Semi-Supervised DFF: Decoupling Detection and Feature Flow for Video Object Detectors

Guangxing Han, Xuan Zhang*, Chongrong Li
Beijing National Research Center for Information Science and Technology (BNRist),
Institute for Network Sciences and Cyberspace (INSC),
Tsinghua University, Beijing, 100084, China.
hgx14@mails.tsinghua.edu.cn,{zhangx,licr}@cernet.edu.cn

ABSTRACT

For efficient video object detection, our detector consists of a spatial module and a temporal module. The spatial module aims to detect objects in static frames using convolutional networks, and the temporal module propagates high-level CNN features to nearby frames via light-weight feature flow. Alternating the spatial and temporal module by a proper interval makes our detector fast and accurate. Then we propose a two-stage semi-supervised learning framework to train our detector, which fully exploits unlabeled videos by decoupling the spatial and temporal module. In the first stage, the spatial module is learned by traditional supervised learning. In the second stage, we employ both feature regression loss and feature semantic loss to learn our temporal module via unsupervised learning. Different to traditional methods, our method can largely exploit unlabeled videos and bridges the gap of object detectors in image and video domain. Experiments on the large-scale ImageNet VID dataset demonstrate the effectiveness of our method. Code will be made publicly available.

CCS CONCEPTS

• **Computing methodologies → Object detection**;

KEYWORDS

Video Object Detection; Feature Flow; Knowledge Distillation

ACM Reference Format:
Guangxing Han, Xuan Zhang*, Chongrong Li. 2018. Semi-Supervised DFF: Decoupling Detection and Feature Flow for Video Object Detectors. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240693

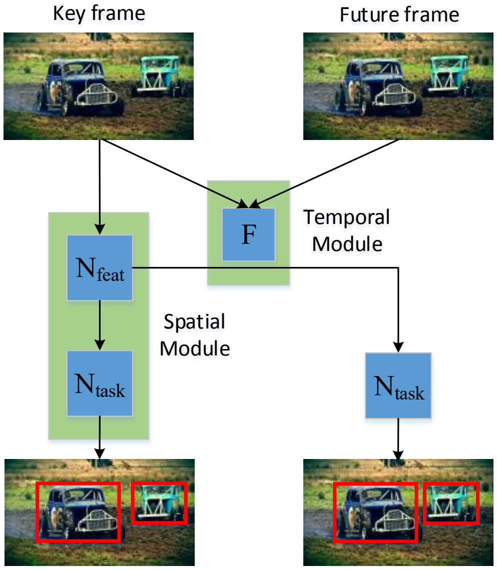

Figure 1: Overview of our video object detector. Our video object detector consists of a spatial module and a temporal module. In spatial module, N_{feat} and N_{task} is short for the feature network and task network respectively, and they detect objects in static frames. The temporal module is the feature flow network F, which propagates CNN features to adjacent frames. Our detector is fast and accurate by combining these two modules efficiently. We also propose a novel semi-supervised learning framework to train our detector by decoupling these two modules.

1 INTRODUCTION

Currently, object detection [6, 17, 35, 42] has achieved significant success due to the rapid development of deep convolutional neural networks (DCNNs [18, 20, 30, 49, 50]). Both fully convolutional architecture design and end-to-end joint training push object detection to real-time speed and state-of-the-art accuracy on several datasets, e.g., PASCAL VOC [8], MS COCO [33] and ILSVRC [45].

Considering the maturity of object detection in static images, it's natural to extend its domain from images to videos. Different from detecting objects in static images, videos provide additional temporal information [57] beyond static appearance features. Naive frame-by-frame independent detection usually requires unaffordable computational cost. Thus exploiting temporal coherence for fast and accurate object detection is critical in many real-time applications, e.g., autonomous driving. In addition, dense labeling in large scale of video data is laborious and difficult to acquire. Only relatively small video snippets (3862 training snippets in [45]) or

*Corresponding author: Xuan Zhang.

MM '18, October 22–26, 2018, Seoul, Republic of Korea
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240693

sparse ground truth labels (1 labeled frame per second in [40]) are available for video object detection tasks in the research community. More attention should be paid to unsupervised or semi-supervised learning in videos.

In this paper, we propose to learn fast and accurate video object detectors by leveraging the rich information in unlabeled video data. We will next briefly describe our motivation and approach in how to learn fast and accurate video object detectors respectively. First, different to the naive per-frame independent detection, we exploit the inherent temporal coherence of video data for fast detection in videos. Consecutive frames in a video usually have similar appearance, and therefore fully exploiting this property is very promising for video object detectors. Fortunately, the elegant design of modern object detectors is particularly suitable to exploit temporal information. Modern object detectors [21], based on R-CNN framework [12], usually consist of two parts: image feature extraction sub-network and RoI-wise classification sub-network. In the current state-of-the-art object detectors such as R-FCN [6], most of the calculations are spent on the feature extraction stage, which usually consists of hundreds of convolutional layers (e.g., ResNet-101 [18]). Convolutional networks can extract high-level semantic features through hierarchical abstraction and naturally preserve spatial correspondences between the images and CNN features. Shelhamer et al. [46] show that high-level features evolve more slowly compared to raw video pixels. Based on these observations, we divide object detectors into a deep, costly feature network and a shallow, cheap task network [11]. Then we only compute features on sparse key frames, and employ light-weight feature flow [63] to propagate CNN features along the temporal dimension through spatial warping [23]. The task network is applied to all the frames. Considering the cheap operations of feature flow and task network, we can largely speed up the running speed of object detection in videos.

Then the key problem can be described as estimating accurate feature flow between adjacent frames given the current and future frames as well as the CNN features of the current frame. Once we get the transformation network, we can easily propagate CNN features to future frames fast. Therefore the quality of the estimated feature flow directly affect the accuracy of our video object detectors. Different from optical flow, feature flow estimates the motion of high-level semantic concepts, and should be more smoother than the motion of original pixels. In addition, unlike optical flow [7], it's challenge to label ground truth feature flow even on synthetically generated datasets. DFF [63] builds the feature flow module and detection module into an end-to-end trainable network, and learns the feature flow network jointly with detection network. Although only sparsely annotated frames in videos are needed, DFF relies on ground truth boxes in videos to supervise the learning. Consequently DFF is restricted to scenarios where labeled videos are available, and cannot fully leverage the power of large scale of unlabeled videos.

To cope with this problem, we propose to decouple the learning of detection and feature flow networks in our video object detection system. Concretely, we learn to detect objects and estimate high-level feature flow separately in a two-stage semi-supervised learning framework. In the first stage, we learn the parameters of the detection network by traditional supervised method [6]. It's rather simple to train state-of-the-art object detectors employing the current techniques [21]. In the second stage, we propose an unsupervised learning algorithm for the feature flow network by leveraging large scale of unlabeled videos. More specifically, we make full use of the trained state-of-the-art object detectors in the first stage (teacher model), and learn feature flow estimation by transferring the knowledge from the teacher model to our video object detectors (student model). Traditional unsupervised optical flow estimation [58] minimizes the loss function of flow warp error calculated by the pixel difference of the warped image and ground truth image. This method is also applicable to our feature flow. However, it's hard to directly minimizing the regression loss of high-dimensional CNN features and estimate feature flow suitable for our detection task. Inspired by perceptual loss in [27] and knowledge distillation in [19], we propose to minimize the semantic loss of warped feature specific to our detection task, which is also guided by our teacher model. Furthermore, we combine these two losses in a two-step optimization method and achieve better results than either of them.

In summary, our proposed semi-supervised DFF, denoted as semi-DFF, decouples the spatial and temporal modules in our video object detector, and is very promising considering its ability to leverage unlabeled videos. Furthermore, semi-DFF bridges the gap of object detection in image and video domain, and can serve as a more general framework for transferring static image recognition networks to video domain. To evaluate our method, we conduct several experiments of different experimental settings on video object detection task. Comprehensive experiments demonstrate the superiority of our semi-DFF on both running speed and detection accuracy compared to a large variety of strong competitors.

Our main contributions are in three folds.

(1) We propose semi-DFF, a novel general semi-supervised learning framework for fast and accurate video object detection, which decouples the learning of detection and feature flow network, and bridges the gap between state-of-the-art object detectors in images and videos.
(2) For unsupervised feature flow estimation, we employ regression loss as well as novel semantic loss in a two-step optimization method, which can fully exploit rich knowledge in unlabeled videos to learn better feature flow.
(3) We comprehensively evaluate our proposed method on ImageNet VID validation dataset. Our method achieves largely speedup (4× faster) and minor accuracy decrease compared to the strong frame-by-frame R-FCN baseline. We also achieve superior performance in accuracy and running speed trade-offs compared to single shot detectors. Furthermore, our semi-supervised method also outperforms the purely supervised DFF given more unlabeled videos.

2 RELATED WORK

Object Detection in Images. Currently, object detectors can be grouped into two main families: two-stage detectors [6, 15, 17, 42] and single shot detectors [16, 34, 35, 41]. Two-stage detectors, e.g., Faster R-CNN [42] and Mask R-CNN [17], divide detection into two stages: region proposal generation (RPN [42]) and RoI-wise classification (R-CNN [11, 12]). This cascade design usually achieves higher detection accuracy. R-FCN [6] further shares the calculation in the 2nd stage by introducing a position-sensitive RoI pooling layer, which greatly reduce the computation time and also enjoys high accuracy. Single shot detectors, e.g., YOLOv2 [41] and SSD [35], directly predict boxes from input images using fully convolutional networks [37]. This single shot design usually achieves higher running speed but has certain descend of accuracy [21]. In this paper, we employ R-FCN [6] as our default object detector considering the balance of speed and accuracy. We also note that our contributions are architecture-independent, and therefore can be applicable for any detection networks mentioned above.

Object Detection in Videos. Video object detection is a relatively new topic and is mainly promoted by ImageNet VID challenge. A large variety of methods [9, 28, 29] focus on associating independent detection results of multiple frames to get more stable and accurate detection accuracy. Different to this, another kind of methods work on feature-level to accelerate detection speed or improve detection accuracy. Shelhamer et al. [46] propose clockwork ConvNets to schedule different layers at different update rates. However, neglecting the evolution of high level features usually lead to worse recognition accuracy. Zhu et al. [63] firstly propose to reuse sparsely sampled CNN features, and propagate the features to adjacent frames via a light-weight flow field. Following this idea, [10, 62] densely aggregate features from adjacent frames to enhance the feature quality at all frames. Recent works [53, 61] combine the merits of [62, 63] and strike a balance between accurate feature aggregation and fast feature propagation modules for all frames. However, all these methods need ground truth boxes in video data for training. We make several contributions for learning feature flow to propagate CNN features between adjacent frames via unsupervised learning. Hence we can transfer arbitrary pre-trained image object detectors to fast and accurate video object detectors using unlabeled video data.

Optical Flow in Video based Applications. Optical flow [7] estimates pixel-level correspondences between two input images, and further captures motion information in videos. Therefore optical flow is widely used in many computer vision tasks such as video interpolation/extrapolation [36], novel view synthesis [60], action recognition [48] and temporal smoothing in dense video processing tasks [3, 14]. We focus on feature prediction of adjacent frames in this paper. While high-level CNN features evolve slowly in video, it's cheep to copy features from nearby frames. [10, 62, 63] employs optical flow to align different frames and then propagate CNN features to adjacent frames, thus improving the accuracy or speed for video processing tasks. We mainly follow this idea in our paper. Accurately estimating flow of high-level features between nearby frames using unlabeled videos is a core part in our video object detection systems.

Predictive Learning. Learning to prediction is an important problem of artificial intelligence, and therefore receives a growing attention in recent years. Researchers have explored a variety of specific problems such as future frame prediction [32, 39, 52], scene parsing prediction [25, 26, 38], feature prediction [51, 54]. The encoder-decoder network [22, 39] is a general architecture for prediction tasks. Novel training techniques like adversarial learning [13, 22] are introduced to learn better predictions beyond traditional L1/L2 regression loss [39, 51]. However, blur is often a problem for these generative techniques. Optical flow based models still show comparable results considering the balance of speed, accuracy and model simplicity [32, 38, 39]. Our problem is also a prediction task, but has slightly different problem settings. The input images of the current and future frames, together with the CNN features of the current frames are known, we attempt to predict CNN features in the future frames. Our work shows promising results for feature prediction using optical flow based method.

Unsupervised Learning of Optical Flow. Usually ground truth motion estimations are not easily to obtain. Therefore it is valuable to exploit rich information in unlabeled videos to learn optical flow. Recently self-supervised learning [24, 55, 57] has been proposed as a novel unsupervised learning method. The key idea is to leverage the inherent structure of raw images or videos to formulate a strong supervision signal for training. Following this idea, [1, 43, 58] learn optical flow from unlabeled videos using assumptions of brightness constancy and spatial smoothness. In this paper, we aim to learn high-level feature flow for fast and accurate video object detectors using self-supervised learning and also make several contributions to learn accurate feature flow in our problem, which demonstrates to be very effective in our experiments.

Knowledge Distillation. Knowledge distillation (KD) [19] is originally proposed to transfer knowledge from large or ensemble networks to a smaller one for efficient deployment. Compared with one-hot labels, softened outputs of the teacher provide extra knowledge of inter-class similarities. However, original KD is only limited to softmax function and classification tasks. Subsequent works [44, 56] attempts to transfer intermediate features of the teacher as 'hint' to train student models. Zagoruyko et al. [59] proposes to transfer spatial attention maps from teacher to student network. Beyond classification task, Chen et al. [4] compress large object detection models into smaller ones. Li et al. [31] employs KD to add new capabilities to an existing model while maintaining performance for old capabilities. Shmelkov et al. [47] employs KD for incremental learning in object detection and avoid catastrophic forgetting problems. Chen et al. [2] leverage rich source-domain knowledge to build target-domain detector in low-shot setting. Inspired by these successful applications of KD, we explore a new task, learning the temporal module of video object detectors from pre-trained image object detectors using unlabeled videos.

3 SEMI-SUPERVISED DFF

In this section, we first give an overview of our proposed semi-DFF in Section 3.1. Then we describe our method in detail, starting with the spatial module, the default object detector R-FCN in static images in Section 3.2. After this, we present the temporal module and describe the detailed architecture of our feature flow network in Section 3.3. Then we introduce our semi-supervised learning framework and the unsupervised feature flow estimation in Section 3.4. Finally we give some important implementation details of our semi-DFF in Section 3.5.

3.1 semi-DFF Overview

Our goal is to learn fast and accurate object detectors in video domain and make full use of unlabeled video data. To achieve this goal, first our video object detector has a spatial module as well as a

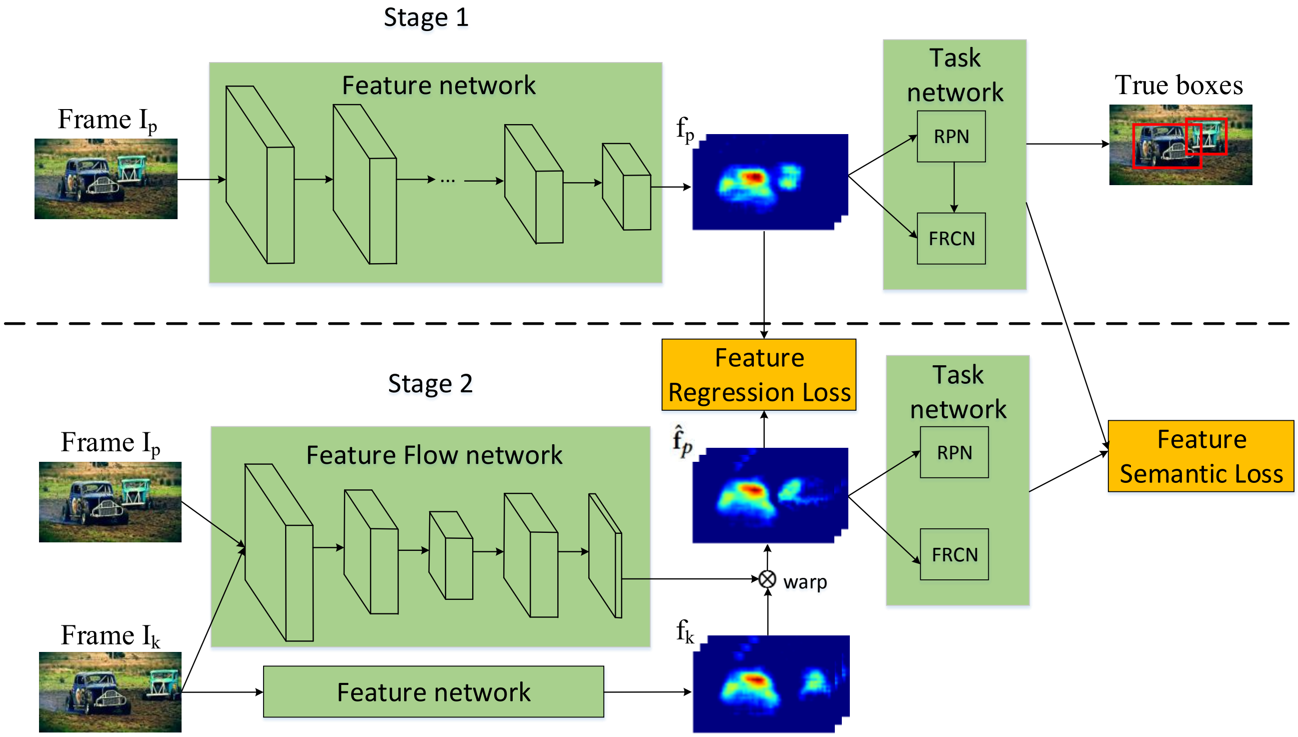

Figure 2: Illustration of our proposed two-stage semi-supervised learning framework. In the first stage, we learn the spatial module using supervised learning with ground truth labeled images. In the second stage, the static image object detector learned in the first stage is regarded as the teacher model. We adopt both feature regression loss and feature semantic loss to learn accurate feature flow using unlabeled videos with the help of the teacher model.

temporal module, and alternately runs two modules at a proper interval. Then we propose a novel semi-supervised learning method to train our detector using unlabeled videos. Our method is a simple and general detection framework which bridges the gap of object detectors in image and video domain, and therefore can easily transfer state-of-the-art object detectors between two domains.

More specifically, we have two decoupled but also closely related modules in our semi-DFF: spatial module and temporal module. The spatial module is a state-of-the-art object detector which can recognize objects in static images. It's accurate but may be too slow if we densely evaluate every frame in a video due to the high computational burden of modern object detectors [21]. High-level CNN feature extraction is the bottleneck for fast detection. However, high-level CNN features usually evolve very slowly due to temporal coherence [46], we thus introduce the temporal module to reuse CNN features for nearby frames in a video. Feature flow [63] is employed to propagate CNN features between adjacent frames in our temporal module. Different from the traditional purely supervised video object detectors [63], our semi-DFF adopts supervised learning for spatial module and unsupervised learning for the temporal module respectively. Hence we can leverage the rich information in unlabeled video data. In the following sections, we will describe the network architecture and semi-supervised learning framework in details.

3.2 Spatial Module: Detection Network

We adopt R-FCN as our default detection network. R-FCN is the state-of-the-art object detection framework for static images considering the balance of running speed and detection accuracy. It can be divided into two parts: image feature extraction network and detection specific networks. With the rapid development of DCNNs [18, 20], hundreds of convolutional layers are employed for strong and robust feature extraction. On top of the feature extraction network, RPN is used to generate proposals in the image with pre-defined multi-scale candidate boxes (anchors in [42]). Position-sensitive maps are built to encode the relative spatial position of objects. Both of RPN and Position-sensitive maps are fully convolutional and nearly cost-free computation. After this, position-sensitive RoI pooling layers further classify proposals and refine their coordinates using position-sensitive maps.

Concretely, given an input frame $\mathbf{I} \in \mathbb{R}^{H_0 \times W_0 \times 3}$, a backbone feature extraction network (e.g., ResNet-101 [18]) is used to obtain CNN features $\mathbf{f} \in \mathbb{R}^{C_l \times H_l \times W_l}$, where C_l, H_l, W_l are the channel, height, width of features output at layer l. In the detection specific networks, object proposals are generated using RPN. Then $k^2(C+1)$ position-sensitive score maps and $4k^2$ position-sensitive regression maps, corresponding to a $k \times k$ spatial grid, are produced for proposal classification (C categories) and (class-agnostic) bounding box regression. Finally, position-sensitive RoI pooling layers are adopted to aggregate position-sensitive maps for final prediction.

3.3 Temporal Module: Feature Flow Network

In our temporal module, we aim to predict high-level CNN features $\hat{\mathbf{f}}_p \in \mathbb{R}^{C_l \times H_l \times W_l}$ of nearby future frame fast and accurately given the current frame (key frames) $\mathbf{I}_k$ and future frame $\mathbf{I}_p$ as well as the high-level CNN features $\mathbf{f}_k$ of the current frame $\mathbf{I}_k$ in a video, and their temporal interval is $\Delta T = p - k$. The temporal interval between key frames is an important parameter in our model. Predicting future CNN features using a light-weight network can largely accelerate our video object detectors. However, it's usually hard to directly hallucinate target CNN features using generative encoder-decoder networks [32, 36]. While high-level features change slowly in a video, we attempt to explicitly model the motion dynamics of nearby frames, and then cheaply propagate CNN features to nearby frames via feature flow.

Therefore estimating accurate feature flow is the key problem in our temporal module T_θ, where θ is the parameters. We employ the modern CNN based optical flow estimation architecture FlowNet [7] (the "Simple" version), and automatically learn to compute the high-level feature flow $\mathcal{F}(\mathbf{I}_k, \mathbf{I}_p)$ using the current frame $\mathbf{I}_k$ and future frame $\mathbf{I}_p$. The estimated feature flow has the same spatial dimensions as the CNN feature maps. Then bilinear interpolation $\mathcal{W}$ is adopted to propagate CNN features simultaneously for all the channels. To better accommodate the amplitudes of CNN features, we multiply the predicted features of all channels with a learned scale field $S(\mathbf{I}_k, \mathbf{I}_p)$ of the same size in an element-wise way. Thus the predicted feature $\hat{\mathbf{f}}_p$ is

$$\hat{\mathbf{f}}_p = T_\theta(\mathbf{I}_k, \mathbf{I}_p, \mathbf{f}_k) = \mathcal{W}(\mathbf{f}_k, \mathcal{F}(\mathbf{I}_k, \mathbf{I}_p)) * S(\mathbf{I}_k, \mathbf{I}_p) \tag{1}$$

where the bilinear interpolation $\mathcal{W}$ is parameter-free and can be differentiated during training [23].

3.4 Semi-Supervised Learning Framework

For training our detector, the proposed two-stage semi-supervised learning framework is illustrated in Figure 2.

In the first stage, we train our spatial module using supervised learning. We assume that ground truth boxes are available for every training images in this stage. Hence we can learn the parameters of R-FCN end-to-end by jointly optimizing RPN and FRCN [11] modules using ground truth boxes. The supervised learning loss consists of two parts:

$$\mathcal{L}_1 = L_{RPN}(P_1, p_1^*, b_1, b_1^*) + L_{FRCN}(P_2, p_2^*, b_2, b_2^*) \tag{2}$$

and the loss function of RPN and FRCN is both the summation of cross-entropy loss and smooth L1 box regression loss [11]:

$$L(P, p^*, b, b^*) = l_{cls}(P, p^*) + l_{reg}(b, b^*) \tag{3}$$

where P and p^* are the predicted probability distribution and ground truth label (one hot label) of anchor box or proposal [42], and therefore $l_{cls}(P, p^*) = -\log P_{p^*}$. b and b^* are the predicted bounding box (bbox) and ground truth bbox respectively. Background proposals are ignored in bbox loss.

In the second stage, we train our temporal module in unsupervised learning with unlabeled videos. Prior works on unsupervised learning of optical flow [58] tries to minimize the loss function of flow warp error between the flow-guided warped image and ground truth image. We can also use this method for unsupervised learning of feature flow. Here, we exploit the pre-trained spatial module R-FCN in the first stage as teacher model to get teacher CNN features $\mathbf{f}_p$ for frame $\mathbf{I}_p$. Then, the teacher CNN features can be directly employed to supervise the predicted features of our temporal module using Euclidean distance loss:

$$\begin{aligned}\mathcal{L}_{reg}(\mathbf{f}_p, \hat{\mathbf{f}}_p) &= \frac{1}{C_l H_l W_l} ||\mathbf{f}_p - \hat{\mathbf{f}}_p||_2^2 \\ &= \frac{1}{C_l H_l W_l} \sum_{c=1}^{C_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} (\mathbf{f}_p(c,h,w) - \hat{\mathbf{f}}_p(c,h,w))^2\end{aligned} \tag{4}$$

However, it's hard to directly minimizing the L1 or L2 regression loss of high-dimensional CNN features (e.g. $1024 * H/16 * W/16$ in our experiments) and also the estimate feature flow may not be suitable for our detection task. Inspired by perceptual loss in [27] and knowledge distillation in [19], we propose to minimize the semantic loss of predicted feature specific to our detection task. which is defined as the summation of RPN and FRCN losses guided by the R-FCN model learned in stage one:

$$\mathcal{L}_{sem} = L_{RPN}(P_{s1}, P_{t1}, b_{s1}, b_{t1}) + L_{FRCN}(P_{s2}, P_{t2}, b_{s2}, b_{t2}) \tag{5}$$

$$L(P_s, P_t, b_s, b_t) = l_{cls}(P_s, P_t) + \lambda l_{reg}(b_s, b_t) \tag{6}$$

where P_s and P_t are the predicted probability distribution of the student and teacher model respectively and T is the temperature parameter introduced in [19] to soften the softmax output:

$$P_s = softmax(\frac{a_s}{T}), \quad P_t = softmax(\frac{a_t}{T}) \tag{7}$$

Here, a_s and a_t are softmax pre-activations. Original softmax is the special case of $T = 1$. Higher temperature T will produce softer probability distribution over classes, and also introduces more noise in learning. We will study the effect of the parameters in ablation experiments. The cross-entropy loss between P_s and P_t is:

$$l_{cls}(P_s, P_t) = -\sum P_t \log P_s \tag{8}$$

In addtion, b_s and b_t are bbox regression results for the student and teacher model, we employ Euclidean distance loss to estimate the semantic loss for bbox refinements:

$$l_{reg}(b_s, b_t) = ||b_s - b_t||_2^2 \tag{9}$$

Different from the supervised learning in the first stage, we adopt unsupervised learning in the second stage without any ground truth labels. We learn the temporal module by forcing the CNN feature predicted to have similar semantic capability as the feature calculated by teacher model. Using semantic loss, we transfer three kinds of knowledge from the teacher model to the student. First of all, **inter-class relationship.** Since our model aims to predict multi-class objects, the student model can learn better inter-class relationship by mimicking the soft target [19] computed by the teacher model for each pre-computed proposal from the teacher. Secondly, **attention maps.** RPN naturally computes heat maps of input images, which represents the probability of foreground at each location. Thus transferring this knowledge in our detection problem actually works in a very similar way of transferring attention which is proposed in [59] for classification problem. Finally, **bbox accurate locations.** We also transfer the knowledge of the bbox regression both in RPN and FRCN due to bbox regression is very important for precise object location [11], and it's also an easier task than classification since bbox refinements are class-agnostic in our model.

Finally, we also experiment with various methods to cooperate these two losses, and find that a two-step optimization give the best results. We first use $\mathcal{L}_{reg}$ to obtain a pretty good initial model. Then we employ $\mathcal{L}_{sem}$ to fine-tune the model, and the task network (RPN and FRCN in spatial module) is also fine-tuned.

3.5 Implementation Details

In our spatial module R-FCN, we adopt ResNet-101 [18] as our backbone network. We resize the input image such that its shorter side is 600 pixels and keep the aspect ratio. The feature stride is reduced from 32 to 16 to produce denser feature maps. A 3×3 convolution is appended to res5c to get the final features with 1024 channels, which is the intermediate feature maps for detection specific network. The first half 512-dimensional of the intermediate feature maps is used for region proposal generation and the second half feature maps for proposal classification and refinement. We mainly follow the hyper-parameters and training details in [6].

In our temporal module, we adopt FlowNetS architecture [7] for feature flow network as default. To keep the same resolution of the flow and CNN features ($\frac{1}{16} \times \frac{1}{16}$ of the original images), additional pooling or convolutional layers with stide of 2 are inserted to the feature flow network. We compute the reverse flow mapping the locations in future frames to the locations in key frames as we want to warp key-frame CNN features. Different from [63], we don't employ any pre-trained FlowNet models in our training by default. Feature flow network is initialized randomly by Normal distribution with standard deviation 0.02. Key frames interval is set to $\Delta T = 10$ by default for the balance of speed and accuracy. In our two-step optimization method, the temperature is set to 1 and also $\lambda = 1$. We adopt RMSprop solver to optimize our model in both steps, and the learning rate is set to 5×10^{-5} by default.

Our implementation adopt the publicly available code of [63] and MXNet deep learning framework [5]. Code will be made publicly available to facilitate future research after publication.

4 EXPERIMENTAL RESULTS

We comprehensively evaluate our method on ImageNet VID dataset. The VID dataset has 3862, 555, and 937 fully-annotated video snippets for the training, validation, and test sets respectively. The frame rate is 25 or 30 fps for most snippets. There are 30 object categories, and they are a subset of the categories in the ImageNet DET dataset. To train R-FCN in static images, we also utilize ImageNet DET training dataset (only the same 30 category labels are used), as a common practice in [29, 63]. Following the protocols in [29, 63], evaluations are performed on the ImageNet VID validation set, and we report our results using the standard mean average precision (mAP) metric over all classes at $IoU = 0.5$ and running speed tested in TITAN X Maxwell GPU.

We evaluate our method in two different experimental settings. First, we only have ground truth boxes for static images. Second, we have true boxes for both static images and video data. In addition, unlabeled videos are available in all experiments. We also conduct ablation experiments in the first experimental setting to analyze deep into our semi-DFF.

Approach	mAP (%)	runtime (fps)
R-FCN frame	59.31	7.5
R-FCN copy	54.12	60
R-FCN flow	55.05	30
semi-DFF reg	57.51	30
semi-DFF sem	58.15	30
semi-DFF	**58.68**	30

Table 1: Performance comparison on the ImageNet VID validation set. All methods use the same R-FCN model which is trained using labeled images in ImageNet DET training set. Our semi-DFF further adopts unlabeled videos in ImageNet VID training set for semi-supervised learning. We compare our semi-DFF with baseline methods described in Section 4.1. The mean average precision over all classes and running speed are shown for a variety of methods.

Approach	mAP (%)	runtime (fps)	labeled videos
R-FCN frame	59.31	7.5	
YOLOv2-416	51.65	67	
SSD300	56.33	45	
DFF [63]	59.54	30	✓
semi-DFF	**58.68**	30	

Table 2: Performance comparison with state-of-the-art methods. All methods use labeled images in ImageNet DET training set except that semi-DFF employs unlabeled videos and DFF [63] needs labeled videos in ImageNet VID training set for training.

4.1 True labels for only images

In this experiment, we only have ground truth labels for static images (totally 53639 training images in ImageNet DET training set of the same category in VID), and this is the default setting in our semi-supervised learning framework. In the first stage, we train R-FCN using the labeled images available via supervised learning. After learning the spatial module, we train the temporal module using unlabeled videos in ImageNet VID training dataset and adopt the settings in 3.5.

Baseline evaluation. We compare our semi-DFF with a variety of baselines and variants as below.

- R-FCN frame. We simply evaluate each frame independently using the trained R-FCN model. This is a strong baseline without any feature propagation module.
- R-FCN copy. In this case, only key frames are evaluated with R-FCN model, The detection results of other frames are just copies of nearby key frames.
- R-FCN flow. The key frames is evaluated by R-FCN model, and CNN features of nearby frames are propagated using the FlowNetS model pre-trained in Flying Chairs [7].
- semi-DFF reg. In this case, we train the temporal module only using feature regression loss in our semi-DFF.
- semi-DFF sem. Here we train the temporal module only using feature semantic loss in semi-DFF.

(a) Feature Prediction Architecture: Our flow-based method is more effective.

Architecture	mAP (%)
generative method	54.79
flow-based method	**57.51**

(b) Feature Regression Loss: Three feature regression losses are explored and L2 loss works best.

Feature Regression Loss	mAP (%)
steady feature loss [24]	54.13
L1 regression loss	55.55
L2 regression loss	**57.51**

(c) Feature Semantic Loss: Ablation study of feature semantic loss.

	Feature Semantic Loss			
inter-class relationship	✓	✓	✓	✓
attention maps		✓		✓
bbox accurate location			✓	✓
mAP (%)	55.54	55.76	57.75	**58.15**

(d) Temperature: Our model is insensitive to T.

Temperature	mAP (%)
$T = 1$	**58.15**
$T = 2$	58.07
$T = 4$	58.16
$T = 10$	57.85
logits regression ($T = \infty$)	57.81

(e) Fine-tune Layers: In our feature semantic loss, we find that training temporal model as well as the task network, other layers frozen, works best.

Fine-tune Layers	mAP (%)
temporal module	57.94
temporal module + spatial module	53.38
temporal module + task network	**58.15**

(f) Training Method: Our two-step optimization method outperforms the joint training method.

Training Method	mAP (%)
joint training	58.08
two-step training	**58.68**

(g) Key-frame Interval: The accuracy and speed trade-offs of our model are shown for a wide range of key-frame intervals.

Interval	mAP (%)	runtime (fps)
5	59.05	20
10	**58.68**	30
15	57.68	34
20	57.35	36

Table 3: Ablations for the second stage learning of our semi-DFF. We use sampled frame pairs from ImageNet VID training set as in [63], test on ImageNet VID validation set, and report mAP and runtime for comparison.

- semi-DFF. Regression loss and semantic loss are employed to learn the temporal module via a two-step optimization method in our semi-DFF, and this is the default model we refer to unless otherwise specified.

The evaluation results of all these methods are show in Table 1. R-FCN frame is a strong baseline and achieves the best results (59.3% mAP) among all these methods. Simple method as R-FCN copy can largely improve inference speed, but brings in large accuracy decrease (5.4% decrease in mAP). R-FCN flow slightly improves R-FCN copy using pre-trained FlowNet to approximate the feature flow. However, feature flow evolves more slowly than optical flow. Our proposed semi-DFF can largely improve detection accuracy by learning better feature flow from unlabeled videos, and also enjoy high running speed (4× speedup and only 0.6% mAP decrease compared to the strong R-FCN frame).

We also explore deep into the semi-DFF. Simply using feature regression loss like prior works on unsupervised optical flow estimation [58] can get a rather good feature flow. However, as we analyse in section 3.4, using regression loss is not easy to learn suitable feature flow for our detection task (1.8 % mAP lower than R-FCN frame). Our proposed semantic loss forces feature flow to preserve the semantic knowledge when propagating, and improves mAP by 0.6%. We then further combine these two losses in a two-step optimization manner to train our temporal module, and can obtain another gain of 0.5%. Note that the gain is significant considering the small gap between the strong baseline R-FCN.

We also compare our semi-DFF with other state-of-the-art methods in Table 2. YOLOv2 [41] and SSD [35] are trained following their original papers using the labeled images, and are evaluated in a densely frame-by-frame manner. These methods are faster than our semi-DFF. On the other hand, our method, based on R-FCN, are especially good at accurate detection while still maintaining real-time speed. Moreover, semi-DFF is independent of detection architecture design, and can further benefit SSD and YOLOv2 by introducing the temporal module. We leave this for the future work. Our method is also closely related with DFF. Here we train DFF with additional labeled videos using 'DFF fix N' method in [63]. We find that DFF only slightly outperform R-FCN frame and semi-DFF given additional labeling in videos. Our method is a very strong competitor when no labels are available in videos.

Ablation experiments. We also conduct comprehensive ablation studies of semi-DFF in this experimental setting in Table 3.

First, we explore the generative encoder-decoder architecture for feature prediction except our flow-based network. To ease the learning of directly predicting future CNN features, we adopt residual learning framework [18] since CNN features of nearby frames are very similar. We use the current and future frames to predict changes (residual) of high-level features between them. We use feature regression loss for training. Results are shown in Table 3a. Generative method obtains meaningful results compared to R-FCN copy, but is inferior to our flow-based method. Estimating motions is more effective for feature prediction between adjacent frames.

We also compare different feature regression methods in Table 3b. Steady feature loss [24] assumes that features change in a similar manner in adjacent time intervals. We thus employ triplet of video frames with equal temporal interval and learn feature flow using this regularization. However, this assumption is too weak than to directly supervise the predicted feature, and we do not observe meaningful results in our experiments. In addition, we also compare L1 and L2 regression, and observe better for the latter.

Then we study deep into our feature semantic loss. We first analyse the effect of each term in our semantic loss. The result is shown in Table 3c. We find that all three kinds of knowledge are crucial for final mAP. Temperature T is another important parameter in our semantic loss, we find that our method is insensitive to it and achieves pretty good results for a wide range of T in Table 3d. For large T (e.g. 10 or ∞), we observe a slightly decrease in mAP. So we set $T = 1$ by default in our experiments. We then explore which layers to train in Table 3e. Only training the temporal module already achieves good results, and fine-tuning the task network (RPN and FRCN in R-FCN) can obtain another 0.2% gain in mAP. However, if we also fine-tune the feature extraction network in spatial module, we find that it's hard to converge and get worse results.

We also explore how to incorporate the two learning loss for our temporal module in Table 3f. We find that our two-step optimization can gain an improvement of 0.6% mAP compared to the joint learning method. In joint training, it's hard to tune the weight between feature regression loss and semantic loss.

Finally, we investigate the speed and accuracy trade-offs for different key-frame intervals in Table 3g. Overall, semi-DFF achieves significant speedup with decent accuracy drop, and smoothly trades in accuracy for speed flexibly. We also notice that too large intervals ($\Delta T >= 10$) can not provide meaningful acceleration. Therefore we recommend the default key-frames interval of 10 frames in our experiments.

4.2 True labels for both images and videos

In this experiment, we have ground truth labels both for static images in ImageNet DET training set (53639 training images) as well as sparse labeled frames for videos in ImageNet VID training set (57834 selected frames are labeled from the raw 3862 video snippets, as in [63]), totally 111473 training images. Similar training method is adopted for all methods as in Section 4.1 except for longer training iterations. Note that labels for sparse frames in VID is only used to expand the training set of R-FCN in semi-DFF, and we do not employ any human labels in our second stage training.

Different to the training set in DET, there are more variant of objects in VID dataset, e.g., rare pose, partial occlusion and motion blur [9, 62]. Given labeled frames in VID dataset, we can learn better spatial module suitable to the video scene. The evaluation results of all these methods are shown in Table 4. We can observe an improvement of 10% $\sim$ 15% in mAP for all methods due to better spatial module. Concretely, our R-FCN baseline achieves 74.1% mAP which is the best among all the methods. Our semi-DFF obtains 72.65%, only 1.5% lower than R-FCN frame but 4× faster. Using feature semantic loss and two-step joint training obtain similar improvement as in section 4.1. Our model also gives a variety of speed and accuracy trade-offs by varying ΔT.

Compared to other state-of-the-art methods, our semi-DFF also obtains meaningful improvements. First, compared to simple baselines, e.g., R-FCN copy and R-FCN flow, our semi-DFF achieves significant improvement (7% and 5.3% mAP higher respectively). Second, we find that single stage object detectors do not perform well on large scale of imagesets as demonstrated on COCO dataset by prior works [35, 41]. Our semi-DFF achieves similar large advantage compared to YOLOv2 and SSD by 12.6% and 7.1% mAP respectively. Third, we also compare our method with state-of-the-art video object detector DFF. Our experimental setting is very suitable for DFF. DFF learns R-FCN and the flow network jointly with labeled videos, and achieves 0.3 mAP higher than ours. Note that DFF and our semi-DFF use the same amount of training data here while DFF employs pre-trained FlowNet on synthetic Flying Chairs dataset [7] for initialization. Thus we also conduct experiments for DFF without initialization for equal competition. DFF only achieves 70.54%, 2.1% mAP lower than semi-DFF. This implies that DFF needs a good initial point for better results. Furthermore, we train semi-DFF++ using more unlabeled video data (3× more training data) in ImageNet VID training set and corresponding more training iterations, and achieve 73.23% mAP, 0.3% higher than DFF. This demonstrates that our semi-supervised learning framework is more promising compared to original DFF.

Approach	mAP (%)	pretrained FlowNet
R-FCN frame	74.10	
R-FCN copy	65.61	
R-FCN flow	67.32	✓
YOLOv2-416	60.00	
SSD300	65.54	
DFF [63]	70.54	
DFF [63]	72.93	✓
semi-DFF reg	71.14	
semi-DFF sem	71.96	
semi-DFF	**72.65**	
semi-DFF (ΔT = 5)	73.59	
semi-DFF (ΔT = 15)	71.31	
semi-DFF (ΔT = 20)	70.42	
semi-DFF++†	**73.23**	

Table 4: Performance comparison on the ImageNet VID validation set. All methods use training data both on ImageNet DET training set (only boxes of categories in VID are used) and ImageNet VID training set (use sampled frames as in [63]). We compare semi-DFF with a variety of state-of-the-art methods as in section 4.1. †: we use more unlabeled videos for training in semi-DFF++.

5 CONCLUSION AND FUTURE WORK

We propose to learn fast and accurate video object detector by decoupling the spatial and temporal modules, and train our detector in a two-stage semi-supervised framework. In the first stage, we learn the spatial module to recognize objects in key frames via supervised learning. In the second stage, we learn the temporal module to recognize objects in adjacent frames fast and accurately via unsupervised learning. Our method is promising for it bridges the gap between object detectors in image and video domain, and can easily learn video object detectors given a pre-trained image object detector. We conduct comprehensive experiments in ImageNet VID dataset and demonstrate the effectiveness of our method.

For future work, we can try much larger dataset in [40] and learn better temporal module with more unlabeled videos. Furthermore, our method can also benefit other video based recognition tasks such as video semantic segmentation, which is also interesting to explore in the future.

REFERENCES

[1] Aria Ahmadi and Ioannis Patras. 2016. Unsupervised convolutional neural networks for motion estimation. In *ICIP*. 1629–1633.

[2] Chao Chen, Yali Wang, and Yu Qiao. 2018. LSTD: A Low-Shot Transfer Detector for Object Detection. In *AAAI*.

[3] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017. Coherent Online Video Style Transfer. In *ICCV*. 1105–1114.

[4] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. 2017. Learning Efficient Object Detection Models with Knowledge Distillation. In *NIPS*. 742–751.

[5] Tianqi Chen, Yutian Li Mu Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. In *NIPSW*.

[6] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *NIPS*. 379–387.

[7] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. 2015. FlowNet: Learning Optical Flow With Convolutional Networks. In *ICCV*. 2758–2766.

[8] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2015. The Pascal Visual Object Classes Challenge: A Retrospective. *IJCV* (2015), 98–136.

[9] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2017. Detect to Track and Track to Detect. In *ICCV*. 3038–3046.

[10] Raghudeep Gadde, Varun Jampani, and Peter V. Gehler. 2017. Semantic Video CNNs Through Representation Warping. In *ICCV*. 4453–4462.

[11] Ross Girshick. 2015. Fast R-CNN. In *ICCV*. 1440–1448.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*. 580–587.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NIPS*. 2672–2680.

[14] Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2017. Characterizing and Improving Stability in Neural Style Transfer. In *ICCV*. 4067–4076.

[15] Guangxing Han, Xuan Zhang, and Chongrong Li. 2017. Revisiting Faster R-CNN: A Deeper Look at Region Proposal Network. In *ICONIP*. 14–24.

[16] Guangxing Han, Xuan Zhang, and Chongrong Li. 2017. Single shot object detection with top-down refinement. In *ICIP*. 3360–3364.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *ICCV*. 2961–2969.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. 770–778.

[19] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*.

[20] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *CVPR*. 4700–4708.

[21] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*. 7310–7311.

[22] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-To-Image Translation With Conditional Adversarial Networks. In *CVPR*. 1125–1134.

[23] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. 2015. Spatial Transformer Networks. In *NIPS*. 2017–2025.

[24] Dinesh Jayaraman and Kristen Grauman. 2016. Slow and Steady Feature Analysis: Higher Order Temporal Coherence in Video. In *CVPR*. 3852–3861.

[25] Xiaojie Jin, Xin Li, Huaxin Xiao, Xiaohui Shen, Zhe Lin, Jimei Yang, Yunpeng Chen, Jian Dong, Luoqi Liu, Zequn Jie, Jiashi Feng, and Shuicheng Yan. 2017. Video Scene Parsing With Predictive Feature Learning. In *ICCV*. 5580–5588.

[26] Xiaojie Jin, Huaxin Xiao, Xiaohui Shen, Jimei Yang, Zhe Lin, Yunpeng Chen, Zequn Jie, Jiashi Feng, and Shuicheng Yan. 2017. Predicting Scene Parsing and Motion Dynamics in the Future. In *NIPS*. 6915–6924.

[27] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *ECCV*. 694–711.

[28] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang. 2017. Object Detection in Videos With Tubelet Proposal Networks. In *CVPR*. 727–735.

[29] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. 2016. Object Detection From Video Tubelets With Convolutional Neural Networks. In *CVPR*. 817–825.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*. 1097–1105.

[31] Zhizhong Li and Derek Hoiem. 2016. Learning Without Forgetting. In *European Conference on Computer Vision (ECCV)*. 614–629.

[32] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P. Xing. 2017. Dual Motion GAN for Future-Flow Embedded Video Prediction. In *ICCV*. 1744–1752.

[33] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*. 740–755.

[34] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. In *ICCV*. 2980–2988.

[35] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. In *ECCV*. 21–37.

[36] Ziwei Liu, Raymond A. Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. 2017. Video Frame Synthesis Using Deep Voxel Flow. In *ICCV*. 4463–4471.

[37] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully Convolutional Networks for Semantic Segmentation. In *CVPR*. 3431–3440.

[38] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. 2017. Predicting Deeper Into the Future of Semantic Segmentation. In *ICCV*. 648–657.

[39] Michael Mathieu, Camille Couprie, and Yann LeCun. 2016. Deep multi-scale video prediction beyond mean square error. In *ICLR*.

[40] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. 2017. YouTube-BoundingBoxes: A Large High-Precision Human-Annotated Data Set for Object Detection in Video. In *CVPR*. 5296–5305.

[41] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: Better, Faster, Stronger. In *CVPR*. 7263–7271.

[42] S. Ren, K. He, R. Girshick, and J. Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*. 91–99.

[43] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. 2017. Unsupervised Deep Learning for Optical Flow Estimation. In *AAAI*. 1495–1501.

[44] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for Thin Deep Nets. In *In Proceedings of ICLR*.

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV* (Dec 2015), 211–252.

[46] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. 2016. Clockwork Convnets for Video Semantic Segmentation. In *ECCVW*. 852–868.

[47] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. 2017. Incremental Learning of Object Detectors Without Catastrophic Forgetting. In *The IEEE International Conference on Computer Vision (ICCV)*. 3400–3409.

[48] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *NIPS*. 568–576.

[49] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*.

[50] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going Deeper with Convolutions. In *CVPR*. 1–9.

[51] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Anticipating Visual Representations From Unlabeled Video. In *CVPR*. 98–106.

[52] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating Videos with Scene Dynamics. In *NIPS*. 613–621.

[53] Tuan-Hung Vu, Wongun Choi, Samuel Schulter, and Manmohan Chandraker. 2018. Memory Warps for Learning Long-Term Online Video Representations. *arXiv:1803.10861* (2018).

[54] Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. 2016. Actions Transformations. In *CVPR*. 2658–2667.

[55] Xiaolong Wang and Abhinav Gupta. 2015. Unsupervised Learning of Visual Representations Using Videos. In *ICCV*. 2794–2802.

[56] Zhenyang Wang, Zhidong Deng, and Shiyao Wang. 2016. Accelerating Convolutional Neural Networks with Dominant Convolutional Kernel and Knowledge Pre-regression. In *Computer Vision – ECCV 2016*. 533–548.

[57] Laurenz Wiskott and Terrence J. Sejnowski. 2002. Slow feature analysis: unsupervised learning of invariances. *Neural computation* (2002).

[58] Jason J. Yu, Adam W. Harley, and Konstantinos G. Derpanis. 2016. Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness. In *ECCV 2016 Workshops, Part 3*. 3–10.

[59] Sergey Zagoruyko and Nikos Komodakis. 2017. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *ICLR*.

[60] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. 2016. View Synthesis by Appearance Flow. In *ECCV*. 286–301.

[61] Xizhou Zhu, Jifeng Dai, Lu Yuan, and Yichen Wei. 2018. Towards High Performance Video Object Detection. In *CVPR*. 7210–7218.

[62] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. 2017. Flow-Guided Feature Aggregation for Video Object Detection. In *ICCV*. 408–417.

[63] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. 2017. Deep Feature Flow for Video Recognition. In *CVPR*. 2349–2358.