Learning Stochastic geometry models and Convolutional Neural Networks. Application to multiple object detection in aerospatial data sets.

Jules Mabon, Inria, Université Côte d'Azur, France
In collaboration with **Mathias Ortner** (Airbus DS) and **Josiane Zerubia** (Inria)

December 20, 2023

Introduction

Application Goal

- Detection (and vectorization) of small objects (🚗) in satellite images 🛰️

Challenges

📷 Low visual information & saliency

- Small sized objects at 0.5 m/px
- Partial occlusions, shadows, noise
- Visually diverse environments and objects
- Variable object density

🔗 Priors on interactions

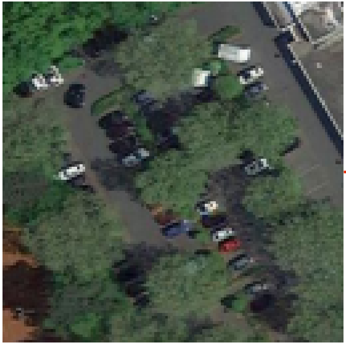
Image from the DOTA[1]dataset

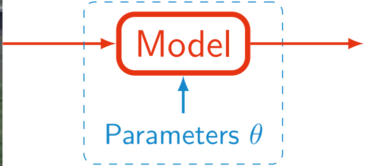
---

[1] Xia *et al.* 2018.

Object detection: basics

Image $\mathbf{X}$





Model

Parameters θ

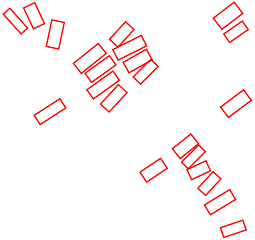
estimated before inferrence

once for all images

using set of ground truth $\mathbf{y}^{GT}$

Set of objects $\mathbf{y}$

estimated $\hat{\mathbf{y}}$

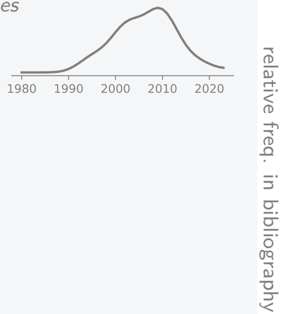


Proposed approach

Introducing CNN and EBM methods to Point Process

● **Point Process (PP)** *Configurations of points as random variables*

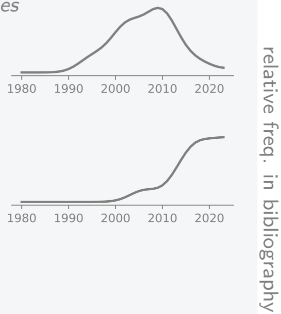
- ✓ Models geometric and interaction priors
- ✗ Tedious manual tuning, application specific



Proposed approach

Introducing CNN and EBM methods to Point Process

- **Point Process (PP)** *Configurations of points as random variables*
  - ✓ Models geometric and interaction priors
  - ✗ Tedious manual tuning, application specific
- **Convolutional Neural Network (CNN)**
  - ✓ Powerful at learning texture and extracting local information
  - ✗ Learning interactions is costly



Proposed approach

Introducing CNN and EBM methods to Point Process

- ● **Point Process (PP)** *Configurations of points as random variables*
  - ✓ Models geometric and interaction priors
  - ✗ Tedious manual tuning, application specific
- ● **Convolutional Neural Network (CNN)**
  - ✓ Powerful at learning texture and extracting local information
  - ✗ Learning interactions is costly
- ● **Energy Based Model (EBM)**
  - ✓ Captures dependencies in scalar energy
  - ✓ Framework to train and sample generative models

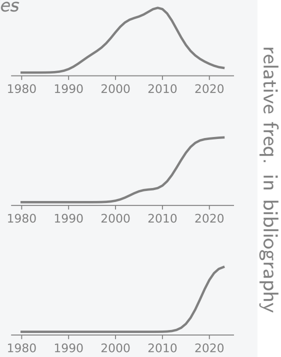


Table of contents

Part I : 📚 **Existing methods**

1. 🔵 Energy Based Models
2. 🟠 Point Process for object detection
3. 🟢 CNN for object detection

Part II: 💡 **Proposed approach**

4. 🟣 Energy model
5. 🟣 Sampling
6. 🟣 Parameters estimation
7. ⚪ Applications

📚 Existing methods

01

1. ● Energy Based Models

2. ● Point Process for object detection

3. ● CNN for object detection

Energy Based Models

Encoding dependencies as scalar energies [2]

- $\mathbf{X}$: observation, $\mathbf{y}$: to be predicted, $U(\mathbf{y},\mathbf{X}) \in \mathbb{R}$: *compatibility*
- *Most compatible* output: $\mathbf{y}^* = \arg\min_{\mathbf{y}\in\mathcal{Y}} U(\mathbf{y},\mathbf{X})$
- Usage from prediction, ranking, detection to generative models

---

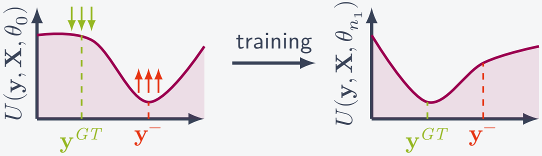
[2] LeCun *et al.* 2006.

Energy Based Models

Encoding dependencies as scalar energies [2]

- $\mathbf{X}$: observation, $\mathbf{y}$: to be predicted, $U(\mathbf{y}, \mathbf{X}) \in \mathbb{R}$: *compatibility*
- *Most compatible* output: $\mathbf{y}^* = \arg\min_{\mathbf{y} \in \mathcal{Y}} U(\mathbf{y}, \mathbf{X})$
- Usage from prediction, ranking, detection to generative models

Energy based training



---

[2] LeCun *et al.* 2006.

02

1. ● Energy Based Models

2. ● Point Process for object detection
   - What is a Point Process ?
   - Application to object detection

3. ● CNN for object detection

Point Process: definition

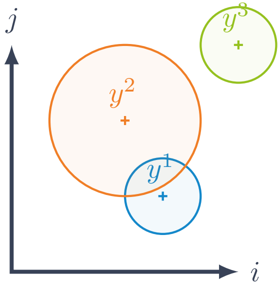
Marked point process

- **Configuration** of points $\mathbf{y} = \{y^1, \dots, y^n\}$
- $\mathbf{y} \in \bigcup_{n=0}^{\infty} \{ \underbrace{\{y^1, \dots, y^n\}}_{\mathcal{Y}_n}, y \in \mathcal{S} \times \mathcal{M} \}$
- $\mathcal{S}$ image space, $\mathcal{M}$ mark space
- $\mathbf{y}$: realization of a random variable in $\mathcal{Y}$

Circles ○

- $\mathcal{S} \subset \mathbb{R}^2$, $\mathcal{M} = \mathbb{R}^+$
- $y = (y_i, y_j, y_r)$

Configuration $\{\mathbf{y} = y^1, y^2, y^3\}$



	y_i	y_j	y_r	
y^1 : (	2	1	0.5	)
y^2 : (	1.5	2	1	)
y^3 : (	3	3	0.5	)
	$\mathcal{S}$		$\mathcal{M}$	

Point Process: definition

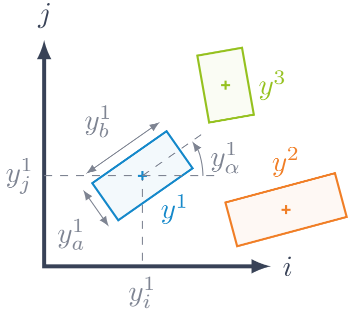
Marked point process

- Configuration of points $\mathbf{y} = \{y^1, \dots, y^n\}$
- $\mathbf{y} \in \bigcup_{n=0}^{\infty} \{ \underbrace{\{y^1, \dots, y^n\}}_{\mathcal{Y}_n}, y \in \mathcal{S} \times \mathcal{M} \}$
- $\mathcal{S}$ image space, $\mathcal{M}$ mark space
- $\mathbf{y}$: realization of a random variable in $\mathcal{Y}$

Oriented rectangles 🚗

- $\mathcal{S} \subset \mathbb{R}^2, \qquad \mathcal{M} = \mathbb{R}^+ \times \mathbb{R}^+ \times [0, \pi]$
- $y = (y_i, y_j, y_a, y_b, y_\alpha)$

Configuration $\{\mathbf{y} = y^1, y^2, y^3\}$



	y_i	y_j	y_a	y_b	y_α	
y^1 : (	1.3	1.2	0.6	1.2	0.6	)
y^2 : (	3.2.	0.7	0.6	1.5	0.3	)
y^3 : (	2.4	2.4	0.6	0.9	1.7	)

$\mathcal{S}$ $\qquad$ $\mathcal{M}$

Point Process: definition

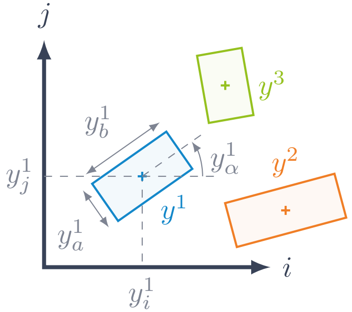
Marked point process

- Configuration of points $\mathbf{y} = \{y^1, \dots, y^n\}$
- $\mathbf{y} \in \bigcup_{n=0}^{\infty} \{\underbrace{\{y^1, \dots, y^n\}}_{\mathcal{Y}_n}, y \in \mathcal{S} \times \mathcal{M}\}$
- $\mathcal{S}$ image space, $\mathcal{M}$ mark space
- $\mathbf{y}$: realization of a random variable in $\mathcal{Y}$

Point process density (Gibbs)

$$h(\mathbf{y}) = \frac{1}{Z}\exp(-U(\mathbf{y}))$$

Configuration $\{\mathbf{y} = y^1, y^2, y^3\}$



	y_i	y_j	y_a	y_b	y_α
y^1 :	1.3	1.2	0.6	1.2	0.6
y^2 :	3.2.	0.7	0.6	1.5	0.3
y^3 :	2.4	2.4	0.6	0.9	1.7

$\mathcal{S}$ $\mathcal{M}$

Point Process: definition

Marked point process

- Configuration of points $\mathbf{y} = \{y^1, \dots, y^n\}$
- $\mathbf{y} \in \bigcup_{n=0}^{\infty} \{ \underbrace{\{y^1, \dots, y^n\}}_{\mathcal{Y}_n}, y \in \mathcal{S} \times \mathcal{M} \}$
- $\mathcal{S}$ image space, $\mathcal{M}$ mark space
- $\mathbf{y}$: realization of a random variable in $\mathcal{Y}$

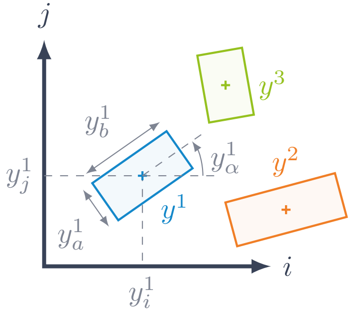
Point process density (Gibbs)

$$h(\mathbf{y}) = \frac{1}{\boxed{Z}} \exp(-U(\mathbf{y}))$$

⚠ Intractable normalizing constant $\in \mathbb{R}^+$

$$Z = \int_{\mathbf{y} \in \mathcal{Y}} \exp(-U(\mathbf{y}))\mu(d\mathbf{y})$$

Configuration $\{\mathbf{y} = y^1, y^2, y^3\}$

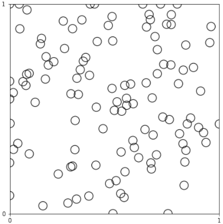


	y_i	y_j	y_a	y_b	y_α	
y^1 : (	1.3	1.2	0.6	1.2	0.6	)
				5	0.3	)
				9	1.7	)

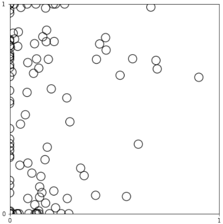
Energy composition

Total energy: sum of per-point energies

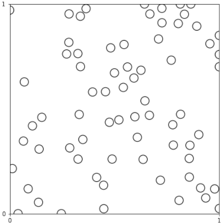
$$U(\mathbf{y}) = \sum_{y \in \mathbf{y}} V(y, \mathcal{N}^{\mathbf{y}}_{\{y\}})$$



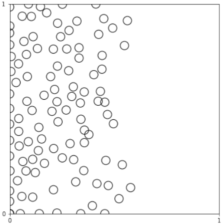
Uniform
$V_a(y) \propto 1$



Gradient
$V_b(y) \propto y_i$



No overlap
$V_c(y) \propto \max_{y' \in \mathcal{N}^{\mathbf{y}}_{\{y\}}} \mathbb{1}_{d(y,y')<r}(y)$



Gradient+no overlap
$V_d(y) \propto V_b + V_c$

Point process for object detection

Density & energy as function of the image $\mathbf{X}$

Point process density (Gibbs)

$$h(\mathbf{y}|\mathbf{X}) \propto \exp(-U(\mathbf{y}, \mathbf{X}, \theta))$$

Building the energy model

Given some annotated data $(\mathbf{X}, \mathbf{y}^{GT})$, we want U with parameters θ such that

$$\mathbf{y}^{GT} \simeq \arg\min_{\mathbf{y} \in \mathcal{Y}} U(\mathbf{y}, \mathbf{X}, \theta)$$

Point process for object detection

Density & energy as function of the image $\mathbf{X}$

Point process density (Gibbs)

$$h(\mathbf{y}|\mathbf{X}) \propto \exp(-U(\mathbf{y},$$

Building the energy model

Given some annotated data $(\mathbf{X}, \mathbf{y}^{GT})$, we want U with parameters θ such that

$$\mathbf{y}^{GT} \simeq \arg\min_{\mathbf{y}\in\mathcal{Y}} U(\mathbf{y}, \mathbf{X}, \theta)$$

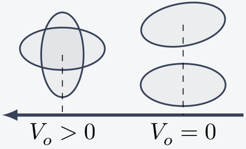
What a model needs:

1. Energy model
   $U : \mathcal{Y} \to \mathbb{R}$
2. Sampling procedure
   $\hat{\mathbf{y}} \simeq \arg\min_{\mathbf{y}\in\mathcal{Y}} U(\mathbf{y}, \mathbf{X}, \theta)$
3. Parameter estimation
   $\theta = F(\mathcal{D})$

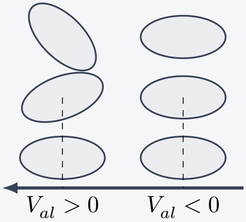
Classical energy model: prior

$$U(\mathbf{y}, \mathbf{X}) = \sum_{y \in \mathbf{y}} V_{data}(y, \mathbf{X}) + \boxed{V_{prior}(y, \mathcal{N}^{\mathbf{y}}_{\{y\}})}$$

Prior term



Overlapping



Alignment

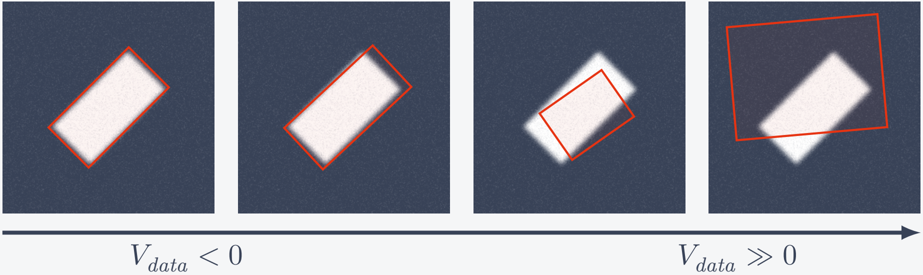
Or others:

- Shape
- Size
- Dynamics
- …

Classical energy model: data

$$U(\mathbf{y}, \mathbf{X}) = \sum_{y \in \mathbf{y}} \boxed{V_{data}(y, \mathbf{X})} + V_{prior}(y, \mathcal{N}^{\mathbf{y}}_{\{y\}})$$

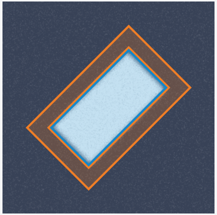
Data term



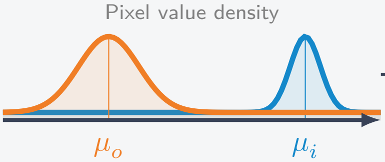
Classical energy model: data

$$U(\mathbf{y}, \mathbf{X}) = \sum_{y \in \mathbf{y}} \boxed{V_{data}(y, \mathbf{X})} + V_{prior}(y, \mathcal{N}^{\mathbf{y}}_{\{y\}})$$

Data term



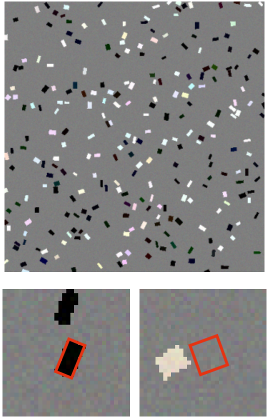
Data potential built from a contrast measure between the interior i and exterior o (e.g., T-test, KL, Image gradient...):



$$\xrightarrow{d(y_i, y_o)} V_{data}(y, \mathbf{X})$$

Contrast measures limitations

Synthetic data



✓object ✗not-object

DOTA sample

✓object ✗not-object

Average Precision (AP) on object/not-object classification task:

Measure	AP synth.	AP DOTA
T-test[a]	0.99	**0.20**
Image gradient[b]	0.99	**0.27**
CNN	0.88	0.99

---

[a] Lacoste *et al.* 2005.

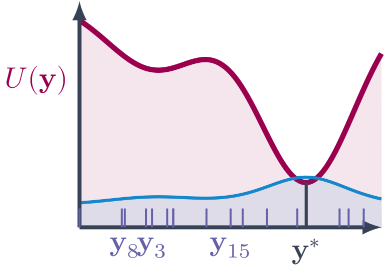
[b] Kulikova *et al.* 2011.

Classical sampling procedure

Looking for $\hat{\mathbf{y}} \simeq \mathbf{y}^* = \arg\min_{\mathbf{y}\in\mathcal{Y}} U(\mathbf{y}, \mathbf{X}, \theta)$

Reversible Jump MCMC [3]

- ▶ **Markov chain** $(\mathbf{y}_t)_{t>0}$, **stationary density**:
$$h(\mathbf{y})^{1/T_t} \propto \exp\left(-\frac{U(\cdot,\mathbf{X},\theta)}{T_t}\right)$$
- ↔ **Local transforms**: within $\mathcal{Y}_n$
- ↕ **Birth and Death**: $\mathcal{Y}_n \leftrightarrow \mathcal{Y}_{n+1}$



---

[3] Green 1995.

Classical sampling procedure

Looking for $\hat{\mathbf{y}} \simeq \mathbf{y}^* = \arg\min_{\mathbf{y}\in\mathcal{Y}} U(\mathbf{y}, \mathbf{X}, \theta)$

Reversible Jump MCMC [3]

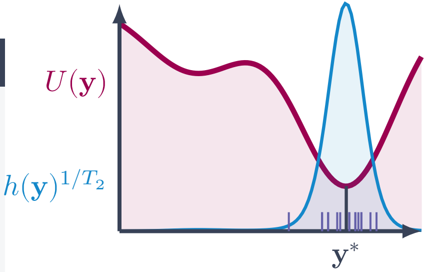
- **Markov chain** $(\mathbf{y}_t)_{t>0}$, **stationary density**:

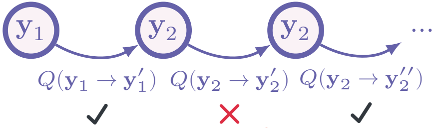
$$h(\mathbf{y})^{1/T_t} \propto \exp\left(-\frac{U(\cdot,\mathbf{X},\theta)}{T_t}\right)$$

- **Local transforms**: within $\mathcal{Y}_n$
- **Birth and Death**: $\mathcal{Y}_n \leftrightarrow \mathcal{Y}_{n+1}$
- **Perturbation** Q, Accepted (✓/✗) with proba:

$$\frac{Q(\mathbf{y}'\to\mathbf{y})}{Q(\mathbf{y}\to\mathbf{y}')} \exp\left(-\frac{\Delta U(\mathbf{y}\to\mathbf{y}')}{T_t}\right)$$

- **Simulated annealing**: $T_{t+1} = 0.99T_t$





---

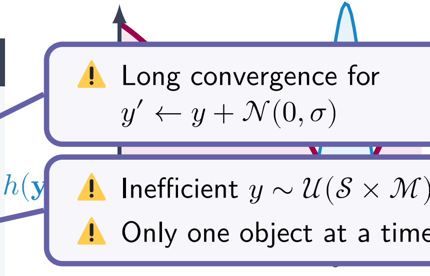
[3] Green 1995.

Classical sampling procedure

Looking for $\hat{\mathbf{y}} \simeq \mathbf{y}^* = \arg\min_{\mathbf{y}\in\mathcal{Y}} U(\mathbf{y}, \mathbf{X}, \theta)$

Reversible Jump MCMC [3]

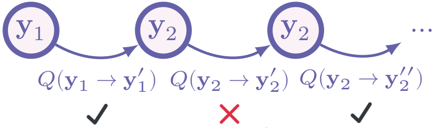
- **Markov chain** $(\mathbf{y}_t)_{t>0}$, **stationary density**:
$$h(\mathbf{y})^{1/T_t} \propto \exp\left(-\frac{U(\cdot,\mathbf{X},\theta)}{T_t}\right)$$
- **Local transforms**: within $\mathcal{Y}_n$
- **Birth and Death**: $\mathcal{Y}_n \leftrightarrow \mathcal{Y}_{n+1}$
- **Perturbation** Q, Accepted (✓/✗) with proba:
$$\frac{Q(\mathbf{y}'\to\mathbf{y})}{Q(\mathbf{y}\to\mathbf{y}')} \exp\left(-\frac{\Delta U(\mathbf{y}\to\mathbf{y}')}{T_t}\right)$$
- **Simulated annealing**: $T_{t+1} = 0.99 T_t$



⚠ Long convergence for $y' \leftarrow y + \mathcal{N}(0, \sigma)$

⚠ Inefficient $y \sim \mathcal{U}(\mathcal{S} \times \mathcal{M})$

⚠ Only one object at a time



---

[3] Green 1995.

Estimating weights w

$$U(\mathbf{y}) = \sum_{y \in \mathbf{y}} w_{data} V_{data}(y) + w_{pr1} V_{pr1}(y) + w_{pr2} V_{pr2}(y) + \ldots$$

Estimation with Linear Programming[4]

- **Negative** samples $\mathbf{y}^{-} \sim Q^{-}(\mathbf{y}^{GT} \rightarrow \cdot)$
  e.g. Birth + Death + transforms
- Get constraints $U(\mathbf{y}^{-}) \geq U(\mathbf{y}^{GT})$
- Solve Linear Programming problem

---

[4] Craciun *et al.* 2015.

Estimating weights w

$$U(\mathbf{y}) = \sum_{y\in\mathbf{y}} w_{data} \boxed{V_{data}(y)} + w_{pr1}V_{pr1}(y) + w_{pr2}V_{pr2}(y) + \dots$$

Estimation with Linear Programming [4]

- **Negative** samples $\mathbf{y}^{-} \sim \boxed{Q^{-}(\mathbf{y}^{GT} \rightarrow \cdot)}$
  e.g. Birth + Death + transforms
- Get constraints $\boxed{U(\mathbf{y}^{-}) \geq U(\mathbf{y}^{GT})}$
- Solve Linear Programming problem

⚠ No estimation of internal parameters (e.g. $V(y,\theta)$)

⚠ User-defined procedure influences the estimated w

⚠ Over-constraining (esp. on noisy GT)

---

[4] Craciun *et al.* 2015.

Point Process for object detection: summary

Point Processes : modelling configurations of points

- ✓ Easy addition of object interaction priors
- ✗ Contrast measures fail in complex settings
- ✗ Sampling is inefficient or requires application specific heuristics
- ✗ Limited parameter estimation method

03

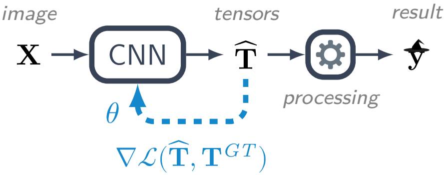
1. ● Energy Based Models
2. ● Point Process for object detection
3. ● CNN for object detection

Object detection with CNN

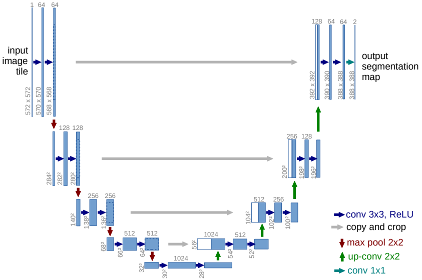
Convolutional Neural Network

- Convolution & pooling

  pattern matching & spatial aggregation
- Learning convolution filters with gradient descent



Unet[5]: simple and Fully Convolutional (FCN)



CenterNet[6]: using heatmaps to locate objects

keypoint heatmap [C] local offset [2] object size [2]

---

[5] Ronneberger *et al.* 2015.

[6] Zhou *et al.* 2019.

CNN for object detection: summary

CNN: efficient pattern extraction

- ✓ Efficient extraction of local image information
- ✓ Transforms images/pixels into new representations
- ✗ Hardly models object interaction
- ✗ Modeling object interactions requires more complexity: (e.g. Transformers → 📈 parameters)

💡 Proposed approach

Key contributions

Leveraging CNN and EBM methods into a Point Process framework

- 🟠 The PP framework allows for lightweight interaction models
- 🟢 Building data terms from simple CNN outputs.
- 🟢 Improved sampling based on CNN pre-computed energy maps and modern computation tools.
- 🔵 Bridging the gap of parameters estimation by introducing EBM training methods.

04

4. ● Energy model
  Generic energy model
  Data terms from a CNN
  Priors as energies
  Final energy model

5. ● Sampling

6. ● Parameters estimation

7. ● Applications

Generic Energy Model

Generic Energy Model

$$U\left(\mathbf{y},\mathbf{X},\theta\right) = \sum_{y\in\mathbf{y}} \left(\overbrace{\boxed{w_0}}^{\text{offset}} + \sum_{e\in\xi} \overbrace{\boxed{w_e}}^{\text{weight}} \overbrace{\boxed{V_e\left(y,\mathbf{X},\mathcal{N}^{\mathbf{y}}_{\{y\}},\theta\right)}}^{\text{energy term}} \right)$$

Energy terms $(e \in \xi)$

- **Data terms** $V_e(y,\mathbf{X})$: Built from CNN output
- **Prior terms** $V_e(y,\mathcal{N}^{\mathbf{y}}_{\{y\}})$: Multiple simple energies combined with w

Generic Energy Model

Generic Energy Model

$$U(\mathbf{y},\mathbf{X},\theta)=\sum_{y\in\mathbf{y}}\left(\overset{\text{offset}}{\boxed{w_0}}+\sum_{e\in\xi}\overset{\text{weight}}{\boxed{w_e}}\overset{\text{energy term}}{\boxed{V_e\left(y,\mathbf{X},\mathcal{N}^{\mathbf{y}}_{\{y\}},\theta\right)}}\right)$$

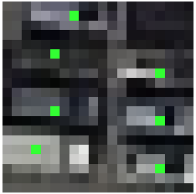
Parameters θ:

Weights: w_0, $\{w_e,\ e\in\xi\}$

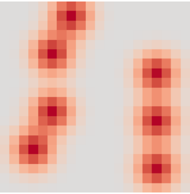
Energy terms $(e\in\xi)$

- **Data terms** $V_e(y,\mathbf{X})$: Built from CNN output
- **Prior terms** $V_e(y,\mathcal{N}^{\mathbf{y}}_{\{y\}})$: Multiple simple energies combined with w

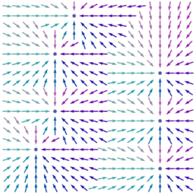
Potentials from a CNN



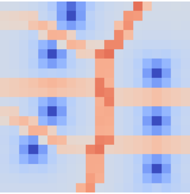
(a) object centers



(b) centers heatmap



(c) vector field



(d) divergence

1 Contrast measure on CNN output[7]

- ✗ Tuning of the contrast measure
- ✗ Connected blobs (b)

2 Divergence on vector field[8]

Image (a) $\xrightarrow[\text{CNN}]{}$ Vector field (c) $\xrightarrow[\text{Div()}]{}$ Energy map (d)

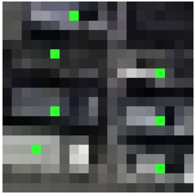
- ✓ High energy boundaries

---

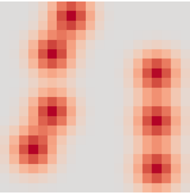
[7] Mabon *et al.* 2021.

[8] Mabon *et al.* 2022a.

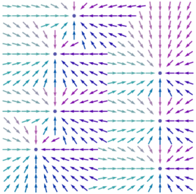
Potentials from a CNN



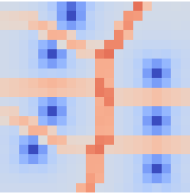
(a) object centers



(b) centers heatmap



(c) vector field



(d) divergence

1 Contrast measure on CNN output[7]

- ✗ Tuning of the contrast measure
- ✗ Connected blobs (b)

2 Divergence on vector field[8]

Image (a) $\xrightarrow{\text{CNN}}$ Vector field (c) $\xrightarrow{\text{Div()}}$ Energy map (d)

- ✓ High energy boundaries

Let's assume we already have a trained CNN

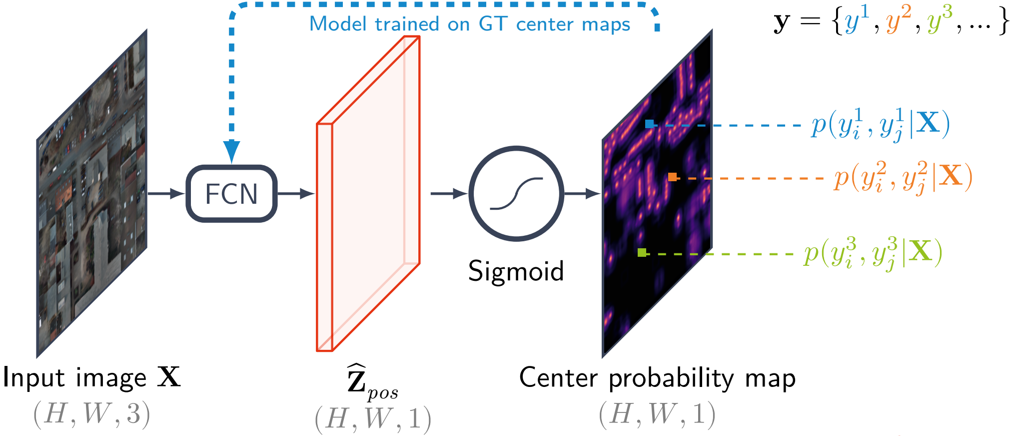
---

[7] Mabon *et al.* 2021.

[8] Mabon *et al.* 2022a.

Pretrained CNN for position energy

Reinterpreting trained CNN outputs[9]

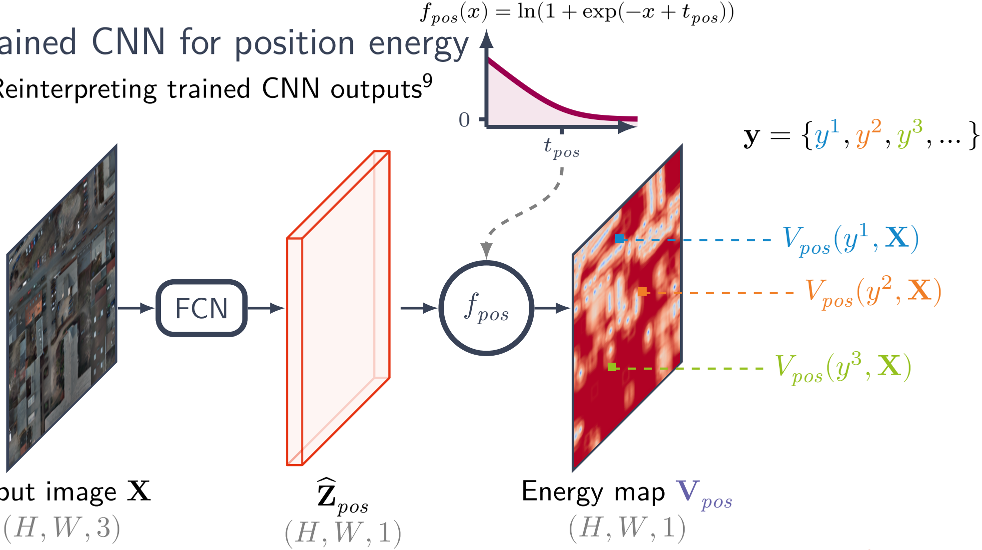


---

[9] Mabon *et al.* 2023a.

Pretrained CNN for position energy

Reinterpreting trained CNN outputs[9]

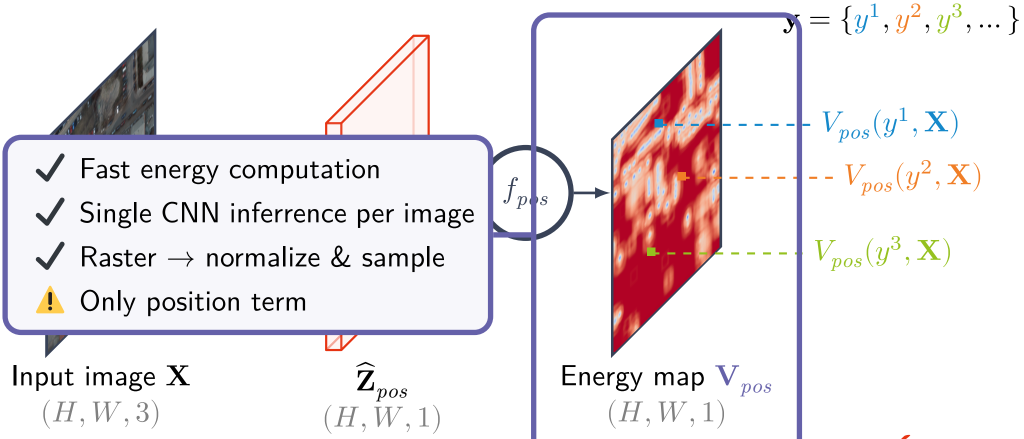


---

[9] Mabon *et al.* 2023a.

Pretrained CNN for position energy

Reinterpreting trained CNN outputs[9]



- ✓ Fast energy computation
- ✓ Single CNN inferrence per image
- ✓ Raster → normalize & sample
- ⚠ Only position term

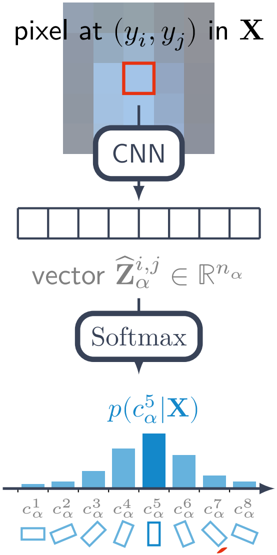
---

[9] Mabon *et al.* 2023a.

Mark energy terms

Classifier for discrete mark values

$$p\left(c|y_i, y_j, \mathbf{X}\right) = \operatorname{Softmax}_c(\widehat{\mathbf{Z}}_\alpha^{i,j}) = \frac{\exp\left(\widehat{\mathbf{Z}}_\alpha^{i,j}[c]\right)}{\sum_{c'=1}^{n_\alpha} \exp\left(\widehat{\mathbf{Z}}_\alpha^{i,j}[c']\right)}$$



Here, for mark $\kappa = \alpha$

---

[10] Grathwohl *et al.* 2019.

Mark energy terms

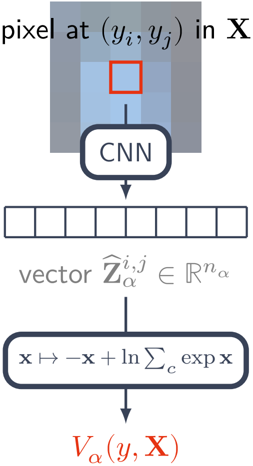
Classifier for discrete mark values

$$p\left(c|y_i,y_j,\mathbf{X}\right)=\mathrm{Softmax}_c(\widehat{\mathbf{Z}}_\alpha^{i,j})$$
$$=\frac{\exp\left(\widehat{\mathbf{Z}}_\alpha^{i,j}[c]\right)}{\sum_{c'=1}^{n_\alpha}\exp\left(\widehat{\mathbf{Z}}_\alpha^{i,j}[c']\right)}$$

Reformulating as energy [10]

$$V_\alpha\left(y,\mathbf{X}\right)=-\widehat{\mathbf{Z}}_\alpha^{i,j}[c_\alpha(y_\alpha)]+\ln\sum_{c=1}^{n_\alpha}\exp\widehat{\mathbf{Z}}_\alpha^{i,j}[c]$$

Here, for mark $\kappa=\alpha$

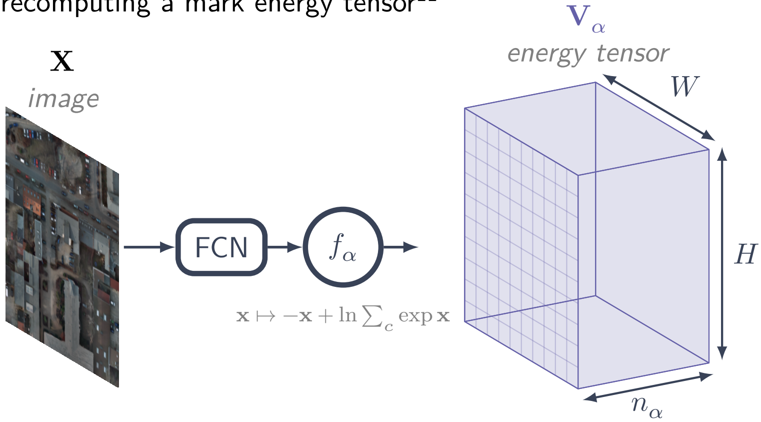


---

[10] Grathwohl *et al.* 2019.

Marks energy term: energy tensor

Precomputing a mark energy tensor[11]

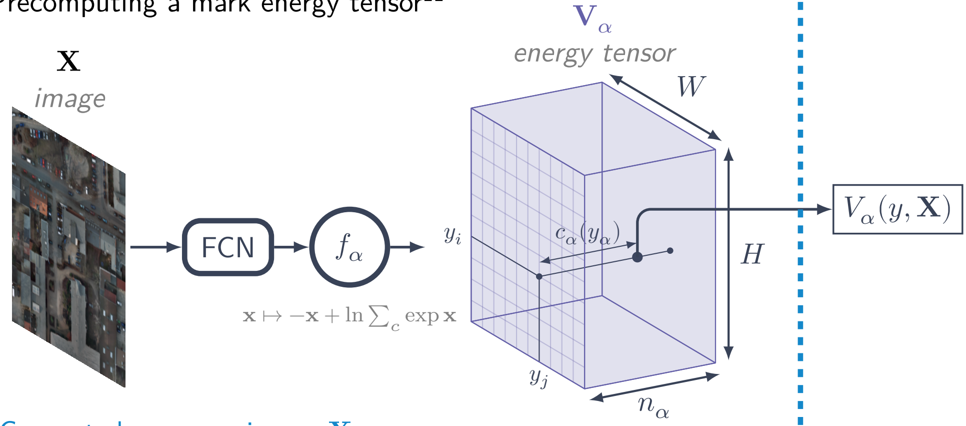


---

[11] Mabon *et al.* 2023a.

Marks energy term: energy tensor

Precomputing a mark energy tensor[11]

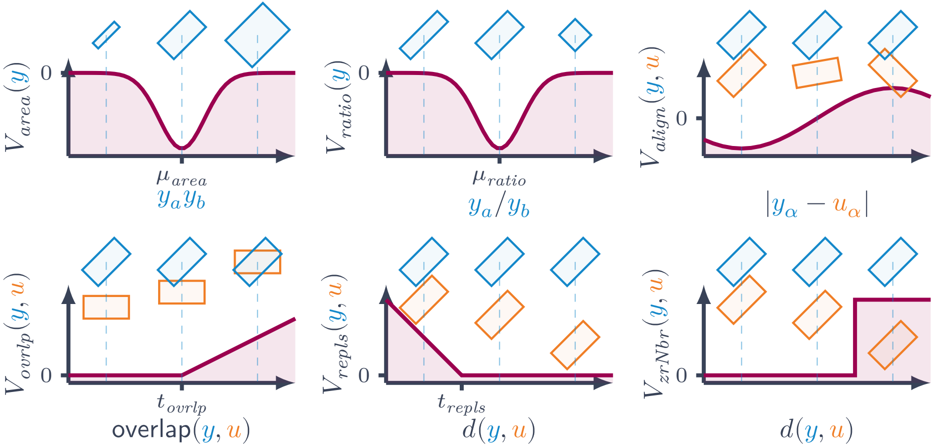


Computed once per image $\mathbf{X}$

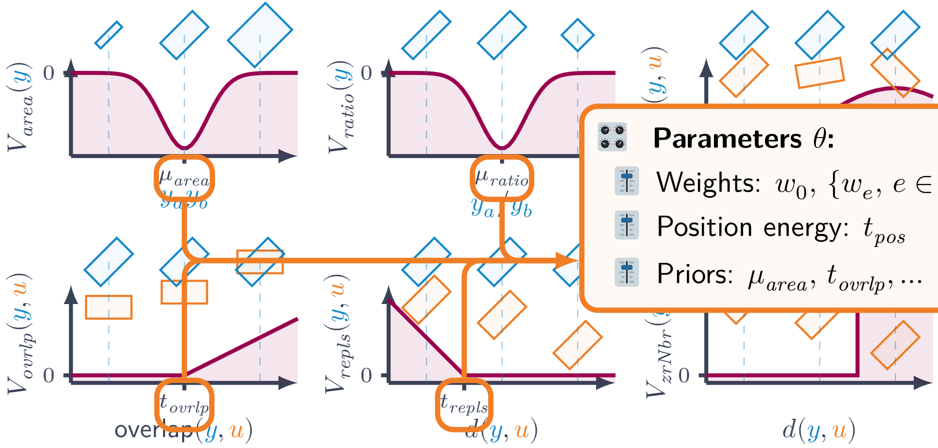
---

[11] Mabon *et al.* 2023a.

Priors on configurations



Priors on configurations



Parameters θ:

- Weights: $w_0, \{w_e, e \in \xi\}$
- Position energy: t_{pos}
- Priors: $\mu_{area}, t_{ovrlp}, \dots$

Energy model

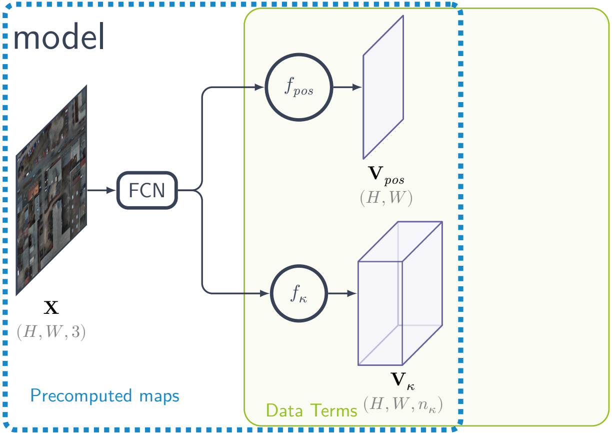
$\mathbf{X}$

$(H, W, 3)$

$U(\mathbf{y}, \mathbf{X})$

$\mathbf{y} = \{y^1, y^2, y^3\}$

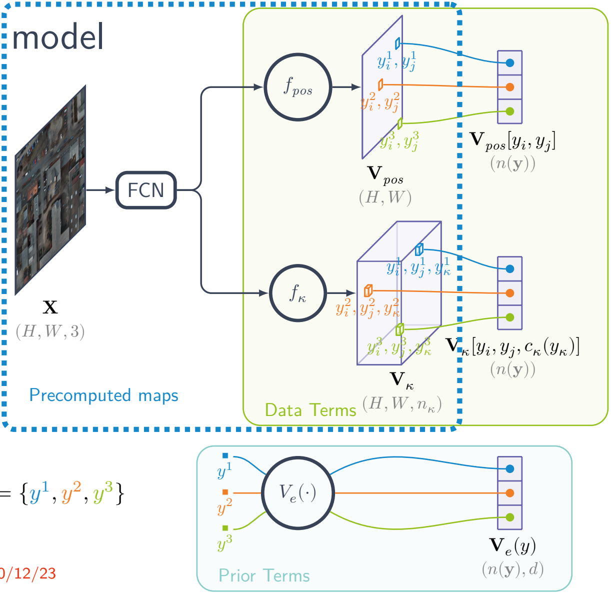
Energy model



$$U(\mathbf{y}, \mathbf{X})$$

$$\mathbf{y} = \{y^1, y^2, y^3\}$$

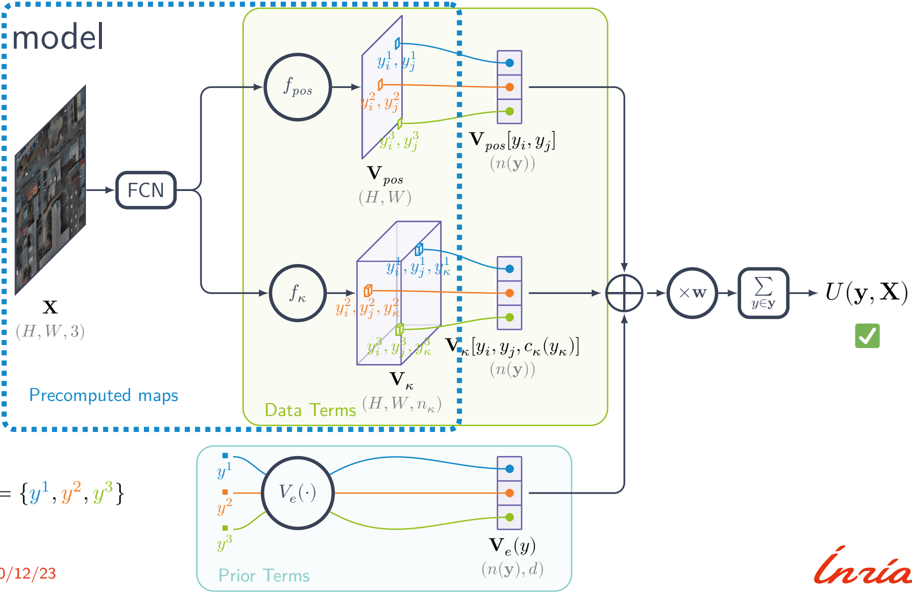
Energy model



$U(\mathbf{y}, \mathbf{X})$

$$\mathbf{y} = \{y^1, y^2, y^3\}$$

Energy model



$$\mathbf{y} = \{y^1, y^2, y^3\}$$

Energy model: summary

Integrating CNN into the the energy model

- Composition of simple priors into complex interaction models using energy weights w.
- Fast computation thanks to pre-computed energy maps.
- Pre-computed raster energy maps ready to use for sampling.
- Implemented as tensor computations (PyTorch, TensorFlow, JAX) :
  - Easy parallelization
  - Automatic differentiation

05

4. ● Energy model

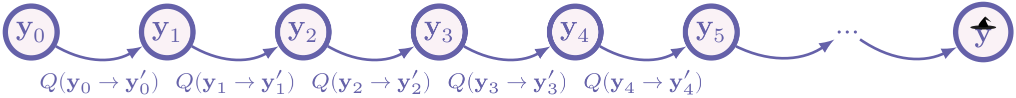
5. ● Sampling
    - Moves based on data
    - Parallel sampling

6. ● Parameters estimation

7. ● Applications

Sampling the model

$$\hat{\mathbf{y}} \simeq \arg\min_{\mathbf{y}\in\mathcal{Y}} U(\mathbf{y},\mathbf{X},\theta)$$



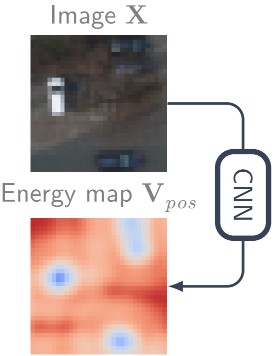
Towards better perturbations Q

- 🟢 Sampling moves from energy maps (from the CNN)
- 🔵 Diffusion on the whole energy model thanks to automatic differentiation
- 🟠 PP is Markovian → can be processed in parallel

Using energy maps for birth densities

Truncated energy as birth map [12]

- Approximating samples from $p(u|\mathbf{y}_t)$
- Energy maps can be normalized and sampled from



---

[12] Mabon *et al.* 2021.

Using energy maps for birth densities

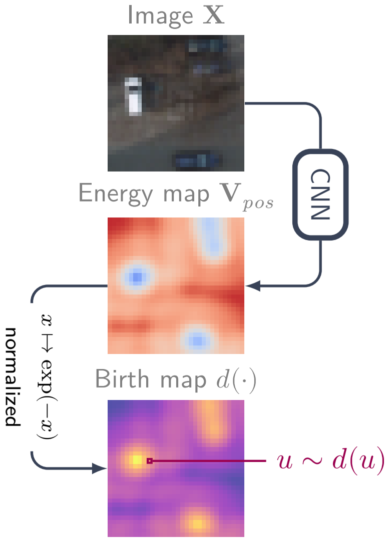
Truncated energy as birth map [12]

- Approximating samples from $p(u|\mathbf{y}_t)$
- Energy maps can be normalized and sampled from

Sampling using the raster energy maps

- Sample u in $\mathcal{S} \times \mathcal{M}$
  $u \sim \frac{1}{Z} \exp\left(-w_{pos}\mathbf{V}_{pos}[u]\right)$

Actually sampled in discrete space $\mathcal{S}_d \times \mathcal{M}_d$



---

[12] Mabon *et al.* 2021.

Jump diffusion

Jump Diffusion [13]

- ↕ Jump: Birth and Death moves, $\mathcal{Y}_n \leftrightarrow \mathcal{Y}_{n+1}$
- ↔ Diffusion / Langevin Dynamics, fixed $\mathcal{Y}_n$

Diffusion on the point process[14]

$$\mathbf{y} \leftarrow \mathbf{y} - \gamma \frac{\partial U(\mathbf{y}, \mathbf{X}, \theta)}{\partial \mathbf{y}} + dw\sqrt{2T_t},\ dw \sim \mathcal{N}(0, \gamma)$$

---

[13] Grenander and Miller 1994.

[14] Mabon *et al.* 2023a.

Jump diffusion

Jump Diffusion [13]

- ⬍ Jump: Birth and Death moves, $\mathcal{Y}_n \leftrightarrow \mathcal{Y}_{n+1}$
- ⬌ Diffusion / Langevin Dynamics, fixed $\mathcal{Y}_n$

Diffusion on the poi

$$\mathbf{y} \leftarrow \mathbf{y} - \gamma \overbrace{\boxed{\frac{\partial U(\mathbf{y}, \mathbf{X}, \theta)}{\partial \mathbf{y}}}}^{\text{energy gradient}} + \overbrace{\boxed{dw\sqrt{2T_t}}}^{\text{noise}}, \; dw \sim \mathcal{N}(0, \gamma)$$

---

[13] Grenander and Miller 1994.

[14] Mabon *et al.* 2023a.

Jump diffusion

Jump Diffusion [13]

- ⬍ Jump: Birth and Death moves, $\mathcal{Y}_n \leftrightarrow \mathcal{Y}_{n+1}$
- ⬌ Diffusion / Langevin Dynamics, fixed $\mathcal{Y}_n$

Diffusion on the point process[14]

$$\mathbf{y} \leftarrow \mathbf{y} - \gamma \boxed{\frac{\partial U(\mathbf{y}, \mathbf{X}, \theta)}{\partial \mathbf{y}}} - d$$

- ✓ Takes into account data and interaction terms
- ✓ Automatic differentiation
  no manual derivation

---

[13] Grenander and Miller 1994.

[14] Mabon *et al.* 2023a.

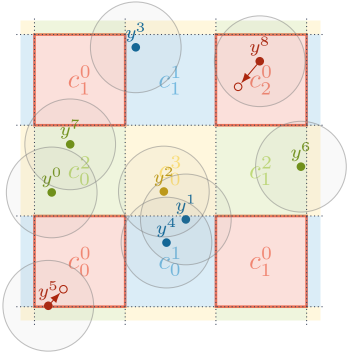
Sampling in parallel

PP Markovianity allows parallelization

Two perturbations distant enough can be done in parallel

Parallelization of perturbations Q

- Space $\mathcal{S}$ split into sets of mutually independent cells [15]
- We pick cells to simulate according to birth map $d(\cdot)$ [16]
- Parallelization is achieved as batched tensor computation[16]



---

[15] Verdié and Lafarge 2012.

[16] Mabon *et al.* 2023c.

Sampling: summary

Leveraging the proposed model for improved sampling

- Precomputed energy maps allows for efficient moves in the Markov chain
- Easy diffusion mechanisms enabled by modern automatic differentiation engines
- Implicit parallelization by defining the model as batched tensor operations guided by the precomputed energy maps

06

4. ● Energy model

5. ● Sampling

6. ● Parameters estimation
   Weights estimation with SVM
   Parameters estimation with Contrastive Divergence

7. ● Applications

Parameters estimation: introduction

Looking for θ, ideally such that $\forall(\mathbf{y}^{GT}, \mathbf{X}) \in \mathcal{D}$:

$$\mathbf{y}^{GT} \simeq \arg\min_{\mathbf{y}\in\mathcal{Y}} U(\mathbf{y}, \mathbf{X}, \theta)$$

Proposed methods

- SVM based method for weights estimation
- Contrastive Divergence for all parameters estimation

Parameters θ

Energy weights

- $w_0,\ \{w_e,\ e \in \xi\}$

"Internal" parameters

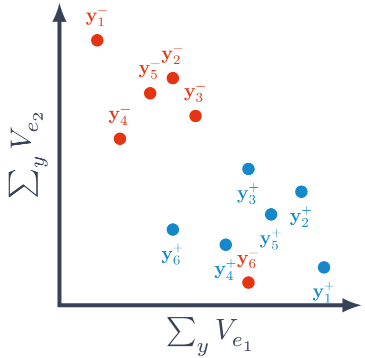
- t_{pos}
- $\mu_{area},\ t_{ovrlp}, \dots$

Estimating weights with Support Vector Machine

$$U(\mathbf{y}, \mathbf{X}, \theta) = \sum_{e \in \xi} w_e \sum_{y \in \mathbf{y}} V_e(y, \ldots) = \mathbf{w} \cdot \mathbf{v}_{\mathbf{y}}$$

Maximizing the energy margin

- **Positive** samples $\mathbf{y}^+ \sim \mathbf{y}^{GT} + \mathcal{N}(0, \sigma)$
  low σ, modeling uncertainty
- **Negative** samples $\mathbf{y}^- \sim Q^-(\mathbf{y}^{GT} \rightarrow \cdot)$
  e.g. Birth + Death + transforms

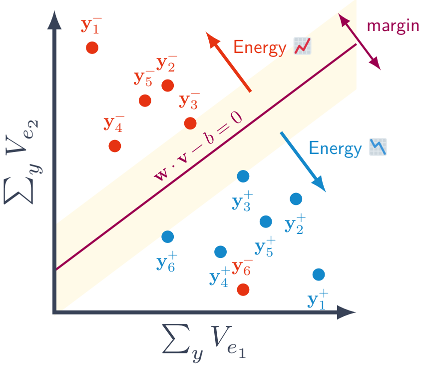


Estimating weights with Support Vector Machine

$$U(\mathbf{y},\mathbf{X},\theta) = \sum_{e\in\xi} w_e \sum_{y\in\mathbf{y}} V_e(y,\dots) = \mathbf{w}\cdot\mathbf{v}_{\mathbf{y}}$$

Maximizing the energy margin

- **Positive** samples $\mathbf{y}^{+} \sim \mathbf{y}^{GT} + \mathcal{N}(0,\sigma)$
  low σ, modeling uncertainty
- **Negative** samples $\mathbf{y}^{-} \sim Q^{-}(\mathbf{y}^{GT} \rightarrow \cdot)$
  e.g. Birth + Death + transforms
- Minimize Hinge loss:
  Compromise between max. margin and good labeling

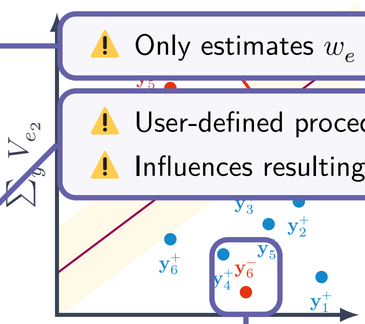


Estimating weights with Support Vector Machine

$$U(\mathbf{y},\mathbf{X},\theta) = \sum_{e\in\xi} \boxed{w_e} \sum_{y\in\mathbf{y}} V_e(y,\dots) = \mathbf{w}\cdot\mathbf{V}_{\mathbf{y}}$$

Maximizing the energy margin

- **Positive** samples $\mathbf{y}^+ \sim \mathbf{y}^{GT} + \mathcal{N}(0,\sigma)$
  low σ, modeling uncertainty
- **Negative** samples $\mathbf{y}^- \sim \boxed{Q^-(\mathbf{y}^{GT} \to \cdot)}$
  e.g. Birth + Death + transforms
- Minimize Hinge loss:
  Compromise between max. margin and good labeling



⚠ Only estimates w_e

⚠ User-defined procedure

⚠ Influences resulting $\mathbf{w}$

✓ Allows outliers

Estimating parameters with Contrastive Divergence

Maximizing likelihood

Estimate θ that maximizes likelihood over the data $\mathcal{D}$
Minimize : $\mathcal{L}_{nll}(\theta, \mathcal{D}) = -\log(P(\mathbf{y}_1^{GT}, \dots, \mathbf{y}_N^{GT} | X_1, \dots, X_N, \theta))$

---

[17] Hinton 2002.
[18] Bottou 2012.

Estimating parameters with Contrastive Divergence

Maximizing likelihood

Estimate θ that maximizes likelihood over the data $\mathcal{D}$
Minimize : $\mathcal{L}_{nll}(\theta, \mathcal{D}) = -\log(P(\mathbf{y}_1^{GT}, \ldots, \mathbf{y}_N^{GT} | X_1, \ldots, X_N, \theta))$

Constrative Divergence [17] (CD)

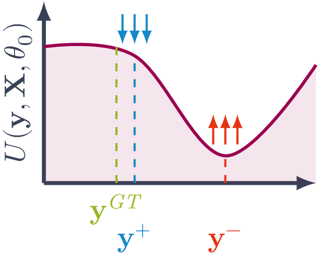
- Update θ_{n+1} with $\nabla \mathcal{L}$ using SGD[18]
- Minimize loss $\mathcal{L}(\theta_n, \mathbf{y}^+, \mathbf{y}^-, \mathbf{X}) = U(\mathbf{y}^+, \mathbf{X}, \theta_n) - U(\mathbf{y}^-, \mathbf{X}, \theta_n)$
- **positive** samples $\mathbf{y}^+ \sim \mathbf{y}^{GT} + \mathcal{N}(0, \sigma)$
- **negative** samples $\mathbf{y}^- \sim \exp(-U(\cdot, \mathbf{X}, \theta_n))$

---

[17] Hinton 2002.

[18] Bottou 2012.

Contrastive Divergence procedure



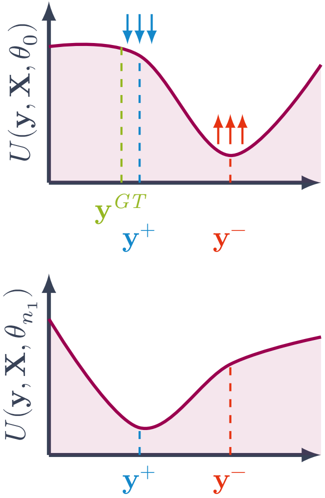
Procedure

1. initialize θ_0
2. For each element $(\mathbf{X},\mathbf{y}^{GT})\in\mathcal{D}$ *(or minibatch)*
   - 2.1 Sample $\mathbf{y}^{+}\sim\mathbf{y}^{GT}+\mathcal{N}(0,\sigma)$
   - 2.2 Sample $\mathbf{y}^{-}\sim\exp(-U(\cdot,\mathbf{X},\theta_n))$
   - 2.3 $\mathcal{L}=U(\mathbf{y}^{+},\mathbf{X},\theta_n)-U(\mathbf{y}^{-},\mathbf{X},\theta_n)$

Contrastive Divergence procedure

Procedure

1. initialize θ_0
2. For each element $(\mathbf{X}, \mathbf{y}^{GT}) \in \mathcal{D}$ *(or minibatch)*
    - 2.1 Sample $\mathbf{y}^+ \sim \mathbf{y}^{GT} + \mathcal{N}(0, \sigma)$
    - 2.2 Sample $\mathbf{y}^- \sim \exp(-U(\cdot, \mathbf{X}, \theta_n))$
    - 2.3 $\mathcal{L} = U(\mathbf{y}^+, \mathbf{X}, \theta_n) - U(\mathbf{y}^-, \mathbf{X}, \theta_n)$
    - 2.4 Update θ_n to θ_{n+1} from $\nabla_{\theta_n} \mathcal{L}$ with Stochastic Gradient Descent
3. Repeat from 2 until convergence



Contrastive Divergence: summary

Linear programming [19]

- ✗ **Constraints** $U(\mathbf{y}^{GT}) < U(\mathbf{y}^{-})$
- ✗ $\mathbf{y}^{-} \sim Q^{-}(\mathbf{y}^{GT} \to \cdot)$ **user defined** Q^{-}
- ✗ Estimates only energy term **weights**

Constrastive divergence [20]

- ✓ **Loss** $\mathcal{L} = U(\mathbf{y}^{+}) - U(\mathbf{y}^{-})$
- ✓ $\mathbf{y}^{-} \sim \exp(-U(\cdot, \mathbf{X}, \theta_n))$ using **current** θ_n
- ✓ **Also** estimates **non-linear parameters**
- ✓ Not limited to linear energy combination (e.g. $U(\mathbf{y}) = \mathrm{MLP}_\theta(\mathbf{v}_\mathbf{y})$)

---

[19] Craciun *et al.* 2015.

[20] Mabon *et al.* 2022a.

07

4. ● Energy model
5. ● Sampling
6. ● Parameters estimation
7. ● Applications

Results on remote sensing datasets

Data

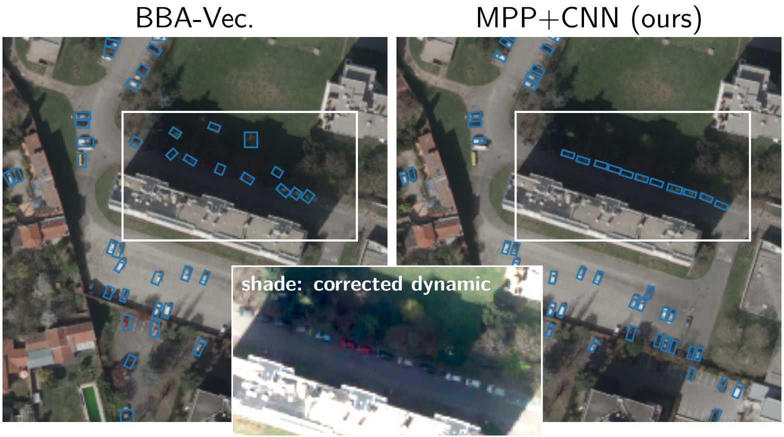
Images subsampled to 50 cm/pixel

- **Benchmarks**:
  - **DOTA**, (Xia *et al.* 2018) labeled with oriented rectangles, training dataset
  - **COWC**, (Mundhenk *et al.* 2016) labeled with centers
- **Airbus aerial images** (unlabeled) *matching CO3D sensors (🚀 2025)*

Models

- ***CNN-PP***$^{\blacklozenge}$/***CNN-PP***$^{\clubsuit}$: **manual**/**learned** weights θ
  trial and error/Contrastive Divergence
- ***CNN-LocalMax.***: CNN model with local maximum
- ***BBA-Vec.*** (Yi *et al.* 2021), ***YOLOV5-OBB*** (Yang and Yan 2022)

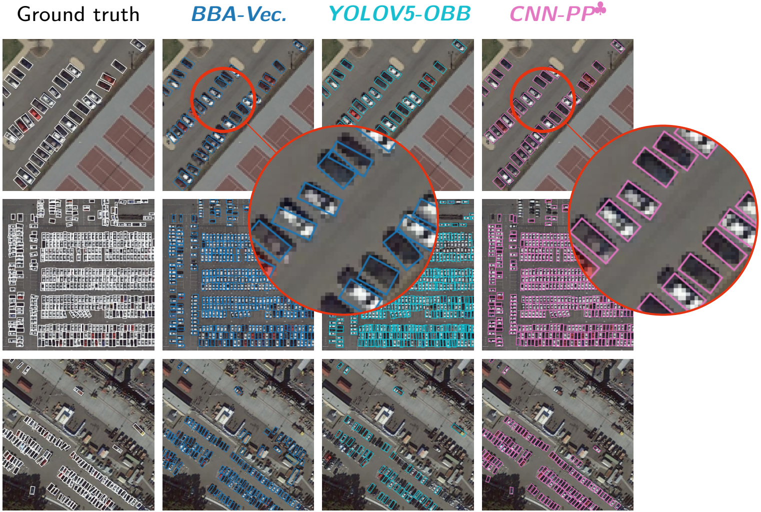
Airbus data, difficult example



BBA-Vec.

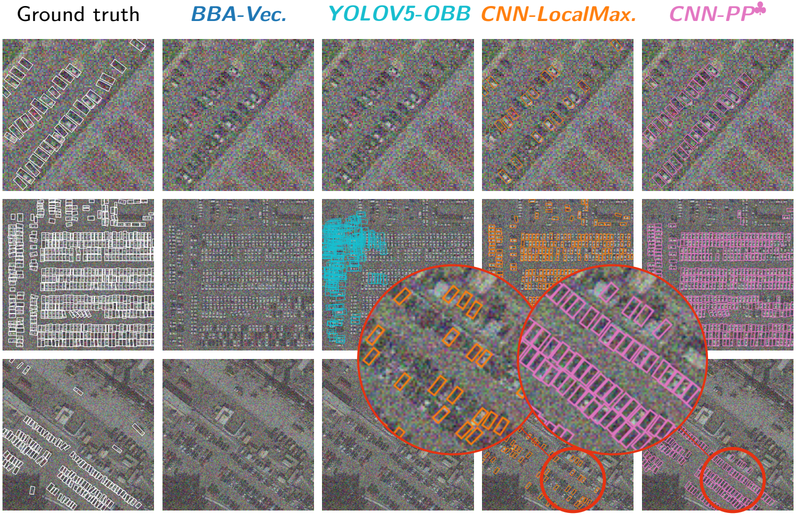
MPP+CNN (ours)

DOTA



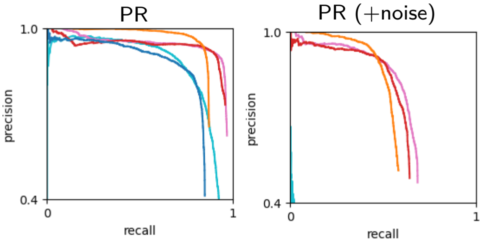
DOTA

DOTA
+Noise



DOTA : Metrics

Method	AP	AP_{+noise}
BBA-Vec.	0.82	0.19
YOLOV5-OBB	0.86	0.10
CNN-LocalMax.	0.86	0.55
CNN-PP♦	**0.91**	**0.58**
CNN-PP♣	**0.92**	**0.62**



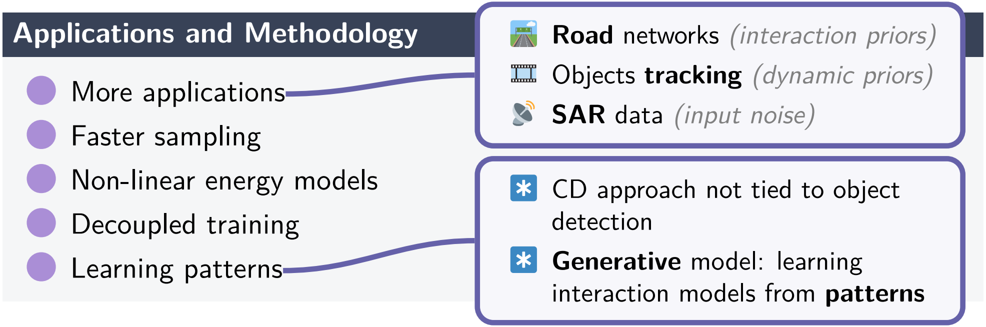
- For PP methods: Points $y \in \mathbf{y}$ are scored using Papangelou conditional intensity

Conclusion

At the crossroad between 🟠PP, 🟢CNN & 🔵EBM

- 🟠 The PP allows for explicit lightweight **interaction model**
- 🟢 Replacing contrast measures with **CNN data terms**
  - ✱ Efficient detection of **small objects** with limited computational **complexity**
- 🟢 Allowing to **improve sampling** methods
  - ✱ Birth map and **parallelism** guided by energy model & **Diffusion** dynamics
- 🔵 Bridging a gap in **parameters estimation**
  - ✱ Estimating **any** differentiable parameter with CD
- ⚪ **Lightweight** model with **performance** comparable to SOTA
  - ✱ **Regularized** configurations & **Robustness** to noise

Perspectives



Applications and Methodology

- More applications
- Faster sampling
- Non-linear energy models
- Decoupled training
- Learning patterns

🛣️ **Road** networks *(interaction priors)*
🎞️ Objects **tracking** *(dynamic priors)*
📡 **SAR** data *(input noise)*

✳️ CD approach not tied to object detection

✳️ **Generative** model: learning interaction models from **patterns**

Publications *(3 nat. / 3 intl. conf. | 1 jrn. to be submitted)*

- J. Mabon *et al.*, “Processus ponctuels et réseaux de neurones convolutifs pour la détection de véhicules dans des images de télédétection,” in *ORASIS 2021 - 18èmes Journées Francophones Des Jeunes Chercheurs En Vision Par Ordinateur*, Saint Ferréol, France: CNRS, Sep. 2021
- J. Mabon *et al.*, “CNN-Based Energy Learning for MPP Object Detection in Satellite Images,” in *2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP)*, Aug. 2022, pp. 1–6
- J. Mabon *et al.*, “Point process and CNN for small object detection in satellite images,” in *SPIE, Image and Signal Processing for Remote Sensing XXVIII*, Sep. 2022
- J. Mabon *et al.*, “Processus ponctuels marqués et réseaux de neurones convolutifs pour la détection d’objets dans des images de télédétection,” in *GRETSI 2022 - XXVIIIème Colloque Francophone de Traitement du Signal et des Images*, Nancy, France, Sep. 2022
- J. Mabon *et al.*, “Apprentissage contrastif de modèles de processus ponctuels pour la détection d’objets,” in *GRETSI 2023 - XXIXème Colloque Francophone de Traitement du Signal et des Images*, Grenoble, France, Aug. 2023
- J. Mabon *et al.*, “Learning point process models for vehicles detection using CNNs in satellite images,” in *17th International Conference on Signal-Image Technology &Internet-Based Systems (SITIS)*, Nov. 2023
- J. Mabon *et al.*, *Learning Point Processes and Convolutional Neural Networks for object detection in satellite images*, to be submitted to IEEE TGRS, Nov. 2023

Other Activities

Seminars and presentations

- Presentation at Inria **PhD seminars**, October 2021.
- Presentation at **journées du RT Geosto-MIA**, Rouen, September 2022.
- Presentation to the **Airbus Defense and Space** teams, Toulouse, September 2022.
- Presentation to the **CNES data Campus** team visiting Centre Inria d'Université Côte d'Azur, September 2022.

Other activities

- **Update** and maintenance of the **Ayana Team website** (2020-2023).
- Helping in **editing** the yearly **Ayana team activity report** (2020-2023).
- Organizing member (2021-2022) and **secretary** (2022-2023) of the **Association Doctorale du campus STIC** (ADSTIC).

🙏 Thank you !

Bibliography

Bibliography I

[1] G.-S. Xia *et al.*, “DOTA: A Large-Scale Dataset for Object Detection in Aerial Images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, Jun. 2018, pp. 3974–3983. arXiv: `1711.10398`.

[2] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. J. Huang, “A Tutorial on Energy-Based Learning,” *Predicting structured data*, p. 59, 2006.

[3] C. Lacoste, X. Descombes, and J. Zerubia, “Point processes for unsupervised line network extraction in remote sensing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1568–1579, Oct. 2005.

[4] M. Kulikova, I. Jermyn, X. Descombes, E. Zhizhina, and J. Zerubia, “A Marked Point Process Model Including Strong Prior Shape Information Applied to Multiple Object Extraction From Images.,” *International Journal of Computer Vision and Image Processing*, vol. 1, no. 2, pp. 1–12, Apr. 2011. (visited on 07/21/2023).

Bibliography II

[5] P. J. Green, “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination,” *Biometrika*, vol. 82, no. 4, pp. 711–732, Dec. 1995. (visited on 11/26/2020).

[6] P. Craciun, M. Ortner, and J. Zerubia, “Joint Detection and Tracking of Moving Objects Using Spatio-temporal Marked Point Processes,” in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, USA, Jan. 2015, pp. 177–184. (visited on 10/09/2023).

[7] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., ser. Lecture Notes in Computer Science, Munich, Germany, Oct. 2015, pp. 234–241.

[8] X. Zhou, D. Wang, and P. Krähenbühl, *Objects as Points*, Apr. 2019. arXiv: `1904.07850 [cs]`. (visited on 10/09/2023).

Bibliography III

[9] J. Mabon, M. Ortner, and J. Zerubia, “Processus ponctuels et réseaux de neurones convolutifs pour la détection de véhicules dans des images de télédétection,” in *ORASIS 2021 - 18èmes Journées Francophones Des Jeunes Chercheurs En Vision Par Ordinateur*, Saint Ferréol, France: CNRS, Sep. 2021.

[10] J. Mabon, M. Ortner, and J. Zerubia, “CNN-Based Energy Learning for MPP Object Detection in Satellite Images,” in *2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP)*, Aug. 2022, pp. 1–6.

[11] J. Mabon, M. Ortner, and J. Zerubia, “Apprentissage contrastif de modèles de processus ponctuels pour la détection d’objets,” in *GRETSI 2023 - XXIXème Colloque Francophone de Traitement du Signal et des Images*, Grenoble, France, Aug. 2023.

[12] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, “Your classifier is secretly an energy based model and you should treat it like one,” in *International Conference on Learning Representations (ICLR)*, virtual, Sep. 2019. (visited on 10/09/2023).

Bibliography IV

[13] U. Grenander and M. I. Miller, "Representations of Knowledge in Complex Systems," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 56, no. 4, pp. 549–581, 1994. (visited on 10/09/2023).

[14] Y. Verdié and F. Lafarge, "Efficient Monte Carlo sampler for detecting parametric objects in large scenes," in *Computer Vision – ECCV*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., Florence, Italy: Springer Berlin Heidelberg, Oct. 2012, pp. 539–552.

[15] J. Mabon, M. Ortner, and J. Zerubia, *Learning Point Processes and Convolutional Neural Networks for object detection in satellite images*, to be submitted to IEEE TGRS, Nov. 2023.

[16] G. Hinton, "Training Products of Experts by Minimizing Contrastive Divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002. (visited on 10/09/2023).

[17] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade: Second Edition*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds., Berlin, Heidelberg: Springer, 2012, pp. 421–436.

Bibliography V

[18] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, “A Large Contextual Dataset for Classification, Detection and Counting of Cars with Deep Learning,” in *Computer Vision – ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., ser. Lecture Notes in Computer Science, Amsterdam, The Netherlands: Springer International Publishing, Oct. 2016, pp. 785–800.

[19] J. Yi, P. Wu, B. Liu, Q. Huang, H. Qu, and D. Metaxas, “Oriented Object Detection in Aerial Images with Box Boundary-Aware Vectors,” in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, virtual, Jan. 2021, pp. 2149–2158. (visited on 10/09/2023).

[20] X. Yang and J. Yan, “On the Arbitrary-Oriented Object Detection: Classification Based Approaches Revisited,” *International Journal of Computer Vision*, vol. 130, no. 5, pp. 1340–1365, May 2022. (visited on 07/03/2023).

[21] J. Mabon, M. Ortner, and J. Zerubia, “Point process and CNN for small object detection in satellite images,” in *SPIE, Image and Signal Processing for Remote Sensing XXVIII*, Sep. 2022.

Bibliography VI

[22] ——,“Processus ponctuels marqués et réseaux de neurones convolutifs pour la détection d’objets dans des images de télédétection,” in *GRETSI 2022 - XXVIIIème Colloque Francophone de Traitement du Signal et des Images*, Nancy, France, Sep. 2022.

[23] ——,“Learning point process models for vehicles detection using CNNs in satellite images,” in *17th International Conference on Signal-Image Technology &Internet-Based Systems (SITIS)*, Nov. 2023.

Fin 👋 !