electronics

MDPI

Article

Even-Order Taylor Approximation-Based Feature Refinement and Dynamic Aggregation Model for Video Object Detection

Liule Chen, Jianqiang Li, Yunyu Li and Qing Zhao *

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; gary27@emails.bjut.edu.cn (L.C.); lijianqiang@bjut.edu.cn (J.L.); liyunyu@emails.bjut.edu.cn (Y.L.)
* Correspondence: zhaoqing@bjut.edu.cn; Tel.: +86-186-1132-9377

Abstract: Video object detection (VOD) is a sophisticated visual task. It is a consensus that is used to find effective supportive information from correlation frames to boost the performance of the model in VOD tasks. In this paper, we not only improve the method of finding supportive information from correlation frames but also strengthen the quality of the features extracted from the correlation frames to further strengthen the fusion of correlation frames so that the model can achieve better performance. The feature refinement module FRM in our model refines the features through the key–value encoding dictionary based on the even-order Taylor series, and the refined features are used to guide the fusion of features at different stages. In the stage of correlation frame fusion, the generative MLP is applied in the feature aggregation module DFAM to fuse the refined features extracted from the correlation frames. Experiments adequately demonstrate the effectiveness of our proposed approach. Our YOLOX-based model can achieve 83.3% AP50 on the ImageNet VID dataset.

Keywords: video object detection; feature refinement; feature aggregation; Taylor series

Citation: Chen, L.; Li, J.; Li, Y.; Zhao, Q. Even-Order Taylor Approximation-Based Feature Refinement and Dynamic Aggregation Model for Video Object Detection. *Electronics* **2023**, *12*, 4305. https://doi.org/10.3390/electronics12204305

Academic Editor: Stefanos Kollias

Received: 1 September 2023
Revised: 10 October 2023
Accepted: 15 October 2023
Published: 18 October 2023

1. Introduction

Object detection is a fundamental and challenging task in many computer vision applications, which aims to localize and classify objects in images. Remarkable progress in object detection has been witnessed in the past few years. Object detectors based on convolutional neural networks (CNN) [1] are constantly evolving. Existing object detection models based on convolutional neural networks can be divided into two categories: one-stage and two-stage detectors. The YOLO series [2–5] and SSD [6] are representative works of one-stage detectors based on convolutional neural networks. One-stage detectors are powerful due to excellent accuracy and speed trade-off, so they are suitable for scenarios where real-time performance is required. Two-stage detectors focus more on accuracy and are therefore more complex in terms of processing than one-stage detectors. Two-stage detectors first select possible regions where objects may be present and then classify the targets in the regions. The difference between the two is whether or not there is an explicit process of extracting RoI features. In the absence of an explicit RoI feature-extraction step, one-stage detectors produce the location and classification directly from the feature map. This approach has led to the diversity of features but also to a low quality of features. A large number of low-quality features are discarded by post-processing to achieve the purpose of screening features. But the quality of the features that were preserved did not improve, while high-quality features are key to dense prediction tasks [7,8]. Thus, one aim of this work is to boost the performance of the model by improving the quality of features. In the pursuit of superior feature quality, series expansion instead of the softmax function is employed to mitigate computational overhead. The even-order Taylor series is utilized, specifically aiming to enhance the inter-cluster distance while simultaneously reducing the intra-cluster distance. The eventual acquisition of more refined classification boundaries is attainable.

Video Object Detection (VOD) is a high-level version of still image object detection. The two have the same task, namely location and classification, but the difference is that VOD can make use of the temporal information of videos. When a target object appears in several different video frames, the relationship between the frames can be used to refine the detection. Therefore, incorporating temporal information into the detectors is the key to enhancing confidence and eliminating ambiguity in the case of image degradation. Figure 1 shows the motion blur and the rare pose of objects that appear in the videos.

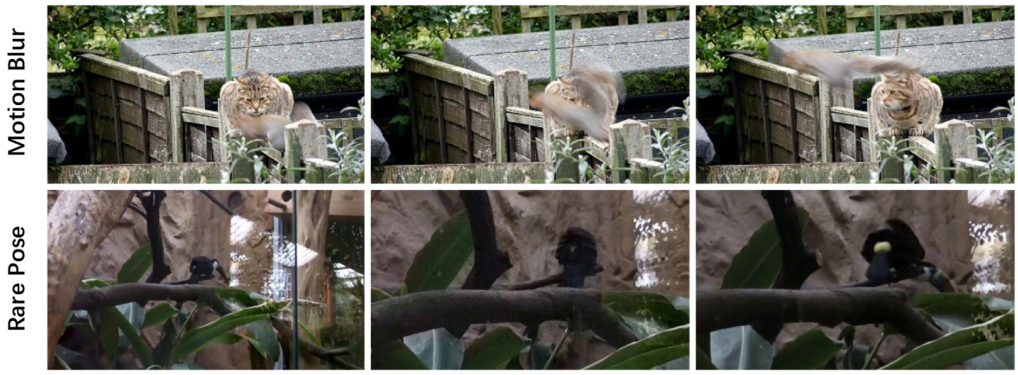

Figure 1. Motion blur occurs when the object is moving at a high speed, and the object may appear in a posture that is difficult to recognize.

Existing VOD methods fuse the temporal information from two perspectives: one correlates the detection results of the underlying detectors through post-processing to produce coherent results, and the other one directly improves the detection accuracy of the detectors by aggregating temporal information at the feature level. Post-processing methods usually apply still image detectors, especially two-stage detectors, to obtain detection results. After obtaining the detection results, the bounding boxes are linked to form a tubelet, and then the results in the same tubelet are refined. Methods that aggregate temporal information directly at the feature level typically utilize features in reference frames to refine the features of the current frame. It is worth mentioning that due to the characteristics of the region proposal networks (RPN) [9], proposals are given explicit representation, so two-stage detectors can more easily migrate to the task of video object detection. However, most of the video object detection methods based on two-stage detectors are overly sophisticated and inherit the defects of still image detectors, such as the low quality of features aforementioned, which do not meet the needs of real-time applications and can have higher accuracy on an image sequence. Thus, a real-time one-stage video object detection method with improved features refining and aggregating is built.

Contribution. In this work, we propose an FRM block based on multi-layer perceptrons (MLPs) and a key–value encoding pattern that is based on even-order Taylor series approximation to refine features. This block guides and improves the quality of features at each scale by extracting the deepest features rich in semantics with the following aggregating global and local features. The channel MLP is adopted to capture the long-range dependencies when the learnable key–value encoder is adopted to enhance the detail of local features. In the experiments, this block is validated as having the ability to improve the feature quality. Then, we propose a DFAM block based on MLP to aggregate features from reference frames to the current frame. This block can generate aggregating matrices for different sets of features selected from video frames. The proposed approach can be applied in the field of autonomous driving, such as enhancing the existing perception algorithm [10,11].

Performance. The proposed method can achieve a promising accuracy of 83.3% AP50 on the ImageNet VID dataset without post-processing. Based on YOLOX-M, FRM gains 2.4% in AP50, and DFAM gains 4.6% in AP50, which fully proves the effectiveness of the method.

2. Related Work

2.1. MLP-Based Network

Recently, MLP-Mixer [12], a completely MLP-based architecture, proposed the use of linear layers for simple token-mixing and channel-mixing to replace the self-attention mechanism [13] and achieve competitive results with faster inference speed in image-classification tasks. Specifically, token-mixing encodes spatial information by interacting between all tokens, while channel-mixing features interact between all channels within each token. The success of MLP-Mixer brought multilayer perceptron architecture back to the fore. At the same time, gMLP [14] uses multiplication gating-based MLP to prove that MLP-based methods can also be a good alternative to self-attention mechanisms in natural language processing tasks. Subsequently, research on the application of MLP to computer vision tasks continued to emerge, and ViP [15] proposed independent modeling along the horizontal and vertical directions to solve the problem of spatial information loss. AS-MLP [16] introduces local token-mixing based on global token-mixing and uses horizontal and vertical space movement to solve the problem of lack of local information exchange. This enables us to pay more attention to the characteristics of the local area. S2-MLP [17] further adopts spatial displacement in four directions to collect local area information more fully. CycleMLP [18] adopts a pseudo-kernel structure to solve the problem wherein MLP architecture is sensitive to the size of the input image. MAXIM [19] uses cross-gating blocks and multi-axis gating blocks to mix local and global information, further improving the performance and computational efficiency of MLP architectures.

2.2. Still-Image Object Detection

In the task of object detection, some representative work in the past, such as the two-stage detector RCNN [20], applied CNNs to object detection for the first time. Fast-RCNN [21] introduced RoI projection to reduce the amount of computation. Faster-RCNN [9] then proposed RPN networks for generating candidate boxes. Cascade-RCNN [22] proposed a multi-stage architecture to address the mismatch problem with different IoU thresholds and the overfitting problem with increasing IoU thresholds. Sparse-RCNN [23] introduced learnable proposals for end-to-end detection. In the one-stage detectors, the YOLO series [2–5], as well as SSD [6], RetinaNet [24], FCOS [25], etc., no longer explicitly extract RoIs, but directly predict from the feature map. Compared with the two-stage detectors, it has a greater advantage in detection speed and competitive detection accuracy. At the same time, with the widespread application of the attention mechanism in visual tasks, some work based on the attention mechanism has also been proposed, such as DETR [26], introducing query to abstract the object detection into a set prediction. Deformable DETR [27], Sparse DETR [28], Conditional DETR [29], DAB-DETR [30], and DINO [31] mainly focus on solving the problems of slow convergence and high computational complexity.

2.3. Video Object Detection

Video object detection tasks have richer information than still-image object detection tasks, but due to the movement of the object and the change of scene, it inevitably introduces the problem of image degradation, such as occlusion, camera defocus, and rare poses. Moreover, video object detection tasks also require spatial and temporal consistency while detecting objects. Most of the existing video object detection methods are based on two-stage detectors, which pay more attention to improving accuracy, and some work has made efforts to improve the inference speed under the premise of ensuring accuracy as much as possible; we divide the existing methods into two directions: focusing on improving accuracy by aggregating temporal information and improving inference speed without significantly losing accuracy.

With regard to the first aspect, D&T [32] proposed introducing the object-tracking task into video object detection, learning the similarity between features in different frames by tracking, and obtaining the displacement of targets between frames to assist detection. MEGA [33] integrates the global and local information to enhance the candidate box

features in the keyframe and store the enhanced features in an external memory bank. FGFA [34] proposed the concept of temporal features aggregation for the first time in the field of video object detection, which produces detection results for non-key frames by fusing and weighting the optical flow features between the current frame and adjacent frames. Similarly, MA-Net [35] first extracts the features of the current frames and adjacent frames and the optical flow information between them, corrects and fuses them at the pixel level and instance levels, and uses the fused features for training. Other optical flow-based algorithms, SELSA [36] and OGEMN [37], further improve the detection accuracy. RDN [38] captures the interactions across the objects in both the spatial and temporal context. HVRNet [39] integrates intra-video and inter-video proposal relations, respectively, to exploit intra and inter-contexts. STSN [40] proposes learning spatial sample features from neighboring frames of the current frame and aggregates the learned features into the features of the current frame for detection. STMM [41] models long-term temporal appearance and motion dynamics to boost detection accuracy. TransVOD [42] introduces the transformer structure to the video object detection task and proposes a temporal transformer to aggregate both the spatial object queries and the feature memories of each frame, which effectively eliminates the need for many hand-crafted components and complicated post-processing steps.

In terms of the second aspect, DFF [43] first proposed the concept of keyframes, which improves the speed of inference by distinguishing between key keyframes and non-keyframes, using keyframes directly for detection while extracting optical flow information from non-keyframes and fusing them into key features. THP [44] achieves the purpose of accelerating inference by extracting features of sparsely distributed keyframes and propagating the features to other non-keyframes in the form of flow fields, and the method has been tested on mobile devices for the first time. LSTM-SSD [45] adopts bottleneck-LSTM to refine and propagate features across frames, which significantly reduces computational cost. ST-Lattice [46] performs detection on sparse keyframes and uses temporal information to assist in the detection of non-keyframes at different spatial scales. CHP [47] propagates previous reliable detection in the form of a heatmap to boost the results of current images. YOLOV [48] integrates the post-processing step into the detection head, selects the features with high confidence in different video frames, and aggregates the features through the multi-head self-attention mechanism to achieve the purpose of refining the features in the current frame. These approaches have made efforts and attempts from various perspectives, but there can be more effective improvements in improving the quality of features and the aggregation of features. Our model first starts from the feature itself, strengthens the feature by key–value coding based on even-order Taylor series, and then fuses the features between frames on this basis. The fusion method uses MLP, which fuses the features in a generative way.

3. Methodology

Following the insight that collecting supportive information from reference frames to enhance the detection of the keyframes is critical to improving the video object detection methods' performance, our work boosts accuracy by improving the quality of features and aggregating the features from multiple frames of a video clip with a confidence level above a preset threshold. Previous work [33,36,38] has focused on how to effectively find and save supportive information from reference frames and use it for the detection of the current frame. While this is important, the quality of features from the reference frame is not paid attention to, whereas high-quality features can further improve the accuracy of detection. Considering that the deepest features have the richest semantic information [49], we use the deepest features to adjust the shallow features. Through specific design, we can make the deepest features retain as much local information as possible, which is quite critical for dense prediction tasks. After collecting a certain number of high-quality features from multiple reference frames, how to effectively aggregate these features is what we will focus on next. Due to the high computational cost of Transformers and the limited

receptive fields of convolutional neural networks, MLP is adopted to aggregate features from reference frames. Different from the traditional MLP linear mapping operation in the spatial dimension or channel dimension, we construct the weights in a generative way—that is, dynamically generate feature aggregation weights in the space-time dimension and channel dimension through linear mapping. The detailed descriptions of our proposed work are given below. Based on YOLOX [5], our work integrates two key components: Feature Refine Module (FRM) and Dynamic Feature Aggregation Module (DFAM). Our proposed method is schematically depicted in Figure 2.

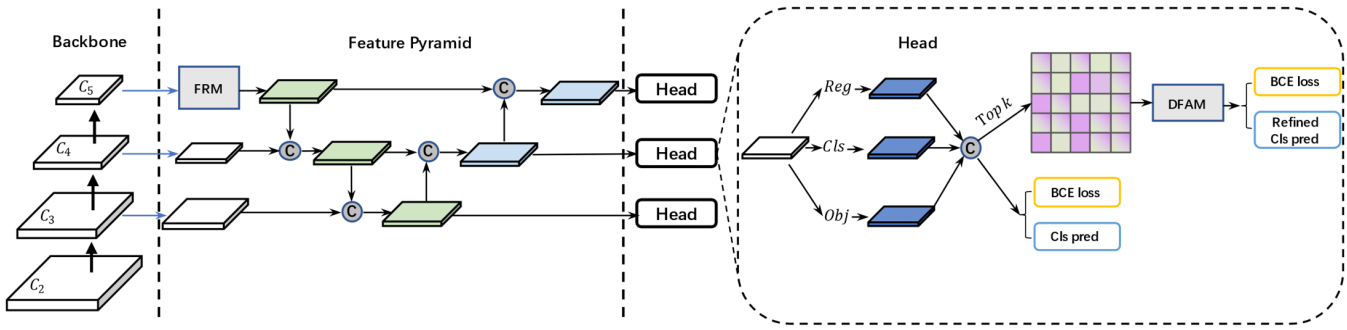

Figure 2. Illustration of the whole architecture of our proposed network. FRM and DFAM are integrated into the model's FPN and detection head, respectively. FRM refines the deepest feature and guides the shallow feature. The detection head selects features by top-k, inputs them into the DFAM module, and calculates the loss after feature aggregation.

FRM: Feature Refining Module. As illustrated in Figure 3, FRM achieves the purpose of improving feature quality through feature interaction between layers, which is realized by an endogenous learnable key–value parameter. This learnable key–value parameter adjusts the multi-scale features of FPN [50] through a special weighted-average method so that the originally ignored local information is strengthened by intra-layer interaction. This module consists of CBR blocks (a 3×3 convolution with a BN layer and ReLU activation function), learnable key–value parameters (a matrix that participates in derivation), and channel MLPs. In forward propagation, this module first uses a 3×3 convolution operation to downsample the input feature map to reduce the cost of the calculation, and the result of downsampling is used as the pyramid input of the two branches. The first branch encodes features using a CBR block with the encoded feature size $N \times C \times W \times H$, where N denotes the batch size, C denotes the channel size, and H and W denote the feature map spatial size in height and width. The size of the learnable key–value parameters is $K \times C$, where K denotes the number of key–value pairs, and C denotes the number of channels. The learnable key–value parameters are defined as $P_{key-value} = \{v_1, v_2, \ldots, v_K\}$, and the encoded features are defined as $X_{in} = \{x_1, x_2, \ldots, x_L\}$, where $L = H \times W$ is the total spatial number of the input features. The query values of x_i in the learnable key–value parameters in the feature map are as follows:

$$Rf\left(x_i, P_{key-value}\right) = \sum_{k=1}^{K} \frac{f^n\left(\|x_i - v_k\|^2\right)}{\sum_{j=1}^{K} f^n\left(\|x_i - v_j\|^2\right)}(x_i - v_k) \tag{1}$$

Therefore, the query value of the encoded feature map in the learnable key–value parameter is as follows:

$$Rf\left(X_{in}, P_{key-value}\right) = \sum_{i=1}^{L}\sum_{k=1}^{K} \frac{f^n\left(\|x_i - v_k\|^2\right)}{\sum_{j=1}^{K} f^n\left(\|x_i - v_j\|^2\right)}(x_i - v_k) \tag{2}$$

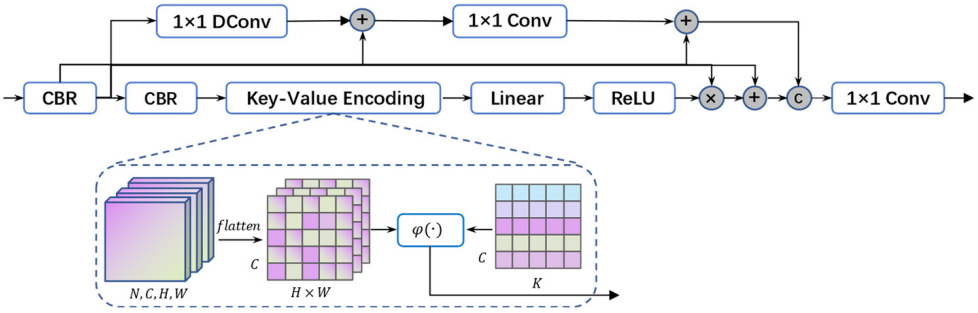

Figure 3. Illustration of DFM. $\varphi(\cdot)$ denotes Equation (3).

It is important to note that f^n is a higher-order Taylor series approximation of e^x, where the orders n are even. The

$$f^n(z) = \sum_{i=0}^{n} \frac{z^i}{i!} \tag{3}$$

The features encoded by the learnable key–value parameters are then entered into an activation function and a linear layer, and the output of the first branch rX_{in1} is obtained by multiplying and then adding the features (the feature before being encoded by key–value parameters and the feature entered into the first branch) in the form of residual linkage. σ in Equation (4) denotes the ReLU activation function.

$$rX_{in1} = \sigma\Big(Linear\Big(Rf\Big(X_{in}, P_{key-value}\Big)\Big)\Big) \odot X_{in} + X_{in} \tag{4}$$

The second branch contains 1×1 depth-wise convolution and 1×1 convolution, and the operation of this branch is defined by the following formulas:

$$X_{in2} = DConv(GN(X_{in})) + X_{in} \tag{5}$$

$$rX_{in2} = \sigma(Conv_{1\times1}(X_{in2})) + X_{in} \tag{6}$$

GN in Equation (5) denotes group normalization, where the group number is 1. σ in Equation (6) denotes the GeLU activation function. The features of the two branch outputs are concatenated in the channel dimension, and then the features are entered into a 1×1 convolution layer to restore the number of channels.

$$rX_{final} = Conv_{1\times1}(Concat(rX_{in1}, rX_{in2})) \tag{7}$$

DFAM: Dynamic Feature Aggregation Module. As illustrated in Figure 4, DFAM completes the aggregation of the features of the reference frame and the current frame by dynamically generating and redistributing the weights. This module contains three branches corresponding to the temporal, spatial, and channel dimensions, each implemented by multiple linear layers for feature encoding, weight generation, and weight redistribution. The dimension of the feature that entered into three branches is $F \times C \times T$, where F denotes the spatial dimension size, C denotes the channel size, and T denotes the temporal dimension size. Corresponding to spatial and temporal dimensions, we first reshape the input features into shape $F \times TC$ (TC equals $T \times C$), and the obtained features X are entered into the linear layers to generate weights corresponding to this dimension (W_f corresponds to the spatial dimension and W_t corresponds to the temporal dimension). It should be noted that $\otimes$ in the following equations stands for the matrix product, and $\odot$ stands for Hadamard product. The weight and the feature entered into this branch are multiplied as the output of this branch, which can be denoted as

$$W_t = Softmax(Linear_t(X)) \tag{8}$$

$$W_f = Softmax(Linear_f(X)) \tag{9}$$

$$X_t = W_t \otimes X \tag{10}$$

$$X_f = W_f \otimes X \tag{11}$$

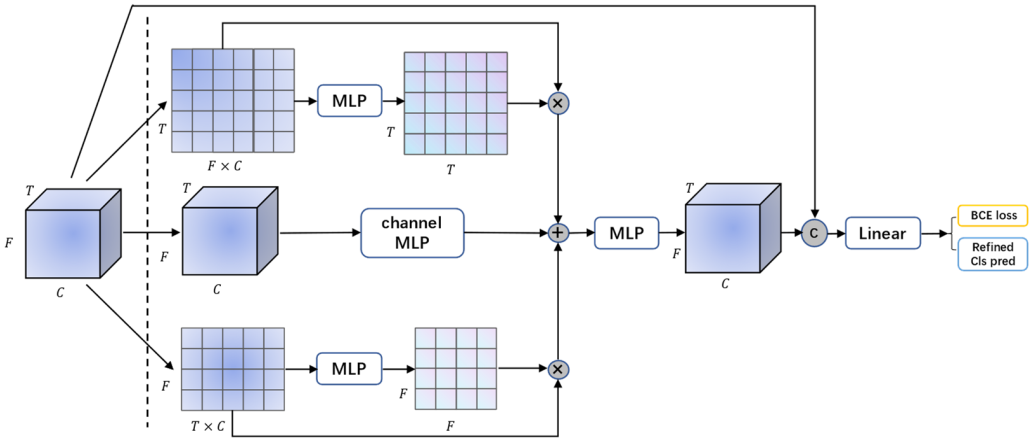

Figure 4. Illustration of DFAM. Input feature with shape $F \times T \times C$ is selected by top-k according to the confidence.

Corresponding to the channel dimension, the output of this branch is generated directly by mapping in the channel dimension using linear layers, defined by the following formula:

$$X_c = Linear(X) \tag{12}$$

We add the outputs of the three branches and calculate the mean for each channel. The result is fed into a linear layer that redistributes the weights. After the redistribution, the number of channels of the feature is tripled, corresponding to three branches, and the reassignment of weights is defined by the following formula:

$$\left[reW_t, reW_f, reW_c\right] = Linear\left(Mean\left(X_t + X_f + X_c\right)\right) \tag{13}$$

$$X_{tfc} = Linear\left(reW_t \odot X_t + reW_f \odot X_f + reW_c \odot X_c\right) \tag{14}$$

The weights and outputs corresponding to the three branches are multiplied, the obtained output features and the features entered into the DFAM are concatenated along the channel dimension as the input of two linear layers, and then these two layers output the aggregated features $aggX$ of DFAM.

$$aggX = Linear\left(Concat\left(X_{tfc},\ X\right)\right) \tag{15}$$

Loss Function. The loss function is Equation (16). We use BCE Loss for training the classification branch (L_{cls}), refined classification branch (L_{rcls}), and object branch (L_{obj}), and IoU Loss for training the regression branch (L_{iou}). λ_{rcls} and λ_{iou} are the corresponding weight coefficients.

$$Loss = L_{cls} + \lambda_{rcls} \cdot L_{rcls} + \lambda_{iou} \cdot L_{iou} + L_{obj} \tag{16}$$

4. Experiment

4.1. Experimental Setup

Datasets and Evaluation. Following [34,36,44,46], experiments were performed on videos in the ImageNet VID [51] and images in the ImageNet DET [51] with the same

classes. The ImageNet VID contains 4416 videos, among which there are 3862 videos for training and 555 videos for validation. The ImageNet VID dataset has 30 categories, which is a subset of the 200 categories of ImageNet DET. Following protocols widely adopted in [32,34,35,42,46], the method proposed is evaluated on the validation set using the mean average precision (mAP) metric.

Network architectures. YOLOX is adopted as our base detector. Following [46,48], the base detector is pretrained on the COCO dataset [52]. It should be clarified that the feature extractor of the base detector is Modified CSPDarknet [53], and the three deepest features are adopted for the detection task. The Feature Refining Module is integrated into the FPN [50], and the Dynamic Feature Aggregation Module is integrated into the YOLOX Head.

Training details. The training procedure is divided into three stages. In the first training stage, the pre-trained base detector YOLOX with backbone MCSP [53] is finetuned on the combination of the ImageNet VID dataset and the ImageNet DET dataset with the same categories. To eliminate redundancy of the video frames, 1/10 frames are sampled in the VID training set. Most training configuration in YOLOX is carried over, except for epochs, which is set to 15, and the learning rate per batch is set to 0.002. While the training epochs number 15, the warmup epochs are decreased to 2, and the number of no-augmentation epochs is 3. In the second training stage, the detection head is integrated with FRM. Then, the detector is trained for 7 epochs by an SGD optimizer with a batch size of 16. A total of 7 training epochs are divided in an orderly fashion into 3 groups: a first epoch for warming up with a learning rate increasing from 0 to 0.01; second to fifth epochs for training with strong augmentation, and a learning rate decreasing from 0.01 to 0.002; and sixth to seventh epochs for training, with strong augmentation disabled and the learning rate remaining unchanged. In the third training stage, DFAM is integrated into the backbone MCSP. Most of the former training configuration is adopted, while the backbone parameters are frozen. Only the linear projection and the DFAM in the detection head are fine-tuned. The proposed approach is implemented using the Pytorch [54] framework. All experiments are performed on 2GPUs RTX A5000 with FP16-precision enabled. In the training phase, the size of input images ranges from 352×352 to 672×672 with a stride of 32×32. In the testing phase, the size of input images is set to 512×512. The inference performance of the model is tested on an RTX GeForce 3090 with FP16-precision enabled.

4.2. Main Results

Comparison with base detector. Our proposed method is first compared with the base detector YOLOX in Table 1. Our proposed methods outperform their respective base detectors by over 5% in AP50. As for the inference time, our proposed method has a small increase, which is within a tolerated range.

Comparison with existing VOD methods. Table 2 shows the detailed comparison information between our proposed method and existing VOD methods. It can be seen from the results that our method achieves 83.3% in AP50. Our proposed method exceeds most two-stage detector-based VOD methods, such as SELSA (+3.0%) and ST-Lattice (+4.3%). Our proposed method also exhibits better performance than the one-stage detector-based VOD method in accuracy. Our proposed method shows better performance when compared with optical flow estimation methods, such as FGFA (+7.0%) and OGEMN (+3.3%). Thanks to feature-refining, our proposed method overcomes the influence of low feature quality on dense prediction tasks to a certain extent. Dynamic feature aggregation also effectively aggregates the temporal features and improves the detection of the current frame in a targeted manner. Figures 5–8 show a comparison between the proposed method and the base detector.

Table 1. Comparison with the base detector YOLOX.

Model	Params	GFLOPs	Time (ms)	AP50 (%)
YOLOX-S	8.95 M	17.58	4.3	68.2
Ours-S	14.40 M	29.53	8.2	75.5
YOLOX-M	25.30 M	48.38	5.4	71.2
Ours-M	37.46 M	75.26	11.7	77.2
YOLOX-L	54.17 M	102.10	9.5	74.8
Ours-L	75.73 M	149.88	15.3	80.9
YOLOX-X	99.02 M	184.93	15.2	77.0
Ours-X	132.66 M	259.57	32.3	83.3

Table 2. Comparison with existing video object detection methods.

Base Detector	Methods	mAP (%)
R-FCN [8]	DFF [43]	73.0
	D&T [32]	75.8
	FGFA [34]	76.3
	THP [42]	78.6
	STSN [38]	78.9
	OGEMN [35]	80.0
	STMM [41]	80.5
Faster-RCNN [9]	ST-Lattice [44]	79.0
	SELSA [34]	80.3
	RDN [36]	81.8
	MEGA [31]	82.9
	HVRNet [37]	83.2
CenterNet [55]	CHP [45]	76.7
Deformable DETR [25]	TransVOD [40]	81.9
YOLOX [5]	Ours	83.3

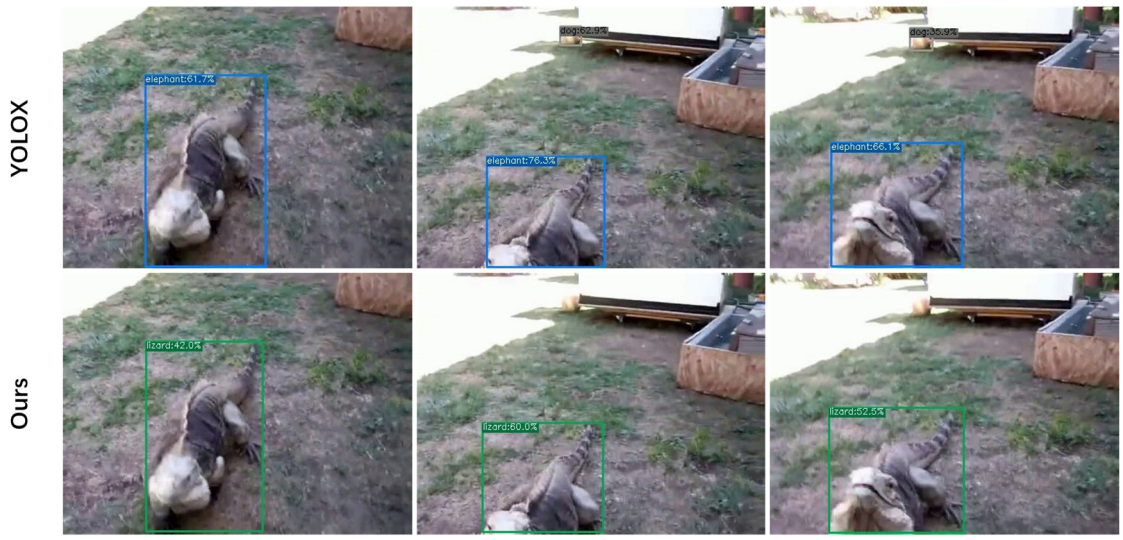

Figure 5. Visualization of the comparison between baseline and proposed method with regard to motion blur.

Figure 6. Visualization of the comparison between baseline and proposed method with regard to defocus.

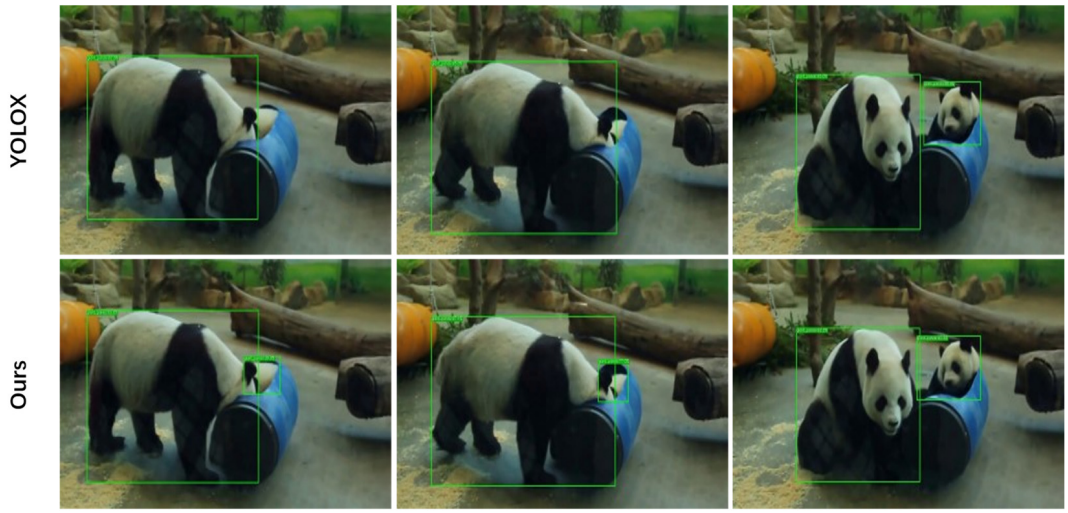

Figure 7. Visualization of the comparison between baseline and proposed method with regard to occlusion.

Figure 8. Visualization of the comparison between baseline and proposed method with regard to rare pose.

4.3. Ablation Study and Analysis

On the effectiveness of FRM and DFAM. To validate the effectiveness of FRM and DFAM, the experiments were performed with YOLOX-M. Table 3 summarizes the effects of FRM and DFAM on the ImageNet VID dataset. As can be observed, YOLOX-M armed with FRM gains 2.4% in AP50. YOLOX-M armed with both FRM and DFAM gains 6.0% in AP50. This ablation study shows the effects of FRM and DFAM.

Table 3. Ablation study with or without FRM and DFAM.

Methods	FRM	DFAM	AP50 (%)
YOLOX-M	-	-	71.2
Ours-M	✓	-	73.6
Ours-M	✓	✓	77.2

On the order of the Taylor series. Table 4 illustrates the effectiveness of the order of the Taylor series. The ablation experiments are performed on YOLOX-S integrated with FRM. While the order is less than 6, AP50 is higher than 70.0%. As can be observed, when the order is 4, our proposed method achieves the best AP50, at 71.0%. But the AP50 drops to 69.7% when the order is 8. It can be seen that the Taylor series of different orders has a certain influence on the performance of the method. It is better to dynamically adjust the order of the Taylor series according to the needs of the actual task. And how the order affects the model's performance on specific tasks is left for our future work.

Table 4. Ablation study on the order of Taylor series.

Order	2	4	6	8
AP50 (%)	70.7	71.0	70.5	69.7

On the contribution of each path in DFAM. Table 5 illustrates the contribution of each path in DFAM. The ablation experiments are performed on YOLOX-S integrated with FRM and DFAM. The specific branch in DFAM is disabled in each experiment. While the temporal branch is disabled, AP50 is 71.1%. It can be observed that disabling the temporal branch decreases the AP50 most. This is followed by the channel branch, which has an AP50 of 72.5% when the channel branch is disabled. The spatial branch contributes the least to DFAM, and the AP50 is 73.3% when the spatial branch is disabled.

Table 5. Ablation study on the contribution of each path in DFAM.

Path	Temporal	Channel	Spatial	Original
AP50 (%)	71.1	72.5	73.3	75.5

5. Conclusions

In this paper, a new video object detection framework is proposed, starting from feature quality, which is critical to dense prediction tasks. This framework refines features through learnable key–value parameters to improve the quality of features for the DFAM, which aggregates temporal features by dynamically generating weights. Experiments and ablation studies have shown the effectiveness and practicality of both modules. This paper improves video object detection from two novel perspectives, hoping to provide some enlightenment for related work.

Author Contributions: Conceptualization, J.L., Q.Z., L.C. and Y.L.; methodology, L.C. and Y.L.; software, L.C. and Y.L.; writing, Q.Z., L.C. and Y.L.; visualization Y.L.; supervision, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This study was partially supported by the National Key R&D Program of China via project No. 2020YFB2104402.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
2. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26–30 June 2016.
3. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
4. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
5. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
6. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016.
7. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
8. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems 29 (NIPS 2016), Barcelona, Spain, 5–10 December 2016.
9. Ren, S.; He, K.; Girshick, R. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015.
10. Xia, X.; Bhatt, N.P.; Khajepour, A.; Hashemi, E. Integrated inertial-LiDAR-based map matching localization for varying environments. *IEEE Trans. Intell. Veh.* **2023**. [CrossRef]
11. Meng, Z.; Xia, X.; Xu, R.; Liu, W.; Ma, J. HYDRO-3D: Hybrid Object Detection and Tracking for Cooperative Perception Using 3D LiDAR. *IEEE Trans. Intell. Veh.* **2023**, *8*, 4069–4080. [CrossRef]
12. Tolstikhin, I.O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Dosovitskiy, A. Mlp-mixer: An all-mlp architecture for vision. In Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS 2021), Online, 6–14 December 2021; pp. 24261–24272.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
14. Liu, H.; Dai, Z.; So, D.; Le, Q.V. Pay attention to mlps. In Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS 2021), Online, 6–14 December 2021; pp. 9204–9215.
15. Hou, Q.; Jiang, Z.; Yuan, L.; Cheng, M.M.; Yan, S.; Feng, J. Vision permutator: A permutable mlp-like architecture for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 1328–1334. [CrossRef] [PubMed]
16. Lian, D.; Yu, Z.; Sun, X.; Gao, S. As-mlp: An axial shifted mlp architecture for vision. *arXiv* **2021**, arXiv:2107.08391.
17. Yu, T.; Li, X.; Cai, Y.; Sun, M.; Li, P. S2-mlp: Spatial-shift mlp architecture for vision. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 4–8 January 2022.
18. Chen, S.; Xie, E.; Ge, C.; Chen, R.; Liang, D.; Luo, P. Cyclemlp: A mlp-like architecture for dense prediction. *arXiv* **2021**, arXiv:2107.10224. [CrossRef] [PubMed]
19. Tu, Z.; Talebi, H.; Zhang, H.; Yang, F.; Milanfar, P.; Bovik, A.; Li, Y. Maxim: Multi-axis mlp for image processing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Waikoloa, HI, USA, 4–8 January 2022.
20. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
21. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
22. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
23. Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Luo, P. Sparse r-cnn: End-to-end object detection with learnable proposals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021.
24. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
25. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
26. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020.
27. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.

28. Roh, B.; Shin, J.; Shin, W.; Kim, S. Sparse detr: Efficient end-to-end object detection with learnable sparsity. *arXiv* **2021**, arXiv:2111.14330.
29. Meng, D.; Chen, X.; Fan, Z.; Zeng, G.; Li, H.; Yuan, Y.; Wang, J. Conditional detr for fast training convergence. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
30. Liu, S.; Li, F.; Zhang, H.; Yang, X.; Qi, X.; Su, H.; Zhang, L. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv* **2022**, arXiv:2201.12329.
31. Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging properties in self-supervised vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
32. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Detect to track and track to detect. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
33. Chen, Y.; Cao, Y.; Hu, H.; Wang, L. Memory enhanced global-local aggregation for video object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
34. Zhu, X.; Wang, Y.; Dai, J.; Yuan, L.; Wei, Y. Flow-guided feature aggregation for video object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
35. Miao, J.; Wei, Y.; Yang, Y. Memory aggregation networks for efficient interactive video object segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
36. Wu, H.; Chen, Y.; Wang, N.; Zhang, Z. Sequence level semantics aggregation for video object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
37. Deng, H.; Hua, Y.; Song, T.; Zhang, Z.; Xue, Z.; Ma, R.; Guan, H. Object guided external memory network for video object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
38. Deng, J.; Pan, Y.; Yao, T.; Zhou, W.; Li, H.; Mei, T. Relation distillation networks for video object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
39. Han, M.; Wang, Y.; Chang, X.; Qiao, Y. Mining inter-video proposal relations for video object detection. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020.
40. Bertasius, G.; Torresani, L.; Shi, J. Object detection in video with spatiotemporal sampling networks. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
41. Xiao, F.; Lee, Y.J. Video object detection with an aligned spatial-temporal memory. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
42. Zhou, Q.; Li, X.; He, L.; Yang, Y.; Cheng, G.; Tong, Y.; Tao, D. TransVOD: End-to-end video object detection with spatial-temporal transformers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 7853–7869. [CrossRef] [PubMed]
43. Zhu, X.; Xiong, Y.; Dai, J.; Yuan, L.; Wei, Y. Deep feature flow for video recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
44. Zhu, X.; Dai, J.; Yuan, L.; Wei, Y. Towards high performance video object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
45. Liu, M.; Zhu, M. Mobile video object detection with temporally-aware feature maps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
46. Chen, K.; Wang, J.; Yang, S.; Zhang, X.; Xiong, Y.; Loy, C.C.; Lin, D. Optimizing video object detection via a scale-time lattice. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
47. Xu, Z.; Hrustic, E.; Vivet, D. Centernet heatmap propagation for real-time video object detection. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020.
48. Shi, Y.; Wang, N.; Guo, X. YOLOV: Making Still Image Object Detectors Great at Video Object Detection. *arXiv* **2022**, arXiv:2208.09686. [CrossRef]
49. Zhang, D.; Zhang, H.; Tang, J.; Hua, X.S.; Sun, Q. Self-regulation for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
50. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
51. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Li, F.-F. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
52. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014.
53. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.

54. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019.
55. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.