# Video representation learning through prediction for online object detection

Masato Fujitake† and Akihiro Sugimoto‡
†Dept. of Informatics, The Graduate University for Advanced Studies, SOKENDAI, Tokyo, Japan
‡National Institute of Informatics, Tokyo, Japan
{fujitake, sugimoto}@nii.ac.jp

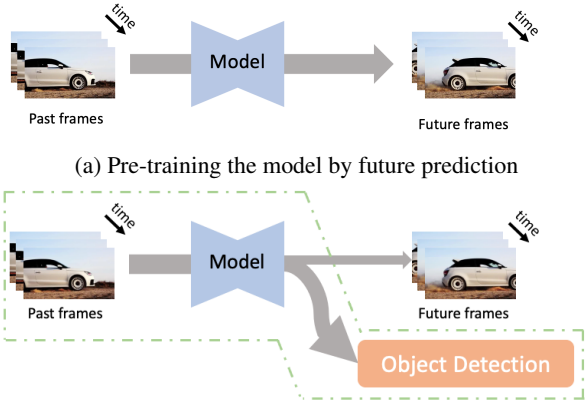
## Abstract

*We present a video representation learning framework for real-time video object detection. Our approach is based on the interesting observation that a powerful prior knowledge of video context helps to improve object recognition, and it can be acquired via learning video representations through stochastic video prediction. Our proposed framework utilizes the stochastic video prediction into object detection so that we first acquire a prior knowledge of videos to have video representations and then adjust them to object detection to improve the accuracy. We validate our proposed method on ImageNet VID and VisDrone-VID2019 datasets to demonstrate the effectiveness of video representation learning via future video prediction. In particular, our extensive experiments on ImageNet VID show that our approach achieves 73.1% mAP at 54 fps with ResNet-50 on commercial GPUs.*

## 1. Introduction

Video object detection, which localizes objects in each frame in a video, is one of the fundamental tasks in computer vision. Different from image object detection, it has different characteristics such as degradation due to motion, and thus poses a new challenge to accurately and stably detect objects. Detecting objects accurately in consecutive frames in a video has been studied along with the improvement of convolutional neural networks [28, 43, 56], and can be broadly divided into offline and online methods. Offline methods [10, 54, 64] have been more studied than online methods [9, 29, 36] because they can make full use of video frames including future ones to improve accuracy. In the live-streaming video scenario such as using surveillance cameras, however, future information is not available to detect objects in a current frame. Online detectors [9, 29, 36] have thus focused on stabilizing feature maps with temporal information from past to the current frames, but have difficulty in achieving the accuracy of offline detectors.

Recently, an online detector [21] has been proposed that predicts the next frame feature map using the past frames and then enriches the current feature map with the information to detect objects at the same time. This approach is more effective than just stabilizing the feature map with past information, since it actively learns how an object moves to predict a feature map at the next frame. While the method improves accuracy, it has the following limitations: (1) Predicting the next frame feature map is too short-term, so the future information is not effectively utilized in the training phase. (2) The accuracy of predicting the next-frame feature map depends on the feature detector, and the feature detector itself is acquired in the training phase where the future information is not effectively utilized. Hence, it is insufficient to leverage future information to train online video detection with the aforementioned method [21].




(a) Pre-training the model by future prediction

(b) Fine-tuning the model to a downstream task

Figure 1: The overview of our proposed framework. To obtain a video representation, we pre-train the model through the future prediction task. Then, we append a detection module to the trained model and transfer it to the detection task. In the inference, only the dotted line area in (b) is used (the future prediction part is not used).

Video pre-training methods are recently studied for video recognition tasks where a video representation is

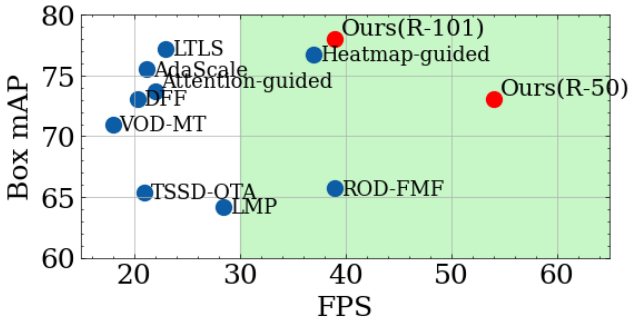



Figure 2: Accuracy-speed trade-off across various online detectors on ImageNet VID (our method is plotted in red, using ResNet backbones). Green color area represents real-time performance.

learned through such as pace prediction [51] or pseudo-label estimation [23]. Predicting future frame images in a video [8,32] can be regarded as a video pre-training method. It can provide a video representation that effectively learns temporal and spatial information, and can be applied to online object detection. However, since existing studies focus only on generating future frames, proposed model structures are task-specific, and their applicability to other tasks is not well investigated.

Based on the above observations, we propose a framework that utilizes stochastic next-frame video prediction [32] into online video object detection. During training to generate future frames, we can obtain the video representation that can effectively stabilize the feature map at the current frame with temporal information from past to longer-term future frames. Figure 1 shows the two-stage training flow of the proposed framework. First, we pre-train the encoder-decoder structure of the detector using stochastic next-frame video prediction [32]. In this pre-training step, the video representation, namely, the context of the video is learned.This step is conducted by self-supervised learning with unlabeled videos, and thus our model can exploit large-scale videos with no annotation. Next, we append the object detection module to the decoder as the downstream task and then fine-tune the model for a detector. We remark that although "pre-training" and "fine-tuning" are often used as applying a model trained on one dataset to another dataset in the same task, they mean in this paper transferring the model to another task. We validate the proposed framework on ImageNetVID [42] and VisDrone-VID2019 [62], and confirm that the proposed framework effectively improves the detection accuracy while maintaining real-time performance. Figure 2 depicts comparison between state-of-the-art models, showing that our model achieves the best trade-off between accuracy and speed.

## 2. Related work

**Object detection in videos.** The main focus of recent methods for video object direction is to improve detection accuracy by exploiting temporal information. These methods are roughly categorized into offline and online.

Since the offline video object detector has access to past and future frames, it exploits all available information to improve the detection accuracy in the current frame. Offline detectors have been proposed by exploiting complementary information [31], future information [5, 26, 27, 52, 64], and relationships between frames [10, 15, 54] and classes [25]. Despite the accuracy improvement, these detectors are slow and do not work in real time.

On the other hand, online video object detectors have recently been studied in applications to live-streaming videos captured by smartphones and robots. In these scenes, only past information can be available, and fast processing speed is required.

To propagate past information to stabilize the current feature map, methods are proposed by utilizing clues such as flow information [63, 65], recurrent neural networks [9, 35, 36], attention mechanism [20, 24, 29, 30, 58, 66], tracking [19, 59], adaptive scaling [11], memories [14], and heatmaps [57]. Aggregating past-frame features in memories using an attention mechanism has also been proposed [29] to improve both the accuracy and run-time. Recently, it has been shown that predicting next-frame feature map improves accuracy while maintaining fast processing time [21].

This paper advances the idea of next-frame feature map prediction [21] by utilizing probabilistic video-frame prediction into online video object detection.

**Next-frame prediction in videos.** To successfully predict future frames from given past frames, two main approaches have been proposed: deterministic prediction and stochastic prediction.

The video prediction by deterministic models generates the next frame by using LSTM [46], ConvLSTM [39], 3D-Convolution [1, 61], and more complex recurrent models [6, 38]. Deterministic models tend to produce blurred images because the output image is the average over all possible images. For this reason, separating a foreground object from the background has been proposed for more accurate generation [4, 16, 48, 50, 55].

Models with stochastic hidden variables such as VAEs, have been proposed [3, 8, 32] to reduce uncertainty in accuracy that increases over time in deterministic models. These models define a prior distribution for a set of latent variables and allow different samples from these latent variables to capture multiple outcomes. It has also been observed that the mean-squared-error loss is based on Gaussian distribution and produces blurred output, so the use of an adversarial loss with GAN is proposed [32].

Our proposed method utilizes stochastic video prediction methods [3, 8, 32] into video object detection via video representation learning. Although CrevNet [61] suggests the video representation learned through video prediction can be directly used for object detection, we show in Section 5 that it is not the case.

## 3. Proposed method

We present our video object detection framework that utilizes Stochastic Adversarial Video Prediction (SAVP) [32]. Our model is modified from SAVP so that it is able not only to predict future frames, but also to detect objects in videos.

### 3.1. Stochastic adversarial video prediction

SAVP [32] combines GANs and VAEs. VAEs produce diverse images while sampling but tend to produce blurry images, while GANs produce clear images but suffer from producing diverse images. Combining VAEs and GANs thus benefits from their complementary strengths.

SAVP consists of a generator $G$ and a discriminator $D$. $G$ is with an encoder-decoder structure conditioned with past frames, and from a current frame and its latent codes at the time, generates the next frame while $D$ optimizes $G$ adversarially. SAVP possesses two distributions for sampling the latent code: the prior distribution and the posterior one, approximated by the learned encoder, in which the posterior distribution is parameterized by a conditional Gaussian distribution using frames of adjacent time steps. At test time, a random latent code $z$ is sampled from the prior distribution independently at each time step.The generator $G$ takes the previous frame and $z$, and then synthesizes a next-step future frame. To generate the next frame, the frame generated in the previous step is fed into $G$ again, and the generation is repeated. During training, $G$ is optimized to predict videos that match the distribution of actual videos using the discriminator $D$. The historical state is accessed via the recurrent neural networks in the generator $G$. SAVP also uses separate discriminators $D$ and $D^{\text{VAE}}$, depending on the distribution used to sample the latent code.

### 3.2. Overall pipeline

The overall pipeline architecture of our proposed framework is depicted in Fig 3. The proposed method consists of five major components. They are (1) recurrent encoder $RE$ for feature extraction from each frame, (2) recurrent decoder $RD$ for generating feature maps from encoded features, (3) detection head for detection,(4) a synthetic head for generating an image from the decoded feature map, and (5) a discriminator for discriminating the generated future frames from the actual future ones.

We have two training steps and one inference step. During "Pre-training", the model acquires the feature representation of videos by self-supervised learning through predicting future frames. The training method is essentially the same as SAVP, but reconstruction of the current frame is also performed for the "fine-tuning" step. Optimization of the generated future frames is performed using a GAN. "Fine-tuning" transfers the model to our downstream task, i.e, detection, using the feature representations acquired in the pre-training step. The difference from pre-training is that the detection head is appended to the decoder.

During "Inference", we do not generate future frames but detect objects. Generating future prediction frames contributes to acquire video representations during training, but is not necessary for detection.

### 3.3. Our prediction and detection network

The task of future prediction takes the current frame $x_t$ (at time $t$) as input with the past $d$ frames $\{c_i\}_{i=t-d}^{t-1}$ as the context, and predicts the future frame $\hat{x}_{t+1}$ at time $t$. However, video object detection requires a detection result in the current frame $x_t$ at the input time $t$ (the output from the decoder must be at the current time $t$). Therefore, to combine video object detection and future prediction, the model must be able to decode feature maps for the current frame and the next frame separately.

We design an encode-decoder network to decode the encoded features at each time independently. The network has the recurrent encoder $RE$ and recurrent decoder $RD$ as shown in Figure 3. The $RE$ and $RD$ are based on ResNet [28] and Feature Pyramid Network (FPN) [33] respectively. Two-layered ConvLSTMs [44] are added to the outputs of each ResNet block (C3, C4, C5, P3, P4, P5). The roles of ConvLSTMs in $RE$ and $RD$ are different. ConvLSTMs in $RE$ are used to generating temporally-aware feature maps [35], whereas those in $RD$ are used to propagate the information over time.

Using the feature map obtained from the encoder at time $t$, the decoder at time $t$ together with the synthetic head at time $t$ first reconstructs the current frame $\hat{x}_t$. Then, it samples the latent code $z_t$ and decodes feature maps with $z_t$ to generate the next frame $\hat{x}_{t+1}$. The states of the recurrent neural network in the decoder are propagated to account for time-stamp information. The decoders at time $t$ and $t+1$ share their weights except the ConvLSTMs states. In order to generate realistic images from the decoder, we append the shared-weighted synthetic heads on top of the output of P3. The synthetic head is a simple network consisting of a convolution layer with 3-size kernel and 2-stride, instance normalization [49], and ReLu, stacked together twice. The output of the final convolution layer is set to 3 dimensions for RGB images. To optimize the generated image, the corresponding ground truth image is resized to the P3 feature-map resolution size.

To enable stochastic sampling for the future frame gen-

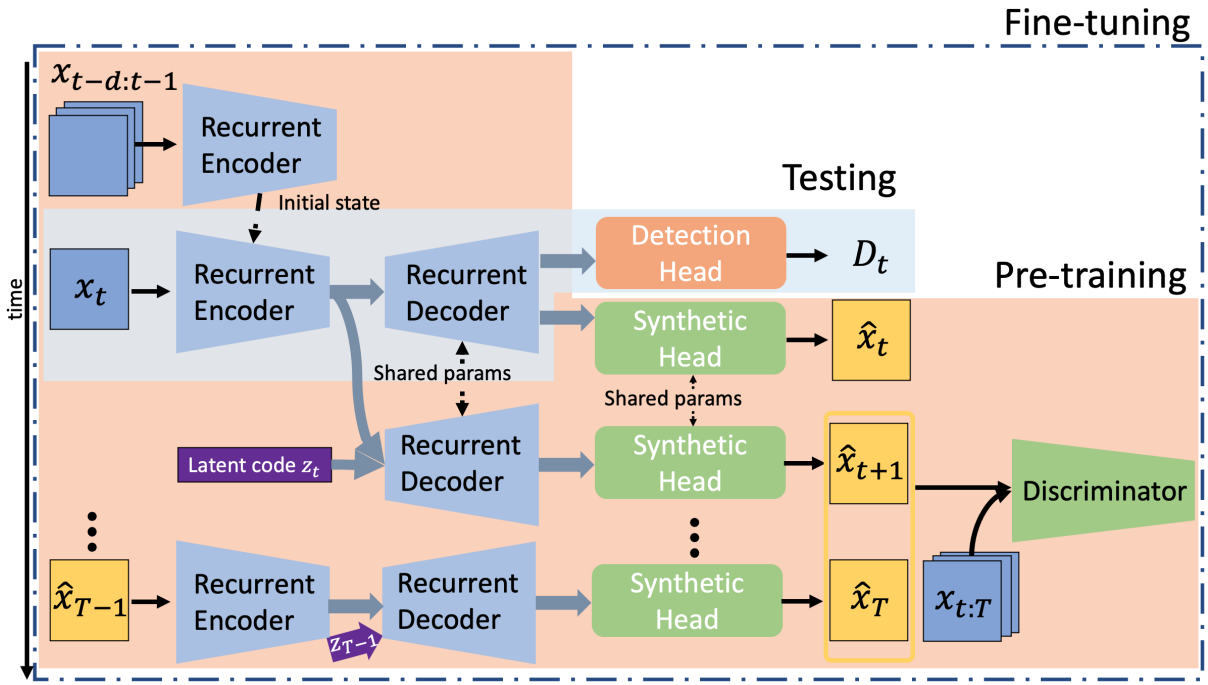



Figure 3: The architecture of the proposed model. The model is pre-trained using the components in the orange-colored area, and then fine-tuned using the whole components. The inference corresponds to blue-colored area.

eration, $RD$ is conditioned on time-varying latent codes. Those codes are sampled at training. Each latent code $z_t$ is a 16-dimensional vector, and is passed through a fully-connected LSTM to facilitate correlations in time of the latent variables. The encoded latent codes are then converted to match the 256 input dimensions of FPN, and added channel-wisely to the all input of FPN during lateral connections. Thus, the latent codes are added to the input of FPN when generating the future frame stochastically, but not when generating the current frame. FPN works as an ordinary object detection module.

### 3.4. Pre-training loss

The current frame image is first reconstructed to enable the model to transfer it to both detection in the fine-tuning step and future frame prediction. Then, the decoder time stamp is increased to generate future frames. The loss function for the pre-training step is almost identical to that for the SAVP training [32]. The only difference is that our loss involves the current frame reconstruction.We use $d = 10$ frames for initialization to predict future as proposed in [32].

The loss function of SAVP is defined with four weights $\lambda_i$ $(i = \{1, \ldots, 4\})$ as follows:

$$\mathcal{L}_{\text{savp}} = \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_{\text{KL}} + \lambda_3 \mathcal{L}_{\text{GAN}} + \lambda_4 \mathcal{L}_{\text{GAN}}^{\text{VAE}}, \quad (1)$$

where $\mathcal{L}_1$ is the L1 norm between the forecasted frames and the ground truth, $\mathcal{L}_{\text{KL}}$ is the KL divergence between the prior and posterior distribution, $\mathcal{L}_{\text{GAN}}$ is adversarial loss for discriminator, and $\mathcal{L}_{\text{GAN}}^{\text{VAE}}$ is analogous to $\mathcal{L}_{\text{GAN}}$ except which use latent codes sampled from the posterior distribution. See [32] for details of these terms. In order to optimize the reconstructed current frame $\hat{x}_t$ from the actual frame $x_t$ at the time $t$, we employ L1 norm. The loss function $\mathcal{L}_{\text{video}}$ for our pre-training is defined as:

$$\mathcal{L}_{\text{video}} = \mathcal{L}_{\text{savp}} + \lambda_5 ||x_t - \hat{x}_t||_1, \quad (2)$$

where $\lambda_5$ is the weights for reconstruction loss of the current frame.

### 3.5. Fine-tuning loss

The pre-trained model will be optimized to be a model for detection. Through fine-tuning, we train the whole weights to fit for detection. Fine-tuning is the same as pre-training except for detection loss and input. The loss for detection defined in FCOS [47] is set to $\mathcal{L}_{\text{det}}$. As a result, the loss function for the current frame in fine-tuning is

$$\mathcal{L}_{\text{finetune}} = \alpha \mathcal{L}_{\text{video}} + \beta \mathcal{L}_{\text{det}}, \quad (3)$$

where $\alpha$ and $\beta$ is the balance weights for $\mathcal{L}_{\text{video}}$ and $\mathcal{L}_{\text{det}}$, respectively.

The difference of input is that in detection, the previous frame may not be obtained due to the timing of video loading, so we select a random value from $[0\ 10]$ (we set $d = 10$) and use the frame corresponding to the value.

### 3.6. Inference step

The encoder and decoder inherit the ConvLSTMs state from the previous frame except for the initial frame, and the detection is performed sequentially using the encoder-decoder with input frames. It is important to note that at the inference step, we neither reconstruct the current frame nor predict future frames. This is because the fine-tuned model already acquires the video representation for detection. Removing reconstruction and prediction functions also contributes to faster inference.

## 4. Experiments

### 4.1. Dataset and Metric

**ImageNet VID [42]:** is a large-scale benchmark for video object detection. It has 30 categories and contains 3,862 training and 555 validation videos with frame rates of 25 and 30 fps. We evaluate our method on the validation set and use the mean average precision (mAP) as the evaluation metric following widely adopted protocols in [63, 64].

**VisDrone-VID2019 [62]:** is a large-scale unmanned aerial vehicle scene benchmark, which is more complex and crowded than ImageNet VID. It includes 56 training, 7 validation, and 16 test video clips with ten categories of object instances from different cities.We use AP, $AP_{50}$, $AP_{75}$, $AR_1$, $AR_{10}$, $AR_{100}$, and $AR_{500}$ metrics for evaluation.

### 4.2. Implementation details

We employ FCOS [47] as the baseline object detector in our proposed model. On ImageNet VID, we use ResNet-50 and ResNet-101 [28] with FPN [33] for the backbone and insert ConvLSTMs as described in Section 3.3. We utilize ResNet-101 as a backbone on Visdrone dataset for a fair comparison with other methods. We follow the hyper-parameters of FCOS [47] and the modifications [2]. The input images are resized to have their smaller side to be 512 pixels on ImageNet VID and 800 pixels on VisDrone-VID2019, respectively.

For pre-training, we follow SAVP [32] with SGD and a batch size of 16 with pre-trained weights of ImageNet. We set $\lambda_1 = 0.25$, $\lambda_2 = 0.0375$, $\lambda_3 = 0.3$, $\lambda_4 = 0.3$, and $\lambda_5 = 0.25$ empirically and use 16 dimensions of latent codes. We utilize the discriminator $D$ as proposed in [32]. We train our model for ten epochs to predict a 10-frame forward future in total, with the learning rate of $10^{-4}$ and $10^{-5}$ in the first six and the last two epochs, respectively.

For fine-tuning, we set $\alpha = 1.0$, $\beta = 1.0$. We then train the pre-trained model for 5 epochs, with the learning rate of $10^{-4}$ and $10^{-5}$ in the first 3.3 and the last one epoch, respectively. Although we trained our model on two RTX 3090 GPUs, we evaluated the speed performance on two 2080 Ti GPUs for a fair comparison with other methods.

Table 1: Performance comparison with the state-of-the-art online and real-time detectors on ImageNet VID `val`.

| Models | Backbone | Base Detector | mAP | FPS | Device |
|---|---|---|---|---|---|
| LMP [66] | MobileNetV2 [43] | RetinaNet [34] | 64.2 | 29 | GTX 1060 |
| TSSD-OTA [9] | VGG-16 [45] | SSD [37] | 65.4 | 21 | Titan X |
| ROD-FMF [21] | MobileNetV2 [43] | SSD [37] | 65.7 | 39 | 2080 Ti |
| VOD-MT [29] | VGG-16 [45] | SSD [37] | 71.0 | 18 | – |
| DFF [65] | ResNet-101 [28] | R-FCN [12] | 73.1 | 20 | K40 |
| AdaScale [11] | ResNet-101 [28] | R-FCN [12] | 75.5 | 21 | 1080 Ti |
| Attention-guided [58] | ResNet-101 [28] | R-FCN [12] | 73.7 | 22 | 1080 Ti |
| LTLS [30] | ResNet-101 [28] | R-FCN [12] | 77.2 | 23 | Titan V |
| Heatmap-guided [57] | ResNet-101 [28] | CenterNet [18] | 76.7 | 37 | – |
| Ours | ResNet-50 [28] | FCOS [47] | 73.1 | **54** | 2080 Ti |
| Ours | ResNet-101 [28] | FCOS [47] | **78.0** | 39 | 2080 Ti |

Table 2: Performance comparison with the state-of-the-art models on VisDrone-VID2019 `test`.

| Methods | AP | $AP_{50}$ | $AP_{75}$ | $AR_1$ | $AR_{10}$ | $AR_{100}$ | $AR_{500}$ |
|---|---|---|---|---|---|---|---|
| Faster R-CNN [41] | 14.46 | 31.80 | 11.20 | 8.55 | 21.31 | 26.77 | 26.77 |
| D&T [19] | 17.04 | 35.37 | 14.11 | 10.47 | 25.76 | 31.86 | 32.03 |
| FGFA [64] | 18.33 | 39.71 | 14.39 | 10.09 | 26.25 | 34.49 | 34.89 |
| FCOS(baseline) [47] | 15.12 | 32.42 | 11.44 | 9.01 | 22.29 | 26.98 | 26.98 |
| Ours | 21.82 | 49.01 | 16.83 | 12.91 | 30.22 | 41.28 | 41.28 |
| DBAI-Det [62] | 29.22 | 58.00 | 25.34 | 14.30 | 35.58 | 50.75 | 53.67 |

### 4.3. Comparison with state-of-the-arts

**ImageNet VID:** We compare our method against several online video object detectors. Table 1 shows their performance comparison. We observe that with ResNet-101 backbone, our method surpasses the strong competitive detector, Heatmap-guided [57] with faster processing speed.

Our method runs at 54 and 39 fps, positioning at the first and second place with ResNet-50 and ResNet-101, respectively. Despite the high-speed processing, it achieves an accuracy of 73.1 mAP, which may be sufficient for reasonable detection accuracy. A comparison between detectors capable of real-time processing confirms that our method runs at high speed while maintaining high accuracy.

In particular, when compared with VOD-MT [29], which is close in accuracy, we see that our method runs more than twice faster. VOT-MT aggregates past feature maps to stabilize detection, and takes time for each frame aggregation. In contrast, our method learns a video representation in advance and transfers it to the detection task, so that no aggregation is required for detection. This difference brings the high accuracy of our method with excellent speed. More detailed comparison with VOT-MT is presented in Section 4.4.

We visualize in Fig. 4 some of our results with those by the original FCOS. We see that our method provides more stable detection than the original FCOS. In particular, our method is found to be robust to blurring and suppresses class switching, achieving higher mAP as seen in Table 3.

**VisDrone-VID2019:** Table 2 shows performance comparison. D&T [19] and FGFA [64] are methods to stabilize the feature map of the still image detector [41] with temporal information. We see that our model significantly improves the accuracy from the baseline and outperforms the compared methods. Our model runs at 23 fps. As a ref-

Table 3: Ablation study of our model.

| Methods | mAP | $mAP_s$ | $mAP_m$ | $mAP_f$ | FPS |
|---|---|---|---|---|---|
| FCOS (baseline) [47] | 68.7 | 79.3 | 68.5 | 43.6 | **56** |
| model w/o prediction (+ConvLSTMs only) | 70.1 | 80.0 | 68.7 | 44.5 | 54 |
| model w/o pre-traing | 71.6 | 83.1 | 69.0 | 47.1 | 54 |
| complete model | **73.1** | **83.3** | **71.7** | **52.7** | 54 |
| complete model w/o GAN | 71.7 | 81.5 | 69.6 | 48.2 | 54 |
| complete model w/o VAE | 70.7 | 80.9 | 69.2 | 44.9 | 54 |

erence, we show DBAI-Det [62] since it is ranked as the first place at the VisDrone-VID2019 competition. Note that DBAI-Det [62] combines heavy backbone [56] and several methods [7, 13] for accuracy, and runs at less than 1 fps.

Fig. 5 visualizes the result on blur scene. We see that our method detect target object robustly.

### 4.4. Detailed analysis on ImageNet VID

To confirm the effectiveness of the proposed method in detail, we conducted ablation studies on the validation set of ImageNet VID. We follow a motion-aware evaluation metric in [64] to evaluate the performance on the categories of slow, medium, and fast objects, where these three categories are divided by their average IoU scores between objects across nearby frames. Slow motion means the case where IoU score is higher than 0.9, and fast motion means that IoU score is lower than 0.7. Medium motion indicates the rest. We note that $mAP_s$, $mAP_m$, $mAP_f$ represent mAP(small), mAP(medium), mAP(fast), respectively.

**Component ablation analysis.** We evaluate the impact of key components of our model on the detection accuracy; see Table 3. Model w/o prediction exploits from past to current frames such as [9] (we append ConvLSTMs only), and model w/o pre-training represents the model is trained for future prediction and object detection simultaneously without pre-training, and the number of training iterations is the same as for pre-training. The lower part of Table 3 shows the models without using VAEs or GANs in the SAVP [32] part. We see that model w/o pre-training outperforms model w/o prediction, meaning that the accuracy is improved by training the recurrent neural network to predict the future, rather than simply propagating features from the past to the present. We also see that our complete model significantly outperforms the model w/o pre-training. This indicates that learning the video representation through prediction and transferring it to detection is a meaningful procedure to improve detection accuracy. This is also supported by the fact that the improved accuracy for fast objects is remarkable because video representations for fast objects are more sensitive changes in context, such as motion. When GANs or VAEs are ablated, the accuracy drops, confirming that they are both important for using long-term future predictions.

**Fair comparison with the same baseline.** To make comparison more fairly with the closely performing method VOT-MT [29], we follow the same configuration of the detector and feature extractor as VOD-MT. Table 4 shows the accuracy and speed comparison of VOD-MT under the same detector, feature extractor, and input frame size. Both methods have improved accuracy from the base detector, but our method achieves higher accuracy and faster processing speed. This is because our method just uses robust feature representations for inference that are learned during training while VOT-MT generates robust feature maps at each inference.

Table 4: Performance comparison of VOD modules with VOD-MT [29] on RetinaNet and ResNeXt-101.

| Methods | mAP | $mAP_s$ | $mAP_m$ | $mAP_f$ | FPS |
|---|---|---|---|---|---|
| RetinaNet [34] | 77.9 | 87.3 | 74.5 | 55.7 | **9.1** |
| VOD-MT [29] | 79.2 | 88.2 | 76.0 | 57.5 | 6.4 |
| Ours | **81.5** | **89.2** | **80.2** | **63.4** | 8.7 |

Table 5: Impact of the number of the future prediction.

| predicted frames ($T$) | 1 | 3 | 5 | 7 | 10 | 15 | 20 |
|---|---|---|---|---|---|---|---|
| mAP | 71.9 | 72.3 | 72.5 | 72.8 | 73.1 | 72.9 | 73.2 |

**Effect of KL divergence.** We change KL loss weight $\lambda_2$ to see how the weighting for VAE affects the detection and generation. Fig. 6 shows under different $\lambda_2$, the detection accuracy and Structural Similarity Index Measure (SSIM) [53]) computed with the ground truth in 10 future frames. When $\lambda_2$ is large (weighting for VAE is large), KL loss prevents the generation as the regularizer (producing poor SSIM). As $\lambda_2$ becomes small, however, the detection accuracy and the generated image become high until a certain point. Then, as $\lambda_2$ becomes even smaller, KL loss does not work well for detection while rendering becomes better until some point and then gradually degraded. Therefore, there is the trade-off and $\lambda_2$ that compromises detection and generation, which should be learned as a well-balanced point. The frame generation accuracy does not increase when the KL loss works well because images are generated stochastically, and they are structurally different from the actual future ones, as seen at the bottom of the row.

**Impact of prediction of future frame.** We investigate how much the long-term future predictions affect the detection accuracy. Table 5 shows the detection accuracy under different number of generated frames during training. We see that the accuracy gradually improves with longer future predictions and becomes saturated with about 10 frame prediction. We confirm that 10 frames are effective and sufficient for long-term prediction.

**Comparison of stochastic and deterministic future predictions.** The proposed method learns feature representation by stochastically predicting what is likely to happen in the future using VAE. We evaluate how stochastic prediction affects the acquisition of video representations with respect to the size of training set (Fig. 7). As a comparison of deterministic prediction, we also show the result without using VAE. Here, we change the ratio of training data against

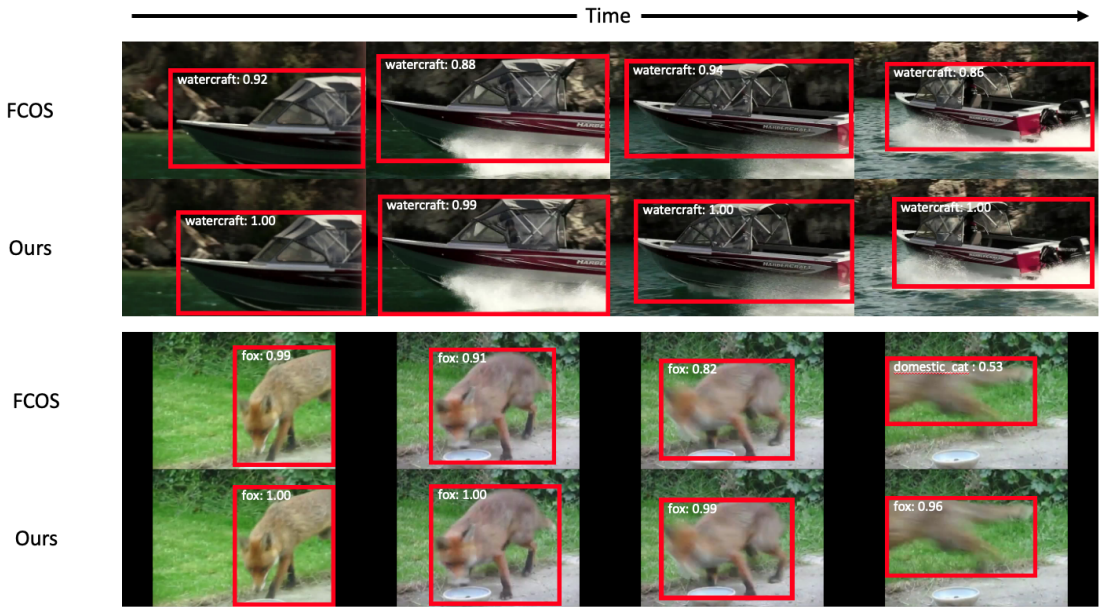



Figure 4: Visualization results on ImageNet VID `val`. The upper row of each sequence corresponds to our baseline, FCOS, and the lower row corresponds to our proposals. Our network learns temporal context to provide significantly stable detection with strict regression across frames.


Figure 5: Visualization results on VisDrone-VID2019 `test`. Our method detects objects with a high degree of confidence even with blur, while the baseline method tends to be unstable due to motion blur in drone images.

the training set of ImageNet VID from 0.5 to 1.0 by 0.25. The number of iterations in training is adjusted not to be affected by the ratio. Fig. 7 shows that stochastic prediction tends to be more accurate as the training data size increases compared to deterministic prediction. We also observe that while the final value of the loss function for deterministic prediction does not change along the training data size, that for stochastic prediction becomes increased. This suggests that stochastic prediction leads to increasing the model capacity for detection by avoiding overfitting that arises for deterministic prediction due to redundant training data [54].

**Impact of pre-training dataset.** Our model does not require an annotated dataset for pre-training. In order to investigate how the size and variation of datasets used for pre-training affect detection accuracy, we exploit two video datasets: YouTube-BB [40] and BDD100K [60]. YouTube-BB is a new large-scale natural scene dataset similar to ImageNet VID and consists of about 380,000 15-20 second videos extracted from publicly available YouTube videos. BDD100K is the largest and most diverse open-driving video dataset to date, consisting of 100,000 videos recorded in different weather conditions such as clear, cloudy, and rainy, and at different times of the day and night.

Table 6: Performance comparison under different pre-training datasets.

| Pre-training dataset | ImageNet VID | YouTube-BB | BDD100K |
|---|---|---|---|
| mAP | 73.1 | (+3.5) 76.6 | (+2.1) 75.2 |

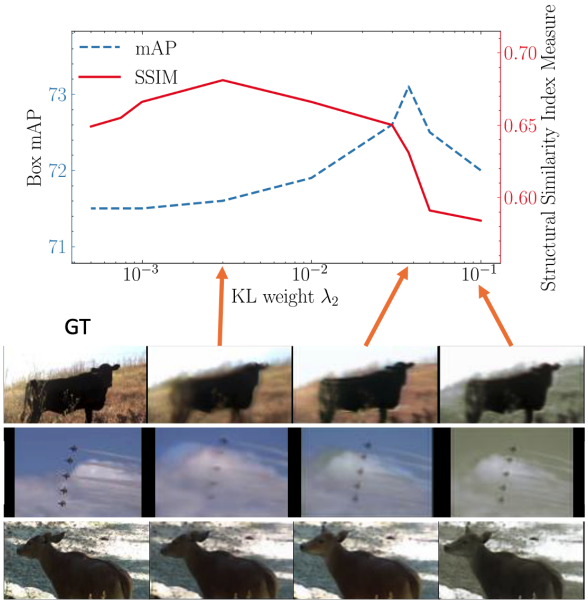



Figure 6: Effect of varying the KL loss weights on the detection and generation accuracy, showing the synthesized 10th frames corresponding to the weights and the corresponding ground truth.

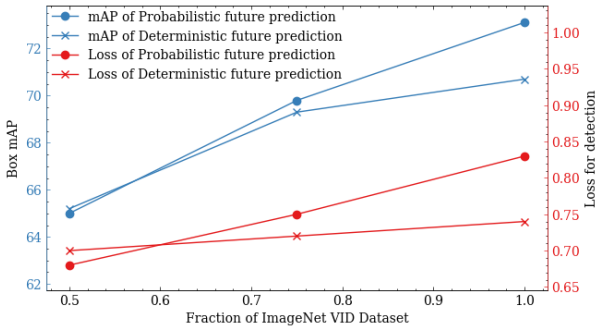



Figure 7: Accuracy impact of different methods of generating future forecasts

Table 6 shows the detection accuracy under different datasets for pre-training. Without any effort, pre-training with YouTube-BB dataset improves the accuracy by 3.5 points. This is a significant gain without increasing any cost of inference. Pre-training with BDD100K also shows the improvement in accuracy by 2.1 points. This interestingly indicates that even pre-training on a completely different-looking dataset improves accuracy. These observations mean that there is great potential for accuracy improvement by using a larger amount of training data for learning video representations through future prediction. A more detailed analysis in this way is left for future work.

## 5. Discussion

CrevNet [61] is a deterministic model to generate future images. It focuses on predicting future frames as accurately as possible by minimizing information loss during feature extraction, but its application to detection is also suggested. Here, we evaluate the accuracy of future image generation and detection to see whether deterministically predicting accurate future images is really required for accurate detection.

Table 7: Prediction accuracy in SSIM on Caltech Pedestrian dataset. Higher SSIM means better prediction accuracy.

| Model | Next-Frame | 3rd | 6th | 9th |
|---|---|---|---|---|
| CrevNet [61] | **0.92** | **0.83** | **0.73** | **0.67** |
| Ours | 0.89 | 0.79 | 0.69 | 0.65 |

Table 8: Detection accuracy in mAP on KITTI

| Methods | Car | | | Pedestrian | | | Cyclist | | | mAP |
|---|---|---|---|---|---|---|---|---|---|---|
| | Easy | Mod | Hard | Easy | Mod | Hard | Easy | Mod | Hard | |
| CrevNet [61] | 91.9 | 91.8 | 86.0 | **89.7** | **83.2** | 75.8 | 87.3 | 80.9 | 72.2 | 84.3 |
| Ours | **95.3** | **93.3** | **91.1** | 88.8 | 80.5 | **75.9** | **89.1** | **81.2** | **73.1** | **85.4** |

We follow the evaluation in CrevNet [61]. To be more specific, we pre-trained our model for video prediction on KITTI [22]. The accuracy of future image generation by the pre-trained model is then evaluated on the Caltech Pedestrian dataset [17] using SSIM [53]. Next, we fine-tuned the model using the detection data on the KITTI. Since the training set of the KITTI dataset provides unlabeled frames of the previous three frames for each annotated detection frame and no future frames, the fine-tuning step of our method is purely for the detection part.

Tables 7 and 8 show accuracy of generating future images on Caltech, and the detection accuracy on KITTI. We see that while our method is less accurate than CrevNet in terms of generating future images in both the short and long-term, it is better in terms of detection; our method significantly outperforms CrevNet. We reason that the video representation learned for generating accurate future images does not match the representation for object detection. Therefore, some uncertainty in video representation brought by stochastic prediction is needed to increase the model capacity for other tasks. Fine-tuning to object detection makes use of this capacity to adjust the video representation to detection.

## 6. Conclusion

We proposed a framework that utilizes stochastic next-frame video prediction into online video object detection. Our model first learns the video representation through future frame prediction and then fine-tunes the representation for object detection via optimizing the appended detector module as the downstream task. By using a single-stage detector, our method achieved 73.1 mAP% on the ImageNet VID dataset with a speed of 54 fps. This pushes forward the trade-off between accuracy and speed.

## References


[1] Sandra Aigner and Marco Körner. Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing gans. *arXiv preprint arXiv:1810.01325*, 2018. 2

[2] aim uofa. Adelaidet. https://github.com/aim-uofa/AdelaiDet. 5

[3] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. In *ICLR*, 2018. 2, 3

[4] Xinzhu Bei, Yanchao Yang, and Stefano Soatto. Learning semantic-aware dynamics for video prediction. In *CVPR*, pages 902–912, 2021. 2

[5] Gedas Bertasius, Lorenzo Torresani, and Jianbo Shi. Object detection in video with spatiotemporal sampling networks. In *ECCV*, pages 331–346, 2018. 2

[6] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *ECCV*, pages 753–769, 2018. 2

[7] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2017. 6

[8] Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *ICCV*, pages 7608–7617, 2019. 2, 3

[9] Xingyu Chen, Junzhi Yu, and Zhengxing Wu. Temporally identity-aware ssd with attentional lstm. *TC*, 50(6):2674–2686, 2020. 1, 2, 5, 6

[10] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. In *CVPR*, pages 10337–10346, 2020. 1, 2

[11] Ting-Wu Chin, Ruizhou Ding, and Diana Marculescu. Adascale: Towards real-time video object detection using adaptive scaling. In *MLSys*, 2019. 2, 5

[12] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, page 379–387, 2016. 5

[13] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 6

[14] Hanming Deng, Yang Hua, Tao Song, Zongpu Zhang, Zhengui Xue, Ruhui Ma, Neil Robertson, and Haibing Guan. Object guided external memory network for video object detection. In *ICCV*, pages 6678–6687, 2019. 2

[15] Jiajun Deng, Yingwei Pan, Ting Yao, Wengang Zhou, Houqiang Li, and Tao Mei. Relation distillation networks for video object detection. In *ICCV*, pages 7023–7032, 2019. 2

[16] Emily Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *NIPS*, page 4414–4423, 2017. 2

[17] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *CVPR*, pages 304–311, 2009. 8

[18] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, pages 6569–6578, 2019. 5

[19] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *ICCV*, pages 3057–3065, 2017. 2, 5

[20] Masato Fujitake and Akihiro Sugimoto. Temporal feature enhancement network with external memory for object detection in surveillance video. In *ICPR*, pages 7684–7691, 2020. 2

[21] Masato Fujitake and Akihiro Sugimoto. Real-time object detection by feature map forecast for live streaming video. In *ICME*, 2021. 1, 2, 5

[22] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 8

[23] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, pages 12046–12055, 2019. 2

[24] Chaoxu Guo, Bin Fan, Jie Gu, Qian Zhang, Shiming Xiang, Veronique Prinet, and Chunhong Pan. Progressive sparse local attention for video object detection. In *ICCV*, pages 3909–3918, 2019. 2

[25] Liang Han, Pichao Wang, Zhaozheng Yin, Fan Wang, and Hao Li. Class-aware feature aggregation network for video object detection. *TCSVT*, pages 1–1, 2021. 2

[26] Mingfei Han, Yali Wang, Xiaojun Chang, and Yu Qiao. Mining inter-video proposal relations for video object detection. In *ECCV*, pages 431–446, 2020. 2

[27] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S. Huang. Seq-nms for video object detection. *arXiv preprint arXiv:1602.08465*, 2016. 2

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 3, 5

[29] Kim Jaekyum, Koh Junho, Lee Byeongwon, Yang Seungji, and Jun Won Choi. Video object detection using object's motion context and spatio-temporal feature aggregation. In *ICPR*, pages 1604–1610, 2020. 1, 2, 5, 6

[30] Zhengkai Jiang, Yu Liu, Ceyuan Yang, Jihao Liu, Peng Gao, Qian Zhang, Shiming Xiang, and Chunhong Pan. Learning where to focus for efficient video object detection. In *ECCV*, pages 18–34, 2020. 2, 5

[31] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang. Object detection in videos with tubelet proposal networks. In *CVPR*, pages 889–897, 2017. 2

[32] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018. 2, 3, 4, 5, 6

[33] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017. 3, 5

[34] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 5, 6

[35] Mason Liu and Menglong Zhu. Mobile video object detection with temporally-aware feature maps. In *CVPR*, pages 5686–5695, 2018. 2, 3

[36] Mason Liu, Menglong Zhu, Marie White, Yinxiao Li, and Dmitry Kalenichenko. Looking fast and slow: Memory-guided mobile video object detection. *arXiv preprint arXiv:1903.10172*, 2019. 1, 2

[37] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37, 2016. 5

[38] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016. 2

[39] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *NIPS*, page 2863–2871, 2015. 2

[40] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *CVPR*, pages 5296–5305, 2017. 7

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI*, 39(6):1137–1149, 2017. 5

[42] 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. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 2, 5

[43] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 1, 5

[44] Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, pages 802–810, 2015. 3

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[46] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, pages 843–852. PMLR, 2015. 2

[47] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019. 4, 5, 6

[48] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, pages 1526–1535, 2018. 2

[49] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 3

[50] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *ICLR*, 2017. 2

[51] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *ECCV*, pages 504–521, 2020. 2

[52] Shiyao Wang, Yucong Zhou, Junjie Yan, and Zhidong Deng. Fully motion-aware network for video object detection. In *ECCV*, pages 542–557, 2018. 2

[53] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4):600–612, 2004. 6, 8

[54] Haiping Wu, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Sequence level semantics aggregation for video object detection. In *ICCV*, pages 9217–9225, 2019. 1, 2, 7

[55] Yue Wu, Rongrong Gao, Jaesik Park, and Qifeng Chen. Future video synthesis with object motion prediction. In *CVPR*, pages 5539–5548, 2020. 2

[56] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017. 1, 6

[57] Zhujun Xu, Emir Hrustic, and Damien Vivet. Centernet heatmap propagation for real-time video object detection. In *ECCV*, pages 220–234, 2020. 2, 5

[58] Yanni Yang, Huansheng Song, Shijie Sun, Yan Chen, Xinyao Tang, and Qin Shi. A feature temporal attention based interleaved network for fast video object detection. *JAIHC*, 2021. 2, 5

[59] Chun-Han Yao, Chen Fang, Xiaohui Shen, Yangyue Wan, and Ming-Hsuan Yang. Video object detection via object-level temporal aggregation. In *ECCV*, pages 160–177, 2020. 2

[60] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, pages 2636–2645, 2020. 7

[61] Wei Yu, Y. Lu, S. Easterbrook, and S. Fidler. Efficient and information-preserving future frame prediction and beyond. In *ICLR*, 2020. 2, 3, 8

[62] Pengfei Zhu, Dawei Du, Longyin Wen, Xiao Bian, Haibin Ling, Qinghua Hu, Tao Peng, Jiayu Zheng, Xinyao Wang, Yue Zhang, et al. Visdrone-vid2019: The vision meets drone object detection in video challenge results. In *ICCVW*, pages 227–235, 2019. 2, 5, 6

[63] Xizhou Zhu, Jifeng Dai, Xingchi Zhu, Yichen Wei, and Lu Yuan. Towards high performance video object detection for mobiles. *arXiv preprint arXiv:1804.05830*, 2018. 2, 5

[64] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, pages 408–417, 2017. 1, 2, 5, 6

[65] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *CVPR*, pages 4141–4150, 2017. 2, 5

[66] Zhifan Zhu and Zechao Li. Online video object detection via local and mid-range feature propagation. In *HuMA*, page 73–82, 2020. 2, 5