CAP5415
Computer Vision

Yogesh S Rawat
yogesh@ucf.edu
HEC-241
Administrative details
Programming assignment 2

• Optional
• BONUS points

• Due on 9th November
Programming assignment 3

• Start date?
  • You will need time for course project

• 2\textsuperscript{nd} Nov
• 5\textsuperscript{th} Nov
• 9\textsuperscript{th} Nov
Questions?
Object Detection

Lecture 13
Agenda

• Object detection
• Sliding window
• Image pyramid
• Evaluation
Object Recognition

• **Problem:** Given an image A, does A contain an image of a person?
Object localization
Human Detection

• UAV images
• Surveillance images
Object detection

- Multiple objects
Why detection?

• Self-driving car
A simple solution

Sliding window
Sliding window

• Slide a box around image and classify each crop

• There are many interesting papers
  • Viola and Jones (2001): Face detector (22K citations)
  • Dalal and Triggs (2005): HOG (33K citations)
  • Felzenswalb et al. (2010): Deformable part-based model (10K citations)
Naïve approach: Template Matching

Find the chair in this image

Output of correlation

This is a chair
Template Matching

Find the chair in this image

Epic fail!
Simple template matching is not going to make it
Sliding window

• General approach
  • Scan all possible locations
  • Extract features
  • Classify features
  • Post-processing
Sliding window

• General approach
  • Scan all possible locations
    • Extract features
    • Classify features
    • Post-processing

• Inspect all windows
• Size of window is fixed
Sliding window

- General approach
  - Scan all possible locations
  - Extract features
  - Classify features
  - Post-processing

- People at different sizes?

Objects can be of very different sizes (scales), even in the same image. How do we deal with that?
Sliding window

• General approach
  • Scan all possible locations
    • Extract features
    • Classify features
    • Post-processing

• Down-scale the image
Sliding window

• General approach
  • **Scan all possible locations**
    • Extract features
    • Classify features
    • Post-processing

• Keep shrinking
Scale-space pyramid

• Full image pyramid
Gaussian pre-filtering

Solution: filter the image, then subsample
Gaussian pre-filtering

Solution: filter the image, *then* subsample
Image pyramid

A bar in the big images is a hair on the zebra’s nose;
in smaller images, a stripe;
in the smallest, the animal’s nose
Gaussian pyramid construction

Repeat
• Filter
• Subsample

Until minimum resolution reached
• can specify desired number of levels (e.g., 3-level pyramid)

The whole pyramid is only 4/3 the size of the original image!
Challenges

• Different aspect-ratio?
Sliding window

• General approach
  • Scan all possible locations
  • Extract features
    • Classify features
    • Post-processing

• HOG/SIFT
Sliding window

• General approach
  • Scan all possible locations
  • Extract features
    • Classify features
    • Post-processing

• Extract features
Sliding window

- General approach
  - Scan all possible locations
  - Extract features
  - Classify features
  - Post-processing

- Linear/SVM/NN
Sliding window

• General approach
  • Scan all possible locations
  • Extract features
  • **Classify features**
  • Post-processing

• Linear/SVM/NN
Sliding window

- General approach
  - Scan all possible locations
  - Extract features
  - **Classify features**
  - Post-processing

- Run classifier at all scales
- Use a threshold
- Detect all positives
Sliding window

• General approach
  • Scan all possible locations
  • Extract features
  • Classify features
  • Post-processing

• Multiple detections
• Non-maxima suppression
Non-maximum suppression

\[ M(x, y) = \begin{cases} 
|\nabla S(x, y) | & \text{if } |\nabla S(x, y) | > |\Delta S(x', y') | \\
& \& |\Delta S(x, y) | > |\Delta S(x'', y'') | \\
0 & \text{otherwise}
\end{cases} \]

\( x' \) and \( x'' \) are the neighbors of \( x \) along normal direction to an edge.
Non-maxima suppression (NMS)

• Iterate over all detections
  • Pick the highest scoring box
• Find overlap
  • with all other boxes
• Remove boxes with high overlap
  • Threshold, usually 0.5

\[
\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}
\]
Non-maxima suppression (NMS)

- Iterate over all detections
  - Pick the highest scoring box
- Find overlap
  - with all other boxes
- Remove boxes with high overlap
  - Threshold, usually 0.5
Non-maxima suppression (NMS)

- Iterate over all detections
  - Pick the highest scoring box
- Find overlap
  - with all other boxes
- Remove boxes with high overlap
  - Threshold, usually 0.5
Non-maxima suppression (NMS)

- Iterate over all detections
  - Pick the highest scoring box
- Find overlap
  - with all other boxes
- Remove boxes with high overlap
  - Threshold, usually 0.5
Non-maxima suppression (NMS)

• Iterate over all detections
  • Pick the highest scoring box
• Find overlap
  • with all other boxes
• Remove boxes with high overlap
  • Threshold, usually 0.5
Non-maxima suppression (NMS)

• Iterate over all detections
  • Pick the highest scoring box
• Find overlap
  • with all other boxes
• Remove boxes with high overlap
  • Threshold, usually 0.5
Evaluation

• How do we know if our approach is doing well?
Evaluation

• Intersection over union

Intersection over union (IoU)

\[
\text{IoU} = \frac{\text{size of } \text{yellow}}{\text{size of } \text{blue}}
\]

“Correct” if IoU ≥ 0.5
Sample IoU scores

0.905
0.532
0.391
0.143
0.0
Evaluation

• True positives
Evaluation

• True positives
• False positives
Evaluation

• True positives
• False positives
Evaluation

• True positives
• False positives
• False negatives

TP

FP

bird

wrong class

IOU < 0.5

no overlap
Evaluation

- True positives
- False positives
- False negatives
Evaluation

• Only one is correct
Evaluation

• Precision
  • Precision is the ability of a model to identify **only** the relevant objects.
  • It is the percentage of correct positive predictions and is given by:

\[
\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}
\]

• Recall
  • Recall is the ability of a model to find all the relevant cases
  • It is the percentage of true positive detected among all relevant ground truths and is given by:

\[
\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}
\]
Evaluation

• Sort all predicted boxes (for all images)
  • According to scores

• For each k (location) in the list
  • Compute recall and precision
Precision recall curve

• Which is better?
Average precision (AP)

**mAP:** \textit{average AP over multiple classes}
Questions?

Sources for this lecture include materials from works by Abhijit Mahalanobis, Alexei Efros, Sanja Fidler, and Fei Fei Li