Real Time Object Detection using Deep Learning

Research · July 2024

DOI: 10.13140/RG.2.2.23126.46408

1 author:

Mohammad Salman Khan
Konya Technical University
8 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Real Time Object Detection using Deep Learning: A Camera BasedTechnique

Mohammad Salman Khan[1]

Konya Technical University

Abstract:

Object detection has emerged as a critical application of deep learning, distinguished by its ability to learn and represent features, unlike traditional object detection methods. Region proposal algorithms are fundamental to object detection networks, helping to hypothesize object regions. Algorithms like SPPnet and Fast R-CNN have significantly reduced the running time of these detection networks. This paper proposes a method for object detection using a webcam, combining a Region Proposal Network (RPN) and Fast R-CNN. The RPN is trained end-to-end to generate high-quality region proposals, which are then utilized by Fast R-CNN for detection. Together, these modules form an object detection system known as Faster R-CNN. Deep models can function both as classifiers and detection devices without the need for hand-engineered technologies. Consequently, deep learning technology holds great promise for object detection. Object detection is a major focus in computer vision technologies, with applications in pedestrian detection for driverless cars, video surveillance, robotics, and counting techniques. The Faster R-CNN models InceptionV2 and ResNet50 have been used for real-time object detection through a webcam.

Key Words: Object Detection, Faster R-CNN, Deep Learning, Real Time, Webcam.

1 Introduction

Object detection is a key research area and focus within computer vision technologies (Erhan et al., 2014), with significant real-time applications in driverless cars, video surveillance, robotics, pedestrian detection (Borji et al., 2015; Tian et al., 2015), and counting and monitoring areas using remote sensing, which is a burgeoning field of application. Deep learning has revolutionized object detection methods, transforming traditional approaches to object identification and detection. Recent advances in object detection using deep learning techniques are largely attributed to region proposal methods (Uijlings et al., 2013) and region-based convolutional neural networks (R-CNNs) (Girshick et al., 2014). Initially, R-CNNs were computationally expensive (Girshick et al., 2014), but their cost has been greatly reduced by sharing convolutions across proposals (He et al., 2014; Girshick, 2015). Fast R-CNNs use very deep networks to achieve near real-time rates (Simonyan and Zisserman, 2015), excluding the time spent on region proposals. Deep neural networks have a strong feature representation capacity (Ouyang et al., 2015) in image processing, making them highly effective as feature extraction modules in object detection. It has been observed that Fast R-CNNs use GPUs while region proposal methods are applied to the CPU.

The aim of deep learning is to emulate brain neural activity functions to recognize specific images within data. Simulated neural networks combine low-level features to produce high-level representations, capturing the distributed characteristics of the data for analysis and learning (Krizhevsky et al., 2012). In this paper, we employ two models of Faster R-CNN: Inception V2 and ResNet50, which are deep learning models for object recognition from real-time data viewed through a webcam. Due to the absence of available datasets, we created our own training and testing image datasets to train the network for object recognition via webcam. Images are used solely for training and testing, as live streaming through a webcam, i.e., video, is essentially a sequence of images.

2 Related Work

Object detection involves identifying objects within a specific scene using certain methods. Before the advent of deep learning technologies, object detection relied on mathematical models (Tang et al., 2017). Some common classical methods in object detection include the Hough transform (Merlin and Farber, 1975), frame-difference method (Singla, 2014), background subtraction method (Lee, 2005), optical flow method (Horn and Schunck, 1981; Barron et al., 1992), sliding window model (Viola and Jones, 2001), and deformable part model (Felzenszwalb et al., 2010; Felzenszwalb et al., 2008).

Traditional machine learning algorithms for image analysis tasks focused on hand-crafted feature extraction, color segmentation, and normalization, often supported by classification algorithms like regression and support vector machines. However, these methods struggled with processing high-dimensional image feature sets, leading to the emergence of neural networks, which offered a viable solution for automated extraction in high-dimensional image sets. Today, neural networks are widely used, with numerous experiments utilizing multi-layered networks (Bezak, 2016).

Classical methods and machine learning applications can be categorized into two groups. The first category, including background subtraction, frame-difference, Hough transform, and optical flow methods, uses specific data characteristics to create mathematical models for object detection. The second category, which includes the deformable part model and the sliding window model, integrates classifier algorithms with supervised features for object detection results (Tang et al., 2017).

In deep learning, region selection is based on specific methods, feature extraction is achieved through convolutional neural networks (CNNs), and classification can be performed by support vector machines (SVM) (Tang et al., 2017). Recently, region proposal-based models have become prominent in deep learning for object detection.

Deep learning object detection models based on region proposals consist of two main parts: extracting regional candidates and building a deep neural network. This approach has been implemented in various ways.

1) R-CNN

R-CNN, introduced by Girshick in 2014 (Girshick et al., 2014), is a region proposal-based convolutional neural network that pioneered the concept of region proposals. R-CNN's

implementation relies on the region segmentation method of selective search (Uijlings et al., 2013) to extract region proposals from an image, identifying potential objects. These proposals are then fed into a CNN to extract feature vectors. After this step, an SVM classifier is used to classify these feature vectors, yielding classification results for each region proposal. The model's outputs are refined using non-maximal suppression (NMS), resulting in accurate object classifications and bounding boxes for object detection. The architecture is as given in Fig. 1.

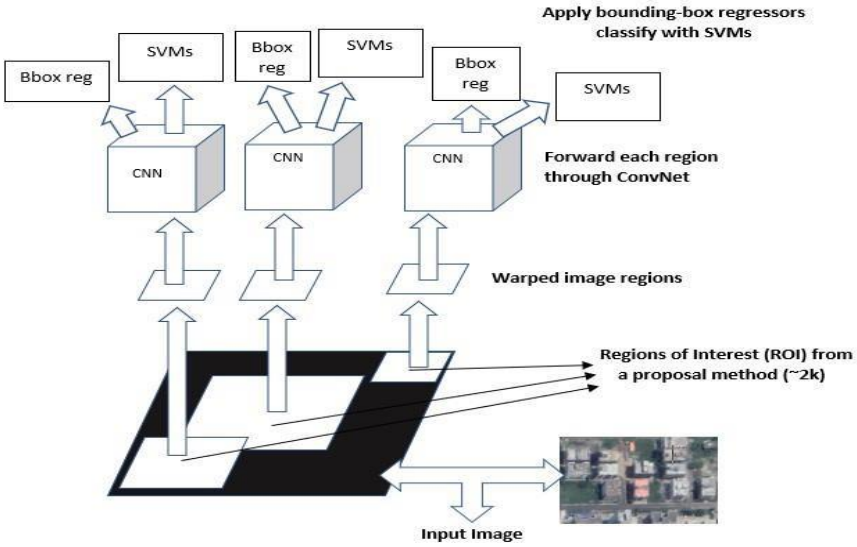

Figure 1 R-CNN Architecture

2) SPP-net

SPP-net, introduced by He et al. in 2014, is a deep artificial neural network that uses spatial pyramid pooling to eliminate the need for the warping operation on the input image used in R-CNN. This allows inputs of various sizes to connect to the fully connected layer, producing feature vectors of the same dimension after passing through the convolution layer. SPP-net addresses issues of image incompleteness and object deformation that could occur in R-CNN. However, it has a significant drawback due to its poor real-time computation methodology, similar to R-CNN.

3) Fast R-CNN

Fast R-CNN, introduced by Girshick in 2015, is an enhanced version of R-CNN that addresses the issue of repeatedly calculating the 2000 region proposals passing through the CNN. Unlike R-CNN, Fast R-CNN improves by extracting region proposals from input images using the selective search algorithm, mapping them to the feature layer of the CNN, and then performing pooling on these mapped proposals to create the ROI pooling feature layer. This ROI pooling helps in extracting feature vectors of a consistent size, which is crucial for successful connection with the CNN. The performance of ROI pooling is similar to SPP-net's spatial pyramid pooling. The way that Fast R-CNN operates is displayed in Fig. 2

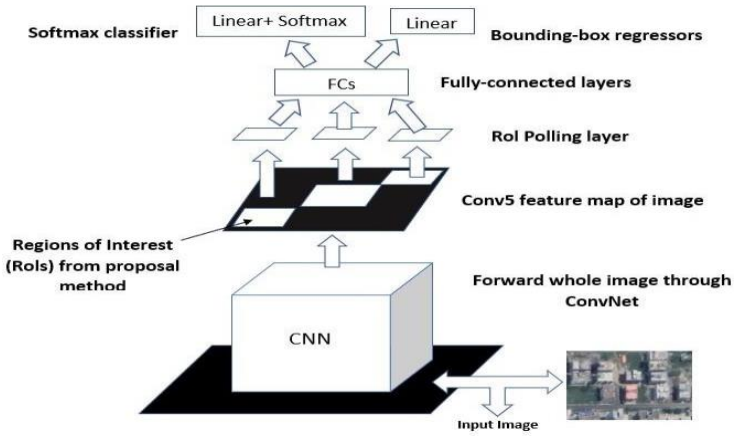

Figure 2 Architecture of Fast R-CNN

Mapping region proposals from input images to the Fast R-CNN feature layer distributes the computation of the convolution, thereby reducing the overall calculation. Additionally, to minimize parameters related to the fully connected layer, Fast R-CNN employs truncated SVD, replacing a single fully connected layer with two smaller fully connected layers, further reducing network computation. It has been noted that during training, Fast R-CNN converges 8.8 times faster than R-CNN and 2.58 times faster than SPP-net. During testing, it is 146 times faster than R-CNN without truncated SVD, 213 times faster than R-CNN with truncated SVD, and 7 times faster than SPP-net without truncated SVD, and 10 times faster with truncated SVD (Tang et al., 2017).

3 Architecture

In this study, object detection through a webcam is implemented using Faster R-CNN (Ren et al., 2015), an improved version of Fast R-CNN. To address the high computational costs and poor real-time performance of the selective search method used in Fast R-CNN and R-CNN, Faster R-CNN employs region proposal networks (RPN). This innovation helps to overcome these obstacles. Faster R-CNN utilizes an end-to-end framework, enabling simpler and faster model training compared to R-CNN and Fast R-CNN.
Faster R-CNN comprises two modules: first, a fully convolutional network (CNN) for proposing regions, and second, Fast R-CNN (Girshick, 2015) as the detector that uses these proposed regions. Together, these modules form a single, unified network for object detection. Figure 3 illustrates the architecture of Faster R-CNN.

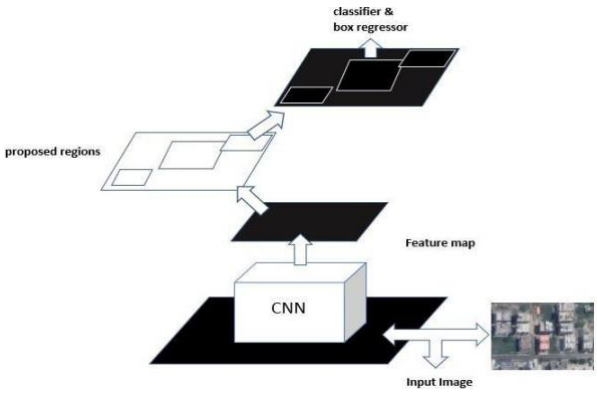

Figure 3: Architecture of Faster R-CNN

The primary function of RPN is to employ an attention mechanism (Chorowski et al., 2015) within neural networks to guide the Fast R-CNN module on where to focus its attention. The operational method of Faster R-CNN is depicted in Figure 3.

When implemented in Faster R-CNN, RPN reduces the number of region proposals from 2000, as used in selective search, to just 300. This significant reduction drastically cuts down the computational time of the model and accelerates the entire network. Experiments have shown that Faster R-CNN operates at a speed of 5 frames per second (fps), which is 10 times faster than Fast R-CNN, while also improving accuracy.

4 Result

The experiments were conducted using an NVIDIA platform and TensorFlow API on Windows 7. Acceleration of the learning process utilized a NVIDIA GeForce GTX 660 GPU with 2GB of RAM, while a NVIDIA Quadro FX 4800 CPU with 8GB RAM was also employed. Various datasets such as MNIST, MSCOCO, CIFAR, and PASCAL VOC were utilized for initial testing before applying models to proprietary datasets. Additionally, several standard models under the CC license, including LeNet, AlexNet, GoogleNet, ResNet, Inception, MobileNet, and NAS, were publicly available. These models vary significantly in architecture and storage requirements. With 2GB of GPU RAM available, identifying models that efficiently utilize memory has been challenging. Through trial and error, we have identified two models that achieve optimal performance within these memory constraints: Inception V2 and ResNet50.

A. Inception V2 model

The initial training involved the Inception V2 model with 3000 training images and 600 validation images (20%), each with dimensions of 500×500 pixels. These images were in RGB color format and encoded as JPG, which is a lossy format with reduced detail retention.

The outputs from image detection via webcam are illustrated in Figures 4, 5, 6, 7, and 8. Validation achieved moderate accuracy of 80%, improving with additional learning epochs. During training, there was a 10% loss, reducing to 5% during validation. The model achieved a prediction accuracy of 88.6%, with an unsuccessful prediction rate of 12.4%. Modern deep learning models competing in Kaggle often achieve over 90% accuracy in object detection tasks using simple RGB imagery.

Hence, there was a need to enhance our neural network model through modifications, model changes, parameter tuning, and experimenting with various models. This effort led to successful results with ResNet50 after increasing the number of training images and refining the approach.

Fig.4

Fig.5

Fig.6

Fig.7

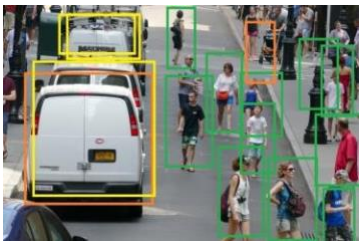

Fig.8

B. ResNet50 model

The final model we trained is ResNet50, utilizing 5000 training images and 1000 validation images (20%), each with dimensions of 500×500 pixels. These images were in RGB color format and encoded as JPG, a format that sacrifices some detail due to its lossy nature.

The outputs from image detection via webcam are illustrated in Figures 4, 5, 6, 7, and 8. Validation achieved high accuracy of 95%, improving with additional learning epochs. During training, there was a 5% loss, reducing to 3% during validation. The model achieved a prediction accuracy of 95%, with an unsuccessful prediction rate of 5%.

5 Discussion

We introduce a deep learning model designed for automated object recognition and detection in live webcam streams, achieving real-time performance. The system leverages advanced deep learning models, particularly Faster R-CNN, along with sophisticated image classification techniques. Traditionally, object detection in photos and videos relied on basic image analysis and classical machine learning methods, which were limited by their use of a predefined set of image features. In contrast, deep learning methods excel at automatically extracting features from images, leading to superior performance.

In recent years, Faster R-CNN, which integrates the Region Proposal Network (RPN), has notably enhanced computational efficiency and accuracy in object detection tasks. However, challenges such as image blurring, varying lighting conditions, and the constraints of training datasets remain areas for potential improvement. The rapid training convergence of deep learning models is critical for achieving real-time object detection capabilities, a capability effectively demonstrated by Faster R-CNN.

6 Conclusion

Comparing the two models, it has been concluded that the ResNet50 model outperforms the Inception V2 model. The proposed model is capable of real-time object detection in live

streaming videos through a webcam. It is scalable and can learn from extensive training datasets. A substantial amount of training images from various angles and of different objects is necessary for effective testing and validation of the model. The advantage of ResNet50 is its real-time implementation capability and its potential for deployment across various platforms, including mobile smart devices. However, a limitation is the restricted memory availability (2GB on GPU), which could hinder its use on memory-constrained platforms such as small smart devices and drones.

References

[1] Barron, J. L., Fleet, D. J., Beauchemin, S. S. (1992). "Performance of optical flow techniques." In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Champaign, IL, USA, 15-18 June, pp. 236-242.

[2] Bezak, P. (2016). "Building Recognition System Based on Deep Learning." In Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR), Lodz, Poland, 19-21 Sept. ISBN: 978-1-46739187-0.

[3] Borji, A., Cheng, M. M., Jiang, H. (2015). "Salient object detection: A benchmark." IEEE Transactions on Image Processing, 24(12), pp. 5706-5722.

[4] Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y. (2015). "Attention-based models for speech recognition." In Conference on Advances in Neural Information Processing Systems (NIPS), Palais des Congrès de Montréal, Montréal, CANADA, 7-12 Dec.

[5] Erhan, D., Szegedy, C., Toshev, A. (2014). "Scalable object detection using deep neural networks." In IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA, 23-28 June, pp. 2155-2162.

[6] Felzenszwalb, P. F., Girshick, R. B., McAllester, D. (2010). "Object detection with discriminatively trained part-based models." IEEE Transactions on Pattern Analysis and Machine Intelligence, 32, pp. 1627-1645.

[7] Felzenszwalb, P., McAllester, D., Ramanan, D. (2008). "A discriminatively trained, multiscale, deformable part model." In IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23-28 June, pp. 1-8.

[8] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation." In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Washington, DC, USA, 23-28 June, pp. 580-587.

[9] Girshick, R. (2015). "Fast R-CNN." In IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7-13 Dec. DOI: 10.1109/ICCV.2015.169.