Object Detection with Reinforcement Learning

Manssouri Ayyoub & Boudellah Omar

January 12, 2024

Abstract

In this project, we address the object localization task in computer vision by implementing a novel active object localization algorithm based on deep reinforcement learning. Leveraging powerful computational techniques, particularly convolutional neural networks (CNNs), we explore the dynamic decision-making nature of object localization. Our experiments delve into hyperparameter settings and network structures, providing insights into the model's performance. A comparative evaluation against a random baseline reveals the algorithm's efficacy, and we discuss potential drawbacks to guide future research. Index Terms—object localization, neural networks, deep reinforcement learning, Markov decision process.

Keywords: object localization · neural networks · deep reinforcement learning · Markov decision process

1 Introduction

In the dynamic realm of computer vision, the convergence of object detection and reinforcement learning heralds a transformative paradigm, poised to elevate the capabilities of intelligent systems. Object detection, a fundamental tenet of visual perception, entails the nuanced task of not only recognizing an array of objects within images but also precisely localizing them through bounding boxes. This article navigates the uncharted territory of "Object Detection with Reinforcement Learning," delving into the symbiotic relationship between deep learning and adaptive decision-making. With a specific focus on applying this fusion to the venerable PASCAL VOC (Visual Object Classes) dataset, a cornerstone in the field, our exploration extends beyond theoretical frameworks. As we unravel the intricacies of algorithm design and implementation, this article seeks to establish a comprehensive understanding

of the synergy between reinforcement learning and object detection, offering both theoretical insights and practical implications for researchers and practitioners at the forefront of contemporary computer vision applications.

2 Dataset

The PASCAL VOC2012 (Visual Object Classes) dataset is a benchmark in the field of object recognition and detection. It was introduced as part of the PASCAL Visual Object Classes Challenge, which aimed to stimulate progress in object detection algorithms. The VOC2012 dataset specifically focuses on 20 object categories, including common classes such as person, car, bird, and chair. The dataset comprises images collected from a variety of sources, encompassing both everyday scenes and specific object instances.

Key features of the PASCAL VOC2012 dataset include:

Object Categories: The dataset covers a diverse set of object categories, making it suitable for evaluating algorithms across different classes.

Annotations: Each image in the dataset is annotated with bounding boxes around the objects, along with class labels. This facilitates the training and evaluation of object detection algorithms.

Image Variability: The dataset includes images with variations in scale, pose, lighting conditions, and occlusions, reflecting real-world challenges encountered in object detection tasks.

Train-Val-Test Split: It is divided into training, validation, and test sets, allowing researchers to train their models on a subset, tune parameters on the validation set, and evaluate performance on the test set.

Challenges: The PASCAL VOC dataset is associated with annual challenges, where participants submit their object detection algorithms and are evaluated based on predefined metrics, including precision, recall, and average precision.

Example of image of DataSet :

3 Related work

3.1 RCNN

The R-CNN model, initiates the object detection process by generating region proposals through the selective search algorithm. This algorithm strategically suggests regions by hierarchically grouping similar areas based on color, texture, size, and shape compatibility. Subsequently, the model constructs a feature vector for each proposed region, representing the image in a significantly reduced dimension using a pre-trained Convolutional Neural Network (CNN).

The CNN, pretrained on extensive datasets such as ImageNet, undergoes

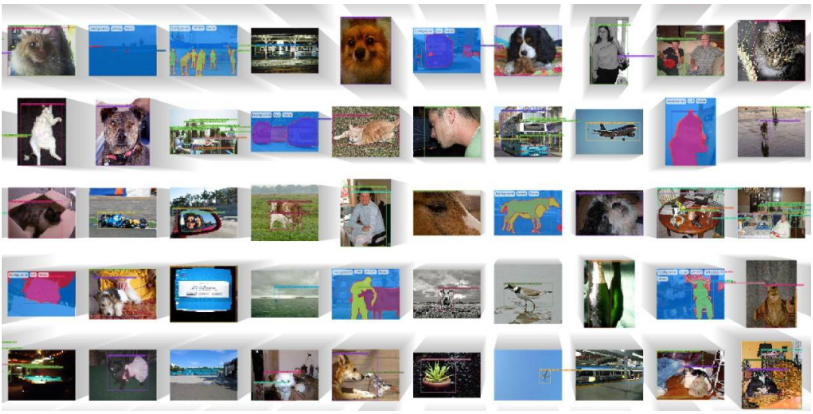

Fig. 1 This is an example of images of PASCAL VOC2012

weight adjustments to fine-tune its performance. For the network to process the image, resizing is necessary to align with the CNN input specifications. Following the extraction of features from the region proposals, an SVM classifier comes into play for object classification within those regions.

It's noteworthy that a distinct SVM is assigned to each object class, resulting in n outputs for a single feature vector, where n signifies the number of distinct objects targeted for detection. The output manifests as a confidence score, indicating the level of certainty that the feature vector accurately represents its assigned class. Both the R-CNN and our implemented approach share a limitation in that they require the use of a pretrained CNN for feature extraction, as simultaneous training of SVM (or, in our case, DQN) and CNN classifiers is not feasible.

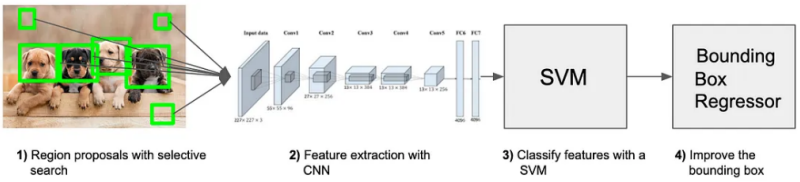

Fig. 2 The R-CNN system

4 Algorithm design and implementation

4.1 States

It is expected that the state space of this prob- lem formulation should be very large due to the complexity of dealing with the image data. As a result, generalization with feature extraction is necessary to represent the states efficiently

and effectively. In this project, the states are constructed with a feature vector o, which summarizes the image information of the current found bounding box, and a history vector h of previous taken actions. The feature vector o is extracted by feeding the current bounded region of the image into the pre-trained VGG-16 network. Since the VGG-16 network requires for inputs with the same dimension, all the attended regions are transformed into a 224x224 image with the OpenCV library. The network then deals with the pixel values of the attended region and extracts corresponding features out of it. The output feature vector is set to be a 4096-dimensional vector. The action history vector h is a binary vector that encodes the actions applied in the past with one-hot encoding method. This vector stores 10 previous actions taken by the agent, and as we have 9 possible actions, the one-hot encoded history vector will have 90 dimensions. By adding this history vector as part of the state representation, the agent is expected to be informed about what has happened in the past and avoid getting stuck in repetitive search cycles.

4.2 Actions

As stated above, the actions of this MDP are mainly the geometric transformations of the bounding boxes. There are nine transformation actions and one terminal action in this MDP formulation . The eight transformation actions can be divided into four categories, which modify the horizontal level, vertical level, scale and aspect ratio of the bounding box, respectively. A figure of all actions can be seen. The transformation actions are executed by calculating a discrete change to the box by a relative factor , using equations 3.

$$\alpha_w = \alpha(x_2 - x_1) \quad \alpha_h = \alpha(y_2 - y_1) \tag{3}$$

Fig. 3 Illustration of the actions in MDP. Image credit

Where x1, y1, x2 and y2 are the coordinates of the top- left corner and the right-bottom corner of the current placed bounding box. With this calculated discrete change, the action can be carried out by adding this value to or subtract this value from the corresponding coordinates.

4.3 Rewards

The reward function needs to be defined corresponding to the actions taken. In the proposed formulation, the reward function R is given based on the

improvement of the Intersection-over-Union (IoU) between the predicted and the ground truth bounding box, during the transition between states. Thus, for non-terminal actions, the reward is set to 1 if the state transition improves the IoU and -1 otherwise, as below formula:

$$Ra(s, s0) = sign(IoU(b0, g) \;\; IoU(b, g)) \quad (5)$$

5 Evaluation

As mentioned in earlier sections, for each class of objects, we trained our agent for 50 epochs of maximum 20 steps on only 100 images that contain that object. This insufficient amount of training caused some biased behaviors in our agent for some categories. For instance for bicycle class we realized most of the predicted boxes have a similar vertical form regardless of the shape of target object in image. This behavior can be due to having several objects with vertical boxes in the training data, that caused our agent to become biased and deform the window to a vertical form regardless of the shape of target in the test images. The average IoU in this scenario was approximately 0.31

figure 1 :

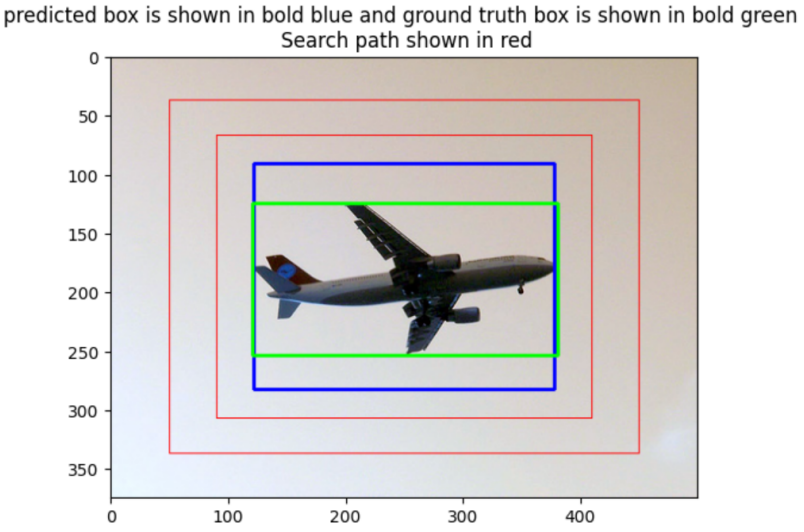

Fig. 4 Image test of our model

6 Summary and perspectives

In this project, we have pioneered an active object localization algorithm employing deep reinforcement learning within the domain of computer vision. By harnessing the capabilities of convolutional neural networks, we redefine object localization as a dynamic decision-making process, showcasing its adaptability and intelligence. Rigorous experiments involving hyperparameter tuning and network structure investigations underscore the model's effectiveness, outperforming a random baseline. This work not only contributes a novel solution to object localization but also provides valuable insights for future enhancements, encouraging ongoing research in the evolution of adaptive computer vision methodologies with broad applications in diverse domains.