PRIYANKA, M., LAVANYA, K., SAI, K.C., ROHIT, K. and BANO, S. 2022. Comprehending object detection by deep learning methods and algorithms. In *Shakya, S., Ntalianis, K. and Kamel, K.A. (eds.) Proceedings of the 3rd International conference on mobile computing and sustainable informatics (ICMCSI 2022), 27-28 January 2022*, Kirtipur, Nepal. Lecture notes on data engineering and communications technologies, 126. Singapore: Springer [online], pages 523-537. Available from: https://doi.org/10.1007/978-981-19-2069-1_36

Comprehending object detection by deep learning methods and algorithms.

PRIYANKA, M., LAVANYA, K., SAI, K.C., ROHIT, K. and BANO, S.

2022

Comprehending Object Detection by Deep Learning Methods and Algorithms

Mallineni Priyanka*, Kotapati Lavanya, K. Charan Sai, Kavuri Rohit, and Shahana Bano

Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, India

Abstract In the real world, computer vision is used for more challenging tasks like detecting of objects in an image or video. There are multiple applications of object detection in various domains like animation, autonomous driving, monitoring of traffic, communicating through video. With the development of new emerging technologies in deep learning, finding accuracy of objects by performing classification and detection became possible. When compared to traditional object detection methods deep learning methods has an ability of feature learning and rendering. This paper is mainly focused on the working procedure of convolutional neural networks in detecting objects that are present in the environment of an image. CNN, R-CNN, and Faster R-CNN are the main models of deep learning which are considered for comparative-based study. Comparison between these models is made by identifying their accuracies, limitations, and speed. Among the three models, Faster R-CNN is identified as ideal one as it has higher accuracy and less expensive in nature when compared with R-CNN whereas CNN model can be only used for image classification (Tripathi in Journal of Innovative Image Processing (JIIP) 3:100–117, 2021), but it cannot localize the objects.

Keywords:- Deep learning; Convolutional neural network (CNN); Region-based CNN; Faster R-CNN; Selective search

1 Introduction

To acquire a proper understanding on an image, just focusing on classifying the images is not the final, but also on estimating the locations in a video or image also plays a key role [1]. This task is referred as object detection. Object detection works by creating bounding boxes around one or more objects in an image or video [2]. The exposure of deep learning technology has drastically transformed the traditional ways of object detection [3, 4]. Among the wide range of applications of object detection [5], some of the trending applications are driverless cars, video surveillance [6], activity recognition, pedestrian detection. Convolutional neural network (CNN) is used to detect objects by achieving high levels of accuracy. CNN is made up of several layers, including an input layer, an output layer, and it also at least consists one hidden layer.[7] They are great for distinguishing patterns like edges, forms, colors, and textures in object detection.

2 Literature Survey

Gavrilescu et al. [8] used the most up-to-date image processing methods to detect traffic signals while driving a car in this work. The R-CNN method, which is faster, produces good accuracy and speed, making it acceptable for usage in applications. RPN and Fast R-CNN are the two modules that make up Faster R-CNN. The major purpose of this study is to demonstrate how the Faster R-CNN method may be used to speed up the detection of objects in images.

Tang et al. [9] explains about the methods in object detection and explicate the similarities and difference between traditional and deep learning methods. Deep learning models based on regional proposals are explored by understanding there working process, limitations, challenges, and solutions.

Abushahma et al. [6] described in a clear way by considering an image which contains single, double, triple objects for detection and compared their average accuracies.

Wei et al. [2] used deep learning algorithms for detecting small objects. Two-stage algorithms are used, and object detection methods based on classification are also known as two-stage algorithms since they are divided into two steps. Two-stage algorithms include R-CNN, SPP-Net, Fast R-CNN, Faster R-CNN. One-stage algorithm is used, and object detection algorithm based on regression is known as one-stage algorithm. It includes YOLO, SSD, and other algorithms. It essentially distinguishes between one-stage and two-stage detection algorithms, with one-stage algorithms having a faster detection speed and two-stage algorithms having a higher detection accuracy.

Malhotra et al. [10] used R-CNN, Fast R-CNN, and YOLO algorithms for object detection. In this R-CNN, Fast R-CNN and YOLO algorithms are compared. When compared to traditional object detection methods, these algorithms train and compute the model's parameters in a faster and more efficient manner, resulting in improved performance. YOLO is faster than R-CNN and Fast R-CNN.

Algorry et al. [11] used YOLO v2 and convolutional neural network for object detection of small and similar figures. It is difficult with YOLO v2 to detect small objects like traffic signs, plants, etc. By using convolutional neural network, it gives good accuracy and faster speed by detecting small objects and similar figures.

Understanding of deep neural networks [12] for object detection is made by reviewing the reference papers. Most of the papers made a comparative based study on convolutional neural network algorithms like R-CNN, faster R-CNN, YOLO, SPP-Net, etc., and how they work on real-world applications. By referencing this papers, CNN, R-CNN, Faster R-CNN models from deep neural networks were taken for object detection as key models for the study that helps us in analyzing the working procedure, limitations, accuracy in detection, advantages, disadvantages of using these models in real-time applications.

3 Convolutional Neural Networks

CNN includes different layers in which it consists an input layer, an output layer, and it also includes at least one hidden layer [13]. Each CNN layer is referred to as a feature map. The effect of applying the filters to an input image is captured by the feature maps of a CNN [14]. The feature map is the output of each layer; i.e., at each layer [15], the feature map is the output of that layer. The goal of visualizing a feature map for a specific input image is to learn more about the features that CNN [10] finds. In this sort of neural network, the hidden layers are convolutional layers [16] that operate as a filter, receiving input, transforming it using a particular pattern, and sending it to the next layer. When there are additional convolutional layers, they are altered in different ways each time a new input is supplied to the next convolutional layer. In CNN [7], at first, an image is taken as input then the image is divided into different parts. After that, each section is treated as a separate image. And all these regions are sent to the CNN in which it will classify into different categories. Once each region is divided into matching class to get original image, combine all the regions with the detected objects.

$$\text{Output} = \frac{I + (2 * \text{Pc}) - K1}{S1} + 1 \tag{1}$$

where in Eq. 1 output is the number of features of output, I refers to number of input features, Pc refers to padding size, $K1$ refers to kernel size, and $S1$ refers to stride size.

Pseudocode for CNN

Step 1: Start
Step 2: Import and load the image dataset
Step 3: Perform pre-processing operations on the data
Step 4: Visualize the training data
Step 5: Standardize the data
Step 6: Create and compile the CNN model
Step 7: Train the model and visualize the training results
Step 8: Predict the accuracy
Step 9: End.

4 Region Proposal Network

The object detection based on region proposal works on two key steps. The process of primary step is to extract the candidates regions [14]. Construction of deep neural networks is the other.

4.1 *R-CNN*

R-CNN [17] is a convolutional neural network based on Girshick's region proposal, which he proposed for the first time in 2014 [2, 10, 18]. The main focus of R-CNN is to enhance the standard of BBs of candidates and to extract high-level feature [18] by taking a deep architecture. For each image, the R-CNN generates approximately 2000 region proposals by adopting selective search [10]. Selective search identifies the different regions. Varying scales, colors, textures, and enclosure are the four basic regions to form an object. In an image, patterns are recognized by using selective search and later as a result various regions are proposed.

Pseudocode for R-CNN

Step 1: Start
Step 2: Import and load the dataset
Step 3: Perform pre-processing operations on the data
Step 4: Extract region proposals nearly 2000 and use selective search for find region of interest
Step 5: Compute CNN feature
Step 6: Classify the regions
Step 7: End.

4.2 *Faster R-CNN*

For object detection, a Faster R-CNN is used, which appears to the user as a single, unified network from beginning to end. It can also determine the positions of various things quickly and precisely. Faster R-CNN is made up of two modules [19]. The first module, rpn, is used to create region proposals, and it employs the neural network concept by directing the quick R-CNN to recognize the items in the image [20]. Fast R-CNN is the second module, which is utilized to recognize objects in the proposed regions.

The input image is initially adjusted so that its longer side does not exceed 1000 px, and its shortest side is 600 px by the region proposal network (RPN) [20], which then passes it to the convolutional neural network [8]. Based on the backbone network, the network's resultant output features [21] are smaller than the input image. The network must learn if the object is present or not in the provided input image at each point in the resultant feature map and estimate its size for each point in the subsequent feature map [22]. All of this is made feasible by putting a set of "Anchors" on the input image and the network's output feature map. The network feature map is the generated feature map.

A CNN backbone, a ROI pooling layer, and fully connected layers are followed by classification and bounding box regression [23] branches in the Fast R-CNN. After that, the input image is fed into the backbone CNN, which generates a feature map. The benefit of weight sharing between the quick R-CNN detector and the RPN backbone is the main argument for choosing an RPN as a proposal generator, aside from test time efficiency. The bounding box suggestions from the RPN are utilized to pool features from the backbone feature map, which are then done by the ROI pooling layer.

Pseudocode for Faster R-CNN

Step 1: Start.
Step 2: Import and load the dataset.
Step 3: Consider an input image, which you can then send to ConvNet, which will provide feature maps for it.
Step 4: On these feature maps, region proposal network are used in order to get object proposals.
Step 5: The ROI pooling layer is used to standardize the size of all proposals.
Step 6: To categorize and predict the bounding boxes for the image. Transmit these proposals to a fully connected layer.
Step 7: End.

5 Discussion

The article first explains about object detection and its important applications in the current trending world and how the deep learning methods have been used in the emerging technologies when compared with traditional methods for object detection. In the aspect of object detection, three deep learning methods were considered and a comparative-based study was performed among them to find a best method for object detection by considering all parameters and situations. The working procedure of CNN, R-CNN, and Faster R-CNN was explained in the form of flow charts and pseudocodes with a brief description. After performing this experiment, graphs for total loss and accuracy were plotted for better understanding of results. CNN detects the class of the testing image with a certain accuracy. R-CNN not only classifies the class of the objects in image but also localize the objects present in the image, whereas Faster R-CNN also detects the objects as same as R-CNN but with a higher accuracy and speed with less expensive nature. The main issue raised during the experimentation of these models is with R-CNN. Even though, CNN models has not justified the purpose of object detection but it in the aspect of speed and RAM usage CNN model works efficiently, whereas in the case of R-CNN, it took a lot of time and RAM for training the model. High RAM is required while working on R-CNN model.

6 Flow Charts

From Fig. 1, first start with importing required libraries and datasets. In the next following steps, data pre-processing steps, visualization of dataset is carried out. Create and train the CNN model using the training data. Classify and predict the accuracy.

From Fig. 2, in R-CNN, first import the libraries that are required and load the annotation csv file, image dataset. Initialize selective search for extraction of 2000 regional proposals from the given input image. Define IOU and perform transfer learning by importing VGG16 model on pre-processed data. Create and train the model by passing dataset into the model. Predict the validation loss and accuracy.

From Fig. 3, import dataset from OIDv4 tool kit that contains 600 different classes. From the tool kit extract the three annotation csv files which contains classes with their label names. Create bounding boxes; dataset classes are defined in class file and extract the images. Format the data into a table which is later converted into train and test data. RPN is used on these feature maps; and as a result, object proposals are obtained. For standardizing the size of proposals, ROI pooling layer is functioned. At last, classify and predict the objects.

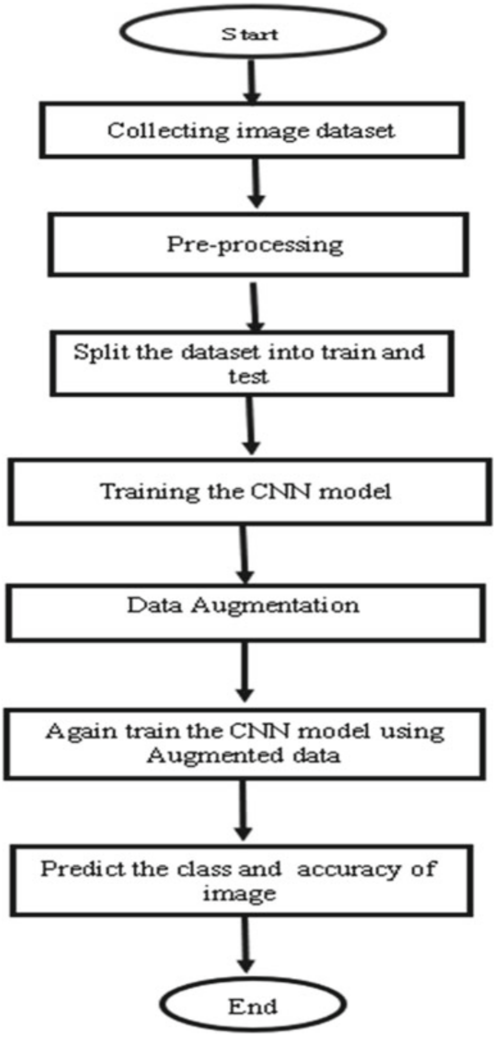
7 Results

Evaluation of deep learning models is performed in an open-source tool Google Collab.

7.1 Accuracy Table for CNN Model

Different datasets like flowers, monkeys, etc., shown in Table 1 were used for analyzing the CNN model. For each dataset, CNN models results the class of the image to which it belongs to with a certain accuracy percentage. The total number of parameters considered for evaluation is 56,320.

Fig. 1 CNN flow chart



7.2 Accuracy Table for R-CNN

R-CNN model is analyzed by using a small dataset due to the expensive nature in the aspects of speed and RAM usage. Accuracy table (Table 2) is plotted based on the testing accuracy and loss. The total number of parameters considered for this experimentation is 138,357,544.

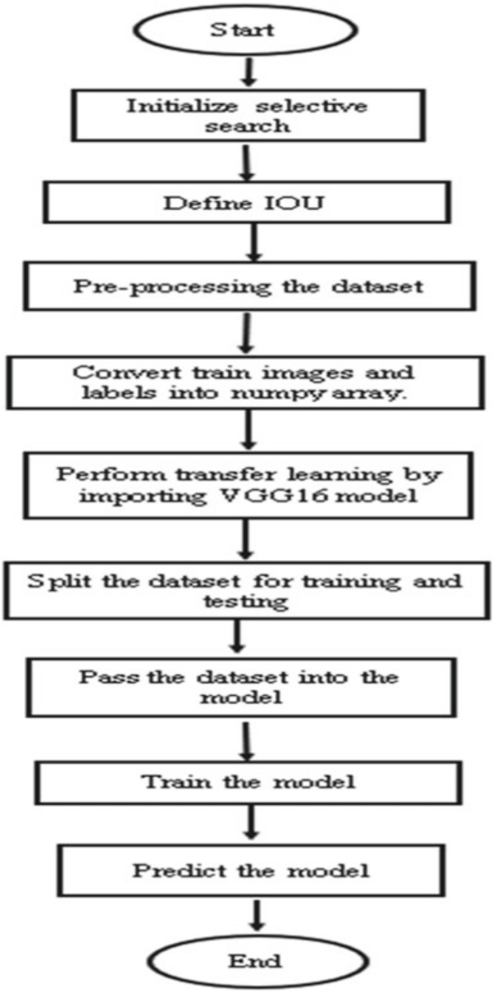


Fig. 2 R-CNN flow chart

7.3 Accuracy Table for Faster R-CNN

For analyzing the Faster R-CNN model, OIDv4 tool kit is used for downloading the dataset. A dataset containing apples, oranges, and chairs as class names was chosen. Table 3 contains accuracy and loss of testing data.

Fig. 3 Faster R-CNN flow chart

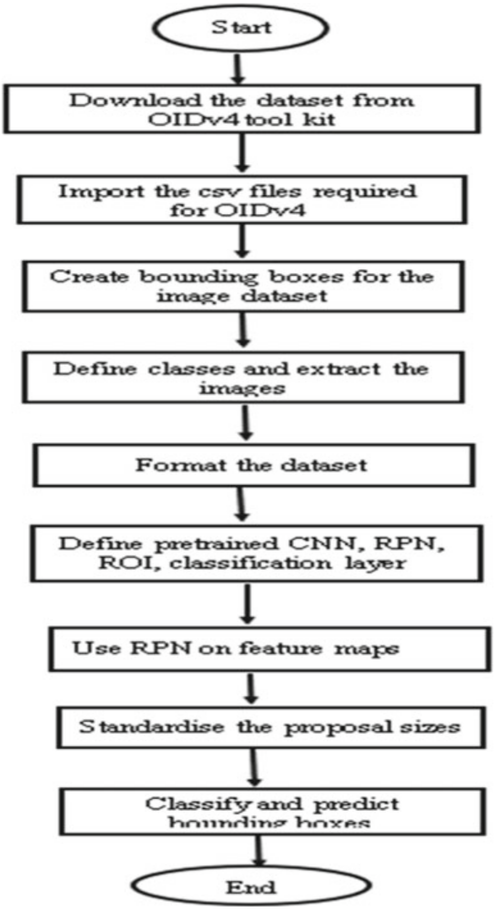


Table 1 CNN accuracy table

Dataset	Accuracy (%)
Flowers	98.95
Monkeys	66.56
Animals	77.61
Fruits	66.70
Expressions	99.09
Different objects	76.14

7.4 CNN

After training the model by considering the training and validation accuracy, loss graphs are plotted in Fig. 4. For better results, data augmentation is performed which

Table 2 R-CNN accuracy table

Epoch	Accuracy (%)	Loss (%)
1	84	33
2	87	33
3	90	21
4	93	17
5	96	8

Table 3 Faster R-CNN accuracy table

Epoch	Accuracy (%)	Total loss (%)
10	82	15
20	87	13
30	89	12
40	92	8

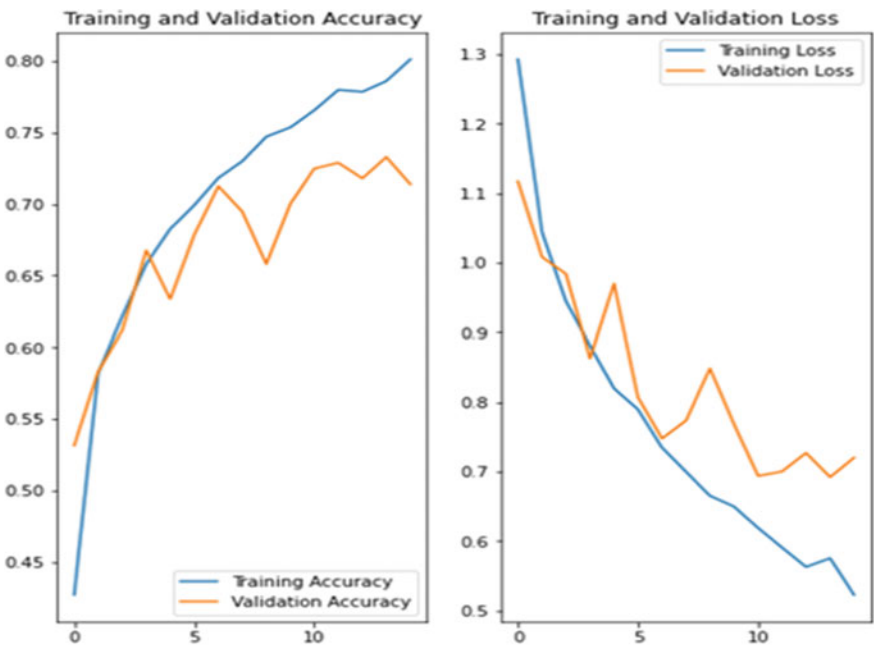


Fig. 4 Accuracy and loss graphs for CNN model

is helpful in expanding the size of training images. After data augmentation, there is an increase in accuracy and decrease in loss when compared to the data which is used before performing data augmentation.

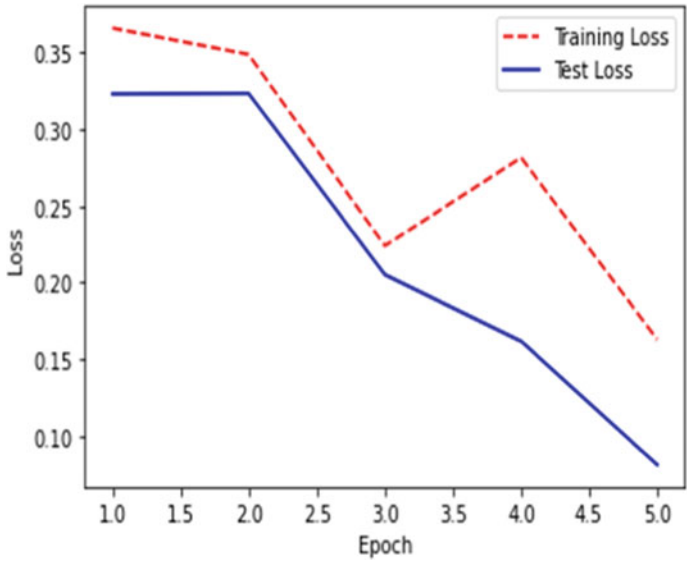


Fig. 5 Loss graph for R-CNN model

7.5 R-CNN

After training the model, graphs are plotted for finding the difference between training and testing loss, accuracy. Figure 5 shows the decrease in loss as the number of training steps increases, the dotted line represents the training loss, and the blue line indicates the testing loss. Figure 6 shows the increase in accuracy as the number of training steps increases, the dotted line represents the training accuracy, and the blue line indicates the testing accuracy.

7.6 Faster R-CNN

For Faster R-CNN, as number of training steps increases, accuracy increases and total loss decreases. Figure 7 shows the graph between accuracy versus number of training steps. Figure 8 shows the graph between loss versus number of training steps.

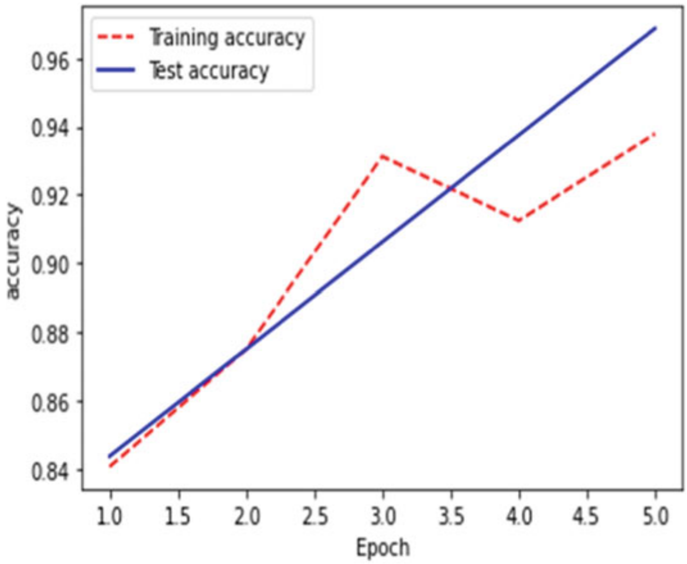


Fig. 6 Accuracy graph for R-CNN model

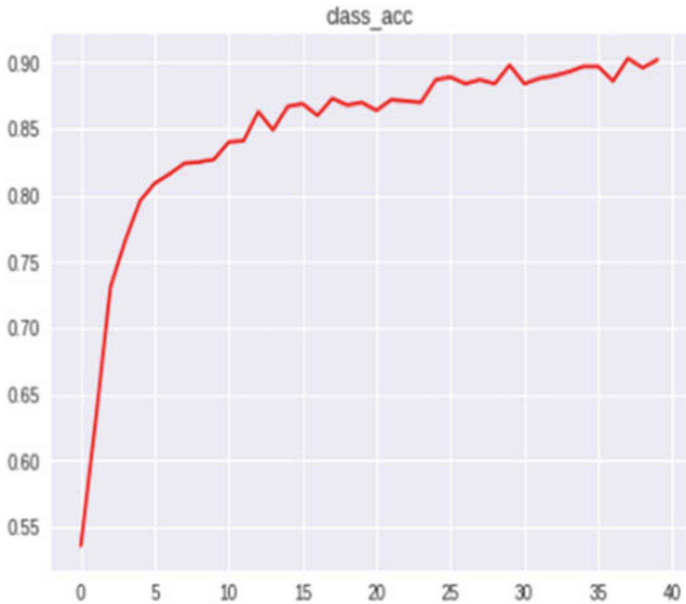


Fig. 7 Accuracy for Faster R-CNN model

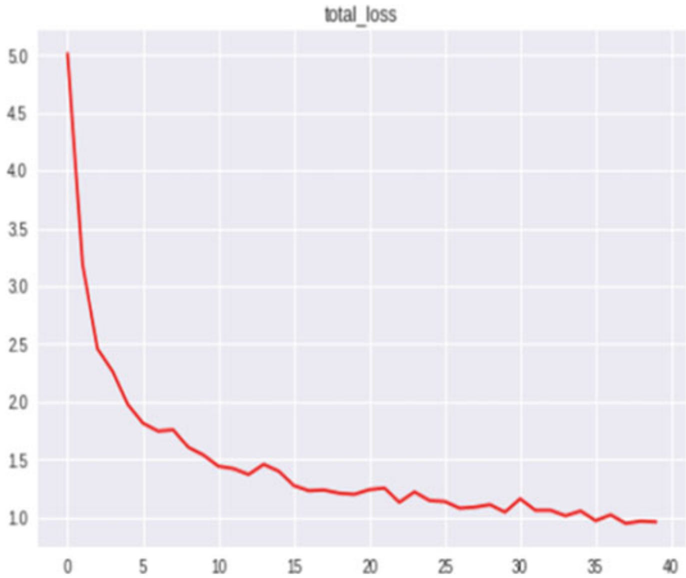


Fig. 8 Loss for Faster R-CNN model

8 Conclusion

In this paper, a conclusion is made that CNN algorithm is mainly used for image classification but R-CNN, in which R stands for Region, is for object detection. Even though a CNN model can yield a good accuracy, it only tells the class of object but it cannot locate the objects in the image, but while using R-CNN model, it also identifies the object location in the given image. In CNN, the image can have different aspect ratios and spatial locations. CNN model results an average accuracy of 85 percent. In R-CNN, for every image, it extracts 2000 region proposals for finding region of interest which takes lot of computational time. R-CNN takes so much of space and time while compiling, whereas Faster R-CNN took less time to run with an accuracy of above 90%. To overcome expensive nature of R-CNN, Faster R-CNN is used for getting better results.

9 Future Scope

These algorithms are flexible and can easily adopt to any changes in the environment and can solve a wide range of complex problems in easy way. The neural network technologies are used for development without complexity.

References

1. Kumar, S.T.: Study of retail applications with virtual and augmented reality technologies. J. Innov. Image Process. (JIIP) **3**(2), 144–156 (2021)
2. Wei, W.: Small object detection based on deep learning. In: 2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), pp. 938–943 (2020). https://doi.org/10.1109/ICPICS50287.2020.9202185
3. Pathak, A.R., Pandey, M., Rautaray, S.: Application of deep learning for object detection. Procedia Comput. Sci. **132**, 1706–1717 (2018). ISSN: 1877-0509. https://doi.org/10.1016/j.procs.2018.05.144
4. Sri, C.G., Bano, S., Deepika, T., Kola, N., Pranathi, Y.L.: Deep neural networks based error level analysis for lossless image compression based forgery detection. In: 2021 International Conference on Intelligent Technologies (CONIT), pp. 1–8 (2021). https://doi.org/10.1109/CONIT51480.2021.9498357
5. Vivekanandam, B.: Evaluation of activity monitoring algorithm based on smart approaches. J. Electron. **2**(03), 175–181 (2020)
6. Abushahma, R.I.H., Ali, M.A.M., Al-Sanjary, O.I., Tahir, N.M.: Region-based convolutional neural network as object detection in images. In: 2019 IEEE 7th Conference on Systems, Process and Control (ICSPC), pp. 264–268 (2019). https://doi.org/10.1109/ICSPC47137.2019.9068011
7. Reddy, B.K., Bano, S., Reddy, G.G., Kommineni, R., Reddy, P.Y.: Convolutional network based animal recognition using YOLO and Darknet. In: 2021 6th International Conference on Inventive Computation Technologies (ICICT), pp. 1198–1203 (2021). https://doi.org/10.1109/ICICT50816.2021.9358620
8. Gavrilescu, R., Zet, C., Foșalău, C., Skoczylas, M., Cotovanu, D.: Faster R-CNN: an approach to real-time object detection. In: 2018 International Conference and Exposition on Electrical And Power Engineering (EPE), pp. 0165–0168 (2018). https://doi.org/10.1109/ICEPE.2018.8559776
9. Tang, C., Feng, Y., Yang, X., Zheng, C., Zhou, Y.: The object detection based on deep learning. In: 2017 4th International Conference on Information Science and Control Engineering (ICISCE), pp. 723–728 (2017). https://doi.org/10.1109/ICISCE.2017.156
10. Malhotra, P., Garg, E.: Object detection techniques: a comparison. In: 2020 7th International Conference on Smart Structures and Systems (ICSSS), pp. 1–4 (2020). https://doi.org/10.1109/ICSSS49621.2020.9202254
11. Algorry, A.M., García, A.G., Wofmann, A.G.: Real-time object detection and classification of small and similar figures in image processing. In: 2017 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 516–519 (2017). https://doi.org/10.1109/CSCI.2017.87
12. Raj, J.S., Vijitha Ananthi, J.: Recurrent neural networks and nonlinear prediction in support vector machines. J. Soft Comput. Paradigm (JSCP) **1**(1), 33–40 (2019)
13. Smys, S., Chen, J.I.Z., Shakya, S.: Survey on neural network architectures with deep learning. J. Soft Comput. Paradigm (JSCP) **2**(3), 186–194 (2020)
14. Zhao, Z.-Q., Zheng, P., Xu, S.-T., Wu, X.: Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. 1–21 (2019)
15. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Region-based convolutional networks for accurate object detection and segmentation. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 1, pp. 142–158 (2016). https://doi.org/10.1109/TPAMI.2015.2437384
16. Adam, E.E.B., Sathesh, A.: Construction of accurate crack ıdentification on concrete structure using hybrid deep learning approach. J. Innov. Image Process. (JIIP) **3**(2), 85–99 (2021)
17. Guru Sai Sarma Chilukuri, N.V.S., Bano, S., Tholeti, G.S.R., Kamma, S.P., Niharika, G.L.: An analytical prediction of breast cancer using machine learning. In: Kumar, A., Senatore, S., Gunjan, V.K. (eds.) ICDSMLA 2020. Lecture Notes in Electrical. Engineering, vol. 783. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-3690-5_17

18. Zou, X.: A review of object detection techniques. In: 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA), pp. 251–254 (2019). https://doi.org/10.1109/ICSGEA.2019.00065
19. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 1–1 (2016)
20. Cao, C., et al.: An improved faster R-CNN for small object detection. IEEE Access **7**, 106838–106846 (2019). https://doi.org/10.1109/ACCESS.2019.2932731
21. Liu, B., Zhao, W., Sun, Q.: Study of object detection based on Faster R-CNN. In: 2017 Chinese Automation Congress (CAC), pp. 6233–6236 (2017). https://doi.org/10.1109/CAC.2017.8243900
22. Galvez, R.L., Bandala, A.A., Dadios, E.P., Vicerra, R.R.P., Maningo, J.M.Z.: Object detection using convolutional neural networks. In: TENCON 2018—2018 IEEE Region 10 Conference (2018)
23. Girshick, R., Donahue, J., Darrell, T., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
24. Tripathi, M.: Analysis of convolutional neural network based image classification techniques. J. Innov. Image Process. (JIIP) **3**(02), 100–117 (2021)