CCNY at TRECVID 2015: Video Semantic Concept Localization

Yuancheng Ye[1], Xuejian Rong[2], Xiaodong Yang[3], and Yingli Tian[1,2]

[1]The Graduate Center, City University of New York
yye@gradcenter.cuny.edu

[2]The City College, City University of New York
{xrong, ytian}@ccny.cuny.edu

[3]NVIDIA Research
xiaodongy@nvidia.com

Abstract

In this paper, we present a novel video-based object localization system, which is developed for the Localization task of TRECVID 2015. Our system is based on the R-CNN which is one of the state-of-the-art image-based object localization algorithms. In our system, in addition to the selective search method, EdgeBoxes algorithm is also applied to generate candidate region proposals. Two CNN models are adopted in this paper: AlexNet and GoogLeNet. The features of each region extracted from these two models are ℓ_2 normalized and concatenated. After that the linear SVM classification model is employed. Since the R-CNN is image based algorithm and does not take temporal information into consideration, we propose a region trajectory algorithm which can keep tracking possible object regions while prune false detections. Our system ranks the first place in the temporal measurement and the third in the spatial measurement. The result demonstrates that our system can robustly and effectively localize objects in videos.

1 Introduction

Video-based objects localization task aims at detecting both the spatial and temporal locations of the targeted objects in videos. Most approaches to tackle this problem are based on algorithms designed for spatial localization of objects in images and then extending to the temporal dimension by treating each frame as one image. For most standard image-based object detection algorithms, there are two main challenges: 1) how to generate sufficient and effective region proposals, which should encompass enough information for all the contents in an image; 2) how to develop a robust and efficient algorithm to extract discriminative features from each region proposal. Traditional approaches use sliding windows to produce region proposals, and then employ hand-crafted feature extraction algorithms to generate feature representations of each region proposal.

However, these approaches are relatively time consuming and generally have many manually defined rigid parameters.

Although support vector machines (SVM) [2] have been widely applied as efficient classifiers for many applications, convolutional neural networks (CNN) restart drawing attention with impressive image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 [8]. CNN has significantly improved the performance of visual recognition and object detection tasks, as compared to the classical methods based on the scale-invariant feature transform (SIFT) [10] and the histogram of oriented gradients (HOG) [3].

Starting with LeNet-5 [9], the typical structure of the convolutional neural networks (CNN) has been well defined, namely, stacked convolutional layers (optional contrast normalization and max-pooling layers in between) followed by several fully-connected layers. Variants of this basic design have been applied to different tasks, and yielded the top-performance results on various datasets. As to the object detection area, Girshick proposed the region-based convolutional neural network (R-CNN) [5] to bridge the gap between image classification and object detection, which dramatically outperforms other methods adopting sliding-window paradigm and HOG-like features on PASCAL VOC Challenge [4]. In this paper, we develop a video-based object localization system upon the R-CNN approach. Aside from the AlexNet, which is employed in the initial version of R-CNN, we also employ another newly proposed network structure, GoogLeNet [14], in our detection algorithms. By concatenating the features extracted from these two CNN models, linear SVM classifier is then employed to learn a more discriminate model to classify each region proposals.

However the image-based object localization R-CNN may not be able to extract important temporal information when applied to individual frames of a video. Therefore, inspired by [16] which applied dense trajectory for action recognition task, we propose a novel region trajectory algorithm to effectively exploit temporal information. With the assumption of the temporal continuity, our proposed algorithm can register missing candidate object regions as well as prune false candidate regions by checking the validity of each region trajectory. The experimental results demonstrate that our region trajectory algorithm can improve the accuracy of object localization both in the temporal and spatial measurements.

This paper is organized as follows. Section 2 describes the approaches employed in our localization system. Specifically, an overview of our system is presented in Section 2.1, and in Section 2.2 the basic ideas and components of R-CNN are discussed. In Section 2.3, two network structures, AlexNet and GoogLeNet, are introduced. Section 2.4 explains in detail about our proposed region trajectory algorithm. In Section 3, all the results of our submitted four runs are presented and discussed. Finally, Section 4 makes a conclusion about our video-based objects localization system in this TRECVID contest.

2 Video-based Object Localization System

In this section, we introduce the structure of our system for object localization in videos and discuss the details of each component respectively.

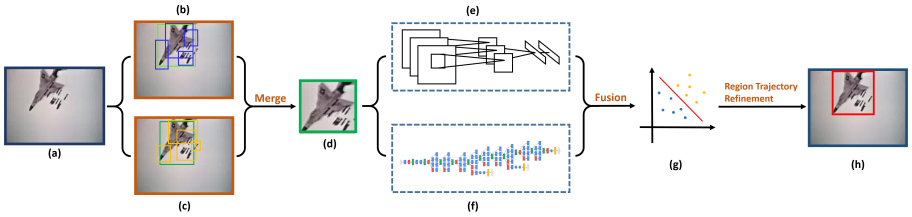

Figure 1: Overview of System Structure. (a) Input image. (b) Region proposals by SelectiveSearch. (c) Region proposals by EdgeBox. (d) Region proposals fused by (b) and (c). (e) Extracting features of region proposals by AlexNet. (f) Extracting features of region proposals by GoogLeNet. (g) Applying linear SVM classifier to the vector concatenating by the ℓ_2 normalization of (e) and (f). (h) Final region box obtained by the region trajectory algorithm.

2.1 System Overview

As illustrated by the Fig. 1, our system consists of four main components: R-CNN, feature fusion, SVM, and region trajectory. In the R-CNN component, two networks are applied: AlexNet and GoogLeNet. This process serves as a feature extractor, which extracts CNN features from each region proposal. Then the features from different CNN networks are concatenated before fed into the linear SVM classifier. After that an effective region trajectory algorithm is employed to predict possible object regions and prune plausible ones based on the temporal information between iframes, which define the starting and ending points of any smooth movement in a video. We will discuss each part in detail in the following sections.

2.2 Region-based Convolutional Network (R-CNN)

R-CNN decomposes the overall detection problem into three modules: 1) utilizing low-level image cues to generate potential category-independent object region proposals; 2) adopting a large pre-trained convolutional neural network to extract a fixed-length feature vector for each proposed region. 3) training a set of category-specific linear SVM classifiers. This kind of three-stage approach well leverages the accuracy of bounding box segmentation with low-level cues, as well as the highly powerful feature extraction capability of the state-of-the-art CNN. The details of the stages in our implementation are describes as follows.

Region Proposals Recently many object detection methods for generating category independent region proposals have been proposed, including EdgeBoxes [17], Objectness [1], Selective Search [15], and etc. We choose to employ Selective Search and EdgeBoxes to generate region proposals based on the evaluation of the influence of different proposal methods on R-CNN [6].

Feature Extraction In our implementation, by forward propagating a mean-subtracted RGB region proposal through a standard CNN architecture, a 4096-dimensional feature vector is extracted from last fully connected layer using

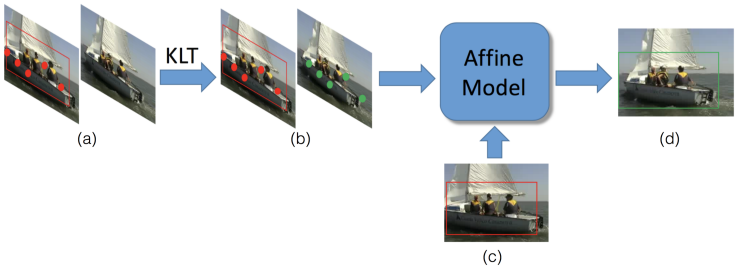

Figure 2: The pipeline of the region trajectory algorithm. (a) Two consecutive frames in the video. The red bounding box in the first frame is the object location detected by R-CNN, while there is no region detected in the second frame. The red dots in the first frame are the interested points detected by [13]. (b) Same as (a), while the green dots in the second frame represent the points tracked from the interested points detected in the first frame (the red points) by KLT algorithm. By applying these two sets of points, an affine model can be trained. (c) The first frame with the region detected by R-CNN. (d) The predicted object region in the second frame obtained by applying the bounding box in the first frame to the affine model trained previously.

Caffe [7], which is a popular deep learning tool. In detail, the size of each region proposal is normalized to meet the input size requirement of CNN, by warping all pixels in a tight bounding box around the candidate region to the required size. Prior to warping, the tight bounding box is dilated to make sure at the warped size there are exactly p pixels of warped image context around the original box ($p = 16$ hereby).

Object Category Classification After the features are extracted, the score of each extracted feature vector for each class is obtained by using the SVM trained for that class. Given all scored regions in an image, a greedy non-maximum suppression is independently applied for each class independently to reject the regions which have an intersection-over-union (IoU) overlap with a higher scoring selected region larger than a learned threshold. The actual threshold used in our system is provided in the experiments section.

2.3 CNN Models: AlexNet and GoogLeNet

In this section, we will discuss in detail the two CNN models employed in our system: AlexNet and GoogLeNet.

AlexNet AlexNet was first proposed in the paper [8], as a milestone to apply CNN in the area of image processing. AlexNet consists of 5 convolution layers and 2 fully connected layers. There are three max pooling layers right after the first, second and fifth convolution layers respectively. The size of the input image is normalized to 227×227. The kernel size of the first convolution layer is 11×11, and 5×5 for the second convolution layer. For the following convolution layers, the kernel sizes are all 3×3. The final feature dimension of AlexNet is 4096.

GoogLeNet GoogLeNet was proposed in the paper [14], which introduced the concept inception in the deep learning area. The feature maps produced by different kernel sizes are concatenated before fed into the next layer. This process can significantly reduce the parameters of each layer, while exploits discriminative power of different kernel sizes. Although this structure is very deep, the computational resources of GoogLeNet are relatively small since it employs many inception layers, which substitutes some of the 5×5 and 3×3 kernels by the simple 1×1 filter.

The experimental results demonstrate that these two CNN networks can extract complemental features by concatenating the output features of the final fully connected layers.

2.4 Region Trajectory

When applying R-CNNs in video-based object localization task, important temporal information may be neglected. To effectively incorporate temporal information, we propose the region trajectory algorithm.

Region trajectory algorithm is based on the affine transformation and Kanade-Lucas-Tomasi Feature Tracker (KLT) [11] methods. The pipeline is demonstrated by the Fig. 2. Firstly, we extract interested points by the Shi-Tomasi corner detector [13] in the object regions detected by the R-CNN algorithm. Then the KLT method is applied to track the extracted points. The points that have no correspondences in the next frame are deleted. After that an affine model can be obtained from the two sets of points in the two consecutive frames respectively. Finally, by applying the region corners of the first frame to the affine model, we can calculate the corner points of the regions in the second frame.

If the biggest IoU of the predicted region and the R-CNN detected region is larger than a threshold (0.5 in our implementation), the R-CNN detected region is replaced by the predicted region and then continue the tracking algorithm. Meanwhile, the R-CNN detected region is deleted from the untracked pool to avoid duplicate tracking.

To prune the plausible region trajectories, we set the threshold of the ratio of the number of R-CNN detected regions and the total number of the regions in the trajectory. We observe that using the average SVM score of each trajectory as the threshold can not improve performance. The reason is that some false regions may have high SVM scores, which may make the average score of the regions of false trajectory higher than correct region trajectories. On the other hand, the value of the ratio can reflect how many regions in the trajectory are not predicted by the R-CNN algorithm, which implies that the higher the ratio, the better the chance that the trajectory is correct.

3 Experimental Results

The TRECVID dataset [12] is comprised of ten concepts: airplane(1003), anchorperson(1005), boat_ship(1015), bridges(1017), bus(1019), computers(1031), motorcycle(1080), telephones(1117), flags(1261), and quadruped(1392). In practice, ten CNN networks are trained corresponding to these ten concepts respectively, therefore we conduct this task as binary classification. We totally

submitted four runs for this task. There are four other teams submitted final results for this task: MediaMill (University of Amsterdam Qualcomm), PicSOM (Aalto University and University of Helsinki), TokyoTech (Tokyo Institute of Technology), Trimps (Third Researh Institute of the Ministry of Public Security, P.R.China). The results are summarized in the Tables 1, 2 and 3.

Run	iframe_fscore	mean_pixel_fscore
1	0.7447	0.4723
2	0.7682	0.4542
3	0.7309	0.5085
4	0.7661	0.4591
MediaMill*	0.7662	0.6557
PicSOM*	0.6643	0.3944
TokyoTech*	0.6699	0.6688
Trimps*	0.7357	0.4760

Table 1: The results of Mean_Per_Run for four submitted runs. $*$ indicates the best results of other teams among all their submitted runs. iframe_fscore and mean_pixel_fscore are the measurements of temporal and spatial accuracy respectively, and the larger the number stands for better performance of the system.

Run	1003	1005	1015	1017	1019	1031	1080	1117	1261	1392
1	0.8227	0.8196	0.8033	0.6710	0.6394	0.7797	0.5595	0.6752	0.8375	0.8397
2	0.8709	0.8159	0.8016	0.7238	0.7026	0.7749	0.6008	0.7250	0.8251	0.8418
3	0.7920	0.8184	0.7905	0.6834	0.5720	0.7138	0.5595	0.6829	0.8583	0.8382
4	0.8709	0.8159	0.7997	0.7189	0.7014	0.7749	0.5902	0.7212	0.8251	0.8426
MediaMill*	0.8219	0.8535	0.7798	0.6974	0.6783	0.7755	0.6074	0.7896	0.8909	0.8576
PicSOM*	0.6936	0.8245	0.7489	0.6109	0.2685	0.6793	0.5498	0.7620	0.8275	0.7681
TokyoTech*	0.7669	0.8501	0.6721	0.5736	0.4244	0.7111	0.6086	0.5395	0.8475	0.7420
Trimps*	0.7959	0.8253	0.7725	0.5521	0.6381	0.8028	0.6741	0.7584	0.8204	0.7998

Table 2: The results of iframe_fscore for each concept. $*$ indicates the best iframe_fscore for each concept achieved by other teams among all their submitted runs.

In the first run, only the AlexNet is employed without integrating the region trajectory algorithm. In the second run, both AlexNet and GoogLeNet are utilized with applying the region trajectory algorithm. The average accuracy of the temporal localization is boosted from 0.7447 to 0.7682, which demonstrates that by combining with GoogLeNet and region trajectory algorithm, some objects that are neglected in the first run can be detected. However, the result of mean_pixel_fscore which stands for the accuracy of spatial localization is decreased. The reason may be that by introducing the region trajectory algorithm, many plausible trajectories are included, which can deteriorate the accuracy of spatial localization. Both in the first run and second run, the threshold of SVM score for the positive candidates is set to -1. In the third run, the threshold of SVM score is increased from -1 to 0, which contributes to the increase of spatial accuracy from 0.4542 to 0.5048 compared with the second run. The reason is that by increasing the threshold of SVM score, many false detected regions are removed, and then alleviate the amount of plausible regions that

Run	1003	1005	1015	1017	1019	1031	1080	1117	1261	1392
1	0.6421	0.6323	0.4235	0.4165	0.4886	0.2807	0.3762	0.4238	0.4796	0.5600
2	0.6081	0.6276	0.4151	0.3790	0.4415	0.3176	0.3248	0.4038	0.4856	0.5389
3	0.6390	0.6310	0.4748	0.3886	0.5637	0.3502	0.4165	0.5119	0.5181	0.5907
4	0.6081	0.6276	0.4247	0.3916	0.4488	0.3176	0.3326	0.4088	0.4856	0.5456
MediaMill*	0.6476	0.6222	0.6340	0.6584	0.7445	0.7179	0.5744	0.7636	0.5985	0.7102
PicSOM*	0.6234	0.2948	0.6469	0.2804	0.7871	0.2426	0.4307	0.4159	0.4140	0.2998
TokyoTech*	0.6947	0.6368	0.7808	0.4905	0.8310	0.5901	0.5724	0.7519	0.6759	0.6637
Trimps*	0.5622	0.6403	0.4582	0.2852	0.4190	0.5198	0.3565	0.5404	0.4939	0.5262

Table 3: The results of mean_pixel_fscore for each concept. $*$ indicates the best mean_pixel_fscore for each concept achieved by other teams among all their submitted runs.

region trajectory algorithm will introduce. In the final run, the experimental setups are almost same as the second run, except for increasing the pruning ratio threshold from 0.25 to 0.35 in the region trajectory algorithm. As the results of such change, the accuracy of temporal localization decreases from 0.7682 to 0.7661, and the accuracy of spatial localization increases from 0.4542 to 0.4591.

For the temporal localization, e.g. the measurement of iframe_fscore, the second run achieves the best results among all the teams. For the spatial localization, e.g. the mean_pixel_fscore, our result ranks the third. The best result of mean_pixel_fscore is achieved by the team MediaMill in the fourth run: 0.6557.

From the results, we observe that the accuracy of temporal localization can be improved by applying the region trajectory algorithm. However, this process also introduces many false positive regions which may lead to a decrease in the measure of spatial score. By increasing the threshold of the ratio of the number of R-CNN detected regions and the total number of the regions in the trajectory in the third run, a trade-off between temporal and spatial accuracies is made and archives the best mean score of iframe_fscore and mean_pixel_fscore.

We also observe that by fusing the features of AlexNet with the ones of GoogLeNet, the performance can be improved. This phenomenon demonstrates that the two different CNN networks provide complementary attributes for object localization.

Our mean_pixel_fscore, which is the spatial measurement, is inferior to some results of other teams. The accuracy of spatial localization can be further improved by implementing regression model to refine the bounding boxes, which is not employed in our implementation due to the time limit.

4 Conclusion

In the localization task of TRECVID 2015, we have designed a video-based object localization system to detect object regions both in spatial and temporal positions. By combining AlexNet and GoogLeNet in the R-CNN algorithm, more discriminative features of each region proposal can be produced. To incorporate temporal information, we have proposed a novel region trajectory algorithm to predict and prune object regions. The experimental results demonstrate the effectiveness and robustness of our system.

5 Acknowledgement

This work was supported in part by ONR grant N000141310450 and NSF grants EFRI-1137172, IIP-1343402, and IIS-140080.

References

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. "Measuring the objectness of image windows". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.11 (2012), pp. 2189–2202.

[2] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory*. ACM. 1992, pp. 144–152.

[3] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 886–893.

[4] Mark Everingham et al. "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2 (2010), pp. 303–338.

[5] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE. 2014, pp. 580–587.

[6] Jan Hosang et al. "What makes for effective detection proposals?" In: *arXiv preprint arXiv:1502.05082* (2015).

[7] Yangqing Jia et al. "Caffe: Convolutional Architecture for Fast Feature Embedding". In: *arXiv preprint arXiv:1408.5093* (2014).

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[9] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[10] David G Lowe. "Object recognition from local scale-invariant features". In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[11] Bruce D Lucas, Takeo Kanade, et al. "An iterative image registration technique with an application to stereo vision." In: *IJCAI*. Vol. 81. 1981, pp. 674–679.

[12] Paul Over et al. "TRECVID 2015 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics". In: *Proceedings of TRECVID 2015*. NIST, USA. 2015.

[13] Jianbo Shi and Carlo Tomasi. "Good features to track". In: *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE. 1994, pp. 593–600.

[14] Christian Szegedy et al. "Going deeper with convolutions". In: *arXiv preprint arXiv:1409.4842* (2014).

[15] Jasper RR Uijlings et al. "Selective search for object recognition". In: *International journal of computer vision* 104.2 (2013), pp. 154–171.

[16] Heng Wang et al. "Action Recognition by Dense Trajectories". In: *IEEE Conference on Computer Vision & Pattern Recognition*. Colorado Springs, United States, June 2011, pp. 3169–3176. URL: `http://hal.inria.fr/inria-00583818/en`.

[17] C Lawrence Zitnick and Piotr Dollár. "Edge boxes: Locating object proposals from edges". In: *Computer Vision–ECCV 2014*. Springer, 2014, pp. 391–405.