Week 2: Tuesday Paper 1: You Only Look Once: Unified, Real-Time Object Detection

explained by Understanding a Real-Time Object Detection Network: You Only Look Once (YOLOv1) - PyImageSearch

Supplemental: YOLOv7: The Fastest Object Detection Algorithm (2024) - viso.ai

1. What are the input and output in "Yolo (You Only Look Once)"? How is the output of the model evaluated? Be precise.

Ans: The YOLO model operates by taking a single image as input and processes it through a neural network to output a set of bounding boxes and associated class probabilities. The model divides the input image into a grid(SxS), and each grid cell predicts multiple bounding boxes and class probabilities. These predictions are then combined into a single output tensor.
The output of the YOLO model is evaluated using mean average precision (mAP). This metric considers both the precision, i.e., how many selected items are relevant, and recall, i.e., how many relevant items are selected, across all classes and averaged.

2. What are the different types of layers used? Give an intuitive description of what role each of them plays.

Ans: The YOLO model mainly consists of three types of layers,

Convolutional Layers: They are used for feature extraction from the input image. Different layers capture various aspects of the image, like edges, textures, and more complex patterns.

Maxpooling Layers: These layers are interspersed with the convolutional layers and are used for downsampling the feature maps. This reduces the spatial resolution of the representation, decreasing the amount of computation and parameters in the network, and helping to make the model more robust.

Fully Connected Layers: These layers come after the convolutional layers and are used for making the final predictions. They take the high-level features extracted by the convolutional layers and use them to predict bounding boxes and associated class probabilities.

We should also note that the activation function used in YOLO is LeakyReLU. It allows a small gradient when the unit is not active, which helps prevent the dying ReLU problem where neurons stop learning.

3. What is a bounding box? How is it defined and used in the paper?

Ans: A bounding box is a rectangular box that is used to define the position and extent of an object within an image. Each bounding box is defined by four parameters: the center coordinates (x, y) of the box, and its width and height.

To quote PyImageSearch that discussed this paper,

Each bounding box outputs five predictions: $\hat{x}, \hat{y}, \hat{w}, \hat{h}, \hat{C}$.

- The target (x, y) coordinates represent the center of the bounding box relative to the bounds of the grid cell, meaning the value of (x, y) would range between
 [0, 1]
. An (x, y) with value
 (0.5, 0.5)
 would mean the object's center is the center of a particular grid cell.
- The target (w, h) is the width and height of the bounding box relative to the entire image. This means that the predicted $(\hat{w}, \hat{h})$ will also be relative to the whole image. The width and height of the bounding box can be greater than
 1
.
- $\hat{C}$ is the confidence score predicted by the network.

Bounding boxes are crucial for localizing objects within an image, and they are used alongside class predictions to provide a complete description of the detected objects.

4. What is *Intersection Over Union (IOU)?* Why is it used in the paper? What would be the result if IOU was not used?

Ans: IOU is a metric used in the YOLO paper to evaluate the accuracy of the bounding boxes predicted by the model. IOU measures the overlap between the predicted bounding box and the ground truth bounding box. It is calculated as the area of overlap between the two boxes divided by the area of their union. IOU is used to quantify how close the model's predictions are to the actual object locations. If IOU was not used, assessing the model's accuracy in predicting the exact location and size of objects in images would be less precise, making it harder to evaluate and improve the model's performance.

5. How is LeakyRelu different from Relu? Why do you think LeakyRelu was used?

Ans: LeakyReLU is a variation of the ReLU activation function used in neural networks. The key difference is that while ReLU sets all negative values to zero, LeakyReLU allows a small, non-zero gradient when the unit is not active. Specifically, LeakyReLU has a small slope for negative values, instead of a flat zero. This is believed to help prevent the dying ReLU problem, where neurons can sometimes become inactive and stop learning entirely. The use of LeakyReLU in YOLO might be motivated by this advantage, as it can help maintain a healthy gradient flow through the network, especially in deeper architectures where the dying ReLU problem is more likely to occur.

6. What do you think are the key reasons that Yolo outperforms R-CNN?

Ans: The key reasons that YOLO outperforms R-CNN are,

Speed: YOLO processes images in a single pass, making it significantly faster than R-CNN, which proposes regions and then runs a classifier on each proposed region.

Detection as Regression: Unlike R-CNN, which uses a two-step process (region proposal and classification), YOLO frames object detection as a regression problem, directly predicting bounding boxes and class probabilities in one evaluation.

Spatial Constraints: YOLO imposes spatial constraints on bounding box predictions, which helps in reducing the number of false positives.

Global Context: YOLO looks at the entire image during training and test time, enabling it to implicitly encode contextual information about classes and their appearance.

7. What are some of Yolo's limitations? How would you address these limitations?

Ans: The limitations include,
Difficulty with Small Objects: YOLO struggles with detecting small objects or objects that appear in groups because of its spatial constraints.

Less Accuracy in Localization: YOLO tends to be less precise in object localization compared to some other methods.

We can address them by,

Improve Spatial Resolution: Modify the architecture to improve the spatial resolution of the feature maps, which can help in better detecting small objects.

Ensemble Models: Combine YOLO with other models that excel in areas where YOLO falls short, like models that are better at detecting small objects.

8. What questions do you have about the paper?

Week 2: Tuesday Paper 2: U-Net: Convolutional Networks for Biomedical Image Segmentation

explained by U-NET Paper Walkthrough, Unet Explained

Supplemental: UNet++: A Nested U-Net Architecture for Medical Image Segmentation

1. What are the input and output in “U-Net”? How is the output of the model evaluated? Be precise.

Ans: In the U-Net model, the input is a biomedical image, and the output is a segmentation map. The output of the model is evaluated based on how accurately these segmentations match the ground truth segmentations in the training data and Intersection over Union was used.

2. What are the different types of layers used? Give an intuitive description of what role each of them plays.

Ans:

Convolutional Layers: These layers extract features from the input image. They apply filters to the image to create feature maps that highlight different aspects of the image.

Max Pooling Layers: These layers reduce the spatial dimensions of the feature maps, making the model more computationally efficient and increasing its field of view.

Up-Convolution Layers: These layers upsample the feature maps to higher spatial resolutions, which is crucial for detailed segmentation.

Concatenation Layers: These layers combine the features from the downsampling path with the upsampled features, which helps the network to retain context and fine-grained details necessary for accurate segmentation.

3. What other evaluation metrics can be used for segmentation tasks? Give limitations of each of them as well.

Ans:
Dice Coefficient: This is similar to IOU and measures overlap but can be more sensitive to small structures.

Limitations: Might not adequately capture the performance in cases of extreme class imbalance.

Pixel Accuracy: This calculates the percentage of correctly classified pixels.

Limitations: Can be misleading in imbalanced datasets where one class dominates.

Sensitivity (True Positive Rate): It measures the proportion of actual positives correctly identified.

Limitations: Doesn't consider false positives, leading to a skewed view in imbalanced datasets.

Specificity (True Negative Rate): It measures the proportion of actual negatives correctly identified.

Limitations: Like sensitivity, it doesn't provide a complete picture when the classes are imbalanced.

4. What are the improvements of U-Net++ over U-Net?

Ans: The following improvements have been made to U-Net++ from U-Net,

Nested and Dense Skip Connections: U-Net++ introduces more intricate skip connections, enhancing feature fusion from different network levels.

Redesigned Skip Pathways: These pathways are redesigned to reduce the semantic gap between the feature maps of the encoder and decoder sub-networks.

Deep Supervision: U-Net++ incorporates deep supervision, allowing gradients to backpropagate through shorter paths, which improves training efficiency and segmentation accuracy.

5. Do you think skip connections are needed? Elaborate.

Ans: Skip connections are indeed crucial in architectures like U-Net. They help in merging low-level feature information from early layers with high-level features from deeper layers. This is particularly important in tasks like segmentation, where fine-grained details are necessary for accurate pixel-level classification. Without skip connections, the network might lose important spatial information during downsampling, leading to less precise segmentation results. Thus, skip connections play a key role in maintaining the integrity of spatial information throughout the network.

6. Is data augmentation used in the paper? If yes, why was it used?

Ans: Yes, data augmentation was used in the U-Net paper. The authors applied data augmentation due to the limited availability of annotated training images. They specifically used elastic deformations to increase the diversity of the training set. This approach allows the network to learn invariance to such deformations, enhancing its ability to generalize to new, unseen images. Data augmentation, particularly with elastic deformations, is useful in biomedical segmentation tasks as it mimics common variations in tissue images.

7. What questions do you have about the paper?