Differential Network for Video Object Detection

Jing Shi
University of Rochester
j.shi@rochester.edu

Chenliang Xu
University of Rochester
chenliang.xu@rochester.edu

Abstract

Object detection in streaming videos has three requirement: consistency, online and real time. For consistency, we adopt incremental Seq-NMS [9] to link the current bounding boxes with previous ones to see whether the object is new or not. And we adopt Deep Feature Flow *(DFF) [34] to achieve on-line and real time. However, DFF is sensitive to the selection of a good key frame, so we proposed the* Differential Network *(DiffNet) to achieve an automatic key frame scheduling. DiffNet combines both information from raw image and optical flow to tell whether two frames need to transfer feature, which is applied ahead of the detection pipeline. We firstly evaluate the classification ability of DiffNet and then examine such method on ImageNet VID dataset and compare it with the fixed-step key frame scheduling.*

1. Introduction

Recent years have witnessed significant progress in object detection [16] in still images. It is natural for people to extend the detection task from image domain to video domain. However, such extension will introduce new challenges. First, applying the deep networks on each video frame leads to prohibitive computational cost. Second, recognition accuracy suffers from deteriorated appearances in videos that are seldom observed in still images, such as motion blur, video defocus, rare poses, etc. Third, adjacent frames has strong temporal correlation which might play a latent role to improve the detection result in video domain. There have been few works on video object detection.

Recently *Deep Feature Flow* (DFF) [34] proposed a view trying to address above challenges. It exploits data redundancy between consecutive frames to reduce the expensive feature computation on most frames and improves the speed. This method divides all frames into two sets: key frame sets and non-key frame sets. The deep network is applied only on key frames to compute their feature. To obtain the features at a non-key frame, an optical flow network [5] estimates the motions between the nearest key frame and the non-key frame. The feature map of key frame is warped to the non-key frame according to the flow motion. The warped feature is shown in Fig. 1(b). Hence, DFF transfers the cumbersome calculation of deep network to the much lighter calculation of optical flow on non-key frame, enabling the detection pipeline with faster speed.

However, the features for non-key frames are only approximated and error-prone; thus hurting the recognition accuracy in further extend. A major limitation of DFF is that the detection accuracy of a non-key frames depends heavily on the feature quality of its corresponding key frame and the reliability of the motion field between them. To put it more concrete, a deteriorated frame as a key frame will propagate its poor feature to others and jeopardize the feature of non-key frames. Likewise, a bad optical flow estimation will distort the feature from the key frame and contribute to a bad feature propagation. Thus, how to choose the key frame plays a critical role in DFF. However, DFF just adopts fixed key frame selecting scheme and fails to take account of the quality of both key frame and the optical flow. Thus it is necessary to formulate a key frame scheduling.

Our model construction obeys two rules: 1) update a new key frame when the estimation of optical flow is bad; 2) the key frame should not be a deteriorated frame. Further we unify the two rules into one principle – the frame with transfered feature should have comparative detection accuracy as it with its own feature. In this work, we propose a *Differential Network* (DiffNet) to remedy improper selection of key frames by following the above principle. DiffNet is applied ahead of the detection network. It takes key frame and current frame as input and outputs a binary classification of whether these two frames should pass feature or not. We firstly generate the passing label for each frame pairs and then use them to train DiffNet. This technique embodies the scheduling principle in the ground truth labeling process. It makes the key frame usage more efficient. The experiments show the classification accuracy for DiffNet is 76% and achieved mAP of 70.7% combined with detection pipeline, which is still 0.5% lower than detection by fixed step key frame scheduling. We expect the results

to be further improved by feature aggregation and box-level inference.

The contribution of this paper is 1) train the DiffNet to formulate an automatic key frame scheduling. 2) propose an adaptive sampling method to remedy the unbalanced classification problem. 3) further combine the image-level and box-level feature to strength the classification ability of DiffNet.

2. Related Work

Speed/accuracy trade-off in object detection. As [16] indicates that speed/accuracy trade-off of modern detection system can be achieved by different feature networks [27, 29, 12, 28, 30, 15, 2, 14, 31] and detection networks [8, 11, 7, 25, 3, 22, 10, 4], or varying some critical parameters such as image resolution, box proposal number. PVANET [20] and YOLO [23] even design specific feature networks for fast object detection. By applying several techniques (e.g. batch normalization, high resolution classifier, fine-grained features and multiscale training), YOLO9000 [24] achieves higher accuracy.

Since our proposed method only considers how to compute higher quality feature faster by using temporal information, and is not designed for any specific feature networks and detection networks, such techniques are also suitable for our proposed method

Video object detection. Existing object detection methods incorporating temporal information in video can be separated into box-level methods [19, 18, 9, 21, 17, 6] and feature-level methods [34, 33] (both are flow-based methods). Box-level methods usually focus on how to improve detection accuracy considering temporary consistency within a tracklet. T-CNN [18, 19] first propagates predicted bounding boxes to neighboring frames according to pre-computed optical flows, and then generates tubelets by applying tracking algorithms. Boxes along each tubelet will be re-scored based on the tubelet classification result. Seq-NMS [9] constructs sequences along nearby high-confidence bounding boxes from consecutive frames. Boxes of the sequence are re-scored to the average confidence, other boxes close to this sequence are suppressed. MCMOT [21] formulates the post-processing as a multi-object tracking problem, and finally tracking confidence are used to re-score detection confidence. TPN [17] first generates tubelet proposals across multiple frames ($\leq$ 20 frames) instead of bounding box proposals in a single frame, and then each tubelet proposal is classified into different classes by a LSTM based classifier. D&T [6] simultaneously outputs detection boxes and regression based tracking boxes with a single convolutional neural networks, and detection boxes are linked and re-scored based on tracking boxes. Feature-level methods usually use optical flow to get pixel-to-pixel correspondence among nearby frames. Although feature-level methods are more principle and can further incorporate with box-level methods, they suffer from inaccurate optical flow. Still ImageNet VID 2017 winner is powered by feature-level methods DFF [34] and FGFA [33]. Our proposed method is an improvement of a feature-level method, which introduces a DiffNet to select the key frames to pass the feature. The most similar work of this paper is [32]. It introduces Spatially-adaptive partial feature updating to fix the inaccurate feature propagation caused by inaccurate optical flow. However, this method uses pixel-wise information to indicate the incorrect optical flow of each pixel and just sum up the bad pixel of the whole image to schedule key frame selection, but fail to consider the box-wise information. In our work, not only the image-level, but the box-level information is used to select the key frame.

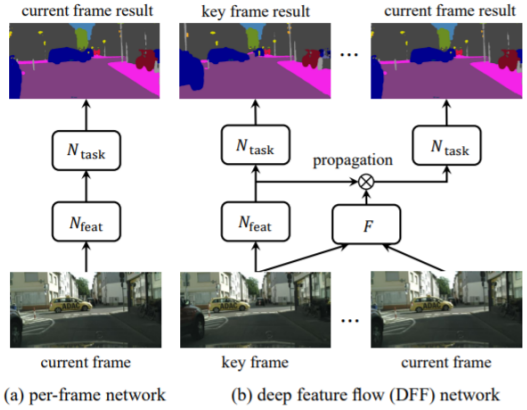

Figure 1. Illustration of video recognition using per-frame network evaluation (a) and deep feature flow (b). (figure cited from [33])

3. Methods

3.1. Deep Feature Flow

Deep feature flow [34] introduces the concept of key frame for video object detection. The motivation is that similar appearance among consecutive frames usually results in similar features, which is demonstrated by experiment that the feature warped by optical flow can still lead to a good detection. It is therefore unnecessary to compute features on all frames.

DFF Structrue. Structure of DFF is shown in Fig 1. During inference, the expensive feature network $\mathcal{N}_{feat}$ is applied only on sparse $keyframes$ (e.g., every 10 frames). The feature maps on any non-key frame i are propagated from its preceding key frame k by per-pixel feature value warping and bilinear interpolation. The between frame pixel-wise motion is recorded in a two dimensional *motion field* $M_{i\to k}$. The propagation from key frame k to frame i

Method	mAP(%)	runtime (fps)
Frame	73.9	1.52
DFF	73.1	20.25

Table 1. The comparison of DFF (key frame step 10) and detection by frame on ImageNet VID. We can see DFF has much faster speed than detection by each frame while maintaining a relative high mAP. Thus DFF is a good method to accelerate detection task in video.

is denoted as

$$F_{k\to i} = \mathcal{W}(F_k, M_{i\to k}) \tag{1}$$

where $\mathcal{W}$ represents the feature warping function, which is a bilinear sampler. Then the detection network $\mathcal{N}_{det}$ works on $F_{k\to i}$, the approximation to the real feature F_i , instead of computing F_i from $\mathcal{N}_{feat}$.

The motion field is estimated by a lightweight flow network, $\mathcal{N}_{flow}(I_k, I_i) = M_{i\to k}$ [5], which takes two frames I_k, I_i as input. End-to-end training of all modules, including $\mathcal{N}_{flow}$, greatly boosts the detection accuracy and makes up for the inaccuracy caused by feature approximation. Compared with the single frame detector, the computation of $\mathcal{N}_{flow}$ and Eq. 1 is much cheaper than feature extraction in $\mathcal{N}_{feat}$. The speed and accuracy comparison can be seen from Table 1.

Module Design. The featrue extraction network$\mathcal{N}_{feat}$ is ResNet-101. The feature warping function $\mathcal{W}$ is bilinear sampler. The flow network $\mathcal{N}_{flow}$ is specifically FlowNet-S in [5]. Here DFF is tasked for detection, thus $\mathcal{N}_{task}$ is detection network $\mathcal{N}_{det}$ which is implemented by R-FCN [3].

3.2. Differential Network

DFF [34] adopts fixed-step key frame scheduling, which might impede the detection result by selecting bad key frames. A good key frame shoulders twofold responsibilities. One is to end the feature transferring from the previous key frame due to large pixel difference and bad quality of optical flow. Another is to restart a good new feature that can be well transfered to its followers. Hence both historical and future information should be considered to draw a decision. However, in the real-time detection system only historical frames are available and the future information can only be inferred from history, rendering the task very hard. Formally, the possibility for current frame I_i to be a key frame is in condition of whole history and is expressed as $\mathbb{P}_{key}(I_i) = \mathbb{P}(I_i|I_1, I_2, ..., I_{i-1})$. To reduce computation complexity, we treat the I_i only dependent on its nearest previous key frame I_k, and the dependency is modeled as

$$\mathbb{P}_{key}(I_i) \approx \mathbb{P}(I_i|I_k) = 1 - \mathcal{N}_{diff}(I_i, I_k, M_{k\to i}) \tag{2}$$

where $\mathcal{N}_{diff}$ is Differential Network (DiffNet) and $M_{k\to i}$ is optical flow from I_k to I_i. DiffNet takes as input current frame, last key frame and optical flow between them and outputs the possibility for them to pass feature. If not passing feature, I_i becomes a new key frame.

Network Structure. Fig. 2 and Fig. 3 show the structure of DiffNet. It firstly concatenates I_i and I_k as an early fusing to get more fine-grained difference feature, and utilizes features and optical flow from FlowNet to learn the quality of motion field. The concatenated feature later goes into the tail part and passes RoI pooling layer to get a uniform size of 7×7, due to varied sizes of inputs. Finally a probability output is obtained through a fully-connect layer followed by a sigmoid function.

Get the training label for the differential network. We label the passing ground truth by the principle that the frame with transfered feature should have comparative detection accuracy as it does with its own feature. Specifically, we start by randomly selecting two frames (I_k and I_i) in a video. And we pass these two frame to feature extraction network to calculate their own features F_k and F_i. Then we use (1) to get I_i's transfered feature $F_{k\to i}$. With $F_{k\to i}$ and F_i, we compare the number of detected objects whose IoU score with the ground truth are greater than 0.5. The labeling method is shown in Table 2. If the number of detected objects from $F_{k\to i}$ is no less than that from F_i, we label passing; if no object is detected from both F_i and $F_{k\to i}$, we abandon such training pair (because there is no information in this situation); else we label not passing.

Adaptive sampling method. Random sampling in videos faces the problem of class unbalance, that is to say, the number of positive samples (passing feature) is much more than the number of negative samples (not passing feature), which is explained by the slow motion in many videos. We address such problem by an adaptive sampling method that can adjust the interval between a pair of images so as to modulate the label for such pair. In each video, 20 image pairs are sampled. In order to make the samples cover the entire video, the first 10 pairs are sampled from the start to the end of the video with equal interval $d = \lfloor \frac{D}{10} \rfloor$, where D is the length of the video and $\lfloor \cdot \rfloor$ is rounding down. The latter 10 pairs are sampled with adaptive interval that is recurrently updated as $d \leftarrow \lfloor \alpha d \rfloor$, where α is defined as

$$\alpha = \begin{cases} 1 & \text{if \#(positive sample) = \#(negative sample)} \\ 2 & \text{if \#(positive sample) } > \text{ \#(negative sample)} \\ \frac{1}{2} & \text{if \#(positive sample) } < \text{ \#(negative sample).} \end{cases}$$

And if the number of sample pairs in one class exceeds 10, we reject sampling that class until both classes are balanced or exceed maximum sampling count.

Training. DiffNet can be regarded as a stand alone module and can be trained separately from the detection part. Let $y_{k_n\to i_n} = \mathcal{N}_{diff}(I_{i_n}, I_{k_n}, M_{k_n\to i_n})$, and $t_{k_n\to i_n}$ be the ground truth label for passing feature. Cross-entropy

condition	label
$n_{k\to i} \geq n_i$ & $n_i > 0$	pass feature
$n_{k\to i} < n_i$	not pass feature
$n_{k\to i} = n_i = 0$	no label

Table 2. Labeling method for key frames, where $n_{k\to i}$ and n_i are the number of detected objects from $F_{k\to i}$ and F_i, respectively

loss is used for training and is written as

$$L = \frac{1}{N}\sum_{n=1}^{N}[t_{k_n\to i_n}\log(y_{k_n\to i_n}) + (1-t_{k_n\to i_n})\log(1-y_{k_n\to i_n})] \quad (3)$$

where n and N represent the index of sample pairs and total number of samples, respectively.

4. Incremental Seq-NMS

In order to find the new object in the video, linking boxes into tubes is a common way to maintain the trajectory history. Seq-NMS [9] provide an suppression method that can link the objects among frames and do a better suppression job. We further modify Seq-NMS into incremental way and thus can link the bounding boxes into cubes with streaming video data.

5. Experiments

The whole detection pipeline is divided into head part (DiffNet) for key frame selection, and body part (R-FCN and FlowNet) for object detection. During training, the parameters in the head part are trained and those in body part remain the same as [34].

5.1. Datasets and Evaluation

We perform experiments on ImageNet VID dataset [26], which is a prevalent large-scale benchmark for video object detection. Model training and evaluation are performed on the 3,862 video snippets from the training set and the 555 snippets from the validation set, respectively. The snippets are fully annotated at frame rates of 25 or 30 fps in general. There are 30 object categories, which are a subset of the categories in the ImageNet DET dataset.

To evaluate our method, the DiffNet was firstly tested on classification 5.3 and then tested on detection pipeline 5.4.

5.2. Implementation Details

When generating label, we set the detection confidence 0.7 to calculate the number of detected objects. 80K image pairs are sampled from training set with ratio of positive number to negative number 2 to 1, due to some short videos whose positive number cannot be sampled as In training DiffNet, images are resized to have the shorter side of 600 pixels and then center cropped into size of 600×600, so as to be trained in minibatch with batchsize 4. In testing, the image is resized with either shorter side of 600 pixels or longer side of 1000 pixels to maintain the aspect ratio, and the batchsize is 1. We implement the model using MxNet [1]. The DiffNet is optimized by Adam, where 3 epoch are performed on 4 GPUs to a convergence with loss 0.3. Testing time is measured on a Tesla K80 GPU.

5.3. Testing DiffNet

Since DiffNet is a stand alone module, it can be tested separately from detection pipeline. Thus we firstly sample 8.5K image pairs from validation set using the adaptive sampling and cut them into equal number of positive and negative samples, to compose a testing set for DiffNet. A good classification result for DiffNet is the premise for a good key frame scheduling and better detection result. Different structures for DiffNet are explored and the result is shown in Table 4, where the classification threshold is set as 0.5. Binary classification is sensitive to data imbalance; thus sensitivity and specificity are utilized for better evaluation. The best classification test accuracy is 76.7% using structure B in Fig. 2 and C1 in Fig. 3.

Need we balance the number of samples in two classes the same in training data? Due to some short videos (e.g. less than 20 frames) and slow motion videos, we cannot sample as much as negative samples as positive samples rendering the ratio of them 2 to 1. We duplicate the negative samples to overcome such class imbalance, which is the experiment (2) in Table 4. Comparison of experiment (1) and (2) indicates such balance will worsen the classification accuracy.

Tune or fix the FlowNet? Comparison of experiment (1) and (3) in Table 4 shows fixing FlowNet will decrease the accuracy about 5%. Thus when we train the DiffNet while tuning the FlowNet. However, the FlowNet tuned in DiffNet cannot guarantee the quality of optical flow, thus we use two FlowNets, one for DiffNet and one for warping feature.

RoI pooling or global average pooling? Experiment (4) and (6) in Table 4 use RoI pooling and global average pooling respectively, telling global average pooling has lower accuracy than RoI pooling.

Early fusion or late fusion? Considering DiffNet is used for see the difference between two frames and tell passing feature or not, the stage when the features of these two frames merge is studied in experiment (3) and (4). Structure B1 uses late fusion that merges the features of two frames in conv4 layer in Fig. 2. Results show early fusion is better.

Need we reduce the channel number of features to concatenate optical flow? Since the channel of optical flow

model	mAP %
NMS	72.93
Incre Seq-NMS	73.88

Table 3. Comparison of Incremental Seq-NMS and vanilla NMS

is 2, directly concatenate it to features with large number of channels might dwarf the effect of optical flow. In experiment (4), channel number in conv4 layer is reduced to 16, compared with experiment (5) where the channel number is 256. The result shows more channel number lead to better classification result.

5.4. Detection Pipeline

Having tested the classification performance of DiffNet, we then apply it as a head network to the detection pipeline, and the detection result is shown in Table 5. If we maintain the classification threshold for DiffNet 0.5, the key frames update at average step of 26, much larger than updating at fixed step 10, but have a lower mAP. In order to have a fair comparison between DiffNet and fixes step method, we adjust threshold of DiffNet to 0.7 and obtains an average step of 18, compared with fixed step 17. But the results reflect that DiffNet still cannot compete fix step method. Thus better methods are needed.

Because we use FlowNet twice in DiffNet detection pipeline, it consumes more time than fixed step method. The time consumption is shown in Table 5. We can see the data loader consume much time, which can be better engineered. And the FlowNet need to be further trained end to end thus it can be shared by DiffNet and DFF to reduce computing burden.

Some detections are visualized in Fig. 4. Four pairs of images are shown and in each pair the left uses DiffNet and the right uses fixed step method. Green boxes are GT and red ones are detections.

5.5. Test Incremental Seq-NMS

Incremental Seq-NMS are tested on fixed key frame scheduling at 10 steps on ImageNet VID. Table 3 shows the comparison of Incremental Seq-NMS and vanilla NMS and indicates the improvement of mAP for 0.95.

6. Improvement

6.1. Object specific Feature

We suspect that the DiffNet only seeing the whole image cannot provide enough cue for the appearance change of each object, thus we implement a method that uses the object specific area information rather than the whole image. The change is we do not pass the whole image into roipooing layer, but send each object region into roipooling and merge the feature after roipooling into uniform-size feature. Firstly, we use GT boxes as the region constraint and test it on classification task. The result is shown in Table 6. We can see DiffNet using object specific feature gets a better accuracy with FlowNet fixed compared with that using the whole image feature. However if we train using object specific feature with flownet tuned , the performance will decrease. Thus this method needs further consideration.

6.2. Feature aggregation

Inspired by [13, 32], features for key frames can be aggregated. In [32], the feature for the current key frame are iteratively aggregated by the feature of preceding key frame, and such method shows an improvement.

7. Conclusion

This paper proposes DiffNet to achieve adaptive key frame schilling, which is a stand alone module. It takes two frames, see the "difference" between them and judge whether passing feature or not. For labeling, an adaptive sampling method is introduced to balance two classes. We show our DiffNet have an mAP 0.5% lower then fixed step method at average step 18, indicating such method needs further improvement.

References

[1] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*, 2015.

[2] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint arXiv:1610.02357*, 2016.

[3] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. *arXiv preprint arXiv:1605.06409*, 2016.

[4] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *arXiv preprint arXiv:1703.06211*, 2017.

[5] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[6] C. Feichtenhofer, A. Pinz, and A. Zisserman. Detect to track and track to detect. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3038–3046, 2017.

[7] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

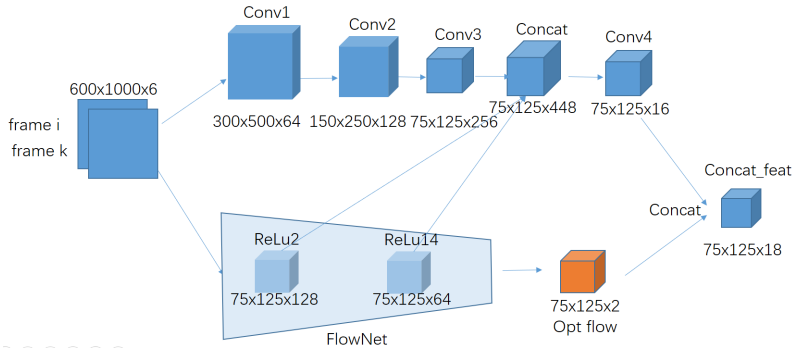

Figure 2. The head part of DiffNet, combining the feature of FlowNet

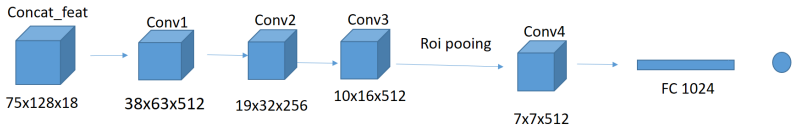

Figure 3. The tail part of DiffNet, with roi pooling

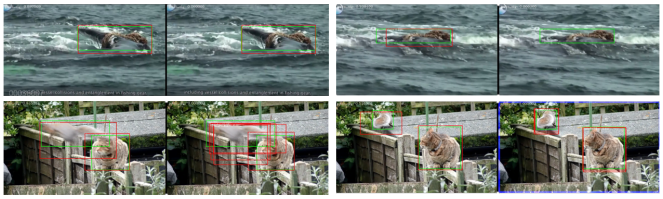

Figure 4. detection result, left is DiffNet, right is fix step

[9] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang. Seq-nms for video object detection. *arXiv preprint arXiv:1602.08465*, 2016.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] C. Hetang, H. Qin, S. Liu, and J. Yan. Impression network for video object detection. *arXiv preprint arXiv:1712.05896*, 2017.

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[15] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[16] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv preprint arXiv:1611.10012*, 2016.

[17] K. Kang, H. Li, T. Xiao, W. Ouyang, J. Yan, X. Liu, and X. Wang. Object detection in videos with tubelet proposal networks. *arXiv preprint arXiv:1702.06355*, 2017.

[18] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *arXiv preprint arXiv:1604.02532*, 2016.

[19] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 817–825, 2016.

[20] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park. Pvanet: Deep but lightweight neural networks for real-time object detection. *arXiv preprint arXiv:1608.08021*, 2016.

[21] B. Lee, E. Erdenee, S. Jin, M. Y. Nam, Y. G. Jung, and P. K. Rhee. Multi-class multi-object tracking using changing point detection. In *European Conference on Computer Vision*, pages 68–83. Springer, 2016.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. *arXiv preprint arXiv:1612.03144*, 2016.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[24] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In

Models	Training Set Accuracy	Training Set Sensitivity	Training Set Specificity	Testing Set Accuracy	Testing Set Sensitivity	Testing Set Specificity	Loss
(1) B+C1,flowtune	86.2%	86.7%	85.6%	76.7%	84.2%	69.3%	0.33
(2)B+C1,flowtune, balanced 1:1	80.81%	83.48%	78.18%	74.38%	83.75%	65.0%	0.37
(3) B+C1,flowfix	74.93%	79.43%	70.51%	71.27%	80.00%	62.55%	0.49
(4) B1+C1, flowfix	74.7%	82.2%	67.3%	70.9%	82.25%	59.74%	0.52
(5) B1+C1, flowfix, noshrink	75.4%	80.2%	70.3%	71.44%	79.9%	62.97%	0.51
(6) B1+C2, flowfix	71.47%	83.18%	59.96%	69.44%	85.36%	53.54%	0.50

Table 4. The test result of DiffNet. B is structure 2 with early feature fusing; B1 uses late feature fusion; C1 is structure 3 with RoI polling; C2 uses global average pooling; flowtune and flowfix means training with FlowNet tuned and fixed; nonshrink is for the channel of conv4 layer 256 rather than 16 in 2

experiment	mAP	ave step	dataloader(s)	DFF(s)	DiffNet(s)	nms(s)	fps	fps(no dataloader time)
DiffNet@0.5	68.8%	26	0.046	0.0222	0.0180	0.0065	10.78	21.42
DiffNet@0.7	70.7%	18	0.046	0.0228	0.0181	0.0064	10.71	21.14
Fix 17 step	71.2%	17	0.045	0.0233	0	0.0065	13.37	33.55
Fix 10 step	72.9%	10	0.046	0.0258	0	0.0065	13.10	30.96

Table 5. The result of detection pipeline on different key frame scheduling, and time consumption on different modules for a single frame.

Advances in neural information processing systems, pages 91–99, 2015.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[30] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995. IEEE, 2017.

[31] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *arXiv preprint arXiv:1707.01083*, 2017.

[32] X. Zhu, J. Dai, L. Yuan, and Y. Wei. Towards high performance video object detection. *arXiv preprint arXiv:1711.11577*, 2017.

[33] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. Flow-guided feature aggregation for video object detection. *arXiv preprint arXiv:1703.10025*, 2017.

[34] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. *arXiv preprint arXiv:1611.07715*, 2016.

Models	Training Set Accuracy	Training Set Sensitivity	Training Set Specificity	Testing Set Accuracy	Testing Set Sensitivity	Testing Set Specificity	Loss
B+C1,flowtune	86.2%	86.7%	85.6%	76.7%	84.2%	69.3%	0.33
B+C1,flowtune,objspec(without train)	84.9%	86.2%	83.6%	76.6%	84.3%	69.0%	-
B+C1,flowtune,objspec(trained)	-	-	-	66.6%	55.8%	77.5%	0.56
B+C1,flowfix	74.93%	79.43%	70.51%	71.27%	80.00%	62.55%	0.49
B+C1,flowfix,objspec(trained)	**78.4**%	77.4%	79.3%	**75.2**%	78.9%	71.5%	0.46

Table 6. The result of object-level DiffNet