Latent SVMs for Human Detection with a Locally Affine Deformation Field

Ľubor Ladický\textsuperscript{1} Phil Torr\textsuperscript{2} Andrew Zisserman\textsuperscript{1}

\textsuperscript{1} University of Oxford \quad \textsuperscript{2} Oxford Brookes University
Human Detection

- Find all objects of interest
- Enclose them tightly in a bounding box
Human Detection

- Find all objects of interest
- Enclose them tightly in a bounding box
HOG Detector

- Sliding window using learnt HOG template
- Post-processing using non-maxima suppression

Dalal & Triggs CVPR05
• Sliding window using learnt HOG template
• Post-processing using non-maxima suppression
• Sliding window using learnt HOG template
• Post-processing using non-maxima suppression
HOG Detector

- Sliding window using learnt HOG template
- Post-processing using non-maxima suppression

Dalal & Triggs CVPR05
HOG Detector

- Sliding window using learnt HOG template
- Post-processing using non-maxima suppression

Dalal & Triggs CVPR05
HOG Detector

• Sliding window using learnt HOG template
• Post-processing using non-maxima suppression

Dalal & Triggs CVPR05
HOG Detector

- Sliding window using learnt HOG template
- Post-processing using non-maxima suppression
Does not fit well!

- Sliding window using learnt HOG template
- Post-processing using non-maxima suppression
• Allows parts to move relative to the centre
• Effectively allows the template to deform
• Multiple models based on an aspect ratio

Felzenszwalb et al. CVPR08
Deformable Part-based Model

- Allows parts to move relative to the centre
- Effectively allows the template to deform
- Multiple models based on an aspect ratio

Felzenszwalb et al. CVPR08
Comparison with other approaches

HOG template
(no deformation)

Part-based model
(rigid movable parts)

Our model
(deformation field)
Classifier response:

\[ H(c) = w^* h(c) + b^* = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sum_{l=1}^{m} w_{ijl}^* h_l(c_{ij}) + b^* \]
Classifier response:

\[ H(c) = w^* \cdot h(c) + b^* = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sum_{l=1}^{m} w_{ijl} h_l(c_{ij}) + b^* \]

The weights \( w^* \) and the bias \( b^* \) learnt using the Linear SVM as:

\[
(w^*, b^*) = \arg\min_{w, b} \lambda \|w\|^2 + \sum_{k=1}^{M} \xi_k \text{ hinge loss} \\
\text{s.t. } \forall k \in \{1, \ldots, M\} : \\
\xi_k \geq 0 \\
\xi_k \geq 1 - \zeta_k \left( w \cdot h(c^k) + b \right),
\]

ground truth labels
Cells $c$ displaced by the deformation field $d$:

$$D^{dij}(c_{ij}) = D^{dij}([x_{ij}, y_{ij}]) = [x_{ij} + d_{ij}^x, y_{ij} + d_{ij}^y]$$
Cells $c$ displaced by the deformation field $d$:

$$D^{d_{ij}}(c_{ij}) = D^{d_{ij}}([x_{ij}, y_{ij}]) = [x_{ij} + d_{ij}^x, y_{ij} + d_{ij}^y]$$

Classifier response:

$$H(c) = \max_d \left( {w^*} \cdot h(D^d(c)) + b^* - R(d) \right)$$

- HOG feature over the deformed template
- Regularization of the deformation field
Cells $c$ displaced by the deformation field $d$:

$$D^{d_{ij}}(c_{ij}) = D^{d_{ij}}([x_{ij}, y_{ij}]) = [x_{ij} + d_{ij}^x, y_{ij} + d_{ij}^y]$$

Classifier response:

$$H(c) = \max_{d} \left( w^* \cdot h(D^d(c)) + b^* - R(d) \right)$$

Regularisation takes the form of a smoothness MRF prior:

$$R(d) = \theta_p \sum_{(ij, fg) \in \mathcal{E}} \psi_p(d_{ij} - d_{fg})$$
Cells $c$ displaced by the deformation field $d$:

$$D_{dij}(c_{ij}) = D_{dij}([x_{ij}, y_{ij}]) = [x_{ij} + d_{ij}^x, y_{ij} + d_{ij}^y]$$

Classifier response:

$$H(c) = \max_d (w^* \cdot \mathbf{h}(D^d(c)) + b^* - R(d))$$

The weights $w^*$ and the bias $b^*$ learnt using the Latent Linear SVM as:

$$(w^*, b^*) = \arg \min_{(w,b)} \lambda ||w||^2 + \sum_{k=1}^{M} \xi_k$$

s.t. $\forall k \in \{1, \ldots, M\}$:

$$\xi_k \geq 0$$

$$\xi_k \geq 1 - z^k \max_d \left( w \cdot \mathbf{h}(D^d(c^k)) + b - R(d) \right)$$
Why hasn’t anyone tried it before?
Why hasn’t anyone tried it before?

• Latent models with many latent variables tend to over-fit
• Inference not feasible for a sliding window
To resolve these problems we propose:

• Flexible constraints on the deformation field which avoid over-fitting
• Feasible inference method under these constraints
• Clustering of the training data into multiple models
Locally Affine Deformation Field

We restrict the deformation field to be locally affine ($d \in A$):

$$\forall i,j \in \{1, .., n_x - 1\} \times \{1, .., n_y - 1\} : d_{i,j} + d_{i+1,j+1} = d_{i+1,j} + d_{i,j+1}$$
Locally Affine Deformation Field

We restrict the deformation field to be locally affine ($d \in A$):

$$\forall i,j \in \{1,..,n_x-1\} \times \{1,..,n_y-1\} : d_{i,j} + d_{i+1,j+1} = d_{i+1,j} + d_{i,j+1}$$

Non-deformed  Stretching in $x$  Stretching in $y$  Globally affine

Mapping $(x,y) \rightarrow (x+f(y),y)$  Mapping $(x,y) \rightarrow (x,y+f(x))$  Locally affine  Unconstrained
We restrict the deformation field to be locally affine ($d \in \mathcal{A}$):

$$\forall ij \in \{1,..,n_x-1\} \times \{1,..,n_y-1\} : d_{i,j} + d_{i+1,j+1} = d_{i+1,j} + d_{i,j+1}$$

- Non-deformed
- Stretching in $x$
- Stretching in $y$
- Globally affine
- Mapping $(x,y) \rightarrow (x + f(y), y)$
- Mapping $(x,y) \rightarrow (x, y + f(x))$
- Locally affine
- Unconstrained
We restrict the deformation field to be locally affine ($d \in A$): 

$$\forall i,j \in \{1,..,n_x - 1\} \times \{1,..,n_y - 1\} : d_{i,j} + d_{i+1,j+1} = d_{i+1,j} + d_{i,j+1}$$
We restrict the deformation field to be locally affine ($d \in \mathcal{A}$):

$$\forall i,j \in \{1,..,n_x-1\} \times \{1,..,n_y-1\} : d_{i,j} + d_{i+1,j+1} = d_{i+1,j} + d_{i,j+1}$$

- Non-deformed
- Stretching in $x$
- Stretching in $y$
- Globally affine

Mapping $(x,y) \rightarrow (x+f(y),y)$
Mapping $(x,y) \rightarrow (x,y+f(x))$
Locally affine
Unconstrained
We restrict the deformation field to be locally affine ($\mathbf{d} \in \mathcal{A}$):

$$\forall i, j \in \{1, \ldots, n_x - 1\} \times \{1, \ldots, n_y - 1\} : d_{i,j} + d_{i+1,j+1} = d_{i+1,j} + d_{i,j+1}$$

- Non-deformed
- Stretching in $x$
- Stretching in $y$
- Globally affine

Mapping \((x, y) \rightarrow (x + f(y), y)\)

Mapping \((x, y) \rightarrow (x, y + f(x))\)

Locally affine

Unconstrained
Locally Affine Deformation Field

We restrict the deformation field to be locally affine ($d \in A$):

$$\forall i, j \in \{1, \ldots, n_x - 1\} \times \{1, \ldots, n_y - 1\} : d_{i,j} + d_{i+1,j+1} = d_{i+1,j} + d_{i,j+1}$$

- Non-deformed
- Stretching in $x$
- Stretching in $y$
- Globally affine

Mapping $(x, y) \rightarrow (x + f(y), y)$
Mapping $(x, y) \rightarrow (x, y + f(x))$

Locally affine

Unconstrained
We restrict the deformation field to be locally affine ($d \in A$):

$$\forall i,j \in \{1, \ldots, n_x - 1\} \times \{1, \ldots, n_y - 1\} : d_{i,j} + d_{i+1,j+1} = d_{i+1,j} + d_{i,j+1}$$
We restrict the deformation field to be locally affine \((d \in A)\):

\[
\forall i, j \in \{1, \ldots, n_x - 1\} \times \{1, \ldots, n_y - 1\} : d_{i,j} + d_{i+1,j+1} = d_{i+1,j} + d_{i,j+1}
\]
We restrict the deformation field to be locally affine ($d \in A$):

$$\forall i, j \in \{1,.., n_x - 1\} \times \{1,.., n_y - 1\}: \ d_{i,j} + d_{i+1,j+1} = d_{i+1,j} + d_{i,j+1}$$
Weights / bias ($w^*, b^*$) and the deformation fields $d^k$ estimated iteratively
Weights / bias \((w^*, b^*)\) and the deformation fields \(d^k\) estimated iteratively

Given the deformation fields the problem is a standard linear SVM:

\[
(w^*, b^*) = \arg \min_{(w,b)} \lambda_w \|w\|^2 + \sum_{k=1}^{M} \xi^k
\]

s.t. \(\forall k \in \{1,..,M\} : \)

\[
\xi^k \geq 0
\]

\[
\xi^k \geq 1 - z^k \left( w \cdot h(D\hat{d}^k(c^k)) + b - R(\hat{d}^k) \right)
\]
Given \((w^*, b^*)\) the problem is a constrained MRF optimisation:

\[
\begin{align*}
d^{k*} &= \arg \max_{d^k \in \mathcal{A}} \left( \hat{w} \cdot h(D^{d^k}(c^k)) - R(d^k) \right) \\
&= \arg \min_{d^k \in \mathcal{A}} \sum_{(i,j) \in \mathcal{E}} \psi_p(d_{ij}^k - d_{fg}^k) - \hat{w} \cdot h(D^{d^k}(c^k))
\end{align*}
\]
Given \((w^*, b^*)\) the problem is a constrained MRF optimisation:

\[
d^{k*} = \arg \max_{d^k \in \mathcal{A}} \left( \hat{w} \cdot h(D^{d^k}(c^k)) - R(d^k) \right)
\]

\[
= \arg \min_{d^k \in \mathcal{A}} \sum_{(ij, fg) \in \mathcal{E}} \psi_p(d^k_{ij} - d^k_{fg}) - \hat{w} \cdot h(D^{d^k}(c^k))
\]

The last can be decomposed as:

\[
\hat{w} \cdot h(D^{d^k}(c^k)) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sum_{l=1}^{m} \hat{w}_{ijl} h_l(D^{d^k}_{ij}(c^k_{ij}))
\]
Given \((w^*, b^*)\) the problem is a constrained MRF optimisation:

\[
d^{k*} = \arg \max_{d^k \in A} \left( \hat{w} \cdot h(D^{d^k}(c^k)) - R(d^k) \right)
\]

\[
= \arg \min_{d^k \in A} \sum_{(ij, fg) \in E} \psi_p(d^k_{ij} - d^k_{fg}) - \hat{w} \cdot h(D^{d^k}(c^k))
\]

The last can be decomposed as:

\[
\hat{w} \cdot h(D^{d^k}(c^k)) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sum_{l=1}^{m} \hat{w}_{i,j,l} h_l(D^{d^k}_{ij}(c^k_{ij}))
\]

By defining \(\psi_u(d^k_{ij}) = -\sum_{l=1}^{m} \hat{w}_{i,j,l} h_l(D^{d^k}_{ij}(c^k_{ij}))\) the optimisation becomes:

\[
d^{k*} = \arg \min_{d^k \in A} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \psi_u(d^k_{ij}) + \sum_{(ij, fg) \in E} \psi_p(d^k_{ij} - d^k_{fg})
\]
Given \((w^*, b^*)\) the problem is a constrained MRF optimisation:

\[
d^{k*} = \arg \max_{d^k \in \mathcal{A}} \left( \hat{w} \cdot \text{h}(D^d(c^k)) - R(d^k) \right)
\]

\[
= \arg \min_{d^k \in \mathcal{A}} \sum_{(i,j,f,g) \in \mathcal{E}} \psi_p(d^k_{ij} - d^k_{fg}) - \hat{w} \cdot \text{h}(D^d(c^k))
\]

The last can be decomposed as:

\[
\hat{w} \cdot \text{h}(D^d(c^k)) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sum_{l=1}^{m} \hat{w}_{ijl} h_l(D^{d_{ij}}(c^k_{ij}))
\]

By defining \(\psi_u(d^k_{ij}) = -\sum_{l=1}^{m} \hat{w}_{ijl} h_l(D^{d_{ij}}(c^k_{ij}))\) the optimisation becomes:

\[
d^{k*} = \arg \min_{d^k \in \mathcal{A}} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \psi_u(d^k_{ij}) + \sum_{(i,j,f,g) \in \mathcal{E}} \psi_p(d^k_{ij} - d^k_{fg})
\]

Didn’t we make the problem harder?
• The location of the cells in the first row and in the first column fully determine the location of each cell

\[ d_{i+1,j+1} = d_{i+1,j} + d_{i,j+1} - d_{i,j} \]
• The location of the cells in the first row and in the first column fully determine the location of each cell

• Any locally affine deformation field can be reached by two moves:
  • move all columns where each column $i$ can move by $(\Delta^c d_i^x, \Delta^c d_i^y)$
  • move all rows where each row $j$ can move by $(\Delta^r d_j^x, \Delta^r d_j^y)$
Optimisation

Columns move

Non-deformed

Stretching in x

Stretching in y

Locally affine

Globally affine

Unconstrained

Mapping \((x, y) \rightarrow (x + f(y), y)\)

Mapping \((x, y) \rightarrow (x, y + f(x))\)
Optimisation

Non-deformed

Mapping \((x, y) \rightarrow (x + f(y), y)\)

Stretching in \(x\)

Mapping \((x, y) \rightarrow (x, y + f(x))\)

Stretching in \(y\)

Globally affine

Decruling

Locally affine

Unconstrained

Rows move
Optimisation

• The location of the cells in the first row and in the first column fully determine the location of each cell

• Any locally affine deformation field can be reached by two moves:
  • move all columns where each column $i$ can move by $(\Delta^c d_i^x, \Delta^c d_i^y)$
  • move all rows where each row $j$ can move by $(\Delta^r d_j^x, \Delta^r d_j^y)$

• These moves do not alter the local affinity

Columns move

\[
\begin{align*}
d_{i,j} + d_{i+1,j+1} &= d_{i+1,j} + d_{i,j+1} \\
\Delta^c d_i + \Delta^c d_{i+1} + \Delta^c d_{i+1} + \Delta^c d_i
\end{align*}
\]

Rows move

\[
\begin{align*}
d_{i,j} + d_{i+1,j+1} &= d_{i+1,j} + d_{i,j+1} \\
\Delta^r d_j + \Delta^r d_{j+1} + \Delta^r d_j + \Delta^r d_{j+1}
\end{align*}
\]
• The location of the cells in the first row and in the first column fully determine the location of each cell

• Any locally affine deformation field can be reached by two moves:
  • move all columns where each column $i$ can move by $(\Delta^c d_i^x, \Delta^c d_i^y)$
  • move all rows where each row $j$ can move by $(\Delta^r d_j^x, \Delta^r d_j^y)$

• These moves do not alter the local affinity

• Both moves can be solved quickly using dynamic programming
• The location of the cells in the first row and in the first column fully determine the location of each cell

• Any locally affine deformation field can be reached by two moves:
  • move all columns where each column $i$ can move by $(\Delta^c d_i^x, \Delta^c d_i^y)$
  • move all rows where each row $j$ can move by $(\Delta^r d_j^x, \Delta^r d_j^y)$

• These moves do not alter the local affinity

• Both moves can be solved quickly using dynamic programming

• Works for any form of pairwise potentials
• We define a similarity measure between two training samples as:

\[ S_{kp} = \max(S_{k \rightarrow p}^*, S_{\text{mirror}(k) \rightarrow p}^*) \]

where

\[ S_{k \rightarrow p}^* = \max_{d \in A} \left( h(c^p) \cdot h(D^d(c^k)) - R(d) \right) \]
Learning multiple poses / viewpoints

• We define a similarity measure between two training samples as:

\[ S_{kp} = \max\left( S_{k \rightarrow p}^*, S_{\text{mirror}(k) \rightarrow p}^* \right) \]

where

\[ S_{k \rightarrow p}^* = \max_{d \in A} \left( \mathbf{h}(c^p) \cdot \mathbf{h}(D^d(c^k)) - R(d) \right) \]

• K-medoid clustering of S matrix clusters the data into multi models
Experiments

• Buffy dataset (typically used for pose estimation)
  • Contains large variety of poses, viewpoints and aspect ratios
  • Consists of 748 images
    • Episode s5e3 used for training
    • Episode s5e4 used for validation
    • Episodes s5e2, s5e5 and s5e6 used for testing

Ferrari et al. CVPR08
Each row corresponds to one model (out of 10 models)
Qualitative results
Qualitative results
### Quantitative results

<table>
<thead>
<tr>
<th>Method</th>
<th>AP</th>
<th>Training time</th>
<th>Test time</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOG detector</td>
<td>47.65%</td>
<td>1h</td>
<td>1.4s</td>
</tr>
<tr>
<td>Part-based detector</td>
<td>72.38%</td>
<td>19.5h</td>
<td>6s</td>
</tr>
<tr>
<td>HOG detector with K-Medoid Clustering</td>
<td>69.10%</td>
<td>1.5h</td>
<td>4.7s</td>
</tr>
<tr>
<td>HOG detector with LADF Clustering</td>
<td>73.78%</td>
<td>2.5h</td>
<td>4.7s</td>
</tr>
<tr>
<td>LADF Detector</td>
<td>76.03%</td>
<td>2.5h</td>
<td>50s</td>
</tr>
</tbody>
</table>

![Precision-Recall Curve](chart.png)

- **HOG Detector**
- **Part-based Detector**
- **HOG Detector with K-Medoid Clustering**
- **HOG Detector with LADF Clustering**
- **LADF Detector**
Conclusion and Further Work

- We propose
  - Novel inference for locally affine deformation field (LADF)
  - Object detector using LADF
  - Clustering using LADF
- Further work
  - Explore usability for other vision problems (tracking, flow)
  - Explore generalisations of LADF where the same inference method is applicable