Measurement: Sensors 31 (2024) 101025

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Faster region based convolution neural network with context iterative refinement for object detection

Kishore Anthuvan Sahayaraj K. [a,*], Balamurugan G. [b]

[a] Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur, 603203, Tamil Nadu, India
[b] Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur, 603203, Tamil Nadu, India

ARTICLE INFO

Keywords:
FR-CNN
Object detection
Context iterative refinement
SSVM
Joint score

ABSTRACT

In this paper, proposed a novel method to improve the localization precision of identified objects. We present a framework for iteratively enhancing image region recommendations to meet ground truth values in this research. The Faster R–CNN (FR-CNN) seems to be an object recognition deep convolutional network. It gives the user the impression that the network is cohesive and single. The network can provide accurate and timely predictions about the whereabouts of a range of objects. We first build a unified model based on rapid predictions to relocate inaccurate area recommendations. Because the emphasis is on object detection, it may be utilised with a wide range of datasets and is compatible with various FR-CNN architectures. Second, we focus on the application of the joint score function to a variety of picture features. This joint score function depicts the location of the concealed object concerning other objects. The picture data and an updated structured production loss function are the only two inputs that influence the parameters of the joint scoring function. The join-score function and iterative context refinement (CIR) are used to generate our final unified model, which is then classified using Smooth Support Vector Machine (SSVM). We measured accuracy using the mean average precision after training FR-CNN + CIR and SSVM on a low-cost GPU using the PASCAL VOC 2012 dataset. Our results are 3.6 % more exact than rival deep learning algorithms on average.

1. Introduction

The process of object detection is often regarded as a challenging endeavour because to its status as one of the "classical issues" of computer vision. The fact that it requires both identifying and categorising portions of an image is one of the primary reasons why object detection is considered to be problematic. In some applications, such as full image categorization, the locating component is unnecessary [1]. In order to identify an item, it is necessary to have some concept of the possible location of the object as well as an understanding of how the image is divided. Object detection has already established itself as a key research topic and the primary emphasis in the field of computer vision. This field of study has potential applications in areas such as robotics, autonomous cars, video surveillance, and the identification of pedestrians [2]. The process of detecting objects is progressing from the identification of a single item to the recognition of several objects. Every item category has its own unique quality that may be used to assist in the categorization of the category [3]. For instance, all circles have a round shape. This unique characteristic is used for object class recognition. In general, techniques for object recognition may be broken down into two categories: those based on machine learning and those based on deep learning. Deep learning approaches are those that are able to accomplish end-to-end object recognition without expressly specifying characteristics. These techniques are often based on CNN and are part of the field of deep learning [4]. The activity of detecting objects, which has a significant number of applications in our way of life. However, it requires a significant amount of time and effort to truly categorise the content of a particular image area each time, and there are limited process capabilities that an agent can spend on classification [5]. In order to swiftly determine which parts of an image need to be processed in more depth and which parts may be disregarded, humans employ a process that is referred to as visual attention. This enables the United States of America to assimilate the vast amount of visual input and to make effective use of the capabilities of our sensory system [6].

A significant issue in computer vision research is object detection. Convolutional neural network-based object recognition, such as that used in Ref. [6] has drawn a lot of attention recently. However, to increase the speed of object localization, CNNs are frequently paired with

* Corresponding author.
E-mail addresses: kishorea1@srmist.edu.in (K.A.S. K.), balamurg1@srmist.edu.in (B. G.).

https://doi.org/10.1016/j.measen.2024.101025
Received 4 January 2023; Received in revised form 26 June 2023; Accepted 2 January 2024
Available online 8 January 2024

object recognition techniques like those described in Ref. [7], which initially produce a limited number of limitless class region recommendations. When it comes to scale invariant, the Regions of Interest (RoI) produced by the object proposal approaches are erroneous. When this RoI is applied to CNNs, some of the inexact RoI may be wrongly recognized in later stages, which affects the performances [8]. In Fig. 1, a bike is encompassed by a bounding box. To address this issue, numerous solutions have been put forth. To obtain the most accurate RoI feasible, for instance pooled many item proposal models [9]. However, it is unsuccessful when dealing with a sizable number of highly overlapping RoI. These methods, when used with various datasets, necessitate retraining and off-line learning for the training.

In this research, a novel technique for iteratively enhancing bounding boxes such that they reflect ground truth is described. This is accomplished by extending it to contextual reasoning [10]. In order to convert erroneous region proposals into their precise positions and sizes, we first develop a unified model with quick approximations. Since it is data-focused, learning-free, and compatible with various FR-CNN architectures, it may be used with a variety of datasets [11]. Second, we use a technique called the joint score to evaluate a lot of potential candidates in the images. This combined scoring algorithm takes an object's relative location into account when assigning points to it. The only two inputs that have any influence on any score function parameter are the image data and the refinement of a structured output loss function [12]. In the end, we decided to combine iterative refinement with the join score function to create one unified model.

Our Motivation is to improve the accuracy, speed, and efficiency of object detection systems. Object detection plays a critical role in various applications, including autonomous driving, surveillance, and robotics. However, traditional object detection methods may struggle to achieve real-time performance and accurate localization of objects in complex scenes. The Faster R–CNN with context iterative refinement aims to address these limitations and provide more robust and efficient object detection capabilities.

The following outline constitutes the framework of the paper. In Section II, information is provided on the works that are linked. In Section III, the process is broken down and described. The findings of the comparison are presented in Section IV. The last section of the paper is section V.

2. Related works

In digital images or videos, object detection looks for instances of objects belonging to particular classes. An object detection model analyses a video or image and identifies what items are there and where they are located. In addition to several other applications, it is crucial for self-driving vehicles and video surveillance. Convolutional neural networks recently outperformed classical algorithms at vehicle detection [13]. The two primary kinds of CNN-based object detection algorithms are regression-based algorithms and region proposal-based algorithms (R–CNN [[14], Fast R–CNN [15], and Faster R–CNN [16]).

Learning at a deep level is required in order to recognise objects inside images, since this task cannot be accomplished without this technique. Sometimes also referred to as deep structured learning or ranked learning, deep learning is a kind of machine learning [17]. As opposed to task-specific algorithms, it belongs to a more general family of machine learning approaches that are centred on the concept of learning representations. One of the most famous disciplines in deep learning is image object analysis [18]. Image object analysis Deep learning is gaining a lot of interest because it is a particularly sophisticated kind of learning that has the potential to be very beneficial for global applications. This is one reason why it has drawn so much attention [19]. Deep models may be created to function as both a classifier and a regression device, and they do not need any special hand-engineered features. Therefore, the technique of deep learning presents a significant opportunity in the object identification field.

A. Region Proposal Based Algorithms.

R–CNN. The region-based convolutional neural network, or R–CNN, uses the "region proposal" technique to extract regions from an image. There are three steps that result in these regional proposals. First, it uses a targeted search to develop a region suggestion. An algorithm that is greedy combines similar regions to retrieve the second feature. The third SVM classification has been completed [20]. Because no learning is done and training to recognise the regions takes longer with selective search, there are problems.

A Region Proposal Network (RPN), is a network that is completely convolutional and can predict object limits as well as objectness scores for each place simultaneously. The RPN receives comprehensive training to develop high-quality region proposals at every stage of the process [21]. Fast R–CNN advances the work of R–CNN by speeding up training and testing while increasing detection accuracy. This step takes place at the very end. The problem with this approach is that it takes a long time to prepare Faster R–CNN. Faster R–CNN is suggested as a solution to the issue with Fast R-region CNN's proposals since it uses neural networks to create regions rather than the former method [22]. It is made up of two modules. The first component, a region proposal network (RPN), suggests regions for the fast R–CNN detector to examine in the second component. With numerous anchors at each sliding window point, a tiny network with RPN is slid across the convolution feature map. The RPN produces bounding boxes (region suggestions) and projected classes.

RPNs are designed to do one thing and one thing only: make suggestions for objects. Images are taken as an input by the RPN, and it generates a series of rectangular object suggestions as its output, along with the likelihood of there being an item in each proposal. RPN extracts a feature map with the assistance of a CNN and then adds a further convolutional layer on top of that map. Following the Convolutional layer is an activation function known as a Rectified linear unit (ReLU), which offers nonlinearity and speeds up the pace at which the network

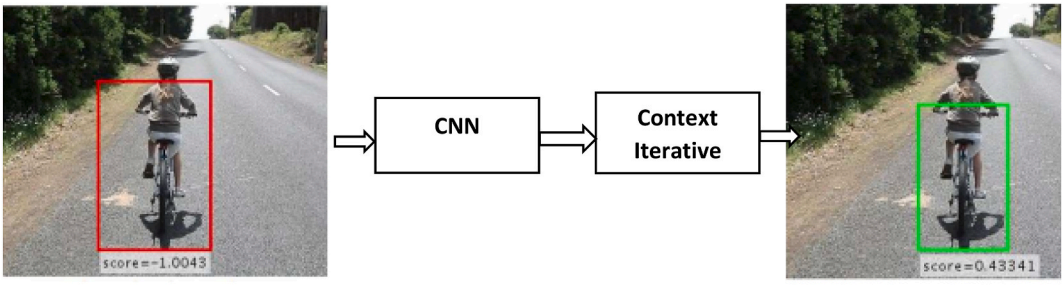

Fig. 1. Optimizing a CNN's localization via iterative processes [29].

converges [23]. This Conv, which is followed by the Relu notation, is responsible for mapping the characteristics of each sliding window to a vector. After that, the regression & softmax layers get this mapping as input. The regression and softmax layers, in turn, provide predictions about the various coordinates of the bounding boxes, as well as the likelihood that an item will be found in each of those bounding boxes [24].

B. Regression Based Algorithms.

The SSD and YOLOYou Only Look Once (YOLO) transforms the detection problem into a regression problem, yielding the Single Shot MultiBox Detector (SSD), which is based on a pre-defined grid and uses a CNN to sequentially forecast class confidence and bounding boxes. This makes it much faster than methods based on region proposals.

The Selective Search suffers from the problems of being an offline method and having a high level of computing complexity. The region proposal network enters the scene at this point. The Region Proposal Network was first presented in Faster R–CNN in order to make use of a limited network in order to create region suggestions [25]. RPN is equipped with a classifier that provides a probability estimate of the area. In addition to that, it comes with a regressor that gives back the parameters of the bounding boxes [26].

In Machine Learning, the forecasting of future values is accomplished with the use of regression methods. Through the application of regression, a broad range of potential future values may be forecasted based on the previous data and input data. The term "label" refers to the variable that is being predicted in machine learning, and regression is used to characterise the connection that exists between the label and the data points [27]. A kind of supervised learning in machine learning known as regression helps map out a prediction link between the labels of data points and the points themselves. Linear, polynomial, logarithmic, stepwise, and other forms of regression algorithms are among the most used in machine learning. Continue reading if you're interested in learning more concerning the most common regression methods.

A statistical approach known as simple linear regression employs the use of a straight line in order to determine the nature of the connection that exists between two variables. Finding the slope and intercept of the line allows one to define the line and reduce mistakes caused by regression [28]. This is how the line is drawn. With pairwise and unary potentials, the procedure in this model creates bounding boxes. The pairwise potential model has the spatial arrangement of RoI in the image. The structured surrogate loss can be adjusted with the help of the joint score function model.

From the above literature studies, the following are challenges in object detection methods. i. Faster R–CNN is a computationally expensive algorithm, especially for large images. This can make it difficult to use in real-time applications. ii. Faster R–CNN is not as good at detecting objects with a variety of shapes. iii. The problem statement involves improving the localization accuracy of object detection models.

This literature review explores the existing research on Faster R–CNN for object detection, focusing on techniques that enhance its speed and accuracy.

3. Proposed methodology

A single model with a quick approximation was built to correct inaccurate region predictions into their precise locations and sizes. As a result of being data-driven, learning-free, and adaptable to various FR-CNN architectures, it may be used with various datasets. Second, we apply the joint score function to numerous candidates in the image graphs. The image data and refining a structured output loss function are the only inputs affecting any score function parameter. The main objective of this research is to improve accuracy of object detection, optimise the computational efficiency that can meet the requirements of real-world applications.

3.1. Faster RCNN algorithm and the region proposal network

This article begins by briefly overviewing the region-based convolutional neural network, also known as R–CNN. R–CNN is the first attempt at developing an object identification model that needs to be extracted using a pre-trained CNN. The next step is a speedy assessment of the Fast R–CNN, which, while it is quicker than the R–CNN, regrettably ignores the process by which region ideas are developed. This problem is eventually overcome by the FR-CNN, which constructs a region-proposal network. This network can create region proposals, which are then given to the detection model (the Fast R–CNN), which searches for objects as shown in Fig. 2.

The RPN performs a binary classification job, which outputs either "object" or "no object," and then generates a score that shows whether the portion of the image in question (the anchor) has a background or a foreground object. This value is referred to as the objectness score. The score for all of the classes into which we would like to place the items is produced by the classifier network and shown.

Algorithm steps for FRCNN:

1. *Initialize the parameters.*
2. *Initialize and process every frame.*
3. *Reading and returning every frame with its coordinate points.*
4. *Creating anchor box from every frame.*
5. *Passing anchor box frame to convert as input.*
6. *Extracting anchor and bounding box to detect object.*
7. *for i = 1 (number of object detections in a range);*
    a. *anchor_box → bounding_box (0,0, i)*
    b. *bounding_mask →masks [i];*
        i. *left_box → int [box * masks [i]];*
        ii. *top_box → int [box * masks [i]];*
        iii. *right_box → int [box * masks [i]];*
        iv. *bottom_box → int [box * masks [i]];*
    c. *square_box → int [box, [start (x,y)], [end(x, y)];*
8. *detecting object in boundary_box [i];*

In the algorithm 1, have been developed so far, the object detection component of the algorithm has been considered independently from the region proposal components. A classifier was given the outputs of the region proposal techniques, and the effectiveness of the classifier was dependent on the effectiveness of the region proposal method. The study that was done on Faster RCNN suggested using a unified strategy for both of the tasks that needed to be completed, which meant that the convolutional features used by the region proposal and the classifier were the same.

A Context Iterative Refinement

Fig. 3 depicts the R–CNN model, which is a quicker. A new bounding box Ri under control in close proximity to the box that was there before (Ri, f (Ri)). It is situated in the middle of the ROI pooling layer and the convolution layer of the neural network. The divide as well as conquer model taking over and continues its iterative refining process when the quicker R–CNN is unable to locate the necessary bounding box. First, the preferred bounding box is searched over an area of interest. Next, the area of interest is recursively broken into searches over increasingly smaller subregions.

Let Ri be the first bounding box in each iteration. Score f (Ri) represents the probability that the ith region of interest is located in the region Ri. The vector [Ti, Bi, Li, Ri] is also known as the rim vector. The

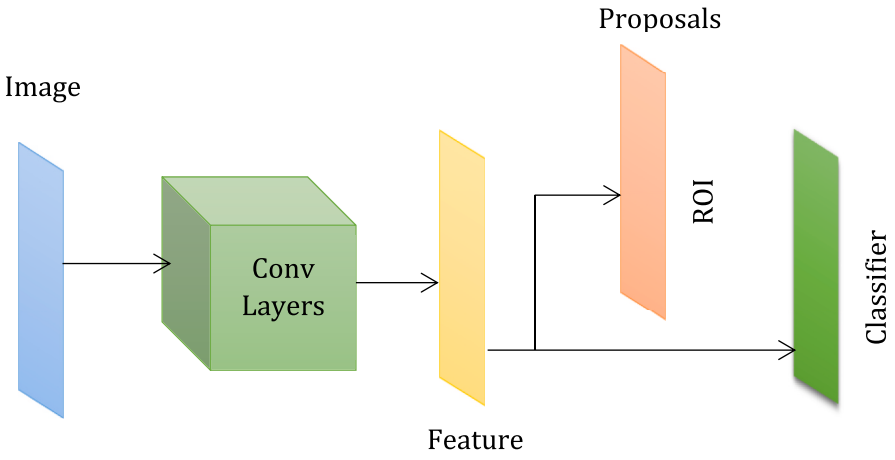

Fig. 2. Object detection using Faster RCNN functions as a single, unified network.

same search process is carried out, but it is carried out across ever smaller subregions. When there is only a little amount of bounding box in the region R (the leaf node), the recursive tree comes to an end. Then, the answers to the fragments are determine the probability P (Ri). Algorithm 2 provides a synopsis of this procedure.

unique regions of interest in this section. The spatial relationship between candidates in the pairwise potential model differs from the score functions defined in the unary potential of prior work, which are well-defined by the local object detector at a matching position. Fig. 5 depicts the join score function model.

The labels 1 and 0 are used to denote the background and the area of

Algorithm 2 Localization Modification Method

1. Describe the search region encompassing ri

2. Evaluate P (Ri).

3. Evaluate f (Ri)

4. Assign the new subordinates to Ri if P(Ri > 0).

5. Continue steps 1–4 through till amount of P(Ri) reaches 0 or even the amount of iterations T exceeds a specified threshold.

6. Change the coordinates to Ri.

The evaluation of the first bounding box score is done with the help of the ROI pooling & classifier. Then, we extract the region suggestions using a selective search technique, selecting the highest-scoring area proposals as the initial bounding box. This procedure is referred to as non-maximal suppression (NMS). Fig. 4 shows how NMS gets the new search areas by using bounding boxes that have been cut and overlapped.

The suggested algorithm's embedded design. Because of this architecture, the context refinement method may be used in both the region proposal generating stage and the final refinement stage of the present two-stage detection pipeline.

The region Ri is known as the leaf node in the recurrence tree. The samples Ri' are taken from Ri and are located within the consistent region of the ascendants of the node in the recurrence tree, as well as being privileged in the region Ri of the node. The solution of the bounding box, f (Ri), which has the formula is used to identify the root node as given in equation (1).

$$f\,(\boldsymbol{Ri}) \sim maximum\{\,\boldsymbol{Ri}|\,\boldsymbol{R}'\,\epsilon\,\boldsymbol{Ri}'\} \tag{1}$$

B *Joint score procedure*

The purpose of the joint score procedure is to jointly reason about interest, respectively. A ground truth label is provided for each region of interest in the training images. Based on how big and where the bounding box is, the edges of the regions are grouped together. The edge cluster index I j) is written as kij 1, …, k. 1, …, k.

The θ_i^U and θ_{ij}^P of unary and pairwise potentials, andthe joint score function $S(y, w)$ that ties all together labels of trainees in the same image, where is the trainable parameter, depend on, and y is the vector of all binary variables.

$$S(y,w) = \sum_{i\in S} y_i \theta_i^U(\omega) + \sum_{(i,j)\in\varepsilon} y_i y_j \theta_{ij,k_{ij}}^P(\omega) \tag{2}$$

Using a feed-forward neural network, the score function Eq. (2) is used to correlate the image with potentials. Using the additional feed-forward networks known as the unary and pairwise networks, the feature vector of an area of interest may be mapped. A area of interest is mapped by a bounding box in the unary potential, while the concatenated pairs of a region of interest are mapped by the pairwise potential in the pairwise potential. The feature extractor for the joint function is derived from the localization refinement approach from the preceding procedure, yielding 2048 features. Even though we used 16 regions of interest from the region proposal for one image, the performance of the validation set did not change.

C Smooth Support Vector Machine

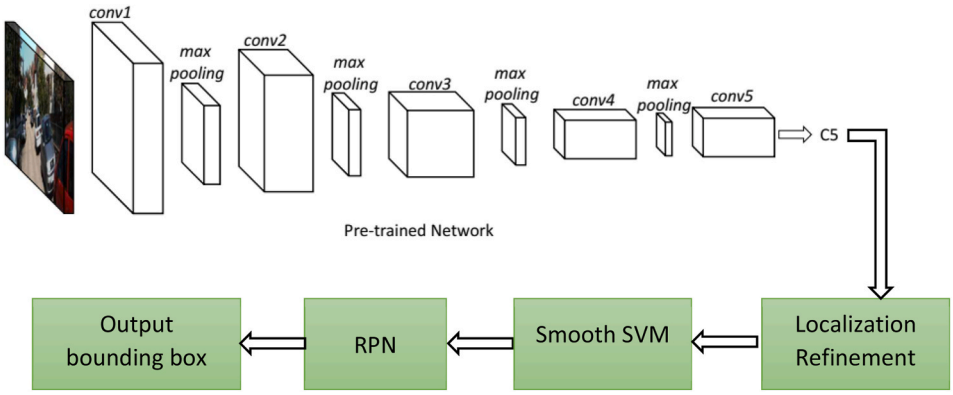

Fig. 3. Improvements to the localization process using a faster R–CNN.

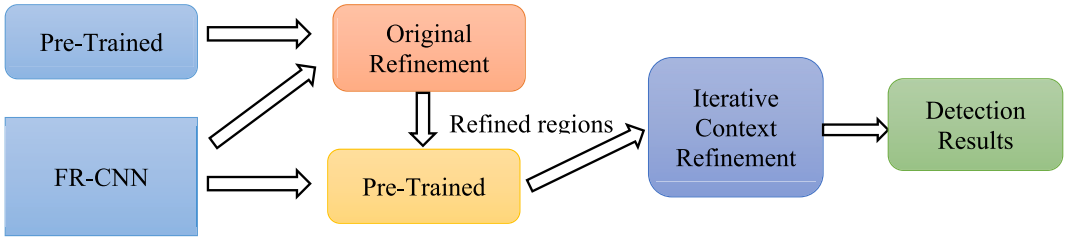

Fig. 4. Context iterative refinement model.

For the purpose of surrogate loss management and area of interest prediction, SSVM is used as a goal. This is because the measure relies on recall. Gradients are computed in accordance with the model parameters to optimise the surrogate loss. The model is trained using the parameters by the process of stochastic gradient descent, which involves minimising a structured surrogate loss as part of the process.

Algorithm 3 for SSVM:

Input: *O = [m, n]; m (input array), n (labels of class)*

Output: *Analyse the system performance and matrix*

function testing_SSVM (m,n, iterations)

initialize: *testing_rate = random.math();*

for testing_rate in itrations

error = 0;

for i in m

*if (n[i] * m[i]) < 1 then*

*update: 1 + testing rate * (n[i] * m[i]) * (1/iterations) * 2*

else

*update: 1 + testing rate * (1/iterations) * 2*

end if

end

end

The SSVM algorithm involves the subsequent phases.

1. Choose the regions of interest by applying the NMS to the Faster R–CNN scores produced first.
2. Carry out the feed forward pass so that the possibilities of the joint scoring function may be determined.
3. Implement the recommendation to calculate the structural loss and associated gradients.
4. Propagate the gradient backwards through the algorithm.

Gradients are computed with regard to the model parameters to optimise the surrogate loss. The back propagation method can be used to achieve this. By back propagating the gradients through the model parameters, the bounding box scores may be calculated precisely. With respect to the Unary and Pairwise Models, Eq. (3), and with respect to the feature extractor, Eq. (4), calculating the loss in the back propagation method.

$$\frac{dl}{d\omega^U} = \sum_{i \in S} \frac{dl}{d\theta_i^U} \frac{d\theta_i^U}{d\omega^U}, \frac{dl}{d\omega^P} = \sum_{(i,j)\epsilon k} \sum_{k=1}^{K} \frac{dl}{d\theta_{ij,k}^P} \frac{d\theta_{ij,k}^P}{d\omega^P} \tag{3}$$

$$\frac{dl}{df_i} = \frac{dl}{d\theta_i^U} \frac{d\theta_i^U}{df_i} + \sum_{j:(i,j)\epsilon\epsilon} \frac{dl}{d\theta_{ij,k_{ij}}^P} \frac{d\theta_{ij,k_{ij}}^P}{df_i} + \sum_{j:(i,j)\epsilon\epsilon} \frac{dl}{d\theta_{ji,k_{ji}}^P} \frac{d\theta_{ji,k_{ji}}^P}{df_i} \tag{4}$$

In this case, w represents the corresponding to the bounding planes, and establishes where those planes are located in relation to the origin as given in Eqs. (5) and (6).

$$x'w - \gamma = +1 \tag{5}$$

$$x'w - \gamma = -1 \tag{6}$$

When the two classes can be divided into strictly linear subgroups—that is, when the slack variable y is equal to 0—the first plane over bounds the points in class 1, and the second plane below limits the points in class −1. The plane serves as the dividing surface for the linear boundary as given in Eq. (7).

$$x'w - \gamma \tag{7}$$

The classification (cls) & regression (reg) losses are added together to form the loss function of the Regional Proposal Network. The entropy loss associated with determining whether an object is in the forefront or background is referred to as the categorization loss. The regression loss

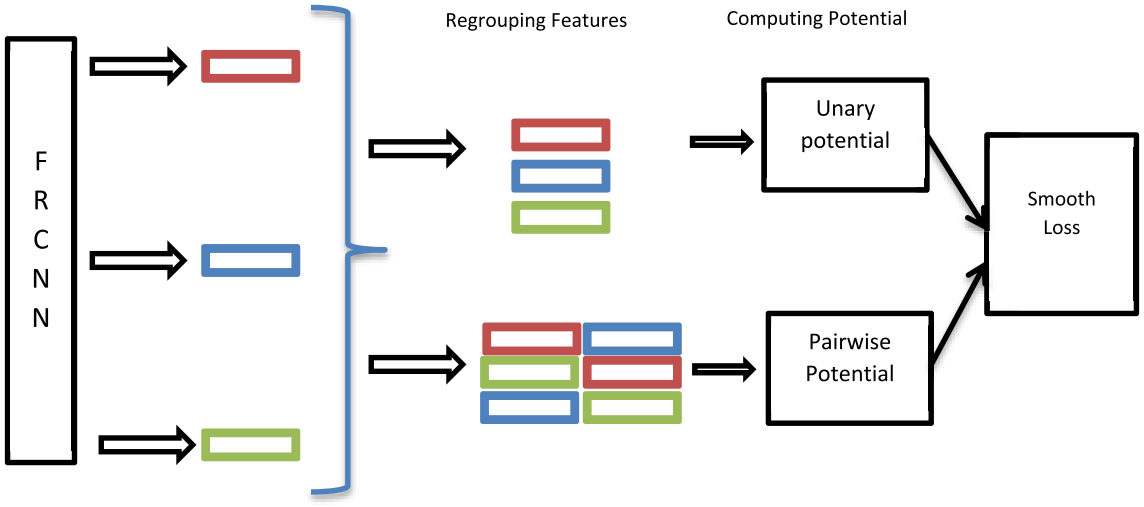

Fig. 5. A model of the technique for joining and scoring.

is calculated by subtracting the regression of the foreground box from the regression calculated using the ground truth box as given in Eq. (8).

$$L(\{Pi\},\{ti\}) = \frac{1}{N\_cls}\sum_i L\_cls(pi, pi^*) + \lambda \frac{1}{N\_reg}\sum_{ipi^*}(pi, pi^*) \quad (8)$$

Where,

i-anchor of index, p – Probability, t-vector of parameterized co-ordinates, N_cls and N_reg – Normalization, λ – Scale classifier

When presented with an image, the first methods of object recognition consisted of two stages:

The image is set into several smaller parts, and then those bits are sent into an image classifier, which determines whether or not the image includes an item. If the answer is true, then identify the fragment from the first image as the item that was spotted.

The SSVM method is one technique to accomplish the first phase, which involves sliding a rectangular window over the original image while using each of the grid boxes as a smaller component to create a new image, similar to the way the white box appears in the image that follows.

In order to do this, grids or windows consisting of several images were generated while taking into account a variety of aspect ratios (sizes), angles, and forms. These were then entered into a Conv net for the classification step. The boundary boxes were made by the window openings. The fact that this step had to be carried out a number of times made the procedure a significant computing challenge.

It can be utilised with a wide range of datasets since it is data-driven, learning-free, and adaptable to multiple FR-CNN architectures. Second, we use the joint score function on a large number of candidates in the picture graphs. The only two inputs that alter any score function parameter are picture data and refining a structured output loss function.

4. Experimental results

In this section, we assess iterative refinement on the PASCAL VOC 2012 dataset for object detection by calculating the mean average precision (mAP). As suggested, the dataset is divided into 5020 trained objects and 4960 validated objects. It also has more than 80,000 tagged traffic objects.

4.1. Data preprocessing

The data collected from various means is in a messy format, and there may be many null values, invalid data values, and unwanted data. Clean all this data and replace it with appropriate or similar data. Delete empty dates and missing data and replace them with some dates. The fixed substitute value is the most important preprocessing step of the data. All these cases need to be checked and replaced with substitute values so that the data is meaningful and useful for further processing. Data should be stored in an organized format.

The Faster R–CNN iterative refinement input is the entire image, and the output is a 248-dimensional vector. A recurrence tree is utilised to refine every vector. The IoU ratio for the maximum number of iterations with at least one ground truth value is 0.3. Using an SGD method with weight decay of 0.0005, momentum of 0.90, and learning rate of 0.0001, the parameters are modified by decreasing the sum of log losses. The weights of the unary and paired potentials were reset using 0.01 standard deviation Gaussians. The structured surrogate objective is optimised using SGD and weight decay values of 0.00005, momentum of 0.90, and a learning rate of 0.00001. To compute the detection performance, we can use the mean average precision metric that is derived from the precision-recall curve. Detections with an IoU greater than or equal to 0.30 are considered positive. The detectors are Fast R–CNN with ZF, but using various proposal methods for training and testing.

The iterative refinement of the Faster R–CNN model is compared to that of the R–CNN model and the SPP-net model in Table 1. The procedure of further refining has the potential to provide more accurate results for the planned territories. In terms of IR efficiency, we find that FR-CNN outperforms both R–CNN & SPP-net by 3.6 mAP a shown in Fig. 6. When bounding box regression is included, the IR in Faster R–CNN produces a mAP value that is 61.32 % greater than R–CNN and FR-CNN.

Iterative refinement is used to change the size and location of the bounding box to their suitable values. The efficacy grows as the number of true positives increases. Fig. 7 displays the detection results achieved from the CIR Faster R–CNN.

Detected Objects:

The model has identified multiple objects and their corresponding bounding box coordinates.

Object: Car Bounding Box: (x1, y1) = (120, 150), (x2, y2) = (300, 250).

Table 1
Detection results on PASCAL VOC 2012 dataset.

Method	mAP
R–CNN	54.3
Fast R–CNN	55.3
Faster R–CNN	59.1
Faster R–CNN + CIR	**62.7**

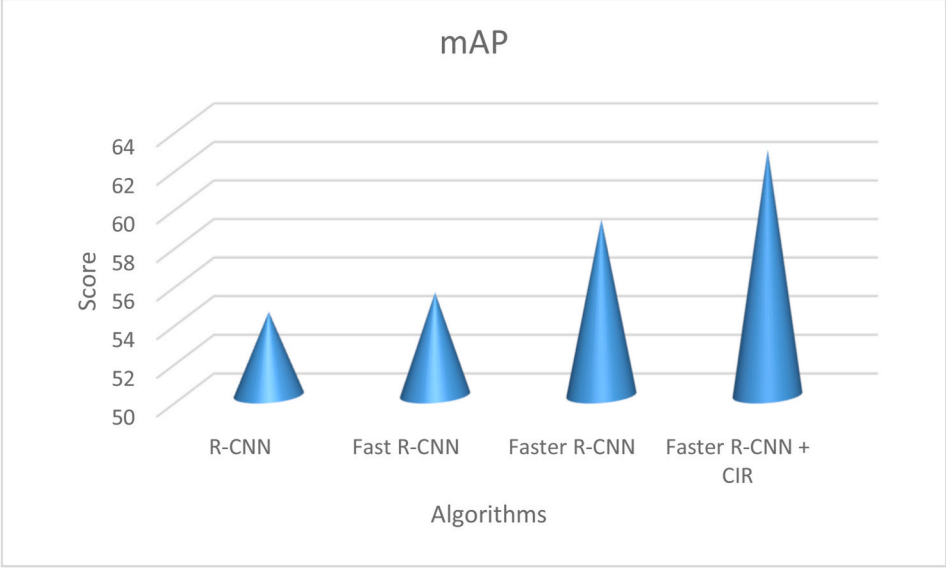

Fig. 6. Algorithm comparison for mAP.

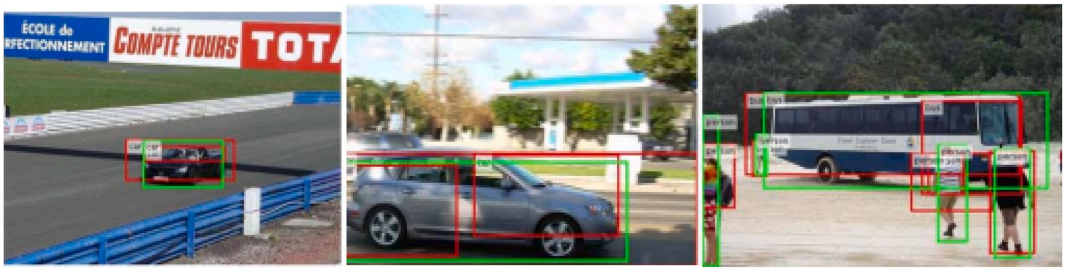

Fig. 7. The detection results obtained using the method suggested (green bounding boxes). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Object: Person Bounding Box: (x1, y1) = (400, 200), (x2, y2) = (500, 350).

Object: Dog Bounding Box: (x1, y1) = (550, 300), (x2, y2) = (650, 400).

The visualization will overlay the bounding boxes on the image, outlining the regions where the objects are detected. This allows for a clear representation of the objects and their spatial locations in the image.

4.2. Objectness score

The CLS layer produces an output consisting of a vector with two elements for each proposed area. In the event where the first element has a value of 1 and the second element has a value of 0, the area proposal has been designated as backdrop. If the first element is 0, then the area does not represent an object. If the second element is 1, then the region does represent an item.

The area of intersection between the anchor and ground-truth boxes is used to compute the IoU, which is given as a ratio to the total surface area of the two boxes together. The IoU may be anything between 0.0 and 1.0. The IoU equals 0.0 when there is no intersection to evaluate. The IoU will continue to rise until it reaches 1.0 (which will occur when the two boxes are 100 % similar to one another), regardless of how near the two boxes get to one another.

The following four circumstances will utilise the IoU to decide whether or not an anchor will be given a positive or negative objectness score, respectively.

1. A positive objectness label is assigned to an anchor with an IoU overlap percentage of more than 0.7 with any ground-truth box.
2. Whenever the IoU overlap for all ground-truth box is less than zero, a non-positive anchor is assigned a negative objectness score.
3. Anchors that may be classified as neither positive nor negative do not assist in any way to the overall goal of the exercise.

Assume that there are three region proposals connected to three anchors, and that the IoU scores of these suggestions with respect to the three ground-truth boxes are stated below. Given that there is an anchor with an IoU value of 0.9, which is more than 0.7, the objectness score associated with that ground-truth box is positive, but the objectness score associated with all of the other boxes is negative. The following equation provides a synopsis of the four conditions as given in Eq. (9).

$$Objectness_{Score}(IoU) = \begin{cases} Positive \rightarrow IoU > 0.7 \\ Positive \rightarrow 0.7 \geq IoU > 0.5 \\ Negative \rightarrow 0.3 > IoU \\ Not\ Negative\ /\ Positive \rightarrow 0.5 \geq IoU \geq 0.3 \end{cases} \quad (9)$$

Note that agreeing to meet the first condition $IoU > 0.7$ is usually all that is required to classify an anchor as positive (i.e., it contains an object), but the authors chose to also mention meeting the second condition ($0.7 \geq IoU > 0.5$) in case there were any of the extremely rare instances in which there was no region with such an IoU of 0.7.

The steady improvement of the average mean accuracy values during the training epochs is seen in Figs. 8 and 9. Both designs effectively minimized the values of their individual loss functions while also achieving high mean average precision (mAP). The smoothness in the

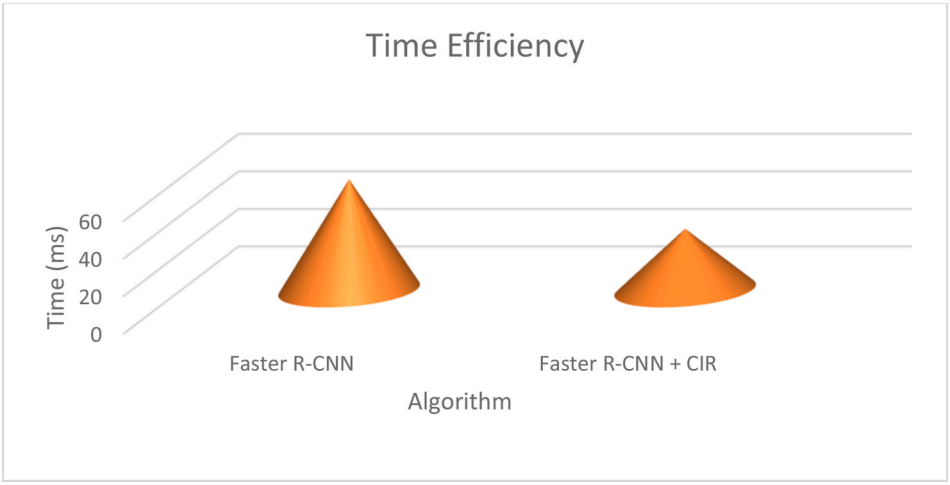

Fig. 8. During the training phase, the amount of time required to carry out each architectural on a single image.

Total time: 33:08

epoch	train_loss	valid_loss	pascal_voc_metric	BBloss	focal_loss
0	1.472294	1.599764	0.158803	0.385957	1.213806
1	1.644351	1.791672	0.138816	0.410458	1.381213
2	1.772430	1.777559	0.132887	0.408994	1.368565
3	1.681056	1.654816	0.150120	0.383538	1.271278
4	1.607463	1.615077	0.164021	0.373200	1.241877
5	1.502257	1.708495	0.136862	0.361128	1.347366
6	1.381117	1.428304	0.204288	0.347032	1.081272
7	1.263347	1.340594	0.228309	0.335232	1.005362
8	1.196914	1.352324	0.209727	0.330933	1.021391
9	1.162093	1.333483	0.213452	0.329912	1.003570

Fig. 9. Total time Taken for object detection.

curve demonstrates that the FRCNN network demonstrated greater and better accuracy stability than its competitors, as can be observed. The SSD came out on top with mean & standard deviation values of 17 ± 217 ±2 ms, while the FRCNN interpreted its high processing complexity into a longer execution time with mean & standard deviation values of 30 ± 230±2 ms, respectively.

The time taken to process the input image and perform forward propagation through the object detection model. This includes passing the image through the convolutional layers, region proposal network, and classification and bounding box regression heads. The time taken to preprocess the input image before feeding it into the object detection model. Preprocessing tasks may include resizing, normalizing, and converting the image to the required format.

The performance of object detection is further explored in Fig. 10, which displays further findings. First, it shows overall mean average precision, that represents the average value of the average precisions for each class.

According to the definition, average precision is the number that is the average of 12 points on the accuracy curve for each potential threshold. This implies that it contains all of the detection probabilities for the same class. Faster R–CNN with context iterative refinement can be complex and time-consuming, often requiring the use of deep learning frameworks such as TensorFlow or PyTorch. Additionally, the availability of pre-trained models and open-source implementations can significantly aid in the implementation process.

This experimental analysis focuses on evaluating the performance and effectiveness of this approach through various experiments and

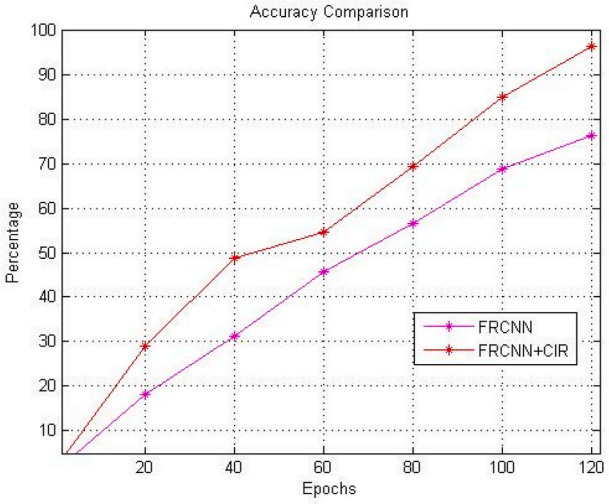

Fig. 10. Accuracy comparison.

benchmarks. It clearly shows that our proposed works performs better than exiting methods.

5. Conclusion

This paper presents a unique technique for enhancing report a gain and localization accuracy of 3.6 mAP. The newly created Iterative Refinement approach for correcting imprecise bounding boxes has been incorporated into the current object proposal methods, widely used in FRCNN-based object recognition. By applying the scoring functions, the objective of the combined score is to bring together several distinct areas of interest. The FRCNN is a deep CNN that is used for object recognition. It gives the impression to the user that the network is unified and single across its whole. The network can provide precise and timely predictions on the locations of various items. In order to have a complete comprehension of Faster R–CNN, we must first get a fundamental familiarity with its progenitor networks, namely RCNN and FRCNN. This paper also contains experimental results on various methods for the detection and identification of objects. It compares each method based on its efficiency. The experiments' findings reveal an increase in Mean Average Precise in connection to the proposed area feature vector. The inclusion of motion information enabling long-term tracking is another possible advancement that may take place in the future. In future, this can be

done by using more efficient neural network architectures, or by using parallel computing techniques.

Declaration of competing interest

The authors declare that they have no known competing for financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] T. Sharma, B. Debaque, N. Duclos, A. Chehri, B. Kinder, P. Fortier, Deep learning-based object detection and scene perception under bad weather conditions, Electronics 11 (4) (2022) 563.

[2] A. Wang, Y. Sun, A. Kortylewski, A.L. Yuille, Robust object detection under occlusion with context-aware compositionalnets, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12645–12654.

[3] Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, Y. Fu, Rethinking classification and localization for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10186–10195.

[4] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.

[5] T. Qu, Q. Zhang, S. Sun, Vehicle detection from high-resolution aerial images using spatial pyramid pooling-based deep convolutional neural networks, Multimed. Tool. Appl. 76 (2017) 21651–21663.

[6] C. Chen, K. Debattista, J. Han, Semi-supervised Object Detection via Virtual Category Learning, 2022 *arXiv preprint arXiv:2207.03433*.

[7] T. Diwan, G. Anirudh, J.V. Tembhurne, Object detection using YOLO: challenges, architectural successors, datasets and applications, Multimed. Tool. Appl. 82 (6) (2023) 9243–9275.

[8] D. Rukhovich, A. Vorontsova, A. Konushin, Imvoxelnet: image to voxels projection for monocular and multi-view general-purpose 3d object detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 2397–2406.

[9] B. Chen, W. Chen, S. Yang, Y. Xuan, J. Song, D. Xie, Y. Zhuang, Label matching semi-supervised object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14381–14390.

[10] S. Kothawade, S. Ghosh, S. Shekhar, Y. Xiang, R. Iyer, Talisman: targeted active learning for object detection with rare classes and slices using submodular mutual information, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII, Springer Nature Switzerland, Cham, 2022, October, pp. 1–16.

[11] X. Li, Y. Zhang, D. Kong, Eˆ 2-PV-RCNN: improving 3D object detection via enhancing keypoint features, Multimed. Tool. Appl. 81 (25) (2022) 35843–35874.

[12] C. Xia, Y. Sun, X. Gao, B. Ge, S. Duan, DMINet: dense multi-scale inference network for salient object detection, Vis. Comput. 38 (9–10) (2022) 3059–3072.

[13] H. Wang, S. Dong, S. Shi, A. Li, J. Li, Z. Li, L. Wang, Cagroup3d: class-aware grouping for 3d object detection on point clouds, Adv. Neural Inf. Process. Syst. 35 (2022) 29975–29988.

[14] Y. Zhang, J. Chen, D. Huang, Cat-det: contrastively augmented transformer for multi-modal 3d object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 908–917.

[15] A.M. Roy, R. Bose, J. Bhaduri, A fast accurate fine-grain object detection model based on YOLOv4 deep neural network, Neural Comput. Appl. (2022) 1–27.

[16] T. Fel, I.F. Rodriguez Rodriguez, D. Linsley, T. Serre, Harmonizing the object recognition strategies of deep neural networks with humans, Adv. Neural Inf. Process. Syst. 35 (2022) 9432–9446.

[17] A.M. Obeso, J. Benois-Pineau, M.S.G. Vázquez, A.Á.R. Acosta, Visual vs internal attention mechanisms in deep neural networks for image classification and object detection, Pattern Recogn. 123 (2022) 108411.

[18] S. Zou, C. Li, H. Sun, P. Xu, J. Zhang, P. Ma, M. Grzegorzek, TOD-CNN: an effective convolutional neural network for tiny object detection in sperm videos, Comput. Biol. Med. 146 (2022) 105543.

[19] S.Y. Alaba, J.E. Ball, Wcnn3d: wavelet convolutional neural network-based 3d object detection for autonomous driving, Sensors 22 (18) (2022) 7010.

[20] V.R.S. Mani, A. Saravanaselvan, N. Arumugam, Performance comparison of CNN, QNN and BNN deep neural networks for real-time object detection using ZYNQ FPGA node, Microelectron. J. 119 (2022) 105319.

[21] A.N. Amudhan, A.P. Sudheer, Lightweight and computationally faster Hypermetropic Convolutional Neural Network for small size object detection, Image Vis Comput. 119 (2022) 104396.

[22] M.C. Chiu, H.Y. Tsai, J.E. Chiu, A novel directional object detection method for piled objects using a hybrid region-based convolutional neural network, Adv. Eng. Inf. 51 (2022) 101448.

[23] Z. Dong, M. Wang, Y. Wang, Y. Liu, Y. Feng, W. Xu, Multi-oriented object detection in high-resolution remote sensing imagery based on convolutional neural networks with adaptive object orientation features, Rem. Sens. 14 (4) (2022) 950.

[24] S.S.A. Zaidi, M.S. Ansari, A. Aslam, N. Kanwal, M. Asghar, B. Lee, A survey of modern deep learning based object detection models, Digit. Signal Process. (2022) 103514.

[25] N. Wang, Y. Wang, M.J. Er, Review on deep learning techniques for marine object recognition: architectures and algorithms, Control Eng. Pract. 118 (2022) 104458.

[26] J. Ren, Y. Wang, Overview of object detection algorithms using convolutional neural networks, J. Comput. Commun. 10 (1) (2022) 115–132.

[27] L.S. Huber, R. Geirhos, F.A. Wichmann, The Developmental Trajectory of Object Recognition Robustness: Children Are like Small Adults but unlike Big Deep Neural Networks, 2022 *arXiv preprint arXiv:2205.10144*.

[28] G. Yang, B. Wang, S. Qiao, L. Qu, N. Han, G. Yuan, Y. Peng, Distilled and filtered deep neural networks for real-time object detection in edge computing, Neurocomputing 505 (2022) 225–237.

[29] V.V. Aroulanandam, T.P. Latchoumi, B. Bhavya, S.S. Sultana, Object detection in convolution neural networks using iterative refinements, Architecture 15 (2019) 17.