Computers, Materials & Continua
DOI:10.32604/cmc.2021.014406
Article

Tech Science Press

Spatial-Resolution Independent Object Detection Framework for Aerial Imagery

Sidharth Samanta[1], Mrutyunjaya Panda[1], Somula Ramasubbareddy[2], S. Sankar[3] and Daniel Burgos[4,*]

[1]Deptartment of CSA, Utkal University, Bhubaneswar, 751004, India
[2]Department of Information Technology, VNRVJIET, Hyderabad, 500090, India
[3]Department of CSE, Sona College of Technology, Salem, 636005, India
[4]Research Institute for Innovation & Technology in Education (UNIR iTED),
Universidad Internacional de La Rioja (UNIR), Logroño, 26006, Spain
[*]Corresponding Author: Daniel Burgos. Email: daniel.burgos@unir.net

Received: 18 September 2020; Accepted: 14 February 2021

Abstract: Earth surveillance through aerial images allows more accurate identification and characterization of objects present on the surface from space and airborne platforms. The progression of deep learning and computer vision methods and the availability of heterogeneous multispectral remote sensing data make the field more fertile for research. With the evolution of optical sensors, aerial images are becoming more precise and larger, which leads to a new kind of problem for object detection algorithms. This paper proposes the "Sliding Region-based Convolutional Neural Network (SRCNN)," which is an extension of the Faster Region-based Convolutional Neural Network (RCNN) object detection framework to make it independent of the image's spatial resolution and size. The sliding box strategy is used in the proposed model to segment the image while detecting. The proposed framework outperforms the state-of-the-art Faster RCNN model while processing images with significantly different spatial resolution values. The SRCNN is also capable of detecting objects in images of any size.

Keywords: Computer vision; deep learning; multispectral images; remote sensing; object detection; convolutional neural network; faster RCNN; sliding box strategy

1 Introduction

Surveillance of a large geographical area through aerial imagery is undoubtedly a faster and less time-consuming process than conventional methods that use a horizontal perspective. Although there are some cases where aerial imagery cannot be used for surveillance, like person or facial detection and pedestrian or vehicle license plate detection, it can be used for detection of the number and types of vehicles in a city or any geographical area. To perform this task using a horizontal perspective, it is very expensive in terms of planning, procurement and execution, but computationally it is quite simple to analyse through an aerial perspective. The field of computer vision has resolved numerous problems of surveillance, irrespective of their type and complexity.

Surveying the Earth from an aerial view by using deep learning has not only reduced the time and cost but has also become more accurate and robust with the availability of training data and computation power. There are many application areas, like the study of vegetation distribution in an area and changes in shape and size of agricultural land, towns, or slums, where the machine outsmarts humans concerning time as well as efficiency.

1.1 Object Detection

Object detection is a computer vision technique widely used in surveillance. It is generally used to determine the number, type and position of a particular object in an image. There are many state-of-the-art object detection frameworks such as the Region-based Fully Convolutional Network (RFCN) [1], Single-Shot Detector (SSD) [2], You Only Look Once (YOLO) [3] and RCNN [4] and its multiple variants, such as Mask RCNN [5], Fast RCNN [6], Faster RCNN [7], YOLO version 2 [8] and YOLO version 3 [9]. Each of these frameworks uses different methods and principles to detect objects, but all are based on deep neural networks. This study uses Faster RCNN rather than SSD and YOLO because of its accuracy [10], although it is slower and more resource-heavy than SSD and YOLO. When detecting objects from an extremely large aerial image, time and computational resources can be traded for accuracy.

The change in the size of the objects in the image makes the detection process more complex for the algorithm. When a trained model processes an input image with higher or lower spatial resolution than the training image dataset, the Region Proposal Network (RPN) of Faster RCNN fails to provide a Region of Interest (RoI). This is because the RPN uses similar sized anchor boxes as evaluated during the training process. For example, an object detection model trained on a dataset with a spatial resolution of 7.5 cm cannot perform well with an image with a spatial resolution of 30 cm. The same thing happens for the size of the image. A model trained on a dataset of images with the dimensions 250 px × 250 px cannot perform accordingly with larger images with the dimensions 1000 px × 1000 px or smaller images with the dimensions 100 px × 100 px.

1.2 Problem Statement

Innovations in optical sensors, storage devices and sensor carriers like satellites, airplanes and drones have revolutionized the remote sensing and Geographic Information System (GIS) industries. These sensors are producing a huge number of multispectral images with different characteristics, such as spatial resolution. The spatial resolution of an aerial image can be defined as the actual size of an individual pixel on the surface, as demonstrated in Fig. 1. Images with lower spatial resolution values seem to be clearer and larger than those with relatively higher spatial resolution values.

An object detection model trained with an arial image dataset will perform accordingly with test images having the same spatial resolution, but its accuracy drops drastically when tested with images having a different spatial resolution. Almost all existing state-of-the-art frameworks fail to detect objects in this scenario. Though image cropping can be used where the spatial resolution of the training image is less than that of the testing image, the reverse (i.e., the spatial resolution of the training image is higher than that of the testing image) cannot be done with this technique.

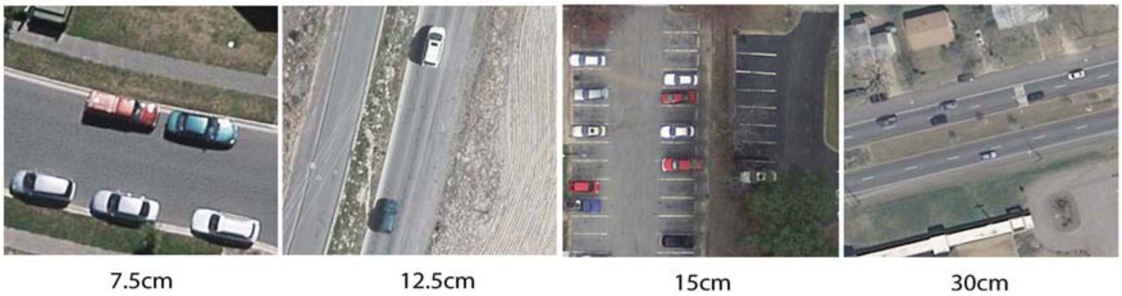

Figure 1: 250 × 250 px of four different images with different spatial resolution values

1.3 Research Contributions

This paper proposes an extension to the state-of-the-art Faster RCNN. It is based on the sliding windows strategy which uses a mathematically-derived optimal window size for precise detection. The primary use cases of the proposed model can be noted as follows:

(i) To detect objects from images of any spatial resolution value and size, such as detection of vehicles in a city [11] and tree crowns in a forest [12].
(ii) For object detection in images captured from drones [13] or aircraft [14], where the elevation is not fixed, as elevation is directly proportional to the spatial resolution value, where the sensor remains constant.
(iii) For the detection of small and very small objects, such as headcounts in protests or social gatherings [15,16].
(iv) It can also be used for microscopic object detection such as cells [17], molecules [18], pathogens [19], red blood cells [20] and blob objects [21].

The rest of the paper is organized as follows. Section 2 provides an overview of some critical works on object detection in remote sensing and aerial imagery and methods to deal with size and resolution. Sections 3 and 4 provide the proposed model and its results, respectively. Finally, Section 5 contains the conclusion.

2 Related Works

Many pieces of literature have reviewed the application of deep-learning-based computer vision techniques in aerial imagery. The authors [22] surveyed about 270 publications related to object detection. This includes the detection of objects by (i) matching the template, (ii) matching the knowledge, (iii) image analysis and (iv) machine learning. They also raised a concern about the availability of labelled data for supervised learning. Han et al. [23] proposed a framework in, where a weakly labelled dataset can be used to extract high-level features. The problem of object orientation in remote sensing imagery is addressed in [24–26].

Diao et al. [27] proposed a deep belief network in, whereas [28] used a convolutional neural network for object detection. In [29], a basic RCNN model is used and in [30] a single-stage densely connected feature pyramid network is used for object detection specifically for very-high-resolution remote sensing imagery. The studies in [31,32] used the SSP and the state-of-the-art YOLO 9000, respectively. Huang et al. used a densely connected YOLO based on the SSP in [33]. The proposed model aims to provide a framework that can process any aerial image with any value of spatial resolution. Although very few studies addressed this problem, the semantics of [34–36] and the method used in [37] are close to the working principle of the proposed model.

3 Proposed Method

This study proposes an extension that is based on the sliding window strategy; therefore, it is called the Sliding Region-based Convolutional Neural Network. In the proposed model, the slider box shown in Fig. 2i(a) will roam all over the input image just like a convolution operation with a determined stride value. The stride value is derived from the spatial resolution of the input image. At each instance of the box position, the model will perform the object detection process according to the stock Faster RCNN on the fragment of the image that falls under the footprint of the slider box as demonstrated in Fig. 2i(b). Fig. 2 shows the architecture of the proposed SRCNN. The proposed SRCNN is divided into three phases.

- Phase 1: Image Analysis
- Phase 2: Image Pre-Processing
- Phase 3: Object Detection

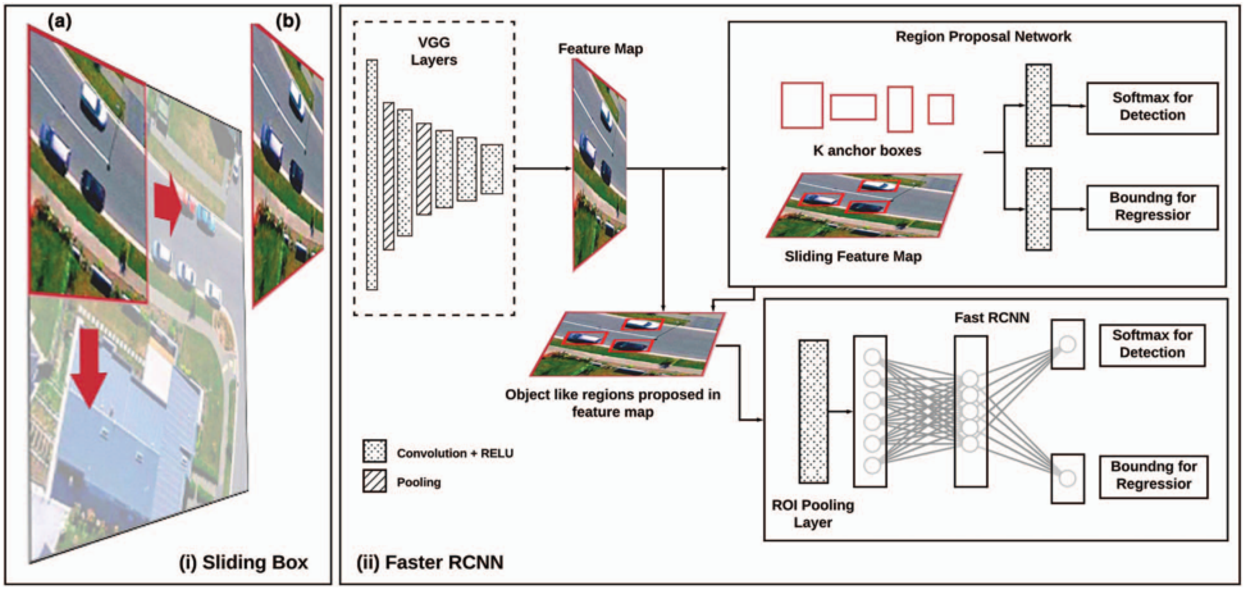

Figure 2: Architecture of proposed sliding RCNN

3.1 Phase 1: Image Analysis

Phase 1 of the proposed model includes data acquisition, data analysis and a box dimension proposal. This phase plays a vital role in normalizing the spatial resolution factor. As illustrated in Fig. 1 in Section 1.2, the size changes according to the spatial resolution value. So, the image has to be scaled in such a way that the size of the object in the training and testing images feels similar in terms of spatial view. In Fig. 3, the visual object size feels very similar in (a) and (b) as the image in (b) is down-scaled almost three times. For the proposed model, the original dimension of the scaled image can be the size of the slider box. The box length m can be derived from the average length s of the input image with dimensions a × b and the spatial resolution of both training image r and testing image R, as follows:

$$s = \frac{(a+b)}{2}$$

$$m = \mathrm{s} \times \left(\frac{\mathrm{r}}{\mathrm{R}}\right) \tag{1}$$

Thus, the slider box width is the product of the training image width and the ratio between the spatial resolution of the training image and the input image. This value is also helpful when cropping a large image to process individually.

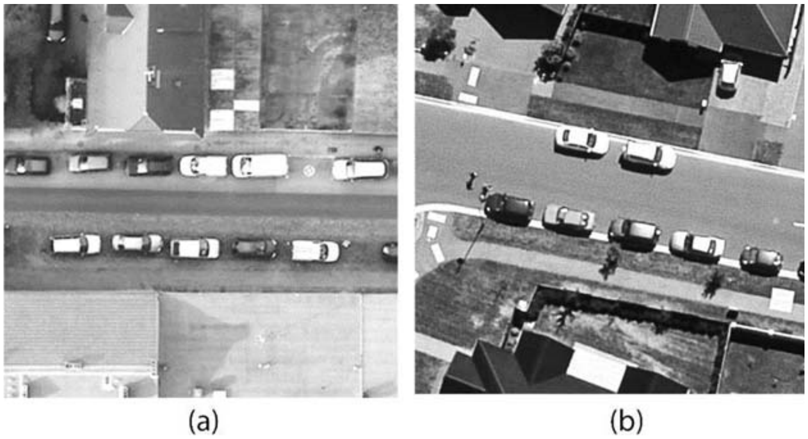

Figure 3: Scaling of image. (a) Spatial resolution: 25 cm/px image size: 250 × 250 px, (b) spatial resolution: 7.5 cm/px image size: 848 × 848 px

3.2 Phase 2: Image Pre-Processing

Phase 2 of the proposed model is image pre-processing, which includes image size analysis and padding. The size of the slider box, evaluated in Section 3.1, depends upon the spatial resolutions of the training and testing image and the dimensions of the training image. But the slider box has to traverse every pixel present in the testing image, so it must be compatible with the image size. In Fig. 4ii, the original image is too short to accommodate the last set of slider boxes. As the image area covered by these last boxes will be exempted from the object detection process, it cannot be ignored. This problem can be solved by either image resizing or image padding, in such a way that the end of the last slider box will converge with the end of the image, as demonstrated in Figs. 4i and 4iii.

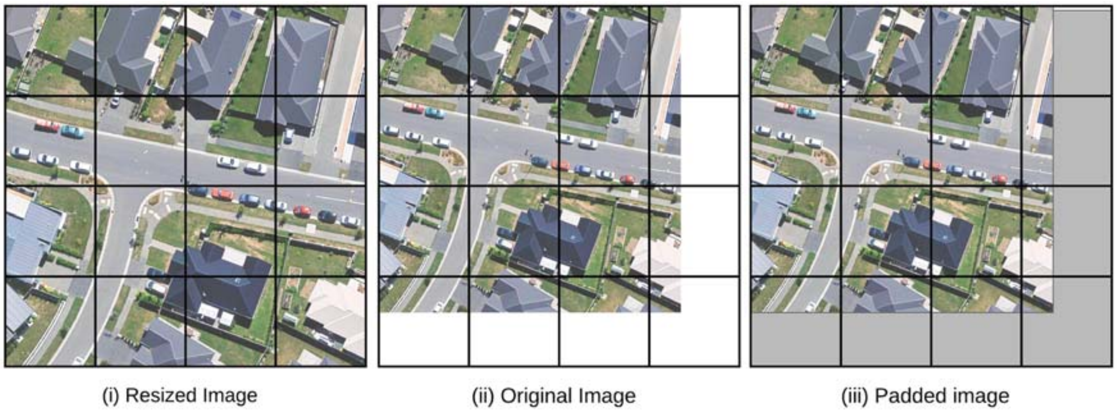

Figure 4: Overlapping problem and difference between resizing and padding

It is observed in Fig. 4 that the object size in the padded image is the same as the original image, but the object size in the resized image is bigger than the original, and this is similar to Fig. 1. This means that resizing the image results in a significant change in spatial resolution. Thus, the proposed model has used the padding method over resizing. The given image needs to be padded with 0s in such a way that the sliding boxes can cover the entire image area. To determine the padding amount, two cases have to be considered for the slider box of length m, which takes p number of steps to cover the image having length n with O percentage of overlapping. The best and worst cases are demonstrated in Fig. 5.

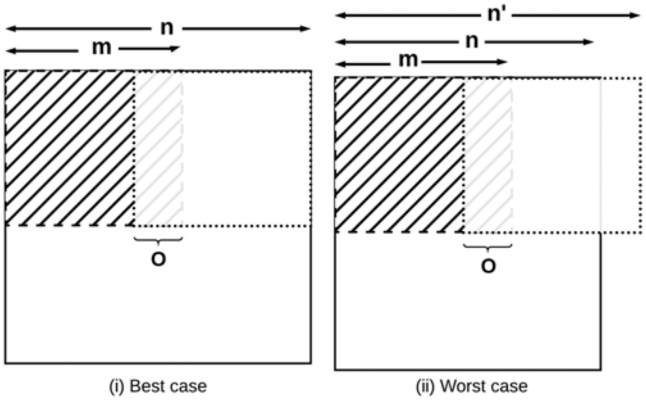

Figure 5: Box sliding demonstration

a) Best Case:

The last box converges perfectly with the image as shown in Fig. 5 (case 1). The size of the image is calculated as follows:

$$n = p \times m - [(p - 1) \times (m \times O)]$$

$$p = \frac{n - m \times O}{m \times (1 - O)} \tag{2}$$

b) Worst Case:

The last box does not converge with the image as shown in Fig. 5 (case 2). The box will take p' number of steps to cover the image.

$$p = \left\lceil \frac{n - m \times O}{m \times (1 - O)} \right\rceil \tag{3}$$

With p′ number of instances, an image of length n′ is needed to converge perfectly like the best case. The same formula is applied for vertical sliding as well.

$$n' = p'm - \left[\left(p' - 1\right) \times (mO)\right] \tag{4}$$

$$\text{Pading Amount} = (n' - n) \tag{5}$$

3.3 Phase 3: Object Detection

Phase 3 of the proposed model is detection. The fraction of the image that falls under the footprint of the slider box is selected and the image matrix is processed by the Faster RCNN to detect the objects. Here, a trained Faster RCNN model is used to detect objects in the input image. Rather than taking the whole image at once, it takes the box image, i.e., the portion of the input image covered by the sliding box. By using Eq. (3), the row instance P_r and column instance P_c can be evaluated for an input image of dimension a × b. The product of P_c and P_r is the total number of iterations I.

$$P_r = \left\lceil \frac{a-(m\times O)}{m\times(1-O)} \right\rceil$$

$$P_c = \left\lceil \frac{b-(m\times O)}{m\times(1-O)} \right\rceil$$

$$I = P_r \times P_c \tag{6}$$

4 Results and Discussion

A computer with an Intel i5 8th generation processor, 8 GB RAM and a dedicated 4 GB NVIDIA GTX 1050ti graphics card is used to train a Faster RCNN model using the TensorFlow open-source library. Pre-trained weights named "faster_rcnn_inception_v2_coco_2018" are used to initialize the parameter for transfer learning. The model was trained for nineteen hours on the benchmark VEDAI dataset [38]. The experimental codes used in this paper for evaluation and weights are available at https://github.com/sidharthsamanta/srcnn.

Four types of images with spatial resolution (sample image with ground truth demonstrated in Fig. 6) 7.5, 12.5, 15.5 and 30.5 cm were used for testing (Fig. 7). Each type contained three images of 256 px × 256 px, the same as the training image dataset. All images were processed under Faster RCNN and the proposed SRCNN to determine the accuracy and the precision of the proposed framework.

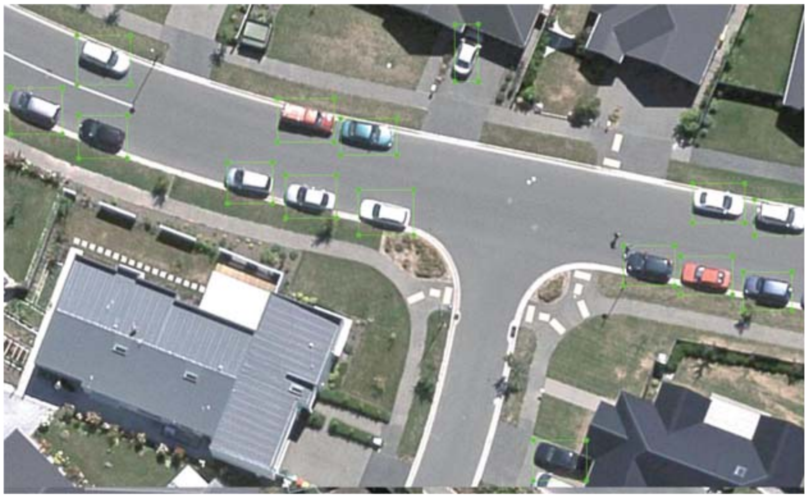

Figure 6: Portion of test Christchurch.jpg with ground truth

i. **Image Analysis:** Details of the testing images are given below in Tab. 1. The Box Size column in the table is the length of the slider box, which is calculated by using the formula derived in Eq. (1).

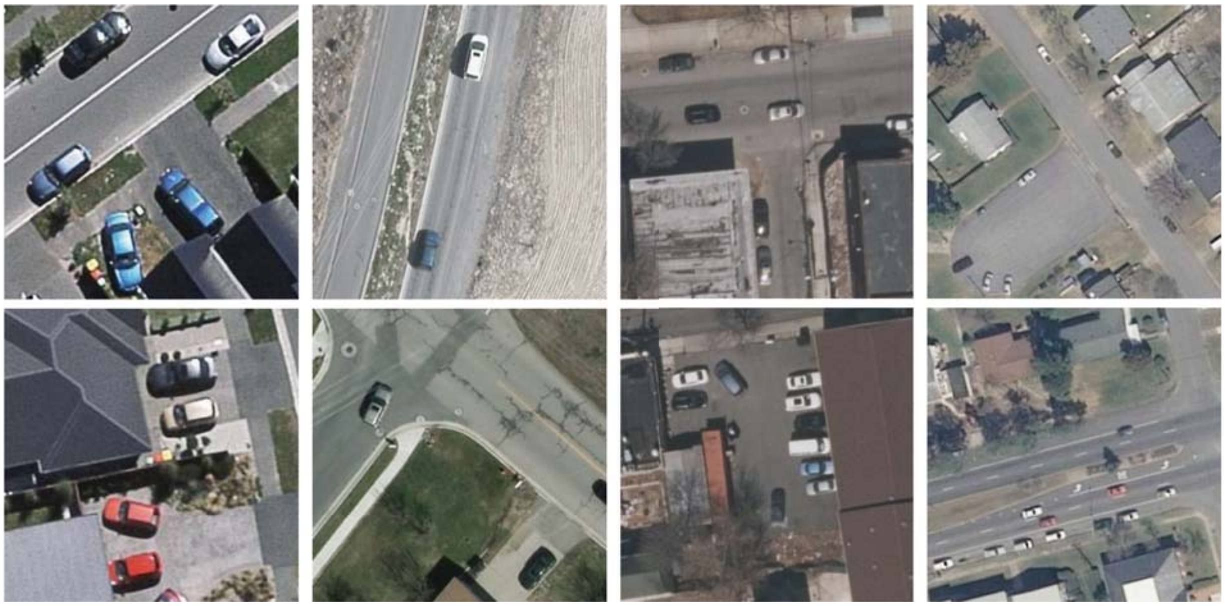

Figure 7: Overview of test images with spatial resolution 7.5, 12.5, 15 and 30.5 cm, respectively

Table 1: Train and test image description

Image name	Spatial resolution	Box size (px × px)	Padding value (px × px)
Christchurch	7.5 cm	426 × 426	230 × 230
VEDAI	12.5	256 × 256	0 × 0
New York	15 cm	213 × 213	159 × 159
Arlington	30 cm	104 × 104	45 × 45

ii. **Image Pre-processing:** The padding amount p is calculated for each image with 5% overlapping by using the mathematical formula from Eq. (6). The Padding Value column of Tab. 1 contains all the padding values for each test image. The first number represents the number of 0s to be added on the right side of the image and the second number represents the number of 0s to be appended at the bottom of the image. 0s can be padded on any side of the image, as there will be no effect on performance.

iii. **Object Detection:** Now the detector is deployed on top of the sliding window to process the image fragment that falls under its footprint. The process continues until the box reaches the vertical and horizontal end. Fig. 8 illustrates the sliding detection process.

iv. **Evaluation:** The outcomes of the proposed model with four sets of input images mentioned in Tab. 1 are compared with the Faster RCNN model in Tab. 2. The confusion matrix is used for calculating the accuracy (Eq. 7) and precision (Eq. 8).

(a) **True Positives (TP):** Objects that are present in the ground truth and correctly detected in the output.
(b) **True Negatives (TN):** Objects that are not present in the ground truth and not detected in the output. For object detection and localization, the TN is always considered 0.

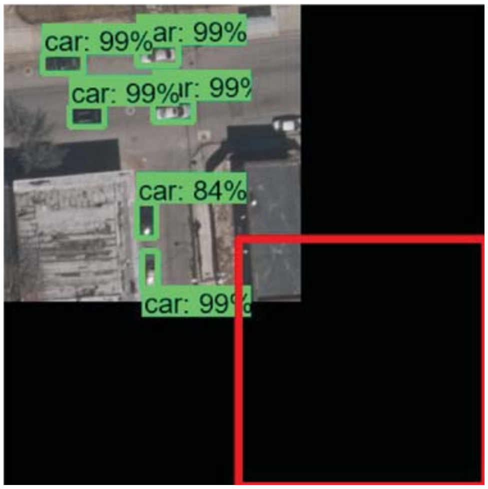

Figure 8: Sliding detection process on a sample image with 15 cm resolution

Table 2: Accuracy and prenecision of faster RCNN and proposed SRCNN

Spatial resolution	Accuracy		Precision	
	FRCNN	SRCNN	FRCNN	SRCNN
7.5 cm	0.5	0.85	1	1
12.5 cm	1	1	1	1
15 cm	0.7	0.87	1	1
30 cm	0	0.7	0	0.86

(c) **False Positives (FP):** Objects that are not present in the ground truth, but detected in the output.

(d) **False Negatives (FN):** Objects that are not present in the ground truth, but detected in the output.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

v. **Discussion:** As the spatial resolution was the same as the training image data, i.e., 12.5 cm, both models performed identically, as both are the same. But when the spatial resolution increased or decreased, the performance of the stock Faster RCNN started to deteriorate. There was a significant change in accuracy as well as in precision when the Faster RCNN dealt with the images having spatial resolution of 7.5 cm and 15 cm. At resolution 30 cm, it performed worse with 0 accuracies and 0 precision, whereas the proposed SRCNN shows the better results for every spatial resolution.

5 Conclusion

Detection of an object is a complex task due to ambiguity in object position, orientation and light source. A small modification of the sensor might change the scale of the objects present over the image. This scaling can be normalized by the proposed method, as it segments the image before detection. The proposed SRCNN outperformed the stock Faster RCNN on image samples with completely different spatial resolution values. It is additionally ascertained that the model can work with images of much smaller or far larger dimensions.

The size problem can also be resolved by using an internal slider box during the convolution operation. However, when an image with very large dimensions undergoes a convolution operation directly, it creates a large range of hyperparameters. Storing and processing these hyperparameters could cause a high configuration personal computer to run out of memory. There is a possibility to implement the extended part of SRCNN in a different state-of-the-art framework, such as YOLO or SSD.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Dai, Y. Li, K. He and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, Barcelona, Spain, pp. 379–387, 2016.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed *et al.,* "SSD: Single Shot multibox detector," in *European Conf. on Computer Vision*, Cham, Springer, pp. 21–37, 2016.

[3] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 779–788, 2016.

[4] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, USA, pp. 580–587, 2014.

[5] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask RCNN," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 2961–2969, 2017.

[6] R. Girshick, "Fast R-CNN," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Santiago, Chile, pp. 1440–1448, 2015.

[7] S. Ren, K. He, R. Girshick and J. Sun, "Faster RCNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, Long Beach, USA, pp. 91–99, 2017.

[8] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Santiago, Chile, pp. 7263–7271, 2015.

[9] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv: 1804.02767, 2018.

[10] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara *et al.,* "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, USA, pp. 7310–7311, 2016.

[11] Z. Zheng, G. Zhou, Y. Wang, Y. Liu, X. Li *et al.,* "A novel vehicle detection method with high resolution highway aerial image," *IEEE Journal of Selected Topics in Applied Earth, Observations and Remote Sensing*, vol. 6, no. 6, pp. 2338–2343, 2013.

[12] D. A. Pouliot, D. J. King, F. W. Bell and D. G. Pitt, “Automated tree crown detection and delineation in high-resolution digital camera imagery of coniferous forest regeneration,” *Remote Sensing of Environment*, vol. 82, no. 2–3, pp. 322–334, 2002.

[13] P. Zhu, L. Wen, X. Bian, H. Ling and Q. Hu, “Vision meets drones: A challenge,” arXiv preprint arXiv: 1804.07437, 2018.

[14] Y. Yang, Z. Lin and F. Liu, “Stable imaging and accuracy issues of low-altitude unmanned aerial vehicle photogrammetry systems,” *Remote Sensing*, vol. 8, no. 4, pp. 316, 2016.

[15] M. Li, Z. Zhang, K. Huang and T. Tan, “Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection,” in *19th Int. Conf. on Pattern Recognition*, Tampa, USA, pp. 1–4, 2008.

[16] A. T. Nghiem, E. Auvinet and J. Meunier, “Head detection using Kinect camera and its application to fall detection,” in *11th Int. Conf. on Information Science, Signal Processing and Their Applications*, Montreal, Canada, pp. 164–169, 2012.

[17] S. Yang, B. Fang, W. Tang, X. Wu, J. Qian *et al.,* “Faster R-CNN based microscopic cell detection,” in *Int. Conf. on Security, Pattern Analysis and Cybernetics*, Shenzhen, China, pp. 345–350, 2017.

[18] K. Fujisaki, A. Hamano, K. Aoki, Y. Feng, S. Uchida *et al.,* “Detection and tracking protein molecules in fluorescence microscopic video,” in *First Int. Symp. on Computing and Networking*, Matsuyama, Japan, pp. 270–274, 2013.

[19] J. S. Park, M. J. Oh and S. Han, “Fish disease diagnosis system based on image processing of pathogens’ microscopic images,” in *2007 Frontiers in the Convergence of Bioscience and Information Technologies*, New York, US, IEEE, pp. 878–883, 2007.

[20] M. Maitra, R. K. Gupta and M. Mukherjee, “Detection and counting of red blood cells in blood cell images using Hough transform,” *International Journal of Computer Applications*, vol. 53, no. 16, pp. 13–17, 2012.

[21] G. Li, T. Liu, J. Nie, L. Guo, J. Malicki *et al.,* “Detection of blob objects in microscopic zebrafish images based on gradient vector diffusion, Cytometry Part A,” *Journal of the International Society for Analytical Cytology*, vol. 71, no. 10, pp. 835–845, 2007.

[22] G. Cheng and J. Han, “A survey on object detection in optical remote sensing images,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 11–28, 2017.

[23] J. Han, D. Zhang, G. Cheng, L. Guo and J. Ren, “Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3325–3337, 2014.

[24] W. Zhang, X. Sun, K. Fu, C. Wang and H. Wang, “Object detection in high-resolution remote sensing images using rotation invariant parts-based model,” *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 1, pp. 74–78, 2013.

[25] G. Cheng, P. Zhou and J. Han, “Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7405–7415, 2016.

[26] K. Li, G. Cheng, S. Bu and X. You, “Rotation-insensitive and context-augmented object detection in remote sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 4, pp. 2337–2348, 2017.

[27] W. Diao, X. Sun, X. Zheng, F. Dou, H. Wang *et al.,* “Efficient saliency-based object detection in remote sensing images using deep belief networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 2, pp. 137–141, 2016.

[28] Y. Long, Y. Gong, Z. Xiao and Q. Liu, “Accurate object localization in remote sensing images based on convolutional neural networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 5, pp. 2486–2498, 2017.

[29] Y. Cao, X. Niu and Y. Dou, “Region-based convolutional neural networks for object detection in very high-resolution remote sensing images,” in *12th Int. Conf. on Natural Computation, Fuzzy Systems and Knowledge Discovery*, New York, US, IEEE, pp. 548–554, 2016.

[30] H. Tayara and K. T. Chong, "Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network," *Sensors*, vol. 18, no. 10, pp. 3341, 2018.

[31] H. Cholakkal, J. Johnson and D. Rajan, "Backtracking spatial pyramid pooling-based image classifier for weakly supervised top-down salient object detection," *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 6064–6078, 2018.

[32] A. Van Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," arXiv preprint arXiv: 1805.09512, 2018.

[33] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo *et al.,* "DC-SPP-YOLO: Dense connection and spatial pyramid pooling-based YOLO for object detection," *Information Sciences*, vol. 522, no. 9, pp. 241–258, 2020.

[34] J. Pang, C. Li, J. Shi, Z. Xu and H. Feng, "R2-CNN: Fast tiny object detection in large-scale remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5512–5524, 2019.

[35] M. Gao, R. Yu, A. Li, V. I. Morariu and L. S. Davis, "Dynamic zoom-in network for fast object detection in large images," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, USA, pp. 6926–6935, 2018.

[36] J. Shermeyer and A. Van Etten, "The effects of super-resolution on object detection performance in satellite imagery," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, Long Beach, USA, 2019.

[37] N. Audebert, B. Le Saux and S. Lefèvre, "Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images," *Remote Sensing*, vol. 9, no. 4, pp. 368, 2017.

[38] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *Journal of Visual Communication and Image Representation*, vol. 34, no. 10, pp. 187–203, 2016.