International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 06 | June-2018 www.irjet.net p-ISSN: 2395-0072

Multiple Object Detection using Deep Neural Networks

Wangkheimayum Madal[1], Dr. Lakhmi Prasad Saikia [2]

[1]M.Tech, Computer Sc & Engg, Assam downtown University,India
[2] Professor, Computer Sc& Engg, Assam downtown University, India

---***---

Abstract - *In this paper, we present a system for the detection of multiple objects in an image. The system takes scenes as input, and outputs a set of objects present in the scene. The system first generates and analyses region proposals using deep convolutional neural networks. Next, a postprocessing algorithm exploits temporal information between consecutive frames to enhance the overall confidence of a detected object. The system can be extended to be used in other domains as well.*

***Key Words*: *Object Detection, Videos, Convolutional Neural Network, RCNN (Region based Convolutional Neural Network), SVM (Support Vector Machine), RPN (*Region Proposal Network)**

1. INTRODUCTION

In this paper, we focus on the development of a multiple object detection system in, that can provide the necessary content to enhance the quality and relevance of video. This system takes scenes of video as input and processes these frames one by one. Deep convolutional neural networks are used in the processing stage of the system. These neural networks are used for two purposes – to generate regions of a frame where it is likely to detect an object, and to detect the objects in those regions. In order to exploit the temporal features which are inherently present in videos, we introduce a post-processing algorithm which analyzes and combines the results of consecutive frames. When a particular frame of the video is, the detected objects are returned as output. We related to Faster R-CNN architecture is optimised by mapping each region of interest onto a large feature map extracted by the convolutional neural network. The features are calculated once over the entire image and the corresponding portion of the feature map is extracted for each region. Further improvements are made to the architecture. Now the region proposal algorithm (selective Search) is replaced by a convolutional neural network. This means that the system uses convolutional neural networks to perform generation as well as processing of regions of interest. The features are calculated once over the whole image and for each region, the corresponding portion of the feature map is extracted. To improve the performance of the object detection system, the SVMs are replaced by a combination of a Region of Interest layer and a softmax output layer.

The paper is structured as follows. In section 2, a brief overview of the related work in the domain of object recognition in videos is described. In section 3, we present the architecture used to build our proposed system and we also discuss the algorithm to detect multiple objects in videos. Next, we analyze the result of our algorithm in section 4, along with the conclusion in section 5.

2. RELATED WORK

The literature related to object detection can be split into two broad domains. In the first domain, we will discuss the primary research papers that focus on detecting multiple objects in images. Next, we will discuss a selection of the research papers related to the task of video classification.

To detect multiple objects in a single image, the most relevant techniques are discussed in [5], [6] and [7]. In [5], the R-CNN architecture has been discussed. Deep convolutional networks are useful tools for the purpose of feature extraction. The first step is the generation of regions of interest, i.e. sub-regions of an image which are likely to contain objects. This is achieved by using a region proposal algorithm, like Selective Search [8]. Then, the convolutional neural network extracts the features for each of these regions of interest. The various features which are extracted are presented in the form of a feature map. Lastly, SVMs (Support Vector Machines) are used to classify the regions of interest as proper objects or just background noise.

In the paper related to Fast R-CNN, the R-CNN architecture is further optimised by mapping each region of interest onto a large feature map extracted by the convolutional neural network. The features are calculated once over the whole image and for each region, the corresponding portion of the feature map is extracted. In order to improve the performance of the system, a combination of two layers, i.e. Region of Interest layer and softmax output layer is used instead of SVMs. The Region of Interest layer uses the features of the regions of interest as input and scales these features to the right input dimensions compatible to the softmax output layer. Then, the softmax output layer classifies the regions of interest on the basis of the input from the Region of Interest layer.

In the second research domain, the primary research is presented in [9] and [10]. Both papers propose three different techniques, i.e. early fusion, late fusion and slow fusion in order to combine the features obtained from the first convolutional layers of the network. This is used for processing information of a number of consecutive frames at once:. In [10], along with the techniques used to merge convolutional features of consecutive frames, a recurrent neural network is introduced. This recurrent neural network encodes the variations of the content of consecutive video frames. This allows a more effective and efficient classification of longer videos.

3. ARCHITECTURE

3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are different from feed-forward neural networks owing to their convolutional and pooling layers. In convolutional (conv) layers, independent filters (usually square) are convolved with all the previous layer's channels. For example, in the very first conv layer, filters are convolved with the red, green, and blue channels of the input image. The size of the filter is denoted as (h,w,c), where h and w represent the height and width of the filter respectively, and c represents the number of input channels (e.g. 3 for the very first conv layer). After the linear filtering process, the responses across channels undergo summation. This process is performed repeatedly for a number of independent filters. The activations of each filter act as the channels for the next layer. Unlike pre-made filters (e.g. Sobel operator), the filter weights are learned and updated during training. The filter parameters are learned using Stochastic gradient descent with backpropagation. Multiple conv layers can be stacked to enable learning of higher order features in the deeper layers [17]. These conv features perform better than the hand-crafted features, which has led to the widespread use of CNNs in computer vision [17].

Pooling is a technique often used in CNNs. Pooling refers to an operator (e.g. max or average) applied to spatial neighborhoods in order to reduce dimensionality. Each channel is pooled independent of the others. Moreover, the pooling is applied in strides, and this corresponds to the magnitude of reduction in dimensionality. For instance, a pooling layer applied with a stride of 2 by 2 will reduce the size of the width and height of the layer input by 2. This reduction of dimensionality is particularly useful because input images are of high dimensions, and hence, applying a large number of convolutions to an image of size 224 by 224 pixels will lead to huge computational costs. Moreover, max pooling selects the largest response in an neighborhood, thereby leading to a degree of translation invariance. The location of an object in the local neighborhood an object is irrelevant, and this leads to translational invariance.

3.2 Fast R-CNN

Fast R-CNN [18] (different from Faster R-CNN, which is described below) improves upon the original R-CNN [6]. It is designed to increase the speed of the detection network. First, this network generates bounding boxes for the image using bounding box proposal methods such as Selective Search [19] or Edge Boxes [20]. The entire image is then passed through the CNN, up through the last conv layer (e.g. through conv5 in VGGNet). Next, Region of Interest (RoI) pooling is applied to the final layer conv features for each bounding box, and this gives a fixed size vector. Then, this RoI vector is given as input to a classification network and regressor network. The classification layer is a fully connected layer followed by a softmax over all object classes. The regressor network is a 2-layer network that fine-tunes the bounding box coordinates and dimensions. The fine-tuning provides significant improvements of the bounding boxes. Finally, non-maxima suppression is applied over all boxes for each class independently.

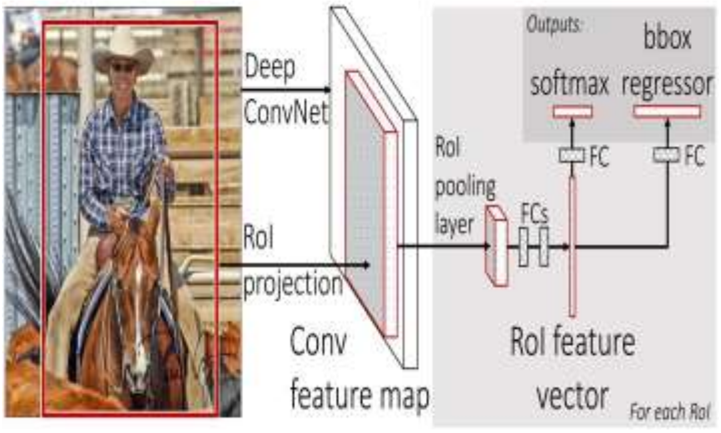

Figure 1: Fast R-CNN architecture (credit [18]).

In RoI pooling, the boundaries of the bounding box are projected onto the conv layer. It is worth noting that this is not a one-to-one mapping. The reason is that there is a significant reduction in the height and width of the final conv layer as compared to the original input dimensions because of the striding in conv and pooling layers. To account for this, let the striding factor S be the product of strides sizes in all the conv and pooling layers (for simplicity, this assumes the horizontal and vertical strides are equal, but it is trivial to extend to non-equal strides). Finally, split the conv bounding box region into H by W equal sized windows and max pool inside each block. This results in an H x W fixed size vector for each channel. H and W are constant hyperparameters that depend on the network architecture requires $H = W = 7$ to serve as input to the fully-connected layer).

The network is trained with a multi-task loss, $L(p,c,t^u,b) = L_{cls}(p,c) + \lambda[c =6 \text{ background}]L_{reg}(t^u,b)$ p is the probability distribution over $K + 1$ object categories predicted by the classification layer. The distribution includes the background category (i.e. no object in box). t^u is the bounding box predicted by the regression network. c is the ground truth class and b is the ground truth bounding box. [c 6= background] is an indicator function which takes a value of 1 iff the proposed bounding box contains an object; otherwise, it is a background box and the indicator function takes a value of 0, so regression loss is not used for training on background boxes.

3.3 Region Proposal Network

In Region Proposal Network (RPN), a sliding window approach is used to generate k bounding boxes for each position in the last layer conv feature map (e.g. conv5 for VGGnet). First, an n by n filter is convolved with last layer conv feature map. Next, the result is projected to a lower dimensional space by convolving with a 1 by 1 filter (which just linearly combines the channels for each position independently). This results in a fixed-size vector for each position. The vector is separately passed into a box-regression layer and a box-classification layer. In the box-regression layer, *k* bounding boxes are generated relative to the current position in the conv feature map (the current anchor point). For example, if the vector for position (x_a, y_a) is passed into the box-regression layer, then the layer will have $4k$ outputs. Each set of of 4 outputs parameterizes a bounding box relative to (x_a, y_a). The four parameters are t_x, t_y, t_w, t_h, where,

$t_x = (x - x_a)/w_a \quad t_y = (y - y_a)/h_a$

$t_w = \log(w/w_a) \quad t_h = \log(h/h_a)$

Each of the *k* generated bounding boxes will have a different fixed w_a and h_a. This gives rise to boxes differing in areas and aspect ratios (because of different absolute and relative magnitudes of w_a and h_a).

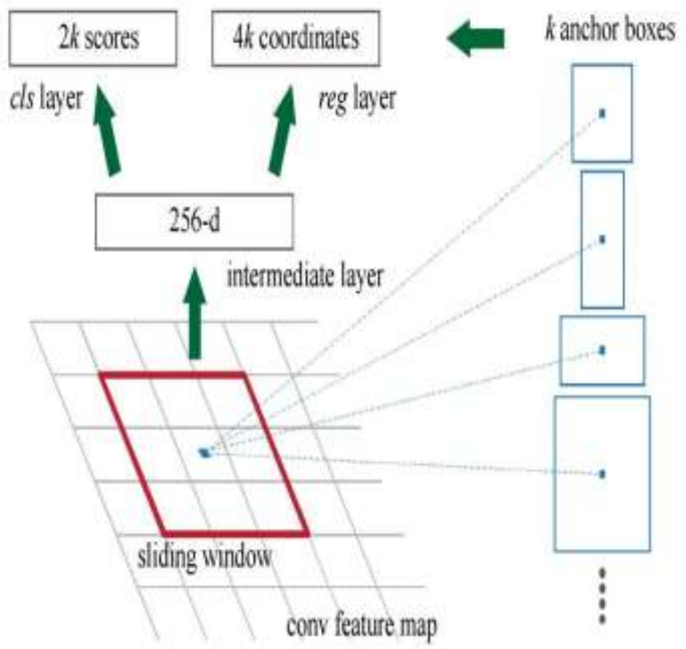

Figure 2: The RPN generating bounding boxes for one position in the final layer convolutional feature map (credit [6]).

The box-classification layer generates $2k$ outputs, where each pair of 2 outputs is the probability that the corresponding bounding box has an object in it or is just background. This means that the sum of each pair of outputs is 1, and is the probability distribution over whether the bounding box contains an object or not. In order to reduce the number of bounding box proposals, non-maxima suppression is used on proposals that have high intersection-over-union (IoU) with each other. The boxes are ranked based on the object probability score. Finally, the top *N* proposals are taken from the remaining proposals, again ranked by object probability score.

The RPN is trained using a multi-task loss which is quite similar to that of the detector Fast R-CNN network. A proposal is designated as a positive training example if it overlaps with a ground-truth box with an IoU greater than a predefined threshold (e.g. 0.7), or if it is the bounding box that has the highest IoU with a ground truth. A proposal is designated as a negative example if its maximum IoU with any ground truth box is less than another predefined threshold (e.g. 0.3). Having a huge margin between the two thresholds avoids learning on ambiguous cases. The ground truth regression outputs t^*_x, t^*_y, t^*_w, and t^*_h are parameterized similar to the the normal outputs (i.e. compared against the anchor location (x_a, y_a), box width w_a, and box height h_a). The multi-task loss has a classification component, and a regression component that is only trained on for positive examples, similar to the Fast R-CNN detector network loss. Then a detector network with separate conv layers is trained using the RPN's proposals. Afterwards, an RPN is trained using the detector network's conv layers. However, the conv layers are fixed, so only the RPN specific layers are trained. Finally, the first detector network's fully connected layers are fine-tuned, using the new RPN's proposals.

3.4 Multiple Object Detection in Videos

In this section, the algorithm for generation, processing and post-processing of a frame of a video is described. As an output, the algorithm returns a list of detected objects. An overview of the algorithm is shown in fig 3.

A. Generate Region Proposals

In order to generate region proposals, we first use the Region Proposal Network to produce an initial set of proposals. The outputs (i.e. proposals) are sorted on the basis of their objectiveness score (likeliness of an object to be present in the proposed region). Next, we introduce extra criteria to further limit the amount of region proposals. These criteria are related to:

a. The size of a region proposal: A minimum size value is set as a threshold for the size of each region. This enables filtering out the small regions that possibly show only parts of an object.

b. The height-width ratio of a region: This value is used as a threshold to filter out regions with unrecognisable objects, which arise due to scaling and cropping operations during the preprocessing stage.

c. The position of the region proposal: Most objects are situated around the center of an image rather than near the corners of the same image. Hence, the regions which contain these objects are positioned around the center of the image as well. To eliminate regions close to the corners of an image, a threshold is applied to the normalised distance between the center of the image and the center of the region.

d. The overlap between regions: When the overlap exceeds a predefined threshold, the smallest region will be eliminated. This limits the amount of region proposals covering the same area of an image. Out of the remaining list of region proposals, we will keep the first 15 proposals to limit the computation time to process all region proposals.

B. Process Region Proposals

During the processing phase, we compute the probabilities over all objects for each region proposal. Before the regions are fed to the object detection network, we apply two pre-processing steps for each region. First, the region is scaled such that the longest side of the region is has a width/length of 256 pixels. On the rescaled region, a center 224x224 crop operation is applied. This change of dimensions of the region is required to fulfill the dimension constraint of the object detection network. This network requires a fixed 224x224 RGB image as input to be able to process the region correctly. During the second pre-processing step, normalization of the image is done by subtracting the mean pixel value of the pixel values of the region. This normalization improves the performance of the object detection network. The preprocessed regions are now given as inputs to the object detection network one by one. The output of the network is stored for each region.

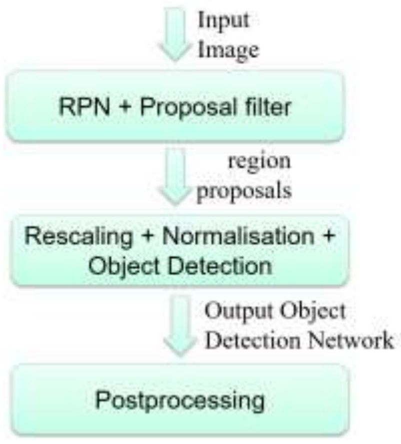
Input Image
RPN + Proposal filter
region proposals
Rescaling + Normalisation + Object Detection
Output Object Detection Network
Postprocessing

Fig. 3. Multiple Object Detection in Videos: algorithm

C. Post-processing

During the post-processing phase, we analyse and combine the results of each region processed in the previous subsection, with the cues present in the previous frame. A cue is a sequence of regions, spread over consecutive frames, covering a similar area. Each cue has a combined output to decide what object is detected in this series of regions. The process of analysing the regions and updating or creating new cues consists of four steps.

First, all non-interesting regions in the current frame are filtered out. This is performed by applying a threshold on the probability that an object is detected in the selected region. When the threshold value is not exceeded, the region is filtered out. Next, the interesting regions are used for the second post-processing step.

During the second step, we merge similar regions in the current frame. To do this, the similarity between the regions is calculated by using two thresholds. The first threshold is based on the overlap between two regions. The second threshold is based on the similarity of these regions. The similarity is calculated by using selective search to acquire interesting region based features. Another similarity measure that is used to calculate the total similarity is the cosine similarity between the probability arrays for each region. The total similarity is the product of both the region similarity and the content similarity.

The similarity is then thresholded and merged with the larger region if the requirement is met. The new probability vector is calculated by using a combination of both probability vectors.

During the third step, the interesting and distinct regions in the current frame are used to create and update cues. For each region, the total distance is calculated between all the existing cues. The total distance is a combination of the similarity of the position of the centers of the region and the last added region of the cue, the dimensions these regions and the similarity between the probability arrays of the region and the cue.

After a comparison between the region and all cues, the minimal distance is chosen. Next, the minimal distance is thresholded. If the minimal distance is too high, a new cue is created. Otherwise, the region is added to the cue with the lowest distance.

When all frames of the scene are processed, the fourth and last step is executed. Now the final cues are thresholded on the maximum probability of each cue, as well as the maximum objectiveness score of the cue. Also the number of regions present in a cue is thresholded. Finally, all the cues with a high certainty are kept and the objects present according to these cues are marked as detected.

4. Results

We trained our Faster-RCNN model on a subset of the training data provid. We heavily modified the open source Faster R-CNN Caffe code released by the authors . We did not have the time or computational resources to train on the entire dataset, so we chose to only train on the initial dataset released by the challenge (the full dataset was released in two parts).

Class	mAP	Class	mAP
Cat	76.4	Chair	50.7
Dog	72.3	Laptop	55.2
Book	75.3	Kite	57.7
Bottle	54.3	Person	70.3
Ball	74.2	Apple	71.3
Pen	58.4	Cup	67.6
Mobile phone	62.7	Table	68.3

Table 1: Class mean average precision for validation set.

We used Faster-RCNN convolutional backend and by applying our algorithm for detecting multiple object in videos, we achieved a good performance on the validation set, we achieved a mAP of 65.33%. We performed well for classes like train and hamster, where the object is often prominently featured in each frame. We did poorly on classes like Chair and bottle, where the object is heavily occluded.

5. CONCLUSIONS

We have developed an object detection system that is capable of detecting multiple objects in videos. The technique we have used to perform this task is the Region Proposal Network. This network is capable of proposing regions that possibly contain an object. Besides the Region Proposal Network we have also introduced extra criteria for each region proposal to limit the amount of final regions that need to be examined by the object classification system. The last part of the developed system introduces an algorithm which exploits temporal information between consecutive frames in a video to improve the accuracy of the system. After fine-tuning the parameters of this algorithm, the overall accuracy measured using a test set of videos is 65.33%. While some categories perform better than others, we must be cautious of the low accuracy for background images. In the future we will need to develop techniques that can significantly increase the accuracy of the background class, since this will also increase the accuracy for the other categories.

REFERENCES

[1] Statista, (2015). Digital ad spend worldwide 2015 — Statistic.

[2] Statista, (2015). Distribution of advertising spending worldwide by medium 2013-2018 — Forecast.

[3] Cisco, (2015). Cisco Visual Networking Index: Forecast and Methodology, 2014-2019 White Paper. [online] Available at http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ ip-ngn-ip-next-generation-network/white paper c11-481360.html

[4] Invespcro.com, (2015). Online Video Advertising Statistics And Trends — The Invesp Blog. [online] Available at: http://www.invespcro.com/ blog/online-video-advertising/

[5] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation CVPR 2014 , pp. 2-9

[6] R. Girshick, Fast R-CNN, arXiv (2015)

[7] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time

Object Detection with Region Proposal Networks, Arxiv (2015), pp. 1-10

[8] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers,A.W.M. Smeulders, Selective Search for Object Recognition, International Journal of Computer Vision (2013), Vol. 104, pp. 154-171

[9] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei Large-scale Video Classification with Convolutional Neural Networks, CVPR2014, pp. 1725-1732

[10] J. Yue-Hei, M. Hausknecht, S. Vijayanarasasimhan, O. Viruyals, R. Mongo, G. Toderici Beyond Short Snippets: Deep Networks for Video Classification, CVPR2015, pp. 4694-4702

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei Imagenet Large Scale Visual Recognition Challenge, 2015, Vol. 115, pp.211-252.

[12] Google Scraper https://github.com/NikolaiT/GoogleScraper

[13] Karen Simonyan, Andrew Zisserman Very Deep Convolutional Networks for Large-Scale Image Recognition 2015

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010.

[15] Yann LeCun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In Proceedings of the IEEE, volume 86, pages 2278–2324, 1998.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural

networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012.

[17] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I, pages 818–833, 2014.

[18] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-theshelf: an astounding baseline for recognition. abs/1403.6382, 2014.

[19] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. International Journal of Computer Vision, 2013.

[20] C. Lawrence Zitnick and Piotr Dollar.´ Edge boxes: Locating object proposals from edges. In ECCV. European Conference on Computer Vision, September 2014.

[21] Justin Johnson, Andrej Karpathy, and Fei-Fei Li. Densecap: Fully convolutional localization networks for dense captioning. abs/1511.07571, 2015.