Deformable Part Models (DPM)
Felzenswalb, Girshick, McAllester & Ramanan (2010)
Slides drawn from a tutorial By R. Girshick

Part 1: modeling

Part 2: learning

Person detection performance on PASCAL VOC 2007
The Dalal & Triggs detector

Image pyramid
The Dalal & Triggs detector

- Compute HOG of the whole image at multiple resolutions
The Dalal & Triggs detector

- Compute HOG of the whole image at multiple resolutions
- Score every window of the feature pyramid

How much does the window at \( p \) look like a pedestrian?
The Dalal & Triggs detector

- Compute HOG of the whole image at multiple resolutions
- Score every window of the feature pyramid
- Apply non-maximal suppression

\[ \text{score}(l, p) = w \cdot \phi(l, p) \]
Detection

number of locations $p \sim 250,000$ per image
Detection

number of locations \( p \sim 250,000 \) per image

test set has \( \sim 5000 \) images

\( \gg 1.3 \times 10^9 \) windows to classify
Detection

number of locations $p \sim 250,000$ per image

test set has $\sim 5000$ images

$>> 1.3 \times 10^9$ windows to classify

typically only $\sim 1,000$ true positive locations
Detection

number of locations \( p \sim 250,000 \) per image

test set has \( \sim 5000 \) images

\( \gg 1.3 \times 10^9 \) windows to classify

typically only \( \sim 1,000 \) true positive locations

Extremely unbalanced binary classification
Detection

number of locations $p \sim 250,000$ per image

test set has $\sim 5000$ images

$\gg 1.3 \times 10^9$ windows to classify

typically only $\sim 1,000$ true positive locations

Learn $w$ as a Support Vector Machine (SVM)

$w$

Extremely unbalanced binary classification
Dalal & Triggs detector on INRIA pedestrians

- AP = 75%
- Very good
- Declare victory and go home?
Dalal & Triggs on PASCAL VOC 2007

AP = 12%
(using my implementation)
How can we do better?

Revisit an old idea: part-based models
“pictorial structures”

Fischler & Elschlager ’73
Felzenszwalb & Huttenlocher ’00
- Pictorial structures
- Weak appearance models
- Non-discriminative training

Combine with modern features and machine learning
Port the success of Dalal & Triggs into a part-based model
Example DPM (most basic version)

Root filter
Part filters
Deformation costs
Recall the Dalal & Triggs detector

- HOG feature pyramid
- Linear filter / sliding-window detector
- SVM training to learn parameters $w$

$$score(l, p) = w \cdot \phi(l, p)$$
DPM = D&T + parts

- Add parts to the Dalal & Triggs detector
  - HOG features
  - Linear filters / sliding-window detector
  - Discriminative training

Image pyramid

HOG feature pyramid

\[ \text{root} \]

\[ p_0 \]

\[ z \]

[FMR CVPR’08] [FGMR PAMI’10]
Sliding window detection with DPM

\[ z = (p_1, \ldots, p_n) \]

\[ \text{score}(l, p_0) = \max_{p_1, \ldots, p_n} \sum_{i=0}^{n} m_i(l, p_i) - \sum_{i=1}^{n} d_i(p_0, p_i) \]

Filter scores \hspace{1cm} Spring costs
DPM detection

test image

model
DPM detection

Root scale  Part scale

repeat for each level in pyramid
DPM detection
DPM detection

\[
\text{score}(l, p_0) = \max_{p_1, \ldots, p_n} \sum_{i=0}^{n} m_i(l, p_i) - \sum_{i=1}^{n} d_i(p_0, p_i)
\]
DPM detection

Generalized distance transform
Felzenszwalb & Huttenlocher '00
DPM detection

\[
\text{score}(l, p_0) = \max_{p_1, \ldots, p_n} \sum_{i=0}^{n} m_i(l, p_i) - \sum_{i=1}^{n} d_i(p_0, p_i) \\
= m_0(l, p_0) + \sum_{i=1}^{n} \max_{p_i} [m_i(l, p_i) - d_i(p_0, p_i)]
\]
DPM detection

All that’s left: combine evidence
DPM detection

max \[ m_i(l, p_i) - d_i(p_0, p_i) \]
Person detection progress

Progress bar:

<table>
<thead>
<tr>
<th>Year</th>
<th>AP</th>
<th>%</th>
<th>Year</th>
<th>AP</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>12%</td>
<td>27%</td>
<td>2009</td>
<td>36%</td>
<td>45%</td>
</tr>
<tr>
<td>2008</td>
<td>27%</td>
<td></td>
<td>2010</td>
<td>45%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>49%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2011</td>
</tr>
</tbody>
</table>

Example detections and learned filters:
One DPM is not enough: \textbf{What are the parts?}
General philosophy: enrich models to better represent the data
Mixture models

Data driven: aspect, occlusion modes, subclasses

Progress bar:

AP 12% 2005
27% 2008
36% 2009
45% 2010
49% 2011
Pushmi–pullyu?

Good generalization properties on Doctor Dolittle’s farm

\[
\frac{(\text{horse} + \text{horses})}{2} = \text{This was supposed to detect horses}
\]
Latent orientation

Unsupervised left/right orientation discovery

Progress bar:

<table>
<thead>
<tr>
<th>Year</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>12%</td>
</tr>
<tr>
<td>2008</td>
<td>27%</td>
</tr>
<tr>
<td>2009</td>
<td>36%</td>
</tr>
<tr>
<td>2010</td>
<td>45%</td>
</tr>
<tr>
<td>2011</td>
<td>49%</td>
</tr>
</tbody>
</table>

horse AP

- 0.42
- 0.47
- 0.57
Summary of results

[DT’05] AP 0.12

[APR’08] AP 0.27

[FGMR’10] AP 0.36

[GFM voc-release5] AP 0.45

Object detection with grammar models

[Girshick, Felzenszwalb, McAllester ’11] AP 0.49

Code at www.cs.berkeley.edu/~rbg/voc-release5
Part 2: DPM parameter learning

given fixed model structure

component 1

component 2
Part 2: DPM parameter learning

given fixed model structure

component 1

component 2

component 3

component 4

training images

\[ y \]

+1
Part 2: DPM parameter learning

given fixed model structure

component 1

component 2

training images

\[ y \]

\[ y \]

\[ +1 \]

\[ -1 \]
Part 2: DPM parameter learning

Parameters to learn:
- biases (per component)
- deformation costs (per part)
- filter weights

given fixed model \textit{structure}  

training images $y$

component 1  component 2
Linear parameterization of sliding window score

$$z = (p_1, \ldots, p_n)$$

$$\text{score}(l, p_0) = \max_{p_1, \ldots, p_n} \sum_{i=0}^{n} m_i(l, p_i) - \sum_{i=1}^{n} d_i(p_0, p_i)$$

Filter scores: $$m_i(l, p_i) = w_i \cdot \phi(l, p_i)$$

Spring costs: $$d_i(p_0, p_i) = d_i \cdot (dx^2, dy^2, dx, dy)$$

$$\text{score}(l, p_0) = \max_{z} w \cdot \Phi(l, (p_0, z))$$
Positive examples \((y = +1)\)

\(x\) specifies an image and bounding box

We want

\[
f_w(x) = \max_{z \in Z(x)} w \cdot \Phi(x,z)
\]

to score \(\geq +1\)

\(Z(x)\) includes all \(z\) with more than 70% overlap with ground truth
Positive examples ($y = +1$)

$x$ specifies an image and bounding box

We want

$$f_w(x) = \max_{z \in Z(x)} w \cdot \Phi(x, z)$$

to score $\geq +1$

$Z(x)$ includes all $z$ with more than 70% overlap with ground truth
Negative examples ($y = -1$)

$x$ specifies an image and a HOG pyramid location $p_0$

We want to score $\leq -1$

$$f_w(x) = \max_{z \in Z(x)} w \cdot \Phi(x, z)$$

$Z(x)$ restricts the root to $p_0$ and allows any placement of the other filters.
Negative examples \( (y = -1) \)

\( x \) specifies an image and a HOG pyramid location \( p_0 \)

\[
f_w(x) = \max_{z \in Z(x)} w \cdot \Phi(x, z)
\]

to score \( \leq -1 \)

\( Z(x) \) restricts the root to \( p_0 \) and allows any placement of the other filters

All configurations score low
Typical dataset

300 – 8,000 positive examples

500 million to 1 billion negative examples
(not including latent configurations!)

Large-scale optimization!
How we learn parameters: latent SVM

\[ E(w) = \frac{1}{2} \|w\|^2 + C \sum_{i} \max\{0, 1 - y_if_w(x_i)\} \]
How we learn parameters: latent SVM

\[ E(w) = \frac{1}{2} \|w\|^2 + C \sum_i \max\{0, 1 - y_i f_w(x_i)\} \]

\[ E(w) = \frac{1}{2} \|w\|^2 + C \sum_{i \in P} \max\{0, 1 - \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\} \]
\[ + C \sum_{i \in N} \max\{0, 1 + \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\} \]

\(P\): set of positive examples
\(N\): set of negative examples
Latent SVM and Multiple Instance Learning via MI-SVM

Latent SVM is mathematically equivalent to MI-SVM (Andrews et al. NIPS 2003)

Latent SVM can be written as a latent structural SVM (Yu and Joachims ICML 2009)

- natural optimization algorithm is concave-convex procedure
- similar to, but not exactly the same as, coordinate descent
Step 1

\[ Z_{Pi} = \arg\max_{z \in Z(x_i)} w(t) \cdot \Phi(x_i, z) \quad \forall i \in P \]

This is just detection:

We know how to do this!
Step 2

$$\min_w \frac{1}{2} \|w\|^2 + C \sum_{i \in P} \max\{0, 1 - w \cdot \Phi(x_i, Z_{P_i})\}$$

$$+ C \sum_{i \in N} \max\{0, 1 + \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\}$$

Convex!
Step 2

\[
\begin{align*}
\min_w & \quad \frac{1}{2} \|w\|^2 + C \sum_{i \in P} \max\{0, 1 - w \cdot \Phi(x_i, Z_{P_i})\} \\
& \quad + C \sum_{i \in N} \max\{0, 1 + \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\}
\end{align*}
\]

Convex!

Similar to a structural SVM
Step 2

\[
\min_w \frac{1}{2} \|w\|^2 + C \sum_{i \in P} \max\{0, 1 - w \cdot \Phi(x_i, Z_{P_i})\} \\
+ C \sum_{i \in N} \max\{0, 1 + \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\}
\]

Convex!

Similar to a structural SVM

But, recall 500 million to 1 billion negative examples!
Step 2

\[
\min_w \frac{1}{2} \|w\|^2 + C \sum_{i \in P} \max\{0, 1 - w \cdot \Phi(x_i, Z_{P_i})\} \\
+ C \sum_{i \in N} \max\{0, 1 + \max_{z \in Z(x)} w \cdot \Phi(x_i, z)\}
\]

Convex!

Similar to a structural SVM

But, recall 500 million to 1 billion negative examples!

Can be solved by a working set method
- “bootstrapping”
- “data mining” / “hard negative mining”
- “constraint generation”
- requires a bit of engineering to make this fast
What about the model structure?

**Given fixed model structure**

- component 1
- component 2

**Model structure**
- # components
- # parts per component
- root and part filter shapes
- part anchor locations

training images

\[ y \]

\[ +1 \]

\[ -1 \]
Learning model structure

1a. Split positives by aspect ratio

1b. Warp to common size

1c. Train Dalal & Triggs model for each aspect ratio on its own
Learning model structure

2a. Use D&T filters as initial $w$ for LSVM training
   Merge components

2b. Train with latent SVM
   Root filter placement and component choice are latent
3a. Add parts to cover high-energy areas of root filters

3b. Continue training model with LSVM
Learning model structure

without orientation clustering

with orientation clustering
Learning model structure

In summary

– repeated application of LSVM training to models of increasing complexity
– structure learning involves many heuristics — still a wide open problem!