Visual Perception for Smart City Defense Administration and Intelligent Premonition Framework Based on DNN

Debosmit Neogi, Nataraj Das, and Suman Deb

Abstract A detailed methodology of object detection in a smart city setting has been illustrated in this chapter. The presented methodology focuses on intelligent uses of machine learning and deep learning algorithms for the effective extraction of the desired ROI from a challenging backdrop. This chapter encompasses a Convolutional Neural Network (CNN)-based architecture and a Faster RCNN-based approach and holistic comparisons are being made between the results obtained from the different approaches. The problem that has been tried to address through this work is the lack of robust algorithms that can detect occluded objects effectively, which may commonly occur in a smart city. So, the primary focus of this work is to devise a methodology that can detect even minute objects camouflaged in a city crowd, which are prevalent in smart cities across India. This work is believed to be beneficial in various sectors, even in the military, to help them with reconnaissance tasks. Further, the proposed methodology is equipped with an alarm system that warns against plausible security breaches and intrusion. This mechanism enhances the security of a smart city using IoT techniques. The entire methodology described in the chapter can be deployed without the use of any additional hardware. Overall, the proposed framework is robust, effective and viable for multi-facet uses in the future and can be effectively deployed in large distributed systems across smart cities of India.

Keywords Alerting framework · Defense · CNN · Faster RCNN · Smart city · Motional feedback · Object detection · ROI · Smart city

1 Introduction

Object detection and stratification always have been an important task. This is due to the vast array of real-life situations where object detection plays an important role. This task becomes even more daunting when the object is camouflaged with its background, making the detection task even more difficult and challenging. The problem addressed in this work is believed to prove extremely valuable when ROI is

D. Neogi · N. Das · S. Deb (✉)
Deptartment of Computer Science and Engineering, NIT Agartala, Agartala, India

V. Piuri et al. (eds.), *AI and IoT for Smart City Applications*, Studies in Computational Intelligence 1002, https://doi.org/10.1007/978-981-16-7498-3_7

perfectly hidden. Such scenarios include army reconnaissance tasks, wildlife census, etc.

In the military, it is very important to track the movements of people. Even small negligence can prove to be deadly. The Indian army usually operates in challenging battle field conditions, usually covered with dense forest. In such a situation, where visibility is extremely low and enemies can easily camouflage in the dense backdrop, the proposed methodology can work very well. The proposed framework is specially built to extract and detect ROI.

The most important scenario where the methodology is tested is urban defense. With the growing urbanization and development of smart cities, there has been a steady surge in crime rates. Theft and burglary in houses are very common. This architecture can resolve this issue to a great extent and help in making a better defense system for houses and smart cities in general. The described framework can very well be deployed in houses which helps in enhancing securities.

We have implemented a novel idea of the alarm system in this methodology. The alarm system uses simple conditional statements and based on that, the alarm rings. This technology was built keeping in mind the use case of the methodology in case of theft and burglary in houses and also in military settings. If our system detects any object in the wild, the alarm raises an alert signal against a potential security breach. This can be very beneficial as it alarms the people during the night and helps in securing the apartments, which are very common in smart cities. With the rapid transformation of smart cities, there is an associated growth of shopping malls, amusement parks and most importantly banks and ATMs. The defense mechanism proposed here can add an extra layer of security in these public places, thus boosting the security by keeping the intruders at bay. Figure 1 describes a potential security issue in banks across smart cities. The incorporation of our framework ensures an extra protective layer with its alarming mechanism giving time to security officials to react and respond to the situation swiftly.

Fig. 1 Bank defense system

The whole architecture is light and is computationally inexpensive. In addition to that, any requirement of additional hardware has been completely removed. Only a high-resolution camera is a prerequisite for the smooth working of the model. The architecture has been tested upon various parameters, and it fares well against other frameworks.

2 Literature Review

A number of researches have been done in the past that made use of several techniques and algorithms in order to build an intelligent and smart surveillance system for the military: [1] used sensors for collecting intelligence for the army, and [1] used several distributed, wireless communicating smart sensors for surveillance, information collection and reconnaissance system. However, the above-stated researches are involved with a high budget as a result of the costly sensors and hardware for achieving the desired output. To this contrast, the stated methodology is believed to be quite economical as it strongly eliminates the use of costly sensors to conduct surveillance; [2] also worked on a similar theme but used a high-resolution camera. Numerous algorithms and techniques were used by researchers for object detection problems. For instance, among recent works, [3] proposed an algorithm that is capable of detecting entities from an unknown environment; [4] have been referred as they contributed significantly toward deep neural approach toward the detection of physical entities. Reference [5] deployed a multi-scale CNN approach for the detection of entities. Last but not the least, [6] have contributed toward the detection algorithm.

Several researchers have worked on camouflaged object detection. Reference [7] used trans-former networks in their approach. Reference [8] used a video-based input detection mechanism for object detection. A group of researchers [9] have worked on multiple object detection. All their results have been nicely covered in our proposed methodology. Reference [10] worked on creating an alarm system to prevent accidents. Our proposed framework utilizes a similar theme and leverages it along with the object detection mechanism.

3 Dataset and Feature Engineering

For the purpose of training the faster RCNN model, Open Image Dataset (OID) v4 has been taken as the dataset. It can be further used to create a custom dataset that detects classes of our choice. This dataset contains sufficient labeled and annotated images in the training and validation sets for custom classes. The entire dataset is cleaned and preprocessed before training the faster RCNN model. The CNN model was trained on a dataset that contains 75,000 images of trucks which were to make sure the model works perfectly. d Prior to feeding the dataset to the model, a number

of feature engineering is applied in order to shape the data for training purposes. All the images were subjected to pixel adjustment of 90 × 90 pixels. The images were then scaled as well as cleaned. Finally a matrix of 24,300 × 75,000 pixels values obtained. The images assumed to be 3 channel RGB that are finally transformed to a comma-separated value(CSV) file and was used as independent training values for feeding the model.

A similar approach was leveraged to train the model on a gun dataset. The dataset contains 10,000 images of guns. Again, feature engineering is applied to the dataset in order to create total features of 90 × 90, resulting in a matrix of 24,300 × 10,000 dimensions. Both the above datasets are concatenated in order to create a single data file. Even this dataset was subjected to several cleaning and preprocessing stages before making it suitable for training.

4 Synopsis of Proposed Methodology

The proposed methodology is the sequential implementation of a CNN-based and a faster RCNN-based model. The very first step is to apprehend optical feedback. A video is a sequence of images being played with an fps of 60 or more. Considering this fact, we have segmented the video into frames that are images and used that image for further analysis. In light of that purpose, we divided the obtained feedback into two frames at a time.

But before doing any further steps, detection of movement is required. To achieve this, a series of actions is performed to detect movement by calculating the absolute difference between 2 successive frames from the camera feedback. Such difference between frames within a fraction of a second was capable of detecting the very minute and trivial movements. Upon receiving the object in motion, certain morphological transformations are applied such as thresholding and Gaussian blur to achieve a robust and effective visual in order to draw out the contours around the moving entities. Depending upon the area of the captured object, contours are drawn around the object in order to extract the ROI.

Then the detected ROI is first passed through a series of convolution layers, max-pool and activation layers. In the next model, the ROI is passed through a whole faster RCNN-based architecture for stratification. The result obtained is further used for analysis and result validation. Figure 2 depicts the entire methodology of the proposed research work.

5 Classification of Extracted ROI

Classification of ROI is the most important step in the entire pipeline. This step has to be accurate and precise. This step ensures the overall viability of the entire framework. Latest deep learning and state-of-the-art algorithms have been used to

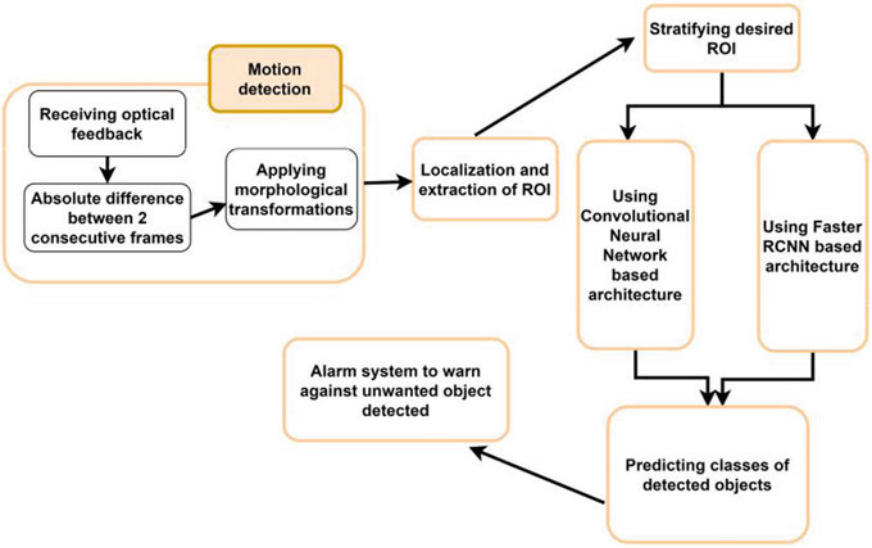

Fig. 2 An overview of the working framework

ensure maximum accuracy such that the framework works well in solving smart city security problems.

5.1 CNN

The very fast stratification of extracted ROI was done through convolution of the image and using artificial neural networks alongside in order to stratify the ROI [11]. To be precise, convolutional neural networks were deployed in order to obtain the desired output. The data feature points were plotted in order to determine the scaling of the issues as well as to draw an initial point of view regarding the classification report.

Figure 3 clearly depicts [12] how well a classification algorithm will perform in order to draw the desired classification report. The desired ROIs are then convoluted with a kernel of specific dimension and value. Since the extracted ROIs were RGB images, they were having a dimension of 3 units. Thus, it was required to perform 2D convolution in order to obtain the feature map. In total, 32 filters were deployed with 3×3 kernel size. Deployment was accompanied by the ReLU activation function. After that, maximum pooling operation was deployed on the earlier created feature map. In order to perform the max-pooling operation, another kernel of size 2×2 was taken into consideration. Finally, the flatten layer was introduced in order to push all the features. The architecture of the classification neurons was considered to be a sequential one and dense network distribution. A total of 6000 neural hidden

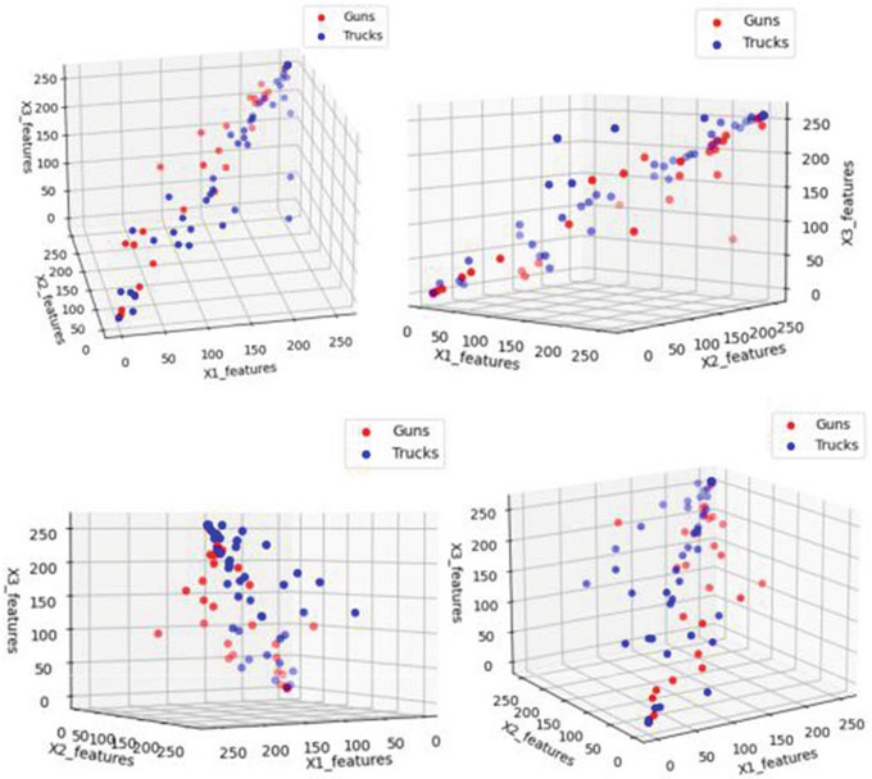

Fig. 3 Feature plot of the data points

layers were created with ReLU as an activation function. The final output layer was designed corresponding to the number of classification targets with Sigmoid as the activation function. Finally, the neural model was compiled with Adam optimizer [13, 14] instead of traditional SGD [15, 16]; Sparse categorical cross entropy as the loss and accuracy as the metrics.

Figure 4 gives a visual of the entire process that has been carried out in order to obtain the desired stratified classes.

5.2 Faster RCNN

Faster RCNN [17] is the most effective and robust model when compared to RCNN or fast RCNN. This is because it solves all the bottlenecks of the Fast RCNN model [18]. There are 2 paramount steps that describe the entire algorithm of Faster RCNN: Region Proposal Network and Fast RCNN as detector algorithm. The first step of the architecture is to feed our image input into the Convolutional Neural Network.

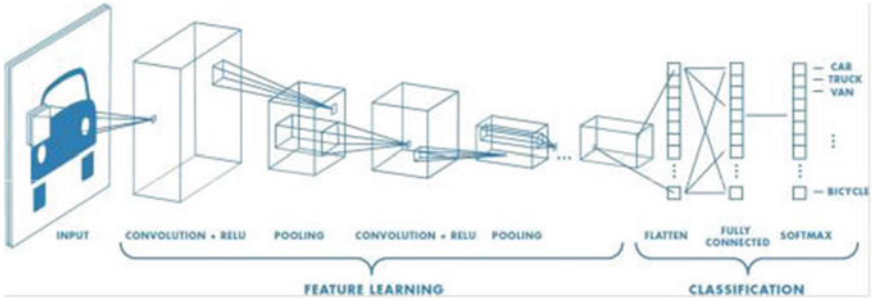

Fig. 4 Algorithmic stratification strategy: CNN

The image size has been resized to 100 × 100 pixels. We have tested our model with 2 different stride values. First, we have set a stride value of 1 both vertically and horizontally. The model is first built upon the ResNet model. But later, VGG-16 was introduced which ultimately gave a better result. There are 4 max-pooling layers in the VGG-16 architecture [19]. Similarly, a stride value of 2 has been considered to test the maximum potential of the detection algorithm. As mentioned earlier, $2^4 = 16$. This means that upon moving a unit step in the feature map, a total of 16 pixels are traversed in input. Another important factor contributing toward this result is the selection of appropriate anchors of various aspect ratios. For this research, a set of aspect ratio values of 1:1, 1:2 and 2:1 have been leveraged (Fig. 5).

Training and loss function

The loss function implemented in this proposed methodology, as mentioned by [17], is

$$L_i(\{p_i\}, \{t_i\}) = 1/N_{cls}\left(\sum L_i(p_i, p_i^*)\right) \tag{1}$$

$$L_i(\{p_i\}, \{t_i\}) = L_i(\{p_i\}, \{t_i\}) + \lambda / N_{reg}\left(\Sigma_i p_i^* L_{reg}(t_i, t_i^*)\right) \tag{2}$$

6 Viable Result Analysis

The proposed methodology has been put through numerous critical analyzing factors and depending on the feedback, the framework has been reconfigured in order to achieve the best possible outcome.

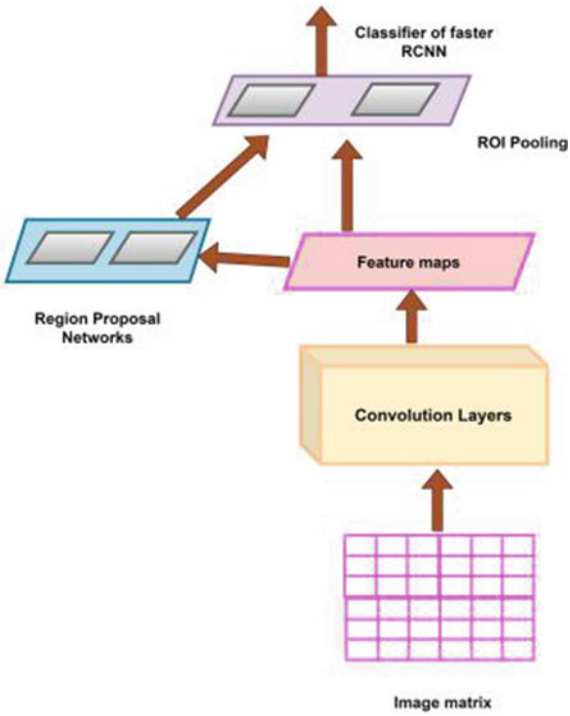

Fig. 5 Algorithmic stratification strategy: faster RCNN

6.1 Alerting for Engagement

During warfare, situations arise where stealth movements occur, which sometimes create a lot of casualties as the oppositions manage to come very close to the operating base [20]. The proposed method also deploys an alarming system that starts ringing as soon as there is some motion on the optical feedback. The algorithm that is used is quite simple which is a rule-based one.

$$Alert\_System = 1;\ \text{if absdiff}_{\text{i}}(\text{f}_{\text{i}} -\!- \text{f}_{\text{i+1}})\,! = 0$$

$$Alert\_System = 0;\ \text{if absdiff}_{\text{i}}(\text{f}_{\text{i}} -\!- \text{f}_{\text{i+1}}) = 0$$

In the above equation, 1 and 0 refer to the logical on and off for the alarm system. Thus, whenever the absolute difference between the frames returns zero, the alert system remains off, but as soon as the return type is not false, i.e. true, the alarm turns on. The deployment of the above algorithm makes the alert system highly robust and precise. However, the problem of alerting against noises was solved using a real ratio of detected motional entities based on their distance from the camera.

6.2 *Performance Analysis of CNN Model*

In order to analyze the result of the proposed framework [21], it was necessary to tune the parameters of the first classification algorithm, i.e. the CNN. The significant hyper-parameters were tuned through hyper-tuning functions. Ultimately, the following parameters were obtained:

Parameter	Value
Model	Sequential
Architecture	Dense
Convolution	2D
Filters	32
Kernel	3 × 3
Stride	32
Pool size	MAX, 3X3
Dropout	0.3
Optimizer	Adam
Loss	S.C.Cr.E
Accuracy	98.7
Loss	2.19

The ReLU function played a vital role as an activation function in both feature map creation as well as in hidden neural layers. Finally, the ROC and AUC of the CNN algorithm were determined and plotted.

Figure 6 depicts the performance of the CNN algorithm in obtaining the stratified ROI.

The precision, robustness and result analysis are carried out by creating a confusion matrix [22].

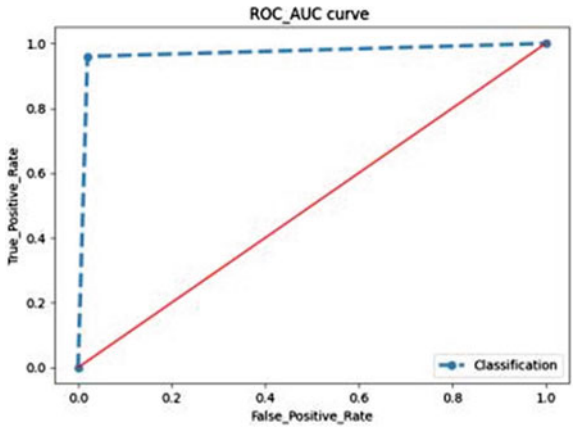

Fig. 6 ROC and AUC for CNN

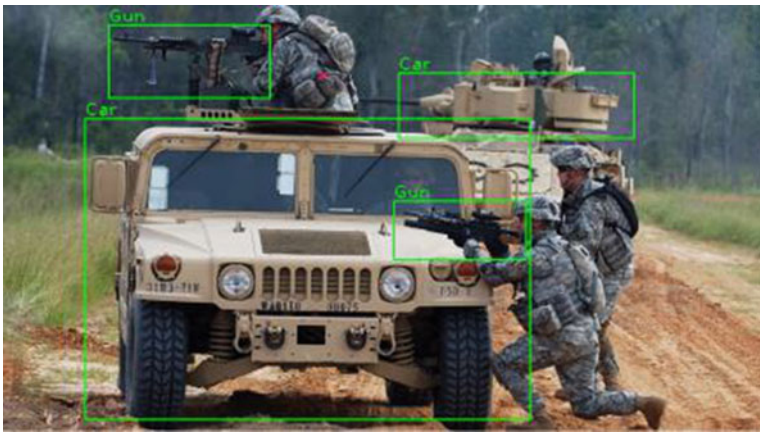

Fig. 7 Classification of extracted ROI using CNN model (Deadlock 1983)

$$A = \begin{bmatrix} 48 & 2 \\ 1 & 49 \end{bmatrix}$$

In the above matrix 6.2 A, the columns are the predicted dataset and rows are the actual ones. Here, the first and second columns represent the gun and truck datasets, respectively. Likewise, the first row represents the actual class of guns and the second row represents trucks. Several parameters have been used to validate the performance of the model [23]. Applying those parameters to the matrix 6.2 A, we have values as the following.

Thus, an average F1 score of 0.97 has been recorded [24]. The model is gives high accuracy of classification. Figure 7 talks about the visual of the physical output of the model. The model gives correct prediction in the image taken in foggy conditions with reduced visibility.

6.3 Performance Analysis of Faster RCNN Model

The model works well when tested under different conditions (Fig. 8). In Fig. 9, the performance of the model is shown. The model does extremely well in detecting the "person" in camouflage in both scenarios. It is equally capable of detecting birds and animals in challenging situations. The model detects a cat from a dark environment and classifies it correctly with a confidence score of 68.75%. It even does well to detect the precise ROI of a parrot in green surroundings and classifies it correctly with a confidence score of 69.92%.

Further, to improve the performance of the proposals generated by the network, we label anchor targets as either 0 or 1. Considering the scenario where anchor's and ground truth's IOU 0.7, the anchor has been labeled as 1. In the proposed

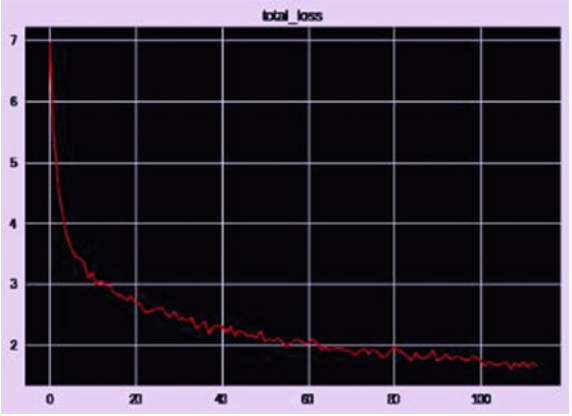

Fig. 8 Epochs versus net loss for faster RCNN training

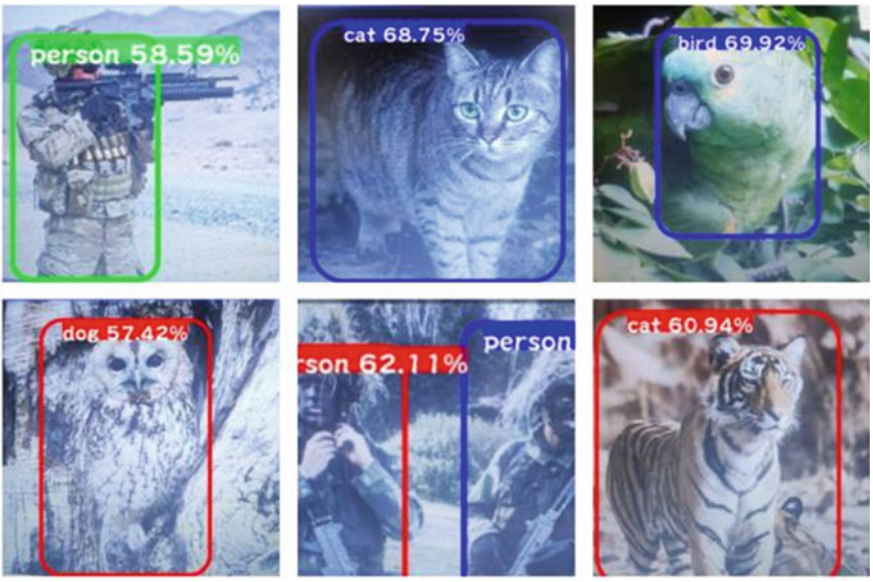

Fig. 9 Object detection using faster RCNN in different conditions (https://unsplash.com/)

methodology, for the object detector network, Binary Cross Entropy [25] has been implemented to calculate the net loss.

The Maximum Average Precision (mAP @ 0.5) and overall mAP score achieved after training the Faster RCNN model on OID v4 were approximately 43.5 and 75%, respectively. The score achieved is quite extraordinary given the challenging conditions on which the model was tested. The initial learning rate (alpha) was set as 0.05 which was tuned over the course of training.

7 Conclusion

In line with the objective of the proposed methodology, a robust object detection and classification algorithm has been achieved. Both CNN-based architecture and Faster RCNN-based models work extremely well when tested in different conditions. The Faster RCNN-based model works particularly well in testing and difficult situations such as low light and objects in camouflage. Both the models have been greatly modified from their respective vanilla architecture, to achieve a state of robustness.

There were some shortcomings too in this methodology. In Fig. 9, the framework misclassified the "owl" and the "tiger" as a dog and a cat, respectively. Although the framework did extremely well in extracting the desired ROI from heavy camouflage, the classification was wrong. This can be solved by taking a stronger dataset to train the models. In the rest of all other cases, the model did a good job in extracting the desired ROI and correctly classifying them. The alerting methodology executed with the rule-based algorithm worked with high precision and robustness. However, an initial problem was encountered, which was unnecessary alerting for minute movements of trees birds, etc. in the optical feedback. This problem was solved by adjusting the contour area on the basis of a specific distance from the camera, keeping in mind the purpose it serves. Maintaining the contour area reduces the unnecessary detection of noise and latent vectors, thus contributing toward its viability.

The main objective of enhancing the security of smart cities by acting as a defensive mechanism has been achieved. The system is tested on different grounds and it has returned high accuracy. The viability of the framework comes from the fact that it has been tested in crowded places. This is a very important aspect as smart cities are usually very crowded due to available job opportunities and promising quality of life. In short, it can be rightly concluded that the proposed framework works with high precision and robustness. The computational power is low and also eliminates all hardware dependencies. The proposed framework, thus, can be deployed in real life in developing and developed smart cities of India such as Agartala, Silchar, Durgapur, Indore and Varanasi. This framework along with minute modifications can be an effective tool in solving a large array of security problems in smart cities that can be classified as detection-related conundrums.

References

1. Astapov S, Preden J-S, Ehala J, Riid A (2014) Object detection for military surveillance using distributed multimodal smart sensors. 08
2. Rajjak SSA, Kureshi AK (2019) Recent advances in object de-tection and tracking for high resolution video: overview and state-of-the-art. In: 2019 5th International conference on computing, communication, control and automation (ICCUBEA), pp 1–9
3. Prasad S, Sinha S (2011) Real-time object detection and tracking in an unknown environment. In: 2011 World congress on information and communication technologies, pp 1056–1061
4. Zhong-Qiu Zhao SX, Zheng P, Wu X. Object detection with deep learning: a review. Neural networks and learning systems

5. Chen Z, Gao L, Cao D (2020) An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks. Hum Cent Comput Inf Sci 10:14
6. Sharma K, Thakur N (2017) A review and an approach for object detection in images. Int J Comput Vis Robot 7:196
7. Mao Y, Zhang J, Wan Z, Dai Y, Li A, Lv Y, Tian X, Fan D-P, Barnes N (2021) Transformer transforms salient object detection and camou aged object detection
8. Mukherjee S (2021) Object detection, pp 159–170
9. Singh A, Kumar T (2021) Multiple object detection, pp 659–664
10. Dr Maturkar, Dudhe K, Roy K (2021) Accident identication and alerting system. Int J Adv Res Sci Commun Technol 766–774
11. Albawi S, Mohammed TA, Al-Zawi S (2017) Understanding of a convolutional neural network, pp 1–6
12. Guyon AEI (2003) An introduction to variable and feature selection. J Mach Learn Res 3(2003):1157–1182
13. Kingma D, Ba J (2014) Adam: A method for stochastic optimization. In: International conference on learning representations
14. Kingma DP, Ba J (2017) Adam: A method for stochastic optimization
15. Wang Y, Zhou P, Zhong W (2018) An optimization strategy based on hybrid algorithm of Adam and sgd. MATEC Web Conf 232:03007
16. Ruder S (2016) An overview of gradient descent optimization algorithms. CoRR, abs/1609.04747
17. Ren RGS, He K, Sun J (2015) Faster r-cnn: towards real-time object detec-tion with region proposal networks. In: Advances in neural information processing systems, pp 91–99
18. Girshick R (2015) Fast r-cnn. arXiv:1504.08083
19. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition
20. Sarkar P, Singh A, Md Islam (2019) Emergency alert system for women's safety. IJIREEICE 7:53–55
21. Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santa-mara J, Fadhel MA, Al-Amidie M, Farhan L, Alzubaidi L, Zhang J (2021) Review of deep learning: concepts, cnn architectures, challenges, applications, future directions
22. Susmaga R (2004) Confusion matrix visualization. In: Mieczys law A Klopotek, Slawomir T Wierzchon, Trojanowski K (eds) Intelligent information processing and web mining, pp 107–116. Springer, Berlin Heidelberg
23. Pianosi F, Beven K, Freer J, Hall JW, Rougier J, Stephenson DB, Wagener T (2016) Sensitivity analysis of environmental models: a systematic review with practical work ow. Environ Model Softw , 79:214–232
24. Sokolova M, Japkowicz N, Szpakowicz S (2006) Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation, vol 4304, pp 1015–1021
25. Ruby U, Yendapalli V (2020) Binary cross entropy with deep learning technique for image classification. Int J Adv Trends Comput Sci Eng 9:10