Reyhane Shahrokhian 99521361

HomeWork8 of Computer Vision Course

Dr. Mohammadi

Q1:

1-1:

By using CNNs with sliding window, there is significant redundancy since many overlapping regions are processed multiple times and high memory is consumed. So, this approach is computationally expensive and also slower due to the repeated calculations for overlapping regions.

But in CNNs without sliding window, instead of dividing the image into smaller windows, the entire image is processed by the CNN in one pass which is more efficient since it avoids redundant calculations for overlapping regions.

There are also some advantages of using CNNs with sliding window. It is easy to implement. Sliding windows can be adjusted in size to handle objects of different scales.

1-2:

- Handling Multiple Classes in Each Grid Cell: YOLO divides the input image into an $S \times S$ grid, with each grid cell predicting multiple bounding boxes and their associated class probabilities. Each grid cell outputs class probabilities and confidence scores for the presence of an object, focusing on the most prominent object centered in the cell.

- Predicting the Probability of Classes with Bounding Boxes: YOLO predicts bounding box coordinates, confidence scores, and class probabilities for each bounding box. The final class-specific confidence score for each bounding box is computed by multiplying the bounding box confidence score by the predicted class probability.
- Anchor Boxes for Detecting Multiple Objects of Different Shapes and Sizes: Anchor boxes are predefined bounding boxes with specific aspect ratios and scales, allowing YOLO to predict bounding boxes for objects of various shapes and sizes. Multiple anchor boxes per grid cell enable the detection of multiple objects within the same cell, accommodating diverse object dimensions and overlaps.

Q2:

2-1:

YOLO:

- Architecture:

  YOLO uses a single convolutional network to predict bounding boxes and class probabilities directly from full images in one evaluation. The image is divided into an S×S grid, and each grid cell predicts B bounding boxes and confidence scores.

  YOLO is trained end-to-end, optimizing both bounding box predictions and classification.
- Speed:

  YOLO is designed for real-time object detection. Its architecture allows it to process images very quickly with fewer computations.
- Accuracy:

YOLO is known for its high precision, especially in detecting larger objects and it can struggle with detecting smaller objects and may produce more localization errors compared to SSD.

SSD:

- Architecture:

  SSD uses multiple feature maps at different scales to detect objects. This allows it to detect objects of various sizes more effectively. For each feature map cell, a set of default boxes (anchor boxes) with different aspect ratios and scales is defined, and predictions are made for each default box. SSD has separate predictions for object classification and bounding box regression.
- Speed:

  SSD is also designed for real-time detection, but its multiple feature maps and the use of default boxes add some overhead which is slightly slower than YOLO.
- Accuracy:

  SSD generally performs better at detecting smaller objects due to its use of multiple feature maps and default boxes. SSD offers a good balance between precision and recall, often achieving higher mean Average Precision (mAP) in benchmarks compared to YOLO.

Scenarios for Better Use:

- YOLO: Ideal for scenarios where speed is crucial, such as real-time video processing, autonomous driving, and interactive systems. Better suited for applications where the primary focus is on detecting larger objects with higher precision.

- SSD: Preferred in scenarios where achieving higher accuracy, especially for small objects, is more important than the fastest possible speed. Effective in applications where objects vary significantly in size and aspect ratio, such as surveillance and image-based search systems.

2-2:

Focal Loss is a modification of the standard cross-entropy loss designed to address the class imbalance problem often encountered in object detection tasks. Focal Loss adds a modulating term to the cross-entropy loss to focus learning on hard examples and down-weight the loss assigned to well-classified examples. Formula: $FL(p_t) =- \alpha_t(1 - p_t)^{\gamma} log(p_t)$

In datasets with class imbalance, the majority class typically dominates the loss, leading the model to be biased towards it. By reducing the loss contribution from easy (majority class) examples, focal loss prevents the model from being overwhelmed by the majority class. By adjusting the focus on hard-to-classify examples, focal loss improves the model's performance on minority classes, which are often underrepresented and harder to classify correctly.

Q3:

Non-Maximum Suppression (NMS) is necessary in object detection to remove redundant bounding boxes that overlap significantly, ensuring that only the most accurate bounding box for each detected object is retained. This helps to reduce noise and improve the precision of the detection results.

Steps:

1. Sorting: Sort the bounding boxes by their confidence scores in descending order.

2. Selection and IoU Calculation: Select the bounding box with the highest score and Compute the Intersection over Union (IoU) between this box and all other remaining boxes.
3. Suppression: Suppress all boxes that have an IoU greater than a predefined threshold with the selected box.
4. Iteration: Repeat the selection and suppression process for the next highest score box until no boxes remain.

The IoU threshold in NMS determines how much overlap is allowed between bounding boxes before one is suppressed. A lower IoU threshold results in more aggressive suppression, reducing false positives but potentially missing close objects. A higher IoU threshold retains more boxes, potentially keeping duplicates but reducing the chance of missing objects that are close together.

Q4:

First we should sort the boxes by their confidence scores:

1. (0.9, (50, 50, 100, 100)) ⇒ highest score box
2. (0.8, (55, 60, 105, 110))
3. (0.7, (100, 100, 150, 150))
4. (0.6, (45, 50, 95, 100))

Now we should compute the IOU of the selected box with others to check if it is larger than 0.5 to remove that box:

1. 1 and 2: $IOU = \frac{45\times40}{2\times50\times50-45\times40} = 0.56 \Rightarrow$ it should be removed.
2. 1 and 3: $IOU = \frac{0}{2\times50\times50} = 0 \Rightarrow$ it should not be removed.
3. 1 and 4: $IOU = \frac{45\times50}{2\times50\times50-45\times50} = 0.82 \Rightarrow$ it should be removed.

So the remaining boxes are:

1. (0.9, (50, 50, 100, 100))
2. (0.7, (100, 100, 150, 150))

Q5:

Center_x = bx × image_width = 0.5 × 416 = 208

Center_y = by × image_height = 0.6 × 416 = 249.6

Bbox_width = bw × image_width = 0.3 × 416 = 124.8

Bbox_height = bh × image_height = 0.4 × 416 = 166.4

Q6: