Sliding Windows

Learning Objectives

After completing this lecture, you should be able to:

- Discuss sliding windows
- Explain sliding windows application
- Understand bonding box prediction

Outlines

- Sliding Windows
- Bonding Box Prediction

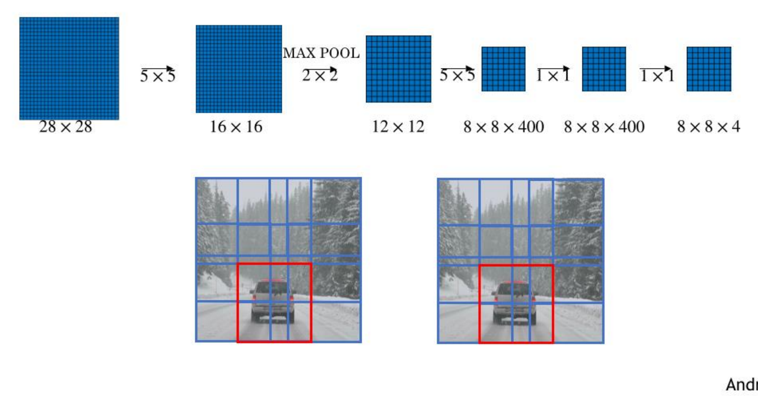
1. Sliding Windows

Definition:

- Sliding windows are a fundamental technique used in various areas of computer vision and signal processing. They involve moving a fixed-size window across an image (or other data types) to analyze different segments of the data sequentially.

Applications:

- **Object Detection:** Identifying the presence and location of objects within an image.
- **Image Classification:** Classifying different regions of an image into categories.
- **Signal Processing:** Analyzing segments of signals for patterns or anomalies.



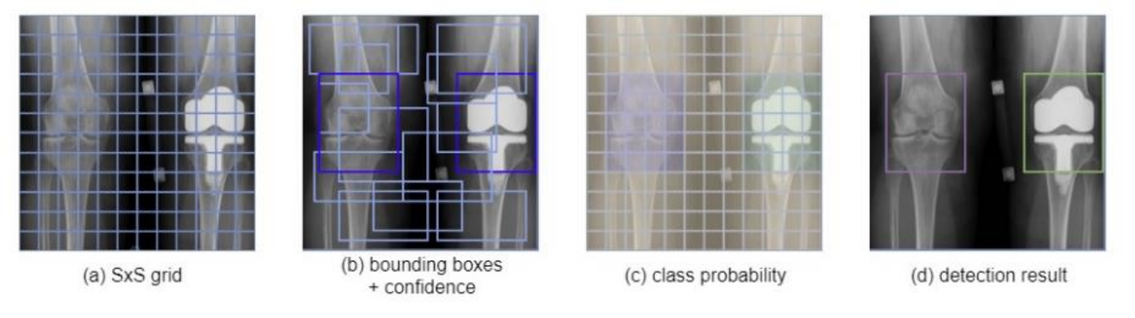
University of Pittsburgh

IEEE COMPUTER SOCIETY

1. Sliding Windows

Algorithm:

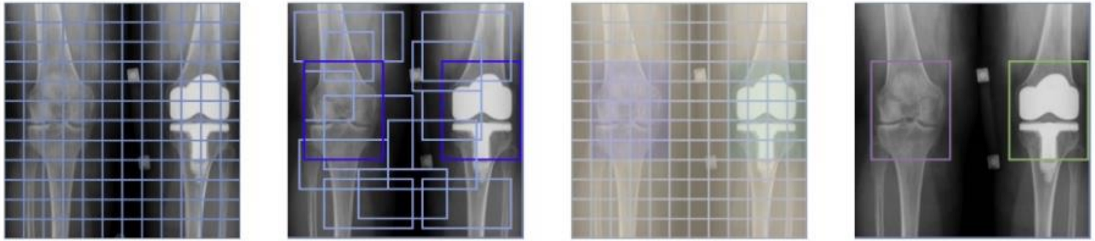
1. **Define the Window Size:** Choose the dimensions of the window (e.g., 32x32 pixels for an image).
2. **Set the Stride:** Determine the step size for moving the window (e.g., moving 1 pixel at a time or larger steps like 5 pixels).
3. **Slide the Window:** Move the window across the image starting from the top-left corner, shifting by the stride amount each time.
4. **Extract Segments:** At each position, extract the segment of the image that falls within the window.
5. **Process Each Segment:** Apply the desired processing to each segment (e.g., pass it through a classifier to detect objects).



1. Sliding Windows

Example:

- Suppose you have a 100x100 pixel image.
- Your sliding window is 10x10 pixels, and your stride is 5 pixels.
- The window starts at the top-left corner (0,0) and captures the segment from (0,0) to (10,10).
- The window then moves 5 pixels to the right, capturing the segment from (5,0) to (15,10), and continues this process across the entire image.



(a) SxS grid (b) bounding boxes + confidence (c) class probability (d) detection result

1. Sliding Windows

Advantages:

- **Simplicity:** Easy to implement and understand.
- **Flexibility:** Can be adapted to different data types and processing tasks.
- **Local Analysis:** Allows for the detailed examination of local regions within the data.

Disadvantages:

- **Computational Cost:** Especially with small strides, the number of segments can be very large, leading to high computational cost.
- **Redundancy:** Overlapping windows mean the same data is processed multiple times, which can be inefficient.
- **Scalability:** Not suitable for large-scale problems without optimization, due to the high number of operations required.

1. Sliding Windows

Optimizations:

- **Adjusting Stride:** Increasing the stride reduces the number of segments but may miss smaller objects or details.
- **Multi-scale Sliding Windows:** Using windows of different sizes to capture objects at different scales.
- **Feature Maps:** Using precomputed feature maps from neural networks to reduce the amount of data processed directly.

2. Convolutional Implementation of Sliding Windows

Overview of Convolutional Neural Networks (CNNs):

- **Structure of CNNs:**
  - **Convolutional Layers:** Apply a set of filters (kernels) to the input image, producing feature maps. Each filter detects specific features like edges or textures.
  - **Pooling Layers:** Reduce the spatial dimensions of feature maps, retaining the most important information and reducing computational complexity.
  - **Fully Connected Layers:** Act as classifiers, combining features extracted by convolutional and pooling layers to make predictions.

2. Convolutional Implementation of Sliding Windows

Difference between Traditional Sliding Windows and Convolutional Implementation:

- **Traditional Sliding Windows:**
  - Each window segment is processed independently.
  - Computationally expensive due to repetitive operations on overlapping windows.
  - Inefficient for large images or high-resolution data.
- **Convolutional Implementation:**
  - Uses a set of learned filters that slide across the entire image simultaneously.
  - Shared weights reduce the number of parameters and computational cost.
  - Can process the entire image in one pass, capturing spatial hierarchies of features.

2. Convolutional Implementation of Sliding Windows

Advantages of Convolutional Implementation:

- **Efficiency:** Shared weights and fewer parameters lead to faster processing.
- **Performance:** Ability to learn hierarchical features improves detection and classification accuracy.
- **Scalability:** Suitable for large-scale problems and high-resolution images.
- **End-to-End Learning:** Filters are learned directly from data, optimizing feature extraction for the specific task.

2. Convolutional Implementation of Sliding Windows

Example in Object Detection:

- **Traditional Sliding Windows Approach:**
  - A classifier is applied to each window segment to determine the presence of an object.
  - Multiple scales and aspect ratios may be used to capture objects of different sizes.
  - Example: Detecting faces in an image by applying a pre-trained face detector to each window segment.
- **Convolutional Implementation Approach:**
  - A CNN processes the entire image to produce feature maps.
  - These feature maps are used by additional layers to predict object locations and classes.

3. Bounding Box Prediction

Definition: Bounding box prediction involves identifying and localizing objects within an image by drawing rectangular boxes around them. Each bounding box is typically defined by its coordinates (x, y) for the top-left corner, along with its width and height.

Techniques for Bounding Box Prediction:

1. **Selective Search:**

- **Overview:** Combines hierarchical grouping of similar regions with exhaustive search to propose object regions.
- **Process:**
  - Start with initial segmentation of the image into superpixels.
  - Merge superpixels based on color, texture, size, and shape.
  - Propose regions (bounding boxes) around merged segments.
- **Advantages:**
  - Does not require training data.
  - Provides a moderate number of object proposals.
- **Disadvantages:**
  - Computationally expensive and slow.
  - May produce redundant proposals.

3. Bounding Box Prediction

2. **Region Proposal Networks (RPNs):**

- **Overview:** Part of the Faster R-CNN architecture, RPNs generate object proposals directly from feature maps.
- **Process:**
  - A small network slides over the convolutional feature map.
  - For each sliding window, it predicts multiple bounding boxes and objectness scores.
- **Advantages:**
  - Integrated into the CNN, allowing for end-to-end training.
  - Faster and more efficient than traditional methods like Selective Search.
- **Disadvantages:**
  - Requires training on large datasets.