Detecting Temporally Consistent Objects in Videos through Object Class Label Propagation

Subarna Tripathi
UCSD
stripathi@ucsd.edu

Serge Belongie
Cornell University
sjb344@cornell.edu

Youngbae Hwang
KETI
ybhwang@keti.re.kr

Truong Nguyen
UCSD
tqn001@eng.ucsd.edu

Abstract

Object proposals for detecting moving or static video objects need to address issues such as speed, memory complexity and temporal consistency. We propose an efficient Video Object Proposal (VOP) generation method and show its efficacy in learning a better video object detector. A deep-learning based video object detector learned using the proposed VOP achieves state-of-the-art detection performance on the Youtube-Objects dataset. We further propose a clustering of VOPs which can efficiently be used for detecting objects in video in a streaming fashion. As opposed to applying per-frame convolutional neural network (CNN) based object detection, our proposed method called Objects in Video Enabler thRough LAbel Propagation (OVERLAP) needs to classify only a small fraction of all candidate proposals in every video frame through streaming clustering of object proposals and class-label propagation. Source code will be made available upon publication.

1. Introduction

Object proposal generation is a common pre-processing step for object detection in images, which is a key challenge in computer vision. Object proposals dramatically decrease the number of detection hypotheses to be assessed. Thus, use of CNN-features [11], which is more effective but computationally expensive, have turned out to be feasible for accurate detection. For the detection of video objects, proposals not only need to consider the space-time complexity, but also need to address the temporal consistency. We propose generating Video Object Proposals (VOP) by scoring candidate windows based on spatio-temporal edge content and show that these VOPs help in learning better video object detectors. Further, we propose an efficient online clustering of these proposals in order to process arbitrary long videos. We show that the joint-analysis of all such windows provides a way towards multiple object segmentations and helps in reducing test time object detection complexity.
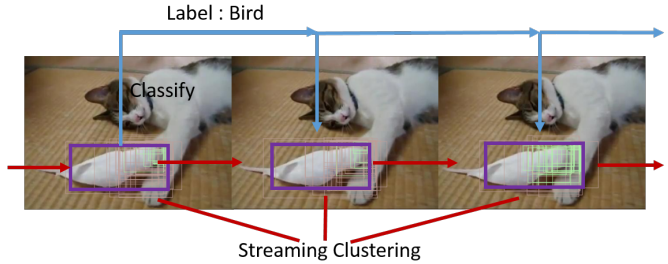
We divide a video into sub-sequences with one-frame

Figure 1. OVERLAP: Object class labels get propagated by streaming clustering of temporally consistent proposals. The system classifies only those VOPs which belong to a new cluster.

overlap in a streaming fashion [37, 34]. We analyze all candidate windows jointly within a sub-sequence followed by affinity-based clustering to produce temporally consistent clusters of object proposals at every video frame. The advantage of performing a streaming spatio-temporal clustering on the object proposals is that it enables an easy label propagation through the video in an online framework. Presumably all object proposals of a cluster have the same object class type. We propose deep-learning based video object detection through objects' class label propagation using online clustering of VOPs. As opposed to applying R-CNN [11]-like approaches, which essentially classify every window based on the expensive CNN features, at every video frame, the proposed method of label propagation requires detection/classification only on video frames which has new clusters (fig. 1) . Our main contributions are as follows:

- We present a simple yet effective Video Object Proposal (VOP) method for detecting moving and static video objects by quantifying the spatio-temporal edge contents;

- We present a **novel algorithm**, "Objects in Video Enabler thRough LAbel Propagation" (OVERLAP), that exploits objects' class label propagation through streaming clustering of VOPs to efficiently detect objects in video with temporal consistency.

We also present the following minor contributions:

1

- Demonstrating VOP's efficacy in learning a better CNN-based video object detector model;
- Object segmentation as a by-product of object detection framework, OVERLAP.

2. Related Works

There are several approaches towards video object detection which broadly fall into three categories : (1) image object proposals for each frame (2) motion segmentation in video (3) supervoxel aggregation.

Object Proposals and detection: Unsupervised category-independent detection proposals are evidently shown to be effective for object detection in images. Some of these methods are Objectness [1], category-independent object proposals [8], SelectiveSearch [35], MCG [2], GOP [21], BING [6], EdgeBoxes [7]. A comparative literature survey on object proposal methods and their evaluations can be found in [15, 14]. Although there is no "best" detection proposal method, EdgeBoxes, which scores windows based on edge content, achieve better balance between recall and repeatabilty.

Applying image object proposals directly for each frame in video may be problematic due to time complexity and temporal consistency. In addition, issues like motion blur and compression artifacts can pose significant obstacles to identifying spatial contours, which degrades the object proposal qualities. Recent advances like SPPnet [13], Fast R-CNN [12], and Faster R-CNN [31] have dramatically reduced the running time by computing deep features for all image locations at the same time and snapping them on appropriate proposal boxes. Per-frame object detection still needs classification of proposal windows and temporal consistency still remains a challenge. The proposed framework dispenses with the need of classifying every candidate window of every video frame through spatio-temporal clustering, thus addressing temporal consistency.

Motion Segmentation in Video: Motion based segmentation is the task of separating moving foreground objects from the background. Several popular methods of motion segmentation include the layered Directed Acyclic Graph (DAG) based framework [39], Maximal weight cliques [26], fast motion segmentation [28], tracking many segments [25], identifying key segments [24], and many more. Although video motion segmentation can detect moving foregrounds robustly, it is not easy to detect multiple objects or if objects suddenly stop or change motion abruptly.

Supervoxel aggregation: Spatio-temporal object proposals have been considered in the context of aggregating supervoxels with spatio-temporal connectivity between neighboring labels. Jain *et al* [18] developed an extension of the hierarchical clustering method of SelectiveSearch [35] to obtain object proposals in video. Even though their independent motion evidence effectively segment objects with

motions from the background, static objects can not be recovered. Oneata *et al.* [27] presented spatio-temporal object proposals by a randomized supervoxel merging process. Sharir *et al.* [32] proposed the extension of category-independent object proposals [8] from image to video by extracting object proposals at each frame and linking across frames into object hypotheses in a framework of graph-based segmentation using higher-order potentials leading to a high computational expense. All these supervoxels often cannot replace the object proposal step for object detection either due to its complexity or the associated over-segmentations.

3. Video Object Proposals

We extend EdgeBoxes [7] from generating image object proposals to video object proposals. In addition to the spatial edge responses, $\mathbf{E_s}$, at every pixel in EdgeBoxes [7], we consider exploiting temporal edge responses, $\mathbf{E_t}$, at every pixel location using mid-range optical flow analysis.

$$\mathbf{E_t}, \mathbf{E_t} \in \mathbb{R}_{\geq 0}^{M \times N} \tag{1}$$

M and N are the height and width of an image.

3.1. Spatio-Temporal Contours and VOP

Optical flow field between pairs of consecutive frames provide approximate motion contours. We perform 2-frame forward optical flow [5] for every consecutive frame-pair. At every pixel, the magnitude of the flow field's gradient and the difference in direction of motion from its neighbor [28], contribute to the measure of the motion contour. To address incompleteness and inaccuracies of two-frame optical flow estimation, we analyze mid-range optical flow over a subset of video frames. Within a sub-sequence, we approximate which pixels consistently reside inside a moving object using inside-outside maps [28]. In our experiments, the inside-outside maps, accumulated over 3 - 5 frames, provide good estimates of time-consistent gross location priors for moving objects. In Section 5.2, We describe how this mid-range accumulation can effectively be exploited for object label propagation through streaming clustering. A simple edge detector on this location prior is called temporal edge, $\mathbf{E_t}$.

EdgeBoxes [7] employs efficient data structures to score millions of candidates based on the difference of number of spatial contours that exist in the box and those that straddle box's boundary. We use a similar scoring strategy, but on spatio-temporal edge $\mathbf{E} \in \mathbb{R}_{\geq 0}^{M \times N}$ which is formed according to Eq 2.

$$\mathbf{E} = \lambda \mathbf{E_t} + (\mathbf{1} - \lambda)\mathbf{E_s}, \lambda \in [0, 1] \tag{2}$$

As the value of λ increases, the system favors detecting only moving objects. One example of spatio-temporal edge is demonstrated in figure 2. A linear combination
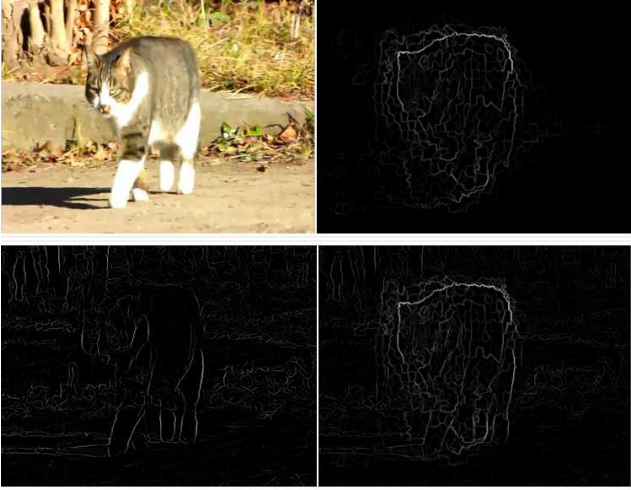
Figure 2. Spatio-Temporal EdgeBoxes. Clock-wise from top-left - (i) A video frame from Youtube-Objects dataset, (ii) Temporal edge $\mathbf{E}_t$ *i.e.* normalized gradient of location prior from motion analysis, (iii) Spatial edge $\mathbf{E}_s$ *i.e.* structured edge detection on video frame, (iv) linear combination of spatial and temporal edge $\mathbf{E}$ brings out the prominent contours of both static and/or moving objects.

of spatial and temporal edge responses represents spatio-temporal contours. This enables a simple yet efficient strategy for scoring based on spatio-temporal edge content through edge groups. We find intersecting edge groups along horizontal and vertical boundaries using two efficient data structures[7]. We use the integral image-based implementation to speed up scoring of boxes in sliding window fashion. As described in the next section, object proposals based on these spatial-temporal contours outperforms those based on only spatial contours for video object detection. The presence of motion blur in spatial edges affects the performance of spatial contour based proposal prediction in video frames. In practice, $\lambda = 0.2$ to 0.5 works well for Youtube-Objects dataset.

4. Learning Video Object Detector Model

We aim for detecting objects in generic consumer videos. Due to the domain shift issues between images and video frames [20], our 10-class video object detection uses supervised pre-training from ImageNet reference model for classification and fine-tuning on annotated frames from Youtube-Objects dataset v2.0 [29, 20, 30] for video objects detection.

Youtube-Objects dataset. The dataset is composed of videos collected from Youtube by querying for the names of 10 object classes of the PASCAL VOC Challenge. It contains 155 videos in total and between 9 and 24 videos for each class. The duration of each video varies between

30 seconds and 3 minutes. However, only 6087 frames are annotated with a bounding-box around an object instance. Hence, the number of annotated samples is approximately 4 times smaller than in PASCAL VOC.

The bottom-up region proposal methods play an important role. Motion blur and compression artifacts affects the quality of spatial edges in video frames, thus, generating good object proposals becomes more challenging. This is to be noted that R-CNN [11], or, Fast R-CNN [12] are fine-tuned for image object detection task, especially for 20-class PASCAL VOC image object categories which is a superset of Youtube-objects categories.

Feature extraction. We extract a 4096-dimensional feature vector corresponding to each region proposal using GPU-based (GeForce GTX 680) the Caffe [19] implementation of the CNN described by Krizhevsky *et al.* [22]. Features are computed by forward propagating a mean-subtracted 227×227 R-G-B image through five convolutional layers and two fully connected layers.

Region Proposals. We use approximately 2000 candidate proposals per video frame to be processed for learning detectors. We investigate the object detection model with different region proposal methods such as selective search[35], EdgeBoxes[7]. As the resolution of different videos varies from VGA to HD, we re-size every video frame to 500×500 before performing proposal generation task.

Training. We discriminatively pre-train the CNN on a large auxiliary dataset (ILSVRC2012 classification) using image-level annotations, followed by domain specific fine-tuning by replacing the last layer of AlexNet [22] model with $10 + 1$ softmax output layer. We use two-step initialization for fine-tuning as described in [4]. As per PASCAL detection criteria, we treat all region proposals with ≥ 0.5 IoU overlap with a ground-truth box as positives for that class of the box and the rest as negatives. Once features are extracted and training labels are applied, we optimize one linear SVM per class.

Test time detection. During test time, approximately 500 to 2000 VOPs are generated. Then forward propagation is performed through the CNN to compute features. Finally, we perform scoring using per-class trained SVM similar to [11] followed by non-maximum suppression.

In the below section 5, we describe the alternate test time detection using OVERLAP through objects' class label propagation.

5. OVERLAP: Objects in Video Enabler thRough LAbel Propagation

Classical approach for object localization has traditionally been image window classification, where each window is scored independent of other candidate windows. Recently, more success in object detection has been reported

by considering spatial relations to all other windows and their appearance similarity [36] with examplar-based associative embedding. Our approach towards video object detection considers spatial relationship and appearance similarity with windows within and even in other nearby video frames, yet in much simpler way through spatio-temporal clustering, to avoid classifying every candidate windows.

5.1. Joint analysis of windows

We aim to detect dissimilarity between VOPs generated within a sub-sequence based on a simple underlying principle: proposals corresponding to the same object exhibit higher statistical dependencies than proposals belonging to different objects.

As a motivation for our approach, we consider generating proposal boxes by perturbing the ground truth locations of PASCAL VOC 2007 objects' bounding boxes and observe the statistical association of those proposal boxes. Let, A and B denote generic features of neighboring proposal windows, where neighborhood is characterized by non-zero Intersection-Over-Union (u). We investigate the joint distribution over pairings A, B. Let, $p(A, B; u)$ be the joint probability of features A and B of windows with spatial overlap value, u. Then, $P(A, B)$ could ideally be computed as :

$$P(A, B) = \frac{1}{Z} \int_u^1 w(u) p(A, B; u) du \qquad (3)$$

where, $u \in [0, 1]$, w is a weighting function, $w(0) = 0$, and Z is a normalization constant. To simplify the process, we use uniform weighting function and work in the discrete (quantized) space of u and replace the integral with summation. We take the marginals of the distribution to get $P(A)$ and $P(B)$. Motivated by the analysis presented in crisp boundary detection work by Isola et al. [17], we model affinity with point-wise mutual information like function:

$$PMI_\rho(A, B) = log \frac{P(A, B)^\rho}{P(A)P(B)} \qquad (4)$$

We choose the value of ρ to be 1.2 which produces best performance in PASCAL VOC dataset with perturbed ground truth proposals. In order to identify the boundary between two features, the model needs to be able to capture the low probability regions of P(A,B). We use a non-parametric kernel (Epanechnikov) density estimator. The number of sample points are the number of overlapping candidate windows. We perform affinity-based clustering afterwards. The affinity matrix, $\mathbf{W}$, for a sub-sequence is created from the affinity function, PMI_ρ, as follows:

$$\mathbf{W}_{i,j} = e^{PMI_\rho(\mathbf{f}_i, \mathbf{f}_j)} \qquad (5)$$

Where i and j are the indices of proposal windows and $\mathbf{f}$ is the feature vector defined for a proposal window. Figure 3 shows an example of spatial clustering of proposal windows. The boxes drawn in same colors correspond to same
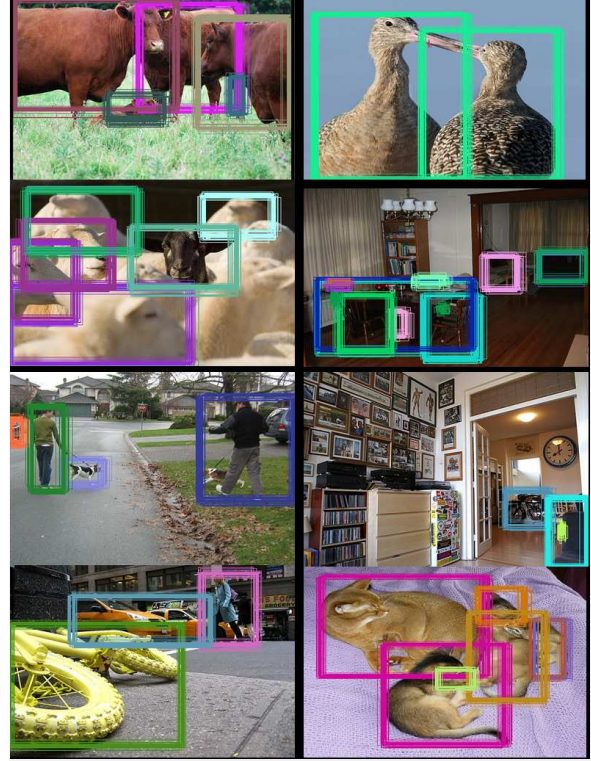

Figure 3. Synthetic experiment using perturbed ground truth locations in PASCAL VOC showing affinity-based clustering of candidate windows. The clustering technique works efficiently with some exceptions where overlapping object instances share very similar color.

clusters. In the streaming VOP clustering framework for Youtube videos, we perform the joint analysis on all proposals within a sub-sequence. Intuitively this is an easy yet effective clustering technique which works on actual test proposals (see Figure 8) which are not simply perturbed ground truth bounding boxes.

This is to be noted that the proposed method measures the affinity between different object proposal windows (of varying sizes) within a video sub-sequence (in different video frames) unlike [17], where the affinity is between the neighboring pixels in an image.

We demonstrate object segmentation can be achieved as a by-product of this clustering algorithm. We cast the segmentation problem through random-field based background-foreground segmentation without manual labeling [23]. Uniformly weighted sum of the location of every window corresponding to a unique cluster defines the foreground location prior for that cluster. However, unlike [23], in our approach, the location prior is not coming from the global neighbors of the image but from within itself and the clustering allows multiple objects segmentations. The segmentation works well if the proposal boxes tightly en-
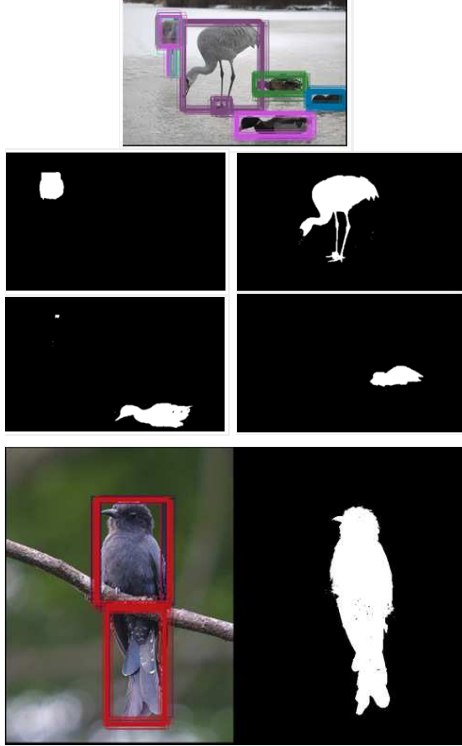
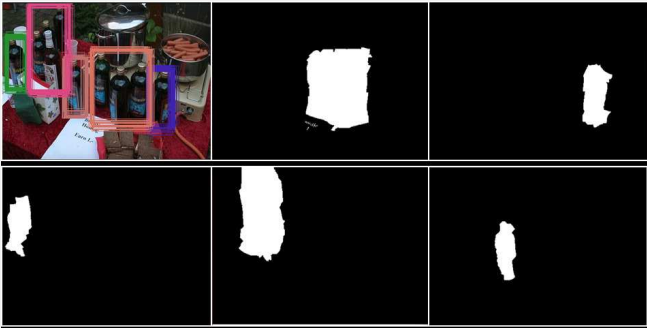Figure 4. Segmentation masks from clustered object proposals.

Figure 5. Segmentation masks generation is not successful where proposal boxes do not tightly enclose the actual object. This happens in cluttered background cases.

close an object as shown in figure 4 with some failure cases as shown in figure 5 where the proposal boxes do not tightly enclose the object in a cluttered background.

Figure 6 shows two segmentation masks generated on individual video frames from the clustered video object proposals generated by the proposed VOP on real videos.

5.2. Streaming clustering of proposals

One of the main contributions of this paper is a simple, principled, and unsupervised approach to spatio-temporal grouping of candidate regions in streaming fashion. We describe a clustering framework which enforces a Marko-
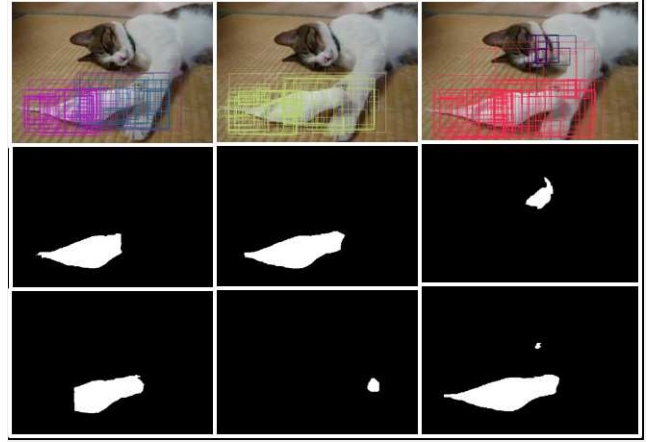

Figure 6. Two segmentation masks from clustered video object proposals from Youtube video "Bird and Cat" for frames #5, #20 and #45.

vian assumption on the video stream to approximate a batch grouping of VOPs. A video is divided into a series of sub-sequences with one frame overlap with the previous sub-sequence as described in [37]. VOP clustering within the current sub-sequence depends on the results from only the previous sub-sequence. We consider sub-sequence length of 3 to 5 frames as a trade-off between quality and complexity. This is the same sub-sequence volume, where mid-range motion analysis is performed for detecting temporal edges (Section 3.1). The color-histogram features are used for estimating joint probability between any overlapping window-pair and affinity-based clustering is performed afterwards. There are two important aspects in this streaming clustering method. The first is generic to any clustering algorithm *i.e.* how to select the number of clusters and the second is specific to streaming method *i.e.* how to associate cluster number of the current sub-sequence with any of the clusters of previous and/or future sub-sequences?

5.2.1 Number of clusters.

Common consumer or Youtube videos contain limited number of moving objects, often less than five. Youtube-Objects dataset contains maximum of 3 object instances and quite often a single moving object. We assume the presence of at most 5 objects to keep the computational complexity tractable and amenable to practical applications. We explore two modes of operations. The first method uses fixed number of clusters, $k = 5$, with careful initialization of cluster centers using k-mean++ [3] during spectral clustering. The second one is the spectral clustering with self-tuning [38]. We observe that while self-tuning outperforms the fixed cluster number case for hypothetically good object windows (such as the perturbed ground truth regions for PASCAL VOC), both modes perform similarly in case of

real object proposals generated by some proposal method.

5.2.2 Cluster Label Association.

In the streaming framework, any sub-sequence except for the first one, needs to address the problem of either associating a cluster with a cluster number in the previous sub-sequence or generating a new one. We perform density estimation using an Epanechnikov kernel using the KD-tree implementation from [16] for every cluster using the 4-dimensional location (2D center, height and width) and 45-bin color histogram (15 for each color channel) of the regions of the proposals corresponding to each and every cluster. If the minimum KL-divergence between a distribution of the current cluster and a cluster from the previous sub-sequence is less than a threshold, we perform the cluster assignment. Otherwise, we create a new cluster.

This is to be noted that, for detecting primarily moving objects in videos, the weight of temporal edges could be as high as 0.7 or more as described in section 3.1. In such cases, considering as low as only 100 VOPs can potentially detect moving objects. Clusters may contain fewer number of proposal windows than the dimension of the original feature space which is 49-dimensional. Thus we perform PCA-based dimensionality reduction before estimating the distribution. Also, we perform scaling of features to ease the process of selecting kernel evaluation points.

5.3. Object Label Propagation.

Time-consistent clustering enables object label propagation through the video. We perform CNN-based object detection *i.e.* classification of every window in CNN feature space at a video frame only when we encounter a new cluster label. An assigned cluster label means the object category is the same as what was already detected in the associated cluster of the previous sub-sequence. We still need to perform the localization, however. In order to address the localization, we fit a 4-D Gaussian distribution on the location parameters i.e. center (x,y), height(h) and width(w), of windows in a cluster. We simply keep track of the distance, d, of the detected final object location (after first-time detection using R-CNN like approach) from the mean of the fitted Gaussian for every cluster. Furthermore, we localize the object by adding d with the mean of the 4-D Gaussian location distribution of the cluster in current sub-sequence. In general videos, new objects do not appear in every video frame. Thus, we do not need to detect objects at every video frame. Even when a new object appears, we need to detect/classify only for the proposals assigned to the new cluster. Thus, OVERLAP framework requires to process CNN features for only a small fraction of the number of proposals generated.

In some sense, the spatio-temporal clustering for object detection is related to tracking. However, a set of windows is tracked instead of a single region/object. Ability to by-pass critical tracker initialization step and the possibility for applying R-CNN like detection at a more frequent and configurable interval to increase the detection accuracy (if needed) are the major advantages.

6. Experimental Results

6.1. Video Object Detector using VOP

We observe that the proposed VOP helps in learning a better object detector model. Table 1 shows the per-class detection accuracy and the mean Average Precision (mAP) for the 10-class Youtube-Objects Test set [29].

classes	R-CNN	DPM	Fine-tune SS	Fine-tune EB	Fine-tune VOP
plane	14.1	28.42	25.57	26.52	**29.77**
bird	24.2	**48.14**	27.27	27.27	28.82
boat	16.9	25.50	27.52	33.69	**35.34**
car	27.9	**48.99**	35.18	36	41
cat	17.9	1.69	25.02	27.05	**33.7**
cow	28.6	19.24	43.01	44.76	**57.56**
dog	12.2	15.84	24.05	27.07	**34.42**
horse	29.4	35.10	41.84	44.82	**54.52**
mbike	21.3	**31.61**	26.70	27.07	29.77
train	13.2	**39.58**	20.48	24.93	29.23
mAP	20.57	29.41	29.67	31.92	**37.413**

Table 1. Object Detection Results on Youtube-Objects test set. Pre-trained R-CNN detector[11] is downloaded from [10]. Detection result using Deformable Parts Model [9] (DPM) are from [20]. Fine-tune SS uses fine-tuning with Selective Search proposals (similar to R-CNN), Fine-tune EB uses fine-tuning with Edge-Boxes proposals and Fine-tune VOP uses fine-tuning with proposed video Object Proposals ($\lambda = 0.2$).

Fine-tuning on Youtube-Objects training data with Selective Search proposals [35] improves the detection results by at least 9% compared with the model fine-tuned for the image dataset PASCAL VOC. The detector learned with EdgeBoxes performs better than the one learned with Selective Search proposals. However, the detector learned with VOP even outperforms the detection rate by another 5.5% and achieves state-of-the-art detection accuracy (37.4%) on this dataset. Although detection accuracy for "cat", "cow", "dog" and "horse" have improved by a huge margin after using CNN features, categories like "train" and "bird" are still best detected using DPM [9] detector.

6.2. Streaming Clustering of VOP

Figure 8 shows the results of frame-level clustering vs streaming clustering at sub-sequence levels on arbitrary videos downloaded from Youtube. In these experiments, a sub-sequence contains 3 video frames, with an overlap of
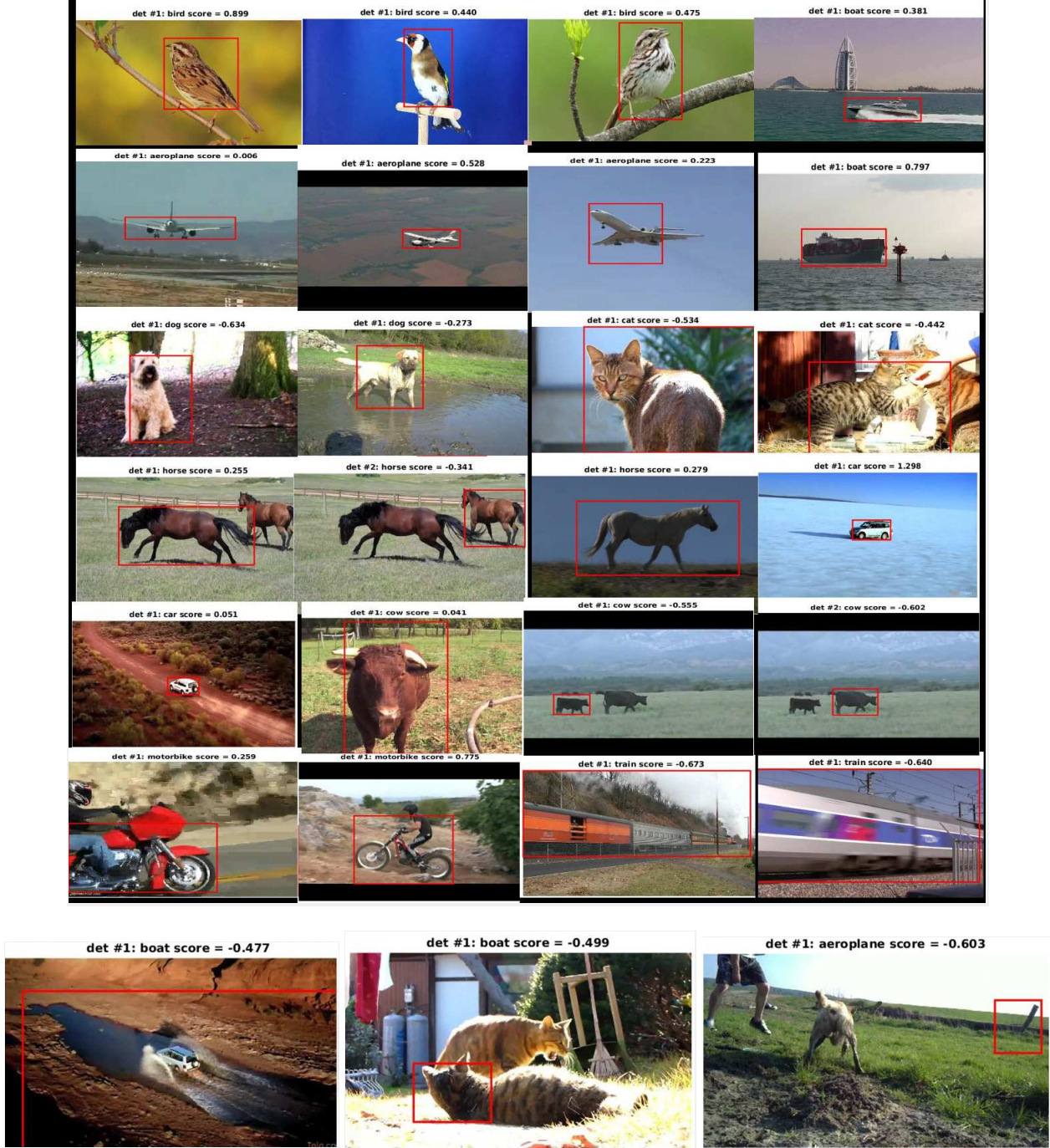
Figure 7. Sample results of Video Object Detection with VOP. First 6 rows show successful detection cases and the last row shows false detection cases.

one frame from the previous sub-sequence. We use high λ value (0.8) to identify only the moving objects with very few number of proposals. For clear visualization, we use only 50 proposal windows at every frame and that makes less than 200 proposals per sub-sequence. We aim to take the advantage of fast approximate spectral clustering al-

gorithm which scales linearly with the problem size. In our current implementation, clustering takes less than 0.1 second for 3 frames streaming-volume with 50 VOPs per frame.

	PF SS	PF EB	PF VOP		OVERLAP							
			CPU OF	GPU OF	CPU Optical Flow				GPU Optical Flow			
					200 V	500 V	1K V	2K V	200 V	500 V	1K V	2K V
Prop time	10	0.3	3.8	1.3	3.8				1.3			
Overall time	30	20.3	23.8	21.3	6.2	9.3	14.6	28.0	3.7	6.8	12.1	26.5
mAP	29.62	31.95	37.72	37.72	28.59	33.59	35.82	36.63	28.59	33.59	35.82	36.63

Table 2. Complexity and accuracy comparison. Proposal generation (Prop time), overall detection time for per-frame (PF) baseline methods with Selective Search (SS) proposals, EdgeBoxes (EB) proposals, proposed VOPs and OVERLAP are shown. Baseline methods use 2000 proposals per frame. GPU OF and CPU OF denote GPU-based and CPU-based optical flow (OF) respectively. mAP increases as number of VOPs increases. About $3\times$ speedup achived with 500 VOPs with only 4% drop in mAP compared to the baseline per-frame detection.
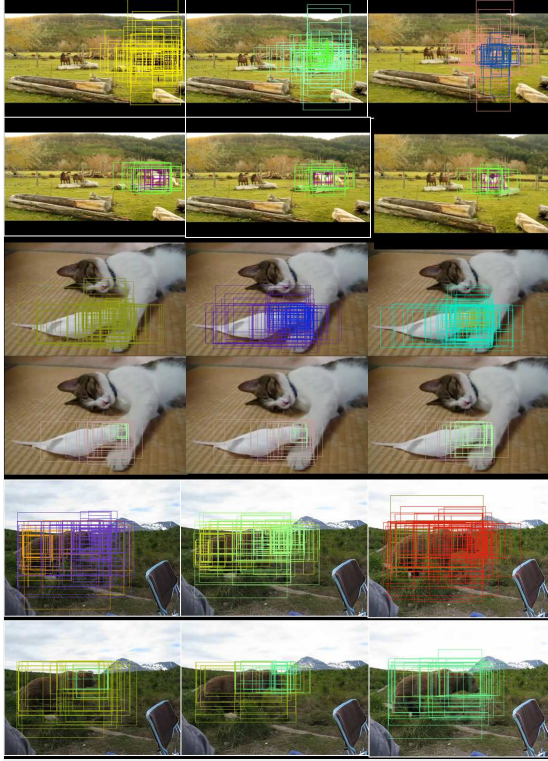

Figure 8. Temporal consistency in streaming clustering of VOPs on arbitrary videos "Horse riding", "Bird-cat" and "Alaskan bear" downloaded from the Internet. Windows drawn in same color belong to same cluster. Top rows of every pair show the proposals clustered on individual frame (frames #2, #10, #25 in each case) level; bottom row shows the results of streaming clustering.

6.3. Video Object Detection

To investigate the relative detection rate with OVERLAP compared with frame-wise R-CNN like approach, we create a subset (955 frames) of Youtube-Objects test-set (1783 frames) where the video frames form a valid video play. We find that for OVERLAP, CNN feature extraction and classification is needed only for 10% to 30% windows among all of them. Table 2 corroborates the fact that the detection accuracy improves as we increase the number of VOPs from 200 to 2000 in OVERLAP at the cost of increased complexity needed for spectral clustering. Difference in mAP is between 1-9% . As an example, compared to per-frame detection, OVERLAP achieves about $3\times$ speedup at the cost of only 4% mean Average Precision (mAP) with 500 VOPs per frame. The non-GPU based spectral clustering implementation in MATLAB makes cases for more than 2000 VOPs even slower than per-frame RCNN.

Per-frame proposal generation with Selective Search and EdgeBoxes takes about 10 seconds [14] and 0.3 seconds [14] respectively. Overall time for object detection per-frame in R-CNN per-frame becomes about 30 seconds and 20.3 seconds with the above corresponding methods. Generation of VOPs requires optical flow which takes 3.5 sec per frame in CPU-implementation and 1 sec per frame for GPU-implementation for about 500×500 resolution frame-pairs. "CPU" and "GPU" in Table 2 denote optical flow implementation in CPU [5] and GPU [33] respectively. The Youtube-Objects dataset mostly contains moving objects. In addition, the test dataset does not contain significant number of test cases where multiple instances of objects with similar appearances are spatially overlapped. Thus, accuracy of OVERLAP successfully approaches the baseline per-frame detection accuracy as we increase the number of VOPs. For the fastest detection (with 200 VOPs only), we use more weight (lambda = 0.6) for temporal edge and still manage to get acceptable detection accuracy with over $5\times$ speedup as shown in the column corresponding to 200 VOP in Table 2. GPU-based spectral clustering can potentially lead to further speed up.

7. Conclusion

Experimental results show that VOP helps in learning a better moving or static video object detector model and achieves state-of-the-art detection accuracy on Youtube-Video dataset. We show that the proposed OVERLAP framework can detect temporally consistent objects in videos through object class label propagation using streaming clustering of VOPs with significant speedup compared with naive per-frame detection with acceptable loss of accuracy. We also show that multiple objects segmentation can also be achieved as a by-product of OVERLAP.

References

[1] B. Alexe, T. Deselares, and V. Ferrari. What is an object? *CVPR*, 2010.

[2] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. *CVPR*, 2014.

[3] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. *ACM-SIAM symposium on Discrete algorithms*, 2007.

[4] S. Branson, G. V. Horn, P. Perona, and S. Belongie. Improved bird species recognition using pose normalized deep convolutional nets. *BMVC*, 2014.

[5] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *PAMI*, 2011.

[6] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. *IEEE CVPR*, 2014.

[7] P. Dollar and C. L. Zitnick. Edge boxes: Locating object proposals from edges. *ECCV*, 2014.

[8] I. Endres and D. Hoiem. Category independent object proposals. In *Proceedings of the 11th European Conference on Computer Vision: Part V*, ECCV'10, pages 575–588, Berlin, Heidelberg, 2010. Springer-Verlag.

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, Sept 2010.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. http://www.cs.berkeley.edu/ rbg/r-cnn-release1-data.tgz.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014.

[12] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[14] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *PAMI*, 2015.

[15] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? *BMVC*, 2014.

[16] A. Ihler and M. Mandel. http://www.ics.uci.edu/ ihler/code/kde.html.

[17] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Crisp boundary detection using pointwise mutual information. *ECCV*, 2014.

[18] M. Jain, J. van Gemert, H. Jegou, P. Bouthemy, and C. Snoek. Action localization with tubelets from motion. In *CVPR*, pages 740–747, June 2014.

[19] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding, 2013. http://caffe.berkeleyvision.org.

[20] V. Kalogeiton, V. Ferrari, and C. Schmid. Analysing domain shift factors between videos and images for object detection. *CoRR*, abs/1501.01186, 2015.

[21] P. Krähenbühl and V. Koltun. Geodesic object proposals. *ECCV*, 2014.

[22] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 2012.

[23] D. Kuettel, M. Guillaumin, and V. Ferrari. Segmentation propagation in imagenet. *ECCV*, 2012.

[24] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011.

[25] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, December 2013.

[26] T. Ma and L. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677, June 2012.

[27] D. Oneata, J. Revaud, J. Verbeek, , and C. Schmid. Spatio-temporal object detection proposals. *ECCV*, 2014.

[28] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstraint video. *ICCV*, 2013.

[29] A. Prest, V. Kalogeiton, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Youtube-objects dataset v2.0, 2014. University of Edinburgh (CALVIN), INRIA Grenoble (LEAR), ETH Zurich (CALVIN).

[30] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. *CVPR*, 2012.

[31] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[32] G. Sharir and T. Tuytelaars. Video object proposals. *CVPR workshop on Large-Scale Video Search and Mining*, 2012.

[33] N. Sundaram, T. Brox, and K. Keutzer. Dense Point Trajectories by GPU-accelerated Large Displacement Optical Flow. In *ECCV*, 2010.

[34] S. Tripathi, Y. Hwang, S. Belongie, and T. Q. Nguyen. Improving streaming video segmentation with early and mid-level visual processing. *WACV*, 2014.

[35] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 2013.

[36] A. Vezhnevets and V. Ferrari. Looking out of the window: object localization by joint analysis of all windows in the image. *CoRR*, abs/1501.01181, 2015.

[37] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. *ECCV*, 2012.

[38] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. *NIPS*, 2004.

[39] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, pages 628–635, June 2013.