R-CNN - ASSIGNMENT 8

Mirko Morello
920601
m.morello11@campus.unimib.it

Andrea Borghesi
916202
a.borghesi1@campus.unimib.it

May 18, 2024

1 INTRODUCTION

In this assignment, we explore and analyze the performance of Faster Region-Based Convolutional Neural Network (Faster RCNN)

2 TECHNOLOGIES USED

2.1 Region-Based Convolutional Neural Network

Region-Based Convolutional Neural Networks (R-CNN) are a class of object detection algorithms that leverage the power of deep convolutional neural networks (CNNs) for visual recognition tasks [2]. These algorithms aim to localize and classify objects within an image by proposing regions of interest and evaluating those regions using a CNN. The R-CNN family consists of several iterations, each building upon the previous version, improving the accuracy, speed, and efficiency of object detection. The core components of R-CNN algorithms are:

Region Proposal: This step involves generating potential object bounding boxes or regions of interest within the input image. Various techniques, such as selective search or region proposal networks (RPNs), are used to propose these regions. Feature Extraction: A pre-trained CNN, such as VGGNet or ResNet, is used to extract features from the proposed regions. These features capture high-level visual information relevant for object recognition. Classification and Bounding Box Regression: The extracted features are fed into a classifier and a bounding box regressor. The classifier predicts the class of the object within the region, while the bounding box regressor refines the proposed bounding box coordinates to better fit the object.

The main variants of R-CNN include:

- R-CNN: The original Region-Based Convolutional Neural Network, which introduced the concept of using CNNs for object detection by extracting features from proposed regions [2].
- Fast R-CNN: An improved version that streamlined the architecture by sharing the computation of the convolutional features across all proposed regions, leading to faster processing times [1].
- Faster R-CNN: This iteration introduced the Region Proposal Network (RPN), which generates region proposals directly from the CNN features, eliminating the need for an external region proposal algorithm and further improving the speed and accuracy of the object detection process [3].

In this assignment we're going to concentrate on Faster R-CNN.

2.2 Architecture Details

Faster R-CNN incorporates several architectural enhancements that make it more efficient and accurate than its predecessors. The main differences and peculiarities of Faster R-CNN are:

- **Region Proposal Network (RPN)**: Faster R-CNN introduces the Region Proposal Network (RPN), which is a dedicated convolutional neural network that generates region proposals directly from the input image. The RPN shares full-image convolutional features with the object detection network, enabling nearly cost-free region proposals [3].
- **Multi-Task Learning**: Faster R-CNN adopts a multi-task learning approach, where a single, unified network is trained for both region proposal generation (RPN) and object detection tasks. This design

allows for end-to-end training, enabling the sharing of representations between the two tasks [3].

- **Shared Convolutional Features**: In Faster R-CNN, the convolutional layers are shared between the RPN and the object detection network. This sharing of convolutional features across the two tasks leads to significant computational savings and improved efficiency [3].

- **Multi-Head Architecture**: The object detection network in Faster R-CNN consists of two parallel heads: a classification head and a regression head. The classification head predicts the class of the object within the proposed region, while the regression head refines the bounding box coordinates for accurate object localization.

- **End-to-End Training**: Faster R-CNN enables end-to-end training of the entire network, including the shared convolutional layers, RPN, and object detection heads. This joint training allows for better optimization and improved performance compared to separate training stages used in earlier versions.

3 Data

Our models have been trained on a dataset composed of Uno cards images, for each image we have an xml file containing the information of 3 ground truth bounding boxes and 3 classes that correspond to the seed of 3 different cards. The challenge is represented by the backgroud of each of them which is continuously changing in form, colour, scale and shape.
The objective of the network is to identify the bounding boxes of the 3 cards in each image and classify the seed of the cards between 16 different classes.

4 Approach

We commenced by training the predictor using the provided code. Since the training data was clearly synthetic, the predictor could rapidly discern its patterns, such as the cards appearing in groups of three and being equally spaced. Despite these repetitive features in the dataset, the provided code ensured proper data augmentation through techniques like flipping, rotating, blurring, and scaling, leading to very high performance. Consequently, only a few training epochs were required to achieve low loss scores during the validation procedures.

After adequately training the predictor, we evaluated its performance and conducted an analysis. To gain more insightful information beyond the loss alone, we computed three scores for each prediction that satisfied a confidence threshold (returned by the prediction procedure). These scores represented how well the prediction aligned with being a true positive, false positive, or false negative concerning the ground truth, determined by assessing the degree of overlap between the predicted and ground truth bounding boxes. We then used these scores to identify which labels were the hardest to identify for the network.

4.1 Metrics and Validation

We established a criterion that a prediction would be considered a true positive if its Intersection over Union (IoU) with the ground truth bounding box exceeded 80% [4]. Predictions not meeting this threshold were classified as either false positives or false negatives, depending on which of these two values was higher. After computing these scores for each prediction, we could identify the labels that proved most challenging by analyzing the mean and standard deviation of true positives, false positives, and false negatives for each label.

5 Results

As expected the performance of the predictor are very high as shown in Table 1 with a precision of 96%.

Metric	Value
True Positives	2459
False Positives	90
False Negatives	143
Precision	0.96
Recall	0.95
F1 Score	0.95

Table 1: Evaluation Scores

Figures 3, 4 and 5 indicate that label 13, representing the block card, was among the most difficult to identify correctly. This can be attributed to its visual similarity with both the digit zero and the number eight, leading to potential confusion. However, upon analyzing the backgrounds of the predictions with the highest error rates, no discernible pattern emerged linking the incorrect labeling to specific background characteristics it was a rare occurrence, an example can be seen in picture 1. By checking the worst prediction according to the IoU score, it is clear that many incorrect predictions are due to the model finding the label at the bottom right corner of the card, rather than the top left one, which is the only one marked as the ground truth, as can be seen in picture 2. This suggests that the obtained scores are higher than the reported one, as this is just a shortcoming of the dataset itself.

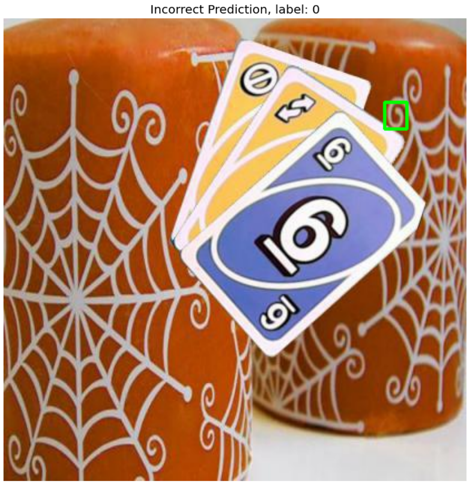

Figure 1: Example of rare background mismatch

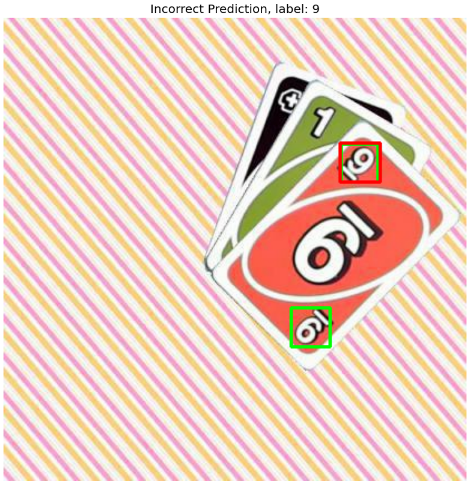

Figure 2: Example of most common mismatch, red box is the ground truth, green box is the match found by the model, in this image two matches can be seen, the one around the ground truth is correct, the other at the bottom is not.

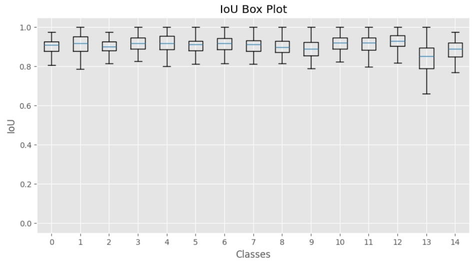

Figure 3: IoU score for each predicted label

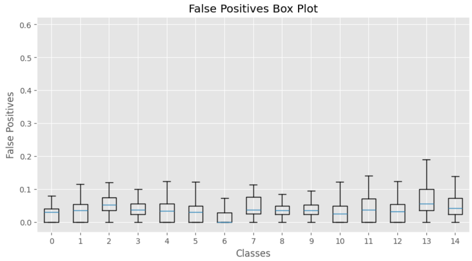

Figure 4: False positive score for each predicted label

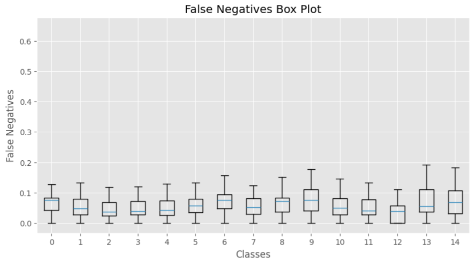

Figure 5: False negative score for each predicted label

References

[1] Ross Girshick. Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[2] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[4] Hamid Rezatofighi, Niki Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019.