Deep Learning and Weakly Supervised Learning
Negative Evidence Models

Séminaire annuel laboratoire CRIStAL, thématique "Image"

Nicolas Thome - Conservatoire National des Arts et Métiers (Cnam)
CEDRIC Lab - Machine Learning Team (MSDMA)
5 Juillet 2018
Outline

1 Context: Big data & Deep Learning

2 Weakly Supervised Learning & Negative Evidence Models

3 Experiments

4 Conclusion
Context: Big Data

- Superabundance of data: images, videos, audio, text, user traces, etc

- Obvious need to access, search, or classify these data: Recognition

- Huge number of applications: mobile visual search, robotics, autonomous driving, augmented reality, medical imaging etc
Recognition of low-level signals: filling the semantic gap

- What we perceive vs
  What a computer sees

<table>
<thead>
<tr>
<th>243</th>
<th>239</th>
<th>240</th>
<th>225</th>
<th>206</th>
<th>185</th>
<th>188</th>
<th>218</th>
<th>211</th>
<th>206</th>
<th>216</th>
<th>225</th>
</tr>
</thead>
<tbody>
<tr>
<td>242</td>
<td>239</td>
<td>218</td>
<td>110</td>
<td>67</td>
<td>31</td>
<td>34</td>
<td>152</td>
<td>213</td>
<td>206</td>
<td>208</td>
<td>221</td>
</tr>
<tr>
<td>243</td>
<td>242</td>
<td>123</td>
<td>58</td>
<td>94</td>
<td>82</td>
<td>132</td>
<td>77</td>
<td>108</td>
<td>208</td>
<td>208</td>
<td>215</td>
</tr>
<tr>
<td>235</td>
<td>217</td>
<td>115</td>
<td>212</td>
<td>243</td>
<td>236</td>
<td>247</td>
<td>139</td>
<td>91</td>
<td>209</td>
<td>208</td>
<td>211</td>
</tr>
<tr>
<td>233</td>
<td>208</td>
<td>131</td>
<td>222</td>
<td>219</td>
<td>226</td>
<td>196</td>
<td>114</td>
<td>74</td>
<td>208</td>
<td>213</td>
<td>214</td>
</tr>
<tr>
<td>232</td>
<td>217</td>
<td>131</td>
<td>116</td>
<td>77</td>
<td>150</td>
<td>69</td>
<td>56</td>
<td>52</td>
<td>201</td>
<td>228</td>
<td>223</td>
</tr>
<tr>
<td>232</td>
<td>232</td>
<td>182</td>
<td>186</td>
<td>184</td>
<td>179</td>
<td>159</td>
<td>123</td>
<td>93</td>
<td>232</td>
<td>235</td>
<td>235</td>
</tr>
<tr>
<td>232</td>
<td>236</td>
<td>201</td>
<td>154</td>
<td>216</td>
<td>133</td>
<td>129</td>
<td>81</td>
<td>175</td>
<td>252</td>
<td>241</td>
<td>240</td>
</tr>
<tr>
<td>235</td>
<td>238</td>
<td>230</td>
<td>128</td>
<td>172</td>
<td>138</td>
<td>65</td>
<td>63</td>
<td>234</td>
<td>249</td>
<td>241</td>
<td>245</td>
</tr>
<tr>
<td>237</td>
<td>236</td>
<td>247</td>
<td>143</td>
<td>59</td>
<td>78</td>
<td>10</td>
<td>94</td>
<td>255</td>
<td>248</td>
<td>247</td>
<td>251</td>
</tr>
<tr>
<td>234</td>
<td>237</td>
<td>245</td>
<td>193</td>
<td>55</td>
<td>115</td>
<td>144</td>
<td>138</td>
<td>138</td>
<td>65</td>
<td>47</td>
<td>156</td>
</tr>
<tr>
<td>248</td>
<td>245</td>
<td>161</td>
<td>128</td>
<td>149</td>
<td>109</td>
<td>138</td>
<td>65</td>
<td>17</td>
<td>7</td>
<td>51</td>
<td>137</td>
</tr>
<tr>
<td>190</td>
<td>107</td>
<td>39</td>
<td>102</td>
<td>94</td>
<td>73</td>
<td>114</td>
<td>58</td>
<td>17</td>
<td>7</td>
<td>51</td>
<td>137</td>
</tr>
<tr>
<td>23</td>
<td>32</td>
<td>33</td>
<td>148</td>
<td>168</td>
<td>203</td>
<td>179</td>
<td>43</td>
<td>27</td>
<td>17</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>25</td>
<td>12</td>
<td>160</td>
<td>255</td>
<td>255</td>
<td>109</td>
<td>22</td>
<td>26</td>
<td>19</td>
<td>35</td>
<td>24</td>
</tr>
</tbody>
</table>
Recognition of low-level signals: input data variations

- Illumination variations
- View-point variations
- Deformable objects
- Intra-class variance
Deep Learning (DL) & Recognition of low-level signals

Before DL:

- **Handcrafted intermediate representations**
  - Needs expertise in each field
  - **Shallow archis**: low-level features

```
Image

x → Local Features (SIFT/ HoG) → h₁ → Coding / Pooling → h₂ → Classifier → y
cat

Handcrafted

speech

x → Local Features (MFCC) → h₁ → Coding / Pooling → h₂ → Classifier → y
phonem

Handcrafted
```
Deep Learning (DL) & Recognition of low-level signals

- **DL: learning intermediate representations**
  - ⊕ Deep: hierarchy, gradual learning
  - ⊕ Common learning methodology, no expertise

![Diagram of image and speech processing with learned intermediate representations](image)
Neural Networks (NN)

- **The formal Neuron**

  ![Diagram of a formal neuron](image)

  \[ y = f(w^T x + b) \]

  - \( x_i \): inputs
  - \( w_i, b \): weights
  - \( f \): activation function
  - \( y \): output of the neuron

  **Figure**: The formal neuron – Credits: R. Herault

- **Neural Networks**: Stacking several formal neurons \( \Rightarrow \) **Perceptron**

- **Soft-max Activation**:

  \[ \hat{y}_k = f(s_k) = \frac{e^{s_k}}{\sum_{k'=1}^{K} e^{s_{k'}}} \]

  \( \Rightarrow \) **Logistic Regression (LR) Model**
Deep Neural Networks (DNN)

- Logistic Regression (LR): limited to linear decision boundaries
- **Multi-Layer Perceptron (MLP):** Stacking layers of neural networks
  - More complex and rich functions
  - Neural network with one single hidden layer ⇒ universal approximator [Cyb89]

- Basis of the “deep learning” field
  - Hidden layers: intermediate representations from data
  - Can be learned with Backpropagation algorithm [Lec85, RHW86] (chain rule)
**Convolutional Neural Networks (ConvNets)**

- **ConvNets**: sparse connectivity + shared weights

- Overcome parameter explosion for Fully Connected Networks on images
- Local feature extraction (≠ FCN), equivariance
Convolutional Neural Networks (ConvNets)

- Convolution on tensors, *i.e.* multidimensional arrays: $T$ of size $W \times H \times D$
  - Convolution: $C[T] = T'$, $T'$ tensor of size $W' \times H' \times K$
  - Each filter locally connected with shared weights ($K$ number of filters)
- An elementary block: Convolution + Non linearity (*e.g.* ReLU) + pooling

- Stacking several Blocks: intuitive hierarchical information extraction
Deep Learning Success since 2010

- 2011: Speech Recognition

<table>
<thead>
<tr>
<th>Acoustic model</th>
<th>Recog WER</th>
<th>RT03S FSH</th>
<th>Hub5 SWB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional features</td>
<td>1-pass</td>
<td>27.4</td>
<td>23.6</td>
</tr>
</tbody>
</table>
| Deep Learning           | 1-pass    | **18.5**  | **16.1**  
|                         | -adapt    | (−33%)    | (−32%)    |
Deep Learning and ConvNet for Image Classification

- ImageNet ILSVRC Challenge (Stanford):
  - 1,200,000 training images, 1,000 classes, mono-label
  - Based on WordNet hierarchy (ontology)
  - Evaluation: top-5 error

- Up to 2012, leading approaches: BoW + SVM

- ILSVRC’12: the deep revolution ⇒ outstanding success of ConvNets [KSH12]

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Error Rate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U. Toronto</td>
<td>0.15315</td>
<td>Deep learning</td>
</tr>
<tr>
<td>2</td>
<td>U. Tokyo</td>
<td>0.26172</td>
<td>Hand-crafted features</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and learning models. Bottleneck.</td>
</tr>
<tr>
<td>3</td>
<td>U. Oxford</td>
<td>0.26979</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Xerox/INRIA</td>
<td>0.27058</td>
<td></td>
</tr>
</tbody>
</table>
2012: the deep revolution

Deep ConvNet success at ILSVRC’12

Two main practical reasons:

1. Huge number of labeled images ($10^6$ images)
   - Possible to train very large models without over-fitting
   - Larger models enables to learn rich (semantic) features hierarchies

2. GPU implementation for training
   - Relatively cheap and fast GPU
   - Training time reduced to 1-2 weeks (up to 50x speed up)
AlexNet [KSH12] in ILSVRC’12

- 60,000,000 parameters
- 650,000 neurons - 630,000,000 connections
- 5 convolutional layers, 3 Fully Connected (FC)
  - Convolution layer: Convolution + non linearity (ReLU) + pooling
  - Full= FC + non linearity - Final FC: 4096-dim
- Trained on 2 GPUs for a week
AlexNet [KSH12] in ILSVRC’12

Architecture of Krizhevsky et al.

- 8 layers total
- Trained on Imagenet dataset [Deng et al. CVPR’09]
- 18.2% top-5 error
- Our reimplementation: 18.1% top-5 error

Credit: R. Fergus
AlexNet [KSH12] in ILSVRC’12

Architecture of Krizhevsky et al.

- Remove top fully connected layer
  - Layer 7

- Drop 16 million parameters

- Only 1.1% drop in performance!

Credit: R. Fergus
AlexNet [KSH12] in ILSVRC’12

Architecture of Krizhevsky et al.

- Remove both fully connected layers
  - Layer 6 & 7

- Drop ~50 million parameters

- 5.7% drop in performance

Credit: R. Fergus
AlexNet [KSH12] in ILSVRC’12

Architecture of Krizhevsky et al.

- Now try removing upper feature extractor layers & fully connected:
  - Layers 3, 4, 6, 7

- Now only 4 layers

- 33.5% drop in performance

→ Depth of network is key

Credit: R. Fergus
Deep Learning in 2012: Representation Learning

Deep: more semantic features

Visualizations

Receptive fields

Layers

1

2

3

4

5

nicolas.thome@cnam.fr - Deep Learning & WSL
Outline

1. Context: Big data & Deep Learning

2. Weakly Supervised Learning & Negative Evidence Models
   - Pooling for Weakly Supervised Learning with ConvNets
   - Weakly Supervised Learning for Deep Structured Prediction

3. Experiments

4. Conclusion
ConvNet and invariance

- Standard ConvNets: limited invariance capacity (small shifts)
- ImageNet: single centered object ≠ other datasets (VOC, MS COCO)

⇒ How to use deep architectures on complex scenes?
How to use deep architectures on complex scenes?

- Learning localized representation
How to use deep architectures on complex scenes?

- Using full (precise) annotation, e.g. BB or segmentation masks
How to use deep architectures on complex scenes?

- Using full (precise) annotation, e.g. BB or segmentation masks
- **BUT:** full annotations expensive [BRFFF16] 
  ⇒ **training with weak supervision**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Notation</th>
<th>Space</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>$x$</td>
<td>$\mathcal{X}$</td>
<td>observed</td>
<td>observed</td>
</tr>
<tr>
<td>Output</td>
<td>$y$</td>
<td>$\mathcal{Y}$</td>
<td>observed</td>
<td>unobserved</td>
</tr>
<tr>
<td>Latent</td>
<td>$h$</td>
<td>$\mathcal{H}$</td>
<td>unobserved</td>
<td>unobserved</td>
</tr>
</tbody>
</table>
Outline

1. Context: Big data & Deep Learning

2. Weakly Supervised Learning & Negative Evidence Models
   Pooling for Weakly Supervised Learning with ConvNets
   Weakly Supervised Learning for Deep Structured Prediction

3. Experiments

4. Conclusion
Deep Architecture for Weakly Supervised Learning

- Adapt deep architecture: **Pooling function** ⇒ global label from local predictions

- \( h \times w \times C \) tensor: Class Activation Maps (CAM)
How to pool?

**Max** [Oquab, CVPR15]

$$y^c = \max_{i,j} z^c_{ij}$$

Use 1 region

**Average (GAP)** [Zhou, CVPR16]

$$y^c = \frac{1}{N} \sum_{i,j} z^c_{ij}$$

Use all regions
Average pooling limitation

- Classifying with all regions
- Not efficient for small objects: lots of “noisy” regions
Max pooling limitation

Max pooling

\[ y^c = \max_{i,j} z^c_{ij} \]  \hspace{1cm} (1)

- Classifying only with the max scoring region
- Loss of contextual information
Max pooling limitation

Max pooling

\[ y^c = \max_{i,j} z^c_{ij} \]  \hspace{2cm} (1)

- Classifying only with the max scoring region
- Loss of contextual information
**max+min pooling**

- **Contribution:** *max+min pooling function*

\[ y^c = \max_{i,j} z^c_{ij} + \min_{i,j} z^c_{ij} \]  

- **\( h^+ \):** presence of the class → high \( h^+ \)
- **\( h^- \):** localized evidence of the absence of class: **negative evidence**

street image \( x \)  

\[ s(\text{street}) = 2 \]  

\[ s(\text{highway}) = 0.7 \]
max+min pooling

- **Negative evidence**: OK pour \( h \Leftrightarrow \text{localization } x \) (MIL):
  - Text

Molecule, e.g. \( x \) DNA, \( h \) DNA region, \( y \) chemical property
  - \( h^- \) inhibition region in DNA for the chemical property
WELDON pooling

- Extension of max+min pooling
- Using several regions, more robust region selection

\[ y_c = s_{k^+}^{\text{top}}(z^c) + s_{k^-}^{\text{low}}(z^c) \]

\[
s_{k^+}^{\text{top}}(z^c) = \frac{1}{k^+} \sum_{i=1}^{k^+} \text{i-th-max}(z^c) \quad s_{k^-}^{\text{low}}(z^c) = \frac{1}{k^-} \sum_{i=1}^{k^-} \text{i-th-min}(z^c)
\]
WILDCAT pooling

- max+min pooling:
  - Both types of region are important
  - Complementary information
  - Not the same importance

- Pooling function

\[
y^c = s_{k^+}^{top}(z^c) + \alpha \cdot s_{k^-}^{low}(z^c)
\]

- \( \alpha \in [0, 1] \): trade off parameter

<table>
<thead>
<tr>
<th>Pooling</th>
<th>( k^+ )</th>
<th>( k^- )</th>
<th>( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GAP</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>max+min</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>WELDON</td>
<td>( k )</td>
<td>( k )</td>
<td>1</td>
</tr>
</tbody>
</table>
WILDCAT architecture

- WELDON: 1 model per class
  - Generalization to $M$ models per class
  - Catch multiple class-related modalities

$$z_{ij}^c = \sum_{m=1}^{M} z_{ij}^{cm}$$ (5)
Outline

1 Context: Big data & Deep Learning

2 Weakly Supervised Learning & Negative Evidence Models
   Pooling for Weakly Supervised Learning with ConvNets
   Weakly Supervised Learning for Deep Structured Prediction

3 Experiments

4 Conclusion
How to use deep architectures on complex scenes?

- **Structured Prediction**: use a structured loss on top of a deep ConvNet
- $\mathcal{X}$ arbitrary input space, $\mathcal{Y}$ discrete output space with **correlated variables** ⇒ **probabilistic graphical models**
- Ex: semantic segmentation ⇒ $\mathcal{Y} = \{1, ..., k\}^D$

- Various applications: NLP (PoS tagging), sequences (e.g. ADN), etc
Structured prediction

Structural SVM (SSVM) [TJHA05]

- \( \Psi(x, y) \in \mathbb{R}^d \): relationship between input \( x \in \mathcal{X} \) and output \( y \in \mathcal{Y} \)
- Scoring function linear in \( \Psi \): \( f_w(x, y) = \langle w, \Psi(x, y) \rangle = s(y) \)
- Prediction or inference: \( \hat{y}(x, w) = \arg\max_{y \in \mathcal{Y}} s(y) \)
  - Output space \( \mathcal{Y} \) generally huge \( \Rightarrow \) exhaustive maximization not tractable
  - Exploit structure (chain, tree), specific scoring functions (sub-modular), etc
- **Training**: a set of \( N \) labeled trained pairs \((x_i, y_i^*)\)
  - Structured loss \( \Delta(\hat{y}_i, y_i^*), \hat{y}_i(x_i, w) \Rightarrow Prior \) knowledge
  - Dependence of \( \Delta \) wrt \( w \) complex (non-convex, non-smooth)
  - **Margin rescaling**: convex upper bound \( \Delta(\hat{y}_i, y_i^*) \leq \ell(x_i, y_i^*, w) \)
    \[ \ell(x_i, y_i^*, w) = \max_{y \in \mathcal{Y}} \left[ \Delta(y_i^*, y) + s(y) \right] - s(y_i) \]
  - \( \tilde{y}_i = \arg\max_{y \in \mathcal{Y}} \left[ \Delta(y_i^*, y) + \langle w, \Psi(x_i, y) \rangle \right] \) "Loss Augmented Inference" (LAI)
    - For computing \( \frac{\partial \ell}{\partial w} = \Psi(x_i, \tilde{y}_i) - \Psi(x_i, y_i^*) \): generally harder than inference

nicolas.thome@cnam.fr - Deep Learning & WSL
Structured Output Ranking

- **Input** $x \in \mathcal{X}$ list of $n$ examples: $x = (d_1, \ldots, d_n)$, $\phi(d_i) \in \mathbb{R}^d$
- **Structured output** $y \in \mathcal{Y}$: ranking of example, represented by matrix $y$ s.t.
  $$y_{ij} = \begin{cases} +1 & \text{if } d_i \prec_y d_j (d_i \text{ is before } d_j \text{ in the sorted list}) \\ -1 & \text{if } d_i \succ_y d_j (d_i \text{ is after } d_j) \end{cases}$$
- **Ranking feature map**: $\Psi(x, y) = \frac{1}{N_+ \cdot N_-} \sum_{d_i \in \oplus} \sum_{d_j \in \ominus} y_{ij} [\phi(d_i) - \phi(d_j)]$, $y^*_{ij} = 1 \ \forall (i, j)$
- **Inference** ($|\mathcal{Y}| \sim 2^{n^2/2}$): exact by sorting example wrt $\langle w; \phi(d_i) \rangle$ [YFRJ07]
- **Training**: LAI with Average Precision (AP) loss: $\Delta_{AP}(y_i, y) = 1 - AP(y)$

- **AP**: Precision $= \frac{TP}{|P|}$ vs Recall $= \frac{TP}{N_+}$
- **$\Delta_{AP}$**: no linear decomposition wrt examples ≠ AUC ROC (TPR vs FPR)
  - Optimal greedy algorithm in $O(N_+ \cdot N_-)$ [YFRJ07], speed-up in [MJK14]
Structured prediction with latent variables

- **Latent Structural SVM (LSSVM)** [*YJ09*]
  - **Prediction:** \( s(y) = \max_{h \in \mathcal{H}} \langle w, \Psi(x_i, y, h) \rangle \Rightarrow \hat{y} = \arg \max_{y \in \mathcal{Y}} s(y) \)
  - **LAI for training:** \( \max_{(y, h) \in \mathcal{Y} \times \mathcal{H}} \left[ \Delta(y^*_i, y) + \langle w, \Psi(x, y, h) \rangle \right] \)

- **Structured AP ranking:** no exact solution LSSVM
  \( \Rightarrow \) Approximate solution in [*BMJK15*]

- **Negative Evidence Models**
  - **MANTRA Prediction:** \( s(y) = \max_{h \in \mathcal{H}} \langle w, \Psi(x_i, y, h) \rangle + \min_{h \in \mathcal{H}} \langle w, \Psi(x_i, y, h) \rangle \)
    - **WELDON:** \( k\text{-max} + k\text{-min} \)
  - **LAI for training:** \( \max_{y \in \mathcal{Y}} \left[ \Delta(y^*_i, y) + s(y) \right] \)

  - **Structured AP ranking:** exact solution!
  - **Symmetrization due to the** \( (k\text{-})\text{max} + (k\text{-})\text{min} \) **scoring**
  - **Decoupling optimization over** \( y \) **and** \( h \), \( \neq [YJ09, BMJK15] \)
WSL Ranking with Deep Negative Evidence Models

- $\Psi(x_i, y, h)$: feature representation for a given image region
- $s(y) = \max_{h \in \mathcal{H}} \langle w, \Psi(x_i, y, h) \rangle + \min_{h \in \mathcal{H}} \langle w, \Psi(x_i, y, h) \rangle$: score for a given output
  - **WELDON**: $k\text{-}\max+k\text{-}\min$
- Learning $\Psi(x_i, y, h)$ with deep ConvNet and AP loss: end-to-end training!
  - Incorporating multiple positive & negative evidence
1. Context: Big data & Deep Learning
2. Weakly Supervised Learning & Negative Evidence Models
3. Experiments
4. Conclusion
### Experimental Setup

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Train</th>
<th>#Test</th>
<th>#Classes</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOC 07</td>
<td>5,011</td>
<td>4,952</td>
<td>20</td>
<td>MAP</td>
</tr>
<tr>
<td>VOC 12</td>
<td>11,540</td>
<td>10,991</td>
<td>20</td>
<td>MAP</td>
</tr>
<tr>
<td>VOC 12 Action</td>
<td>2,296</td>
<td>2,292</td>
<td>10</td>
<td>MAP</td>
</tr>
<tr>
<td>MS COCO</td>
<td>82,783</td>
<td>40,504</td>
<td>80</td>
<td>MAP</td>
</tr>
<tr>
<td>MIT67</td>
<td>5,360</td>
<td>1,340</td>
<td>67</td>
<td>accuracy</td>
</tr>
<tr>
<td>CUB-200</td>
<td>5,994</td>
<td>5,794</td>
<td>200</td>
<td>accuracy</td>
</tr>
<tr>
<td>ILSVRC 2012</td>
<td>1,281,167</td>
<td>50,000</td>
<td>1000</td>
<td>accuracy</td>
</tr>
</tbody>
</table>

- Feature extraction network: ResNet-101 pretrained on ImageNet

nicolas.thome@cnam.fr - Deep Learning & WSL
## Classification Results

<table>
<thead>
<tr>
<th>Method</th>
<th>VOC 2007</th>
<th>VOC 2012</th>
<th>MS COCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-101</td>
<td>89.8</td>
<td>89.2</td>
<td>72.5</td>
</tr>
<tr>
<td>Deep MIL</td>
<td>-</td>
<td>86.3</td>
<td>62.8</td>
</tr>
<tr>
<td>ProNet</td>
<td>-</td>
<td>89.3</td>
<td>70.9</td>
</tr>
<tr>
<td>SPLeaP</td>
<td>88.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>WILDCAT</strong></td>
<td><strong>95.0</strong></td>
<td><strong>93.4</strong></td>
<td><strong>80.7</strong></td>
</tr>
</tbody>
</table>

### ImageNet Top-5 error

<table>
<thead>
<tr>
<th>ImageNet</th>
<th>Top-5 error</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-101 (1 crop)</td>
<td>6.21</td>
</tr>
<tr>
<td>ResNet-200 (10 crops)</td>
<td>4.93</td>
</tr>
<tr>
<td>ResNeXt-101 (1 crop)</td>
<td>4.4</td>
</tr>
<tr>
<td>Inception-ResNet-v2 (12 crops)</td>
<td><strong>4.1</strong></td>
</tr>
<tr>
<td><strong>WILDCAT (M = 1)</strong></td>
<td>4.23</td>
</tr>
</tbody>
</table>

nicolas.thome@cnam.fr - Deep Learning & WSL
AP Ranking Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>VOC07</th>
<th>VOCAct</th>
<th>MS</th>
<th>COCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>max + classif. loss</td>
<td>86.8</td>
<td>71.8</td>
<td>77.4</td>
<td></td>
</tr>
<tr>
<td>max + AP loss (LAPSVM [BMJK15])</td>
<td>87.9</td>
<td>73.3</td>
<td>77.9</td>
<td></td>
</tr>
<tr>
<td>max+min + classif. loss</td>
<td>89.9</td>
<td>78.5</td>
<td>77.7</td>
<td></td>
</tr>
<tr>
<td>max+min + AP loss</td>
<td><strong>91.2</strong></td>
<td><strong>80.7</strong></td>
<td>78.7</td>
<td></td>
</tr>
</tbody>
</table>

- Optimizing the evaluation metric during training is important
Pooling analysis

- max / LSSVM
- max+min / MANTRA
- k-max+k-min / WELDON
- average / GAP
- soft-max / LSE / HCRF

nicolas.thome@cnam.fr - Deep Learning & WSL
Pooling analysis

Unified pooling function

\[
s_w(\alpha, \beta^+_h, \beta^-_h)(x, y) = \frac{1}{2\beta^+_h} \log \left( \frac{1}{|\mathcal{H}|} \sum_{h \in \mathcal{H}} \exp[\beta^+_h \langle w, \Psi(x, y, h) \rangle] \right) + \alpha \frac{1}{2\beta^-_h} \log \left( \frac{1}{|\mathcal{H}|} \sum_{h \in \mathcal{H}} \exp[\beta^-_h \langle w, \Psi(x, y, h) \rangle] \right)
\]
Weakly Supervised Experiments
Weakly supervised localization

<table>
<thead>
<tr>
<th>Method</th>
<th>VOC 2012</th>
<th>MS COCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep MIL [Oquab, CVPR15]</td>
<td>74.5</td>
<td>41.2</td>
</tr>
<tr>
<td>ProNet [Sun, CVPR16]</td>
<td>77.7</td>
<td>46.4</td>
</tr>
<tr>
<td>WSLocalization [Bency, ECCV16]</td>
<td>79.7</td>
<td>49.2</td>
</tr>
<tr>
<td>WILDCAT</td>
<td>82.9</td>
<td>53.4</td>
</tr>
</tbody>
</table>

- Pointwise metric [Oquab, CVPR15]
Weakly supervised segmentation

- Test architecture

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIL-FCN</td>
<td>24.9</td>
</tr>
<tr>
<td>MIL-Base+ILP+SP-sppxl</td>
<td>36.6</td>
</tr>
<tr>
<td>EM-Adapt + FC-CRF</td>
<td>33.8</td>
</tr>
<tr>
<td>CCNN + FC-CRF</td>
<td>35.3</td>
</tr>
<tr>
<td>WILDCAT + FC-CRF</td>
<td><strong>43.7</strong></td>
</tr>
</tbody>
</table>

nicolas.thome@cnam.fr - Deep Learning & WSL
Weakly supervised segmentation
Outline

1 Context: Big data & Deep Learning

2 Weakly Supervised Learning & Negative Evidence Models

3 Experiments

4 Conclusion
Negative Evidence Models: Conclusion

- Local evidence of class absence
- State-of-the-art for many image classification datasets
- Applicable for weakly supervised localization & segmentation
- Application on different type of data: image, text, molecule
- **Structured output prediction:** AP ranking

```
true class
painted bunting
```

```
wrong class
indigo bunting
```
Resources

[1] Thibaut Durand, Nicolas Thome, and Matthieu Cord
MANTRA: Minimum Maximum Latent Structural SVM for Image Classification and Ranking.
In IEEE International Conference on Computer Vision (ICCV), 2015.

WELDON: Weakly Supervised Learning of Deep ConvNets.
In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

WILDCAT: Weakly Supervised Learning of Deep ConvNets for Image Classification, Pointwise Localization and Segmentation.
In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

In IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI), 2018.

Code available on GitHub:
- MANTRA: https://github.com/durandtibo/mantra-python
- WELDON: https://github.com/durandtibo/wsl.resnet.torch
- WILDCAT: https://github.com/durandtibo/wildcat.pytorch
Thank you for your attention!

Thibaut Durand  Nicolas Thome  Matthieu Cord

- Cnam Paris - CEDRIC Lab / MSDMA Team
- Sorbonne Université Associate member - LIP6 Lab / MLIA Team (P. Gallinari)

Questions?


