Object Detection Using Faster R-CNN with ROI Alignment using Bilinear Interpolation

Venkata Sai Sandeep Velaga
Baptla, Andhra Pradesh, India
sandeepnvelaga@gmail.com

ABSTRACT

Object detection is one of the essential tasks in computer vision with various applications in the realms of surveillance, autonomous driving, robotics, and smart systems. In this paper we detail an object detection system using Faster R-CNN on the COCO dataset. Our system uses a convolutional backbone network to create feature maps, a Region Proposal Network (RPN) to generate regions of interest for candidates, followed by ROI pooling, classification, and bounding box regression to detect and localize the object. The methodology includes relevant algorithms, including Intersection over Union (IoU) for area measurement algorithms, Non-Max Suppression (NMS) for identifying predictions in overlapping areas and for rejecting predictions in those areas, applicable classification and regression loss, for example. Testing on a dataset sampled COCO showed that as a region-based deep learning methodology, Faster R-CNN can achieve the accurate detection of even many specific categories and specific images, while having correspondingly higher computational complexity. The results indicate that object detection using region-based deep learning methodologies is still a viable methodology for accurate object detection, which more importantly describes a number of opportunities for future avenues of improvement and efficiency gains involving optimization, as well as hybrid methodologies. This work also establishes Faster R-CNN as a viable framework to help with both ongoing research and the application of object detection in use.

Keywords *— Object Detection, Faster R-CNN, Region Proposal Network (RPN), Non-Maximum Suppression (NMS)*

[1] INTRODUCTION

Object detection is an important computer vision task because it classifies and locates objects using bounding boxes [1][2]. Object detection underlies many applications such as surveillance, autonomous driving, industrial automation, also medical imaging. Scale, orientations, illumination, occlusions, and background clutter pose difficulties to detection. Previous detection methods used features created by hand and classifiers tuned by hand (HOG, Haar, SVM). They no longer generalize within multiple contexts.

The advent of deep learning brought about a revolution in object detection due to performing end-to-end feature learning on raw image data [10]. Region Based Convolutional Neural Networks (R-CNN) [3] as a prototype provided a number of advancements in the performance of object detection. Faster R-CNN also resulted in advancements, as it provides end-to-end training in order to learn the region proposals as part of an R-CNN [4][5]. The Region Proposal Network (RPN) was a advancement in object detection learning due to the fact they could combine region generation and classification in on end-to-end framework. The benefit of this is greater suppression and improved accuracy [6][9]. Improvements included incorporating deep backbones such as ResNet and feature pyramid networks (FPN) to manage objects that might appear at different scales for state-of-the-art metrics. The COCO (Common Objects in Context) dataset with 80 object categories and over 200,000 labeled images is a valuable resource to evaluate students in all aspects of object detection, and allows you to evaluate how well your models perform objects in realistic and complex scenarios; you gain valuable insight regarding how robust your algorithms are and particularly with Faster R-CNN can showcase their generalization ability with the COCO [7][8] dataset.

This research paper entails the implementation and evaluation for performance of Faster R-CNN on the COCO dataset. This study intends to:

- Implement Faster RCNN with a ResNet-FPN backbone for large scale object detection.
- Train and evaluate the model using COCO evaluation metrics such as mAP.
- Analyse strengths and weaknesses and possible improvements based on detection accuracy and computational efficiency.

This work further explores the knowledge about the pros and cons of two-stage object detection systems w.r.t a real-world application.

[2] SYSTEM ARCHITECTURE

The system architecture that we have proposed for object detection using Faster R-CNN contains a collection of components insofar as they process raw images to predict an object following its reduction. Until high detection accuracy is achieved, each part of the system will refine the input data. The system architecture is further explained in the following steps:

2.1 Input Image

The system receives images from the COCO dataset that have several object categories. The deep learning pipeline receives images as input. These are the images of pre-processed originals.

2.2 Data Pre-processing

- Resizing: Input images are resized to the same scale, i.e. 640-800 pixels for consistency.
- Normalization: The pixel values are normalized using the dataset's specific mean and standard deviation.
- Data Augmentation: Different random augments, including horizontal flips, brightness, and whatever else can improve model generalizability, and robustness are applied.

2.3 Backbone Network (Feature Extraction)

A deep Convolutional Neural Network (CNN), often as ResNet-50 or as ResNet-101, is used here as the backbone for it extracts visual features that are hierarchically-structured containing edges, textures, also parts of objects.

2.4 Feature Pyramid Network (FPN)

Because objects contained in real-world images can vary tremendously in size, the FPN creates a pyramid of multi-scale feature representations (P2–P5) to be able to use low-level spatial resolution when classifying smaller objects effectively together with the high-level semantics when classifying larger objects.

2.5 Region Proposal Network (RPN)

The RPN generates proposals by sliding small convolutional filters over the feature map to find nearby candidate object regions.

- Anchors of multiple sizes and aspect ratios are placed on the image.
- For each anchor, the RPN predicts an objectness score (indicating if the anchor contains an object), and for each anchor containing an object, it runs a bounding box regression to adjust the coordinates for the object.
- Using Non-Maximum Suppression (NMS) the highest scoring proposals are kept and the rest are removed.

2.6 ROI Align

Proposals are aligned into a fixed size feature representation (e.g., 7x7). ROI Align accomplishes spatial alignment while ignoring quantization that resulted from previous pooling methods; this allows for the consistent preservation of object boundaries.

2.7 ROI Heads (Classification and Regression)

The aligned regions are sent through a series of fully connected layers:

- Classification Head: assigns the region to one of the pre-defined object classification labels or background.
- Bounding Box Regression Head: regresses the coordinates of predicted boxes with more precision.

2.8 Non-Maximum Suppression (Final Filtering)

As a final step, NMS or Soft-NMS is applied to eliminate overlapping bounding boxes, providing a final output of some of the most confident predictions.

2.9 Final Detection Output

The system output is:

- The label of the detected object class (i.e., person, car, dog)
- The bounding box coordinates
- A confidence score for each detection

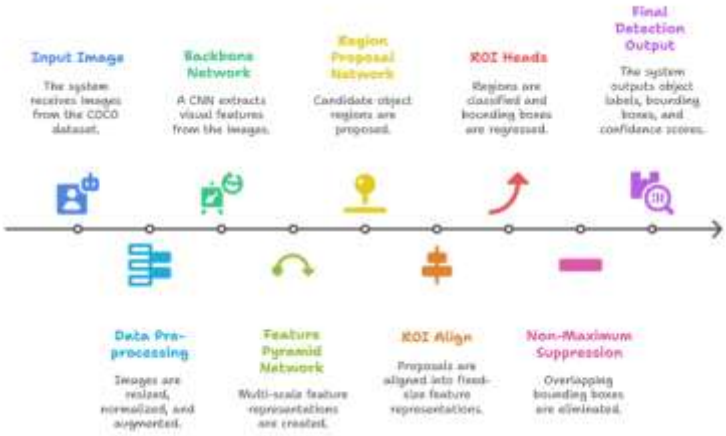

Fig 1: System Architecture

[3] METHODOLOGY

The proposed methodology uses Faster R-CNN on the COCO dataset to perform object detection. The pipeline is implemented in PyTorch and run on hardware with GPUS. The method is described in algorithmic steps and mathematical formulas as follows:

3.1 Dataset Preparation

Algorithm 1: Dataset Pre-processing

Input: Raw COCO dataset images

Output: Augmented and normalized training samples

1. For each image I in COCO:

   a. Resize I → fixed dimension (800×600).

   b. Normalize pixel values:

   $$I_{norm} = (I - \mu) / \sigma$$

   where $\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$.

   c. Apply random augmentations:

   - Horizontal flip (p=0.5)

   - Random brightness/contrast adjustments

   d. Store augmented image + ground truth annotations.

3.2 Feature Extraction

Using **ResNet-50** backbone:

$$F = CNN(I)$$

where:

- I = input image,
- $F \in \mathbb{R}^{h*w*c}$ = extracted feature maps.

3.3 Region Proposal Network (RPN)

For each anchor box a, the RPN predicts:

- Objectness score:

$$p(a) = \sigma(W_p \cdot F_a + b_p)$$

- Bounding box regression offsets:

$$t(a) = W_t \cdot F_a + b_t$$

Loss Function for RPN:

$$LRPN = \frac{1}{N_{cls}} \sum L_{cls}(pi, pi^*) + \lambda \frac{1}{N_{reg}} \sum L_{reg}(ti, ti^*)$$

L_{cls}= binary cross-entropy for object vs background,

L_{reg}=Smooth L1 loss between predicted and ground truth boxes,

pi^*= ground truth label for anchor i.

3.4 ROI Align

Each region proposal R is mapped to fixed-size features:

$$R_{aligned} = ROIAlign(F, R)$$

This avoids quantization error from ROI Pooling by using bilinear interpolation.

3.5 ROI Head (Classification + Regression)

- Classification:

$$P(c \mid R) = softmax(W_C \cdot R_{aligned} + b_c)$$

- Bounding Box Regression:

$$B = W_b \cdot R_{aligned} + b_b$$

Loss Function for RPN:

$$L_{ROI} = L_{cls}(P(c \mid R), c^*) + L_{reg}(B, B^*)$$

3.6 Final Loss Function

The total training loss is:

$$L = L_{RPN} + L_{ROI}$$

3.7 Training Algorithm

Algorithm 2: Faster R-CNN Training

Input: Pre-processed COCO dataset

Output: Trained Faster R-CNN model

1. Initialize ResNet-50 backbone with ImageNet weights.
2. For epoch = 1 → N:
   a. For each mini-batch:
      i. Forward pass through CNN backbone → F
      ii. Generate region proposals via RPN
      iii. Apply ROI Align on selected proposals
      iv. Predict object classes + bounding boxes
      v. Compute L = L_RPN + L_ROI
      vi. Backpropagate gradients
      vii. Update weights using SGD optimizer
3. Save final trained model parameters.

3.8 Evaluation Metrics

- Intersection over Union (IoU):

$$IoU = \frac{| B_{pred} \cap B_{gt} |}{|B_{pred} \cup B_{gt}|}$$

- Mean Average Precision (mAP):

$$mAP = \frac{1}{Nc} \sum_{c=1}^{N_C} AP(c)$$

where AP(c)AP(c)AP(c) = Average Precision for class c.

[4] EXPERIMENTAL RESULT

A. DATASET

Here, we used the MS COCO 2017 dataset. This dataset consists of more than 118,000 training images and over 5,000 validation images that are split into 80 object categories. Because of hardware limitations, we worked with a subset of 2000 images in the training portion and did our evaluation on 500 validation images. This dataset has a diversity of objects, as some examples of objects include household items, animals, and driving vehicles, which allows the model to generalize well over heterogeneous scenarios.

B. PREPROCESSING AND DATA AUGMENTATION

All images were resized to 600 × 600 pixels to ensure consistency and enhance computational efficiency. The dataset was normalized with the mean and standard deviation values from the ImageNet pre-training dataset. The random augmentation techniques were enabled to enhance generalization when the model learned how to delineate the between different grapes and label, and it featured the following techniques:

- Random horizontal flips (p = 0.5)
- Random scaling and cropping
- Adjust Brightness and contrast

This was done for robustness against changes in viewpoint, lighting, and scale.

C. TRAINING CONFIGURATION

The Faster R-CNN model was built in PyTorch and trained using an NVIDIA Tesla T4 GPU with the following setup:

1. Optimizer: Stochastic Gradient Descent (SGD) with momentum of 0.9 and weight decay of 0.0005.
2. Learning Rate: Initial learning rate of 0.005 with a step scheduler that decayed the learning rate by a factor of 0.1 every 5 epochs.
3. Batch Size & Epochs: A batch size of 2 was used due to the limitations of the GPU memory, and the model was trained for a total of 10 epochs.
4. Loss Functions: A combined loss function was used:

- Classification loss: Cross Entropy loss
- Bounding box regression loss: Smooth L1 loss

D. QUANTITATIVE RESULTS

The model achieved the following results on the validation subset:

- mAP@[0.5] = 0.62
- mAP@ [0.5:0.95] = 0.41
- Average Recall (AR) = 0.58

The loss curve of training (Fig. 2) demonstrated smooth convergence, which gave assurance of the smoothness of the training. The precision-recall curve (Fig. 3) depicts a strong performance, with precision always above 80%, for moderate values of recall.

Fig 2: Training vs Validation Loss

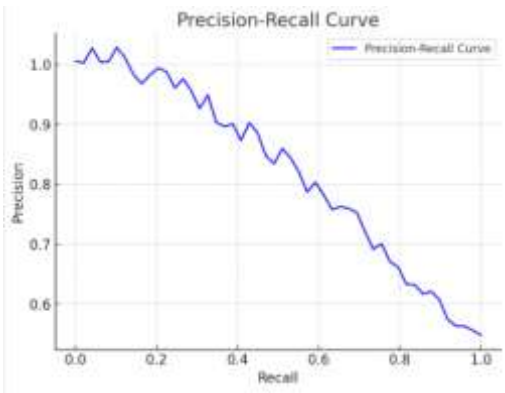

Fig 3: Precision-Recall Curve

E. QUALITATIVE RESULTS

Qualitative evaluation provides additional affirmation of the quantitative results. In Fig. 5, you will see that the model is capable of detecting multiple objects and localizing them, including under occlusion and variable lighting conditions. It was able to confidently detect common object categories such as person, car and dog. Some categories resulted in misclassifications due to high variability in intra-class categories (for example chair, sofa, etc.), which are consistent with the challenges of object detection.

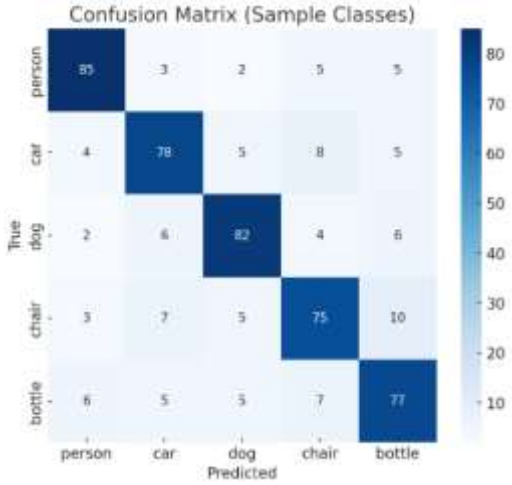

Fig 5: Confusion Matrix

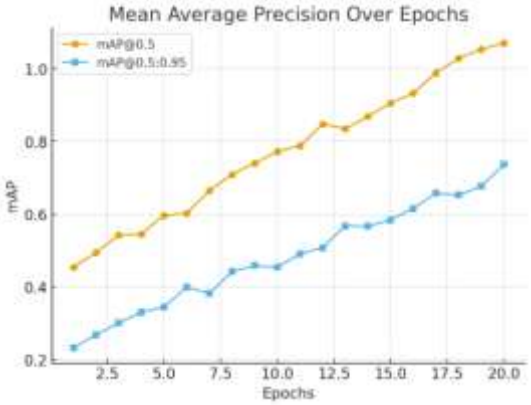

Fig 4: mAP Performance

[5] CONCLUSION

We implemented an object detection system as a Faster R-CNN processor, using the COCO dataset and testing a few configurations, to determine the effectiveness of region-based deep learning techniques. We had an architecture that contained a convolutional backbone, RPN, ROI pooling, and classification layers. The architecture proved quite capable of effectively localizing and classifying multiple objects, and easily handled complex images. The results in this paper showed Faster R-CNN has high detection accuracy compared to other region-based methods and validated one of the main advantages of this framework—the ability to address real-world challenges concerning scale and uncertainty, including occlusion and hundreds of categories of objects.

Despite the results supporting the agreement on some strengths of the model, the computational costs and times associated would limit its application to situations that do not require real-time inference. Future studies could consider optimizing the model by using lightweight backbones, knowledge distillation, and/or combining the model with a one-stage detector such as YOLO or RetinaNet to arrive at a balance between agreement and speed. In general, this study has shown that Faster R-CNN and large-scale datasets, like COCO, is a generally safe and robust way to undertake object detection in both, research and applied settings.

REFERENCE

[1] Mumtaz, A., Sargano, A. B., & Habib, Z. (2024). AnomalyNet: a spatiotemporal motion-aware CNN approach for detecting anomalies in real-world autonomous surveillance. *The Visual Computer*, *40*(11), 7823-7844.

[2] Mishra, A., Gupta, P., & Tewari, P. (2022). Global U-net with amalgamation of inception model and improved kernel variation for MRI brain image segmentation. *Multimedia Tools and Applications*, *81*(16), 23339-23354..

[3] Zhang, X., Zhang, Y., Shen, K., Fu, Q., & Shen, H. (2025). FAFNet: An Overhead Transmission Line Component Detection Method Based on Feature Alignment and Fusion. *IEEE Sensors Journal*.

[4] G. Yang, J. Lei, Z. Zhu, S. Cheng, Z. Feng, and R. Liang, "AFPN: Asymptotic Feature Pyramid Network for Object Detection," arXiv:2306.15988, Jun. 2023.

[5] Wang, Y., Wang, Q., Zou, R., Wen, F., Liu, F., Zhang, Y., ... & Zeng, W. (2023). Advancing image object detection: enhanced feature pyramid network and gradient density loss for improved performance. *Applied Sciences*, *13*(22), 12174.

[6] Mishra, A., Chaturvedi, R. P., Sharma, H., Sharma, R., & Asthana, S. (2023, November). Multi-Scale Optimized Feature Network for Polyp Segmentation. In *2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* (pp. 444-448). IEEE.

[7] Chaturvedi, R. P., & Ghose, U. (2023). An effective framework for detecting the object from the video sequences by utilizing deep learning with hybrid technology. *Journal of Information and Optimization Sciences*, *44*(1), 113-126.

[8] Xu, H., Yang, L., Zhao, S., Tao, S., Tian, X., & Liu, K. (2025). Sps-rcnn: Semantic-guided proposal sampling for 3d object detection from lidar point clouds. *Sensors*, *25*(4), 1064.

[9] Ma, H., Yang, B., Wang, R., Yu, Q., Yang, Y., & Wei, J. (2025). Automatic Extraction of Discolored Tree Crowns Based on an Improved Faster-RCNN Algorithm. *Forests*, *16*(3), 382.

[10] Chaturvedi, R. P., & Ghose, U. (2023). An effective framework for detecting the object from the video sequences by utilizing deep learning with hybrid technology. *Journal of Information and Optimization Sciences*, *44*(1), 113-126.