IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 September 2023, accepted 24 October 2023, date of publication 30 October 2023, date of current version 3 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3328341

RESEARCH ARTICLE

DiffusionVID: Denoising Object Boxes With Spatio–Temporal Conditioning for Video Object Detection

SI-DONG ROH, (Graduate Student Member, IEEE), AND KI-SEOK CHUNG, (Member, IEEE)
Department of Electronic Engineering, Hanyang University, Seoul 04763, South Korea

Corresponding author: Ki-Seok Chung (kchung@hanyang.ac.kr)

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korea Government (MSIT) (Development of Intelligent Edge Computing Semiconductor For Lightweight Manufacturing Inspection Equipment) under Grant 2021-0-00131.

ABSTRACT Several existing still image object detectors suffer from image deterioration in videos, such as motion blur, camera defocus, and partial occlusion. We present DiffusionVID, a diffusion model-based video object detector that exploits spatio-temporal conditioning. Inspired by the diffusion model, DiffusionVID refines random noise boxes to obtain the original object boxes in a video sequence. To effectively refine the object boxes from the degraded images in the videos, we used three novel approaches: cascade refinement, dynamic coreset conditioning, and local batch refinement. The cascade refinement architecture progressively extracts information and refines boxes, whereas the dynamic coreset conditioning further improves the denoising quality using adaptive conditions based on the spatio-temporal coreset. Local batch refinement significantly improves the inference speed by exploiting GPU parallelism. On the standard and widely used ImageNet-VID benchmark, our DiffusionVID with the ResNet-101 and Swin-Base backbones achieves 86.9 mAP @ 46.6 FPS and 92.4 mAP @ 27.0 FPS, respectively, which is state-of-the-art performance. To the best of the authors' knowledge, this is the first video object detector based on a diffusion model. The code and models are available at https://github.com/sdroh1027/DiffusionVID.

INDEX TERMS Conditioning, coreset, diffusion model, spatio–temporal, video object detection.

I. INTRODUCTION

Video object detection (VOD) is one of the most fundamental areas of computer vision that aims to detect objects in a temporally continuous sequence of images. With the increasing popularity of mobile phones, drones, cars, and action cameras, and the widespread use of social media platforms such as YouTube, Facebook, and Tiktok, the role of object detection in videos has become critical. However, most existing object detectors for still images cannot achieve sufficiently high accuracy in the real world, mainly because of their vulnerability to image degradation, such as motion blur, occlusion, and camera defocus. To overcome the limitations of still image detectors, existing VOD methods retrieve reference information by tracking the movement of objects or computing the similarity between objects.

The associate editor coordinating the review of this manuscript and approving it for publication was Sudhakar Radhakrishnan.

The attention mechanism was invented for language modeling [5], [6] and is widely used in computer vision, including image classification [7], [8], restoration [9], [10], generation [11], [12], [13], and object detection [8], [14] tasks. In the field of VOD, the use of attention mechanisms has also been studied to model the relationships of object-level features [15], [16], [17]. The performance of VOD was significantly improved by aggregating information from global and local frames. However, because they are based on two-stage object detectors such as Faster-RCNN [18], their performance heavily depends on the quality of the initial object suggestions extracted from a region proposal network (RPN). To address this shortcoming, pixel-level attention methods have been investigated [1], [2], [19]. They performed pixel-level attention between the feature pixels of the current image and those of the reference image, such that each current feature pixel has more pertinent information and makes a better region proposal. To reduce the computational

TABLE 1. Comparison of DiffusionVID and previous methods.

Methods	RCNN-based [1], [2]	DETR-based [3], [4]	**DiffusionVID (ours)**
Sparse proposals?	No	Yes	**Yes**
Utilize global information?	Yes	Yes	**Yes**
Costly pixel-level attention?	Yes	Yes	**No**
Adjustable latency vs accuracy trade-off at inference?	No	No	**Yes**

cost, they leveraged sparse style of pixel-level attention, but suffered from a low inference speed because of the computation of a large number of feature pixels generated per image. Afterward, DETR-based methods achieved high performance utilizing transformer-like detection architecture and deformable attention [3], [4], [19]. However, they also suffered from low inference speed because of the use of costly pixel-level attention.

Recently, diffusion models have been applied to various vision domains [12], [13], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. Among them, DiffusionDet [29] is the first diffusion model applied to the object detection domain. It achieved high detection performance with a small number of object queries and its the number of queries can be freely adjusted in the inference stage. However, despite these advantages, no attempt has been made to introduce the generative model paradigm for video object detection domain. Furthermore, although diffusion models in generative tasks utilize conditioning mechanisms from various sources to obtain high-quality results [21], [30], [31], [32], no such attempt has been made in the object detection domain. In a video object detection task, global information can be collected across images that can be used to condition the sampling process. Therefore, we propose DiffusionVID to overcome previous limitations. The main contributions of our method are as follows:

- We propose an object-centric cascade refinement structure based on multiple self-refinement modules to efficiently gather information on object proposals. Object queries are initialized from random boxes and progressively improved using feature pixels in their box locations, eventually becoming more robust than the results of the traditional region proposal network.
- We introduce dynamic coreset conditioning (DCC), which combines the coreset concept and attention mechanism to generate condition vectors that support the box refinement (reverse diffusion) process. We build a spatio-temporal coreset of a video sequence and generate condition vectors of object queries, ensuring good performance and low computational cost. To determine the optimal architecture, several existing conditioning mechanisms are evaluated.
- Local batch refinement (LBR) is introduced to maximize inference speed. By removing local attention stages that inhibit data parallelism, local frames can be processed in parallel in a single GPU, maximizing GPU utilization of the inference stage.
- We propose a first diffusion model-based approach for video object detection. Similar to previous diffusion models, our model can improve the quality of the detection results by exploiting additional reverse diffusion processes.

Our method achieves both low computational cost and high accuracy by applying the diffusion model paradigm and coreset-based conditioning mechanism. The comparison of DiffusionVID and existing methods is shown in Table 1. Experiments on ResNet-101 and Swin Transformer backbones show that our method achieved up to 86.9 mAP and 92.4 mAP on the popular ImageNet-VID validation dataset, respectively, with inference speeds of 46.6 FPS and 27.0 FPS, respectively, providing the best performance and speed balance. Furthermore, increasing the number of samplings with an additional execution time further increased the accuracy to 87.1 mAP and 92.5 mAP. Based on these results, we expect our study to set a new baseline for the challenge of video object detection.

II. RELATED WORKS

A. VIDEO OBJECT DETECTION

VOD is more challenging than still image object detection because of image degradation, such as motion blur, camera defocus, partial occlusion, and rare poses, caused by object and camera dynamics. To address this problem, early video-object detectors offered an intuitive approach. They predicted visual motion and use it to aggregate features between the current frame and its neighbors. Some methods used optical flow information to aggregate information from spatially adjacent feature pixels [33], [34], [35]. References [36] and [37] tracked object-level motion to refine the detection results. Reference [38] used deformable convolution to capture the pixel-level dynamics of features in videos.

Most existing video object detectors use R-CNN-based architectures. They extracted object-level features using a region proposal network (RPN) and employed several methods to model the relations between objects. This type of framework leverages an object-level attention mechanism, in which object features are aggregated using a memory structure that collects information from adjacent frames [2], [15], [16], [17]. However, these approaches rely heavily on the performance of the RPN, which occasionally generates false-positive or false-negative suggestion and fails to accurately model the relations. To address this limitation, recent methods improved feature pixels by exploiting pixel-level attention [1], [2], [19] and deformable attention [3], [4]. However, pixel-level attention deals with hundreds and thousands of feature pixels collected per frame to gather information

from surrounding frames. The amount of computation further increases when using recently proposed multilevel feature architecture, such as feature pyramid networks [39], which may limit the applicability.

B. DIFFUSON MODEL

Diffusion models [12], [13], [20] were initially developed in the field of image denoising; however, they are now actively utilized in various fields such as image synthesis [12], [13], [21], [22], [23], video synthesis [24], [25], and other tasks [26], [27], [28]. In contrast, DiffusionDet [29] is the first diffusion model applied to the object detection domain; it detects objects by refining boxes of arbitrary location and size, unlike existing object detectors that use learned queries specialized for detecting specific locations [14]. However, although there are several diffusion models for generative tasks such as video synthesis, there is no relevant research in the field of video object detection. To the best of our knowledge, our method is the first diffusion model-based video object detector.

We now briefly describe the formulation of diffusion models. The operation of diffusion models consists of the forward (diffusion) and reverse (sampling) processes. The forward process defines the iterative addition of small Gaussian noises to an original data sample x_0, which is formulated as

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where $t \in \{0, 1, \ldots, T\}$ is the diffusion time step, and $\beta_t \in (0, 1)$ is a fixed variance schedule that controls the step size. A sample with arbitrary time step x_t can be obtained in a closed form as

$$q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{2}$$

where $\alpha_t := 1 - \beta_t$, $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$, and $q(x_T)$ is an isotropic Gaussian distribution when T is sufficiently large.

The reverse (sampling) process is a generative process that progressively restores the original sample x_0 from noisy input x_t. According to Bayes' theorem, it is found that the posterior $q(x_{t-1}|x_t)$ is a Gaussian distribution. However, since the reverse step $q(x_{t-1}|x_t)$ is intractable, we train a neural network f_θ by minimizing the training objective as follows:

$$\mathcal{L} = \frac{1}{2}||f_\theta(x_t, t) - x_0||^2. \tag{3}$$

The clean sample x_0 can be reconstructed from x_T by applying the updating rule T times using f_θ [12]. According to [13], the number of sampling steps can be reduced to less than T when using the updating rule:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\cdot\epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\epsilon_\theta(x_t, t) + \sigma_t\epsilon_t, \tag{4}$$

$$\epsilon_\theta(x_t, t) = \frac{x_t - \sqrt{\bar{\alpha}_t}\cdot f_\theta(x_t, t)}{\sqrt{1-\bar{\alpha}_t}} \tag{5}$$

where ϵ_θ is the noise prediction. Our method also uses this approach to improve the computational efficiency.

III. PROPOSED APPROACH

A. PRELIMINARIES

DiffusionVID aims to detect objects in a given image frame of a video sequence. A video sequence with N image frames is denoted as $V \in \mathbb{R}^{N\times H\times W\times 3}$, where an RGB image with ith frame is $I_i \in \mathbb{R}^{H\times W\times 3}, i \in \{1, 2, \ldots, N\}$. The detection outputs of an image I_i and N^q queries consist of class predictions $c_i \in \mathbb{R}^{N^q\times N^c}$ and bounded box regressions $b_i \in \mathbb{R}^{N^q\times 4}$, where a bounding box coordinate of the n^{th} query $b_{i,n} \in \mathbb{R}^4$ consists of the center position (c_x, c_y), width and height (w, h). Since we aim to solve the object detection task using the diffusion model, we initialize the data samples of the i^{th} frame with a set of noise boxes ($x_T = b_i^0$), and design a neural network f_θ which refines b_i so that they become ground truth boxes ($x_0 = b_i$). An overview of the proposed method is presented in Fig.1.

B. QUERY INITIALIZATION

This subsection introduces the initialization of object-level queries for box candidates. The initial queries are obtained by extracting features from the interior regions of arbitrary boxes b_i^0, eventually covering the the most of the area of an image frame. The initial queries are enhanced in cascade refinement stages (III-C) and can be decoded to obtain the box coordinates and the class output. The initial object queries $z_i^0 \in \mathbb{R}^{N^q\times D}$ in an image I_i that exists at certain box coordinates b_i^0 can be initialized as follows:

$$z_i^0 = \mathcal{E}(b_i^0, f_i), \tag{6}$$

$$f_i = \mathcal{F}(I_i), \tag{7}$$

where $f_i \in \mathbb{R}^{H'\times W'\times D}$ is a feature map of the image frame. We extract a high-dimensional feature map from an image using a feature extractor $\mathcal{F}$ and image I_i. Here, $\mathcal{F}$ can be a backbone, such as an ImageNet-pretrained ResNet or Swin-Transformer, which are used in various computer vision tasks. Next, we describe a space-specific extractor $\mathcal{E}$. We applied RoI-Align using the initialized boxes b_i^0 and feature map f_i to extract the regional information of the objects. The pooled results are then averaged in the spatial direction to generate the initial object queries z_i^0 where $z_{i,n}^0 \in \mathbb{R}^D$ is a query of a box $b_{i,n}^0$.

C. CASCADE REFINEMENT

Because initial queries are extracted from arbitrary regions, the information in each query is not sufficient for precise box refinement. To address this issue, we propose a novel method called cascade refinement. The cascade refinement ensures that each query targets an object by using iterative self-refinement stages. self-refinement enhances queries by aggregating information from other queries and referencing the information of areas within the predicted bounding boxes. Enhanced queries can localize and classify objects

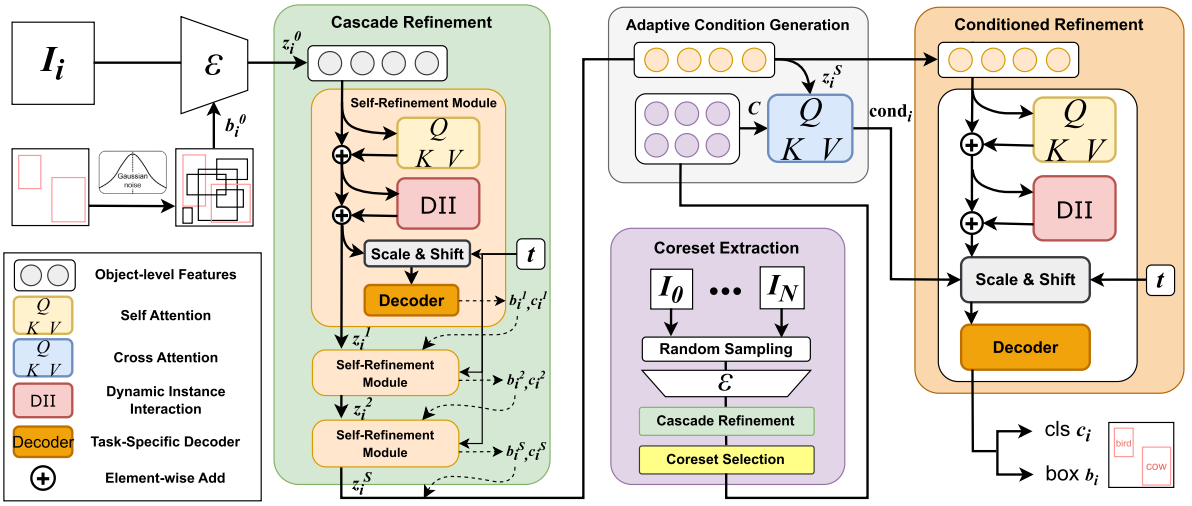


FIGURE 1. Overview of the proposed DiffusionVID. DiffusionVID is based on a diffusion model, taking noise boxes as input and outputting classes and boxes pointing to objects. The encoder inputs random boxes from Gaussian noise and outputs initial object queries. The queries are then processed through cascade refinement, which consists of multiple self-refinement modules. To generate adaptive condition vectors, the cross-attention is conducted between the refined queries and a spatio-temporal coreset. Conditioned refinement further refines object queries using adaptive condition vectors. The decoder outputs the final object class and box coordinates using final object queries.

in an image and allows the subsequent dynamic coreset conditioning (III-D) method to operate with a better data distribution.

1) SELF-REFINEMENT MODULE

A self-refinement module takes queries, bounding boxes, and a feature map as inputs and outputs an improved queries set z_i^s:

$$z_i^s = \mathcal{S}^s(z_i^{s-1}, b_i^{s-1}, f_i, t), \quad s \in \{1, \ldots, S\} \tag{8}$$

where s is the number of the self-refinement stage, b_i^{s-1} denotes a set of bounding boxes, f_i is a feature map, and t is the diffusion time step.

Each self-refinement stage operates as follows. First, the module takes a query set z_i for n boxes in an image and computes the self-attention for the query set. Then, utilizing f_i and b_i, 7×7 RoI-Align is performed to obtain the RoI features. The RoI feature has detailed pixel-level fine-grained features corresponding to each box region. Next, to enhance queries with RoI features, a special type of operation called dynamic instance interaction [40] is applied. In this process, each query references the RoI feature corresponding to the previously predicted box. Each RoI feature is multiplied by the parameters to generate an enhanced query. The parameters are dynamically extracted from the query using fully connected layers. Subsequently, it takes the diffusion time step t as the input and uses linear operations to generate time embeddings and coefficients to normalize (scale and shift) the query. This allows for a multistep reverse process according to the time step, as in the DDIM [13] method.

Finally, decoding is processed to obtain refined results from the enhanced queries. The decoder consists of heads that predict the bounding boxes b_i^{s+1} and classes c_i^{s+1} of the objects, respectively:

$$\{b_i^{s+1}, c_i^{s+1}\} = \mathcal{D}^s(z_i^s), \tag{9}$$

where b_i^{s+1} is used in the next self-refinement stage.

D. DYNAMIC CORESET CONDITIONING

The performance of the cascade refinement can be limited by image deterioration because it uses only the current image information. Inspired by the coreset concept, we propose a new method called dynamic coreset conditioning (DCC). The goal of the DCC is to improve the quality of the refinement process by generating and exploiting customized condition vectors for each query based on the summarized video information.

1) CONSTRUCTING CORESET

Coreset construction is finding a small set of data points while preserving the basic statistical properties of the original dataset. The coreset approach has been studied previously in the field of active learning [41]. They selected a coreset from additional real-world datasets and trained it to avoid overfitting and reduce the computation required for training. In VOD, referencing every object query is computationally inefficient because a significant amount of information redundancy exists. We generate a coreset of objects within the entire video to produce a compact set of features while avoiding excluding a small number of distinctive objects.

We construct a global coreset of object queries as in [42]. First, a global object query set U is generated from N frame images. To exclude the background, we filtered N_k queries per frame based on the decoded class probabilities for a total of $N \times N_k$ queries. Then, we randomly select one query out of all the queries and move it to coreset C for

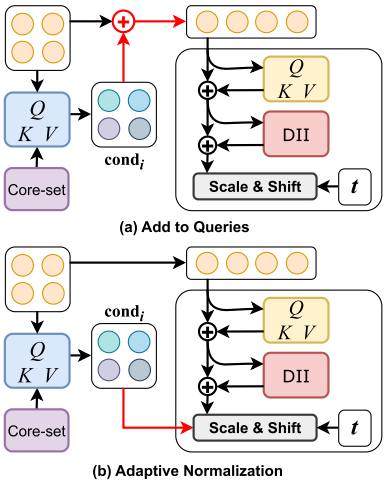


FIGURE 2. Comparison of Conditioned Refinement mechanisms.

initialization. Next, we use the Euclidean distance between queries to select the next query of C based on the following formula:

$$\arg\max_{x \in U} d_{x,C}, \tag{10}$$

where $d_{x,C}$ is the distance between x and C, defined as the distance between x and any sample closest to x among all samples in C. The selected query is then moved from U to C. We iterate this process a number of times corresponding to the size of C we are targeting to obtain a coreset that implies video information. However, the process of constructing a coreset involves computing distances and sorting, which can incur significant overhead, particularly as the size of set U increases and the frequency of updating the coreset increases. Therefore, instead of using the full N frames, we choose N_{samp} randomly selected image frames to prepare U. For fast inference, we reuse the initial coreset C for all frame inferences without updating it.

2) ADAPTIVE CONDITION GENERATION

A coreset contains diverse types of information; however, each query attempts to detect a single object. Therefore, it is important to select the most appropriate information for each query for effective conditioning. Therefore, we propose an adaptive selection method based on the attention mechanism as follows:

$$\text{cond}_\text{i} = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{11}$$

where $\text{cond}_\text{i} \in \mathbb{R}^{N_q \times D}$ represents the condition matrix of self-refined queries z_i^S. $Q \in \mathbb{R}^{N_q \times D}$ and $V \in \mathbb{R}^{N_q \times D}$ are derived from the linear transformation of z_i^S. $K \in \mathbb{R}^{N_q \times D}$ is derived from the linear transformations of the coreset C. Cross-attention searches and gathers information customized for each query to generate robust condition vectors, which are used in the following conditioned refinement.

3) CONDITIONED REFINEMENT

Once the condition vectors are generated, they are given as inputs to the conditioned refinement module to further improve the queries. conditioned refinement works similarly to self-refinement (self-attention, Dynamic Instance Interaction, and normalization) but is modified to inject condition vector information into the queries. This can be implemented in a various ways: `Add`, `Concat`, and `Adaptive Norm`. Among them, `Concat` has the disadvantage of incresing the hidden dimension of the query and thus doubles the subsequent conditioned refinement stages. Therefore, we discarded `Concat` and investigated `Add` and `Adaptive Norm`. The two methods are compared in Fig. 2. `Add` is conducted by adding adaptive conditions to the immediate queries z_i^S. For `Adaptive Norm`, we modify the normalization process of conditioned refinement: use cond_i instead of t to get the shift vector. The performances of the two methods are compared in the ablation study section (Table 6).

E. LOCAL BATCH REFINEMENT

Recent works [2], [15], [16], [17] have utilized cross-attention with multiple local frames to obtain spatio-temporal information from the previous and future. The utilization of more local frames results in larger performance gains. However, because a local frame must be prepared for each frame, multiple current frames cannot be inferred together, thereby limiting parallel inference. By contrast, our method utilizes the same prepared coreset for all frame inferences, allowing for additional parallelization. We propose a local batch refinement (LBR) method that exploits intra-GPU batch parallelism to improve inference speed. As shown in Fig. 3, the previous methods are inefficient because the same frame features are computed multiple times during the inference process. However, our proposed LBR bundles multiple current frames into a "local batch" to perform adaptive condition generation and conditioned refinement operations in bulk, reducing redundant computation and increasing per-GPU utilization.

IV. EXPERIMENTS

In this section, we evaluate the proposed DiffusionVID on the widely used ImageNet-VID dataset and conduct ablation studies to assess the importance of each component in our method. To compare the detection performance, we reported the mean average precision (mAP) using an intersection over union (IoU) threshold greater than or equal to 0.5.

A. IMPLEMENTATION DETAILS

1) DATASETS

We conducted experiments using the DiffusionVID dataset, which consists of 3862 videos for training and 555 videos for validation. As in [33], The ImageNet-DET and ImageNet-VID training sets are combined in order to obtain sufficient samples for the training set. Among the 200 classes in the ImageNet-DET dataset, we used 30 that overlapped with the

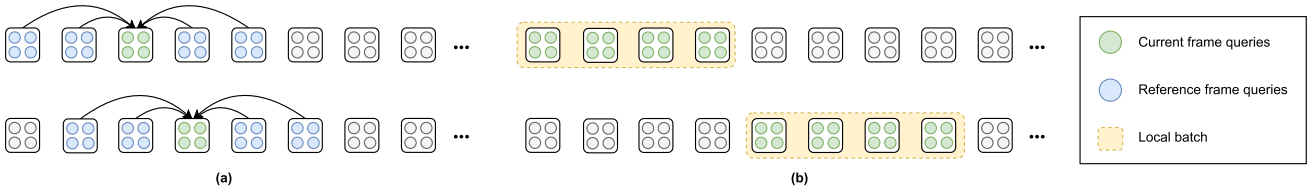


FIGURE 3. Comparison of batch inference methods of VOD. (a) Previous methods require adjacent frames of the current frame, which limits intra-GPU batch-level acceleration. (b) Proposed Local Batch Refinement infers boxes on a multi-frame batch basis.

TABLE 2. Accuracy and speed comparison of the video object detection methods on the ImageNet-VID validation set. 'x4' means inference with 4 sampling steps (default is x1). The inference speed is measured on an RTX 3090 GPU. *reports the results of papers.

Methods	Backbone	Base Detector	Params (M)	mAP	Inference Runtime (ms)
FGFA [33]	R101	R-FCN	-	76.3	-
MANet [36]	R101	R-FCN	-	78.1	-
THP [35]	R101+DCN	R-FCN	-	78.6	-
STSN [38]	R101+DCN	R-FCN	-	78.9	-
OGEMN [1]	R101+DCN	R-FCN	-	80.0	-
SELSA [16]	R101	Faster-RCNN	85.6	80.3	123.7
RDN [15]	R101	Faster-RCNN	161.4	81.8	93.1
MEGA [17]	R101	Faster-RCNN	164.5	82.9	121.8
MAMBA [2]	R101	Faster-RCNN	-	84.6	110.3*
DAFA [42]	R101	Faster-RCNN	161.5	84.5	108.1
VSTAM [19]	R101	Faster-RCNN	-	86.2	95.2*
TransVOD Lite [3]	R101	Deformable DETR	-	80.5	31.0*
TransVOD++ [3]	R101	Deformable DETR	-	82.0	-
PTSEFormer [4]	R101	Deformable DETR	61.8	88.1	314.0
DiffusionVID	R101	DiffusionDet	96.8	86.9	21.5
DiffusionVID(x4)	R101	DiffusionDet	96.8	**87.1**	41.3
SELSA [16]	X101	Faster-RCNN	-	83.1	-
RDN [15]	X101	Faster-RCNN	-	83.2	-
MEGA [17]	X101	Faster-RCNN	201.3	84.1	-
MAMBA [2]	X101	Faster-RCNN	-	85.4	-
DAFA [42]	X101	Faster-RCNN	198.3	85.9	143.2
YOLOV [4]	MCSP	YOLOX	-	85.0	22.7*
VSTAM [19]	SwinB	Deformable DETR	-	87.6	-
TransVOD Lite [3]	SwinB	Deformable DETR	101.4	90.1	51.7
TransVOD++ [3]	SwinB	Deformable DETR	138.9	90.0	670.3
DiffusionVID	SwinB	DiffusionDet	138.6	92.4	37.0
DiffusionVID(x4)	SwinB	DiffusionDet	138.6	**92.5**	56.7

ImageNet-VID dataset. For the ImageNet-VID, 15 frames per snippet were selected to avoid redundant information in adjacent frames. For the evaluation, the ImageNet-VID validation set is utilized rather than the test set because the ground truth is not available.

2) MODEL

We experimented with two backbone models: ResNet-101 and Swin-Base. After the backbone, we constructed a feature pyramid network (FPN) with 3-layer and 256 hidden dimensions. Lastly, we build detection head which consists of three cascading self-refinement stages and one dynamic conditioned refinement stage. Our detection head has 53.8M parameters and costs 14.25 GFLOPs.

3) TRAINING DETAILS

The experiments are performed with 4 RTX 3090 GPUs. We trained two batches for each GPU; therefore, the total batch size is eight. For each batch, one current frame image and four reference frame images are prepared. Reference frames are randomly extracted from the same video sequence as that of the current image. The reference frame of the ImageNet-DET can not be obtained; therefore, we duplicate the current frame and use it as a reference frame. The same augmentation is performed on both the current and reference frames. According to [16], random resizing and cropping, horizontal flipping, and photometric distortion are applied. For random cropping, the image is cropped to include at least one object to ensure balance between the foreground and background objects. The initial learning rate is set to 1e-4, and the backbone learning rate is 1e-5. The weight decay is 1e-4. For the ResNet-101 backbone, the model is trained for 130,000 iterations, and the learning rate decreases by a factor of 10 at the $80{,}000^{th}$ and $120{,}000^{th}$ iterations. A model with the Swin-Base backbone is trained for 70,000 iterations, and the learning rate decreases by a factor of 10 in the $40{,}000^{th}$ and $60{,}000^{th}$ iterations. We warm up both models for the first 18,000 iterations to ensure training stability. For the ResNet-101 backbone, we used an ImageNet-pretrained backbone, and the FPN and detector parts were trained from scratch. For the Swin-Base backbone, we used COCO-pretrained DiffusionDet because the Transformer-based model requires more data samples for training.

B. STATE-OF-THE-ART COMPARISON

Table 2 shows a comparison of the accuracy of the proposed DiffusionVID and existing state-of-the-art methods. For a fair comparison, all the methods were compared without applying

TABLE 3. Net effects on the accuracy (mAP) by adopting various combinations of proposed modules. *a/b* stages means *a* Self-Refinement stages with *b* Conditioned Refinement stages.

Model	DCC	Stages	mAP	Runtime(ms)
DiffusionDet	✗	6/0	79.8	22.9
	✗	4/0	79.4	20.2
DiffusionVID (ours)	✓	3/3	86.0	24.1
	✓	2/2	85.9	21.3
	✓	3/1	**86.9**	21.5

TABLE 4. Net effects on the accuracy (mAP) and the runtime by adopting various sources of conditions.

Condition	(a)	(b)	(c)	(d)
Local Frame Queries		✓	✓	
Global Coreset			✓	✓
mAP	79.4	83.8	86.0	**86.9**
Runtime (ms)	20.2	54.1	55.0	21.5

TABLE 5. Comparision of the global coreset updating strategy in the inference stage.

Update Strategy	Per-batch Update	Presampling		
N_{samp}	24	12	24	36
mAP	86.5	86.5	**86.9**	86.6
Runtime (ms)	27.0	20.9	21.5	22.1

any post-processing methods, except for non-maximum suppression (NMS). The top part of the table compares the models based on ResNet-101 backbone. The following methods are listed: FGFA, MANet, THP, and STSN, based on motion prediction; SELSA, RDN, MEGA, and DAFA, based on object-level attention; and MAMBA, OGEMN and VSTAM, based on pixel-level attention. In general, attention-based methods outperform than motion-based methods. TransVOD Lite, TransVOD++, and PTSEFormer are the latest deformable DETR-based methods, which are based on deformable attention modules that simultaneously learn the pixel location for reference and the attention weight. In the experiment using the ResNet-101 backbone shown at the top of Table 2, our method outperforms most of the compared methods with a score of 86.9 mAP. While PSTEFormer surpasses our detection performance, our method is far superior in terms of the speed-accuracy trade-off given that our inference speed is 14.6 times faster (21.5 ms vs 314.0 ms). Because our method refines boxes using a backward DDIM process, we can refine the box over multiple time steps greater than the default value of 1. When the box is refined over 4-time steps, our method improves by 0.2 mAP to 87.1 mAP.

The bottom part of Table 2 shows the results using more powerful backbones. The ResNeXt [43], Swin-Transformer [8], and MCSP which is used for one-stage detector [44] are added for comparison. Our DiffusionVID with Swin-Base backbone achieves an accuracy of 92.4 mAP and can get an additional 0.1 mAP increase with the 4-step option to achieve 92.5 mAP, which indicates the best performance among all the compared methods, achieving state-of-the-art performance. Compared to the other best Swin-Base backbone-based competitor, TransVOD Lite, we show a performance improvement of up to 2.5 mAP and a 28.4% reduction in execution time, indicating the superiority of our method.

C. ABLATION STUDIES

We evaluate the speed and accuracy of the DiffusionVID methods based on the ResNet-101 backbone.

1) COMBINATION OF MODULES

Table 3 shows the accuracy and speed results when various combinations of the methods in the DiffusionVID (self-refinement and conditioned refinement modules) are applied. First, the "Stages" column shows the number of self-refinement and conditioned refinement modules. If only the self-refinement modules are present, the model becomes similar to the DiffusionDet model. Note that when using the DCC, at least one Conditioned Refinement module must be present. When using only the Self-Refinement modules, there is a slight improvement in accuracy with an increase in the execution time as the number of modules increases. However, accuracy enhancement is limited because of the absence of spatio-temporal information. On the other hand, utilizing DCC considerably improves the accuracy. Reducing 3/3 stages to 2/2 results in a negligible difference in accuracy, but it does reduce the execution time significantly. While maintaining the total number of stages at four, changing from 2/2 to 3/1 stage combinations enhances the accuracy by 1.0 mAP with the same execution time. This indicates that, as the number of self-refinement stages increases, the quality of the queries utilized in the coreset improves, leading to generating a better condition vector. When comparing the 4/0 and 3/1 stage combinations to determine the effect of the DCC, the accuracy differs by 7.5 mAP.

2) SOURCE OF THE CONDITION

Table 4 compares the accuracy and speed with respect to the source of the queries collected for coreset construction. For models using local frame queries, we collected query information from -12 to 12 frames relative to the current frame. If both local and global sources are used, two-stage multi-head attention is used for adaptive condition generation. The experimental results show that using only the global coreset (d) results in a higher detection accuracy than using both the local and global coresets (c). In addition, when the local frame queries are used, local batch refinement will be unavailable. Consequently, the runtime will increase significantly. Therefore, we choose (d) as our default because it shows the best trade-off between accuracy and speed.

3) CORESET UPDATE STRATEGY

Table 5 presents the comparison results in terms of both accuracy and speed with respect to the size of U and the coreset update strategy. `Per-batch Update` refreshes the coreset using local batch queries generated as the local

TABLE 6. Comparison of Conditioned Refinement mechanisms.

Cond. Mechanism	None	Add	Adaptive Norm
mAP	79.4	86.1	**86.9**
Runtime (ms)	20.2	21.6	21.5

TABLE 7. Speed comparisons with respect to various local batch sizes. The time unit is ms.

Local Batch Size	1	4	8	**12**	16
Runtime (ms)	44.2	24.1	21.5	**20.9**	20.9

batch is processed. We update the coreset according to [42], obtaining the union of the existing coreset C and the newly generated candidates to create a new set U, which is then used to create a new coreset. The local batch size was set as 12. `Presampling` initializes the coreset before the first local batch inference starts and then does not update it any further. N_{samp} is the number of frames randomly collected from the video to create U. `Per-batch Update` takes the longest execution time (27.0 ms), where as `Presampling` significantly reduces the computation time, with an execution time of 22.1 ms or less execution time. The execution time tends to increase as N_{samp} increases. The highest detection accuracy is achieved when $N_{samp} = 24$, so we choose it as the default value.

4) CONDITIONED REFINEMENT MECHANISMS

Table 6 compares the accuracy and speed of different designs of the conditioned refinement module. `None` is identical to the self-refinemnet module without using the condition vector. The designs of `Add` and `Adaptive Norm` are described in Section III-D. The accuracy of the `Adaptive Norm` is 0.8 mAP better than that of `Add`, with a slight decrease in the execution time owing to the reduced computation of time embedding generations. Therefore, we choose `Adaptive Norm` as the default.

5) LOCAL BATCH SIZE

The parallel processing capability of GPUs can be better exploited by local batch refinement. Table 7 lists the execution times for the various local batch sizes. Note that local batch refinement only improves the execution speed of the algorithm. Therefore, the detection accuracy remains constant regardless of the local batch size. A local batch size of one implies the absence of local batch refinement. The execution time typically decreases with a larger local batch size but remains consistent above 12. When comparing the execution times with and without local batch refinement, we observe a time reduction of up to 52.7%.

D. QUALITATIVE ANALYSIS

Fig. 4 shows the visualization results for the two snippets. In snippet (a), the fast movement of the squirrel causes motion blur, and simultaneously, the position of the squirrel causes partial occlusion which obscures the cat. In snippet (b), the fast movements of the fox and cat cause motion blur and rare poses. For each snippet, the first row shows the results of Faster-RCNN, and the second row shows the results of DiffusionDet. Both models are still image object detectors. The third row shows the results of MEGA, which is a previous SOTA method. The last two rows show the results of the proposed DiffusionVID model based on two different backbones (ResNet101 and Swin-Base, respectively).

Compared to the two single frame-based models (Faster-RCNN and DiffusionVID), video object detectors generally perform better because they utilize spatio-temporal information. However, still image-based models fail to detect or misclassify objects, where as the video object detectors can classify and localize more accurately.

By comparing SOTA video object detectors (MEGA and DiffusionVID), we observe in snippet (a) that MEGA continues to misclassify the squirrel as a cat, mainly because of the overlap of the cat and the squirrel, while DiffusionVID is more robust. For snippet (b), MEGA tends to output overly high confidence scores for objects, and sometimes outputs false detection results (classifying a bowl as a car). DiffusionVID, however, generally outputs relatively low confidence scores but achieves better localization and classification results.

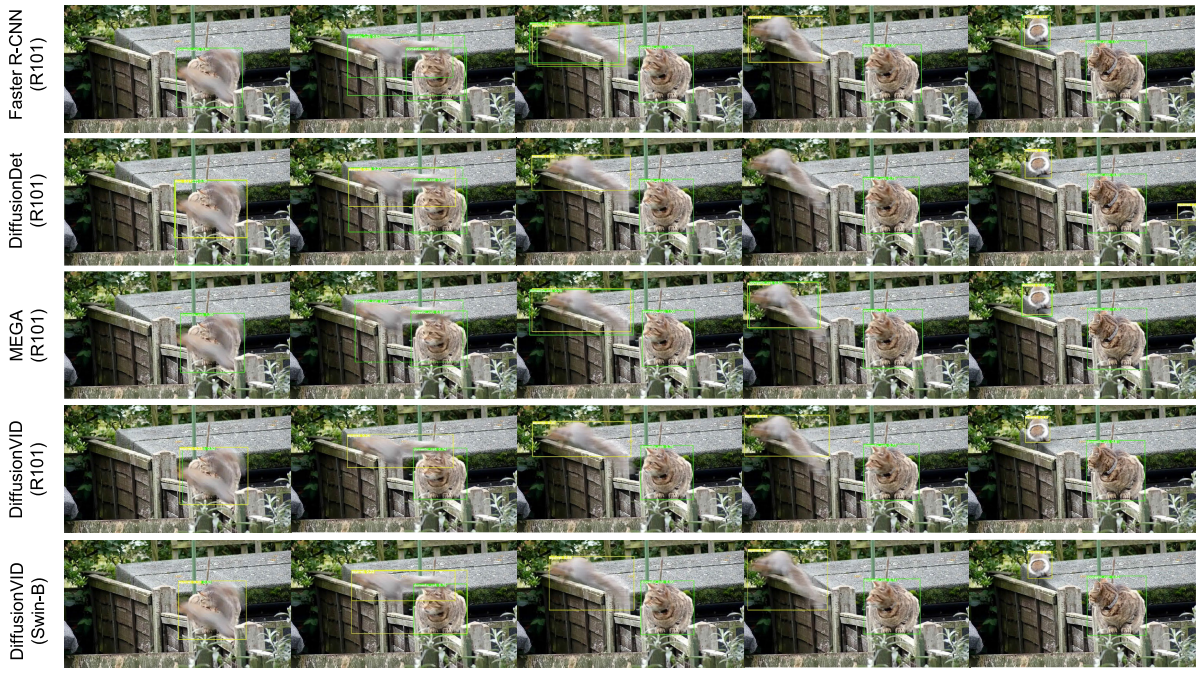
E. EVALUATION ON YOUTUBE OBJECTS DATASET

To further investigate generalization performance, we further evaluated DiffusionVID on the YouTube-Objects (YTO) dataset [48]. The YTO v2.2 dataset contains 155 videos with 720,152 frames and 10 categories, which are subset classes of ImageNet VID. Each class contains between 9 and 24 videos. The dataset is sparsely annotated; 6,087 frames were annotated with 6,975 bounding boxes. Among the annotated frames, the number of test set annotations is 1781. To evaluate the performance, we reused parameters of DiffusionVID trained on the ImageNet-DET and VID datasets without additional training. The object localization accuracy is evaluated using CorLoc [49]. CorLoc is calculated by dividing the number of correctly localized images by the number of ground-truth images. Mean average precision (mAP) results are also reported for detection performance evaluation.

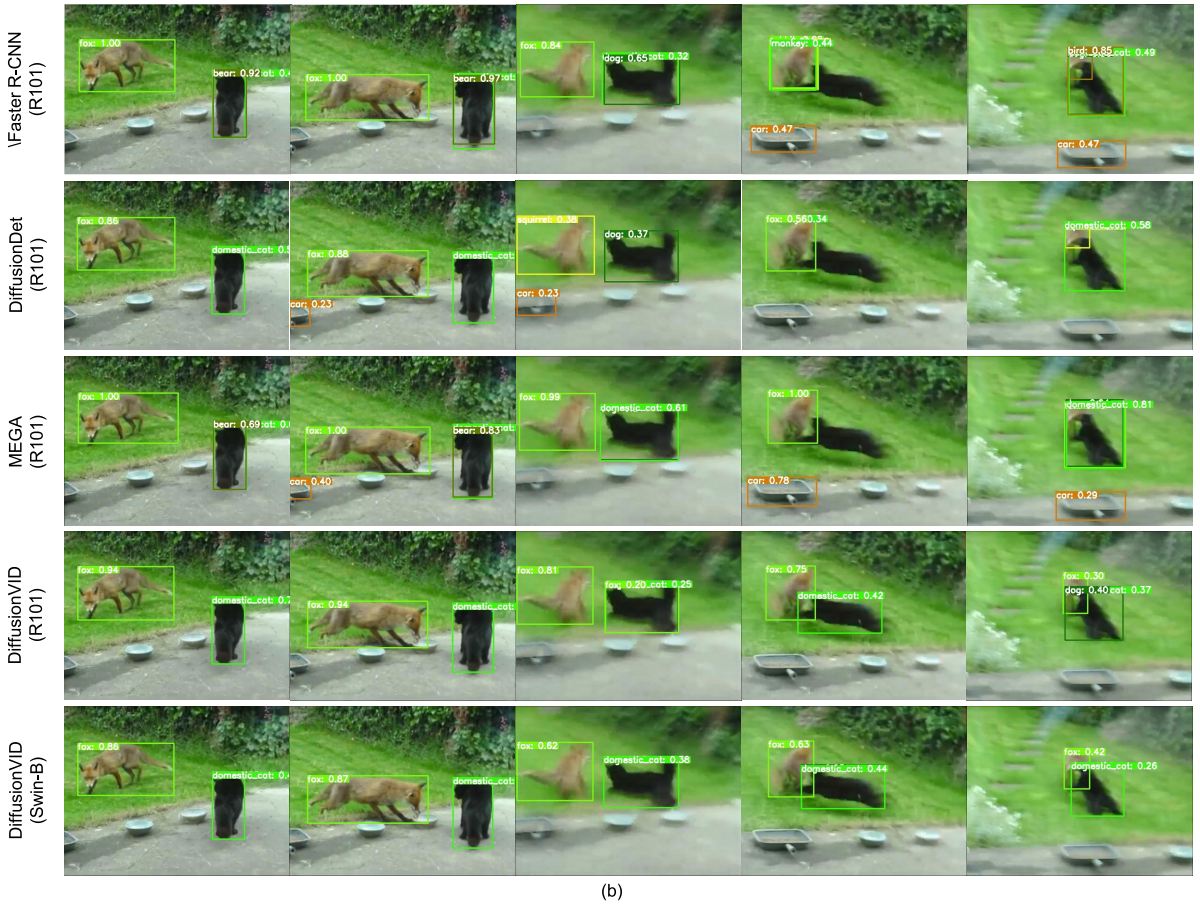
Results are shown in Table 8. For fairness, we compared previous methods with the same ResNet-101 backbone if possible. Faster-RCNN, MEGA, DAFA [17], [42], [50] use ResNet-101 backbone, while others [45], [46], [47] use weaker feature extractors (HoG and GoogleNet). DiffusionVID shows similar or better localization performance (92.8 in CorLoc) compared to most methods while outperforming all other methods in detection (89.7 mAP). When a stronger backbone (Swin Base) is used, our method outperforms other methods by a large margin in both localization and detection.

V. LIMITATIONS

Despite of superior performance of our method, our method has limitations. When there is significant motion blur



(a)



(b)

FIGURE 4. **Qualitative Results. Each row shows the detection results by the model indicated on the left. The detection results are presented in time-ordered frames in the form of bounding boxes with classes and confidence scores.**

or partial occlusion, our method may extract low-quality semantic information from the image and fail to relate the current frame objects and global information. This can lead to the production of low-quality conditions and poor detection

TABLE 8. Localization and detection performance on the YouTube-Objects Dataset. CorLoc and mean average precision (mAP) is used for evaluation metric.

Method	airplane	bird	watercraft	car	cat	cattle	dog	horse	motorcycle	train	CorLoc (Avg.)	mAP
Kwak [45]	56.5	66.4	58.0	76.8	39.9	69.3	50.4	56.3	53.0	31.0	55.7	-
TCN [46]	94.1	69.7	88.2	79.3	76.6	18.6	89.6	89.0	87.3	75.3	76.8	-
T-CNN [47]	91.8	98.7	85.4	95.0	92.2	100	95.7	93.4	93.9	84.2	93.0	-
FasterRCNN	97.8	100	94.9	96.9	76.4	87.3	75.1	78.8	82.6	85.4	87.5	84.3
MEGA [17]	98.9	100	94.4	98.0	89.1	100	91.3	88.3	83.6	87.3	93.1	87.9
DAFA [42]	99.4	100	96.1	98.8	89.1	100	93.1	97.2	80.8	88.6	94.3	88.3
DiffusionVID_R101	99.4	100	96.1	98.5	86.1	96.5	86.1	84.0	91.6	89.9	92.8	89.7
DiffusionVID_Swin	100	100	100	99.5	93.9	100	96.0	84.7	92.5	99.3	96.6	95.1

(a) (b)

FIGURE 5. We show two failure cases. Boxes are the predictions of DiffusionVID with ResNet-101 backbone. In case (a), the 'lizard' is moving quickly, causing severe motion blur. Our method fails to detect the lizard in third frame. Case (b) shows the partial occlusion of a 'cattle' as the camera moves. Our method misclassify hind legs of a 'cattle' to 'antelope' in second frame, and recovers at following frames.

results. Examples of such failure cases are shown in Fig. 5. It can be overcome by utilizing additional methods such as motion-based feature propagation [37], [51] or tublet rescoring [15], [52].

VI. CONCLUSION

Still image object detectors on video suffer from various image deteriorations. We propose DiffusionVID, the first diffusion model-based video object detector. DiffusionVID leverages spatio-temporal conditioning to detect object boxes in images. DiffusionVID comprises three key components: cascade refinement, dynamic coreset conditioning, and local batch refinement. Experimental results show that the proposed DiffusionVID achieves 92.5 mAP on the ImageNet-VID benchmark dataset, demonstrating state-of-the-art performance. Our method still achieves SOTA with 86.9 mAP at 46.6 FPS, even with settings focusing on the accuracy-speed trade-off.

REFERENCES

[1] H. Deng, Y. Hua, T. Song, Z. Zhang, Z. Xue, R. Ma, N. Robertson, and H. Guan, "Object guided external memory network for video object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6678–6687.

[2] G. Sun, Y. Hua, G. Hu, and N. Robertson, "MAMBA: Multi-level aggregation via memory bank for video object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2620–2627.

[3] Q. Zhou, X. Li, L. He, Y. Yang, G. Cheng, Y. Tong, L. Ma, and D. Tao, "TransVOD: End-to-end video object detection with spatial–temporal transformers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7853–7869, Jun. 2023.

[4] H. Wang, J. Tang, X. Liu, S. Guan, R. Xie, and L. Song, "PTSEformer: Progressive temporal-spatial enhanced transformer towards video object detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 732–747.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–22.

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[9] J. Shi, Y. Wang, Z. Yu, G. Li, X. Hong, F. Wang, and Y. Gong, "Exploiting multi-scale parallel self-attention and local variation via dual-branch transformer-CNN structure for face super-resolution," *IEEE Trans. Multimedia*, early access, Aug. 3, 2023, doi: 10.1109/TMM.2023.3301225.

[10] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image restoration using Swin transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1833–1844.

[11] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7354–7363.

[12] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.

[13] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–22.

[14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision—ECCV 2020*. Glasgow, U.K.: Springer, Aug. 2020, pp. 213–229.

[15] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, "Relation distillation networks for video object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7023–7032.
[16] H. Wu, Y. Chen, N. Wang, and Z.-X. Zhang, "Sequence level semantics aggregation for video object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9217–9225.
[17] Y. Chen, Y. Cao, H. Hu, and L. Wang, "Memory enhanced global-local aggregation for video object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10337–10346.
[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.
[19] M. Fujitake and A. Sugimoto, "Video sparse transformer with attention-guided memory for video object detection," *IEEE Access*, vol. 10, pp. 65886–65900, 2022.
[20] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2256–2265.
[21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10684–10695.
[22] O. Avrahami, D. Lischinski, and O. Fried, "Blended diffusion for text-driven editing of natural images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18208–18218.
[23] W.-C. Fan, Y.-C. Chen, D. Chen, Y. Cheng, L. Yuan, and Y.-C. F. Wang, "Frido: Feature pyramid diffusion for complex scene image synthesis," 2022, *arXiv:2208.13753*.
[24] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," 2022, *arXiv:2204.03458*.
[25] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. Wook Kim, S. Fidler, and K. Kreis, "Align your latents: High-resolution video synthesis with latent diffusion models," 2023, *arXiv:2304.08818*.
[26] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "WaveGrad: Estimating gradients for waveform generation," 2020, *arXiv:2009.00713*.
[27] G. Mittal, J. Engel, C. Hawthorne, and I. Simon, "Symbolic music generation with diffusion models," 2021, *arXiv:2103.16091*.
[28] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4713–4726, Apr. 2023.
[29] S. Chen, P. Sun, Y. Song, and P. Luo, "DiffusionDet: Diffusion model for object detection," 2022, *arXiv:2211.09788*.
[30] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, "ILVR: Conditioning method for denoising diffusion probabilistic models," 2021, *arXiv:2108.02938*.
[31] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 8780–8794.
[32] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. G. Lopes, B. K. Ayan, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, "Photorealistic text-to-image diffusion models with deep language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 36479–36494.
[33] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 408–417.
[34] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2349–2358.
[35] X. Zhu, J. Dai, L. Yuan, and Y. Wei, "Towards high performance video object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7210–7218.
[36] S. Wang, Y. Zhou, J. Yan, and Z. Deng, "Fully motion-aware network for video object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 542–557.
[37] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3038–3046.
[38] G. Bertasius, L. Torresani, and J. Shi, "Object detection in video with spatiotemporal sampling networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 331–346.
[39] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[40] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, and P. Luo, "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14454–14463.
[41] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," 2017, *arXiv:1708.00489*.
[42] S.-D. Roh and K.-S. Chung, "DAFA: Diversity-aware feature aggregation for attention-based video object detection," *IEEE Access*, vol. 10, pp. 93453–93463, 2022.
[43] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.
[44] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
[45] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid, "Unsupervised object discovery and tracking in video collections," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3173–3181.
[46] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 817–825.
[47] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, and W. Ouyang, "T-CNN: Tubelets with convolutional neural networks for object detection from videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2896–2907, Oct. 2018.
[48] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3282–3289.
[49] T. Deselaers, B. Alexe, and V. Ferrari, "Localizing objects while learning their appearance," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 452–466.
[50] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10526–10535.
[51] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.
[52] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang, "Seq-NMS for video object detection," 2016, *arXiv:1602.08465*.

SI-DONG ROH (Graduate Student Member, IEEE) received the B.S. degree in electronic engineering from Hanyang University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree in electronic engineering. His research interests include video-object detection and image-based defect inspection.

KI-SEOK CHUNG (Member, IEEE) received the B.S. degree in computer engineering from Seoul National University, Seoul, South Korea, in 1989, and the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, in 1998. He was a Senior Research and Development Engineer with Synopsys Inc., Mountain View, CA, USA, from 1998 to 2000; and a Staff Engineer with Intel Corporation, Santa Clara, CA, from 2000 to 2001. He was also an Assistant Professor with Hongik University, Seoul, from 2001 to 2004. Since 2004, he has been a Professor with Hanyang University, Seoul. His research interests include low-power embedded system design, multi-core architecture, image processing, reconfigurable processors and DSP design, SoC-platform-based verification, and system software for MPSoCs.