IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DEEP LEARNING TECHNOLOGIES FOR INTERNET OF VIDEO THINGS

Received February 26, 2021, accepted March 14, 2021, date of publication March 22, 2021, date of current version March 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3067928

Noisy-LSTM: Improving Temporal Awareness for Video Semantic Segmentation

BOWEN WANG[1], LIANGZHI LI[1], YUTA NAKASHIMA[1], (Member, IEEE), RYO KAWASAKI[2], HAJIME NAGAHARA[1], (Member, IEEE), AND YASUSHI YAGI[3], (Senior Member, IEEE)

[1]Institute for Datability Science (IDS), Osaka University, Suita 565-0871, Japan
[2]Graduate School of Medicine, Osaka University, Suita 565-0871, Japan
[3]Institute of Scientific and Industrial Research, Osaka University, Ibaraki 567-0047, Japan

Corresponding author: Bowen Wang (bowen.wang@is.ids.osaka-u.ac.jp)

This work was supported in part by the Council for Science, Technology and Innovation (CSTI), in part by the Cross-Ministerial Strategic Innovation Promotion Program (SIP), in part by the National Institute of Biomedical Innovation, Health and Nutrition (NIBIOHN) through the Innovative AI Hospital System, and in part by the Japan Society for The Promotion of Science (JSPS) KAKENHI under Grant 19K10662 and Grant 20K23343.

ABSTRACT Semantic video segmentation is a key challenge for various applications. This paper presents a new model named Noisy-LSTM, which is trainable in an end-to-end manner, with convolutional LSTMs (ConvLSTMs) to leverage the temporal coherence in video frames, together with a simple yet effective training strategy that replaces a frame in a given video sequence with noises. Our training strategy spoils the temporal coherence in video frames and thus makes the temporal links in ConvLSTMs unreliable; this may consequently improve the ability of the model to extract features from video frames and serve as a regularizer to avoid overfitting, without requiring extra data annotations or computational costs. Experimental results demonstrate that the proposed model can achieve state-of-the-art performances on both the CityScapes and EndoVis2018 datasets. The code for the proposed method is available at https://github.com/wbw520/NoisyLSTM.

INDEX TERMS Video semantic segmentation, noisy training, temporal awareness.

I. INTRODUCTION

The ever-increasing importance of video semantic segmentation has attracted increasing attention from an extensive number of computer vision researchers. Due to the rapid development of convolutional neural networks (CNNs) [1], [2], it is fair to say that the performances of video semantic segmentation methods have been dramatically improved. A simple yet effective approach is to treat video frames as independent images and use image segmentation models. This approach benefits from many existing image segmentation models [3]–[5] and a large number of datasets available for training [6], [7].

However, these methods usually suffer from segmentation errors, such as inaccurate object boundaries, incomplete regions that only cover parts of certain objects, and overcomplete regions that cover neighboring objects. Due to the deteriorated imaging quality caused by video capturing and encoding processes, these segmentation errors occur much more frequently in video semantic segmentation tasks. An important observation is that these errors only exist in some frames, while other frames, including adjacent frames, may still yield accurate predictions.

Based on this observation, researchers have developed new models dedicated to video semantic segmentation that utilize temporal coherence. There are some works that use optical flow [8]–[10], but the computation of optical flow itself is a nontrivial problem that depends greatly on the motion dynamics in adjacent frames. It is difficult to design a robust and accurate method for estimating optical flow for a variety of videos.

Another possible way to leverage the temporal coherence is to introduce temporal structures into models. One pioneering approach is to use conditional random fields (CRFs) on top of a model for a single image [11], [12]. However, their CRFs have no access to the internal representations in the CNNs, possibly spoiling their potential to improve the segmentation results. Recurrent neural networks (RNNs) provide further flexibility, and there have been a series of works [13]–[16] using RNNs to model the dependency among adjacent frames. However, additional links in the temporal dimension introduce more model parameters to be trained,

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

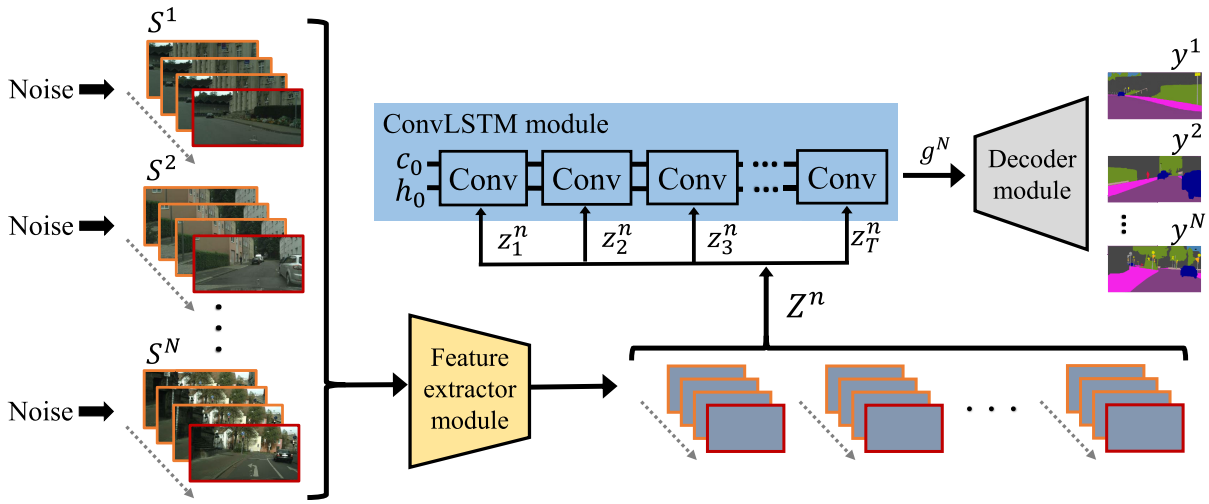

FIGURE 1. Overview of the proposed Noisy-LSTM model for video semantic segmentation. For one input sequence in the left, the context frames are marked with an orange outline and the target frame is marked with red. The noise will be inserted into the context frames.

and this may require more training data. In particular, most RNN-based models need a large amount of labeled data for training, and such data may not always be available for many applications.

Data augmentation is a possible way to fix these kinds of problems. Recent techniques for training neural networks sometimes use noises. For example, dropout and its related techniques [17] inject noises into latent representations to regularize the training process. Some methods add noises even to input images for data augmentation purposes [18]. Xie *et al.* [19] proposed to use unlabeled data, which served as noises for training, in a teacher-student framework. The experimental results in these works demonstrate that using noises during training is an easy yet effective way to improve the performance.

In this paper, we propose a new method, named Noisy-LSTM, which uses convolutional LSTM (ConvLSTM [20]) to facilitate the temporal continuity for improving video semantic segmentation tasks. ConvLSTM can be directly applied to existing semantic segmentation models. We add it into PSPNet [5] and ICNet [21]. Inspired by [19], we adopt a new noisy training strategy to further improve the ability of our method to utilize temporal coherence. It is a new kind of data augmentation that can be used with other data augmentation techniques simultaneously. As shown in Fig. 1, the Noisy-LSTM model is based on a feature extractor and extended with ConvLSTM to leverage the temporal coherence. Noisy-LSTM can be applied to all common semantic segmentation models. Noisy-LSTM uses multiple sequences as input, into which noise frames are added. All frames in these sequences are compiled into a single batch and are fed into a shared CNN, in which batch normalization stabilizes the training process. The resulting feature maps are rearranged into the original sequences, and each of them goes through the ConvLSTM module to make use of their temporal dynamics for prediction. Ultimately, the decoder generates semantic segmentation results.

Our three main contributions are as follows:

- We develop a video semantic segmentation method that makes use of the temporal coherence in video frames with ConvLSTM. In addition, we adopt the batch norm for video sequences, which can stabilize the model training.
- We also enhance the model's temporal awareness by using a noisy training strategy. Without any extra data annotation or computational costs, this strategy can control the reliability of temporal connections. The model is robust to occasional and rare changes in frames, which cannot be handled by a ConvLSTM-based network alone.
- We experimentally demonstrate the performance of our method on the Cityscapes and EndoVis2018 datasets.

II. RELATED WORK

In this section, we briefly review the representative literature.

A. TIME-SEQUENCE SEMANTIC SEGMENTATION

Most approaches are designed only for image segmentation and not for video tasks. This means that the temporal coherence of a given video is not considered and that each frame of a video sequence is predicted independently.

A common approach to deal with temporal coherence is to use RNN-based structures such as long short-term memory (LSTM) networks [22]. In addition to fully convolutional networks (FCNs) [23], Valipour *et al.* introduced the recurrent fully convolutional network (RFCN) [13]. They added a recurrent unit between the encoder and the decoder in an FCN and achieved better performances on the SegTrack, Davis, and Moving MNIST datasets. Yurdakul and Yemez [14] evaluated different kinds of RNN-based structures, such as

ConvRNN, ConvGRU, and ConvLSTM, on the virtual KITTI dataset [24] and concluded that ConvLSTM had the best performance. Nilsson and Sminchisescu [9] used optical flow to represent changes between adjacent frames and applied a ConvGRU structure to encode the temporal continuity. Besides, they used unlabeled frames to further improve the prediction performance. Rochan *et al.* [15] adopted bidirectional ConvLSTM for future frame prediction. They added a ConvLSTM structure between each layer in the encoder and the decoder, merging temporally adjacent feature maps to predict the target frame. Pfeuffer *et al.* [16] applied ConvLSTM at different positions in some state-of-the-art models and demonstrated that ConvLSTM worked well when inserted in most positions.

Different from previous works that process a video sequence during one training iteration, our method adopts batch dimensions in the ConvLSTM structure to increase the training's stability, which is proven to be important in our experiments. We also apply the noisy strategy to video sequences, and this further improves the prediction accuracy.

B. TRAINING WITH NOISES

For the training processes of deep models, insufficient training data is a crucial issue that causes overfitting. To avoid this issue, various ways to use noises during training have been proposed. Dropout [25] is one of them, and it is performed by adding noises to latent representations in neural networks. Some variants of dropout have also been proposed [17]. Data augmentation by additional noises has also been considered [18], where the equivalence between data augmentation by noise and dropout was noted [26]. Recently, using unlabeled data to improve the model performance was proven. Xie *et al.* [19] proposed a self-training method, named Noisy Student, to improve the classification performance on the ImageNet dataset. 300M unlabeled images, many of which were from different domains, were used to enhance the feature extraction ability of the student model. They applied a teacher-student framework in semantic segmentation tasks for images and presented a new model compression method that can result in models with a good performance while requiring far fewer parameters.

In this paper, we also use unlabeled data to improve the segmentation performance of our model. One of the biggest differences is that, unlike teacher-student frameworks, our strategy does not require a dual-network structure, temporarily generated labels, and an iterative training process, which cost additional time and resources. Our noisy training strategy is a new kind of data augmentation approach for temporal data that is different from any existing method. This is a simple training strategy but the experimental results prove its effectiveness. We borrow the insight that adding noises during training enhances the feature extraction capability of a model. With this strategy, we expect that the model is robust to occasional and rare changes in frames. These are scenarios that cannot be handled by a ConvLSTM-based network alone. Our experiments in Section IV-Cd prove the importance of such robustness.

III. METHODOLOGY

A. OUR MODEL

As shown in Fig. 1, the proposed model mainly consists of three components: a feature extraction module, a ConvLSTM module, and a decoder module. It takes multiple sequences in a batch $S = \{S^n | n = 1, \ldots, N\}$ as input, where N is the batch size for sequences, and it produces a single segmentation result for each sequence as output. An input sequence $S^n = \{s_t^n | t = 1, \ldots, T\}$ contains T frames, where the last frame s_T^n is the target frame, for which our model produces the segmentation result y^n, and other frames s_t^n for $t \neq T$ contextualize s_T^n. In other words, $\{s_t^n | t \neq T\}$ are the context frames for the target frame s_T^n in an input video sequence. Our noisy training strategy replaces some of the s_t^n's with noise frames. This will be described in Section III-C. T is fixed in our implementation, and thus all input sequences have the same length T. Note that s_t^n and s_{t+1}^n are not necessarily consecutive in the original video sequence, but they can be frames separated by a fixed number of frames.

For the PSPNet-based model, our feature extraction module adopts ResNet-101 [27] as the backbone network. We replace the last two convolution layers of ResNet-101 with dilated convolutions [1] of size 3×3 at rates of 2 and 4 to enlarge the receptive field and remove fully connected layers in the original ResNet-101. Batch normalization (BN) is of great value for training deep models [16], but it requires diversity in an input batch; otherwise, it may cause severe performance degradation [28]. This is a serious problem for models that deal with temporal sequences because they only input frames from the same video sequences, and these may not offer sufficient diversity. This is the main reason why most LSTM-based video segmentation models [9], [16] do not have BN layers. To address this issue (shown in Fig. 1), in the training stage, we sample target frames s_T^n randomly from all frames (with labels) in the training set and then aggregate context frames for each target frame to form a sequence S^n. Additionally, the feature extraction module is not aware of the sequence structure; it flattens all sequences into a set of $T \times N$ frames so that we can easily apply BN for feature extractor with batch size $T \times N$. We denote the feature maps obtained from s_t^n, which is the output of the second dilated convolution layer, by z_t^n. All z_t^n are rearranged into Z^n with the same sequence structure as that of the model input. Therefore, the ConvLSTM module can process multiple sequence data with a batch size N. We show the importance of BN for model performance in our experiments in Section IV-Cc.

The ConvLSTM module encodes the temporal sequence into a single feature map; this process will be detailed in the next section. A previous work [16] proved that ConvLSTM can be used for various stages (*i.e.*, layers) in various model architectures. We place the ConvLSTM module between the

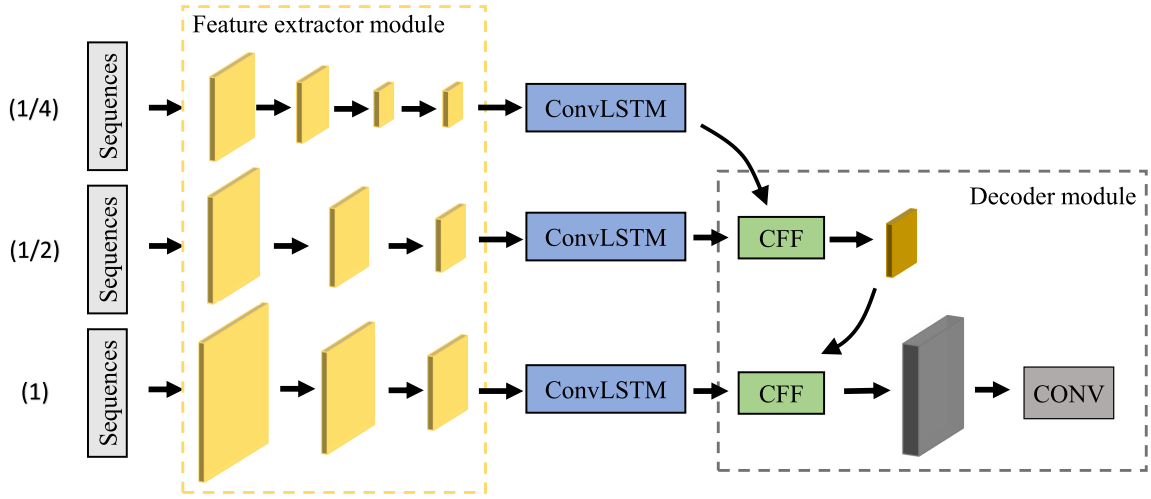

FIGURE 2. ICNet-based Noisy-LSTM. We add the ConvLSTM module after the feature extractor of each branch, and the output features are aggregated by the CFF module.

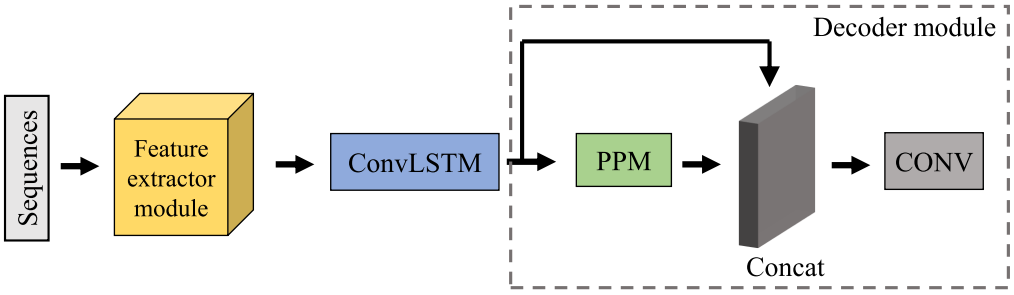

FIGURE 3. PSPNet-based Noisy-LSTM. We add ConvLSTM after the CNN feature extractor, and the output features go through the PPM to generate the final prediction.

feature extractor module and the decoder module. The output of the ConvLSTM module can be represented by

$$g^n = \text{ConvLSTM}\left(Z^n\right), \tag{1}$$

where $Z^n = \{z_t^n | t = 1, \ldots, T\}$.

Finally, the decoder module takes output g^n from the ConvLSTM module and produces semantic segmentation result y^n for target frame s_T^n of input sequence S^n.

In this paper, we apply Noisy-LSTM to ICNet [21] and PSPNet [5] and the model structures are shown in Fig. 2 and Fig. 3, respectively. For the ICNet-based Noisy-LSTM, we directly add a ConvLSTM module at the end of each branch and the output features are aggregated by the cascade feature fusion (CFF) module. In the PSPNet-based Noisy-LSTM, a ConvLSTM module is placed between the CNNs feature extractor module and the decoder module consisting of a pyramid pooling module (PPM), two convolutional layers, and an upsampling layer.

In what follows, we detail our network design for modeling the temporal dependency through ConvLSTM and for enhancing temporal awareness by the noisy training strategy.

B. ENCODING TEMPORAL DEPENDENCY

It has been proved that ConvLSTM is a powerful tool for capturing spatio-temporal dependency, which is important for semantic segmentation in video [16]. LSTM cells can learn how to handle information from preceding frames during training and are able to obtain temporal information over a certain period. In contrast to LSTMs for fully connected layers [29], ConvLSTMs use a convolutional layer as the latent state, and this is more suitable for vision tasks. We use a single ConvLSTM module and set the kernel size to 3×3. The segmentation result for the target frame ($t = T$) is given based on its own feature maps and those of the preceding ($t = 1, \ldots, T-1$) frames.

As shown in Fig. 1, the feature map from each input frame is sequentially fed into the ConvLSTM layer to obtain the feature map, based on which the segmentation result for the target frame is computed. Formally, from the feature map z_t for the t-th frame in the input sequence S (we omit the superscript n for notation simplicity), g is computed as the last latent state of the ConvLSTM layer as follows:

$$\begin{aligned} i_t &= \sigma(W_\text{i} * z_t + V_\text{i} * h_{t-1} + U_\text{i} \otimes c_{t-1} + b_\text{i}) \\ f_t &= \sigma(W_\text{f} * z_t + V_\text{f} * h_{t-1} + U_\text{f} \otimes c_{t-1} + b_\text{f}) \end{aligned}$$

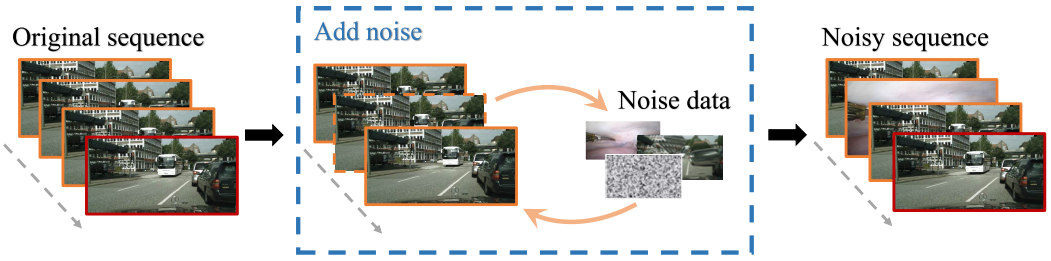

FIGURE 4. Our noisy training strategy introduces noises in the time domain during the training process by replacing some context frames in the sequence with noise frame. (In this sample, among context frames, the dotted line marked frame is replaced by an unrelated image).

$$
\begin{aligned}
c_t &= f_t \otimes c_{t-1} + i_t \otimes \tanh(W_{\mathrm{c}} * z_t + V_{\mathrm{c}} * h_{t-1} + b_{\mathrm{c}}) \\
o_t &= \sigma(W_{\mathrm{o}} * z_t + V_{\mathrm{o}} * h_{t-1} + U_{\mathrm{o}} \otimes c_t + b_{\mathrm{o}}) \\
h_t &= o_t \otimes \tanh(c_t), \qquad (2)
\end{aligned}
$$

where $*$ and $\otimes$ are the convolution operations and the element-wise product, respectively; σ and tanh are the sigmoid and hyperbolic tangent nonlinearities. i_t, f_t, and o_t are the input, forget, and output gates, respectively; c_t and h_t are the cell and the latent state, where $g = h_T$. W_l and V_l for $l \in \{\mathrm{i}, \mathrm{f}, \mathrm{c}, \mathrm{o}\}$ are trainable convolution kernels; U_l and b_l are trainable parameters of the same size as z_t. Multiple ConvLSTM modules can be stacked and temporally concatenated to form highly complex structures, and this may further improve performance. In our network, we only use a single ConvLSTM module.

C. ENHANCING TEMPORAL AWARENESS

For video tasks, the temporal coherence between frames is often leveraged for improving model performance. However, there might be some cases in which this negatively affects performance. For example, in surgery videos, consecutive frames usually have small motions and occasionally exhibit large motions. Such rare events may not be effectively learned with RNN-based models.

For neural network training, a number of attempts have been made to utilize noises in various ways for the sake of regularization [26], [30]. Recently, some studies demonstrated that a large amount of unlabeled data, which may serve as noise during training, can improve the performance of teacher-student networks for semantic segmentation and classification [19], [31]. Inspired by these works, we propose a noisy training strategy that replaces some frames in the input sequences with unlabeled and random images. This noise injection in the time domain stochastically spoils the temporal dependency in the original sequence and may consequently improve the capability of the model to perform feature extraction from individual frames as the temporal continuity is no longer reliable; thus, we can expect improved temporal awareness in the model.

Specifically, for each sequence, we replace some context frames with random frames, which are unlabeled random images with much different content, as shown in Fig. 4. For example, we may use handwriting images, frames in TV drama series, or medical images as noise to replace frames when dealing with street-view sequences. Even a random tensor can be used as one type of noises. We attempt to use three kinds of noises. They are described in Section IV-Cb. The target frame is not replaced, so we can still use its ground-truth label. In addition, due to the structural characteristics of our model, the feature maps from context frames are used solely for enhancing the target frame's feature map, and the output from the model is the segmentation result for the target frame. This means that there is no need to generate, *e.g.*, pseudo labels for noises, which were required in [19], [31]. Therefore, the noisy training strategy requires no extra computation or annotation.

To add noises, each context frame (*i.e.*, $s_1, \ldots, s_{T-1}$) is randomly replaced with a noise frame at a probability of p, which is set to 50% in our implementation. We also limit the number of frames to be replaced to half of the sequence length (*i.e.*, $T/2$). This means that there are no more than two replaced frames in our experiment (We set T as 4 in this paper).

IV. EXPERIMENTS

To evaluate our model trained with the noisy training strategy, we used two video semantic segmentation datasets in completely different domains, *i.e.*, Cityscapes [33] and EndoVis2018 [34]. Frames in one dataset were used as noise when training our model for the other dataset, whereas labels in the dataset used as noise were not used during this process. We adopted data augmentation including rotation (with angles between -10 and 10), random horizontal flipping, and so on, in all the experiments. When training with temporal data, all the input frames in one sequence were calculated by the same data augmentation. The mean of class-wise intersection over union (mean IoU) is used to evaluate the performance.

We used cross entropy as the loss function and Adam [35] as optimizer with an initial learning rate of 10^{-4}; this rate was decreased by a factor of 10 halfway through the training process. The training was terminated after 40 epochs for Cityscapes and 30 epochs for EndoVis2018. The length T of the sequence was set to 4, and the number of sequences N was also set to 4. The hidden state h_0 and cell c_0 were zero initialized. The model was implemented in the Pytorch [36]

TABLE 1. Comparison among ours and state-of-the-art methods on the Cityscapes and EndoVis datasets in mIoU (%). The best performance for each configuration is highlighted in bold.

	Cityscapes		EndoVis2018
Models	Validation	Test	Validation
FCN-8s [23]	64.3	-	47.9
DeepLab-v3 [3]	71.8	-	56.2
DANet [4]	68.7	-	56.0
PSPNet (baseline) [5]	71.6	71.0	59.8
ICNet (baseline) [21]	60.0	59.5	52.1
DynamicCRF [32]	64.5	-	-
Pfeuffer *et al.* [16]	62.3	-	-
GRFP [9]	**73.6**	**72.8**	-
Noisy-LSTM (ICNet) (*w/o* noisy training)	61.2	60.5	53.6
Noisy-LSTM (ICNet)	62.5	61.6	54.8
Noisy-LSTM (PSPNet) (*w/* noisy training)	72.2	71.7	61.1
Noisy-LSTM (PSPNet)	73.0	**72.8**	**62.3**

framework and we ran the model on a Tesla V100 GPU with 32 GB memory.

A. CITYSCAPES DATASET

The Cityscapes dataset contains a total of 5,000 video sequences of high-resolution frames ($2,048 \times 1,024$), and it is partitioned into training, validation, and test sets with 2,975, 500, and 1,525 sequences, respectively. The videos were captured in different weather conditions across 50 different cities in Germany and Switzerland. There are 30 categories in total in the Cityscapes dataset; however, following the previous research, only 19 of them were used.

We tried with different lengths of the frame interval and found that we can achieve the best performance with an interval of 0.12s (more details can be found in Section IV-Ca, which was adopted for all methods. Our model needs multiple video sequences as input for batch normalization. This increases the number of frames handled in one iteration. Because of the limited GPU memory, we resized the original images into 1024×512 for the PSPNet-based model and applied a sliding window with a size of 448×448. This is a commonly used strategy for evaluation [10], [15]. For the ICNet-based model, we maintained the original resolution and adopted a sliding window with a size of 512×1024. We first trained the network without the ConvLSTM module for 40 epochs. After that, the whole network was trained for another 40 epochs. The results of the best-performing model on the validation set were submitted to the Cityscapes test server.

The results are summarized in Table 1. For comparison, we evaluated FCN-8s [23], DeepLab-v3 [3], and DANet [4] as our baselines, all of which were re-implemented and trained with the same configuration. This resulted in different scores than those in the original papers. We applied a smaller input size due to the GPU memory limitation, which causes a decrease in prediction accuracy. We also report the results presented in previous studies that used temporal methods. All the Noisy-LSTM models achieved a better performance than baseline models. The PSPNet-based Noisy-LSTM model got improvements of 1.4% on the validation set and 1.8% on the test set. The ICNet-based Noisy-LSTM model got improvements of 2.5% on the validation set and 2.1% on the test set. The noisy training strategy also improved the performance on both the validation and test sets compared to the models without noisy training. We also show some qualitative results from the validation set in Fig. 5. Each column shows an input frame, its ground truth label, and predictions by some models. All the notable differences are highlighted in orange boxes. This shows that Noisy-LSTM can generate accurate predictions on some challenging objects, for example, the human body in the first column (marked in red), the wall in the second column, and the bus in the third column. Actually, all these objects exist in the previous frames. We can see that Noisy-LSTM can obtain information from these frames and mitigate incorrect segmentation. In this case, the noisy training strategy can help the network obtain these kinds of temporal information efficiently.

B. EndoVis2018 DATASET

We also evaluated Noisy-LSTM and compared it with other methods on the EndoVis2018 dataset [34]. The EndoVis2018 dataset includes 19 sequences, which are split into 15 and 4 sequences for training and testing, respectively. We selected two sequences (sequences #5 and #10) from the training set and used them as the validation set. We resized the image to 520×416 for the PSPNet-based model (the ICNet-based model used the original resolution as input) during training and recovered it in its original resolution for evaluation. Each pixel in the frames was annotated with one of 11 class labels, including organ tissues and surgical instruments.

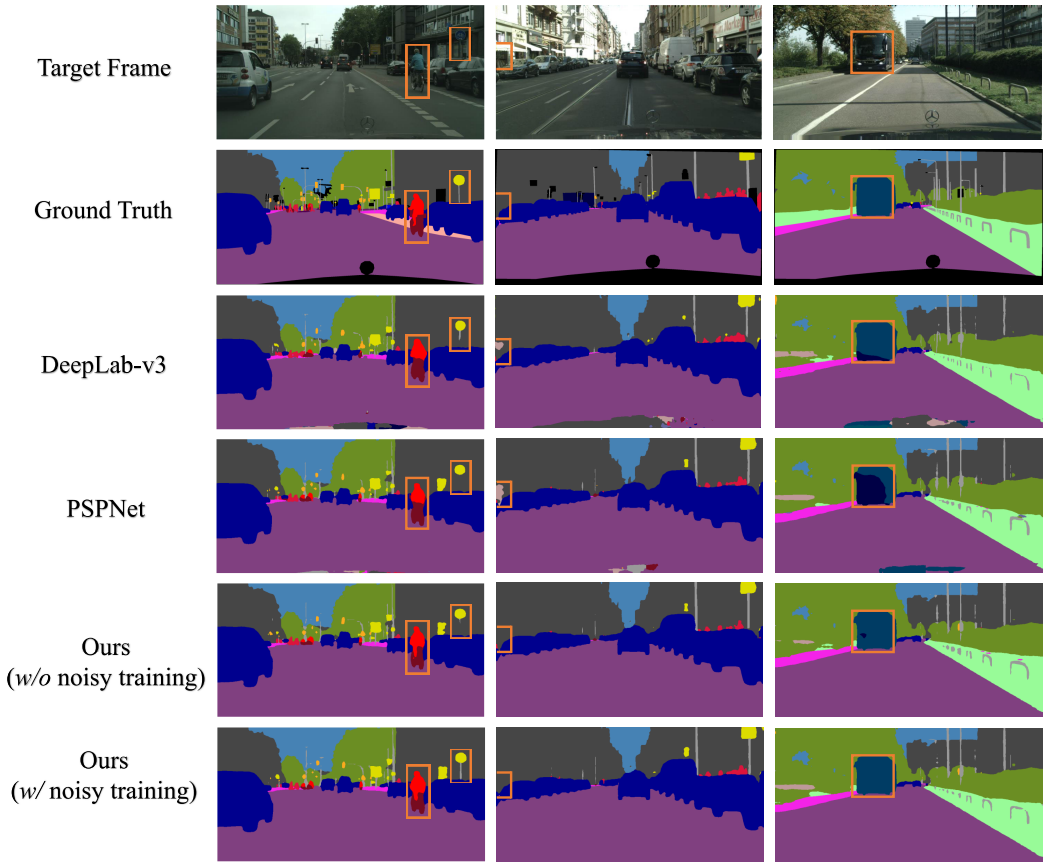

FIGURE 5. Example segmentation results on the Cityscapes dataset using the PSPNet-based models. Our models with the noisy training strategy are able to alleviate incorrect segmentation by favorably obtaining information from the previous frames. Notable differences are marked with orange boxes.

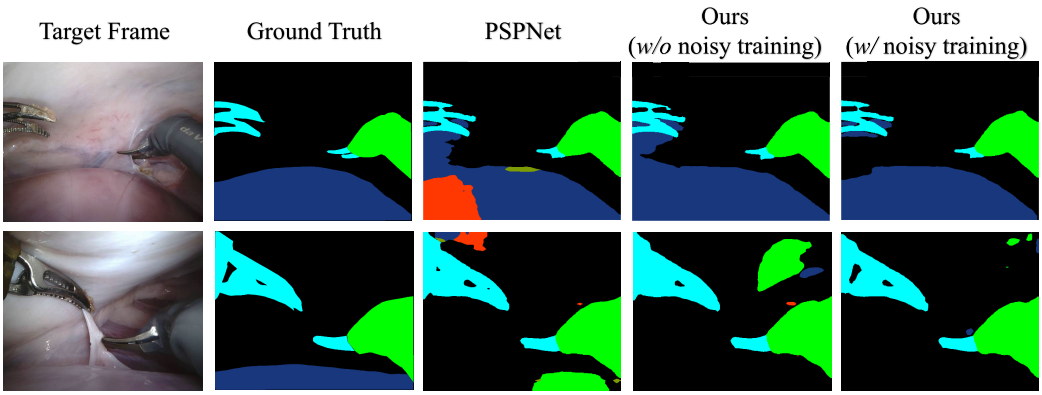

FIGURE 6. Example segmentation for the EndoVis2018 dataset using the PSPNet-based models. The Noisy-LSTM model obtains more accurate segmentation results on the body tissues in the first row, and the surgical instruments in the second row.

Table 1 shows that the Noisy-LSTM model can also outperform other methods on this dataset. Some examples are presented in Fig. 6. Similar to the Cityscapes dataset, on the EndoVis2018 dataset, our Noisy-LSTM provides accurate segmentation results even for small regions.

C. EFFECTS OF THE HYPERPARAMETERS

There are some important hyper-parameters related to the performance. This section gives experimental results to demonstrate the effects of the frame interval, the number of input sequences, and noise types over the Cityscapes dataset's validation set.

1) FRAME INTERVAL

For Cityscapes, each video sequence has 30 frames at 16.7 fps, and the 20th frame is annotated. The Noisy-LSTM model contextualizes the target frame with $T-1$ precedence frames, and context frames can be chosen arbitrarily. In our

implementation, we resampled the context frames from the video sequence, *i.e.*, there are a constant number of frames between s_t and s_{t+1}. We evaluate the cases when the context frames are sampled every 1, 2, and 5 frames, which correspond to frame intervals of 0.12s, 0.18s, and 0.36s, respectively. Table 2 shows the results of the proposed model with or without noisy training using different frame intervals (using unrelated frames as noise). The best result is obtained with an interval of 0.12s and with a noisy training strategy. This shows that a longer interval leads to a decrease in the segmentation performance. Additionally, for all temporal intervals, noisy training methods always show correction capability. This fact proves that the noisy training strategy enhances the temporal awareness of the deep learning models and provides them with an improved ability to extract useful information from previous frames.

TABLE 2. The performance (in mIoU) with different intervals between the input frames, evaluated with PSPNet-based model on the Cityscapes validation set.

	mIoU (%)	
Frame Interval(s)	*w/o* Noisy Training	*w/* Noisy Training
0.06	72.0	72.7
0.12	72.2	**73.0**
0.18	71.6	72.6
0.36	71.2	71.9

2) TYPES AND PROBABILITIES OF NOISE

Noisy training is the key to this work. Thus, we evaluate the effects of different types and intensities of noises with the PSPNet-based model. We demonstrate the experiment results in Table 3. We add three different types of noises: unrelated frames, random tensors, and extreme augmentation (distortion or Gaussian blur). They are described as follows:

TABLE 3. The performance (in mIoU) under different noise types and probability *p*. PSPNet-based models are evaluated on the Cityscapes validation set.

	Noise Type			
Probability p	Unrelated Frames	Random	Distortion	Gaussian Blur
0%	72.2	72.2	72.2	72.2
25%	72.4	72.5	71.6	71.4
50%	**73.0**	72.9	71.2	71.7
75%	72.8	72.4	71.0	71.5
100%	72.5	71.0	71.2	71.3

a: UNRELATED FRAME

This includes images in a totally different domain, extracted from another dataset. For Cityscapes street view images, we utilize medical images from EndoVis2018 as noise frames. When adding noises, a context frame is replaced by an unrelated frame.

b: RANDOM TENSOR

A tensor is initialized with Gaussian noises in the same shape as the input frame. When adding noises, a context frame is replaced by the random tensor.

c: EXTREME AUGMENTATION

Extreme augmentation is a strong image modification that can spoil the original image structure. We adopt distortion (piece-wise affine transformation) and Gaussian blur in this paper. When adding noises, instead of replacing a context frame, we apply this extreme augmentation to it.

The noise frame is randomly selected from the context frames without replacement with probability p (mentioned in Section III-C). In this experiment, we used 25%, 50%, 75%, and 100% for p. 100% means two of the previous context frames are processed with a noisy strategy (we set the maximum number of frames to be processed as $T/2$, and T is set as 4). The result shows that both unrelated frames and random initialization can improve the prediction accuracy of the model, while extreme augmentation did not work well. When the unrelated frame is used as noise, $p = 50\%$ gives the best performance, although the difference is marginal. We can see the same tendency for random tensors, while $p = 100\%$ significantly degrades the performance.

3) NUMBER OF INPUT SEQUENCES IN A BATCH

This parameter is critical for BN and thus can affect the performance of the method. We evaluate the effect of the number of sequences on both PSPNet-based and ICNet-based Noisy-LSTM models. We train the models with different numbers of sequences on single or multiple GPUs, and the results are shown in Table 4. We can see that for the PSPNet-based Noisy-LSTM, a larger batch size leads to a better prediction performance. We also train our model with a batch size larger than 8. However, the performance is not noticeably improved while bringing more training costs. When the batch is set to 1, a great performance drop occurs. For the ICNet-based Noisy-LSTM, improving the batch size can slightly improve the performance. The experimental results prove the necessity of BN for training.

TABLE 4. The performance (in mIoU) for different batch sizes with one or two GPUs over the Cityscapes validation set.

# GPUs	1				2	
# Sequences	1	2	3	4	4	8
ICNet-based	61.3	61.9	61.7	62.5	61.8	62.0
PSPNet-based	68.4	71.9	72.4	73.0	72.6	73.3

D. ROBUSTNESS TO NOISES

For video tasks, some disturbances (blurring, motion distortion, *etc.*) may cause inconsistency in temporal features. In this case, the prediction of the target frame can be affected, and the performance of the model may decrease. Our noisy training strategy can mitigate this problem and generate accurate segmentation. In Table 5, we show the robustness of our models with our training strategy for both the

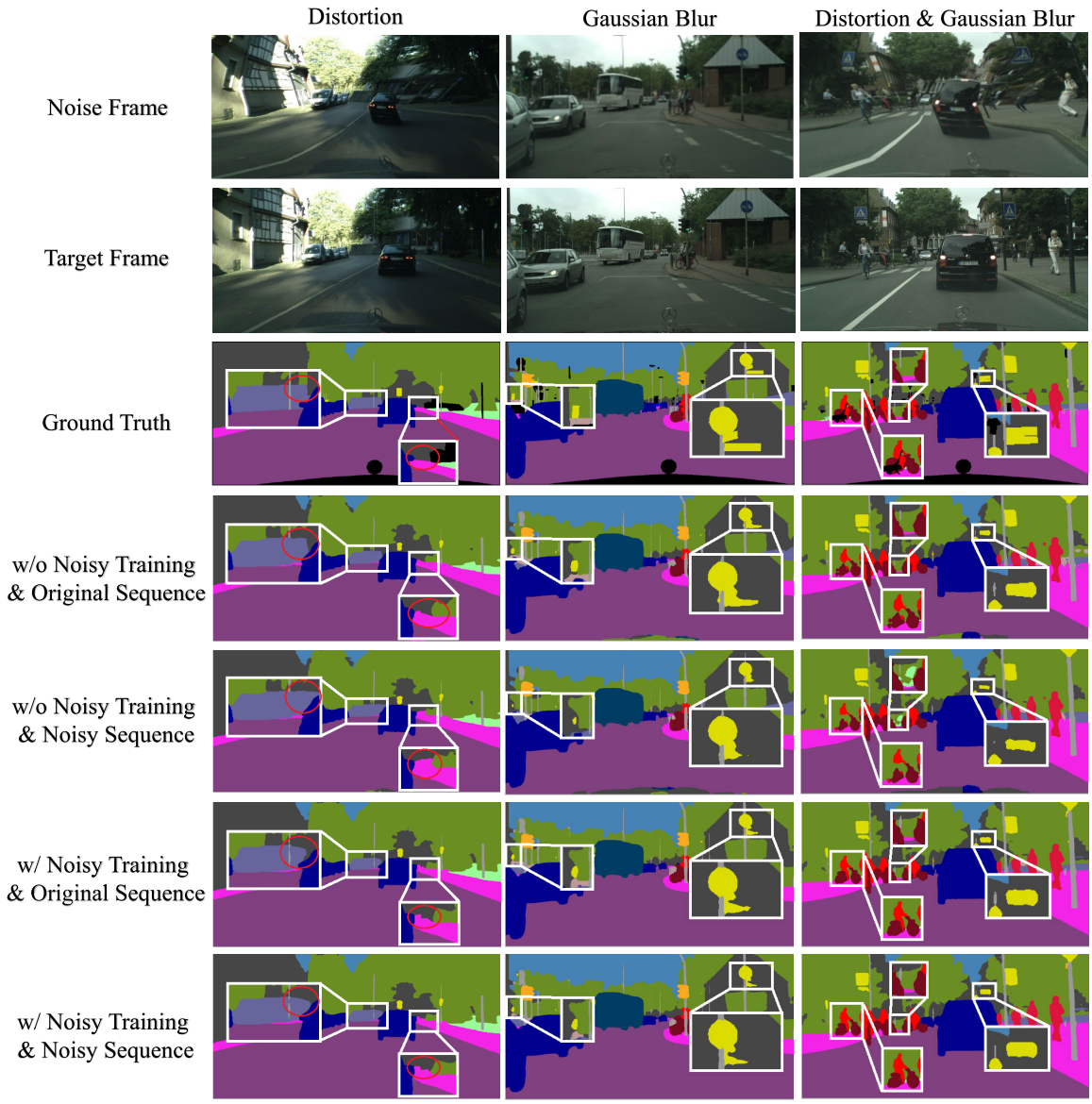

FIGURE 7. Visualization of predicted labels for original input frames (the first row) and noisy frames (the second row) when trained without (the fourth and fifth columns) and with (the sixth and seventh columns) our noisy training strategy. Piece-wise affine transformation and/or Gaussian blur is applied. Some parts of frames (marked as are magnified.

ICNet-based and PSPNet-based models on the validation set of the Cityscapes dataset. We manually introduce such disturbances to the context frames in the validation set. We apply two kinds of noises (Gaussian blur and distortion) and add them to the first and third frames of the input sequence. All the models trained with the noisy training strategy adopted unrelated frames as noises.

When the validation sequences are not altered, the models with our strategy give better results than normal training. When the sequences are altered, the model trained normally experiences a drop in prediction accuracy. However, the prediction accuracy of the model with noisy training almost does not change. These results demonstrate the robustness of our noisy training strategy to disturbances that cannot be well handled by the original ConvLSTM models.

The noisy images are shown in Fig. 7. We find that the models without noisy training are influenced by noisy inputs, while the noisy training strategy can lessen this performance degradation. We also provide some example results in Fig. 7 (some significant differences are highlighted with magnification). All these results are generated by the PSPNet-based model. The target frame is the last frame of the four continuous frames in the input sequence and the third frame is replaced with noises. We used distortion, Gaussian blurring, and both distortion and Gaussian blurring from the first to the third columns, respectively.

TABLE 5. The performance (in mIoU) under different type of noises to evaluate the robustness to noises in target frames.

Model	Noise Type		
	Distortion	Gaussian Blur	No noise
ICNet-based (*w/o* noisy training)	57.5	58.8	61.2
ICNet-based	61.7	62.0	62.5
PSPNet-based (*w/o* noisy training)	70.8	71.1	72.2
PSPNet-based	72.6	72.4	73.0

Without noisy training, the replaced frames cause incorrect predictions. For example, in the red circle in the first column of Fig. 7, the area of *building* (gray) in ground truth frame is partly misclassified as *wall* (slate blue). The end of the sidewalk (fuchsia) also failed. We think these errors are due to the distortion of object regions in the previous frame that are inconsistent with the target frame. In contrast, the models with noisy training were only slightly affected. Similarly, in the second column, Gaussian blur obfuscates the outlines of small objects and even blends them into the background. For noisy target frames, the models without noisy training yielded incomplete predictions of the *signboard* region (yellow), while the models with noisy training have better performances.

V. CONCLUSION

In this paper, we propose a model named Noisy-LSTM, which is trainable in an end-to-end manner, for semantic video segmentation. Noisy-LSTM is capable of utilizing the temporal dependencies in video sequences to improve its segmentation performance. It employs a single convolutional LSTM module to encode spatio-temporal features. In addition, we propose the noisy training strategy, which introduces noises during training to avoid excessive reliance on precedence frames; thus, this technique is expected to improve the feature extraction ability of our model. Our experimental results demonstrate that this strategy further improves the performance without extra data annotations or computational costs, achieving state-of-the-art performances on the Cityscapes and EndoVis2018 datasets. Our future plan is to further explore the type of noises and the way to inject noises in model training. We also plan to apply this method to medical surgery video for accurate semantic analysis, which often suffers from low image quality in some frames due to issues like appliance reflection, *etc*.

REFERENCES

[1] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. ICLR*, 2016.

[2] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285–1298, May 2016.

[3] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: http://arxiv.org/abs/1706.05587

[4] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. CVPR*, 2019, pp. 3146–3154.

[5] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. CVPR*, 2017, pp. 2881–2890.

[6] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, vol. 2014, pp. 740–755.

[8] R. Gadde, V. Jampani, and P. V. Gehler, "Semantic video CNNs through representation warping," in *Proc. ICCV*, 2017, pp. 4453–4462.

[9] D. Nilsson and C. Sminchisescu, "Semantic video segmentation by gated recurrent flow propagation," in *Proc. CVPR*, 2018, pp. 6819–6828.

[10] X. Jin, H. Xiao, X. Shen, J. Yang, Z. Lin, Y. Chen, Z. Jie, J. Feng, and S. Yan, "Predicting scene parsing and motion dynamics in the future," in *Proc. NeurIPS*, 2017, pp. 6915–6924.

[11] C. Xu, C. Xiong, and J. J. Corso, "Streaming hierarchical video segmentation," in *Proc. ECCV*, 2012, pp. 626–639.

[12] A. Kundu, V. Vineet, and V. Koltun, "Feature space optimization for semantic video segmentation," in *Proc. CVPR*, 2016, pp. 3168–3175.

[13] S. Valipour, M. Siam, M. Jagersand, and N. Ray, "Recurrent fully convolutional networks for video segmentation," in *Proc. WACV*, 2017, pp. 29–36.

[14] E. Emre Yurdakul and Y. Yemez, "Semantic segmentation of RGBD videos with recurrent fully convolutional neural networks," in *Proc. ICCV*, 2017, pp. 367–374.

[15] S. Nabavi, M. Rochan, and Y. Wang, "Future semantic segmentation with convolutional LSTM," in *Proc. BMVC*, 2018, p. 137.

[16] A. Pfeuffer, K. Schulz, and K. Dietmayer, "Semantic segmentation of video sequences with convolutional LSTMs," in *Proc. Intell. Vehicles Symp.*, 2019, pp. 1441–1447.

[17] A. Labach, H. Salehinejad, and S. Valaee, "Survey of dropout methods for deep neural networks," 2019, *arXiv:1904.13310*. [Online]. Available: http://arxiv.org/abs/1904.13310

[18] T. S. Nazaré, G. B. P. da Costa, W. A. Contato, and M. Ponti, "Deep convolutional neural networks and noisy images," in *Iberoamerican Congress on Pattern Recognition*. Springer, 2017, pp. 416–424.

[19] Q. Xie, E. Hovy, M.-T. Luong, and Q. V. Le, "Self-training with noisy student improves ImageNet classification," in *Proc. CVPR*, 2020.

[20] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. NeurIPS*, 2015, pp. 802–810.

[21] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proc. ECCV*, 2018, pp. 405–420.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, 2015, pp. 3431–3440.

[24] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proc. CVPR*, 2016, pp. 4340–4349.

[25] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, *arXiv:1207.0580*. [Online]. Available: http://arxiv.org/abs/1207.0580

[26] H. Noh, T. You, J. Mun, and B. Han, "Regularizing deep neural networks by noise: Its interpretation and optimization," in *Proc. NeurIPS*, 2017, pp. 5109–5118.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[28] S. Singh and S. Krishnan, "Filter response normalization layer: Eliminating batch dependence in the training of deep neural networks," in *Proc. CVPR*, 2020, pp. 11237–11246.

[29] A. Graves, "Generating sequences with recurrent neural networks," 2013, *arXiv:1308.0850*. [Online]. Available: http://arxiv.org/abs/1308.0850

[30] C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116, 1995.

[31] J. Xie, B. Shuai, J.-F. Hu, J. Lin, and W.-S. Zheng, "Improving fast segmentation with teacher-student learning," in *Proc. BMVC*, 2018.

[32] F. G. Zanjani, M. van Gerven, and P. de With, "Improving semantic video segmentation by dynamic scene integration," in *Proc. Netherlands Conf. Comput. Vis. (NCCV)*, 2016.

[33] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes dataset for semantic urban scene understanding," in *Proc. CVPR*, 2016, pp. 3213–3223.

[34] M. Allan *et al.*, "2018 robotic scene segmentation challenge," 2020, *arXiv:2001.11190*. [Online]. Available: http://arxiv.org/abs/2001.11190

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.
[36] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," Facebook, Menlo Park, CA, USA, 2017.

BOWEN WANG was born in China. He received the B.CS. degree in computer science from Anhui University, China, and the M.M. degree in medical information research with Osaka University, where he is currently pursuing the Ph.D. degree with the Institute for Datability Science. His research interests include computer vision and medical AI research. He has received the Best Paper Award from APAMI 2020.

LIANGZHI LI received the B.Sc. and M.Eng. degrees in computer science from the South China University of Technology (SCUT), China, in 2012 and 2016, respectively, and the Ph.D. degree in engineering from the Muroran Institute of Technology, Japan, in 2019. He is currently a Researcher with the Institute for Datability Science, Osaka University, Japan. His main research interests include computer vision, deep learning, and medical imaging. He has received the Best Paper Award from the FCST 2017 and IEEE Sapporo Section (2018).

YUTA NAKASHIMA (Member, IEEE) received the B.E. and M.E. degrees in communication engineering and the Ph.D. degree in engineering from Osaka University, Japan, in 2006, 2008, and 2012, respectively. From 2012 to 2016, he was an Assistant Professor with the Nara Institute of Science and Technology. He was a Visiting Scholar with The University of North Carolina at Charlotte, in 2012, and Carnegie Mellon University, from 2015 to 2016. He is currently an Associate Professor with the Institute for Datability Science, Osaka University. His research interests include computer vision and machine learning, and their applications. His main research interest includes video content analysis using machine learning approaches. He is a member of ACM, IEICE, and IPSJ.

RYO KAWASAKI is currently a Clinician-Scientist in ophthalmology with the Osaka University Hospital. He is jointly appointed as an Adjunct Professor of medical data science in ophthalmology with Southern Denmark University. He has a strong passion aiming to achieve the ultimate goal of blindness prevention using epidemiology, data science, behavioral science, and AI. He has published more than 200 peer-reviewed articles. He serves as an Editorial Board Member for *Ophthalmic Epidemiology* and *Japanese Journal of Ophthalmology*.

HAJIME NAGAHARA (Member, IEEE) received the Ph.D. degree in system engineering from Osaka University, Suita, Japan, in 2001. From 2001 to 2003, he was a Research Associate with the Japan Society for the Promotion of Science. From 2003 to 2010, he was an Assistant Professor with the Graduate School of Engineering Science, Osaka University. In 2005, he was a Visiting Associate Professor with CREA, University of Picardie Jules Verns. From 2010 to 2017, he was an Associate Professor with the Faculty of Information Science and Electrical Engineering, Kyushu University. He was a Visiting Researcher with Columbia University, from 2007 to 2008 and from 2016 to 2017. Since 2017, he has been a Professor with the Institute for Datability Science, Osaka University. His research interests include computational photography and computer vision. He was a recipient of the ACM VRST2003 Honorable Mention Award in 2003, the IPSJ Nagao Special Researcher Award in 2012, the ICCP2016 Best Paper Runners-up, and the SSII Takagi Award in 2016.

YASUSHI YAGI (Senior Member, IEEE) received the Ph.D. degree from Osaka University, in 1991. In 1985, he joined the Product Development Laboratory, Mitsubishi Electric Corporation, where he worked on robotics and inspections. He was with Osaka University as a Research Associate in 1990, a Lecturer in 1993, an Associate Professor in 1996, and a Professor in 2003, where he was also the Director of the Institute of Scientific and Industrial Research, from 2012 to 2015, and the Executive Vice President, from 2015 to 2019. He is currently a Professor with the Institute of Scientific and Industrial Research. His research interests include computer vision, medical engineering, and robotics. He is a Fellow of IPSJ and a member of IEICE and RSJ. He was awarded the ACM VRST2003 Honorable Mention Award, the IEEE ROBIO2006 Finalist of T. J. Tan Best Paper in Robotics, the IEEE ICRA'2008 Finalist for Best Vision Paper, the MIRU'2008 Nagao Award, and the PSIVT'2010 Best Paper Award. He has served as the Chair for International conferences, such as FG'1998 (a Financial Chair), OM-INVIS'2003 (an Organizing Chair), ROBIO'2006 (the Program Co-Chair), ACCV'2007 (the Program Chair), PSVIT'2009 (a Financial Chair), ICRA'2009 (a Technical Visit Chair), ACCV'2009 (the General Chair), ACPR'2011 (the Program Co-Chair), and ACPR'2013 (the General Chair). From 2007 to 2011, he has also served as an Editor for the IEEE ICRA Conference Editorial Board. He is an Editorial Member of *International Journal of Computer Vision* and the Editor in-Chief of the *IPSJ Transactions on Computer Vision and Applications*.