Accelerated Training of Linear Object Detectors

Charles Dubout
François Fleuret
Idiap Research Institute
EPFL

June 24, 2013
Plan

Exact Acceleration of Linear Object Detectors (ECCV 2012)
- Feature planes
- Standard convolution process
- Fourier convolution processes

Accelerated Training of Linear Object Detectors
- Linear object detector
- Fourier based gradient computation process
- Results
Feature planes

The image features can be seen as organized in planes, containing distinct features from each grid cell.
The computational cost to convolve a HOG image of size $M \times N$ with $L$ filters of size $P \times Q$ across $K$ features is:

$$C_{\text{std}} = O(KLMNPQ)$$
The computational cost to convolve a HOG image of size $M \times N$ with $L$ filters of size $P \times Q$ across $K$ features is:

$$C_{std} = \mathcal{O}(KLMNPQ)$$
Fourier based convolutions

The computational cost to convolve a HOG image of size $M \times N$ with $L$ filters of size $P \times Q$ across $K$ features is:
Fourier based convolutions

The computational cost to convolve a HOG image of size $M \times N$ with $L$ filters of size $P \times Q$ across $K$ features is:

$$C_{FFT} = O(KMN \log MN) + O(KLMN) + O(KLMN \log MN)$$

- **Forward FFTs**
- **Multiplications**
- **Inverse FFTs**
Fourier based convolutions

The computational cost to convolve a HOG image of size $M \times N$ with $L$ filters of size $P \times Q$ across $K$ features is:

$$C_{opt} = \mathcal{O}(KMN \log MN) + \mathcal{O}(KLMN) + \mathcal{O}(KLMN \log MN)$$

$$\approx \mathcal{O}(KLMN)$$
Linear object detector

- The simplest form of the detection score is:

\[ f_r(i, j) = \sum_{a, b} w(a, b) \Phi_r(i + a, j + b) \]

where \( \Phi_r(i, j) \) is the feature at location \((i, j)\) and \(w\) are the model weights.

- Can be extended to more complex models: mixtures, DPMs, etc.
Per sample loss

- We consider the data-driven term of training loss to be of the form

$$L(w) = \sum_r \sum_{i,j} l(y_r(i, j)f_r(i, j))$$

where $y_r(i, j) \in \{-1, 1\}$ is the label of the sub-window located at $(i, j)$ in image $r$, and $l$ is the per sample loss.

- Typical per sample losses are: hinge loss, logistic loss, exponential loss, etc.
Gradient of the loss

\[ \nabla L(a, b) = \frac{\partial L(w)}{\partial w(a, b)} \]  
(1)

\[ = \sum_r \sum_{i,j} \frac{\partial l(y_r(i, j)f_r(i, j))}{\partial w(a, b)} \]  
(2)

\[ = \sum_r \sum_{i,j} y_r(i, j) \frac{\partial l'(y_r(i, j)f_r(i, j))}{\partial w(a, b)} \frac{\partial f_r(i, j)}{\partial w(a, b)} \]  
(3)

\[ = \sum_r \sum_{i,j} (y_r \cdot l'(y_r \cdot f_r))(i, j) \Phi_r(a - i, b - j) \]  
(4)

\[ = \left( \sum_r y_r \cdot l'(y_r \cdot f_r) * \Phi_r \right)(a, b) \]  
(5)
Gradient of the loss

• Hence:

\[ \nabla L = \sum_r y_r \cdot l'(y_r \cdot f_r) \ast \Phi_r \]

where \( \overline{\alpha}(i, j) = \alpha(-i, -j) \) and the operator \( \cdot \) stands for the pointwise multiplication.

• The gradient can thus also be computed using the Fourier transform.
The computational cost to compute the gradient of $L$ filters of size $P \times Q$ over $R$ HOG images of size $M \times N$ with $K$ features is:

$$C_{\text{opt}} = O(RLNM \log(MN)) + O(KLRMN) + O(KLRMN \log(MN)) \approx O(KLRMN).$$
Fourier based gradient computation process

The computational cost to compute the gradient of $L$ filters of size $P \times Q$ over $R$ HOG images of size $M \times N$ with $K$ features is:

$$C_{\text{opt}} = \mathcal{O}(RLMN \log(MN)) + \mathcal{O}(KLRMN) + \mathcal{O}(KLRMN \log(MN))$$

$$\approx \mathcal{O}(KLRMN)$$
Results

We trained a linear detector on the Pascal VOC 2007 dataset using the *hinge* loss, and report the gradient computation time.

<table>
<thead>
<tr>
<th></th>
<th>1 scene per mini-batch</th>
<th>10 scenes per mini-batch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal filters</td>
<td>Large filters</td>
</tr>
<tr>
<td>Standard (ms)</td>
<td>41.3</td>
<td>70.9</td>
</tr>
<tr>
<td>Ours (ms)</td>
<td>7.2</td>
<td>7.4</td>
</tr>
<tr>
<td>Std. sparse (ms)</td>
<td>1.1</td>
<td>1.3</td>
</tr>
</tbody>
</table>

**Table**: Average time to compute the gradient of the loss for one stochastic gradient descent iteration.
Conclusion

+ Leverages the FT to make the overall training computational cost independent of the filters’ sizes

+ Relieves all the constraints inherent to sparse and approximate methods

- Contrary to our previous contribution, not very useful when using the hinge loss/bootstrapping
Thank you for your attention!

Questions?

Contact me at charles.dubout@idiap.ch