Hierarchical Temporal Attention Network for Thyroid Nodule Recognition using Dynamic CEUS Imaging – *Supplementary Material*

Peng Wan, Fang Chen, Chunrui Liu, Wentao Kong, Daoqiang Zhang

In the main text, we develop a *hierarchical temporal attention network* (HiTAN) for thyroid nodule recognition using dynamic CEUS imaging. In this Supplementary file, we include the implementation details of different types of competing methods, network structures of ablation experiments, and comparison of different nodule detection methods.

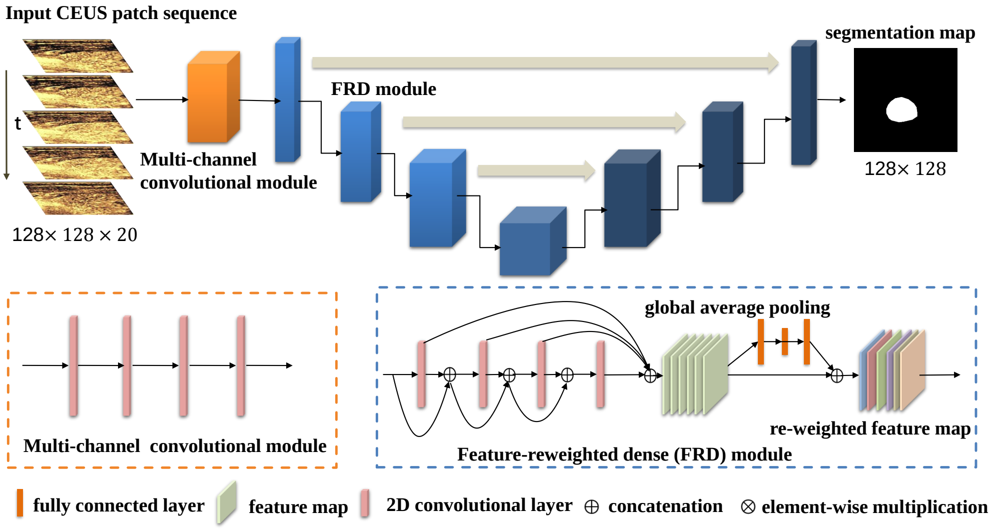


Fig. S1. The flowchart of CEUS-Net architecture used for nodule detection. The multi-channel convolution module is composed of multiple 2D convolution layers and the number of channels is 6, 16, 32, and 64, respectively. As can be seen, each layer in the feature-reweighted dense (FRD) module fully reuses information from all preceding layers. All the generated feature maps accumulate at the end of every dense unit, creating a very rich set of representations.

A. Implementations Details

1) CEUS-Net: We used the default architecture of CEUS-Net[1] in our experiments. This architecture includes a multi-channel convolution module and a U-net module infused with feature reweighted dense blocks. Detailed network structure is presented in Fig S1. To boost the sample diversity and reduce GPU memory load, we adopt the patch-based training strategy with a small batch size of 4. In the training stage, patches of $128 \times 128 \times 20$ were randomly cropped from the input CEUS sequence; In the testing stage, patches of $256 \times 256 \times 20$ were sequentially cropped with 50% overlap, and the corresponding outputs were assembled into a probability map by average, which is converted into a binary mask using an empirical value $\tau = 0.5$. As for parameter update, we use the cross-entropy loss and RMSProp Optimizer with an initial learning rate of 0.0001 and decay of 0.995. The maximal training epochs is set to 50 and the training process would be terminated if the improved accuracy (in terms of mean intersection over union, mIOU) is below 0.001 on validation set. Additionally, we adopt the standard pipeline of data augmentation, including random flipping, rotating, and scaling, etc.

This work was supported by the National Natural Science Foundation of China (Nos. 61876082, 61732006, 61861130366, 61901214, U20A20389, 81671701), the National Key R&D Program of China (Grant Nos. 2018YFC2001600, 2018YFC2001602, 2018ZX10201002), the Royal Society-Academy of Medical Sciences Newton Advanced Fellowship (Nos. NAF\R1\180371), the China Postdoctoral Science Foundation funded project (Nos. 2020M671484), the Nanjing Medical Science and technique Development Foundation (Nos. YKK19054).

P. Wan, F. Chen, D. Zhang are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Nanjing 211106, China.

C. Liu, W. Kong are with the Department of Ultrasound, Affiliated Drum Tower Hospital, Medical School of Nanjing University, Nanjing 210008, China.

Corresponding author: W. Kong (e-mail: breezewen@163.com) and D. Zhang (e-mail: dqzhang@nuaa.edu.cn).

2) Multiple Kernel Learning (MKL): The implementation of MKL algorithm is based on open-source python library MKLpy[1], and the API EasyMKL [2] is used to learn the optimal combination coefficient of base kernels. Notably, we treat the radial basis functions (i.e., $e^{-\gamma_k\|x-y\|_2^2}$) computed in three different perfusion phases as to-be-combined base kernels, thus fusing dynamic enhancement features in CEUS videos. As for optimal parameter selection of γ_k and λ that balances the importance of inter-class discrepancy and inter-centroid distance, we perform a coarse grid search within $\{0.1, 0.2, \cdots, 5\}$ and $\{0.05, 0.1, \cdots, 1\}$ via cross-validation.

3) Deep Canonical Correlation Analysis (DCCA): The implementation of DCCA algorithm is based on the released code run on Pytorch. For each pair of views, 66-dimensional texture features are projected into a common latent space by a multilayer perceptron (MLP) consisting of four fully-connected layers. The output dimension of four layers is 52, 48, 32, and 24, respectively. To maximize the correlation of two views, network weights are updated using a RMSprop optimizer with a learning rate of 0.001 and a large batch size of 100. The number of maximal training epochs is set to 50 and the training process is terminated until achieving the highest correlation on validation set. In this way, we could obtain six-view features by aligning each pair of three phases, and then these features are fed into a multi-kernel SVM classifier (i.e., EasyMKL) to boost the diagnostic accuracy.

4) Deep Learning-based Methods: All deep learning-based video recognition models are implemented on the Pytorch framework, and run on a single GPU (i.e., NVIDIA TITAN RTX 24GB). For a fair comparison, these comparative methods are trained on the same preprocessed data, i.e., the cropped CEUS sequence with the length $T = 20$, and evaluated using the same 5-fold cross-validation protocol. Alternately, one fold is chosen as the testing set while the remaining folds are used as the training and validation set in the ratio of 4:1. To avoid model overfitting, initial weights of 2D(3D) convolution filters of each competing method are learned via auto-encoders[10], where the original backbone is treated as the encoder and a symmetric decoder is appended to reconstruct CEUS frames or sequences under the mean squared (MSE) loss. Besides, the introduced random temporal sampling could considerably augment the diversity of CEUS sequences.

To clarify the training step of each competing method, we present the related details in Table S1, including the number of maximal training epochs, batch size, optimizer, learning rate, implementation details, and reference codes.

B. Network Structures of Ablation Study

To evaluate the effectiveness of the hierarchical lesion recognition module as well as the Local-to-Global temporal aggregation (LGTA) operator used in this module, we design another four variants of our HiTAN method as baselines.

- The first variant Non-local temporal aggregation network (NL-TAN) only preserves the non-local temporal aggregation (NLTA) operator for dynamic enhancement patterns fusion. As shown in Fig. S3, pathological prediction is made at one step without taking label dependency of different diagnostic stages into consideration;
- The other three variants preserve GRUs-based hierarchical lesion recognition mechanism but exploit other temporal fusion manners (i.e., the local-to-local, global-to-local, and global-to-global temporal aggregation) at the characterization level. Therefore, the corresponding variants are named as LL-HCN, GL-HCN, and GG-HCN, respectively. For simplicity, Fig. S4 only illustrates the framework of GG-HCN as an example, and the other two implementations only replace the characterization-level GGTA operator with fusion operators shown in Fig. S5.

C. Comparison of different nodule detection methods

In this group of experiments, we test four different detection methods to evaluate the impact of nodule detection on the eventual recognition performance, 1) We add another two classical detection networks, Faster-RCNN[11] and YoLov3[12] as comparison; 2) We adopt the pre-trained CEUS-Net directly without parameters fine-tuning; 3) We fine-tune the CEUS-Net by sampling 40% and 60% training set as network inputs; 4) We manually correct the detection results produced by fine-tuned CEUS-Net if the mean intersection over union (mIOU) compared with the ground truth is below 0.5. We adopt a 5-fold cross-validation strategy and report the average results over five folds.

Implementation Details: As for Faster-RCNN and YoLov3, both models are implemented on the latest open-source platform MMDetection[2][13] and trained on a single GPU (NVIDIA TITAN RTX, 24GB). Following the same pre-processing steps as HiTAN, the dimension of input contrast sequence is $360 \times 400 \times 20$. The ground-truth bounding box B_g is generated on the pixel-wise segmentation mask. For Faster-RCNN and YoLov3, we use Resnet-34 and Darknet-53 as backbones respectively, and the number of channels is halved to make networks lightweight. As in CEUS-Net, we modify the number of input channel to 20 and the number of classes to 1 for both models. For Faster-RCNN, we adopt the *1x learning schedule* provided by MMDetection. Based on that, we set the learning rate to 2.5×10^{-3} and the number of maximal training epochs to 50. For YoLov3, we modify the learning rate to $2. \times 10^{-3}$ and the number of maximal training epochs to 100. All convolution layers in backbones are updated with a small batch size of 2. The resolutions of rescaling operation in training and testing

[1] https://mklpy.readthedocs.io/en/latest/

[2] https://github.com/open-mmlab/mmdetection

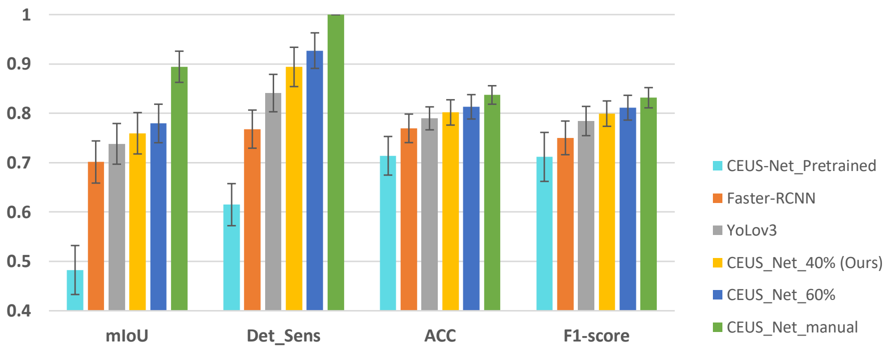


Fig. S2. Comparison of different methods on nodule detection and its impact on classification performance.

pipeline are set to $[(380, 380), (420, 420)]$ and $[(420, 420)]$ by keeping aspect ratio. In order to determine appropriate sizes and aspect ratios of candidate boxes for YoLov3, we performed a K-means clustering on the bounding boxes in training set with the distance of IOU. The chosen 9 clusters are (15,22), (26,35), (32,37), (47,45), (55,62), (67,51), (69,74), (121,105), and (166,142), respectively. As for Faster-RCNN, we set the base sizes and aspect ratios to $[30, 60, 110]$ and $[0.7, 0.9, 1.2]$, respectively.

Evaluation Metrics: For detection models, we take the predicted ROI with the highest category confidence as the final detection result. For CEUS-Net, the predicted ROI is determined by generating a bounding box that encloses the foreground of segmentation mask. To measure the detection performance as well as its impact on nodule classification, we adopt the intersection over union (IOU) between the predicted bounding box B_p and the ground-truth B_g, $\mathrm{IOU} = \frac{B_p \cap B_g}{B_p \cup B_g}$. The predicted ROI with $\mathrm{IOU} \geq 0.5$ is regarded to be true positive (TP), otherwise false positive (FP). The detection sensitivity is calculated as, $\mathrm{Sen} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$, where FN is the number of false negatives, $TP \times FN$ equals the number of to-be-detected ground-truth ROIs. As for classification performance, we use the accuracy (ACC) and F1-score as metrics, $\mathrm{F1} = 2 \times (\mathrm{P} \times \mathrm{R})/(\mathrm{P} + \mathrm{R})$, where P and R are precision and recall scores respectively.

From Fig. S2, several observations could be summarized. 1) Compared with the state-of-the-art detection models (Faster-RCNN and YoLov3), CEUS-Net fine-tuned with 40% training samples (108 instances) has achieved superior detection performance with mIoU of 0.759 and Det_Sens of 0.894, which confirms the effectiveness of CEUS-Net from our previous work in the fundamental task of nodule detection; One possible reason is that segmentation mask could provide more refined boundary information of nodules than a rectangular bounding box; 2) As expected, superior nodule detection performances consistently lead to better pathological classification results, which validates the importance of reliable nodule detection to subsequent enhancement characteristics extraction and fusion; For example, when we manually correct those false detections to the highest mIoU with 0.894, a further improvement of mean classification accuracy can be observed from 0.802 to 0.837. Nevertheless, when we apply a pre-trained CEUS-Net without parameter fine-tuning, the resulting classification accuracy experiences a dramatic drop to 0.714 along with the lowest mIoU of 0.483 and detection sensitivity of 0.615; 3) Classification performance gains brought by the improved detection results gradually decrease when detection accuracy reaches a certain level, such as $\mathrm{Det\_Sense} \geq 0.8$. For example, the increased detection performances of our CEUS-Net method in comparison to YoLov3 are 2.8% and 6.3% in terms of mIoU and Det_Sens, respectively, while the leading classification gains are merely 1.5% and 1.9% in terms of accuracy and F1-score, respectively. One possible reason is that, in our implementation, bounding boxes are enlarged by a fixed factor $\gamma = 1.2$ to incorporate part of surrounding tissues as diagnostic basis. In this way, those misaligned detection results might be partially corrected to include informative regions, which makes the requirement of accurate nodule localization less strict; 4) Although a larger number of annotated samples could further improve the nodule detection performance of fine-tuned CEUS-Net, the leading classification performances improvement of HiTAN method is relatively limited from the sampling rate from 40% to 60% (around 0.01 for ACC and F1-score). Therefore, we could state that 40% should be an appropriate sampling rate which largely includes varying nodule sizes, shapes, and locations, as well as accounts for the pixel-wise annotation cost.

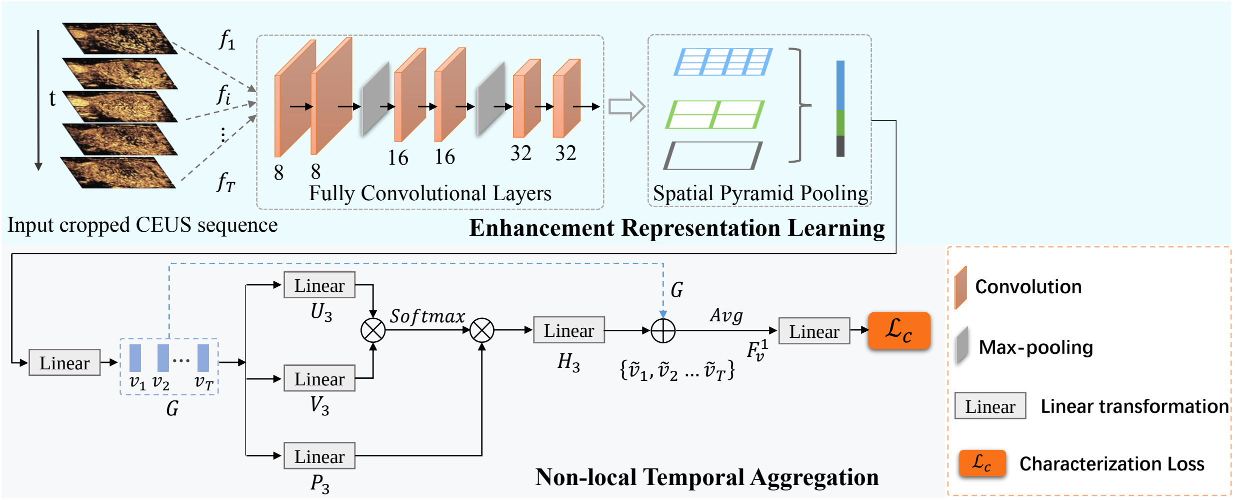


Fig. S3. Illustration of the framework of Non-local Temporal Aggregation Network (NL-TAN). Two major components are included: 1) Enhancement representation learning module, and 2) Non-local temporal aggregation operator. In comparison to the original HiTAN, we remove the fundamental hierarchical lesion recognition mechanism in NL-TAN. For pathological recognition, we preserve the non-local temporal aggregation (NLTA) operator to capture the long-term temporal dependency in sequential enhancement descriptors.

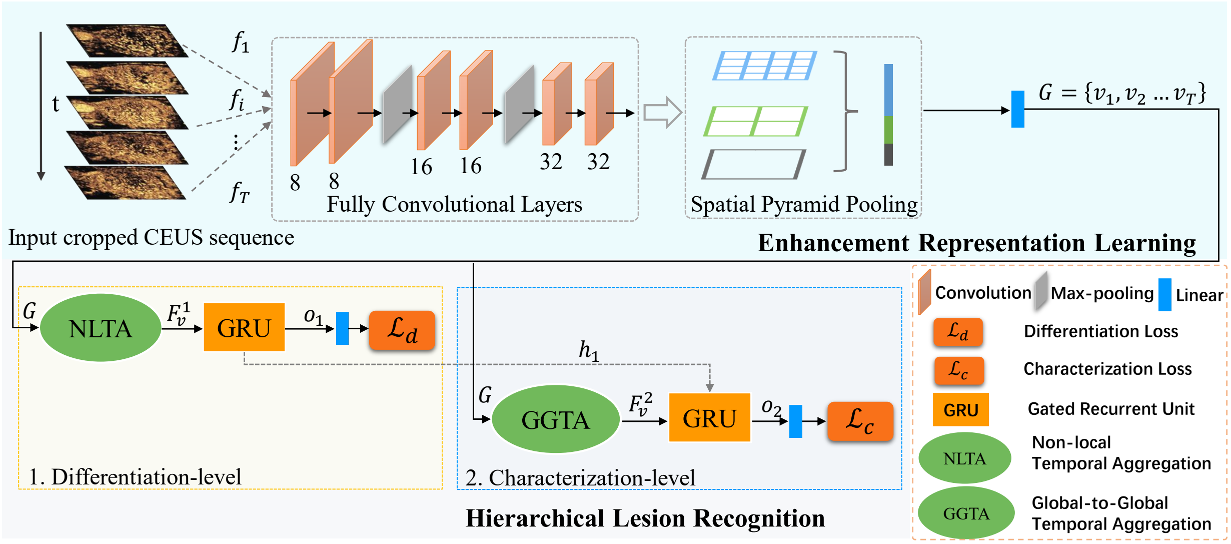
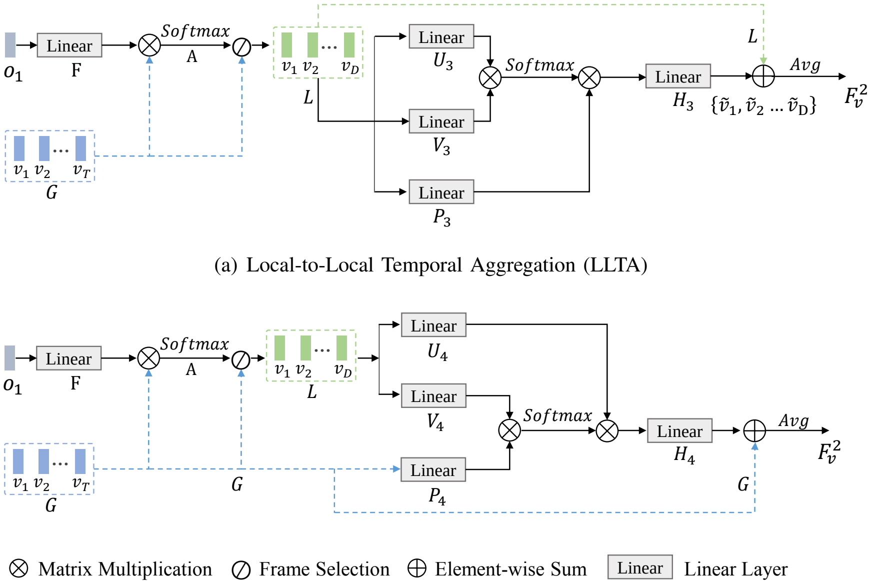


Fig. S4. Illustration of the framework of Hierarchical Convolution Network with Global-to-Global Temporal Aggregation at the characterization level, named as GG-HCN. The major difference from HiTAN is to substitute the original LGTA operator with the GGTA operator at the characterization level, and o_1 is no longer treated as the source of temporal guidance signal.

TABLE S1
DETAILED TRAINING SETTING AS WELL AS REFERENCE IMPLEMENTATIONS OF DEEP LEARNING-BASED COMPETING METHODS.

Method	Epochs	Batch Size	Optimizer	Learning Rate	Implementation Details	Link
C3D[3]	30	10	SGD	0.003, divided by 10 after every 10 epochs	The backbone of C3D consists of four $3 \times 3 \times 3$ convolution(Conv) layers with unit stride, and the number of channels for Conv1 to Conv4 is halved to 32, 64, 128, and 256. Each Conv layer is followed by a $2 \times 2 \times 2$ max-pooling layer with stride of 2. The dropout layer with the rate of 0.2 is appended to the fully-connected layers. Considering the varying size of cropped CEUS sequences, we adopt the same spatial pyramid pooling layer to aggregate three-dimensional feature tensors at the top.	https://github.com/DavideA/c3d-pytorch
2Plus1D[4]	40	8	SGD	0.001, divided by 10 after every 4 epochs	In our implementation, R2Plus1D contains one 7×7 Conv layer with stride of 2 and four $3 \times 3 \times 3$ Conv layers with stride of 2. The number of channels for Conv1 to Conv4 is 32, 64, 128, and 256, respectively.	https://github.com/irhum/R2Plus1D-PyTorch
TCN[5]	40	10	Adam	0.001, divided by 10 after 12 epochs	In our implementation of TCN, dynamic CEUS sequence is fed into the same enhancement representation learning module as in HiTAN, producing a sequence of fixed-length enhancement descriptors. Then, frame-level features are aggregated by three stacked temporal convolution residual blocks, with 1×2 kernel size and unit stride. The dilation rate of three blocks is set to $[1, 2, 4]$, and the number of channel is reduced to the half of input dimension. The drop out rate is set to 0.2. Finally, temporal average pooling is used to produce the video-level embedding.	https://github.com/locuslab/TCN
CNN-LSTM[6]	40	8	Adam	0.005, divided by 10 every after 10 epochs	Similar to TCN, dynamic CEUS sequence is transformed into sequential 128-dimensional enhancement vectors via the backbone of HiTAN. The LSTM module used for temporal aggregation has two stacked recurrent layers. The dimension of hidden state is reduced to 64 and the dropout rate is set to 0.1.	https://github.com/doronharitan/human_activity_recognition_LRCN
TSN[7]	50	4	SGD	0.01,divided by 10 every after 10 epochs	Following the study in [7], each snippet comprises a contrast frame randomly sampled from the corresponding segment, RGB difference, and the 2-channel optical flow field calculated by TV-L^1 algorithm[8] and discretized to $[0-255]$. In our implementation, we adopt the ResNet18 as the backbone for perfusion feature extraction. Considering the varying size of cropped CEUS sequences, we resize images to a common size 224 pixels by keeping aspect ratio. The parameter K for temporal segment division is set to 4, and confidence scores from different snippets are averaged to obtain the final prediction.	https://github.com/yjxiong/tsn-pytorch
TSM[9]	50	4	SGD	0.01, divided by 10 after every 10 epochs	To make the network lightweight, we replace the original backbone ResNet50 with ResNet18. In our implementation, we apply the bi-directional temporal shift operation on each spatial-temporal feature tensor, such that temporal receptive field at each time step is 3. The proportion of shifted channels is selected as the default $1/4$. To preserve spatial information, the resulting feature is also combined with the input via a residual connection. For long-term temporal modeling, temporal shift module (TSM) is embedded into the classic temporal segment network (TSN), where the number of temporal segment division is set to 4.	https://github.com/mit-han-lab/temporal-shift-module



(a) Local-to-Local Temporal Aggregation (LLTA)

(b) Global-to-Local Temporal Aggregation (GLTA)

Fig. S5. Fig. S5(a) Local-to-Local Temporal Aggregation (LLTA) operator: it only considers pairwise interactions within the identified contrast subsequence $\mathbf{L}$; Fig. S5(b) Global-to-Local Temporal Aggregation (GLTA) operator: enhancement characteristics from identified keyframes $\mathbf{L}$ are integrated into each enhancement descriptor v_t instead of embedding global features $\mathbf{G}$ into identified key patterns $\mathbf{L}$.

References

[1] P. Wan, F. Chen, X. Zhu, C. Liu, Y. Zhang, W. Kong *et al.*, “CEUS-Net: Lesion segmentation in dynamic contrast-enhanced ultrasound with feature-reweighted attention mechanism,” in *Proceedings of the IEEE International Symposium on Biomedical Imaging, ISBI 2020*, 2020, pp. 1816–1819.

[2] F. Aiolli and M. Donini, “Easymkl: a scalable multiple kernel learning algorithm,” *Neurocomputing*, vol. 169, pp. 215–224, 2015.

[3] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE Conference on International Conference on Computer Vision, ICCV 2015*, 2015, pp. 4489–4497.

[4] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, 2018, pp. 6450–6459.

[5] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks for action segmentation and detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, pp. 1003–1012.

[6] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko *et al.*, “Long-term recurrent convolutional networks for visual recognition and description,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, 2017.

[7] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang *et al.*, “Temporal segment networks for action recognition in videos,” *CoRR*, vol. abs/1705.02953, 2017.

[8] C. Zach, T. Pock, and H. Bischof, “A duality based approach for realtime tv-L^1 optical flow,” in *Pattern Recognition, 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007, Proceedings*, ser. Lecture Notes in Computer Science, F. A. Hamprecht, C. Schnörr, and B. Jähne, Eds., vol. 4713. Springer, 2007, pp. 214–223.

[9] J. Lin, C. Gan, and S. Han, “TSM: temporal shift module for efficient video understanding,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV 2019*, 2019, pp. 7082–7092.

[10] G. B. Cavallari, L. Ribeiro, and M. Ponti, “Unsupervised representation learning using convolutional and stacked auto-encoders: A domain and cross-domain feature space analysis,” in *Proceedings of the Conference on Graphics, Patterns and Images, SIBGRAPI 2018*, 2018, pp. 440–446.

[11] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[12] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *CoRR*, vol. abs/1804.02767, 2018.

[13] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li *et al.*, “Mmdetection: Open mmlab detection toolbox and benchmark,” *CoRR*, vol. abs/1906.07155, 2019.