Pictorial structures

Deva Ramanan

UC Irvine
Original plan

Inference of deformable part models
“Pictorial structures”
(This morning)

Learning for deformable part models
“Latent SVMs”
(This afternoon)
Revised plan

“Core” deformable part model system
(This morning)

“Extensions” of deformable part models
(This afternoon)
Goal: detect objects in cluttered images

person, plant, cat, dog, chair, sofa, car, bicycle, motorbike, table, plane, ...

Wednesday, August 7, 2013
Why is finding objects (e.g. people) difficult?

Variation in pose, viewpoint
Variation in appearance
Variation in illumination
Occlusion & clutter

Classic “nuisance factors” for general object recognition
Historical approaches

Geometric models (1970s-1990s)

Hand-coded models
**Historical approaches**

- **Geometric models** (1970s-1990s)
  - Hand-coded models

- **Statistical classifiers** (2000s-present)
  - Large-scale training
  - Appearance-based representations

**Learned model**

\[ f_w(x) = w \cdot \Phi(x) \]

- **Training**
  - **Training data** consists of images with labeled bounding boxes
  - Need to learn the model structure, filters and deformation costs

- **Positives**
  - Positive weights

- **Negatives**
  - Negative weights
Historical approaches

Geometric models (1970s-1990s)

Hand-coded models

Statistical classifiers (2000s-present)

Appearance-based representations

Large-scale training

Need to learn the model structure, filters and deformation costs

T

raining data consists of images with labeled bounding boxes

The relative improvement of our approach is 42%, indicating the quality of our flexible part mixture representation.

Our total performance of 89.6% compares favorably to the best previous result of 82.8%.

We also beat all previous results learned from training data.

Type z where the score associated with each combination decomposes into a tree and so is efficient to search over.

Figure 7: [Visualization of our full body model for geometric models for the hand are spread apart to capture a larger variety of relative orientations.]

Figure 5: We take a "data-driven" approach to orientation modeling by clustering the relative locations of parts with respect to their parents.

These clusters are used to generate mixture labels for parts during training.

For example, heads tend to be upright, and so the associated mixture models focus on upright orientations.

Because hands articulate to a large degree, we emphasize that our representation allows for the composition of any part type with any other part type.
Evaluating performance

Columbia Dataset (1996)

Caltech 101/256
Image Net

Flickr dataset (05-12)
“In-the-wild”
5 years of PASCAL people detection

Discriminative mixtures of star models 2007-2010
Felzenszwalb, McAllester, Ramanan *CVPR* 2008
Felzenszwalb, Girshick, McAllester, and Ramanan *PAMI* 2010

1% to 45% in 5 years
Benchmark evaluation

PASCAL VOC 2008 Average Precision Rankings

<table>
<thead>
<tr>
<th></th>
<th>aero</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>pers</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASIA_Det</td>
<td>25.2</td>
<td>14.6</td>
<td>9.8</td>
<td>10.5</td>
<td>6.3</td>
<td>23.2</td>
<td>17.6</td>
<td>9.0</td>
<td>9.6</td>
<td>10.0</td>
<td>13.0</td>
<td>5.5</td>
<td>14.0</td>
<td>24.1</td>
<td>11.2</td>
<td>3.0</td>
<td>2.8</td>
<td>3.0</td>
<td>28.2</td>
<td>14.6</td>
</tr>
<tr>
<td>Jena</td>
<td>4.8</td>
<td>1.4</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
<td>1.0</td>
<td>1.3</td>
<td>-</td>
<td>-</td>
<td>0.1</td>
<td>4.7</td>
<td>0.4</td>
<td>1.9</td>
<td>0.3</td>
<td>3.1</td>
<td>2.0</td>
<td>0.3</td>
<td>0.4</td>
<td>2.2</td>
<td>6.4</td>
</tr>
<tr>
<td>PlusClass</td>
<td>36.5</td>
<td>34.3</td>
<td>10.7</td>
<td>11.4</td>
<td>22.1</td>
<td>23.8</td>
<td>36.6</td>
<td>16.6</td>
<td>11.1</td>
<td>17.7</td>
<td>15.1</td>
<td>9.0</td>
<td>36.1</td>
<td>40.3</td>
<td>19.7</td>
<td>11.5</td>
<td>19.4</td>
<td>17.3</td>
<td>29.6</td>
<td>34.0</td>
</tr>
<tr>
<td>MPI_struct</td>
<td>25.9</td>
<td>8.0</td>
<td>10.1</td>
<td>5.6</td>
<td>0.1</td>
<td>11.3</td>
<td>10.6</td>
<td>21.3</td>
<td>0.3</td>
<td>4.5</td>
<td>10.1</td>
<td>14.9</td>
<td>16.6</td>
<td>20.0</td>
<td>2.5</td>
<td>0.2</td>
<td>9.3</td>
<td>12.3</td>
<td>23.6</td>
<td>1.5</td>
</tr>
<tr>
<td>Oxford</td>
<td>33.3</td>
<td>24.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>UoCTTIUCI</td>
<td>32.6</td>
<td>42.0</td>
<td>11.3</td>
<td>11.0</td>
<td>28.2</td>
<td>23.2</td>
<td>32.0</td>
<td>17.9</td>
<td>14.6</td>
<td>11.1</td>
<td>6.6</td>
<td>10.2</td>
<td>32.7</td>
<td>38.6</td>
<td>42.0</td>
<td>12.6</td>
<td>16.1</td>
<td>13.6</td>
<td>24.4</td>
<td>37.1</td>
</tr>
<tr>
<td>XRC_E_Det</td>
<td>26.4</td>
<td>10.5</td>
<td>1.4</td>
<td>4.5</td>
<td>0.0</td>
<td>10.8</td>
<td>4.0</td>
<td>7.6</td>
<td>2.0</td>
<td>1.8</td>
<td>4.5</td>
<td>10.5</td>
<td>11.8</td>
<td>13.6</td>
<td>9.0</td>
<td>1.5</td>
<td>6.1</td>
<td>1.8</td>
<td>7.3</td>
<td>6.8</td>
</tr>
</tbody>
</table>

UoCTTIUCI 1st on 7 classes, 2nd on 8

Test: ~ 2 second / image  
Train: ~ 4 hours

All code online

PASCAL VOC Lifetime Achievement Award 2010 
Invited application paper in ICML 2010 
Invited article in Communications of ACM 2011

Wednesday, August 7, 2013
Image features:

Histograms of oriented gradients (HOG)

Bin gradients from 8x8 pixel neighborhoods into 9 orientations

(Dalal & Triggs *CVPR 05*)
Scanning-window templates

Dalal and Triggs CVPR05 (HOG)
Papageorgiou and Poggio ICIP99 (wavelets)

$w = \text{weights for orientation and spatial bins}$

$w \cdot x > 0$

Train with a linear classifier (perceptron, logistic regression, SVMs...)
Scanning-window templates

Dalal and Triggs CVPR05 (HOG)
Papageorgiou and Poggio ICIP99 (wavelets)

$w = \text{weights for orientation and spatial bins}$

$w \cdot x > 0$

Train with a linear classifier (perceptron, logistic regression, SVMs...)
How to interpret positive and negative weights?

\[ w \cdot x > 0 \]

\[ (w_{\text{pos}} - w_{\text{neg}}) \cdot x > 0 \]

\[ w_{\text{pos}} \cdot x > w_{\text{neg}} \cdot x \]

Right approach is to compete pedestrian, pillar, doorway... models

Background class is hard to model - easier to penalize particular vertical edges
Out-of-core learning

Our test set distribution is highly imbalanced; so should be the training set
(hundreds of positives, hundreds of millions of negatives)
Out-of-core learning

Our test set distribution is highly imbalanced; so should be the training set
(hundreds of positives, hundreds of millions of negatives)

SVMs are attractive because they generate sparse learning problems
(One can solve problems that are too big to fit in memory)
Large-scale learning

1. Train SVM with subset of training data
2. Use model to find margin violations on all training data
3. If no new violations are found, model is optimal!

(More in afternoon’s talk)
How to model large variations in appearance?
Mixtures of templates

Train “sub-category” templates for each type of pose, body-shape, etc.
But how to handle...

Long-tail distribution of poses

We need lots of templates, and will likely have little data of ‘yoga twist’ poses
Deformable part models

Figure 5: We take a "data-driven" approach to orientation modeling by clustering the relative locations of parts with respect to their parents. These clusters are used to generate mixture labels for parts during training. For example, heads tend to be upright, and so the associated mixture models focus on upright orientations. Because hands articulate to a large degree, mixture models for the hand are spread apart to capture a larger variety of relative orientations.

Figure 7: Visualization of our full-body model for $T = 4$ trained on the Parse dataset. Note that we show them as 6 separate models, but we emphasize that our representation allows for the composition of any part type with any other part type, where the score associated with each combination decomposes into a tree and so is efficient to search over and is learned from training data.

Table 3: We compare our model to all previous published results on the Parse dataset using the standard criteria of $P + P$. Our total performance of 89% compares favorably to the best previous result of 82%. We also beat all previous results on a part-by-part basis except for torso and lower arm detection, for which we are second. $R$ uses the same HOG feature set as our approach, but embedded in a classic articulated pictorial structure. The relative improvement of our approach is 42%, indicating the quality of our flexible part mixture representation.
History over 40 years

Model encodes local appearance + pairwise geometry

Pictorial Structures (Fischler & Elschlager 73, Felzenswalb and Huttenlocher 00)
Cardboard People (Yu et al 96)
Body Plans (Forsyth & Fleck 97)
Active Appearance Models (Cootes & Taylor 98)
Constellation Models (Burl et all 98, Fergus et al 03)
Relationship to other deformable models

Active appearance models
Continuous parameterization of shape
Continuous matching algs
“Local search”

Deformable parts
Discrete parameterization of shape
Combinatorial matching algs
“Brute-force search”
Here of the following form.

In the case of one of our star models, our second class of models represents each object category by a mixture of star models. To train models using partially labeled data, we use a latent variable formulation of MI4SVM. In a latent SVM each example \( x \) is scored by a function \( \Phi(x, z) \)

\[
S(x, z) = x = \text{image} \\
z_i = (x_i, y_i) \\
z = \{z_1, z_2, \ldots\}
\]
Here shows a mixture model for the bicycle category information components of the score of that component model at the given location. In this case the latent SVM $\Phi(x, z) = \max_{i} \beta_{i} \cdot \phi(x, z_i)$ is scored by a function $\phi(x, z_i)$.

The score of one of our mixture models at a given position and scale is the maximum over the concatenation of subwindows from a feature pyramid and part deformation features.

We note that $\beta_{i}$ can handle very general forms of latent information. For example, it could specify a derivation under a rich visual grammar.

To obtain high performance using discriminative training it is often important to use large training sets. In the case of object detection the training problem is highly unbalanced because there is vastly more background than objects. This motivates a process of searching through the background to find a relatively small number of potential false positives.

A methodology of training sets is defined by a coarse root filter $f_{i,fl}$ and several higher resolution part filters $f_{i,fl}$ and a spatial model for the location of each part relative to the root $f_{i,fl}$. The filters specify and deformation cost weights.

To train models using partially labeled data we use a latent variable formulation of MI4SVM.

In a latent SVM each example $\beta(x, z)$ is a vector of model parameters. For example, $z = \{z_{1}, z_{2}, \ldots\}$ could specify a component label and a configuration for that component. Figure shows a mixture model for the bicycle category information components of the score of that component model at the given location. In this case the latent SVM $\Phi(x, z) = \max_{i} \beta_{i} \cdot \phi(x, z_i)$ is scored by a function $\phi(x, z_i)$.
Scoring function

\[ S(x, z) = \sum_i w_i \cdot \phi(x, z_i) + \sum_{i,j \in E} w_{ij} \cdot \psi(z_i, z_j) \]

\[ \psi(z_i, z_j) = [dx \quad dx^2 \quad dy \quad dy^2]^T \]

\[ E = \text{relational graph} \]

\( x = \text{image} \)
\( z_i = (x_i, y_i) \)
\( z = \{z_1, z_2, \ldots\} \)
Deformation modes

\[
\sum_{i,j \in E} w_{ij} \cdot \psi(z_i, z_j) = (z - \mu)^T \Lambda (z - \mu)
\]

where \((\mu, \Lambda)\) are functions/reparameterizations of \(\{w_{ij}\}\) and \(\Lambda\) is the block-sparse inverse of a shape “covariance” matrix.
Deformation modes

\[
\sum_{ij \in E} w_{ij} \cdot \psi(z_i, z_j) = (z - \mu)^T \Lambda (z - \mu)
\]

where \((\mu, \Lambda)\) are functions/reparameterizations of \(\{w_{ij}\}\) and \(\Lambda\) is the block-sparse inverse of a shape “covariance” matrix.
Scoring function

\[ S(x, z) = \sum_i w_i \cdot \phi(x, z_i) + \sum_{ij \in E} w_{ij} \cdot \psi(z_i, z_j) \]

Score is linear in local templates \( w_i \) and spring parameters \( w_{ij} \)

\[ S(x, z) = w \cdot \Phi(x, z) \]
Learning structured linear parameters

\[ S(x, p) = w \cdot \Phi(x, p) \]

(pos)

(neg)

(Apply same sparse learning tricks to deal with exponential set of negatives!)

Wednesday, August 7, 2013
Learning structured linear parameters

\[ S(x, p) = w \cdot \Phi(x, p) \]

• pos
  • neg

(Apply same sparse learning tricks to deal with exponential set of negatives!)
Inference
Inference

[Image of a snowy scene with blue boxes indicating parts of the scene]
Inference

K parts with L possible positions: score all $L^K$ configurations
Inference: \( \max_{z} S(x,z) \)

Felzenszwalb & Huttenlocher 05

- \( L \) candidate locations, \( K \) parts
- Dynamic programming reduces search from \( O(L^k) \) to \( O(KL^2) \) for trees
- For each candidate torso, independently estimate best arm and leg
- In practice, no more expensive than scoring each part independently
Inference: max $S(x,z)$

1) Initialize nodes with match score
2) Initialize edges with spring score
3) Find best path from left to right

In practice, (1) is bottleneck
General formulation

\[ S(x, z) = \sum_i \phi_i(z_i, x) + \sum_{ij \in E} \psi_{ij}(z_i, z_j, x) \]

Local and pairwise potentials can be arbitrary nonlinear functions of image and position

(e.g., neural net part model)

(e.g., intervening contour cue on part pairs)
The score of one of our mixture models at a given position and scale is the maximum over weights for histogram of oriented gradients features. Their visualization shows the positive weights at different orientations. The higher resolution part filters $\beta$ and a spatial model for the location of each part relative to the root $\gamma$. The filters specify a derivation under a rich visual grammar.

Inference

Visualization of the spatial models reflects the “cost” of placing the center of a part at different locations relative to the root. In the case of one of our mixture models, $\beta$ is a vector of model parameters.
Classification

\[ f_w(x) = w \cdot \Phi(x) \]

\[ f_w(x) > 0 \]
Latent-variable classification

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

\[ f_w(x) > 0 \]
Comparison

\[ f_w(x) = w \cdot \Phi(x) \]

Score \( f_w(x) \) is linear in \( w \)

\[ f_w(x) = \max_z w \cdot \Phi(x, z) \]

?
SVMs

Given positive and negative training windows \( \{x_n\} \)

\[
L(w) = ||w||^2 + \sum_{n \in \text{pos}} \max(0, 1 - f_w(x_n)) + \sum_{n \in \text{neg}} \max(0, 1 + f_w(x_n))
\]

\[
f_w(x) = w \cdot \Phi(x)
\]

\( L(w) \) is convex (Quadratic Program)
Latent SVMs

Given positive and negative training windows \( \{x_n\} \)

\[
L(w) = ||w||^2 + \sum_{n \in \text{pos}} \max(0, 1 - f_w(x_n)) + \sum_{n \in \text{neg}} \max(0, 1 + f_w(x_n))
\]

\[
f_w(x) = \max_z w \cdot \Phi(x, z)
\]

\( L(w) \) is “almost” convex
Latent SVMs

\[ f_w(x) = \max_z w \cdot \Phi(x, z) \]

Given positive and negative training windows \( \{x_n\} \)

\[ L(w) = ||w||^2 + \sum_{n \in \text{pos}} \max(0, 1 - f_w(x_n)) + \sum_{n \in \text{neg}} \max(0, 1 + f_w(x_n)) \]

\[ w \cdot \Phi(x_n, z_n) \]

“almost-convex” - \( L(w) \) is convex if we fix latent values for positives
1) Given positive part locations, learn $w$ with a convex program

$$w = \arg \min_w L(w) \quad \text{with fixed} \quad \{z_n : n \in \text{pos}\}$$
Coordinate descent

1) Given positive part locations, learn $w$ with a convex program

$$w = \arg\min_w L(w) \quad \text{with fixed} \quad \{z_n : n \in \text{pos}\}$$

2) Given $w$, estimate part locations on positives

$$z_n = \arg\max_z w \cdot \Phi(x_n, z) \quad \forall n \in \text{pos}$$

The above steps perform coordinate descent on a joint loss. Can be seen as an instance of the CCCP algorithm (Yuille)
Treat ground-truth labels as partially latent

Allows for “cleaning up” of noisy labels (in blue) during iterative learning
Initialization

Learn root filter with SVM

Initialize part filters to regions in root filter with lots of energy
Example models
Example models

Figure 5: Some results from the PASCAL 2007 dataset. Each row shows detections using a model for a specific class: Person, Bottle, Car, Sofa, Bicycle, Horse. The first three columns show correct detections while the last column shows false positives. Our system is able to detect objects over a wide range of scales, such as the cars, and poses, such as the horses. The system can also detect partially occluded objects such as a person behind a bush. Note how the false detections are often quite reasonable, for example detecting a bus with the car model, a bicycle sign with the bicycle model, or a dog with the horse model. In general, the part filters represent meaningful object parts that are well localized in each detection, such as the head in the person model.
Example models
Example models

Some results from the PASCAL 2007 dataset. Each row shows detections using a model for a specific class: Person, Bottle, Car, Sofa, Bicycle, Horse. The first three columns show correct detections while the last column shows false positives. Our system is able to detect objects over a wide range of scales such as the cars and poses such as the horses. The system can also detect partially occluded objects such as a person behind a bush. Note how the false detections are often quite reasonable, for example detecting a bus with the car model, a bicycle sign with the bicycle model, or a dog with the horse model. In general the part filters represent meaningful object parts that are well localized in each detection, such as the head in the person model.
Example models
Example models

Fig. 10 Examples of high-scoring detections on the PASCAL 2007 dataset. The framed images (last two in each row) illustrate false positives for each category. Many false positives, such as for person and cat, are due to the bounding box scoring criteria.

False positive due to imprecise bounding box
Google’s Pet Emoticon Detector

Our system!
Extensions: latent sub-categories

Frontal cars

Side / three-quarters view cars

Felzenswalb, Girshick, McAllester, and Ramanan *PAMI* 2010
Extensions: how do we find multiple objects?

Apply NMS to root scores after dynamic programming
Extensions: how do we find multiple objects?

Apply NMS to root scores after dynamic programming
But will it work for ...?

Perhaps we want to use additional contextual information to resolve (global depth ordering, temporal info, etc...)
N-best decoding

Generate N high-scoring candidates with simple (tree) model, and evaluate with complex model

Popular in speech, but why not vision?
N-best decoding

Generate N high-scoring candidates with simple (tree) model, and evaluate with complex model

Popular in speech, but why not vision?
N-best maximal decoding

N-best with “NMS” or “mode-finding”

Park and Ramanan, ICCV11
Yadollahpour et al. ECCV12
N-best maximal decoding

Intuition: backtrack from all part “max-marginals”, not just root
(can we done without any noticeable increase in computation)

Park and Ramanan, ICCV 2011
N-best maximal decoding

Philosophy: Delay hard decisions as much as possible

Candidate interest points
Candidate parts
Candidate poses
A look back: part models as mixture models

\[ S(x, z) = \sum_i w_i \cdot \phi(x, z_i) + \sum_{ij \in E} w_{ij} \cdot \psi(z_i, z_j) \]

Each distinct placement of parts yields a unique global template

\[ S(x, z) = w_z \cdot x + b_z \]
Parts as mixture models

Spatial model defines bias or “prior”

\[ f(x) = \max_{z \in \mathbb{Z}} w_z \cdot x + b_z \]
Parts as mixture models

Part models allow us to represent an exponentially-large family of global templates

\[ f(x) = \max_{z \in Z} w_z \cdot x + b_z \]
Deformation modes
Deformation modes
DPMs as large-mixture models

\[ f(x) = \max_{z \in Z} w_z \cdot x + b_z \]

- “Double-counting” manifests simply as too strong of a weight

- Suggests jointly learning parts is crucial
  (more on that this afternoon)
Revisit latent (vs linear) classification

\[ f_w(x) = w \cdot \Phi(x) \]

Score is linear in \( x \)

Positive set \( \{ x : f_w(x) > 0 \} \)
  is half-space

\[ f_w(x) = \max_z w_z \cdot x \]

Score is ?

Positive set is ?
Revisit latent (vs linear) classification

\[ f_w(x) = w \cdot \Phi(x) \]
Score \( f_w(x) \) is linear in \( x \)
Positive set \( \{x: f_w(x) > 0\} \) is half-space

\[ f_w(x) = \max_z w_z \cdot x \]
Score \( f_w(x) \) is convex in \( x \)
Positive set \( \{x: f_w(x) > 0\} \) is concave
DPMs vs explicit mixtures

Mixtures of rigid templates
“Exemplar SVMs”
Malisiewicz et al ICCV 11

Part model
DPMs vs explicit mixtures

Mixtures of rigid templates

“Exemplar SVMs”
Malisiewicz et al ICCV 11

Compared to a mixture of exemplars, part models...

1) Share parameters across templates
2) Synthesize new templates not seen during training
3) Efficiently search over templates using dynamic programming
Mixtures of rigid templates

Mixtures of rigid templates with tied parameters (given by parts)

Part model

1) Share parameters across mixtures
2) “Synthesize” new rigid templates not seen during training

To examine (1) vs (2), lets define mixture of exemplars with sharing
An analysis of part models

Zhu, Vondrick, Ramanan & Fowlkes,
“Do we need more training data or better models?”
BMVC 2012
An analysis of part models

Zhu, Vondrick, Ramanan & Fowlkes,
“Do we need more training data or better models?”
BMVC 2012
An analysis of part models

“Synthesis” of unseen (rare) templates is even more beneficial than sharing

Wednesday, August 7, 2013
An argument against “big-data”

One can train a state-of-art face detector (c.f. Google Picassa & Facebook’s face.com) with 100 faces!
A look back

Geometric statistical models

Wednesday, August 7, 2013