks of increasing number of weights $\mathcal{W}$

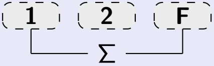
- $\mathcal{W} = 4 \cdot 10^5$ weights net: 2 blocks, 1 skip connection



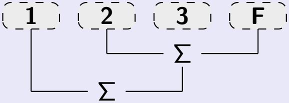
Neural network architecture

Three proposed networks of increasing number of weights $\mathcal{W}$

- $\mathcal{W} = 4 \cdot 10^5$ weights net: 2 blocks, 1 skip connection



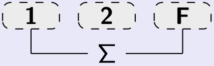
- $\mathcal{W} = 2 \cdot 10^6$ weights net: 3 blocks, 2 skip connections



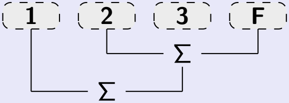
Neural network architecture

Three proposed networks of increasing number of weights $\mathcal{W}$

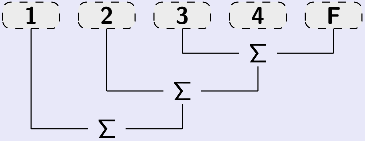
- <u>$\mathcal{W} = 4 \cdot 10^5$ weights net</u>: 2 blocks, 1 skip connection



- <u>$\mathcal{W} = 2 \cdot 10^6$ weights net</u>: 3 blocks, 2 skip connections



- <u>$\mathcal{W} = 8 \cdot 10^7$ weights net</u>: 4 blocks, 3 skip connections



Learning dataset of piecewise monofractal textures

Geometry of underlying true label maps

Learning dataset: $\left\{\left(\boldsymbol{X}^{(s)}, \boldsymbol{M}^{(s)}\right), s = 1, \ldots, \mathcal{S}\right\}$, $\mathcal{S} = 2000$ with

$\boldsymbol{X}^{(s)}$: piecewise homogeneous texture, $\boldsymbol{M}^{(s)}$ true label map

Learning dataset of piecewise monofractal textures

Geometry of underlying true label maps

Learning dataset: $\left\{\left(\boldsymbol{X}^{(s)}, \boldsymbol{M}^{(s)}\right), s = 1, \ldots, \mathcal{S}\right\}$, $\mathcal{S} = 2000$ with

$\boldsymbol{X}^{(s)}$: piecewise homogeneous texture, $\boldsymbol{M}^{(s)}$ true label map

Learning dataset of piecewise monofractal textures

Geometry of underlying true label maps

Learning dataset: $\left\{\left(\boldsymbol{X}^{(s)}, \boldsymbol{M}^{(s)}\right), s = 1, \ldots, \mathcal{S}\right\}$, $\mathcal{S} = 2000$ with

$\boldsymbol{X}^{(s)}$: piecewise homogeneous texture, $\boldsymbol{M}^{(s)}$ true label map

Purpose: **Texture** segmentation with $Q \in \{2, 3, 4\}$ classes ...

Learning dataset of piecewise monofractal textures

Geometry of underlying true label maps

Learning dataset: $\left\{ \left(\boldsymbol{X}^{(s)}, \boldsymbol{M}^{(s)} \right), s = 1, \ldots, \mathcal{S} \right\}$, $\mathcal{S} = 2000$ with

$\boldsymbol{X}^{(s)}$: piecewise homogeneous texture, $\boldsymbol{M}^{(s)}$ true label map

Purpose: **Texture** segmentation with $Q \in \{2, 3, 4\}$ classes ...

... necessary that no **shape** is learned.

Learning dataset of piecewise monofractal textures

Geometry of underlying true label maps

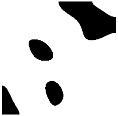
Learning dataset: $\left\{\left(\boldsymbol{X}^{(s)}, \boldsymbol{M}^{(s)}\right), s = 1, \dots, \mathcal{S}\right\}$, $\mathcal{S} = 2000$ with

$\boldsymbol{X}^{(s)}$: piecewise homogeneous texture, $\boldsymbol{M}^{(s)}$ true label map

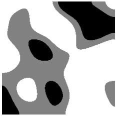
Purpose: **Texture** segmentation with $Q \in \{2, 3, 4\}$ classes . . .

. . . necessary that no **shape** is learned.

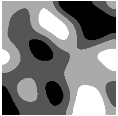
Random segmentation masks



Two classes ($Q = 2$)



Three classes ($Q = 3$)



Four classes ($Q = 4$)

Learning dataset of piecewise monofractal textures

Geometry of underlying true label maps

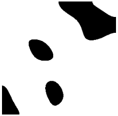
Learning dataset: $\left\{\left(\boldsymbol{X}^{(s)}, \boldsymbol{M}^{(s)}\right), s = 1, \ldots, \mathcal{S}\right\}$, $\mathcal{S} = 2000$ with

$\boldsymbol{X}^{(s)}$: piecewise homogeneous texture, $\boldsymbol{M}^{(s)}$ true label map

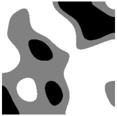
Purpose: **Texture** segmentation with $Q \in \{2, 3, 4\}$ classes . . .

. . . necessary that no **shape** is learned.

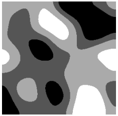
Random segmentation masks



Two classes ($Q = 2$)



Three classes ($Q = 3$)



Four classes ($Q = 4$)

Partition: $\Omega = \cup_{q=1}^{Q} \Omega_q^{(s)}$, where $\Omega_q^{(s)} = \left\{\underline{n} \,|\, M^{(s)}(\underline{n}) = q\right\}$

Learning dataset of piecewise monofractal textures

Fractal texture samples

Learning dataset: $\left\{ \left(\boldsymbol{X}^{(s)}, \boldsymbol{M}^{(s)} \right), s = 1, \ldots, \mathcal{S} \right\}$, $\mathcal{S} = 2000$ with

$\boldsymbol{X}^{(s)}$: piecewise homogeneous texture, $\boldsymbol{M}^{(s)}$ true label map

Learning dataset of piecewise monofractal textures

Fractal texture samples

Learning dataset: $\left\{ \left(\boldsymbol{X}^{(s)}, \boldsymbol{M}^{(s)} \right), s = 1, \ldots, \mathcal{S} \right\}$, $\mathcal{S} = 2000$ with

$\boldsymbol{X}^{(s)}$: piecewise homogeneous texture, $\boldsymbol{M}^{(s)}$ true label map

Texture configuration $\left\{ \left(\sigma_q^2, h_q \right), q = 1, \ldots, Q \right\}$

Region Ω_q: fractal texture characterized by $\left(\sigma_q^2, h_q \right)$

Learning dataset of piecewise monofractal textures

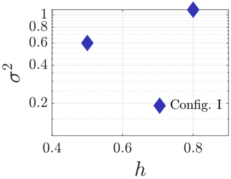
Fractal texture samples

Learning dataset: $\left\{ \left(\boldsymbol{X}^{(s)}, \boldsymbol{M}^{(s)} \right), s = 1, \ldots, \mathcal{S} \right\}$, $\mathcal{S} = 2000$ with

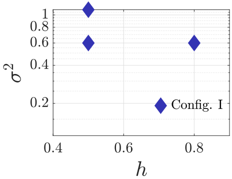
$\boldsymbol{X}^{(s)}$: piecewise homogeneous texture, $\boldsymbol{M}^{(s)}$ true label map

Texture configuration $\left\{ \left(\sigma_q^2, h_q \right), q = 1, \ldots, Q \right\}$

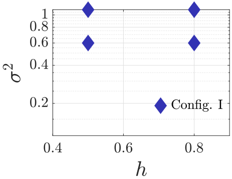
Region Ω_q: fractal texture characterized by $\left(\sigma_q^2, h_q \right)$



$Q = 2$



$Q = 3$



$Q = 4$

Config. I: large $\Delta h = h_q - h_{q'}$, small $\Delta \sigma^2 = \sigma_q^2 - \sigma_{q'}^2$

Learning dataset of piecewise monofractal textures

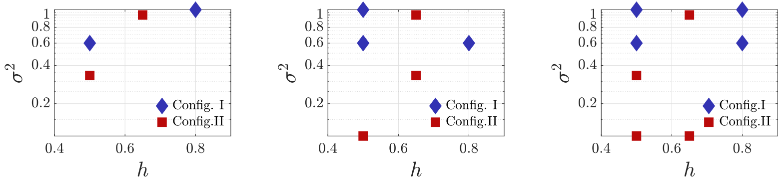
Fractal texture samples

Learning dataset: $\left\{\left(\boldsymbol{X}^{(s)}, \boldsymbol{M}^{(s)}\right), s = 1, \ldots, \mathcal{S}\right\}$, $\mathcal{S} = 2000$ with

$\boldsymbol{X}^{(s)}$: piecewise homogeneous texture, $\boldsymbol{M}^{(s)}$ true label map

Texture configuration $\left\{\left(\sigma_q^2, h_q\right), q = 1, \ldots, Q\right\}$

Region Ω_q: fractal texture characterized by $\left(\sigma_q^2, h_q\right)$



$Q = 2$ $Q = 3$ $Q = 4$

Config. I: large $\Delta h = h_q - h_{q'}$, small $\Delta \sigma^2 = \sigma_q^2 - \sigma_{q'}^2$

Config. II: small $\Delta h = h_q - h_{q'}$, large $\Delta \sigma^2 = \sigma_q^2 - \sigma_{q'}^2$

Unsupervised texture segmentation

Tuning of the regularization parameters (λ, α)

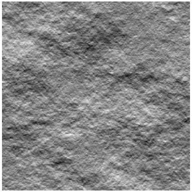
$$\left(\widehat{\boldsymbol{v}}, \widehat{\boldsymbol{h}}\right) = \underset{\boldsymbol{v},\boldsymbol{h}}{\operatorname{argmin}} \sum_j \underbrace{\|\boldsymbol{v} + j\boldsymbol{h} - \ell_j(\boldsymbol{X})\|^2}_{\substack{\textbf{Least-Squares} \\ \rightarrow \text{ estimate fractal attributes}}} + \lambda \underbrace{\mathcal{P}_\alpha(\boldsymbol{v}, \boldsymbol{h})}_{\substack{\textbf{Total Variation} \\ \rightarrow \text{ favors piecewise constancy}}}$$

Unsupervised texture segmentation

Tuning of the regularization parameters (λ, α)

$$\left(\widehat{\boldsymbol{v}}, \widehat{\boldsymbol{h}}\right) = \underset{\boldsymbol{v},\boldsymbol{h}}{\operatorname{argmin}} \underbrace{\sum_j \|\boldsymbol{v} + j\boldsymbol{h} - \ell_j(\boldsymbol{X})\|^2}_{\substack{\textbf{Least-Squares} \\ \to \text{ estimate fractal attributes}}} + \underbrace{\lambda\, \mathcal{P}_\alpha(\boldsymbol{v}, \boldsymbol{h})}_{\substack{\textbf{Total Variation} \\ \to \text{ favors piecewise constancy}}}$$

Textured image $\boldsymbol{X}^{(1)}$



True label map $\boldsymbol{M}^{(1)}$

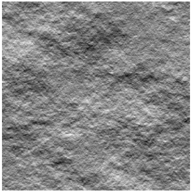
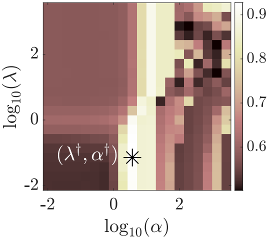
Unsupervised texture segmentation

Tuning of the regularization parameters (λ, α)

$$\left(\widehat{\boldsymbol{v}}, \widehat{\boldsymbol{h}}\right) = \underset{\boldsymbol{v},\boldsymbol{h}}{\operatorname{argmin}} \; \underbrace{\sum_j \|\boldsymbol{v} + j\boldsymbol{h} - \ell_j(\boldsymbol{X})\|^2}_{\text{Least-Squares}} + \lambda \underbrace{\mathcal{P}_\alpha(\boldsymbol{v}, \boldsymbol{h})}_{\text{Total Variation}}$$

Least-Squares → estimate fractal attributes

Total Variation → favors piecewise constancy

Textured image $\boldsymbol{X}^{(1)}$	True label map $\boldsymbol{M}^{(1)}$	Segmentation score
		

Unsupervised texture segmentation

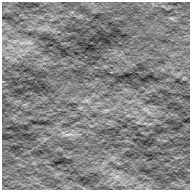
Tuning of the regularization parameters (λ, α)

$$\left(\widehat{\boldsymbol{v}}, \widehat{\boldsymbol{h}}\right) = \underset{\boldsymbol{v},\boldsymbol{h}}{\operatorname{argmin}} \sum_j \underbrace{\|\boldsymbol{v} + j\boldsymbol{h} - \ell_j(\boldsymbol{X})\|^2}_{\textbf{Least-Squares}} + \lambda \underbrace{\mathcal{P}_\alpha(\boldsymbol{v}, \boldsymbol{h})}_{\textbf{Total Variation}}$$

Least-Squares → estimate fractal attributes

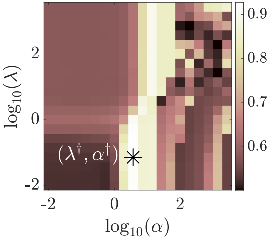
Total Variation → favors piecewise constancy

Textured image $\boldsymbol{X}^{(1)}$



True label map $\boldsymbol{M}^{(1)}$

Segmentation score



Grid search minimizing the segmentation error on $\boldsymbol{X}^{(1)}$

$\longrightarrow$ optimal $\left(\lambda^\dagger, \alpha^\dagger\right)$

Unsupervised texture segmentation

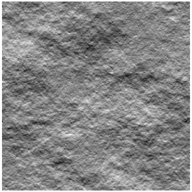
Tuning of the regularization parameters (λ, α)

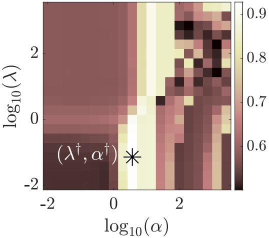
$$\left(\widehat{\boldsymbol{v}}, \widehat{\boldsymbol{h}}\right) = \underset{\boldsymbol{v},\boldsymbol{h}}{\operatorname{argmin}} \ \underbrace{\sum_j \|\boldsymbol{v} + j\boldsymbol{h} - \ell_j(\boldsymbol{X})\|^2}_{\text{Least-Squares}} + \lambda \underbrace{\mathcal{P}_\alpha(\boldsymbol{v}, \boldsymbol{h})}_{\text{Total Variation}}$$

Least-Squares → estimate fractal attributes

Total Variation → favors piecewise constancy

Textured image $\boldsymbol{X}^{(1)}$	True label map $\boldsymbol{M}^{(1)}$	Segmentation score





Grid search minimizing the segmentation error on $\boldsymbol{X}^{(1)}$

⟶ optimal $(\lambda^\dagger, \alpha^\dagger)$

⟶ **frozen** for computing performance on a testing set of 100 images

Supervised learning

Minimization of the loss $\widehat{\theta} = \underset{\theta \in \mathbb{R}^{\mathcal{W}}}{\arg\min} \sum_{s=1}^{\mathcal{S}} d\left(\mathcal{R}_\theta(\boldsymbol{X}^{(s)}), \boldsymbol{M}^{(s)}\right)$

using backward propagation of the gradient

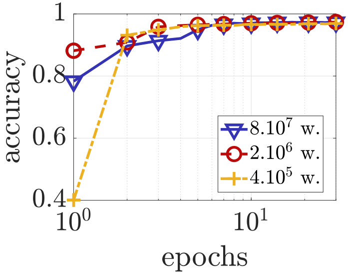
- ADAM optimizer with AMSGrad
- learning rate $2 \cdot 10^{-4}$
- batch size 20 images
- 30 epochs

Supervised learning

Minimization of the loss $\widehat{\theta} = \underset{\theta \in \mathbb{R}^{\mathcal{W}}}{\arg\min} \sum_{s=1}^{\mathcal{S}} d\left(\mathcal{R}_\theta(\boldsymbol{X}^{(s)}), \boldsymbol{M}^{(s)}\right)$

using backward propagation of the gradient

- ADAM optimizer with AMSGrad
- learning rate $2 \cdot 10^{-4}$
- batch size 20 images
- 30 epochs



Segmentation performance on a testing set of 100 images

Same configuration for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Segmentation performance on a testing set of 100 images

Same configuration for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Two regions $\boldsymbol{Q = 2}$

Config. I

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$93.2 \pm 0.8\%$	$97.3 \pm 0.6\%$	$97.4 \pm 0.6\%$	$96.9 \pm 0.7\%$

Segmentation performance on a testing set of 100 images

Same configuration for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Two regions $\boldsymbol{Q = 2}$

<u>Config. I</u>

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$93.2 \pm 0.8\%$	$97.3 \pm 0.6\%$	$97.4 \pm 0.6\%$	$96.9 \pm 0.7\%$

<u>Config. II</u>

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$97.8 \pm 0.2\%$	$99.1 \pm 0.2\%$	$99.0 \pm 0.2\%$	$99.1 \pm 0.2\%$

Segmentation performance on a testing set of 100 images

Same configuration for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Three regions $Q = 3$

Config. I

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$69.3 \pm 2.8\%$	$97.8 \pm 0.3\%$	$98.1 \pm 0.3\%$	$98.0 \pm 0.3\%$

Segmentation performance on a testing set of 100 images

Same configuration for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Three regions $Q = 3$

Config. I

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$69.3 \pm 2.8\%$	$97.8 \pm 0.3\%$	$98.1 \pm 0.3\%$	$98.0 \pm 0.3\%$

Config. II

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$95.2 \pm 3.1\%$	$98.3 \pm 0.3\%$	$98.5 \pm 0.3\%$	$98.4 \pm 0.3\%$

Segmentation performance on a testing set of 100 images

Same configuration for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Four regions $\boldsymbol{Q = 4}$

Config. I

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$58.6 \pm 1.5\%$	$97.1 \pm 0.4\%$	$96.8 \pm 0.5\%$	$96.5 \pm 0.5\%$

Segmentation performance on a testing set of 100 images

Same configuration for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Four regions $Q = 4$

Config. I

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$58.6 \pm 1.5\%$	$97.1 \pm 0.4\%$	$96.8 \pm 0.5\%$	$96.5 \pm 0.5\%$

Config. II

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$64.9 \pm 1.4\%$	$95.7 \pm 0.5\%$	$95.6 \pm 0.5\%$	$95.2 \pm 0.6\%$

Segmentation performance on a testing set of 100 images

Different configurations for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Two regions $Q = 2$

Trained on Config. I, tested on Config. II

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$79.2 \pm 2.9\%$	$91.2 \pm 2.1\%$	$87.9 \pm 2.5\%$	$81.8 \pm 3.8\%$

Segmentation performance on a testing set of 100 images

Different configurations for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Two regions $\boldsymbol{Q = 2}$

Trained on Config. I, tested on Config. II

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$79.2 \pm 2.9\%$	$91.2 \pm 2.1\%$	$87.9 \pm 2.5\%$	$81.8 \pm 3.8\%$

Trained on Config. II, tested on Config. I

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$90.9 \pm 2.8\%$	$56.2 \pm 13.5\%$	$55.1 \pm 14.0\%$	$55.5 \pm 13.8\%$

Segmentation performance on a testing set of 100 images

Different configurations for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Three regions $\boldsymbol{Q = 3}$

Trained on Config. I, tested on Config. II

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$95.2 \pm 1.2\%$	$65.7 \pm 7.2\%$	$69.0 \pm 7.6\%$	$65.2 \pm 7.2\%$

Segmentation performance on a testing set of 100 images

Different configurations for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Three regions $Q = 3$

Trained on Config. I, tested on Config. II

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$95.2 \pm 1.2\%$	$65.7 \pm 7.2\%$	$69.0 \pm 7.6\%$	$65.2 \pm 7.2\%$

Trained on Config. II, tested on Config. I

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$66.7 \pm 2.5\%$	$73.5 \pm 8.2\%$	$74.9 \pm 8.2\%$	$72.6 \pm 8.1\%$

Segmentation performance on a testing set of 100 images

Different configurations for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Four regions $\boldsymbol{Q = 4}$

Trained on Config. I, tested on Config. II

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$66.3 \pm 1.1\%$	$55.6 \pm 3.4\%$	$50.8 \pm 4.0\%$	$46.4 \pm 3.7\%$

Segmentation performance on a testing set of 100 images

Different configurations for the training and testing sets

Percentage of well-classified pixels over testing set averaged over the testing set

Four regions $Q = 4$

Trained on Config. I, tested on Config. II

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$66.3 \pm 1.1\%$	$55.6 \pm 3.4\%$	$50.8 \pm 4.0\%$	$46.4 \pm 3.7\%$

Trained on Config. II, tested on Config. I

Joint TV	FCNN $8 \cdot 10^7$	FCNN $2 \cdot 10^6$	FCNN $4 \cdot 10^5$
$52.0 \pm 1.5\%$	$50.9 \pm 3.9\%$	$51.3 \pm 4.3\%$	$50.2 \pm 3.8\%$

Conclusion and perspectives

- FCNN provides very accurate texture segmentations

Conclusion and perspectives

- FCNN provides very accurate texture segmentations
- Supervised networks outperform unsupervised *Joint* TV

Conclusion and perspectives

- FCNN provides very accurate texture segmentations
- Supervised networks outperform unsupervised *Joint* TV
  $\longrightarrow$ especially when the number of classes Q is large

Conclusion and perspectives

- FCNN provides very accurate texture segmentations
- Supervised networks outperform unsupervised *Joint* TV
  $\longrightarrow$ especially when the number of classes Q is large
- Reduced complexity $\mathcal{W}$ does not degrade performance

Conclusion and perspectives

- FCNN provides very accurate texture segmentations
- Supervised networks outperform unsupervised *Joint* TV
  $\longrightarrow$ especially when the number of classes Q is large
- Reduced complexity $\mathcal{W}$ does not degrade performance
- FCNN not robust to mismatch between training and testing sets

Conclusion and perspectives

- FCNN provides very accurate texture segmentations
- Supervised networks outperform unsupervised *Joint* TV
  $\longrightarrow$ especially when the number of classes Q is large
- Reduced complexity $\mathcal{W}$ does not degrade performance
- FCNN not robust to mismatch between training and testing sets
  $\longrightarrow$ for small Q *Joint* TV is more robust

Conclusion and perspectives

- FCNN provides very accurate texture segmentations
- Supervised networks outperform unsupervised *Joint* TV
  $\longrightarrow$ especially when the number of classes Q is large
- Reduced complexity $\mathcal{W}$ does not degrade performance
- FCNN not robust to mismatch between training and testing sets
  $\longrightarrow$ for small Q *Joint* TV is more robust
- FCNN provide very irregular contours

True label map — Joint TV — FCNN $8 \cdot 10^7$

Conclusion and perspectives

- FCNN provides very accurate texture segmentations
- Supervised networks outperform unsupervised *Joint* TV
  $\longrightarrow$ especially when the number of classes Q is large
- Reduced complexity $\mathcal{W}$ does not degrade performance
- FCNN not robust to mismatch between training and testing sets
  $\longrightarrow$ for small Q *Joint* TV is more robust
- FCNN provide very irregular contours

True label map — Joint TV — FCNN $8 \cdot 10^7$

LISTA *Learning Iterative Shrinkage and Thresholding Algorithm*
$\longrightarrow$ interpretation of sparse coding minimization scheme as a CNN