# Fully Motion-Aware Network for Video Object Detection

Shiyao Wang[1], Yucong Zhou[2], Junjie Yan[2], and Zhidong Deng[1]

[1] State Key Laboratory of Intelligent Technology and Systems
Beijing National Research Center for Information Science and Technology
Department of Computer Science, Tsinghua University, Beijing 100084, China
[2] SenseTime Research Institute
sy-wang14@mails.tsinghua.edu.cn, zhouyucong@sensetime.com
yanjunjie@sensetime.com, michael@tsinghua.edu.cn

**Abstract.** Video objection detection is challenging in the presence of appearance deterioration in certain video frames. One of typical solutions is to enhance per-frame features through aggregating neighboring frames. But the features of objects are usually not spatially calibrated across frames due to motion from object and camera. In this paper, we propose an end-to-end model called fully motion-aware network (MANet), which jointly calibrates the features of objects on both pixel-level and instance-level in a unified framework. The pixel-level calibration is flexible in modeling detailed motion while the instance-level calibration captures more global motion cues in order to be robust to occlusion. To our best knowledge, MANet is the first work that can jointly train the two modules and dynamically combine them according to the motion patterns. It achieves leading performance on the large-scale ImageNet VID dataset.

**Keywords:** Video Object Detection · Feature Calibration · Pixel-level · Instance-level · End-to-end

## 1 Introduction

Object detection is a fundamental problem in image understanding. Deep convolutional neural networks have been successfully applied to this task, including [22, 2, 20, 21, 18, 19, 29]. Although they have achieved great success in object detection from static image, video object detection remains a challenging problem. Frames in videos are usually deteriorated by motion blur or video defocus, which are extremely difficult for single-frame detectors.

To tackle the challenges in deteriorated frames, one of straightforward solutions is to consider the spatial and temporal coherence in videos and leverage information from nearby frames. Following this idea, [8, 15, 14, 5] explore hand-crafted bounding box association rules to refine the final detection results. As post-processing methods, those rules are not jointly optimized. As contrast, FGFA [30] attempts to leverage temporal coherence on feature level by aggregating features of nearby frames along the motion paths. They use flow estimation

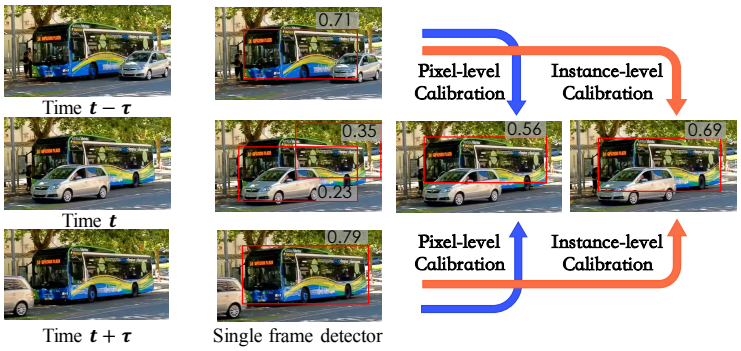



**Fig. 1.** Examples of occlusion in video object detection. When the bus is occluded by a passing car, the single frame detector fails to produce an accurate box. Pixel-level calibration can help improve the results but it is still influenced due to occlusions. Instance-level calibration performs the best among these results.

to predict per-pixel motion which is hereinafter referred to as pixel-level feature calibration. However, such pixel-level feature calibration approach would be inaccurate when appearance of objects dramatically changes, especially as objects are occluded. With inaccurate flow estimation, the flow-guided warping may undesirably mislead the feature calibration, failing to produce ideal results. Thus, the robustness of feature calibration is of great importance.

In this paper, our philosophy is that accurate and robust feature calibration across frames play an important role in video object detection. Besides existing pixel-level methods, we propose an instance-level feature calibration method. It estimates the motion of each object along time in order to accurately aggregate features. Specifically, for each proposal in the reference frame, the corresponding motion features are extracted to predict the relative movements between nearby frames and the current frame. According to the predicted relative movements, the features of the same object in nearby frames are RoI-pooled and aggregated for better representation. Compared to the pixel-level calibration, the instance-level calibration is more robust to large temporal appearance variations such as occlusions. As shown in Figure 1, when the bus in the reference frame is occluded, the flow estimation fails to predict such detailed motion. The warped features of nearby frames can be used to improve the current result, but they are still affected by occluded pixels. In contrast to the pixel-level calibration, the instance-level calibration considers an object as a whole and estimate the motion of the entire object. We argue that such high-level motion is more reliable to use especially when the object is occluded.

Moreover, taking a closer look at above two calibrations, we find the pixel-level and instance-level calibration can work collaboratively depending on different motion patterns. The former one is more flexible for modeling non-rigid motion, particularly for some tiny animals. And high-level motion estimation can

well describe regular motion trajectory (*e.g.* car). On the basis of observation, we develop a motion pattern reasoning module. If the motion pattern is more likely to be non-rigid and any occlusion does not occur, the final result relies more on the pixel-level calibration. Otherwise, it depends more on the instance-level calibration. All above modules are integrated in a unified framework that can be trained end-to-end.

In terms of the baseline model R-FCN, the proposed instance-level calibration and the MANet improve the mAP 3.5% and 4.5%, respectively, on ImageNet VID dataset.

In summary, the contributions of this paper include:

- We propose an instance-level feature calibration method by learning instance movements through time. The instance-level calibration is more robust to occlusions and outperforms pixel-level feature calibration.

- By visualizing typical samples and conducting statistical experiments, we develop a motion pattern reasoning module to dynamically combine pixel-level and instance-level calibration according to the motion. We show how to jointly train them in an end-to-end manner.

- We demonstrate the MANet on the large-scale ImageNet VID dataset [23] with state-of-the-art performance. Our code is available at: `https://github.com/wangshy31/MANet_for_Video_Object_Detection.git`.

## 2 Related Work

### 2.1 Object Detection from Still Images

Existing state-of-the-art methods for general object detection are mainly based on deep CNNs [16, 25, 27, 10, 26, 11, 1]. Based on such powerful networks, a lot of works[7, 6, 22, 2, 18, 3, 24] have been done for further improvement in performance of detection. [7] is a typical proposal based CNN detector by using Selective Search [28] to extract proposals. Different from the above multi-stage pipeline, [6] develops an end-to-end training method though applying spatial pyramid pooling [9]. Faster R-CNN [22] further incorporates proposal generation procedure into CNNs with most parameters shared, leading to much higher proposal quality as well as computation speed. R-FCN [2] is another fully convolutional detector.To address the lack of position sensitivity, R-FCN introduces position-sensitive score maps and a position-sensitive RoI pooling layer.We use R-FCN as our baseline and further extend it for video object detection.

### 2.2 Object Detection in Videos

Unlike those methods of object detection in still images, detectors for videos should take the temporal information into account. One of the main-stream approaches aims to explore bounding box association rules and apply heuristic post-processing. And the other stream of previous work is to leverage temporal coherence on feature level and seek to improve the detection quality in a principled way.

**For post-processing**, the main idea is to use high-scoring objects from nearby frames to boost scores of weaker detections within the same video. The major difference among these methods is the mapping strategy of linking still image detections to cross-frame box sequences. [8] links cross-frame bounding boxes iff their IoU is beyond a certain threshold and generate potential linkages across the entire clip. Then they propose a heuristic method for re-ranking bounding boxes called "Seq-NMS" . [14, 15] focus on tubelet rescoring. Tubelets are bounding boxes of an object over time. They apply an offline tracker to revisit the detection results and then associate still-image object detections around the tubelets. [15] presents a re-scoring method to improve the tubelets in terms of temporal consistency. Moreover, [14] proposes multi-context suppression (MCS) to suppress false positive detections and motion-guided propagation (MGP) to recover false negatives. D&T [5] is the first work to joint learn ROI tracker along with detector. The cross-frame tracker is used to boost the scores for positive boxes. All above approaches focus on post-processing that can be further collaborated with feature-level methods. We will prove it by combining Seq-NMS [8] with our model to reinforce each other and further improve performance.

**For feature-level learning**, [31, 30, 13] propose end-to-end learning frameworks to enhance the feature of individual frames in videos. [30] presents flow-guided feature aggregation to leverage temporal coherence on feature level. In order to spatially calibrate the features across frames, they apply an optical flow network [4] to estimate the per-pixel motion between the nearby frames and the reference frame. All the feature maps from nearby frames are then warped to the reference frame so as to enhance the current representations. Similar to this work, [31] also utilizes an optical flow network to model the correspondences in raw pixels. The difference is that they use it to achieve significant speedup. However, the low-level motion prediction is lack of robustness especially in the presence of occlusion [12]. Such individual pixel-wise prediction without considering context may suffer from local consistency [17]. Different from still image proposals, [13] provides a novel tubelet proposal network to efficiently generate spatiotemporal proposals. The tubelet starts from static proposals, and extracts multi-frame features, in order to predict the object motion patterns relative to the spatial anchor. The detector extends 2-D proposals to spatiotemporal tubelet proposals. All those methods will be our strong baselines.

## 3 Fully Motion-Aware Network

### 3.1 Overview

We first briefly overview the entire pipeline. Table 1 summarizes the main notations used in this paper. The proposed model is built on standard still image detector which consists of the feature extractor $\mathcal{N}_{feat}$, the region proposal network $\mathcal{N}_{rpn}$ [22] and the region-based detector $\mathcal{N}_{rfcn}$ [2]. The key idea of the proposed model is to aggregate neighboring frames through feature calibration.

First, $\mathcal{N}_{feat}$ will simultaneously receive three frames $\boldsymbol{I}_{t-\tau}$, $\boldsymbol{I}_t$ and $\boldsymbol{I}_{t+\tau}$ as input, and produce the intermediate features $\boldsymbol{f}_{t-\tau}$, $\boldsymbol{f}_t$ and $\boldsymbol{f}_{t+\tau}$. As shown in

| | |
|---|---|
| $t-\tau, t, t+\tau$ | video frames index |
| $i$ | proposal index |
| $(x, y, w, h)$ | proposal location described by center $(x, y)$, height and width |
| $(\Delta_x, \Delta_y, \Delta_w, \Delta_h)$ | normed proposal movements |
| $\boldsymbol{I}$ | video frame |
| $\boldsymbol{p}, \boldsymbol{q}$ | 2D location |
| $\boldsymbol{f}, \boldsymbol{s}$ | output feature maps and score maps |
| $\mathcal{N}_{feat}, \mathcal{N}_{rpn}, \mathcal{N}_{rfcn}$ | CNNs for feature extractor, RPN and R-FCN |
| $\mathcal{F}$ | Functions of flow estimation |
| $\mathcal{W}, G$ | Bi-linear interpolation $\mathcal{W}$ with its kernel function $G$ |
| $\phi, \psi$ | ROI pooling and position-sensitive ROI pooling |

**Table 1.** Notations.

Figure 2, the horizontal line running through the middle of the diagram produces the reference features $\boldsymbol{f}_t$. The top and bottom lines are nearby features $\boldsymbol{f}_{t-\tau}$ and $\boldsymbol{f}_{t+\tau}$. These single frame features will be spatially calibrated through the following two steps.

Second, the pixel-level calibration will be first applied to calibrate $\boldsymbol{f}_{t-\tau}$ and $\boldsymbol{f}_{t+\tau}$, generating $\boldsymbol{f}_{t-\tau\to t}$ and $\boldsymbol{f}_{t+\tau\to t}$. These features are then aggregated as $\boldsymbol{f}_{pixel}$. The elaborated formulations are in Section 3.2. $\boldsymbol{f}_{pixel}$ is subsequently delivered to $\mathcal{N}_{rpn}$ to produce proposals, as well as $\mathcal{N}_{rfcn}$, waiting to be further combined with instance-level calibrated features.

Third, the instance-level calibration is conducted on the position-sensitive score maps in $\mathcal{N}_{rfcn}$. Specialized convolutional layers are applied on $\boldsymbol{f}_{t-\tau}$, $\boldsymbol{f}_t$ and $\boldsymbol{f}_{t+\tau}$ to produce a bank of $k^2$ position-sensitive score maps $\boldsymbol{s}_{t-\tau}, \boldsymbol{s}_t$ and $\boldsymbol{s}_{t+\tau}$. For the $i$-th proposal $(x_t^i, y_t^i, w_t^i, h_t^i)$ of $\boldsymbol{s}_t$, we introduce a procedure to regress the corresponding proposal location $(x_{t-\tau}^i, y_{t-\tau}^i, w_{t-\tau}^i, h_{t-\tau}^i)$ for $\boldsymbol{s}_{t-\tau}$ and $(x_{t+\tau}^i, y_{t+\tau}^i, w_{t+\tau}^i, h_{t+\tau}^i)$ for $\boldsymbol{s}_{t+\tau}$. As formulated in Section 3.3, with these predicted proposal, features in nearby frames are RoI-pooled and aggregated as $\boldsymbol{s}_{insta}^i$.

At last, motion pattern reasoning is carried out to decide how to combine the different calibrated features. Since $\boldsymbol{f}_{pixel}$ is also fed into $\mathcal{N}_{rfcn}$, it produces $\boldsymbol{s}_{pixel}^i$ for the $i$-th proposal. Such module is designed to combine $\boldsymbol{s}_{insta}^i$ and $\boldsymbol{s}_{pixel}^i$ according to dynamic motion pattern. It is described in Section 3.4.

In our method, all the modules, including feature extractor $\mathcal{N}_{feat}$, $\mathcal{N}_{rpn}$, $\mathcal{N}_{rfcn}$, pixel-level calibration, instance-level calibration and motion pattern reasoning are trained end-to-end.

### 3.2 Pixel-level Calibration

As motivated by [31] and [30], given a reference frame $\boldsymbol{I}_t$ and a neighbor frame $\boldsymbol{I}_{t-\tau}$ (or $\boldsymbol{I}_{t+\tau}$), we can model the pixel-level calibration through optical flow estimation. Let $\mathcal{F}$ be a flow estimation algorithm, such as FlowNet [4], and $\mathcal{F}(\boldsymbol{I}_{t-\tau}, \boldsymbol{I}_t)$ indicates the flow field estimated through such network from frame $\boldsymbol{I}_t$ to $\boldsymbol{I}_{t-\tau}$ . Then we can warp the feature maps from the neighbor frames to the

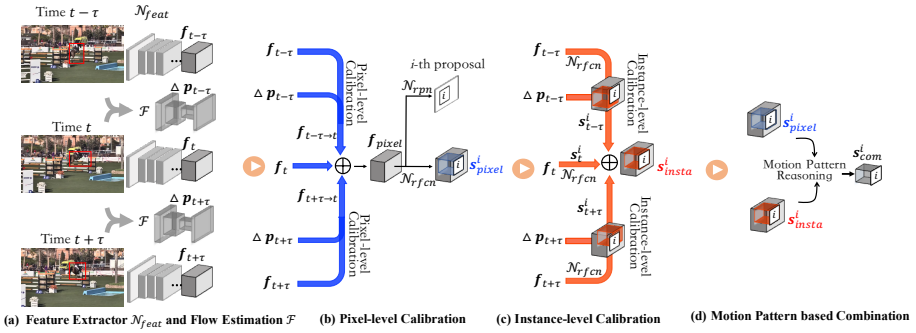



**Fig. 2.** (Better viewed in color) The overall framework of the proposed fully motion-aware network (MANet). It composes the four steps below: (a) single frame feature extraction and flow estimation whose results are fed to the next two steps; (b) the pixel-level calibration by per-pixel warping; (c) the instance-level calibration through predicting instance movements; (d) the motion pattern based feature combination.

current frame as follows:

$$\begin{aligned} \boldsymbol{f}_{t-\tau} &= \mathcal{N}_{feat}(\boldsymbol{I}_{t-\tau}) \\ \boldsymbol{f}_{t-\tau\to t} &= \mathcal{W}(\boldsymbol{f}_{t-\tau}, \mathcal{F}(\boldsymbol{I}_{t-\tau}, \boldsymbol{I}_t)) \end{aligned} \tag{1}$$

where $\boldsymbol{f}_{t-\tau}$ denotes feature maps extracted by $\mathcal{N}_{feat}$ and $\boldsymbol{f}_{t-\tau\to t}$ is the warped features from time $t-\tau$ to time $t$. The warping operation $\mathcal{W}$ is implemented by bi-linear function which is applied on each location for all the feature maps. It projects a location $\boldsymbol{p}+\Delta\boldsymbol{p}$ in the nearby frame $t-\tau$ to the location $\boldsymbol{p}$ in the current frame. We formulate it as:

$$\begin{aligned} \Delta\boldsymbol{p} &= \mathcal{F}(\boldsymbol{I}_{t-\tau}, \boldsymbol{I}_t)(\boldsymbol{p}) \\ \boldsymbol{f}_{t-\tau\to t}(\boldsymbol{p}) &= \sum_{\boldsymbol{q}} G(\boldsymbol{q}, \boldsymbol{p}+\Delta\boldsymbol{p}) \boldsymbol{f}_{t-\tau}(\boldsymbol{q}) \end{aligned} \tag{2}$$

where $\Delta\boldsymbol{p}$ is the output of flow estimation at location $\boldsymbol{p}$. $\boldsymbol{q}$ enumerates all spatial locations in the feature maps $\boldsymbol{f}_{t-\tau}$, and $G(\cdot)$ denotes bi-linear interpolation kernel as follow:

$$G(\boldsymbol{q}, \boldsymbol{p}+\Delta\boldsymbol{p}) = max(0, 1-||\boldsymbol{q}-(\boldsymbol{p}+\Delta\boldsymbol{p})||) \tag{3}$$

After obtaining calibrated features of nearby frames, we average these features as the low-level aggregation for the updated reference features:

$$\boldsymbol{f}_{pixel} = \frac{\sum_{j=t-\tau}^{t+\tau} \boldsymbol{f}_{j\to t}}{2\tau+1} \tag{4}$$

where $\boldsymbol{f}_{pixel}$ is generated by the nearby frames from time $t-\tau$ to time $t+\tau$. [30] proposes an adaptive weight to combine those nearby features. But we find that averaging motion guided features has the similar performance with less computation cost. As a result, we adopt average operation in our model.

Through the pixel-wise calibration, the features of nearby frames are spatially-temporally calibrated so as to provide diverse information for the reference frame. It alleviates several challenges in videos such as motion blur and video defocus.

### 3.3 Instance-level Calibration

The pixel-level feature calibration is flexible for modeling non-rigid motion, which needs precise per-pixel correspondence. But the low-level calibration may be inaccurate when object is occluded. In this subsection, we extend it to instance-level motion modeling which has much more tolerance of occlusions.

The instance-level calibration is conducted on score maps of R-FCN. R-FCN uses specialized convolutional layers to produce position-sensitive score maps $\boldsymbol{s}_t$. In order to aggregate scores for the $i$-th proposal $\boldsymbol{s}_t^i$, we should obtain the $\boldsymbol{s}_{t-\tau}$, $\boldsymbol{s}_{t+\tau}$ and proposal movements. $\boldsymbol{s}_{t-\tau}$ and $\boldsymbol{s}_{t+\tau}$ can be easily yielded by feeding $\boldsymbol{f}_{t-\tau}$ and $\boldsymbol{f}_{t+\tau}$ to the R-FCN. The problem is how to learn the relative movements of the $i$-th proposal, which is the prerequisites for calibrating instance-level features.

We employ the flow estimation and proposals of reference frame as input, and produce movements of each proposal between the neighboring frame and the current frame. The relative movements require motion information. Although per-pixel motion prediction by FlowNet is not accurate due to occlusion, it is capable of describing the motion tendency. We use this motion tendency as input, and output the movements of the entire object. Similar to the Section 3.2, we only formulate the relationship between $\boldsymbol{I}_{t-\tau}$ and $\boldsymbol{I}$, and $\boldsymbol{I}_{t+\tau}$ is in a similar way.

First, we utilize the RoI pooling operation to generate the pooled features $\boldsymbol{m}_{t-\tau}^i$ of the $i$-th proposal at location $(x_t^i, y_t^i, h_t^i, w_t^i)$:

$$\boldsymbol{m}_{t-\tau}^i = \phi(\mathcal{F}(\boldsymbol{I}_{t-\tau}, \boldsymbol{I}_t), (x_t^i, y_t^i, h_t^i, w_t^i)) \tag{5}$$

where $\phi(\cdot)$ indicates the RoI pooling [6] and $\mathcal{F}(\boldsymbol{I}_{t-\tau}, \boldsymbol{I}_t)$ is the flow estimation produced by shared FlowNet in Section 3.2. RoI pooling uses max pooling to convert the features inside any valid region of interest into a small feature map with fixed spatial extent.

Then regression network $R(\cdot)$ is exploited to estimate the movement of the $i$-th proposal between the frame $t-\tau$ and $t$ according to the $\boldsymbol{m}_{t-\tau}^i$:

$$(\Delta_{x_{t-\tau}}^i, \Delta_{y_{t-\tau}}^i, \Delta_{w_{t-\tau}}^i, \Delta_{h_{t-\tau}}^i) = R(\boldsymbol{m}_{t-\tau}^i) \tag{6}$$

where $(\Delta_{x_{t-\tau}}^i, \Delta_{y_{t-\tau}}^i, \Delta_{w_{t-\tau}}^i, \Delta_{h_{t-\tau}}^i)$ is relative movements and $R(\cdot)$ is implemented by a fully connected layer. The remaining problem is how to design proper supervisions for learning the relative movements. Since we have the track-id of each object within a video, we are able to generate the relative movements in terms of the ground-truth bounding boxes. We believe that the proposals should have consistent movement with the ground-truth objects. Thus, the above regression target is assigned the ground-truth box movement if the proposal overlaps with a ground-truth at least by 0.5 in intersection-over-union (IoU). In other word, only the positive proposals will learn to regress the movements among

consecutive frames. We use the normed relative movements as regression targets.

Once we obtain the relative movements, we are able to calibrate the features across time and aggregate them to enhance the feature of the current frame. The proposal of frame $\boldsymbol{I}_{t-\tau}$ can be inferred as:

$$\begin{aligned} x^i_{t-\tau} &= \Delta^i_{x_{t-\tau}} \times w^i_t + x^i_t \qquad & y^i_{t-\tau} &= \Delta^i_{y_{t-\tau}} \times h^i_t + y^i_t \\ w^i_{t-\tau} &= exp(\Delta^i_{w_{t-\tau}}) \times w^i_t \qquad & h^i_{t-\tau} &= exp(\Delta^i_{h_{t-\tau}}) \times h^i_t \end{aligned} \tag{7}$$

Based on the estimated proposal locations for nearby frames, the aggregated feature of the $i$-th proposal can be calculated as:

$$\boldsymbol{s}^i_{insta} = \frac{\sum_{j=t-\tau}^{t+\tau} \psi(\boldsymbol{s}_j, (x^i_j, y^i_j, h^i_j, w^i_j))}{2\tau + 1} \tag{8}$$

where $\boldsymbol{s}_j$ denotes the neighboring score maps, $\psi$ indicates position-sensitive pooling layer introduced by [2], and $\boldsymbol{s}^i_{insta}$ is the instance-level calibrated feature of the $i$-th proposal.

**Discussion about the regression of relative movements.** In [13], they have the similar movement regression problem when generating tubelets. They utilize pooled multi-frame visual features from the same spatial location of proposals to regress the movements of the objects. However, these features within the same location across time without explicit motion information make the regression difficult for training. In our instance-level movements learning, we use flow estimation as input to predict movements. It can regress the movements of all the proposals simultaneously without any extra initialization tricks. [5] proposes a correlation based regression. Compared to this additional correlation operation, we adopt a shared FlowNet to model two kinds of motions (both pixel-level and instance-level) simultaneously. This brings two advantages: 1) the feature sharing saves computation cost (shown in Section 4.6). 2) the supervision for instance-level movement regression provides additional motion information and improves flow estimation as well.

### 3.4 Motion Pattern Reasoning and Overall Learning Objective

Section 3.2-3.3 give two motion estimation methods. Since they have respective advantages on different motion, the key issue of combination is to measure the non-rigidity of the motion pattern. Intuitively, when the boundingbox's aspect ratio $\frac{x^i_t}{y^i_t}$ changes rapidly across time, the motion pattern is more likely to be non-rigid. Thus, we use the central-difference $\delta(\frac{x^i_t}{y^i_t})$ to express the change rate of aspect ratio at current time. In order to provide more stable estimates, we use average operation over a short snippet to produce the final descriptor of motion

pattern:

$$\begin{aligned} \delta(\frac{x_t^i}{y_t^i}) &= (\frac{x_{t+1}^i}{y_{t+1}^i} - \frac{x_{t-1}^i}{y_{t-1}^i})/2 \\ p_{nonri}^i &= \frac{\sum_{j=t-\tau+1}^{t+\tau-1} \delta(\frac{x_j^i}{y_j^i})}{2\tau - 1} \end{aligned} \tag{9}$$

where $p_{nonri}^i$ is the motion pattern descriptor for the $i$-th proposal. The corresponding proposals in the nearby frames can be obtained from Section 3.3.

Additionally, occlusion is another important factor when combining these two calibrations. We exploit the visual feature within the proposal to predict the probability of the object being occluded:

$$p_{occlu}^i = R(\phi(\boldsymbol{f}_t, (x_t^i, y_t^i, h_t^i, w_t^i))) \tag{10}$$

where $R(\cdot)$ is also implemented by a fully connected layer and $p_{occlu}^i$ is the probability of occlusion for the $i$-th proposal. Notice that Equation 10 is similar to Equation 6, but Equation 6 uses motion features from FlowNet to regress movements while Equation 10 adopts visual features to predict occlusion. It is mainly due to the fact that occlusion is more related to appearance.

Considering these two factors, we use learnable soft weights to combine the two calibrated features:

$$\boldsymbol{s}_{com}^i = \boldsymbol{s}_{insta}^i \times \alpha(\frac{p_{occlu}^i}{p_{nonri}^i}) + \boldsymbol{s}_{pixel}^i \times (1 - \alpha(\frac{p_{occlu}^i}{p_{nonri}^i})) \tag{11}$$

where $\alpha(\cdot) : \mathbb{R} \rightarrow [0, 1]$ is the mapping function that controls the adjustment range for the weight.

The overall learning objective function is given as:

$$\begin{aligned} \mathcal{L}(I) = &\frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{cls}(p^i, c_{gt}^i) + \\ &\frac{1}{N_{fg}} \sum_{i=1}^{N} \mathbf{1}\{c_i^{gt} > 0\} (\mathcal{L}_{reg}(b^i, b_{gt}^i) + \mathcal{L}_{cls}(p_{occlu}^i, c_{o\_gt}^i)) + \\ &\lambda \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \mathcal{L}_{tr}(\Delta^i, \Delta_{gt}^i) \end{aligned} \tag{12}$$

where $c_{gt}^i$ is the ground-truth class label. $p^i$ and $b^i$ stand for the predicted category-wise softmax score and bounding box regression based on $\boldsymbol{s}_{com}^i$. $p_{occlu}^i$ and $\Delta^i$ are occlusion probability and relative movement. $\mathbf{1}\{c_{gt}^i > 0\}$ denotes that we only regress the foreground proposals and $N_{tr}$ indicates that only positive proposals will learn to regress the movement targets. $\mathcal{L}_{cls}$ is the cross-entropy loss while $\mathcal{L}_{reg}$ and $\mathcal{L}_{tr}$ are defined as the smooth $L1$ function. The FlowNet is supervised by both the movement targets and the final detection targets.

Given the overall objective function, the whole architecture, including pixel-level calibration, instance-level calibration, motion pattern reasoning, bounding box classification and regression, is learned in an end-to-end way.

# 4 Experiments

## 4.1 Dataset Sampling and Evaluation Metrics

We evaluate the proposed framework on the ImageNet [23] object detection from video (VID) dataset that contains 30 classes. It is split into 3862 training videos and 555 validation videos. The 30 categories are labeled with ground-truth bounding boxes and track IDs on all the video frames. We report all results on the validation set and use the mean average precision (mAP) as the evaluation metric by following the protocols in [30, 31, 13].

The 30 object categories in ImageNet VID are a subset of the 200 categories in the ImageNet DET dataset. Although there are more than 112,000 frames in VID training set, the redundancy among video frames make the training procedure less efficient. Moreover, the quality of frames in video is much poorer than the still images in DET dataset. Thus we follow previous approaches and train our model on an intersection of ImageNet VID and DET set - 30 categories. To sum up, we sample 10 frames from each video in VID dataset and at most 2K images per class from DET dataset as our training samples.

## 4.2 Training and Evaluation

Our model is trained by SGD optimization with momentum of 0.9. During the training, we use a batch size of 4 on 4GPUs, where each GPU holds one mini-batch. The two-phase training is performed. In the first phase, the model is trained on the mixture of DET and VID for 12K iterations, with learning rates of $2.5 \times 10^{-4}$ and $2.5 \times 10^{-5}$ in the first 80K and 40K iterations, respectively. In the second phase, the movement regression along with the R-FCN are learned for another 30K iteration on VID dataset in order to be more adapted to VID domain. The feature extractor ResNet101 model is pre-trained for ImageNet classification as default. FlowNet (the "Simple" version) is also pre-trained on synthetic Flying Chairs dataset in [4] in order to provide motion information. They are jointly learned during the above procedure. In both training and testing, we use single scale images with shorter dimension of 600 pixels. For testing we aggregate in total of 12 frames nearby to enhance the feature of the current frame by using the Equation 4 and Equation 9. Non-maximum suppression (NMS) is applied with intersection-over-union (IoU) threshold 0.7 in RPN and 0.4 on the scored and regressed proposals.

## 4.3 Ablation Study

In this section, we conduct an ablation study so as to validate the effectiveness of the proposed network. To make better analysis, we follow the evaluation protocols in [30] where the ground-truth objects are divided into three groups in accordance with their motion speed. They use object' averaged intersection-over-union(IoU) scores with its corresponding instances in the nearby frames as measurement. It means that the lower the motion IoU($< 0.7$) is , the faster

| Feature Extractor | ResNet-101 | | | | |
|---|---|---|---|---|---|
| Methods | (a) | (b) | (c) | (d) | (e) |
| multi-frame feature aggregation? | | ✓ | ✓ | ✓ | ✓ |
| Pixel-level Calibration? | | | ✓ | | ✓ |
| Instance-level Calibration? | | | | ✓ | ✓ |
| mAP(%) | 73.6 | 73.4 $\downarrow_{0.2}$ | 76.5 $\uparrow_{2.9}$ | 77.1 $\uparrow_{3.5}$ | **78.1** $\uparrow_{\mathbf{4.5}}$ |
| mAP(%)(slow) | 81.8 | 83.8 $\uparrow_{2.0}$ | 85.0 $\uparrow_{3.2}$ | 85.5 $\uparrow_{3.7}$ | **86.9** $\uparrow_{\mathbf{5.1}}$ |
| mAP(%)(medium) | 71.3 | 75.7 $\uparrow_{4.4}$ | 74.9 $\uparrow_{3.6}$ | 76.1 $\uparrow_{4.8}$ | **76.8** $\uparrow_{\mathbf{5.5}}$ |
| mAP(%)(fast) | 52.2 | 45.2 $\downarrow_{7.0}$ | 56.6 $\uparrow_{4.4}$ | 55.4 $\uparrow_{3.2}$ | **56.7** $\uparrow_{\mathbf{4.5}}$ |

**Table 2.** Accuracy of different methods on ImageNet VID validation, using ResNet-101 feature extraction networks.

the object moves. Otherwise, the larger Motion IoU ($score > 0.9$) expresses the object moves slowly. The rest is medium speed.

***Method (a)*** is the single-frame baseline. It achieves 73.6% mAP by using ResNet-101. All the other experiments keep the same setting as this baseline. Note that we only use the single model and do not add bells and whistles.

***Method (b)*** is carried out conducted by averaging multi-frame features. Even we use the same feature extractor in an end-to-end training manner, the model is even worse than our baseline result. It indicates the importance of motion guidance.

***Method (c)*** incorporates the pixel-level feature calibration. The pixel-wise motion information effectively enhances the information from nearby frames in feature aggregation.

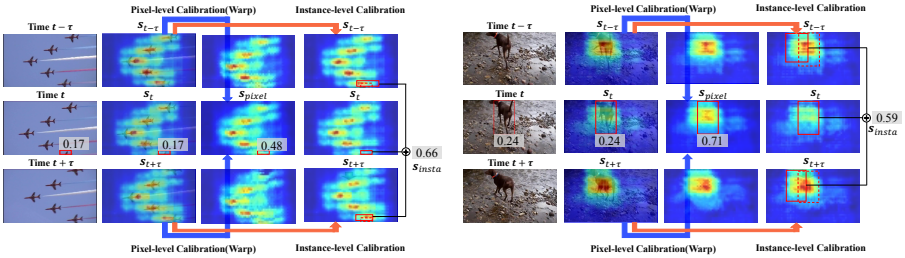
***Method (d)*** is the proposed the instance-level calibration. It aligns the proposal features by predicting the movements among consecutive frames, and finally aggregate them across time. It improves the overall performance by 3.5%, even better than the pixel-wise motion guided features in Method (c).

***Method (e)*** is conducted to prove the pixel-wise motion guided(Method (c)) and the instance-wise motion guided features (Method (d)) are complementary and they are able to collaboratively improve the model. We utilize the motion pattern reasoning (introduced by Section 3.4) to adaptively combine these two kinds of calibrated features, and it helps to further enhance the performance from 77.1% to 78.1%.

To sum up, aggregating the multi-frame features by explicitly modeling the motion is quite necessary, and the combination of these two calibration modes is capable of promoting the final feature representations collaboratively. Through the above modules, the overall mAP is improved from 73.6% to 78.1%.

### 4.4 Case Study and Motion Pattern Analysis

We attempt to take a deeper look at detection results. In order to prove that two calibrated features have respective strengths, we split the validation dataset




(a) Occluded airplane (b) Non-rigid motion

**Fig. 3.** (Better viewed in color) Visualization of two typical examples: occluded and non-rigid objects. They show respective strengths of the two calibration methods.

| Motion Pattern | Pixel-level | Instance-level | Combine |
|---|---|---|---|
| Occlusion | 73.0 | 74.1 | 75.3 |
| Rigid | 81.0 | 81.9 | 82.3 |
| Non-rigid | 52.8 | 51.6 | 53.2 |

**Table 3.** Statistical analysis on different validation sets. The instance-level calibration is better when objects are occluded or move more regularly while the pixel-level calibration performs well on non-rigid motion. Combination of these two module can achieve best performance.

into different subsets that include different typical samples. The first row in Table 3 shows the performance of occluded samples. We select 87,195 images from validation, where more than half bounding boxes are occluded. The instance-level calibration achieves better performance (74.1%) than pixel-level calibration (73.0%). In terms of motion pattern, we use $p_{nonri}$ to divide the dataset. The objects in a snippet whose $p_{nonri}$ are greater than pre-define $thresh$ will be considered as non-rigid motion, otherwise the rigid motion. $Thresh$ is set to 0.02 in our experiments. From the second and third rows of Table 3, the instance-level calibration is better for modeling rigid motion while pixel-level calibration has advantages of modeling non-rigid patterns. In particular, the adaptive combination distills their advantages and obtain the best performance.

We visualize the learned feature maps in order to better understand the two calibration methods. Figure 3(a) show an occluded airplane which is at the bottom of the current frame. When using a single frame detector, the confidence of category *"airplane"* is 0.17. When applying pixel-level calibrated features, it can be improved to 0.48 (the third column). However, due to the occluded part, the quality of warped feature is undesirably reduced. The last column is instance-level calibration. Since it uses original feature maps of nearby frames, the confidence of category *"airplane"* achieves 0.66. For non-rigid objects in Figure 3(b), both of the direction and trajectory are changed through the time, and the parts of dogs may have different motion tendencies. So it is difficult for instance-level module to produce correct movements of the whole dog. The

corresponding locations in the nearby frames are not accurate, leading to the unsatisfactory score 0.59. By contrast, the pixel-level calibration is flexible of modeling dog's motion and appearance, so it can achieve higher confidence 0.71.

### 4.5 Comparison with State-of-the-art Systems

We compare our model to the existing state-of-the-art methods which can be divided into two groups: end-to-end learned feature methods [2, 30, 13, 31] and post-processing based methods [15, 14, 5]. In terms of feature-level comparison, the proposed MANet achieves the best performance among these methods. [13] has the similar regression target with our instance movements learning. But it is much inferior to our calibrated features. [30, 31] are pixel-level feature aggregation and our model is better than these methods mainly due to the robustness of motion prediction. It has been analysed in Section 4.4.

| Methods | airplane | antelope | bear | bicycle | bird | bus | car | cattle | dog | d_cat | elephant | fox | g_panda | hamster | horse | lion |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R-FCN[2] | 90.5 | 80.1 | 83.0 | 69.6 | 73.4 | 72.4 | 57.2 | 62.5 | 69.0 | 81.6 | 77.3 | 85.0 | 80.7 | 87.0 | 72.5 | 41.6 |
| TPN+LSTM[13] | 84.6 | 78.1 | 72.0 | 67.2 | 68.0 | 80.1 | 54.7 | 61.2 | 61.6 | 78.9 | 71.6 | 83.2 | 78.1 | 91.5 | 66.8 | 21.6 |
| D (& T loss)[5] | 89.4 | 80.4 | 83.8 | 70.0 | 71.8 | 82.6 | 56.8 | 71.0 | 71.8 | 76.6 | 79.3 | 89.9 | 83.3 | 91.9 | 76.8 | 57.3 |
| DFF[31] | 84.6 | 82.1 | 84.1 | 67.1 | 71.1 | 76.1 | 56.5 | 67.8 | 65.0 | 82.3 | 76.3 | 87.8 | 81.9 | 91.3 | 70.3 | 47.7 |
| FGFA[30] | 89.4 | 85.1 | 83.9 | 69.8 | 73.5 | 79.0 | 60.6 | 70.7 | 72.5 | 84.3 | 79.9 | 89.8 | 81.0 | 93.3 | 72.3 | 50.5 |
| MANet | 90.1 | 87.3 | 83.4 | 70.9 | 73.0 | 75.6 | 62.0 | 74.0 | 73.3 | 85.3 | 79.6 | 91.6 | 83.5 | 96.5 | 74.5 | 70.5 |
| TCN [15] | 72.7 | 75.5 | 42.2 | 39.5 | 25.0 | 64.1 | 36.3 | 51.1 | 24.4 | 48.6 | 65.6 | 73.9 | 61.7 | 82.4 | 30.8 | 34.4 |
| TCNN[14] | 83.7 | 85.7 | 84.4 | 74.5 | 73.8 | 75.7 | 57.1 | 58.7 | 72.3 | 69.2 | 80.2 | 83.4 | 80.5 | 93.1 | 84.2 | 67.8 |
| D (& T loss)($\tau$= 1)[5] | 90.2 | 82.3 | 87.9 | 70.1 | 73.2 | 87.7 | 57.0 | 80.6 | 77.3 | 82.6 | 83.0 | 97.8 | 85.8 | 96.6 | 82.1 | 66.7 |
| MANet (+[8]) | 88.7 | 88.4 | 86.9 | 71.4 | 73.0 | 78.9 | 59.3 | 78.5 | 77.8 | 90.6 | 79.1 | 96.3 | 84.8 | 98.5 | 77.4 | 75.5 |

| Methods | lizard | monkey | motor | rabbit | r_panda | sheep | snake | squirrel | tiger | train | turtle | watercraft | whale | zebra | mAP(%) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R-FCN[2] | 78.0 | 52.2 | 81.2 | 66.6 | 81.5 | 57.3 | 70.5 | 53.1 | 90.8 | 82.3 | 79.1 | 64.6 | 75.0 | 91.2 | 73.6 |
| TPN+LSTM[13] | 74.4 | 36.6 | 76.3 | 51.4 | 70.6 | 64.2 | 61.2 | 42.3 | 84.8 | 78.1 | 77.2 | 61.5 | 66.9 | 88.5 | 68.4 |
| D (& T loss)[5] | 79.0 | 54.1 | 80.3 | 65.3 | 85.3 | 56.9 | 74.1 | 59.9 | 91.3 | 84.9 | 81.9 | 68.3 | 68.9 | 90.9 | 75.8 |
| DFF[31] | 76.5 | 45.7 | 78.1 | 62.8 | 77.8 | 55.8 | 74.5 | 50.5 | 90.2 | 81.7 | 77.9 | 65.8 | 66.2 | 89.5 | 72.8 |
| FGFA[30] | 80.8 | 52.3 | 83.0 | 72.7 | 84.0 | 57.8 | 77.1 | 55.8 | 91.9 | 83.8 | 83.3 | 68.7 | 75.9 | 91.1 | 76.5 |
| MANet | 82.0 | 54.4 | 81.6 | 67.0 | 89.3 | 73.3 | 77.4 | 54.3 | 91.9 | 82.9 | 80.3 | 69.3 | 75.4 | 92.4 | **78.1** |
| TCN[15] | 54.2 | 1.6 | 61.0 | 36.6 | 19.7 | 55.0 | 38.9 | 2.6 | 42.8 | 54.6 | 66.1 | 69.2 | 26.5 | 68.6 | 47.5 |
| TCNN[14] | 80.3 | 54.8 | 80.6 | 63.7 | 85.7 | 60.5 | 72.9 | 52.7 | 89.7 | 81.3 | 73.7 | 69.5 | 33.5 | 90.2 | 73.8 |
| D (& T loss)($\tau$= 1)[5] | 83.4 | 57.6 | 86.7 | 74.2 | 91.6 | 59.7 | 76.4 | 68.4 | 92.6 | 86.1 | 84.3 | 69.7 | 66.3 | 95.2 | 79.8 |
| MANet(+[8]) | 84.8 | 55.1 | 85.8 | 76.7 | 95.3 | 76.2 | 75.7 | 59.0 | 91.5 | 81.7 | 84.2 | 69.1 | 72.9 | 94.6 | **80.3** |

**Table 4.** Performance comparison with state-of-the-art systems on the ImageNet VID validation set. The average precision (in %) for each class and the mean average precision over all classes are provided.

Since the MANet aims to improve the feature quality in video frames, it can further incorporate bounding-box post-processing techniques to improve the recognition accuracy. Thus using post-processing based methods and combined with [8], the MANet achieves better performance (from 78.1% to 80.3%) that still outperforms other strong baselines [15, 14, 5].

To sum up, the comparison among feature based methods is more related to our motivation. Our model focuses on the end-to-end feature learning and has obvious advantages among these methods. In addition, we also demonstrate that the MANet can be further improved by post processing and achieves the state-of-the art performance.

### 4.6 Performance and Time-consuming Evaluation

Assume that $O(\cdot)$ is denoted as the time spent for the main model $\mathcal{N}$ ($\mathcal{N}_{feat} + \mathcal{N}_{rpn} + \mathcal{N}_{rfcn}$), $\mathcal{F}$ as the flow estimation, $\mathcal{W}$ as the pixel-level feature warping, $Ins$ as the instance-level regression and $Ocu$ as the occlusion predicting. When aggregating 1 adjacent frame, we have:

$$\begin{aligned} &O(\mathcal{N}) = (82.8ms) \gg O(\mathcal{F}) = (6.8ms) > \\ &O(Ocu) = (2ms) > O(Ins) = (1.5ms) > O(\mathcal{W}) = (0.8ms) \end{aligned} \tag{13}$$

where the aggregation modules take negligible time-consuming compared to $\mathcal{N}$.

For testing, we aggregate $k$ nearby frames to enhance the reference frame. The performance and time for varying $k$ are listed in Table 5. Notice that aggregating nearby 4 frames, our model can achieve 77.58 % mAP, which exceeds the performance of [30] where nearby 20 frames are aggregated.

| $k$ | 0 | 4 | 8 | 12 | 16 | 18 |
|---|---|---|---|---|---|---|
| mAP(%) | 73.57 | 77.58 | 77.96 | 78.09 | 78.08 | 78.07 |
| runtime(ms) | 87.4 | 126.8 | 161.3 | 201.8 | 241.1 | 269.7 |

**Table 5.** Results obtained by using different $k$ in inference. The runtime contains data processing which is measured on an NVIDIA Titan X Pascal GPU.

## 5 Conclusions

We propose an end-to-end learning framework for video object detection by aggregating multi-frame features in a principled way. We model the motion among consecutive frames in two different ways and combine them to further improve the performance of the model. We conduct extensive ablation study to prove the effectiveness of each module in our model. In addition, we also give in-depth analysis of their respective strengths on modeling different motion. The proposed model achieves 80.3% mAP on the large-scale ImageNet VID dataset with backbone network ResNet101, which outperforms existing state-of-the-art results.

## Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant No. 2017YFB1302200 and by Joint Fund of NORINCO Group of China for Advanced Research under Grant No. 6141B010318.

## References


1. Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., Feng, J.: Dual path networks. CoRR **abs/1707.01629** (2017)
2. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems. pp. 379–387 (2016)
3. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 764–773 (2017)
4. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2758–2766 (2015)
5. Feichtenhofer, C., Pinz, A., Zisserman, A.: Detect to track and track to detect. In: International Conference on Computer Vision (ICCV) (2017)
6. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
7. Girshick, R.B., 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, CVPR 2014, Columbus, OH, USA, June 23-28, 2014. pp. 580–587 (2014)
8. Han, W., Khorrami, P., Paine, T.L., Ramachandran, P., Babaeizadeh, M., Shi, H., Li, J., Yan, S., Huang, T.S.: Seq-nms for video object detection. arXiv preprint arXiv:1602.08465 (2016)
9. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2015)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 2261–2269 (2017)
12. Hur, J., Roth, S.: Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 312–321 (2017)
13. Kang, K., Li, H., Xiao, T., Ouyang, W., Yan, J., Liu, X., Wang, X.: Object detection in videos with tubelet proposal networks. In: CVPR (2017)
14. Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., Zhang, C., Wang, Z., Wang, R., Wang, X., et al.: T-cnn: Tubelets with convolutional neural networks for object detection from videos. IEEE Transactions on Circuits and Systems for Video Technology (2017)
15. Kang, K., Ouyang, W., Li, H., Wang, X.: Object detection from video tubelets with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 817–825 (2016)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

17. Li, Y., Min, D., Do, M.N., Lu, J.: Fast guided global interpolation for depth and motion. In: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III. pp. 717–733 (2016)
18. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 936–944 (2017)
19. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 2999–3007 (2017)
20. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
21. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788 (2016)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
23. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
24. Shrivastava, A., Gupta, A., Girshick, R.B.: Training region-based object detectors with online hard example mining. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 761–769 (2016)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
26. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA. pp. 4278–4284 (2017)
27. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
28. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. International Journal of Computer Vision **104**(2), 154–171 (2013)
29. Zeng, X., Ouyang, W., Yang, B., Yan, J., Wang, X.: Gated bi-directional cnn for object detection. In: European Conference on Computer Vision. pp. 354–369. Springer (2016)
30. Zhu, X., Wang, Y., Dai, J., Yuan, L., Wei, Y.: Flow-guided feature aggregation for video object detection. In: ICCV (2017)
31. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: CVPR (2017)