International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-10, Issue-2, Mar 2022
https://doi.org/10.55524/ijircst.2022.10.3.55
Article ID CST1026, Pages 268-273
www.ijircst.org

Convolutional Neural Network-based Object Detection

Dr. Ashish Oberoi

Assistant Professor, Department of Computer Science & Engineering, RIMT University, Mandi Gobindgarh, Punjab, India

Correspondence should be addressed to Dr. Ashish Oberoi; ashishoberoi@rimt.ac.in

ABSTRACT- In the midst of the efforts in an item identification, region CNNs (rCNN) stands out as the most impressive, combining discriminatory exploration, CNNs, sustenance vector machines (SVM), and bounding box regression to achieve excellent object detection performance. We propose a new method for identifying numerous items from pictures using convolution neural nets (CNNs) in this presented study. The authors of the presented study use the edge box technique to create region suggestions from edge maps for each picture in our model, and then forward pass all of the proposals through a well-accepted CaffeNet prototype. Then we extract the yield of softmax that generally is most recent layer of CNN, to determine CNNs score for every proposal. One of the greedy suppression methodology referred to as non-maximum suppression (NMS) method is then used to combine the suggestions for each class separately. Finally, we assess each class's mean average precision (mAP). On the PASCAL 2007 test dataset, our model has a mAP of 37.38 percent. In this work, we also explore ways to enhance performance based on our model.

KEYWORDS- Convolutional Neural Network, Datasets, Object discovery, Region proposal, Regression.

I. INTRODUCTION

Because of its great skill in accurately categorizing pictures, convolutional neural nets (CNNs) have remained frequently utilized in pictorial reorganization since 2011 [1,2]. In the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), authors of the paper exhibit a noteworthy advancement in picture categorization accurateness [3]. And CNNs have emerged as the preferred scheme for resolving picture or photo grouping problems. In addition to image classification, researchers have applied CNNs to a variety of other visual recognition tasks, including localization, segmentation, phrase generation from images, and object identification.

Our research focuses primarily on the problem of object detection, which has a wide range of applications in our daily lives. Object detection aims to recognize numerous items in a solitary pic, not individually to yield class sureness aimed at every item, but also to forecast the bounding boxes for each object[4]

We will provide an alternate method to object detection in this study by decreasing the complexity of the rCNN. To begin, instead of using rCNN's selective search, we utilize edge box, a recently released technique to create region suggestions. Despite the fact that the mean average precision (mAP) of edge boxes as well as the discriminating exploration are virtually identical, edge boxes are much quicker than discriminating exploration. Second, we eliminate all class-specific SVMs and utilize the output of softmax in the last layer of CNN as our score. The authors very sensibly construct out training data to well-acceptable CNN to compensate for the probable performance degradation generated by eliminating SVMs. Our model is depicted in Fig. 1 as an overview.

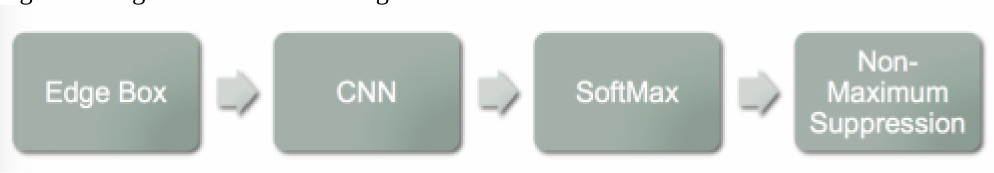


Fig. 1: The general idea of the author's item recognition method.

The authors also build an old-style model as a reference point in addition to the prior model. We utilize a sliding window to produce suggestions in this model, and histogram orientation gradient (HOG) characteristics in order to effectively define them. Thereafter, the authors rate each suggestion using our trained linear SVM.

The remainder of this work may be broken down into the following sections. In part 2, we review prior work in the field of item recognition and bring together state-of-the-art techniques. In section 3, we'll go through the technical aspects of our model in greater depth, including the overall structure, theoretical basis, and performance assessment measures. Also, we will give the findings of our experiment and discuss them briefly. Section 4 will wrap up our research and explain how we might enhance our model's performance further.

II. LITERATURE REVIEW

In the recent decade, object detection has been a popular issue in the field of visual recognition. To increase the detection performance, individuals often create features from raw pictures in the early stages. SIFT and HOG features are the most popular of these options [5][6]. The pedestrians were effectively identified from pictures using these characteristics in combination with SVMs. When the above discussed prototypes are smeared to several classes and object recognition in a solitary pic, however, the results are not as good as we had hoped. Other studies have attempted to employ collaborative SVMs in addition to concealed SVMs grounded on HOG feature descriptions instead of linear SVMs. Object detection performance, on the other hand, scarcely increases.

People are focusing on CNNs since there is a considerable gain in classification accuracy when deep CNNs are used [7][8]. Unlike classification, the detection job also needs us to supply a bounding edge box to locate the item. As a result, we can't employ CNNs for object detection without first addressing the problem of localization. tries to approach the challenge of localization as a regression problem. However, the performance has only slightly improved. The sliding window coupled with CNNs is then used by other researchers as a viable solution. This approach, however, cannot be applied in practice as a result of the time-consuming environment of slithering windows having great computation intricacy. Although the sliding window approach is not practical, it does give a concept for addressing localization by categorizing picture region suggestions.

Objectness, discriminatory exploration, group autonomous object applications, inhibited parametric min-cuts (IPMC), and edge are all methods that can produce suggestions efficiently instead of using a sliding window. The authors of the rCNN article chose selective search as the proposal generation method because of its short processing time. Instead of using selective search, we will use a newly released method called edge in this study to detect objects. Last year, other researchers used poorly supervised knowledge gaining & adaptive CNNs to discover the entities.

III. DISCUSSION

A. Line of attack:

In the present part of the paper, the authors shall shelter the particulars of their item recognition prototype, and give the findings of our experiment and discuss them briefly.

a) Proposal Generation:

The authors shall utilize the edge boxes as their scheme generating method in this study. Edge boxes' main concept is that this method produces and ranks proposals based on the image's edge map. In particular, we should create an edge map with a structured edge detector in the first phase, with each pixel including magnitude and orientation information for the edge. After that, we use a greedy method to group the edges together in such a way that the sum of the orientations of all the edges in the group is smaller than pi/2. Following that, we'll compute the affinity between two edge groups, which is crucial for scoring. Next, we'll compute the bounding box's score based on the edge groups fully within the box for a particular bounding box. Finally, we combine the bounding box using non-maximal suppression to obtain the suggestions.

When compared to the selective search option in the rCNN, the mAP of edge boxes for the VOC 2007 dataset is 21.9 percent, which usually to some extent is higher than the discriminatory exploration mAP of 21.8 percent. Edge boxes most of the time have the benefit of being significantly quicker than the bulk of proposal generating techniques in terms of runtime. The average duration for edge boxes is 1.2 minutes, whereas discriminatory exploration takes around 2 minutes. As a result, the edge boxes reduce the interval intricacy devoid of compromising enactment. As a result, the edge boxes are chosen as the scheme group procedure.

b) Training Procedure:

We'll go through how to prepare our training data and fine-tune the Caffe model in this paragraph. The author shall very prudently construct their data for the training purposes, principally for the contextual set of data, to compensate for the removal SVMs from the rCNN, as we indicated in section 1. The authors use separate training data sets to train both CNN and class specific SVMs in the rCNN model [9][10]. For CNN, the bounding box with IoU greater than 0.5 is considered positive data, while the rest is considered negative (background) data. To enhance localization precision, SVMs employ the ground-truth as positive data, IoU less than 0.3 as negative data, and all other instances are ignored.

All of the training data is taken from the raw pictures in our method. We didn't utilize IoU of 1.01 to separate positive and negative data for CNN since we didn't want to reduce localization performance by using SVMs. The following are our plans. Throughout testing period, the amount of proposals is generally significantly higher than the number of affirmative ideas for the detection problem. As a result, we require more background data for training than positive data. As a result, we acquire four times as much background data as positive data. The main point to consider is that the author shall not add more background data because this would create an imbalance in the training dataset, making it more difficult for a classifier to categorize it.

The data, particularly the background one is then divided into four folders: 1,2,3,4. The intersection over union with base fact for folder 1 is between 0.5 and 0.7, the IoU with ground truth for folder 2 is between 0.3 and 0.5, and the IoU with ground truth for folders 3 and 4 is less than 0.3. Positive data is extracted at random from the raw picture, and if the IoU with the ground truth is greater than 0.7, it is positive data with the same class label as the ground truth. The final step is to shuffle all of the data. We may achieve exact localisation without using class specific SVMs by structuring the training data in this manner. The pre-trained CNN models are then fine-tuned using the produced training data, as stated above. The CaffeNet model was chosen as our pre-trained model. This model is a copy of AlexNet, with 5 convolutional layers and a pre-trained dataset called ILSVRC2012. The productivity amount of the most recent layers is changed to 21. The CaffeNet model tweaking procedure and outcomes will be described.

c) Testing Procedure:

During testing, we first generate regional suggestions using edge boxes, and then run each & every schemes over the well-organized CNN for a forward pass. The suggestions with varied forms are shrunk to the appropriate shape before forward pass as the involvement of the CaffeNet prototype is stable at 227 x 227 pixels. The output of the softmax will then be extracted as a 21-element vector for each proposal, with each item representing confidence of associated proposal in each class.

To eliminate redundant suggestions, we use the non-maximum suppression (NMS) method. The main idea behind this algorithm is to rank ideas by confidence (also known as score) and then discard suggestions that overlap with a higher-scoring proposal. The IoU between two proposals is commonly used to establish the overlap threshold. It's worth noting that the IoU threshold has an impact on the enactment of the indicator, which essentially shall be fine-tuned to get the required results. We use the mean average accuracy to assess the detection's performance (mAP). The integral over the precision-recall curve p is equal to the mAP (r).

$$mAP = \int_0^1 p(r)dr$$

We must first compute the true positive and false positive values of our forecast in order to determine the precision-recall curve. To establish whether or not detection is successful, we utilize the IoU (Intersection of Union).

d) Baseline Model:

As a starting point, we experimented with utilizing features to generate proposals. Localization and categorization are two independent processes. For localization, HOG features, a binary-classification SVM, and Non-Maximum Suppression are employed first, and then CNN is utilized to classify the output boxes from the preceding procedure. Analogous quantities of item & background boxes are utilized in the SVM training process. To train the weights, the HOG characteristics of such boxes are sent into the SVM. SVM will detect if a given box includes objects or is just background. To acquire scores for each label, HOGs of slithering gaps of three forms and three measures are input towards proficient SVM during the testing procedure. Then, using NMS, bounding boxes that most of the time overlay others with higher scores are removed. The localisation of the items will be one of the ideas that remain after this procedure. After then, the CNN will be used to classify these areas. Based on this model, we found a mAP of 22.6 percent. Please see our CS 231A final project report for additional information on this model.

B. Experiment Results:

a) Dataset Description:

In this research, we use the VOC 2007 dataset, which is a widely used dataset for classification, detection, and segmentation. For the purpose of detection and classification, this dataset comprises 5011 pictures. All of the photos are split into two groups: training and validation. We have 2501 pictures in the training dataset & 2322 pics in the substantiation dataset.

The items in the set of data may generally be divided into 15 different categories. Every picture comprises several objects, not all of them belong to the same class. This dataset contains a total of 12608 items, including 5422 in the trained data and 5423 in the endorsement data. There will be 2.51 items per picture on average. As a result, this dataset is a good fit for the detection issue. Regular objects and challenging objects may also be used to describe the items. The challenging items are usually obscured by other objects, making them harder to see. Without additional information, such as the image's perspective, these items tagged as challenging cannot be readily identified or discovered. This isn't part of the project's scope. As a result, unlike most other efforts, we chose to disregard the problematic things in our project.

We pre-processed the training and validation data for fine-tuning CNN, as we indicated in the previous section. We created 29456 training data that comprises of almost more than 5000 positive pictures labelled by 20 classes and 23116 contextual pics, totalling 30120 training data. Additionally, the author acquire 29563 authentication data, which includes 5064 pictures and 23562 contextual pics. The examples of shrunk training data with different labels are shown in Fig. 2. The VOOC 2006 has published entire test data, which includes 3962 pictures & the truth bounding box for every item. In the test dataset, there is no object classified as "difficult."

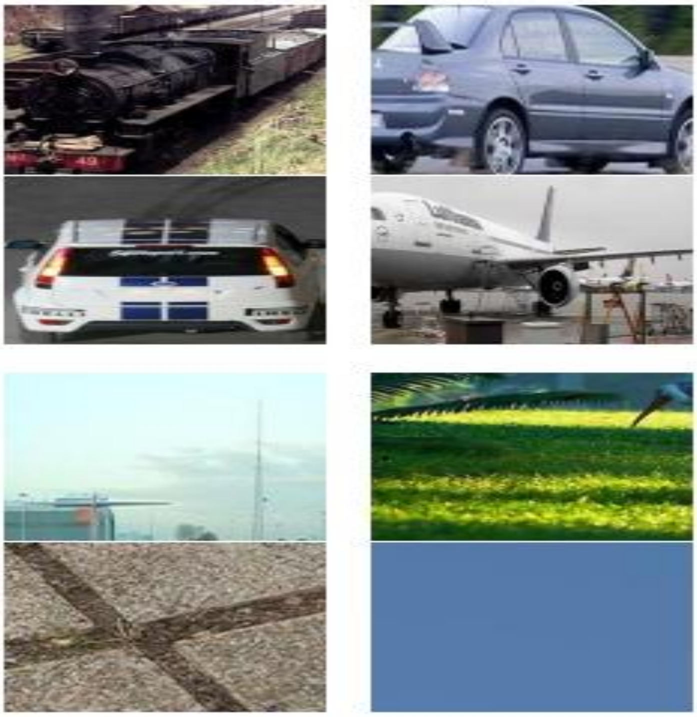
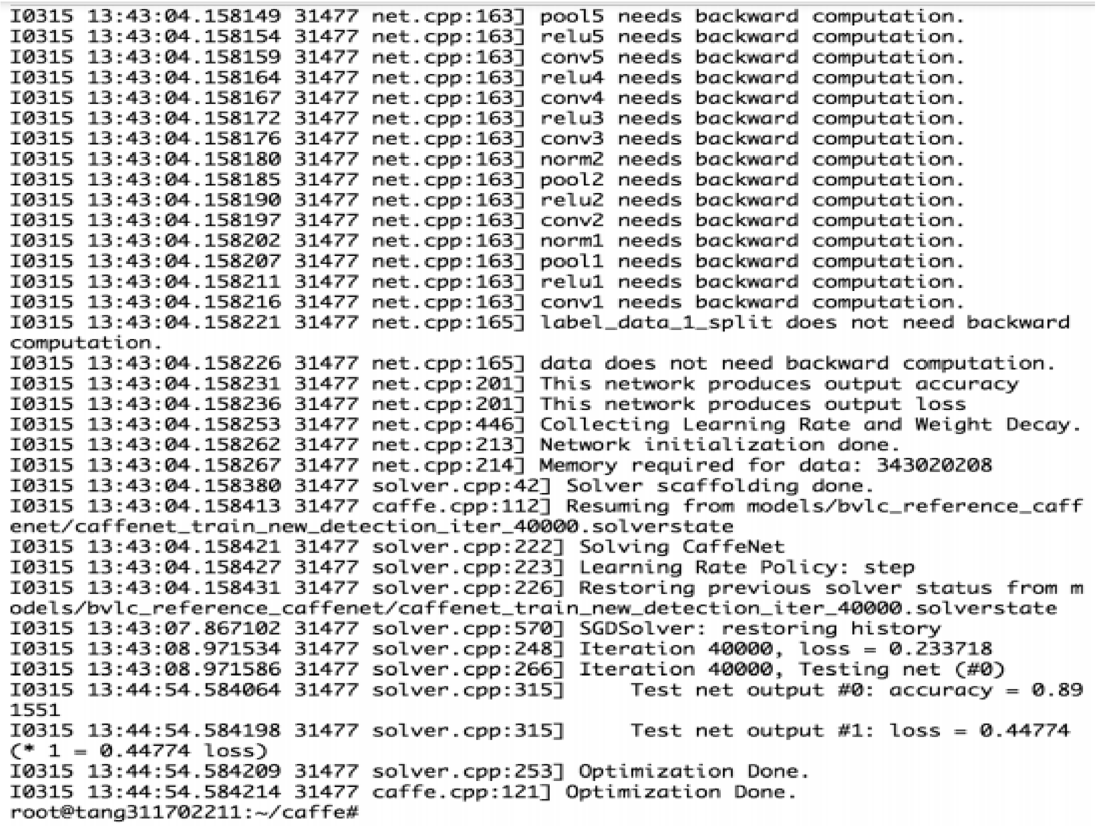


Fig. 2: The illustrations showing the pics including both positive & negative re-sized into 256×256 pixels.

b) *Fine-tuning Caffe:*

The aim of the training procedure is to well-organize the CaffeNet model that has already been trained on our dataset. Our method is to freeze all of the layers except the final (softmax layer), then train the last layer from scratch using a rather aggressive learning rate. This technique is similar to training a softmax classifier using 4096 dimension CNN features. The validation accuracy is 0.780875 after 10000 iterations, with a loss of 0.758219.

Thereafter, for the last well-organized phase, the authors release all of the layers and train them all at once. They start with a low learning rate of 5e-6 and lower it by a factor of 0.9 after every 2000 iterations. We can achieve an 89 percent validation accuracy after 30000 (40000 total) iterations. As seen in Fig. 3, the loss is 0.4474.



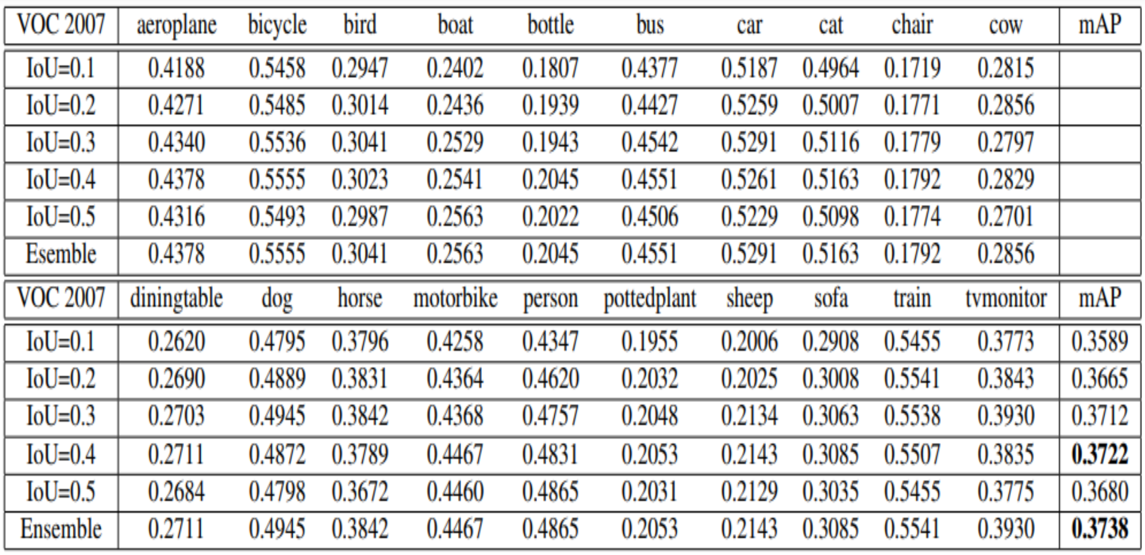
```
I0315 13:43:04.158149 31477 net.cpp:163] pool5 needs backward computation.
I0315 13:43:04.158154 31477 net.cpp:163] relu5 needs backward computation.
I0315 13:43:04.158159 31477 net.cpp:163] conv5 needs backward computation.
I0315 13:43:04.158164 31477 net.cpp:163] relu4 needs backward computation.
I0315 13:43:04.158167 31477 net.cpp:163] conv4 needs backward computation.
I0315 13:43:04.158172 31477 net.cpp:163] relu3 needs backward computation.
I0315 13:43:04.158176 31477 net.cpp:163] conv3 needs backward computation.
I0315 13:43:04.158180 31477 net.cpp:163] norm2 needs backward computation.
I0315 13:43:04.158185 31477 net.cpp:163] pool2 needs backward computation.
I0315 13:43:04.158190 31477 net.cpp:163] relu2 needs backward computation.
I0315 13:43:04.158197 31477 net.cpp:163] conv2 needs backward computation.
I0315 13:43:04.158202 31477 net.cpp:163] norm1 needs backward computation.
I0315 13:43:04.158207 31477 net.cpp:163] pool1 needs backward computation.
I0315 13:43:04.158211 31477 net.cpp:163] relu1 needs backward computation.
I0315 13:43:04.158216 31477 net.cpp:163] conv1 needs backward computation.
I0315 13:43:04.158221 31477 net.cpp:165] label_data_1_split does not need backward
computation.
I0315 13:43:04.158226 31477 net.cpp:165] data does not need backward computation.
I0315 13:43:04.158231 31477 net.cpp:201] This network produces output accuracy
I0315 13:43:04.158236 31477 net.cpp:201] This network produces output loss
I0315 13:43:04.158253 31477 net.cpp:446] Collecting Learning Rate and Weight Decay.
I0315 13:43:04.158262 31477 net.cpp:213] Network initialization done.
I0315 13:43:04.158267 31477 net.cpp:214] Memory required for data: 343020208
I0315 13:43:04.158380 31477 solver.cpp:42] Solver scaffolding done.
I0315 13:43:04.158413 31477 caffe.cpp:112] Resuming from models/bvlc_reference_caff
enet/caffenet_train_new_detection_iter_40000.solverstate
I0315 13:43:04.158421 31477 solver.cpp:222] Solving CaffeNet
I0315 13:43:04.158427 31477 solver.cpp:223] Learning Rate Policy: step
I0315 13:43:04.158431 31477 solver.cpp:226] Restoring previous solver status from m
odels/bvlc_reference_caffenet/caffenet_train_new_detection_iter_40000.solverstate
I0315 13:43:07.867102 31477 solver.cpp:570] SGDSolver: restoring history
I0315 13:43:08.971534 31477 solver.cpp:248] Iteration 40000, loss = 0.233718
I0315 13:43:08.971586 31477 solver.cpp:266] Iteration 40000, Testing net (#0)
I0315 13:44:54.584064 31477 solver.cpp:315]     Test net output #0: accuracy = 0.89
1551
I0315 13:44:54.584198 31477 solver.cpp:315]     Test net output #1: loss = 0.44774
(* 1 = 0.44774 loss)
I0315 13:44:54.584209 31477 solver.cpp:253] Optimization Done.
I0315 13:44:54.584214 31477 caffe.cpp:121] Optimization Done.
root@tang311702211:~/caffe#
```

Fig. 3: The accurateness & forfeiture for well-organized CaffeNet for VOOC 2006 item recognition job.

c) *Testing Results:*

During the testing process, edge boxes are used to create suggestions for each test picture. Each image takes about 0.3 seconds to run on average. However, the number of recommendations for each image varies from 2000 to 6000. The forward pass for each proposal takes around 54 ms per proposal and 1.8 to 5.4 minutes per image when using the terminal GPU+Caffe instance (including the time of load and save files). Based on the analysis above, we can estimate that the overall runtime to pass all of the test pictures in the worst scenario is around 445.68 hours. As a result, for this course project, we need to heuristically minimize the amount of ideas per image. The majority of the items in the VOC 2007 test dataset are big. As a result, we may exclude suggestions with small areas, which are unlikely to be acceptable candidates for object bounding boxes. We also tested if cutting down suggestions with areas less than 2000 square pixels (equivalent to 44.7444.7 pixels picture) reduces the total number of proposals created by edge boxes to about 2000 per image, implying that over half of the proposals generated by edge boxes are tiny. As a result, because the majority of the objects in the test dataset are huge, we may safely eliminate suggestions with areas lower than 2000 square pixels to speed up our computation without reducing performance too much.

Table 1: Illustrates the mAP enactment of the model



VOC 2007	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	mAP
IoU=0.1	0.4188	0.5458	0.2947	0.2402	0.1807	0.4377	0.5187	0.4964	0.1719	0.2815	
IoU=0.2	0.4271	0.5485	0.3014	0.2436	0.1939	0.4427	0.5259	0.5007	0.1771	0.2856	
IoU=0.3	0.4340	0.5536	0.3041	0.2529	0.1943	0.4542	0.5291	0.5116	0.1779	0.2797	
IoU=0.4	0.4378	0.5555	0.3023	0.2541	0.2045	0.4551	0.5261	0.5163	0.1792	0.2829	
IoU=0.5	0.4316	0.5493	0.2987	0.2563	0.2022	0.4506	0.5229	0.5098	0.1774	0.2701	
Esemble	0.4378	0.5555	0.3041	0.2563	0.2045	0.4551	0.5291	0.5163	0.1792	0.2856	

VOC 2007	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor	mAP
IoU=0.1	0.2620	0.4795	0.3796	0.4258	0.4347	0.1955	0.2006	0.2908	0.5455	0.3773	0.3589
IoU=0.2	0.2690	0.4889	0.3831	0.4364	0.4620	0.2032	0.2025	0.3008	0.5541	0.3843	0.3665
IoU=0.3	0.2703	0.4945	0.3842	0.4368	0.4757	0.2048	0.2134	0.3063	0.5538	0.3930	0.3712
IoU=0.4	0.2711	0.4872	0.3789	0.4467	0.4831	0.2053	0.2143	0.3085	0.5507	0.3835	**0.3722**
IoU=0.5	0.2684	0.4798	0.3672	0.4460	0.4865	0.2031	0.2129	0.3035	0.5455	0.3775	0.3680
Ensemble	0.2711	0.4945	0.3842	0.4467	0.4865	0.2053	0.2143	0.3085	0.5541	0.3930	**0.3738**

we send all of the suggestions to CNN, who calculates the softmax scores for each one. After that, we run the NMS for each class to eliminate any overlapping proposals. Finally, we calculate the mAP. It's worth noting that the mAP in NMS is dependent on the IoU threshold. As a result, we've adjusted the IoU from 0.1 to 0.5 and discovered that IoU = 0.4 produces the greatest mAP performance of 0.3722. The mAP may be increased further by selecting the greatest average precision for each class compared over all IoU values, bringing the mAP to 0.3738. Table 1 shows a summary of the findings.

IV. CONCLUSION

Overall, we gained hands-on experience with CNN in this project, including network debugging, transmission knowledge, and dealing with the Caffe. The author also use CNNs to address the uncovering issue, & they attempt to enhance existing models like rCNN. In this work, we present a novel CNN-based object detection model. The edge boxes method is used to create suggestions in this model, and a fine-tuned CaffeNet model is used to calculate the score for each proposal. Then we combine NMS's suggestions. On the VOC 2007 dataset, our model obtains a 0.3738 mAP. We will utilize all of the suggestions produced by the edge boxes rather than throwing the small proposals as we do in this article to enhance this model beyond the scope of this research. We'll also tweak a deeper network to enhance classification accuracy, as well as include ground truth bounding boxes into the training data to improve localization accuracy.

REFERENCES

1. Shah M, Kapdi R. Object detection using deep neural networks. In: Proceedings of the 2017 International Conference on Intelligent Computing and Control Systems, ICICCS 2017. 2017.
2. Xiao F, Deng W, Peng L, Cao C, Hu K, Gao X. Multi-scale deep neural network for salient object detection. IET Image Process. 2018;
3. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet Large Scale Visual Recognition Challenge. Int J Comput Vis. 2015;
4. Zhang X, Chen F, Huang R. A combination of RNN and CNN for attention-based relation classification. In: Procedia Computer Science. 2018.
5. Joseph S, Pradeep A. Object Tracking using HOG and SVM. Int J Eng Trends Technol. 2017;
6. Wang W, Zhu Y, Wang Z, Tu H. Intelligent robot object detection algorithm based on spatial pyramid and integrated features. Jisuanji Jicheng Zhizao Xitong/Computer Integr Manuf Syst CIMS. 2017;

7. 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. 2017;

8. Chen Y, Li W, Sakaridis C, Dai D, Van Gool L. Domain Adaptive Faster R-CNN for Object Detection in the Wild. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2018.

9. Guo MW, Zhao YZ, Xiang JP, Zhang C Bin, Chen ZH. Review of object detection methods based on SVM. Kongzhi yu Juece/Control and Decision. 2014.

10. Liu J, Huang Y, Peng J, Yao J, Wang L. Fast Object Detection at Constrained Energy. IEEE Trans Emerg Top Comput. 2018;