# Implicit Motion Handling for Video Camouflaged Object Detection

*Xuelian Cheng[1], *Huan Xiong[3], †Deng-Ping Fan[4], Yiran Zhong[6,7],
Mehrtash Harandi[1,8], Tom Drummond[1], Zongyuan Ge[1,2,5]

[1]Faculty of Engineering, Monash University, [2]eResearch Centre, Monash University
[3]Mohamed bin Zayed University of Artificial Intelligence, [4]CVL, ETH Zurich,
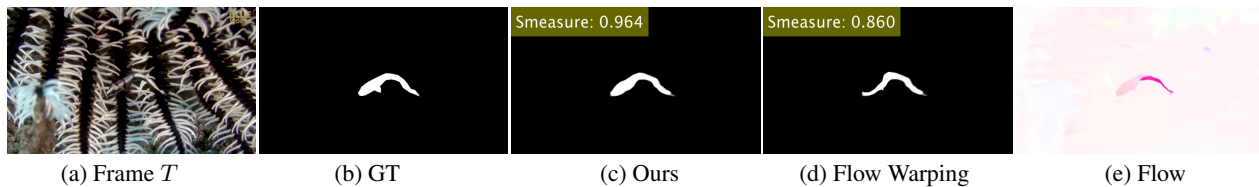[5]Airdoc Research Australia, [6]SenseTime Research, [7]Shanghai AI Laboratory, [8]Data61, CSIRO

Figure 1. Implicit (ours) *vs* Explicit (optical flow based) motion handling. Our result (c) of frame $T$ is generated by the frame $T-1$ and $T$. Figure (d) is the warped result using our result of frame $T-1$ and a pre-computed optical flow (e). As we will discuss, the accumulated error in explicit methods (*i.e.*, optical flow) for camouflage objects can lead to a tangible drop in the accuracy (dropped by 10.79%).

## Abstract

*We propose a new video camouflaged object detection (VCOD) framework that can exploit both short-term dynamics and long-term temporal consistency to detect camouflaged objects from video frames. An essential property of camouflaged objects is that they usually exhibit patterns similar to the background and thus make them hard to identify from still images. Therefore, effectively handling temporal dynamics in videos becomes the key for the VCOD task as the camouflaged objects will be noticeable when they move. However, current VCOD methods often leverage homography or optical flows to represent motions, where the detection error may accumulate from both the motion estimation error and the segmentation error. On the other hand, our method unifies motion estimation and object segmentation within a single optimization framework. Specifically, we build a dense correlation volume to implicitly capture motions between neighbouring frames and utilize the final segmentation supervision to optimize the implicit motion estimation and segmentation jointly. Furthermore, to enforce temporal consistency within a video sequence, we jointly utilize a spatio-temporal transformer to refine the short-term predictions. Extensive experiments on VCOD benchmarks demonstrate the architectural effectiveness of our approach. We also provide a large-scale VCOD dataset*

*named **MoCA-Mask** with pixel-level handcrafted ground-truth masks and construct a comprehensive **VCOD benchmark** with previous methods to facilitate research in this direction. Dataset Link:* `https://xueliancheng.github.io/SLT-Net-project`.

## 1. Introduction

Video Camouflaged Object Detection (VCOD) is the task of discovering objects in a video that, appearance-wise, exhibit a great deal of similarity to the background scene. Despite enjoying wide applications (*e.g.*, surveillance and security [24], autonomous driving [4, 32], medical image segmentation [11, 42], locust detection [17] and robotics [28]), the problem of Camouflaged Object Detection (COD) is a daunting task as camouflaged objects are often indistinguishable to naked-eyes. This, in turn, has made VCOD a relatively under-explored problem in computer vision, as compared to several related problems such as video object detection (VOD) [1, 47], video salient object detection (VSOD) [15], and video motion segmentation (VMS) [16, 46].

In most computer vision tasks (*e.g.*, instance segmentation [51], saliency detection [49]), it is assumed that objects have clear boundaries. This allows us to formulate the problem at the image level and even consider improvements if motion information is available. In contrast, object boundaries are ambiguous and indistinguishable when it comes to detecting camouflaged objects. This not only makes detection from images challenging, but also results

in inaccurate estimation of optical flow and motion cues in videos [36, 37, 52].

The lack of clear boundaries means that the appearance of the camouflaged object resembles the background. This shows itself as two fundamental difficulties: **1)** the object boundaries are often seamlessly blended into the background and is observable only when the object moves; **2)** the object usually has repetitive textures similar to the environment; hence determining the movement of pixels across frames to estimate the motion (*e.g.*, as done in optical flow) is erratic and erroneous. As the first difficulty, to successfully address VCOD, a neural network needs to effectively discover the nuances between the camouflaged object and the background with the help of motion information. Moreover, the motion information is inherently noisy and inaccurate according to the second difficulty, as shown in Figure 1. As such, employing VOD, VSOD, and VMS techniques may fail miserably if naively used or combined to address the VCOD problem.

In this work, we introduce **SLT-Net**, a new method to address VCOD that utilizes short-term dynamics and long-term temporal consistency to detect camouflaged objects in videos. Specifically, we employ a short-term dynamic module to *implicitly* capture the motion between consecutive frames. Rather than using optical flow to *explicitly* represent motions, we use a full-range correlation pyramid strategy to represent them implicitly. The primary motivation behind using a correlation pyramid is that even SOTA optical flow algorithms fail to estimate motions for camouflaged objects and their errors accumulate over the video's duration. Also, it allows us to jointly optimize the motion estimation (implicitly) and the predictions with only the detection supervision. To provide a stable estimation, we further introduce a long-term refinement module to alleviate accumulated inaccuracies in the short-term dynamic module.

We realize the SLT-Net as a hybrid neural network with both transformer and CNN components. In particular, we use a transformer structure to encode features for constructing a correlation pyramid. Aside from its design flexibility, features extracted by the transformer contain global contextual information with long-range dependencies and less inductive bias [31, 40], which we observe to be more distinguishable in estimating the motion.

While the correlation pyramid strategy can effectively capture motions for detecting camouflaged objects, it cannot scale gracefully to long video sequences due to its computational complexity. To solve this issue, we adopt a sequence-to-sequence model with a spatial-temporal transformer to refine the pair-wise prediction with long-term consistency across the videos as we empirically find it is more accurate than the standard ConvLSTM model [44, 53].

Moreover, being a less-explored problem, large-scale datasets are not available to evaluate and benchmark VCOD

systems. To promote new developments in this domain, we have curated a large-scale VCOD dataset based on the Moving Camouflaged Animals (MoCA) [18]. The new dataset, or **MoCA-Mask** for short, contains 87 video sequences with 22,939 frames in total with pixel-wise ground truth masks. MoCA-Mask encapsulates a variety of challenges, such as complex backgrounds and tiny and well-camouflaged objects. We provide annotations, bounding boxes, and dense segmentation masks for every five frames for all the videos in the dataset. We also provide the first comprehensive benchmark for existing VCOD methods. In a nutshell, our contributions are as follows:

- We propose a new VCOD framework that can effectively model short-term dynamics and long-term temporal consistency from videos, where the motion and the camouflage object segmentation can be jointly optimized through a single optimization target.
- We collect the first large-scale VCOD dataset, the **MoCA-Mask** dataset, to promote developments in VCOD as well as a comprehensive VCOD benchmark to facilitate research in VCOD.
- We set a new state-of-the-art on the VCOD task, outperforming a previous SOTA method [45] by 9.88%.

## 2. Related Work

**COD.** Without any prior, even humans can easily miss camouflaged objects. However, once informed that a camouflaged object exists in an image, we can carefully scan the entire image to identify it. Inspired by this fact, ANet [19] incorporated classification stream as the awareness of camouflaged objects and segmentation stream. Sharing a similar idea, SINet [10] and PFNet [27] addressed the problem by first positioning coarse camouflaged objects and then refining it by segmentation. SINet-v2 [8] extended this idea by incorporating the reverse guidance before learning complementary regions. MGL [48] incorporated edge details into the segmentation stream via two graph-based modules. By modeling the conspicuousness of camouflaged objects against backgrounds, Lv *et al.* [25] introduced two new tasks, namely camouflaged object ranking and camouflaged object localization, along with relabeled NC4K dataset.

**VSOD.** To detect salient objects in videos, DLVS [38] introduced fully convolutional networks for pixel-wise saliency prediction. DSR3 [20] exploited an end-to-end 3D neural network to produce video sequences, which incorporates 3D CNN modules combined with recurrent refinement units to predict saliency maps. To better learn temporal information over frames, following works considered SpatioTemporal CRF [21], pyramid dilated convLSTM [33] in the design of their networks. FGRN [23], RCRNet [45] adopted extra flow-guided networks to improve temporal coherence. Later, SSAV [12] specifically focused on the saliency shift phenomenon and established a comprehensive benchmark
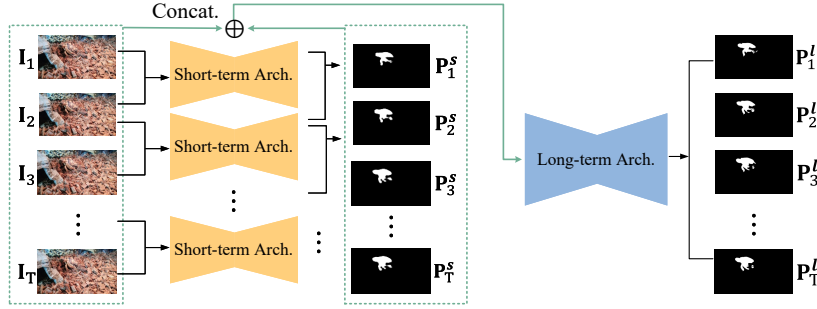
Figure 2. The overall pipeline of the SLT-Net. The SLT-Net consists of a short-term detection module and a long-term refinement module. The short-term detection module takes a pair of consecutive frames and predicts the camouflaged object mask for the reference frame. The long-term refinement module takes T predictions from the short-term detection module along with their corresponding referenced frames to generate the final predictions.

for VSOD. FSNet [15] leveraged the mutual constraints of appearance and motion cues, demonstrating superior performances to many existing methods.

**VMS.** The task of VMS focuses on discovering moving objects in videos. Traditional methods usually address this problem by extracting motion boundaries in the flow field and then refining the initial estimate with appearance features [29], or combining motion and appearance cues by a fusion architecture [13]. Another line of work explicitly leverages optical flow as the input to train a CNN-based network and generate pixel-level motion labels based on supervised learning [35] or in an unsupervised manner [46].

**VCOD.** Different from VMS, visual cues of camouflage objects are considered less effective than motion cues. Prior works mainly relied on homography or optical flows to detect motion patterns. Bidau *et al.* proposed to segment moving objects from the environment by approximating different motion models computed from dense optical flow [2, 3]. In particular, in [2] authors proposed a two-step segmentation algorithm, which first compensated for the camera rotation and then segmented the angle of the optical flow into objects and the background. Although each motion model is updated with optical flow orientations over time, the initial motion is heuristic. In [3], authors used a network to segment the angle field rather than raw optical flow. [18] proposed a video registration and motion segmentation framework, along with a larger camouflaged dataset (MoCA) labeled by bounding boxes for every five frames. The explicit alignment method by optical flow builds spatial correspondence between neighboring frames. However, the optical flow estimation may not be accurate enough to support effective alignment, particularly in dynamic scenes with fast object motions.

## 3. Proposed Framework

The input of our SLT-Net is a video clip containing camouflaged objects, and the output is a set of pixel-wise bi-

nary masks of the camouflaged objects for each frame in the video. Specifically, denote the video clip with T frames by $\{\mathbf{I}^t\}_{t=1}^T, \mathbf{I}^t \in \mathbb{R}^{3 \times H \times W}$, where $H, W$ are the height and the width of the frame. Our network is to assign a binary mask $\mathbf{M}^t \in \{0, 1\}^{H \times W}$ for the video frame $\mathbf{I}^t$ at time $t$.

### 3.1. Overview

The overall framework of the SLT-Net is shown in Figure 2. The SLT-Net consists of a short-term detection module and a long-term refinement module. The short-term detection module takes a pair of consecutive frames and predicts the camouflaged object mask for the reference frame. A sequence-to-sequence translation module is adapted to jointly refine the input video clip frame results with temporal consistency priors. It takes T predictions from the short-term detection module as well as their corresponding referenced frames to generate the final prediction results. To the best of our knowledge, we are the first ones to formulate this dense prediction refining process as a sequence-to-sequence modeling problem.

To train the SLT-Net, we adopt a two-stage strategy. We first train the short-term detection module using pixel-wise annotations only. Once the model converges, we attach the long-term refinement module to the SLT-Net and train the whole model while fixing the short-term detection module.

### 3.2. Short-term Architecture

We illustrate our short-term architecture in Figure 3. It takes two consecutive frames as input from a video and predicts a binary mask of the reference frame. Our model consists of three main modules: (1) **Transformer Encoder** for feature extraction; (2) **Short-term Correlation Pyramid** for capturing short-term dynamics; and (3) **CNN Decoder** to predict the short-term segmentation. Below we describe the details of each module.

**1. Transformer Encoder.** We adopt a Siamese structure with the pyramid vision transformer (PVT) [39] to extract
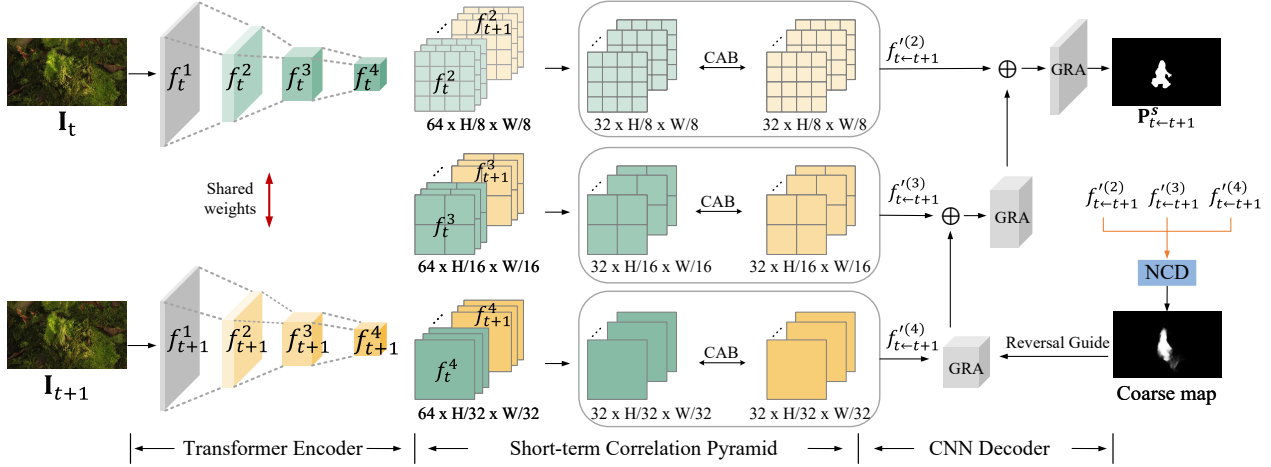
Figure 3. The overview of our short-term pipeline. The network first extracts features from the input frames by a transformer encoder, then computes a full-range volumetric correspondence between the reference frame $\mathbf{I}_t$ and its neighboring frame $\mathbf{I}_{t+1}$ to form a correlation volume pyramid. A CNN decoder is used to predict the final prediction from the motions captured by the short-term correlation pyramid.

features from two consecutive frames. The encoder consists of four stages that generate feature maps at four different scales. All stages share a similar structure, including a patch embedding layer and transformer blocks. The sizes of the features at each stage are $C_i \times H/2^{i+1} \times W/2^{i+1}$, $i \in \{1, 2, 3, 4\}$, where the $H, W, C$ represent the height, the width and the channels. We set $C = 32$ in our experiments. Following [8], we adapt three texture enhanced modules (TEM) for the features from the last three stages. To attain more discriminative feature representations, each TEM includes four parallel residual branches.

**2. Short-term Correlation Pyramid.** Prior works (*e.g.*, [35, 46]) explicitly incorporate motion by taking optical flow from consecutive frames as the inputs into a deep network. However, the inaccurate optical flow may result in error accumulation at subsequent predictions. If we would like to optimize the optical flow module with the segmentation module jointly, the ground truth of optical flow is required. To solve this issue, inspired by [22], we propose a correlation pyramid to capture motion information implicitly. As shown in Figure 3, the CNN decoder directly takes the correlation pyramid as its only input. It means the network can only estimate correct segmentation with correct motion estimation. Also, since the features used to form the correlation pyramid will be updated with the segmentation ground truth, we can use the segmentation ground truth to optimize motion estimations and detection results jointly.

We illustrate the core unit of our correlation pyramid, namely correlation aggregation block (CAB) $\mathbf{C}$, in Figure 4. Given a pair of frame features$\{f_t, f_{t+1}\} \in \mathbb{R}^{C \times H' \times W'}$, the 4D correlation volume $\mathbf{C}(\mathbf{I}_t, \mathbf{I}_{t+1}) \in \mathbb{R}^{H' \times W' \times H' \times W'}$ is defined as:

$$\mathbf{C}(\mathbf{I}_t, \mathbf{I}_{t+1})_{xyuv} = \exp\left(\sum_c \mathcal{F}_\theta(\mathbf{I}_t)_{xyc} \cdot \mathcal{F}_\theta(\mathbf{I}_{t+1})_{uvc}\right), \quad (1)$$

with $c$ being the index along the channel dimension of frame features. With all neighboring features are paired up with correlations, we can find correspondences at a global scale. To reduce the computational complexity, we downsample the adjacent frame by max-pooling over features while keeping the resolution of the reference frame. This design helps the model to learn multi-scale displacement while maintaining high-resolution image details.

Next, we normalize the feature correlation volume $\mathbf{C}(\mathbf{I}_t, \mathbf{I}_{t+1})_{xyuv}$ along the last two dimensions $uv$ over their sum, as they represent the correspondence between the reference and downsampled neighboring feature frame in all the spatial position. The normalized correlation volume is computed as follows:

$$\tilde{\mathbf{C}}(\mathbf{I}_t, \mathbf{I}_{t+1})_{xyuv} = \frac{\mathbf{C}(\mathbf{I}_t, \mathbf{I}_{t+1})_{xyuv}}{\sum_u \sum_v \mathbf{C}(\mathbf{I}_t, \mathbf{I}_{t+1})_{xyuv}}. \quad (2)$$

We apply a convolution operation $\phi(\cdot)$ to selectively consider channel-wise information, and thus obtain a refined feature map $\phi(\mathbf{I}_{t+1}) \in \mathbb{R}^{C \times H' \times W'}$. Specifically, the ag-
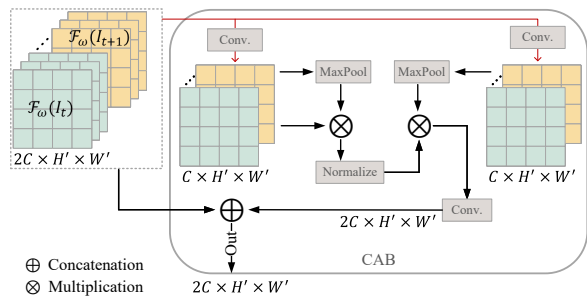


Figure 4. Correlation aggregation block (CAB) computes the normalized correlation volume of feature maps between the reference frame (green blocks) and the neighboring frame (yellow blocks).
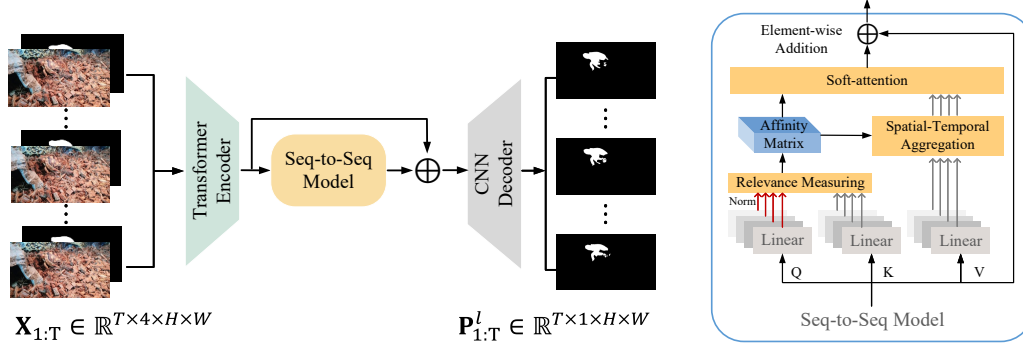
Figure 5. The overview of the proposed long-term consistency architecture. It formulates the process as a seq-to-seq modeling problem and refines the pair-wise predictions with a sequence-to-sequence transformer.

gregated features $f'_{t\leftarrow t+1} = \rho(\mathbf{I}_{t\leftarrow t+1}) \in \mathbb{R}^{C\times H'\times W'}$ was computed as follows:

$$\rho(\mathbf{I}_{t\leftarrow t+1}) = \tilde{\mathbf{C}}(\mathbf{I}_t, \mathbf{I}_{t+1})\phi(\mathbf{I}_{t+1}). \qquad (3)$$

Figure 4 only shows a correlation on one scale. To make the network learn more detailed information, we construct a correlation pyramid $\{\mathbf{C}^i\}, i \in \{2, 3, 4\}$ by incorporating the extracted multi-scale features from the transformer encoder (See details in supplementary materials (Supp) ).

**3. CNN Decoder.** As shown by [8], the neighbor connection decoder is more reliable than conventional connection decoder (*i.e.*, densely connection or short connection). In addition, the group-reversal attention (GRA) strategy used in [8] can provide more accurate segmentation results around the object boundaries. Based on these, we directly feed features from the short-term correlation pyramid, *i.e.*, $\{f'^{(i)}_{t\leftarrow t+1}\} \in \mathbb{R}^{C\times H/2^{i+1}\times W/2^{i+1}}, i \in \{2, 3, 4\}$ into the GRA blocks, and generate refined feature maps. The neighbor connection decoder (NCD) is used to generate a coarse map, which could provide reversal guidance of rough location of the camouflaged object. In this way, we gather the low-level features from the CNN decoder and the high-level features from the correlation pyramid.

**Learning Strategy.** We train the short-term training stage by minimizing the loss below:

$$\mathcal{L} = \mathcal{L}^w_{ce} + \mathcal{L}^w_{iou}. \qquad (4)$$

The weighted cross-entropy loss $\mathcal{L}^w_{ce}$ increases the weights of hard pixels to emphasize their importance. The weighted intersection-over-union loss $\mathcal{L}^w_{iou}$ pays more attention to hard pixels rather than assigning all pixels with equal weights. Readers could refer to prior work [41] to find more details regarding the definitions of these two loss functions.

### 3.3. Long-term Consistency Architecture

Given a sequence of $\mathbf{I}_{1:T} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_T\}$ and the pixel-wise predictions of $\mathbf{P}^s_{1:T} = \{\mathbf{P}^s_1, \mathbf{P}^s_2, \ldots, \mathbf{P}^s_T\}$ from our short-term architecture, we formulate the long-term

consistency refinement process as a seq-to-seq problem. Figure 5 illustrates the long-term consistency architecture. We use the same backbone as the short-term architecture, *i.e.*, transformer encoder and CNN decoder modules, since it has been already pre-trained on camouflaged datasets that could largely accelerate the long-term training processing. For each frame of the input sequence, we concatenate the color frame $\mathbf{I}_t$ with its corresponding prediction $\mathbf{P}^s_t, t \in [1 : T]$ on the channel dimension, and then stack every concatenated frame within the sequence to form a 4D tensor $\mathbf{X}_{1:T} \in \mathbb{R}^{T\times 4\times H\times W}$. The network takes $\mathbf{X}_{1:T}$ as the input and output the final prediction sequence $\mathbf{P}^l_{1:T} \in \mathbb{R}^{T\times 1\times H\times W}$.

There are two kinds of seq-to-seq modeling architecture: one uses convLSTM to model the temporal information, and the other uses a transformer-based seq-to-seq modeling network. We implement both architectures and compare their results in Section 4.4. We empirically find that using the transformer structure can lead to better results, so we select it as our seq-to-seq modeling network to enforce the long-term consistency.

We show the details of the seq-to-seq modeling network on the right side of Figure 5. For each target pixel, to reduce the complexity for building a dense spatial-temporal affinity matrix, we select a fixed number of relevance measuring blocks to construct the affinity matrix within a constrained neighborhood of it. We apply the hybrid loss [9] during the training:

$$\mathcal{L}_{hybrid} = \mathcal{L}^w_{ce} + \mathcal{L}^w_{iou} + \mathcal{L}_e, \qquad (5)$$

where $\mathcal{L}_e$ is the Enhanced-alignment loss, the hybrid loss can guide the network to learn pixel-, object- and image-level features.

## 4. Experiments

This section performs a thorough evaluation of our proposed framework on the CAD dataset and our proposed MoCA-Mask dataset. We also provide a comprehensive VCOD benchmark to facilitate the research of VCOD.

Table 1. Quantitative results on our MoCA-Mask with (w/) and without (w/o) our pseudo labels. The best performing method of each category is highlighted in **bold**. Noting that MG [46] performs unsupervised learning that are trained without labels.

| Models | MoCA-Mask w/o pseudo labels | | | | | | w/ pseudo labels | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $S_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $E_\phi \uparrow$ | $M \downarrow$ | mDic | mIoU | $S_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $E_\phi \uparrow$ | $M \downarrow$ | mDic | mIoU |
| EGNet [50] | 0.547 | 0.110 | 0.574 | 0.035 | 0.143 | 0.096 | 0.546 | 0.105 | 0.573 | 0.034 | 0.135 | 0.090 |
| BASNet [30] | 0.561 | 0.154 | 0.598 | 0.042 | 0.190 | 0.137 | 0.537 | 0.114 | 0.579 | 0.045 | 0.135 | 0.100 |
| CPD [43] | 0.561 | 0.121 | 0.613 | 0.041 | 0.162 | 0.113 | 0.550 | 0.117 | 0.613 | 0.038 | 0.147 | 0.104 |
| PraNet [11] | 0.614 | 0.266 | 0.674 | 0.030 | 0.311 | 0.234 | 0.568 | 0.171 | 0.576 | 0.045 | 0.211 | 0.152 |
| SINet [10] | 0.598 | 0.231 | 0.699 | 0.028 | 0.276 | 0.202 | 0.574 | 0.185 | 0.655 | 0.030 | 0.221 | 0.156 |
| SINet-v2 [8] | 0.588 | 0.204 | 0.642 | 0.031 | 0.245 | 0.180 | 0.571 | 0.175 | 0.608 | 0.035 | 0.211 | 0.153 |
| PNS-Net [14] | 0.544 | 0.097 | 0.510 | 0.033 | 0.121 | 0.101 | 0.576 | 0.134 | 0.562 | 0.038 | 0.189 | 0.133 |
| RCRNet [45] | 0.555 | 0.138 | 0.527 | 0.033 | 0.171 | 0.116 | 0.597 | 0.174 | 0.583 | 0.025 | 0.194 | 0.137 |
| MG [46] | 0.530 | 0.168 | 0.561 | 0.067 | 0.181 | 0.127 | 0.547 | 0.165 | 0.537 | 0.095 | 0.197 | 0.141 |
| **SLT-Net (Ours)** | **0.631** | **0.311** | **0.759** | **0.027** | **0.360** | **0.272** | **0.656** | **0.357** | **0.785** | **0.021** | **0.397** | **0.310** |



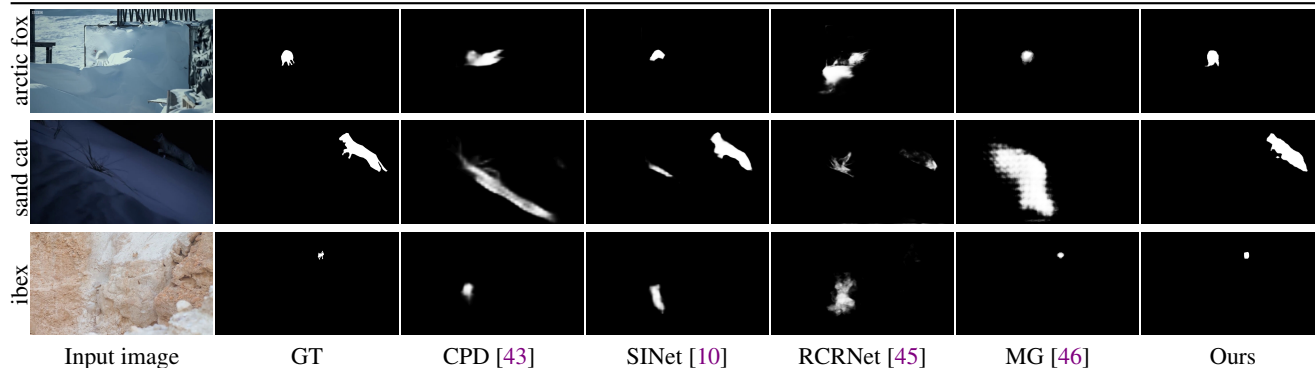| | Input image | GT | CPD [43] | SINet [10] | RCRNet [45] | MG [46] | Ours |

Figure 6. Qualitative results on our MoCA-Mask benchmark. Our model provides more accurate prediction of camouflaged objects in various challenging situations, *i.e.*, unclear appearance (arctic fox), low lighting condition (sand cat), and tiny object (ibex).

## 4.1. Datasets

**COD10K.** We pre-train all still image-based methods as well as the encoder of the video-based methods on COD10K [8]. It is currently the largest COD dataset which consists of 5,066 camouflaged images (3,040 for training, 2,026 for testing), and is divided into five super-classes and 69 sub-classes. This dataset also provides high-quality annotation, reaching the level of matting.

**CAD.** Camouflaged Animal Dataset (CAD) is a small set of camouflaged animals, first introduced by [2]. It includes nine short video sequences in total that were extracted from YouTube videos and accompanying hand-labeled ground-truth masks on every $5^{th}$ frame. We also provide pseudo GT masks by a bidirectional consistency check strategy [34] to enable future studies on this dataset.

**MoCA-Mask.** The original Moving Camouflaged Animals (MoCA) Dataset [18] includes 37K frames from 141 YouTube Video sequences with resolution and sampling rate of $720 \times 1280$ and 24fps in the majority of cases. The dataset covers 67 types of animals moving in natural scenes, but some are not camouflaged animals. Also, the ground truth of the original dataset is bounding boxes rather than dense segmentation masks, which makes it hard to evaluate the VCOD segmentation performance. To this end, we reorganize the dataset as *MoCA-Mask* and build a comprehensive benchmark with more comprehensive evaluation criteria. The modifications could be found in Supp.

## 4.2. Benchmarks

**Metrics.** We adopt the following evaluation metrics to measure the pixel-wise masks: (1) MAE ($M$), which assesses the pixel-level accuracy between prediction and labeled masks. (2) Enhanced-alignment measure ($E_\phi$) [7], which simultaneously evaluates the pixel-level matching and image-level statistics. This metric is naturally suited for assessing the overall and localized accuracy of the camouflaged object detection results. Note that we report mean $E_\phi$ in the experiments. (3) S-measure ($S_\alpha$) [6], which evaluates region-aware and object-aware structural similarity. (4) Weighted F-measure $F_\beta^w$ [26] can provide more reliable evaluation results than the traditional $F_\beta$. (5) mean Dice, which measures the similarity between two sets of data. (6) meanIoU, which measures the overlap between two masks.

**Baseline.** We select nine cutting-edge baselines, including **I.** six image based methods *i.e.*, EGNet [50], BASNet [30], CPD [43], PraNet [11], SINet [10], SINet-v2 [8], and **II.** three video based methods, *i.e.*, PNS-Net [14], RCRNet [45], and MotionGroup [46]. Please refer to the Supp for the implementation details.

**Settings.** We compare our method primarily with the top-performing single image and video baselines. As network architectures, input resolution, modality, pre-processing, and post-processing are all different, we try our best to conduct the comparison as fairly as possible. For sin-

gle image baselines, we adopt the same data pre-processing as [8, 10] for all the compared methods. Specifically, the input images are resized to $352 \times 352$, after random flip, random rotation, and color enhance augmentation. In the training phase, we apply random pepper noise on the GT images. As EGNet [50] requires extra edge/boundary information for training, we adopt the same pre-processing techniques in their paper to obtain the edge maps. This extra information could also be found in our reorganized version of the MoCA-Mask dataset.

Most of the video approaches, *e.g.*, PNS-Net [14], RCR-Net [45], employ a multistage training pipeline. The model is pre-trained using still image datasets and then equipped with temporal modules to process video datasets. We follow this training strategy and pre-train all methods on the COD10K [8] training set, except MotionGroup [46] which does not have a static model. Also, per our practical experience, loading pre-trained weights on the COD10K dataset could further improve the model performance on MoCA-Mask. Compared with the COD10K image dataset, the video dataset MoCA-Mask is more challenging due to the camera motions, blurring images, small ratio of animals, and their tiny body structures, such as slim torso/limbs. In some video sequences, the animals make up a tiny proportion of the entire frame, which makes them extremely hard to be identified (see, for example, ibex in Figure 6). Based upon the considerations above, we provide the results based on the following setting: (a) Training the models on COD10K; (b) Fine-tuning the models on MoCA-Mask, with pre-trained weights on COD10K; (c) Evaluate the models on the whole CAD, the test set of MoCA-Mask.

## 4.3. Results

**Performance on MoCA-Mask.** In Table 1, our approach outperforms all the studied methods by a significant margin, notably by 9.88% on $S_\alpha$ over the best one in this evaluation, RCRNet [45], and 92.97% on $F_\beta^w$ metric over SINet [10]. We also provide the qualitative comparisons of our method and other baselines in Figure 6. Our model can accurately locate and segment camouflaged objects in many challenging situations, such as objects with the tinny torso or complex appearance textures, blur, or abrupt motions. We provide more details, *i.e.*, per-sequence quantitative and qualitative results in the Supp, to illustrate the consistent success over the consecutive frames.

**Performance on CAD.** In Table 2, we assess different approaches by studying their cross-dataset generalization on the CAD dataset. Again, the proposed network obtains the best performance in terms of all six evaluation metrics, further demonstrating its robustness. As shown in Figure 7, our model achieves sharper boundaries with more fine-grained visual details. This benefits from constructing pixel-level correlation pairs in the feature space.

Table 2. Quantitative results on CAD dataset. Bold indicates the best. Our model consistently achieves better performance than other competitors on all metrics.

| Models | $S_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $E_\phi \uparrow$ | $M \downarrow$ | mDic | mIoU |
|---|---|---|---|---|---|---|
| EGNet [50] | 0.619 | 0.298 | 0.666 | 0.044 | 0.324 | 0.243 |
| BASNet [30] | 0.639 | 0.349 | 0.773 | 0.054 | 0.393 | 0.293 |
| CPD [43] | 0.622 | 0.289 | 0.667 | 0.049 | 0.330 | 0.239 |
| PraNet [11] | 0.629 | 0.352 | 0.763 | 0.042 | 0.378 | 0.290 |
| SINet [10] | 0.636 | 0.346 | 0.775 | 0.041 | 0.381 | 0.283 |
| SINet-v2 [8] | 0.653 | 0.382 | 0.762 | 0.039 | 0.413 | 0.318 |
| PNS-Net [14] | 0.655 | 0.325 | 0.673 | 0.048 | 0.384 | 0.290 |
| RCRNet [45] | 0.627 | 0.287 | 0.666 | 0.048 | 0.309 | 0.229 |
| MG [46] | 0.594 | 0.336 | 0.691 | 0.059 | 0.368 | 0.268 |
| **SLT-Net (Ours)** | **0.696** | **0.481** | **0.845** | **0.030** | **0.493** | **0.401** |

Table 3. Ablation on the short-term and long-term modules of SLT-Net on the MoCA-Mask dataset. Bold indicates the best.

| Backbone | Short-term | Long-term | $S_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $E_\phi \uparrow$ | $M \downarrow$ |
|---|---|---|---|---|---|---|
| $\checkmark$ | | | 0.648 | 0.330 | 0.748 | 0.025 |
| $\checkmark$ | $\checkmark$ | | **0.662** | 0.350 | 0.766 | 0.021 |
| $\checkmark$ | $\checkmark$ | $\checkmark$ | 0.656 | **0.357** | **0.785** | **0.021** |

Table 4. Comparing different temporal information handling strategies. We swap the encoder of the RCRNet [45] with our transformer-based encoder to evaluate the performance gain caused by different handling strategies. We use "T" to represent the transformer encoder, "S" for single frame, "V" for video input and $\Delta$ for the improvement. Bold indicates the best.

| Model | $S_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $E_\phi \uparrow$ | $M \downarrow$ |
|---|---|---|---|---|
| RCRNet-TS | 0.597 | **0.206** | **0.618** | 0.043 |
| RCRNet-TV | **0.606** | 0.204 | 0.617 | **0.040** |
| RCRNet-$\Delta$ | 1.51% | -0.97% | -0.16% | 6.98% |
| SLT-Net-TS | 0.648 | 0.330 | 0.748 | 0.025 |
| SLT-Net-TV | **0.656** | **0.357** | **0.785** | **0.021** |
| SLT-Net-$\Delta$ | 1.23% | 8.18% | 4.94% | 16.00% |

## 4.4. Ablation Studies

We perform ablation studies on the MoCA-Mask dataset. In particular, we look into functionality analysis for our short-term and long-term modules, the choice of sequence-to-sequence model, and our pseudo masks.

**Short-term and Long-term Modules.** We evaluate the effectiveness of our short-term and long-term modules in two aspects. We first perform an ablation study on the short-term and the long-term modules on the MoCA-Mask dataset and show the results in Table 3. By adding the short-term module, our performance is improved by 2.16% on $S_\alpha$, 6.06% on $F_\beta^w$, 2.41% on $E_\phi$, 16.00% on $M$, 4.53% on mDic, and 4.84% on mIoU. By adding the long-term module, we further improve our performance by 2% on $F_\beta^w$, while a slight drop 0.91% on $S_\alpha$.

We then swap the encoder of a SOTA VSOD method RCRNet [45] with our transformer based encoder to compare the effectiveness of the temporal information handling strategies between ours and the RCRNet in Table 4. In terms of its spatiotemporal coherence model, it shows both positive and negative gains on the evaluated metric, *i.e.*, 1.51%
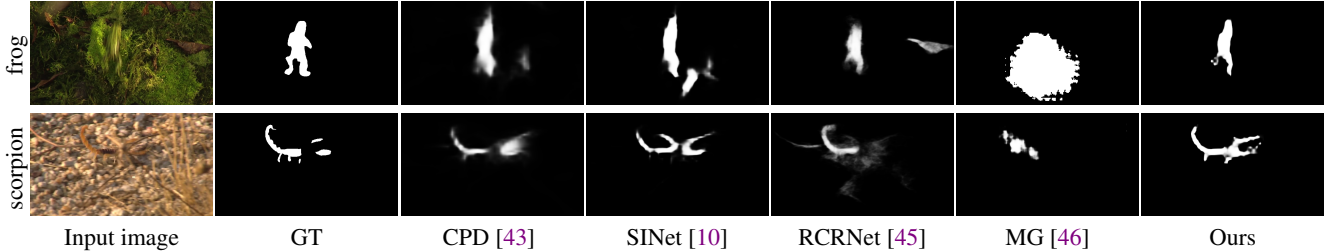
Figure 7. Qualitative results on CAD dataset. As shown, our model can predict more fine-grained detail (scorpion) and work for abrupt motion (frog), which benefits from dense correspondence pair of the feature volume.

Table 5. Ablation studies of different long-term architectures on MoCA-Mask test set. The input resolution is $256 \times 448$.

| Arch. Variant | Params | $S_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $E_\phi \uparrow$ | $M \downarrow$ |
|---|---|---|---|---|---|
| ConvLSTM | 179.03 MB | 0.651 | 0.348 | 0.767 | **0.021** |
| Transformer | 82.30 MB | **0.656** | **0.357** | **0.785** | **0.021** |

on $S_\alpha$, -0.97% on $F_\beta^w$, -0.16% on $E_\phi$, 6.98% on $M$.

**Transformer *v.s.* ConvLSTM.** We evaluate two different approaches for constructing long-term architecture, namely transformer based model, and ConvLSTM based model. For the latter ConvLSTM network variant, we adopt a sequence model proposed in [5] but modify the original VGG-style network for the CNN encoder and decoder with our transformer-style backbone network. From the Table 5, we can observe that the transformer variant is more accurate than the ConvLSTM model in all four metrics, with a much smaller number of parameters.

**Pseudo Masks.** As shown in Table 1, although the generated pseudo labels contain some noises, they can improve the performance of video approaches as they can leverage temporal information to suppress the label noises. For still image baselines, almost all of them are seriously effected by the label noises, leading to worse performance than the one without pseudo labels. It also proves that the motion estimation error can not be overlooked in the VCOD problem and we should jointly optimize it with the segmentation error for a better performance.

**Trained from scratch on MoCA-Mask.** For the sake of completeness, we provide the accuracy of our network with/without pre-trained weights in Table 6. It shows that the gap between the train-from-scratch and the pre-trained model is minor, *i.e.*, only a slight drop 0.15% on $S_\alpha$.

Table 6. Comparison of trained from scratch and using pre-trained models on MoCA-Mask.

| Model | $S_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $E_\phi \uparrow$ | $M \downarrow$ |
|---|---|---|---|---|
| Trained from scratch | 0.655 | 0.351 | 0.764 | 0.024 |
| Pre-trained | **0.656** | **0.357** | **0.785** | **0.021** |

**Generalization.** Our model can be applied to the more general video object detection problem, such as video instance segmentation. Except a detailed comparison with MG [46] in Table 1, we compare with [18] on DAVIS16

Table 7. Comparison with [18] on DAVIS16.

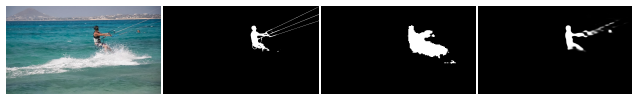| Model | $\mathcal{J}_{Mean} \uparrow$ | $\mathcal{J}_{Recall} \uparrow$ | $\mathcal{F}_{Mean} \uparrow$ | $\mathcal{F}_{Recall} \uparrow$ |
|---|---|---|---|---|
| [18] | 65.3 | 77.3 | 65.1 | 74.1 |
| SLT-Net | 77.96 | 95.49 | 78.65 | 92.08 |



Figure 8. Qualitative Results on DAVIS16. MG [46] fails where the splash created by the person is incorrectly included in the predicted segment. This mainly due to the inaccurate optical flow estimation which cannot be optimized during model training.

(Table 7) and demonstrate superiority of our method.

## 5. Conclusion

We presented a new SLT-Net framework for learning to segment camouflaged objects in a video. Specifically, we proposed a short-term module to implicitly capture motions between consecutive frames which allows us to learn motion estimation and segmentation in a single optimization target. We also proposed a long-term module with a sequence-to-sequence transformer to enforce temporal consistency in video sequence. To promote the development of this field, we rebuild a new dataset called **MoCA-Mask** with 87 high-quality video sequences, including 22,939 frames in total. It is the largest-scale pixel-level annotated dataset that allows object-level benchmark in video camouflaged object detection (VCOD). Compared with existing state-of-the-art baselines, our proposed network achieves fascinating results on two VCOD benchmarks.

**Broader Impact.** Camouflaged object can be used to detect and protect rare animal species, prevent wildlife trafficking, medical applications (*e.g.*, detecting polyp or lung infection) and search-and-rescue work to name a few. Please note that our MoCA-Mask dataset does not contain any military or sensitive scenes. Aside from its important use-cases as mentioned above, our paper takes a solid step into understanding video contents when motion information is noisy.

# References

[1] Sara Beery, Guanhang Wu, Vivek Rathod, Ronny Votel, and Jonathan Huang. Context r-cnn: Long term temporal context for per-camera object detection. In *CVPR*, 2020. 1

[2] Pia Bideau and Erik Learned-Miller. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In *ECCV*, 2016. 3, 6

[3] Pia Bideau, Rakesh R Menon, and Erik Learned-Miller. Moa-net: self-supervised motion segmentation. In *ECCV Workshops*, 2018. 3

[4] Xuelian Cheng, Yiran Zhong, Yuchao Dai, Pan Ji, and Hongdong Li. Noise-aware unsupervised deep lidar-stereo fusion. In *CVPR*, 2019. 1

[5] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *ICML*, 2018. 8

[6] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A New Way to Evaluate Foreground Maps. In *ICCV*, 2017. 6

[7] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment Measure for Binary Foreground Map Evaluation. In *IJCAI*, 2018. 6

[8] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE TPAMI*, 2021. 2, 4, 5, 6, 7

[9] Deng-Ping Fan, Ge-Peng Ji, Xuebin Qin, and Ming-Ming Cheng. Cognitive vision inspired object segmentation metric and loss function. *SSI*, 2021. 5

[10] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *CVPR*, 2020. 2, 6, 7, 8

[11] Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Pranet: Parallel reverse attention network for polyp segmentation. In *MICCAI*, 2020. 1, 6, 7

[12] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *CVPR*, 2019. 2

[13] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, 2017. 3

[14] Ge-Peng Ji, Yu-Cheng Chou, Deng-Ping Fan, Geng Chen, Huazhu Fu, Debesh Jha, and Ling Shao. Progressively normalized self-attention network for video polyp segmentation. In *MICCAI*, 2021. 6, 7

[15] Ge-Peng Ji, Keren Fu, Zhe Wu, Deng-Ping Fan, Jianbing Shen, and Ling Shao. Full-duplex strategy for video object segmentation. In *ICCV*, 2021. 1, 3

[16] Pan Ji, Yiran Zhong, Hongdong Li, and Mathieu Salzmann. Null space clustering with applications to motion segmentation and face clustering. In *ICIP*, 2014. 1

[17] Karthika Suresh Kumar and Aamer Abdul Rahman. Early detection of locust swarms using deep learning. In *MLCI*. 2021. 1

[18] Hala Lamdouar, Charig Yang, Weidi Xie, and Andrew Zisserman. Betrayed by motion: Camouflaged object discovery via motion segmentation. In *ACCV*, 2020. 2, 3, 6, 8

[19] Trung-Nghia Le, Tam V. Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *CVIU*, 2019. 2

[20] Trung-Nghia Le and Akihiro Sugimoto. Deeply supervised 3d recurrent fcn for salient object detection in videos. In *BMVC*, 2017. 2

[21] Trung-Nghia Le and Akihiro Sugimoto. Video salient object detection using spatiotemporal deep features. *IEEE TIP*, 2018. 2

[22] Dongxu Li, Chenchen Xu, Kaihao Zhang, Xin Yu, Yiran Zhong, Wenqi Ren, Hanna Suominen, and Hongdong Li. Arvo: Learning all-range volumetric correspondence for video deblurring. In *CVPR*, 2021. 4

[23] Guanbin Li, Yuan Xie, Tianhao Wei, Keze Wang, and Liang Lin. Flow guided recurrent neural encoder for video salient object detection. In *CVPR*, 2018. 2

[24] Ting Liu, Yao Zhao, Yunchao Wei, Yufeng Zhao, and Shikui Wei. Concealed object detection for activate millimeter wave image. *IEEE TIE*, 2019. 1

[25] Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *CVPR*, 2021. 2

[26] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *CVPR*, 2014. 6

[27] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. Camouflaged object segmentation with distraction mining. In *CVPR*, 2021. 2

[28] Jeff Michels, Ashutosh Saxena, and Andrew Y Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In *ICML*, 2005. 1

[29] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013. 3

[30] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *CVPR*, 2019. 6, 7

[31] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. In *ICLR*, 2022. 2

[32] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *CVPR*, 2019. 1

[33] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*, 2018. 2

[34] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 6

[35] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *CVPR*, 2017. 3, 4

[36] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, Kaihao Zhang, Nikolai Smolyanskiy, and Hongdong Li. Deep two-view structure-from-motion revisited. In *CVPR*, 2021. 2

[37] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Kaihao Zhang, Pan Ji, and Hongdong Li. Displacement-invariant matching cost learning for accurate optical flow estimation. In *NeurIPS*, 2020. 2

[38] Wenguan Wang, Jianbing Shen, and Ling Shao. Video salient object detection via fully convolutional networks. *IEEE TIP*, 2017. 2

[39] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv preprint arXiv:2106.13797*, 2021. 3

[40] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021. 2

[41] Jun Wei, Shuhui Wang, and Qingming Huang. F$^3$net: Fusion, feedback and focus for salient object detection. In *AAAI*, 2020. 5

[42] Yu-Huan Wu, Shang-Hua Gao, Jie Mei, Jun Xu, Deng-Ping Fan, Rong-Guo Zhang, and Ming-Ming Cheng. Jcs: An explainable covid-19 diagnosis system by joint classification and segmentation. *IEEE TIP*, 2021. 1

[43] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection. In *CVPR*, 2019. 6, 7, 8

[44] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, 2015. 2

[45] Pengxiang Yan, Guanbin Li, Yuan Xie, Zhen Li, Chuan Wang, Tianshui Chen, and Liang Lin. Semi-supervised video salient object detection using pseudo-labels. In *ICCV*, 2019. 2, 6, 7, 8

[46] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *ICCV*, 2021. 1, 3, 4, 6, 7, 8

[47] Zhao Yang, Qiang Wang, Luca Bertinetto, Weiming Hu, Song Bai, and Philip HS Torr. Anchor diffusion for unsupervised video object segmentation. In *CVPR*, 2019. 1

[48] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan. Mutual graph learning for camouflaged object detection. In *CVPR*, 2021. 2

[49] Jing Zhang, Deng-Ping Fan, Yuchao Dai, Xin Yu, Yiran Zhong, Nick Barnes, and Ling Shao. Rgb-d saliency detection via cascaded mutual information minimization. In *ICCV*, 2021. 1

[50] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet:edge guidance network for salient object detection. In *ICCV*, 2019. 6, 7

[51] Yiran Zhong, Yuchao Dai, and Hongdong Li. 3d geometry-aware semantic labeling of outdoor street scenes. In *ICPR*, 2018. 1

[52] Yiran Zhong, Pan Ji, Jianyuan Wang, Yuchao Dai, and Hongdong Li. Unsupervised deep epipolar flow for stationary or dynamic scenes. In *CVPR*, 2019. 2

[53] Yiran Zhong, Hongdong Li, and Yuchao Dai. Open-world stereo video matching with deep rnn. In *ECCV*, 2018. 2