

CenterNet Heatmap Propagation for Real-time Video Object Detection

Zhujun Xu^[0000-0002-6867-0401], Emir Hrustic, and Damien Vivet^[0000-0003-1909-5591]

ISAE-SUPAERO, Université de Toulouse, Toulouse, France
{zhujun.xu,emir.hrustic,damien.vivet}@isae.fr

Abstract. The existing methods for video object detection mainly depend on two-stage image object detectors. The fact that two-stage detectors are generally slow makes it difficult to apply in real-time scenarios. Moreover, adapting directly existing methods to a one-stage detector is inefficient or infeasible. In this work, we introduce a method based on a one-stage detector called CenterNet. We propagate the previous reliable long-term detection in the form of heatmap to boost results of upcoming image. Our method achieves the online real-time performance on ImageNet VID dataset with 76.7% mAP at 37 FPS and the offline performance 78.4% mAP at 34 FPS.

Keywords: Video object detection, Real-time, Heatmap propagation, One-stage detector

1 Introduction

Image object detection benefits a lot from the development of Convolutional Neural Networks (CNNs) over the last years. As a fundamental element of many vision tasks, such as visual surveillance, autonomous driving, etc., many CNN based structures [5, 9, 13, 8, 17, 19] have been proposed, which achieve excellent performance in still images. However, many real world applications require video object detection. When directly applying these still image detectors to a video stream, the accuracy suffers from sampled image quality problem caused by motion blur or incomplete object appearance.

Previous works [3, 7, 18, 20] have been conducted to compensate the loss by using temporal information naturally provided by videos. Most of them are developed on the base of two-stage detectors like Region-based CNN (R-CNN). On one hand, it's well known that most two-stage detectors are too slow to achieve real-time performance. On the other hand, adapting existing temporal information merging methods to one-stage detectors is challenging or even infeasible. Indeed, the representation of object bounding boxes differs a lot between these two categories of detectors and, moreover, some methods manipulate on Region of Interests (RoIs) pooled features which do not exist in one-stage structures.

In this paper, we propose a heatmap propagation method as an effective solution for video object detection. We implement our method on a one-stage

detector called CenterNet [19] which outputs a heatmap to detect the center of all objects in an image of different classes. For one frame of a video clip, we transform the stable detected objects to a propagation heatmap. In the obtained heatmap, we highlight potential positions of each object’s center with its confidence score of corresponding class. For the next frame, a balanced heatmap is generated considering both the propagation heatmap and the network output heatmap. This is similar to generate an online tracklet of each object and we update the confidence score by each frame detection result. The rest of this paper is structured as follows. Section 2 presents the object detection state of the art. Section 3 details our contribution and section 4 presents the implementation details. Finally, section 5 presents the results of our approach.

2 Related Work

2.1 Image Object Detection

One-stage and two-stage detectors Generally speaking, there are two types of state-of-the-art image object detectors, the two-stage detectors and the one-stage detectors. Two-stage detectors first use a Region Proposal Network (RPN) to detect RoIs which potentially contain an object. Then, based on pooled features inside RoIs, separated detection heads of network identify the class of object and regress the bounding box. In contrast, one stage detectors directly achieve the classification and regression on the entire feature map. Usually, sophisticated two-stage detectors are more accurate, but slower compared to one-stage detectors. However, several works [13, 14, 19] have proven that the accuracy of one-stage detectors can be competitive with even real-time speed.

Anchor based and Heatmap based detectors From the perspectives of object bounding box representation, the detectors can be divided into two groups: anchor based and heatmap based. Anchor based detectors like R-CNN [9, 8, 15], R-FCN [5] have prefixed bounding boxes at every output spatial position which are called anchor boxes. The regression values estimated by the network are the differences between anchor boxes and object real bounding boxes, typically a scale-invariant translation of the centers and a log-space translation of the width and height. [9] Heatmap based detectors like CornerNet [13], CenterNet [19], detect keypoints of object bounding box, like the vertices or the center. The network outputs heatmaps of keypoints and several regression values for offset or raw bounding box size depending on different structures. To cover as much object shapes as possible, anchor based methods will prefix different size and height/width ratio for anchor boxes. This will obviously increase output dimension which turns out to be a computational burden. Thus the compromise between robustness and speed remains hard to solve. However, the heatmap based detectors avoid such dilemma, as all regression values are raw pixel coordinate values without the need of any preset.

2.2 Video Object Detection

Considering video object detection, a trivial solution is to apply directly an image object detector on the video. However, videos usually contain moving objects or represent a motion as the camera is moving. This results in low image quality, which has undesirable influence on the detection performance. Nevertheless, videos contain temporal information, such as the consistency of the same object in consecutive frames. Using such information to compensate image quality defect is worth considering. D&T method [7] uses feature map correlation to regress bounding boxes variances of the same object across consecutive frames. A viterbi algorithm is applied to do box level association. FGFA [20] uses optical flow information predicted by a pretrained flow network [6, 12] to align and aggregate relevant features from consecutive frames. However, the pretrained optical flow network does not always generalize to new datasets. Spatiotemporal Sampling Networks (STSN) [1] uses deformable convolutions to aggregate relevant features which is more generalized. AdaScale [3] reshapes the image size to an adaptive resolution which produces better accuracy and speed. The fact that most of these works use a two-stage anchor based detector makes it hard to achieve real-time detection. Scale-Time Lattice method [2] proposes a framework of temporal propagation and spatial refinement to extend the detection results on sparse key frames to dense video frames. Our method can also be integrated into this framework. Our approach is inspired by [18], where the author proposed a method of re-scoring tracklet to improve single frame detection. Still it is implemented with R-CNN detectors.

3 Proposed Method

3.1 Background: CenterNet

CenterNet [19] is a one-stage heatmap based object detector. The principle of this method is to predict the position of the center and the size of objects in images. Given an input RGB image of width w and height h , $I \in R^{w \times h \times 3}$, the network outputs a downsampled heatmap $\hat{Y} \in [0, 1]^{\frac{w}{R} \times \frac{h}{R} \times C}$, where R is output stride and C is the number of classes. We note $W = \frac{w}{R}$, $H = \frac{h}{R}$ as the output spatial size. A prediction $\hat{Y}_{x,y,c} = 1$ corresponds to the center of an object of class c at position (x, y) , while $\hat{Y}_{x,y,c} = 0$ corresponds to background. In addition, the network predicts a local offset $\hat{O} \in R^{W \times H \times 2}$ to recover the discretization error by the output stride and a regression $\hat{S} \in R^{W \times H \times 2}$ for object size.

As shown in Fig. 1, the entire network contains 3 components. A general convolutional network, N_{feat} , like ResNet [11] extracts feature maps from input image. A deconvolutional network, N_{decv} , is built of 3×3 deformable convolutional layers (DCLs) [4] and up-convolutional layers. It refines the feature maps into output spatial scale. Finally, 3 separate heads, N_{head} , share the same backbone feature maps and output \hat{Y} , \hat{O} and \hat{S} . Due to computational cost, all classes share the same prediction of offset and object size. So, the final output size of network is $W \times H \times (C + 4)$.

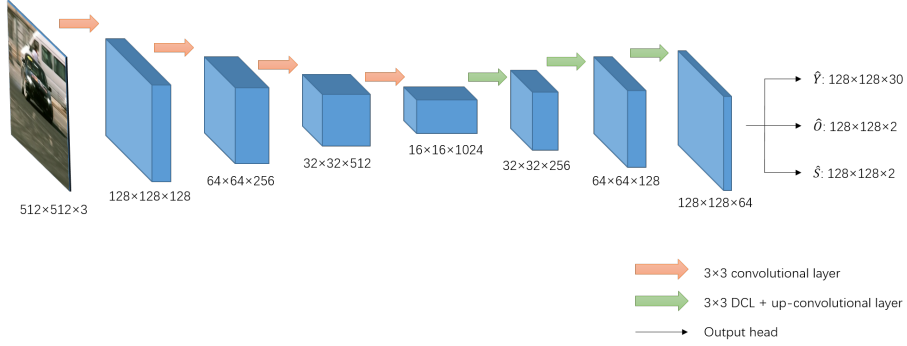


Fig. 1. CenterNet with ResNet-101 backbones. Orange arrows represent N_{feat} . Green arrows represent N_{decv} . Black line arrows represent 3 separate heads N_{head} .

At inference time, by applying a 3×3 max pooling operation, all peaks whose value is greater or equal to its 8 neighbors in the heatmap of each class will be extracted from \hat{Y} . Only the top 100 peaks will be kept. For a peak at position (\hat{x}_i, \hat{y}_i) , we use offset prediction $\hat{O}_{x_i, y_i} = (\delta\hat{x}_i, \delta\hat{y}_i)$ and size prediction $\hat{S}_{x_i, y_i} = (\hat{w}_i, \hat{h}_i)$ to produce the bounding box of object as:

$$\begin{aligned} &(\hat{x}_i + \delta\hat{x}_i - \hat{w}_i/2, \quad \hat{y}_i + \delta\hat{y}_i - \hat{h}_i/2, \\ &\hat{x}_i + \delta\hat{x}_i + \hat{w}_i/2, \quad \hat{y}_i + \delta\hat{y}_i + \hat{h}_i/2). \end{aligned} \quad (1)$$

We refer the reader to the original work of CenterNet [19] for further detail.

3.2 Heatmap Propagation

Still-image object detectors like CenterNet are very effective to process static images. However, due to quality problems of sampled images like blurring or object occlusion, such detectors may produce unstable results when directly applied to consecutive images of video clips. We propose an online real-time method heatmap propagation (HP) for video object detection by propagating previous long-term stable detection results to upcoming image.

Let $D^t = \{d_i^t\}_{i=1}^m$ be the set of m predicted objects in frame t of a video. For each object d_i^t detected at time t , we count the number of consecutive frames up to frame t where the object appears and define the number as tracklet length l_i^t . For a new object detected in frame t , $l_i^t = 1$. We also have the predicted bounding box size $s_i^t = (w_i^t, h_i^t)$ for each object.

As presented in Section 3.1, each predicted object is generated by one peak (\hat{x}_i, \hat{y}_i) of class c_i in output heatmap. To propagate result of frame t to frame $t+1$, each peak value $\hat{Y}_{\hat{x}_i \hat{y}_i c_i}^t$ is dilated with a square kernel of size $2P+1$ resulting in $(2P+1)^2 - 1$ positions. As shown in Fig. 2 (a) and (b), this produces an

extended heatmap $H_i^t = \{h_{i,xyz}^t\} \in [0, 1]^{W \times H \times C}$ where:

$$h_{i,xyz}^t = \begin{cases} \hat{Y}_{\hat{x}_i \hat{y}_i c_i} & \text{if } c = c_i, |x - \hat{x}_i| \leq P, |y - \hat{y}_i| \leq P, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

We overlap all \mathbf{m} extended heatmaps into one propagation heatmap by keeping the maximum value in each position and class: $\bar{H} = \{\bar{h}_{xyz}\} \in [0, 1]^{W \times H \times C}$ where $\bar{h}_{xyz} = \max_{i \in [1, m]} (h_{i,xyz}^t)$ (See Fig. 2 (b) and (c)). Although occlusion of objects may exist, the centers of objects are rarely located at the same point. Thus keeping the maximum value remains an effective way to collect all detection results. The propagation heatmap will inherit tracklet length from frame t , $\bar{L} = \{\bar{l}_{xyz}\}$ where:

$$\bar{l}_{xyz} = \begin{cases} l_i^t & \text{where } i = \arg \max_{i \in [1, m]} (h_{i,xyz}^t) \text{ if } \bar{h}_{xyz} > 0 \\ 0 & \text{if } \bar{h}_{xyz} = 0 \end{cases} \quad (3)$$

Similarly, the bounding box size information will also be inherited, $\bar{S} = \{\bar{s}_{xyz}\} \in R^{W \times H \times C \times 2}$ where:

$$\bar{s}_{xyz} = \begin{cases} s_i^t & \text{where } i = \arg \max_{i \in [1, m]} (h_{i,xyz}^t) \text{ if } \bar{h}_{xyz} > 0 \\ 0 & \text{if } \bar{h}_{xyz} = 0 \end{cases} \quad (4)$$

We combine the network output heatmap of frame $t+1$: $\hat{Y}^{t+1} = \{\hat{Y}_{xyz}^{t+1}\} \in [0, 1]^{W \times H \times C}$ and propagation heatmap from frame t : $\bar{H} = \{\bar{h}_{xyz}\}$ to a long-term heatmap in the following way:

$$\bar{Y}_{xyz}^{t+1} = \frac{\hat{Y}_{xyz}^{t+1} + \beta \bar{h}_{xyz} \bar{l}_{xyz}}{1 + \beta \bar{l}_{xyz}}, \quad (5)$$

where β is a confidence parameter for long-term detection ($\beta = 2$ by default). Equation 5 serves as a temporal average with update prediction \hat{Y}_{xyz}^{t+1} . To be robust to large variance in image, we set the final heatmap as a balance between the long-term heatmap and instant detection heatmap of network:

$$\tilde{Y}_{xyz}^{t+1} = (1 - \alpha) \hat{Y}_{xyz}^{t+1} + \alpha \bar{Y}_{xyz}^{t+1} \quad (6)$$

where α is a balance parameter ($\alpha = 0.98$ by default). The necessity of this balance equation will be analysed in Section 5.3. These 2 steps are shown in Fig. 2 (c),(d) and (e).

For bounding box prediction, we calculate a weighted size by combining propagated size information \bar{S} and network output size $\hat{S} = \{\hat{S}_{xy}\} \in R^{W \times H \times 2}$ based on \bar{H} and \hat{H}^{t+1} :

$$\tilde{S}_{xyz}^{t+1} = \frac{\hat{Y}_{xyz}^{t+1} \hat{S}_{xy}^{t+1} + \bar{h}_{xyz} \bar{s}_{xyz}}{\hat{Y}_{xyz}^{t+1} + \bar{h}_{xyz}}. \quad (7)$$

To be clear, we use class-agnostic box size prediction in network. To propagate the size information with class-specific heatmap, we broadcast the same size prediction in all class channel for Equation 7. Unlike Equation 5, we don't involve tracklet length here, because the size of object projection in image may change during time due to the relative movement between camera and object. Thus we only use the previous and the current estimation.

Finally, we apply the same procedure as CenterNet on the balanced heatmap to produce detection in frame $t+1$, $D^{t+1} = \{d_j^{t+1}\}_{j=1}^n$. We update the tracklet length in the following way: $l_j^{t+1} = \bar{l}_{x_j y_j c_j} + 1$, where (x_j, y_j) is the center position of d_j^{t+1} and c_j is its class.

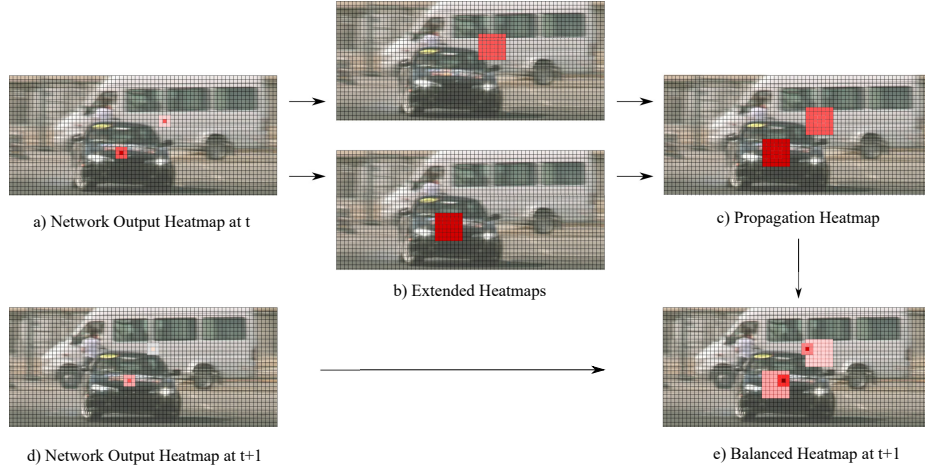


Fig. 2. Illustration of HP (Heatmap Propagation). In a), two cars are detected with high scores at frame t . In d), the detection scores become lower due to image quality at frame $t+1$. In e), after the HP operation, response at relative positions of heatmap has been enhanced. Detection with higher scores can be extracted.

4 Implementation Details

4.1 Architecture

CenterNet In this work, we use ResNet-101 as feature extraction network, N_{feat} , for the purpose of fair comparison with other methods. Following the same structure as original paper, the N_{decv} is built by 3 upsampling layers with 256, 128, 64 channel, respectively. One 3×3 deformable convolutional layer is added before each up-convolution with channel 256, 128, 64, respectively. Each output head is built by a 3×3 convolutional layer with 64 channel followed by a 1×1 convolution with corresponding channel (C for \hat{Y} or 2 for \hat{O} and \hat{Y}) to generate desired output.

Heatmap Propagation A peak point extraction is proposed in [19] to take the place of Non-Maximum Suppression (NMS). However, we still observe a slight performance improvement when applying NMS after the extraction. The dilatation of heatmap is efficiently implemented by a $(2P + 1) \times (2P + 1)$ max pooling layer. The peak extraction operation in CenterNet is implemented by a 3×3 max pooling layer. We notice that this may produce false detection at the edge of dilated square as the extraction is too local. A bigger kernel of size $(2P + 1) \times (2P + 1)$ is used in our case and provides better performances than the original one.

Seq-NMS Seq-NMS [10] is an effective off-line post-processing to boost scores of the weak detection in video. In original work, this method is applied on all proposals. In our case, the CenterNet only keeps the top 100 peaks and we ignore all bounding boxes with a score under 0.05. For each frame, we apply a very limited number of bounding boxes on Seq-NMS, which makes it faster.

4.2 Dataset

We use ImageNet [16] object detection from video (VID) dataset to evaluate our method. The dataset has 3862 training and 550 validation video with framerate at 25 - 30 fps. There are 30 classes of moving objects, which are a subset of 200 classes in ImageNet object detection (DET) dataset. Following the protocol in [7, 20, 18], we train the network on an intersection of ImageNet DET and VID dataset by sampling at most 2K images per class (only using 30 VID classes of moving objects) from DET set and 10 frames of each video from VID set. As the test annotation is not publicly available, we measure the performance of our method by calculating the mean average precision (mAP) on the validation set.

4.3 Training and Inference

For both training and inference, we resize all input image to 512×512 with zero padding for non-square shape images. With the output stride $R = 4$, the output resolution is 128×128 . Most two-stage detectors resize input image to a shorter side of 600 pixels. We don't use this configuration in our work for two reasons. Firstly, the output heatmap size of our CenterNet is proportional to input resolution. A larger input resolution will increase the runtime throughout the entire network. This is different from two-stage detectors, where the increase of resolution only brings extra burden to the part before the Region Proposal Network. The runtime after RoIPooling is only affected by the number of RoIs. Actually, if we use the larger input size, our method's runtime will be about 27 FPS, which is slightly below 30 FPS real-time criteria. Secondly, when using larger resolution, we noticed a slight decrease in AP_{50} with dataset ImageNet VID. This is also analyzed in original CenterNet paper, where they marked 0.3 decrease in AP_{50} with dataset MS coco. Even though they marked a 0.9

increase in AP_{75} , and a 0.1 increase in comprehensive AP , the official evaluation of ImageNet VID is AP_{50} .

During training for CenterNet, random flip, random scaling from 0.6 to 1.4 are used as data augmentation and SGD is used as optimizer. We train the whole network with a batch-size of 32 (on 2 GPUs) and a learning rate of 10^{-4} for 50 epochs followed by a learning rate of 10^{-5} for 30 epochs.

For inference, no augmentation is applied. We use the full validation set for experiments. We apply NMS with IoU threshold of 0.3.

5 Experiments

5.1 Quantitative Result

We show the results for our methods and state-of-the-art in Table 1. All results are conducted with ResNet-101 as backbones. First, we compare our method with the baseline CenterNet. Our method improves 3.1% mAP with an extra 6ms runtime. We also compare results after Seq-NMS to prove the effectiveness of our method and a 2.5% mAP improvement is observed. In both two case, our method maintains a real-time performance. Even combined with Seq-NMS, our method works at 34 FPS. Next, we compare with the state-of-the-art and our method achieves competitive accuracy. However, most of the two-stage based detectors can not achieve the 25-30 FPS standard of ImageNet VID dataset. The AdaScale works at 21 FPS. Our method achieves better accuracy with 1.7 times faster runtime. The Scale-Time Lattice framework reaches the real-time performance by doing sparse key frame detection. Their base detector is still a Faster R-CNN. Our method can be integrated into this frame for a better tradeoff.

We also conduct experiments with DLA-34 as backbones. Unlike original work of CenterNet, the DLA-34 baseline only achieves a mAP of 69.1% in our test. Nevertheless, our method raises the mAP to 71.3%. We believe this is relative to the dataset, as MS coco has more classes and images are clearer.

In purpose of fair comparison, we conduct a simple interpolation between CenterNet’s detection outputs of consecutive frames. However, we don’t see a remarkable improvement in performance. Actually, this is also a special case of our method. We set $P = 0$. This makes the extended heatmap for one object become one point. So the overlap of all extended heatmaps becomes a simple superposition. If there is a detected object at position (x,y,c) in frame t, we have:

$$\bar{h}_{xyc} = \max_{i \in [1, m]} (h_{i, xyc}^t) = \hat{Y}_{xyc}^t \quad (8)$$

Set $\beta \rightarrow \infty$, Equation 5 becomes:

$$\bar{Y}_{xyc}^{t+1} = \bar{h}_{xyc} \quad (9)$$

Set $\alpha = 0.5$, Equation 6 becomes:

$$\tilde{Y}_{xyc}^{t+1} = 0.5(\hat{Y}_{xyc}^{t+1} + \bar{Y}_{xyc}^{t+1}) \quad (10)$$

Finally, we have:

$$\tilde{Y}_{yc}^{t+1} = 0.5(\hat{Y}_{yc}^{t+1} + \hat{Y}_{yc}^t) \quad (11)$$

Table 1. Performance comparison on the ImageNet VID validation set. For methods marked with -, the runtime is not provided in the original papers. With the fact that these methods add computational components on two-stage detectors baseline, the runtime shall be longer than that of R-FCN.

Methods	$mAP_{50}(\%)$	Runtime(ms)	FPS
R-FCN [3]	74.2	75	13
AdaScale [3]	75.6	47	21
FGFA/+Seq-NMS [20]	76.3/78.4	733/873	1.2/1.1
D&T/+Viterbi [7]	75.8/79.8	141/187	6.8/5.5
STSN/+Seq-NMS [1]	78.9/80.4	-	-
Tracklet-Conditioned/+FGFA [18]	79.4/83.5	-	-
Scale-Time Lattice [2]	79/79.8	16.1/200	62/5
CenterNet/+Seq-NMS	73.6/75.9	21/23	47/43
CenterNet+interpolation/+Seq-NMS	73.6/75.9	22/24	45/42
CenterNet+HP/+Seq-NMS	76.7/78.4	27/29	37/34

5.2 Qualitative Result

We also conduct qualitative experiments to better explain the mechanism of our method. Fig. 3 shows some typical examples where our HP method improves detection result compared to baseline CenterNet. One typical case is that the still image detector can miss some detection in certain sample images of a video clip. This usually happens to non consecutive images. The HP can easily boost the score at missing position if the previous image includes a long-term stable detection. This largely solves the problem of transient object lost in long video. Another typical case is that the still image detector can generate inferior bounding boxes when the object blends into background. This happens frequently with motion blur which makes it difficult to distinguish between the textures of the object and the background. Our method assumes small displacement of the object’s center and maintain a smooth variation of bounding box size along the time as we calculate the weighted size with the previous bounding box of the same object. This helps to correct the mentioned problem and maintain stable detection in the aspect of bounding box size.

In Fig. 4, we compare the evolution of detection scores by baseline and our method. In both video clips, our method recovers cliff drops in several frames (e.g. frame 60, 80-100, 120-140 in video 1. frame 110-160, 240-300 in video 2) and the result is much smoother. This means our method gives less false positives than vanilla baseline. Thus when we apply off-line process like Seq-NMS, the

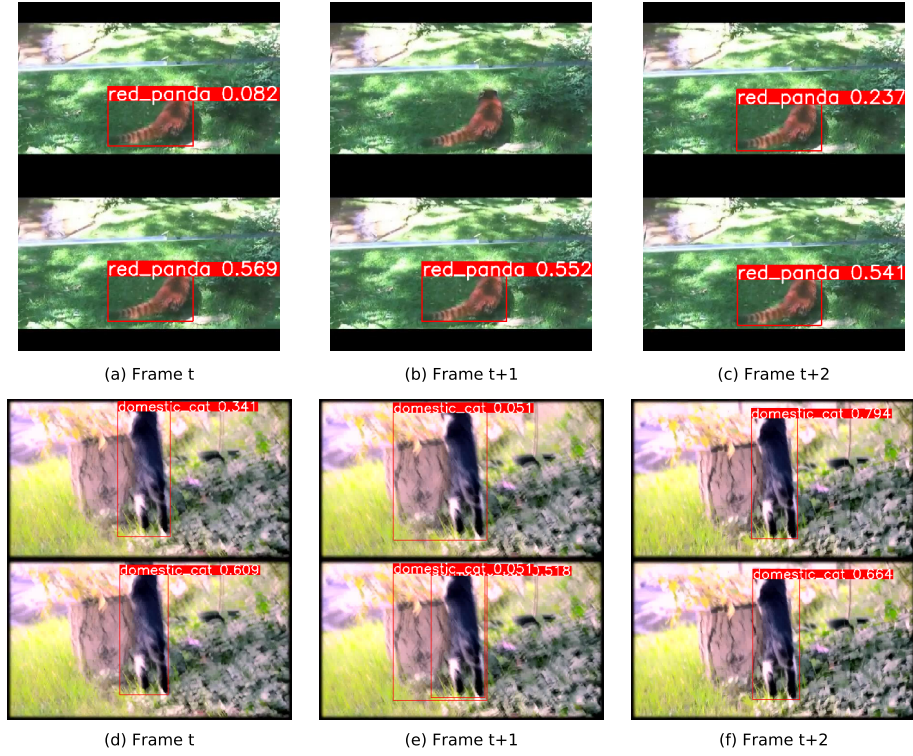


Fig. 3. Examples of qualitative result. For each of the 6 images, the upper part is CenterNet result, the lower part is result after HP method. Images (a),(b),(c) is a scenario where the target object is transiently lost. In frame t+1, the network output score is below the detection threshold 0.05. In frame t+2, the object is detect again. With HP, we can keep detecting the object with better confidence. Images (d),(e),(f) is a scenario where the object blends into background. In frame t+1, the detected center is totally biased. After HP, we maintain the correct center with detection in frame t. Although the score drops from 0.609 to 0.518, it rises back to 0.664 as frame t+2 has a clear detection. Different from a tremendous score variance (0.341 - 0.051 - 0.794), we keep a stable detection with better bounding boxes.

average of detection scores turns out to be more reliable. More stable detection is also preferred in many further vision applications like visual object tracking.

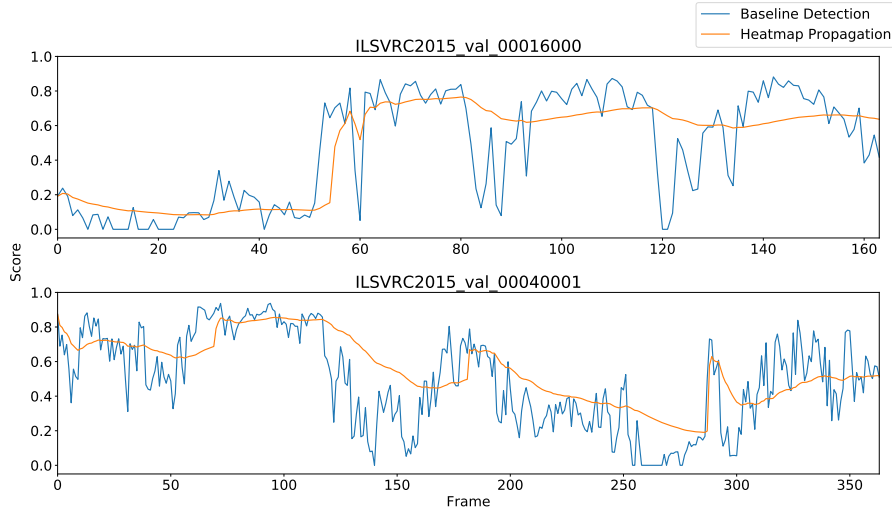


Fig. 4. Evolution of detection scores of 2 video clips. Green lines are CenterNet results. Red lines are results with HP. In both video, our method produces the smoother line, which means the more stable detection.

5.3 Ablation Study

Extended Heatmap Size P In our method, each detected object in output heatmap will be extended into a square with side length $2P+1$. As we increase P , one tracklet will boost a boarder space in final heatmap, which turns out to be more robust to fast movement. However, this may create some false positives (FP). A typical example: due to image quality, the baseline detector makes a correct detection with a high score and a parasitic detection (biased center) with low score for the same object. Usually, this weak FP will disappear rapidly. If P is too large, the precise detection in previous frame will boost both the correct detection and the parasitic detection in the current frame. So that the FP will last for longer. We calculate the maximum displacement of the center in output resolution $\Delta_i = \max(|x_i^{t+1} - x_i^t|, |y_i^{t+1} - y_i^t|)$ for each object pair in ImageNet VID validation set. One object pair means that the same object appears in two consecutive frames. There are 272038 ground-truth (GT) pairs in the whole set. 1220 pairs among them have a Δ_i greater than 5, which represent 0.45% cases. Thus we test P from 0 to 5, and results are shown in Table 2.

For more detail of our method’s performance concerning this parameter, we break down the results into 3 motion speeds, based on whether the ground-truth

Table 2. Ablation study of extended heatmap size P. Uncovered GT pairs stands for the proportion of GT pairs with $\Delta_i > P$.

P (default marked by *)	0	1	2*	3	4	5
Uncovered GT pairs (%)	60.4	11.3	3.7	1.6	0.79	0.45
$mAP_{50}(\%)$	75	76.5	76.7	76.4	75.8	75.5

object pair’s motion is slow ($\Delta_i < 2$), median ($2 \leq \Delta_i < 4$), or fast ($\Delta_i \geq 4$). Some classes usually have fast motion, while others have the opposite, so we use the class-agnostic precision ($\frac{TP}{TP+FP}$) vs recall ($\frac{TP}{TP+FN}$) to better display the results. As shown in Fig. 5, small value of P has better performance in slow motion and vice versa. This is consistent with our previous explanation. The default value of P in this paper is adapted to general scenes. For applications with high-speed scenes, a greater value of P is favorable. Besides, our method surpasses the baseline (and the simple interpolation version) in all 3 subsets. Table 3 presents a direct number comparison between baseline and our method.

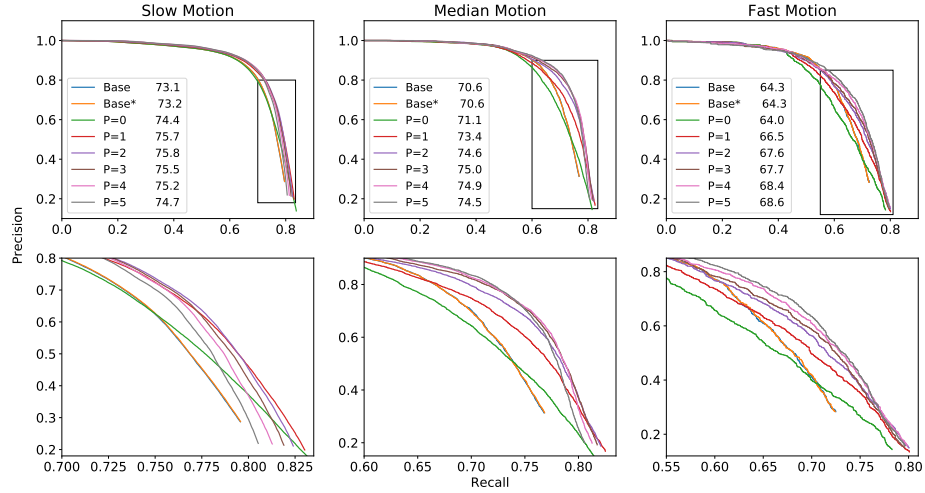
**Fig. 5.** Precision vs recall of different object pair motions. We add a zoom-in version below each figure which provides a better view of content in the black square. Each legend is followed by the corresponding average precision. Base stands for CenterNet and Base* stands for the simple interpolation version.

Table 3. Numerical comparison of 75% recall and 65% recall. Compared values are shown in the form of “baseline/our method in default setting”. Our method outperforms the baseline with both recall configurations in all 3 subsets and the whole valid set.

		GT	FN	FP	Precision (%)
75% Recall	Slow	254190	63548	114741/67930	62.4/73.7
	Median	14820	3705	14499/5219	43.4/68.0
	Fast	3028	757	5473/3734	29.3/37.8
	Overall	272038	68010	131973/75648	60.7/73.0
65% Recall	Slow	254190	88967	22199/17797	88.2/90.3
	Median	14820	5187	1891/1504	83.6/86.5
	Fast	3028	1060	1264/911	60.9/68.4
	Overall	272038	95214	24560/19814	87.8/89.9

Parameter α and β We also investigate the influence of the parameter α and β in Equation 6 and Equation 5, respectively. The results are shown in Table 4 and Table 5.

If we only use Equation 5 in our design ($\alpha = 1$), we observe a remarkable decrease of accuracy. This is the case where we are completely confident with long-term heatmap. This can produce catastrophic results when an object with long length tracklet has one large spatial variance. The HP may create a false detection near previous center and it will last for a long time. Another typical example is when a long tracked object quits the camera’s field of view. In that case, the HP will keep boosting the last position for a long time. For the above reasons, the balance operation with parameter α is necessary.

Different from α which has a linear effect on the heatmap, β imports a non-linear effect with the tracklet length. With greater value of β , we put more confidence on the long-term part in update score. If we only use Equation 6 in our design ($\beta \rightarrow \infty$), we face a contradiction in the choice of α . If α is not large enough, we can hardly recover the cliff fall of score, which is a typical case due to image quality problem. For example, $\alpha = 0.5$, we track an object with score of 0.9 for 20 frames, and the baseline output score drops to 0.1 at 21st and 22nd frame. Equation 6 alone can only boost the score to 0.5 and 0.3 at 21st and 22nd frame. However, by weighing the average by the tracklet length, Equation 5 (with $\beta = 2$) gives 0.88 and 0.86, respectively. To achieve a similar effect with Equation 6 alone, we need to set $\alpha \geq 0.97$. In that case, a temporary false positive (FP) detection will last too long. E.g. we have a FP with score of 0.8 for 1 frame, the score is below 0.1 for the following frames. Our method will attenuate the score to 0.4 in 3 frames, but Equation 6 ($\alpha = 0.97$) alone needs 27 frames.

Table 4. Ablation study of parameters α

α (default marked by *)	0.5	0.6	0.7	0.8	0.9	0.95	0.98*	1
$mAP_{50}(\%)$	74.6	74.9	75.1	75.3	76.1	76.3	76.7	75.3

Table 5. Ablation study of parameters β

β (default marked by *)	0.5	1	2*	5	10	100	1000
$mAP_{50}(\%)$	75.7	76.4	76.7	76.4	76	75.1	75

6 Conclusion

In this paper, we introduce a real-time video object detection method Heatmap Propagation based on CenterNet. Compared with state-of-the-art methods which are mainly based on two-stage detectors and far from real-time performance, our method achieves competitive results with real-time speed. Compared with our baseline CenterNet, our method achieves better accuracy with only 6ms extra runtime per frame and produces smoother and more stable results for further applications. Our future work will include experiments of Heatmap Propagation on other object detection approaches or semantic segmentation for video.

Acknowledgements This work was supported by the Agence Nationale de la Recherche (ANR-the French national research agency) (ANR-17-CE22-0001-01) and by the French FUI (FUI STAR: DOS0075476 00)

References

1. Bertasius, G., Torresani, L., Shi, J.: Object detection in video with spatiotemporal sampling networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 342–357. Springer International Publishing, Cham (2018)
2. Chen, K., Wang, J., Yang, S., Zhang, X., Xiong, Y., Loy, C.C., Lin, D.: Optimizing video object detection via a scale-time lattice. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7814–7823 (2018)
3. Chin, T.W., Ding, R., Marculescu, D.: AdaScale: Towards Real-time Video Object Detection Using Adaptive Scaling. arXiv preprint arXiv:1902.02910 (Feb 2019)
4. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 764–773 (2017)
5. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp.

- 379–387. Curran Associates, Inc. (2016), <http://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf>
6. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazirbas, C., Golkov, V., v. d. Smagt, P., Cremers, D., Brox, T.: FlowNet: Learning optical flow with convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 2758–2766 (2015)
7. Feichtenhofer, C., Pinz, A., Zisserman, A.: Detect to track and track to detect. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 3057–3065 (2017)
8. Girshick, R.: Fast r-cnn. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 1440–1448 (2015)
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587 (2014)
10. Han, W., Khorrani, P., Le Paine, T., Ramachandran, P., Babaeizadeh, M., Shi, H., Li, J., Yan, S., Huang, T.S.: Seq-NMS for Video Object Detection. arXiv preprint arXiv:1602.08465 (Feb 2016)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
12. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1647–1655 (2017)
13. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 765–781. Springer International Publishing, Cham (2018)
14. Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2999–3007 (2017)
15. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(6), 1137–1149 (2017)
16. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.F.: Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* **115**, 211–252 (2014)
17. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9626–9635 (2019)
18. Zhang, Z., Cheng, D., Zhu, X., Lin, S., Dai, J.: Integrated Object Detection and Tracking with Tracklet-Conditioned Detection. arXiv preprint arXiv:1811.11167 (Nov 2018)
19. Zhou, X., Wang, D., Krähenbühl, P.: Objects as Points. arXiv preprint arXiv:1904.07850 (Apr 2019)
20. Zhu, X., Wang, Y., Dai, J., Yuan, L., Wei, Y.: Flow-guided feature aggregation for video object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 408–417 (2017)