Lecture 16

Object detection
Administrative

A4 is out
   - Due May 23th

A5 is out
   - Due May 30th
Administrative

Recitation this friday
- Recognition review
- Jieyu Zhang
So far: A simple recognition pipeline

Training Images

Training Labels

Image Features → Training → Learned Classifier

Test Image

Image Features → Learned Classifier → Prediction
So far: PCA versus LDA

We want a projection that maximizes:

\[ J(w) = \max \frac{\text{between class scatter}}{\text{within class scatter}} \]
So far: Bag of words features

- Every image now becomes a k-dimensional histogram representation.
- We can use these features for any recognition task.
So far: Bag of words + pyramids

Locally orderless representation at several levels of spatial resolution

level 0

level 1

level 2
Today’s agenda

- Object detection
  - Task and evaluation
- A simple detector
- Deformable parts model
Today’s agenda

● Object detection
  ○ Task and evaluation
● A simple detector
● Deformable parts model
Object Detection

• What do you see in the image?

Credit: Flickr user neilalderney123
Object Detection

- **Problem**: Detecting and localizing generic objects from various categories, such as cars, people, etc.

- **Challenges**:
  - Illumination,
  - viewpoint,
  - deformations,
  - Intra-class variability
Object Detection Benchmarks

- PASCAL VOC Challenge

- 20 categories
- Annual classification, detection, segmentation, … challenges
Object Detection Benchmarks

- PASCAL VOC Challenge
- ImageNet Large Scale Visual Recognition Challenge (ILSVRC)
  - 200 Categories for detection
Object Detection Benchmarks

- PASCAL VOC Challenge
- ImageNet Large Scale Visual Recognition Challenge (ILSVR)
- Common Objects in Context (COCO)
  - 80 Object categories
How do we evaluate object detection?

- predictions
- ground truth
Defining what is a good versus bad detection

IoU is a metric used to decide good from bad predictions.

Given a predicted box and ground truth box:

\[ \text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \]

IoU = intersection between the two boxes over (divided by) the union of the two.
Defining what is a good versus bad detection

We say a prediction was good if it has $\text{IoU} > 0.5$ with any of the ground truth boxes.

0.5 is a threshold that is generally accepted as a good heuristic.
How do we evaluate object detection?

True positive:
- The overlap of the prediction with the ground truth is MORE than 0.5
How do we evaluate object detection?

- **True positive:**
- The overlap of the prediction with the ground truth is LESS than 0.5

- **False positive:**
How do we evaluate object detection?

- True positive:
- False positive:
- False negative:
  - The objects that our model doesn’t find
How do we evaluate object detection?

- True positive:
- False positive:
- False negative:
  - The objects that our model doesn’t find

What is a True Negative?
<table>
<thead>
<tr>
<th></th>
<th>Predicted 1</th>
<th>Predicted 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>True 1</td>
<td>true positive</td>
<td>false negative</td>
</tr>
<tr>
<td>True 0</td>
<td>false positive</td>
<td>true negative</td>
</tr>
</tbody>
</table>

\[
\text{precision} = \frac{TP}{TP + FP}
\]

\[
\text{recall} = \frac{TP}{TP + FN}
\]
How do we evaluate object detection?

- True positive: 1
- False positive: 2
- False negative: 1

Q. What is the precision?
How do we evaluate object detection?

True positive: 1
False positive: 2
False negative: 1

Q. What is the precision?

Q. What is the recall?
How to intuitively understand precision versus recall

● Precision:
  ○ how many of the predicted detections are correct?

● Recall:
  ○ how many of the ground truth objects are detected?
In reality, our model makes a lot of predictions with varying scores between 0 and 1.

Here are all the boxes that are predicted with score > 0.

From this, we see that:
- Recall is perfect!
- But our precision is BAD!
How do we evaluate object detection?

Here are all the boxes that are predicted with score > 0.5

We are using a threshold of 0.5

Q. What happens to precision if threshold is high?
How do we evaluate object detection?

Here are all the boxes that are predicted with score > 0.5

We are using a threshold of 0.5

Q. What happens to recall if threshold is high?
Precision – recall curve (PR curve)
Which model is the best?
Which model is the best?
True positives - detecting person
False positives - detecting person

UoCTTILSVM-MDPM

MIZZOU_DEF-HOG-LBP

NECUIUC_CLS-DTCT
Near misses: IoU falls short of 0.5
True positives - detecting **bicycle**
False positives - detecting bicycle

UoCTTI_Lsvm-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT
Today’s agenda

● Object detection
  ○ Task and evaluation
● A simple detector
● Deformable parts model
Dalal-Triggs method

sliding window
At every patch as the window slides

1. Convert the image patch into your favorite feature representation
   a. For example:
      i. HoG,
      ii. HoG with PCA,
      iii. RGB with LDA,
      iv. Bag of words on RGB
      v. etc.

2. Use a trained classifier to determine if it is a specific class
   a. e.g. kNN classifier

3. Accumulate the predictions over all the patches
Sliding window + hog features

- Slide through the image and check if there is an object at every location

No person here
Sliding window + hog features

- Slide through the image and check if there is an object at every location

YES!! Person match found
Sliding window + hog features

- But what if we were looking for buses?

No bus found
Sliding window + hog features

But what if we were looking for buses?

No bus found
Sliding window + hog features

- We will never find the object if we don’t choose our window size wisely!

No bus found
Sliding window + hog features

- We need to do multi-scale sliding windows with pyramids
Computationally, we first resize the image to different sizes and then extract features at each size.
Today’s agenda

- Object detection
  - Task and evaluation
- A simple detector
- Deformable parts model
Recap – bag of words

• We can present images as a set of “words”
  ○ Where each word represents a part of the image.

• Can we use the location of these patches to find objects within those images?
Deformable Parts Model

● Represents an object as a “collection of parts” arranged in a “deformable configuration”
● Each part represents local appearances
● Spring-like connections between certain pairs of parts

Fischler and Elschlager, Pictoral Structures, 1973
Deformable parts model

- The parts of an object form pairwise relationships.
- We can model this using a “star model” where every part is defined relative to a root.
Detecting a person with their parts

• For example, a person can be modelled as having a head, left arm, right arm, etc.
• All parts can be modelled relative to the global person detector, which acts as the root.
Deformable parts model

- Each model will have a **global** filter. And a set of **part** filters. Here is an example of a global person filter with it’s ‘head’ part filter:
5-part bicycle model

“side view” bike model component

Root filter

Part filters
Deformable parts model

- Mixture of deformable part models
- Each component has global component + deformable parts
- Part filters have finer details
DPM for person model with 5 parts

If the head is here, the penalty is low

If the head is here, the penalty is high
DPM for person model with 5 parts

If the arm is here, the penalty is low

If the arm is here, the penalty is high
Multiple DPM for person model with 6 parts
DPM for car with 6 parts

side view

frontal view

root filters (coarse)  part filters (fine)  deformation models
How do we use the parts to make a detection?

Intuition:

1. First, use the sliding windows at different pyramid scales to detect each part (and the root).
2. Each part gives you a score for where the person might be.
3. Accumulate the global and part scores and penalize the deformation of the parts.
Example for detecting people

Image input

A feature template for person
First extract features

Image input

Features

A feature template for person

Features at 2x resolution
Calculate scores for part templates

Global scores for where a person might be

Convolution
Calculate scores for global template

Features at 2x resolution

convolution

Scores for head

Scores for right arm
After step 1, we have scores for all parts and global template

Global scores  Scores for head  Scores for right arm
Allowing each part to deform and guess where the entire body is.

- Given the location for the detected head, we can guess where the body should be.
- The body should be in the direction \( v_i \) predefined in the model.
- Bodies can be of different sizes and shapes. So we allow it to deform by some variable \( d_i \).
- This deformation spreads the scores to potential locations of the body.
Step 2: each part gives you a score for where the person might be.

Global scores

Scores for head

Scores for right arm

Each part is allowed to deform. So it deforms to where the person might be.

Intuition: If the head is here, where is the whole person likely to be?
Step 3: Add up the scores for the final detections

Global scores

Scores for head

Scores for right arm

Add up final scores
Formally, DPM is defined as:

- A model for an object with \( n \) parts is a \( (n + 2) \) tuple:

\[
(F_0, P_1, \cdots, P_n, b)
\]

- Each part-based model defined as:

\[
(F_i, v_i, d_i)
\]

- \( F_i \) filter for the \( i \)-th part
- \( v_i \) “anchor” position for part \( i \) relative to the root position
- \( d_i \) defines a deformation cost for each possible placement of the part relative to the anchor position
$d_i$ can be defined in many ways. We will use a Gaussian filter to define it.

If the head is here, the penalty is low.

If the head is here, the penalty is high.
Calculating the score for a detection

The score for a detection is defined as the sum of scores for the global and part detectors \textit{minus} the sum of deformation costs for each part.

This means that if a detection’s parts are really far away from where they should be, it’s probably a false positive.
Deformable Parts Model (DPM) - bicycle

- Root filters: coarse resolution
- Part filters: finer resolution
- Deformation models
DPM with HoG features - person
DPM - bottle

root filters
coarse resolution

part filters
finer resolution

deformation models
Results – car detection

high scoring true positives

high scoring false positives
Results – Person detection

high scoring true positives

high scoring false positives (not enough overlap)
Results – horse detection

high scoring true positives

high scoring false positives
DPM - discussion

● **Approach**
  ○ Manually selected set of parts - Specific detector trained for each part
  ○ Spatial model trained on part activations
  ○ Evaluate joint likelihood of part activations

● **Pros**
  ○ Parts have intuitive meaning.
  ○ Standard detection approaches can be used for each part.
  ○ Works well for specific categories.

● **Disadvantages**
  ○ Parts need to be selected manually
  ○ Some parts don’t have a simple appearance
  ○ No guarantee that some important part hasn’t been missed
  ○ When adding a new category, it takes a lot of manual effort
Extensions - From star shaped model to constellation model

“Star” shape model

Fully connected shape model
Today’s agenda

- Object detection
  - Task and evaluation
- A simple detector
- Deformable parts model
Next lecture

Motion and Tracking
Calculating the score for a detection

The score for a detection is defined as the sum of scores for the global and part detectors minus the sum of deformation costs for each part.

\[
\text{detection score} = \sum_{i=0}^{n} F_i \phi(p_i, H) - \sum_{i=1}^{n} d_i(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2)
\]
Calculating the score for a detection

\[\text{detection score} = \sum_{i=0}^{n} F_i \phi(p_i, H) - \sum_{i=1}^{n} d_i(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2)\]

Scores for each part filter + global filter (similar to Dalal and Triggs).
Remember from Dalal and Triggs

Filter $F$

Score of $F$ at position $p$ is $F \cdot \phi(p, H)$

$\phi(p, H) =$ concatenation of HOG features from subwindow specified by $p$
Deformable parts calculates a score for each part along with a global score

\[ p_i = (x_i, y_i, l_i) \] specifies the level and position of the \( i \)-th filter

\[ z = (p_0, \ldots, p_n) \]

- \( p_0 \): location of root
- \( p_1, \ldots, p_n \): location of parts
Detection pipeline

Now apply the spatial costs for each part:

\[
\text{detection score} = F_i \phi(p_i, H) - d_i(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2)
\]
Detection pipeline

Now add the global filter:

\[ \text{detection score} = F_0 \phi(p_i, H) + \sum_{i=1}^{n} F_i \phi(p_i, H) - \sum_{i=1}^{n} d_i(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2) \]
Calculating the score for a detection

\[
detection\ \text{score} = \sum_{i=0}^{n} F_i \phi(p_i, H) - \sum_{i=1}^{n} d_i(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2)
\]

The deformation costs for each part.

\(\Delta x_i\) measures the distance in the x-direction from where part \(i\) should be.

\(\Delta y_i\) measures the same in the y-axis direction.

\(d_i\) is the weight associated for part \(i\) that penalizes the part for being away.
Calculating the score for a detection

\[
\text{detection score} = \sum_{i=0}^{n} F_i \phi(p_i, H) - \sum_{i=1}^{n} d_i(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2)
\]

If \(d_i = (0, 0, 1, 0)\). What does this mean?
What we will learn today

- Naïve Bayes
Naïve Bayes

- Classify image using histograms of occurrences on visual words:

\[
x = \sum_{i=1}^{m} x_i
\]

- where:
  - \( x_i \) is the event of visual word \( v_i \) appearing in the image,
  - \( N(i) \) the number of times word \( v_i \) occurs in the image,
  - \( m \) is the number of words in our vocabulary.

Csurka Bray, Dance & Fan, 2004
Naïve Bayes - classification

• Our goal is to classify that the image represented by $x$ is belongs class that has the highest posterior probability:

$$c^* = \arg \max_c P(c \mid x)$$
Naïve Bayes – conditional independence

• Naïve Bayes classifier assumes that visual words are conditionally independent given object class.

• Therefore, we can multiply the probability of each visual word to obtain the joint probability.

• Model for image $x$ under object class $c$:

$$P(x \mid c) = \prod_{i=1}^{m} P(x_i \mid c)$$

• How do we compute $P(x_i \mid c)$?

Csurka Bray, Dance & Fan, 2004
Naïve Bayes – prior

• Class priors $P(c)$ encode how likely we are to see one class versus others.
• Note that:

$$
\sum_{c} P(c) = 1
$$

Csurka Bray, Dance & Fan, 2004
Naïve Bayes - posterior

- With the equations from the previous slides, we can now calculate the probability that an image represented by $x$ belongs to class category $c$.

$$P(c \mid x) = \frac{P(c) P(x \mid c)}{\sum_{c'} P(c') P(x \mid c')}$$

Bayes Theorem
Naïve Bayes – posterior

- With the equations from the previous slides, we can now calculate the probability that an image represented by $x$ belongs to class category $c$.

\[ P(c \mid x) = \frac{P(c) P(x \mid c)}{\sum_{c'} P(c') P(x \mid c')} \]

\[ P(c \mid x) = \frac{P(c) \prod_{i=1}^{m} P(x_i \mid c)}{\sum_{c'} P(c') \prod_{i=1}^{m} P(x_i \mid c')} \]
Let’s break down the posterior

The probability that $x$ belongs to class $c_1$:

$$P(c_1 \mid x) = \frac{P(c_1) \prod_{i=1}^{m} P(x_i \mid c_1)}{\sum_{c'} P(c') \prod_{i=1}^{m} P(x_i \mid c')}$$

And the probability that $x$ belongs to class $c_2$:

$$P(c_2 \mid x) = \frac{P(c_2) \prod_{i=1}^{m} P(x_i \mid c_2)}{\sum_{c'} P(c') \prod_{i=1}^{m} P(x_i \mid c')}$$
Both their denominators are the same

The probability that $x$ belongs to class $c_1$:

$$P(c_1 \mid x) = \frac{P(c_1) \prod_{i=1}^{m} P(x_i \mid c_1)}{\sum_{c'} P(c') \prod_{i=1}^{m} P(x_i \mid c')}$$

And the probability that $x$ belongs to class $c_2$:

$$P(c_2 \mid x) = \frac{P(c_2) \prod_{i=1}^{m} P(x_i \mid c_2)}{\sum_{c'} P(c') \prod_{i=1}^{m} P(x_i \mid c')}$$
Both their denominators are the same

- Since we only want the max, we can ignore the denominator:

\[
P(c_1 | x) \propto P(c_1) \prod_{i=1}^{m} P(x_i | c_1)
\]

\[
P(c_2 | x) \propto P(c_2) \prod_{i=1}^{m} P(x_i | c_2)
\]
For the general class $c$, 

$$P(c \mid x) \propto P(c) \prod_{i=1}^{m} P(x_i \mid c)$$
For the general class $c$, 

$$P(c \mid x) \propto P(c) \prod_{i=1}^{m} P(x_i \mid c)$$

We can take the log:

$$\log P(c \mid x) \propto \log P(c) + \sum_{i=1}^{m} \log P(x_i \mid c)$$
Naïve Bayes - classification

- We can now classify that the image represented by $\mathbf{x}$ is belongs class that has the highest probability:

$$c^* = \arg \max_c P(c \mid \mathbf{x})$$

$$c^* = \arg \max_c \log P(c \mid \mathbf{x})$$
Naïve Bayes - classification

- So, the following classification becomes:

\[
c^* = \arg \max_c P(c \mid \mathbf{x})
\]
\[
c^* = \arg \max_c \log P(c \mid \mathbf{x})
\]
\[
c^* = \arg \max_c \log p(c) + \sum_{i=1}^{m} \log p(x_i \mid c)
\]