Object Detection using Deep Learning

Sriram S[1], Yogesh K[2], Santhosh Kumar D[3], Gayathri R[4]

[1,2,3]Undergraduate, Department of Computer Science and Engineering,
[4]Assistant Professor, Department of Computer Science and Engineering,
Sri Venkateswara College of Engineering, India.

---***---

Abstract - *It has received much research attention in recent years because of the close relationship between object detection and video analysis and image comprehension. Traditional methods of object detection are based on hand-crafted features and architectures that are flawlessly trainable. Their performance stagnates easily by constructing complex ensembles that combine multiple low-level image features with high-level context from object detectors and scene classifiers. With the rapid growth of deep learning, more efficient techniques are implemented to solve the problems inherent in conventional architectures, capable of learning semantic, high-level, and deeper features. These models act differently in the context of network design, training strategy, and optimization.*

Key Words: Deep Learning, Object Detection, Network Design.

1. INTRODUCTION

We should not get a full understanding of the picture. Concentrate not on the description of different images but Often seek to guess exactly the definitions and positions of every image contain artifacts. This function is an entity Detection, normally composed of different subtasks. Examples include face recognition, pedestrian recognition, and finding skeletons. As one of its foundation's Problems with computer vision, object recognition can have Valuable knowledge for image semblance comprehension and images. In many applications, including the classification of pictures, the study of human behaviour, Autonomous driving, and face recognition. Meanwhile, the advancement in these areas, inherited from neural networks and related learning systems, should improve neural network algorithms and also have significant impacts on object detection techniques that can be considered as learning systems. However, due to broad variations in positions, poses, occlusions, and lighting conditions, it is challenging to perform object detection correctly with an additional function of the object's position.

1.1 Informative region selection

Informative region selection. As different objects can appear at any image location and have various aspect ratios or sizes, scanning the entire image with a multi-scale sliding window is a reasonable choice. Although this exhaustive strategy can evaluate all possible locations of the objects, its drawbacks are also evident. It is computationally costly due to a large number of candidate windows and generates too many redundant windows. However, if only a set number of sliding window models are used, the regions are unsatisfactory could be generated.

1.2 Feature Extraction

Object recognition is the definition of a set of related machine views Tasks that include tasks such as digital picture recognition of objects. Classification of photographs involves tasks such as one-class estimation object in the image. Localization of artifacts refers to the location of one or more picture artifacts drawing an abundant box around their picture Extension. Target detection blends these two features and Localizes one or more objects on an image and classifies them. When a customer is active or Practitioner refers to the word "object recognition" and they mean "object detection".

Object detection in Computer Vision is a challenging and exciting task. Detection can be hard because there are all sorts of differences in orientation, Lighting, context, and occlusion, which can lead to entirely different Images of exactly the same thing. Now with the development of deep learning and Neural network, eventually, we can solve these problems without coming up with them various real-time heuristics. We had the Faster R-CNN model developed and trained on the platform for Tensorflow Deep learning. The Region-based Faster RCNN, a tool for detecting neural networks. First, we used a network of area proposals (RPN) to produce ideas for detection[1]-[6].

2. LITERATURE SURVEY

2.1 Improving object detection with deep convolutional networks via Bayesian optimization and structured prediction

According to Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee, Object detection systems based on the profoundly convolutional neural network (CNN) have recently made ground-breaking progress on many benchmarks for object detection. Although the characteristics learned from these high-capacity neural networks are egalitarian for categorization, a major source of detection error is still inaccurate localization. Built on high-capacity CNN architectures, we answer the position problem by 1) using a Bayesian optimization search algorithm which sequentially proposes candidate regions for an object bounding box, and 2) training the CNN with a formal loss that specifically penalizes the inaccuracy of the position[1].

2.2 Subcategory-aware convolutional neural networks for object proposals and detection

According to P. Druzhkov and V. Kustikova, in methods of detection of artifacts based on CNN, area proposal becomes a bottleneck when artifacts show large variance in size, occlusion, or truncation. Moreover, these methods concentrate primarily on 2D object detection and cannot estimate accurate object properties. In this paper, we suggest subcategory-aware CNNs for the detection of objects. We implement a new area proposal network using subcategory information to direct the proposal generation process, and a new detection network for joint identification and classification of subcategories. We achieve state-of-the-art efficiency on both detection and pose estimation on widely used benchmarks by using subcategories related to object pose[2].

2.3 Low-complexity approximate convolutional neural networks

Following on from P. F. Felzenszwalb, D. McAllester, R. B. Girshick, and D. Ramanan, they think of the question of generic detection and localization Objects in static pictures, from categories such as people or vehicles. This is a bit of a difficult question since objects in these categories can differ considerably Semblance. Variations occur not only from shifts in the lighting and viewpoint but also because of non-rigid deformations and instability in intraclass Shape and other visual characteristics. People wear varying clothing, for example, and take a variety of poses as the cars come in various shapes and colors[3].

2.4 Object detection via a multi-region and semantic segmentation-aware cnn model

According to S. Gidaris and N. Komodakis, we propose a method for object detection that relies on a profoundly convolutional neural network (CNN) of multi-region that also encodes semantic segmentation-aware features. The resulting CNN-based representation attempts to capture a diverse collection of discriminative appearance variables and exhibits sensitivity to localization, which is important for the precise location of objects. By implementing it on an iterative localization system that alternates between scoring a box proposal and refining its position with a deep CNN regression model, we leverage the above-mentioned properties of our recognition module. Thanks to the efficient use of our modules, we are detecting objects with very high precision in localization[6].

2.5 Face detection using deep learning: an improved faster RCNN approach

According to X. Sun, P. Wu, and S. C. Hoi, via tuned several key hyper-parameters in the Faster RCNN architecture, where they have found that, among others, the most crucial one seems to be the number of anchors in the RPN part. Traditional Faster RCNN uses nine anchors, which sometimes fails to recall small objects. For face detection tasks, however, small faces tend to be fairly common, especially in the case of unclear face detection. Therefore, instead of using the default setting, we add a size group of 64 × 64, thus increasing the number of anchors to 12 and proposed a new method for face detection using deep learning techniques. We extended the state of-the-art Faster RCNN framework for generic object detection, and proposed several effective strategies for improving the Faster RCNN algorithm for resolving face detection tasks, including feature concatenation, multi-scale training, hard negative mining, and configuration of anchor sizes for RPN[5].

2.6 Imagenet classification with deep convolutional neural networks

According to A. Krizhevsky, I. Sutskever, and G. E. Hinton, with the neural network, which lots of neurons, consists of five convolutional layers, some of which are followed by special layers, and three fully-connected layers with a final 1000-way softmax, the network has learned by computing its top-5 predictions on test images. Probing the network's knowledge is to consider the feature activations induced by an image at the last, a dimensional hidden layer.Computing similarity by using Euclidean distance between two 4096-dimensional, real-valued vectors is inefficient, but it could be made efficient by training an auto-encoder to compress these vectors to short binary codes.Results show that a large, deep convolutional neural network is achieving record breaking results on a highly challenging dataset. It can be noted that our network's performance degrades if a single convolutional layer is removed. Note that the net can identify even artifacts that are off-center, such as the mite in the top-left. Trying out the visual awareness of the network is to find the function activations that an image induces at the last, 4096-dimensional hidden layer[4].

3. PROPOSED WORK

The proposed system is designed for multiple moving tracking in real time. We use the bounding rectangular box for labelling the objects. The initial stage of the system starts with the collection of images and generation of the train and test dataset. The Cnn model comprises of the convolutional layer and pooling layer with regional propositional network for region generation. The feature maps are generated from the input image and fed into RoI layer with the regions generated. The output of the system provides labelling of the objects in the test image with the representation of the rectangular anchor boxes. The system also provides labelling of overlapping of objects based on the region mapped with the image.

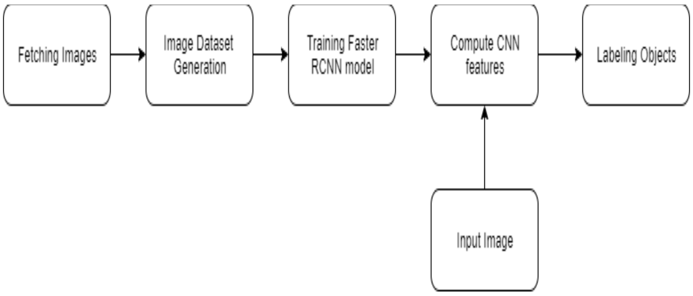


Fig.1 Architecture Diagram

3.1 Modules

The structure of the overall system can be defined using Modular design. Modularity is a common practice typically described as the extent to which a system's segments may be divided and recombined. The system consists of the following modules:

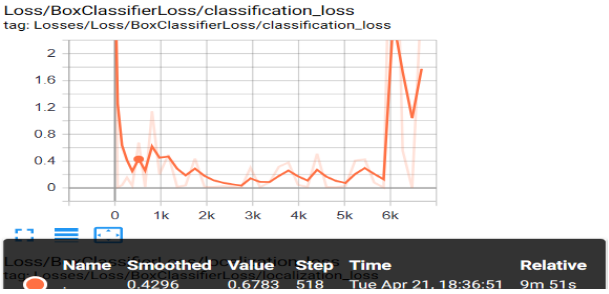
a. Generation of Training and Testing Dataset
b. Generation of Training Model
c. Testing the Model

3.1.1 Generation of the Training and Testing

The Faster R-CNN model is configured with an object dimension of 600 x 1024.The maximum detection per class set to 70. The training record is generated from the training dataset, the maximum number of classes detected around 27. The Label file for the classes is generated and the training model is built with the labeled records. The model is trained with the generation of protobuf file.

3.1.2 Generation of Training the model

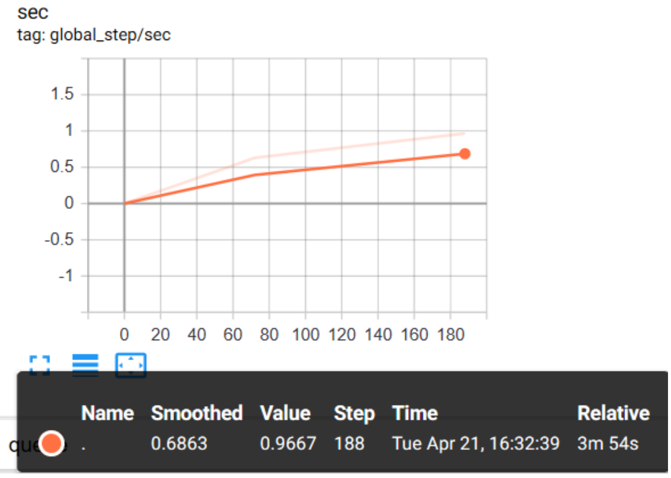
The images are collected from the COCO repository. The gathered images are labeled using the LabelImg tool. The labeled images are then converted to a CSV file using the XML to CSV tool. For training and testing, 80% of the dataset is used for training and the remaining 20% of the dataset is used for testing to check the accuracy.


Loss/BoxClassifierLoss/classification_loss
tag: Losses/Loss/BoxClassifierLoss/classification_loss
2
1.6
1.2
0.8
0.4
0
0
1k
2k
3k
4k
5k
6k

	Name	Smoothed	Value	Step	Time	Relative
●	.	0.4296	0.6783	518	Tue Apr 21, 18:36:51	9m 51s

Chart 1. Classification Loss

The loss occurred during classification of the image category among the available image data. Ideally the loss should be less for a good trained cnn model.



	Name	Smoothed	Value	Step	Time	Relative
●	.	0.6863	0.9667	188	Tue Apr 21, 16:32:39	3m 54s

Chart 2. Training Iteration

global_step/sec - global_step represents the specific iteration while training the object detection model with the number of batches processed per second.

3.1.3 Generation of Testing the model

The Trained model is loaded as the protobuf file as '.pb' and the labels of the trained images are loaded for providing the labels of the test image. The test image is converted as feature maps for detecting the objects in the image. The feature maps are then combined with a regional network for generating equal dimensional feature maps. The objects that are present in the image are labelled with the anchor boxes for live object tracking.

3.2 Model Implementation

3.2.1 Faster regional convolutional Neural network

Faster R-CNN (frcnn for short) makes further progress than Fast RCNN. The selective research process is to restore by Region Proposal Network (RPN).R-CNN is the first step for Faster R-CNN. It uses a particular search to 20 finds out the regions of interests and passes them to a Convolutional Neural Network. The procedure is related to the R-CNN algorithm. Instead of serving the region proposals to the CNN, we supply the input image to the CNN to generate a convolutional feature map. From the convolutional feature map, we recognize the field of plans and warp them within squares, and by using an RoI pooling layer, we reshape them toward a firm size so that it can be served into an utterly connected layer. From the RoI feature vector, we utilize a softmax layer to prognosticate the class of the stated field and the offset values for the bounding box. The reason "Fast R-CNN" is quicker than R-CNN is because you don't have to serve 2000 area plans to the convolutional neural network each time. Alternatively, the convolution process is performed solely once per image, and a feature map is produced from it.

3.2.2 Softmax

The softmax classifier provides "probabilities" for each class. Unlike the SVM, which measures uncalibrated and challenging to elucidate rates for all classes, the Softmax classifier permits us to estimate "chances" for all tags. For illustration, given an image, the SVM classifier might provide you rates [12.5, 0.8, -23.0] for the types "rat, "cat," and "ship." The softmax classifier can preferably measure the three tags' possibilities as [0.9, 0.09, 0.01], which enables you to evaluate its reliance in each class. In both cases, we estimate the equal score vector f (e.g., by matrix multiplication in this section). The variation is in the analysis of the scores in f: The SVM describes these as class scores, and its dropping function supports the right class (class 2, in blue) to have a score higher by a margin than the 22 other class scores. The Softmax classifier alternatively explains the scores as (unnormalized) log-likelihoods for every class. It then encourages the (normalized) log probability of the correct class to be high (equivalently the negative of it to below). The final loss for this case is 1.58 for the SVM and 1.04 for the Softmax classifier, but note that these numbers are not comparable; they are only meaningful concerning loss computed within the same classifier and with the same data.

3.3 Data Evaluation Module

3.3.1 Train-Test Split

The dataset is split into 2 parts before modelling. They are Training Dataset and Testing Dataset. The model is trained using the Training Dataset and in order to identify that the model is getting trained appropriately, the model is again tested using the Testing Dataset.

Here, we have split the data in the ratio 70:30, i.e 70% of the dataset is given as a Training dataset and 30% is given as testing dataset. This split is done by importing the library from scikit-learn, sklearn.model_selection.train_test_split.

3.3.2 Metrics employed for analysis

In this research, the metrics used for assessing the model are Confusion Matrix, Accuracy, and Classification Reports from sklearn.metrics.

A confusion matrix determines the number of true and false prognostications created by the classification model differentiated to the real outcomes (target value) in the data. The matrix is NxN, where N is the number of target values (classes). The performance of such models is commonly evaluated using the data in the matrix. It is represented in eqn.(1).

$$Confusion\ matrix\ in\ 2x2\ format = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

(1)

Where TN is True Negative, FP is False Positive, FN is False Negative, and TP is True Positive.

Classification Accuracy: It is the proportion of the total number of prognostications, which was correct. It is represented in eqn.(2).

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \qquad (2)$$

Positive Predictive Value or Precision: The proportion of affirmative cases which is correctly identified.

F1 Score: F1 score unites recall and precision compared to a specific positive class -The F1 score can be expounded as a weighted mean of the recall and precision, where an F1 score stands its best value at 1 and worst at 0. It is represented in eqn.(3).

$$F1 - Score = \left(2 * \frac{(precision * recall)}{(precision + recall)}\right)$$

(3)

4. CONCLUSION

In this paper, an accurate and efficient object detection system has been developed which achieves comparable metrics with the utilization of the Faster CNN. This project uses recent techniques within the field of computer vision and deep learning. A custom dataset was created using labelImg and also the evaluation was consistent. This could be employed in real-time applications that require object detection for pre-processing in their pipeline. A crucial scope would be to coach the system on a video sequence for usage in tracking applications. The addition of a temporary constant interface would facilitate smooth detection and more optimal than per-frame detection.

5. FUTUREWORK

Discovering of object is a very time exhausting process to draw large quantities of bounding boxes manually. To release this burden, semantic prior unsupervised object discovery multiple instance learning and deep neural network prediction can be integrated to make the best use of image-level supervision to cast object category tags to similar object regions and improve object limits. Furthermore, this model is loaded into android, and objects are detected in mobile camera, and those object names are spelled out by the voice assistant API in the app which is helpful for blind people to navigate oneself.

REFERENCES

[1] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee, “Improving object detection with deep convolutional networks via bayesian optimization and structured prediction,” in CVPR, 2019

[2] P. Druzhkov and V. Kustikova, “Subcategory-aware convolutional neural networks for object proposals and detection,” in WACV, 2018.

[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan,

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in NIPS, 2012.

[5] X. Sun, P. Wu, and S. C. Hoi, “Face detection using deep learning: An improved faster rcnn approach,” arXiv:1701.08289, 2017.

[6] S. Gidaris and N. Komodakis, “Object detection via a multi-region and semantic segmentation-aware cnn model,” in CVPR, 2015.