A Spatiotemporal Excitation Classifier Head for Action Recognition Applications

Dinh Nguyen, Siying Liu, Vicky Sintunata, Yue Wang, Jack Ho,
ZhaoYong Lim, Ryan Lee, Karianto Leman
*Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR)*
1 Fusionopolis Way, #21-01 Connexis South Tower, Singapore 138632, Republic of Singapore
{nguyen_van_dinh;liusy1;Vicky_Sintunata;Wang_Yue;Jack_Ho;Lim_Zhao_Yong;Ryan_Lee;karianto}@i2r.a-star.edu.sg

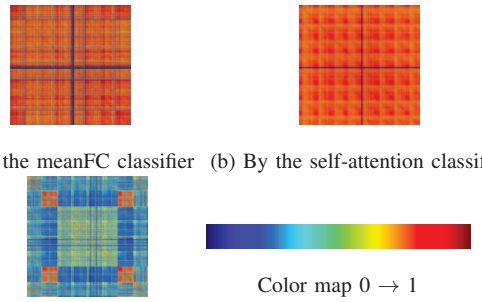
***Abstract*—Transfer learning is a convenient approach to quickly adapt state-of-the-art deep learning models to specific applications with small datasets. Typically, network backbones are fixed, and only the last layer as a classifier is modified to match with a new number of targeted classes. The performance of the models is then limited by model-predefined structures. In this research, we are going to overcome this constraint by studying the effect of the common classifier layer and then proposing an extension classifier module for action recognition applications. By focusing on local spatiotemporal representation of deep features encoded by pre-trained models, we exploit further this local representation in the proposed classifier to enrich deep features representation. In addition, the extension classifier was designed so that it can plug on top of any image or video encoders to perform action recognition. A public dataset TinyVIRAT2 and two private datasets Scratch and AtomicA were adopted for evaluation and the experiments show significant performance improvement caused by the proposed extension classifier.**

***Index Terms*—Spatiotemporal feature, temporal inception, action recognition, extension classifier head**

I. Introduction

By leveraging pre-trained models in public model zoos, we can inherit the generalization capacity of models. However, the drawback is that we have to stick with pre-defined model structures. To overcome this limitation, pre-trained models should be considered as data encoders. Classifier heads that support specific applications can be designed on top of them, capturing essential representation among encoded multi-dimension features to maximize task performances. Flexibility in designing classifier heads also provides opportunities to adopt pre-trained models from different tasks. In developing action recognition in surveillance systems, the proposed action classifier heads were designed to focus on spatiotemporal representation among multidimensional features of videos encoded by state-of-the-art (SOTA) models, which were trained for either image or video classification applications.

The most common classifier head consists of a single fully connected layer following a channel-wise mean function transforming compressed deep features into a single deep feature, a normalization layer standardizing distribution of feature values, and a dropout layer overcoming overfitting. We named this popular classifier head as **meanFC**. The objective of this design is to extract location-invariant features, meaning that encoded deep features at different spatiotemporal locations are expected to be the same after training. However, due to local representation, this goal seems unachievable. Fig 1.a and Fig 1.b are the heat maps of Pearson correlation coefficient scores among deep features encoded by a pre-trained model (VideoVIT [1]) using meanFC and self-attention as a classifier, respectively. In general, almost all features are well correlated to each others (red colour) but there are existing features that do not, indicating in blue and green colours. Therefore, diving deeper into local spatiotemporal representation should benefit entire model performance. Consequently, when we apply the proposed classifier head (STE), deep features of VideoVIT model becomes divergent, indicating by dominance of blue and green colours in the generated heat map Fig 1.c.



(a) By the meanFC classifier (b) By the self-attention classifier

(c) By the STE classifier

Fig. 1: The heat maps of Pearson correlation coefficient score among encoded features extracted by the pre-trained model VideoVIT [1] when it combines with (a) meanFC, (b) self-attention, and (c) our proposed head (STE) as a classifier.

Temporal and spatiotemporal representations depicting relationships among frames in input videos were considered essential representations for action recognition. Many proposed research works captured this representation using convolutional layers, TSN [2], I3D [3], PAN [4], TEA [5], TDN [6], EAN [7], as well as transformer attention layers, UniformerV2 [8], Hiera [9], VideoMAEV2 [1] TimeSformer [10], DirectFormer [11], TubeVit [12]. In TSN, I3D, PAN, TEA, TDN, and EAN, temporal representation extraction layers, usually conv3D, temporal difference, and temporal shift, were inserted in the middle of spatial representation extraction layers. Spatiotem-

This work was supported by the A*STAR Computational Resource Centre through the use of its high performance computing facilities.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00020

poral representation gradually grows from local representation to global representation along model depth. Uniformer model [8] is a conv-transformer hybrid architecture that inserts convolutional layers in the middle of transformer layers to capture the local representation of encoded features which they noticed that this representation is sparse in most of the transformer models, such as VideoMAE2. TimeSformer [10] tried to separate temporal attention and spatial attention modules and provided ablation studies on different combinations of these modules. TubeVit [12] proposed local spatiotemporal attention in tubes. On the other hand, the Hiera model [9] focuses on local and global spatiotemporal representations using pure transformer blocks.

Drawing inspiration from the inception modules in GoogleNet [13], that manage adeptly multi-scale spatial representation, we propose a temporal inception component handling multi-scale temporal representation. Different numbers of consecutive frames are involved in capturing temporal representation. Furthermore, we replace the popular meanFC classifier with a proposed patch_base_classifier, implemented by applying convolution 3D layers to exploit further spatiotemporal representation among encoded deep features. Combining these two proposed components with a spatial extraction component and a multilayer perception component (MLP), we form a spatiotemporal excitation classifier head (STE_head) that is a main contribution in this research. This extension classifier head is designed so that it can be plugged on top of any feature extraction backbones, including spatial feature extractors and spatiotemporal feature extractors, to perform action recognition tasks.

As both VideoMAEV2 [1] and UniformerV2 [8] stands as state-of-the-art models cross various action recognition benchmarks such as somethingV2, kinetic400, and kinetic600, we utilize their architectures, excluding the meanFC classifier head, as video feature extractors in all our experiments. We opt for the base version of VideoMAEV2 since it is the largest version available with pre-trained weight [14]. Different classifier heads are applied on the top of these feature extractors to form different variants. These variants are evaluated on three different action datasets: a public dataset, TinyVIRAT2 [15], and two private datasets, Scratch and AtomicA, which are designed for two practical applications. The variants formed by the proposed STE_head and two video feature extractors always provide the top performance. Excitingly, these variants provide a giant leap in performance improvement, more than 10%, on dataset AtomicA.

II. A SPATIOTEMPORAL EXCITATION CLASSIFIER HEAD

In this section, we describe the architecture of the proposed action classifier head (STE_head) that intends to magnify local spatiotemporal representation presented in deep features encoded from images/videos. This proposed head consists of two main parts: spatiotemporal excitation blocks (STEb) and a patch_base_classifier. A number STEb defines the depth of the STE_head. This STE_head requests deep features input shape of [n, c, h, w]. The value **n** is the number of video frames from input videos. The values **c, h, w** are the numbers of channels, height, and width of deep features, respectively.

A. A STE block

A STE block (**STEb**) includes three main components: a temporal extraction component, a spatial extraction component (SE), and a multilayer perception component (MLP). Each component extracts different representations from encoded features and embeds these extracted representations back into the encoded features.

$$TI(x_0) = x_0 + f_{conv311}(x_0) + f_{conv511}(x_0) + f_{conv711}(x_0) \quad (1)$$

$$x_1^{'} = SE(x_0 + TI(x_0)) \quad (2)$$

$$x_1 = x_1^{'} + MLP(x_1^{'}) \quad (3)$$

Where x_0 and x_1 are an input and an output of a STEb.

The temporal extraction component is a temporal inception module (TI). The goal is to enrich video-encoded features x_0 with multiple scales of its local temporal representation. We adopted 3D convolution layers to design this module. The convolution kernels' temporal dimensions vary from three to ***t***, targeting encoded features of three to t consecutive frames. The kernels' spatial dimensions maintain as one. The value t might need to be adjusted for optimal recognition performance. In this work, the proposed TI module is configured with three values of t as 3,5,7, forming three conv3d layers with kernel sizes of 3x1x1, 5x1x1, and 7x1x1, shown in equation (1).

The spatial extraction component follows the temporal extraction component to capture further local spatial representation in encoded features. In this work, we applied 2D convolution with a kernel size 3x3. We then enrich the encoded features by representation from its higher dimension space, extracted by the MLP component, which is very common in transformer blocks.

B. A patch_base_classifier

To avoid the aforementioned drawback of the common classifier meanFC, we proposed a **patch_base_classifier**. Encoded features are finally divided into N patches (cubes); each patch contains its local spatiotemporal representation. Patches are then passed through a single 3D convolution layer with a kernel size equal to the patch size to perform action classification. The output of this classifier is in the shape of (N,num_class). A function channel-wise mean is then applied to the output to form the final classification. It is worth noting that the number of output channels equals the number of classes.

$$y = \mathbb{E}(Conv3d(x)) \qquad x \in \mathbb{R}^{N,d} \Rightarrow y \in \mathbb{R}^{n\_cls} \quad (4)$$

III. EXPERIMENTS

A. Datasets

To evaluate the effectiveness of the proposed classifier head, we used three datasets: TinyVIRAT2 dataset [15], Scratch dataset, and AtomicA dataset.

The TinyVIRAT2 dataset was proposed to optimize action recognition models for security surveillance applications. The dataset was created by cropping out human activities regions in dataset VIRAT. Cropped clips have a wide resolution range from 10x10 to 128x128 pixels, averaging 70x70 pixels. This dataset is multi-label data, meaning that there are multiple targeted actions per clip. The dataset focuses on 26 daily activities and has 20258 samples for training and 7425 samples for testing. In this study, we only used the training samples as there were no public labels for the testing samples. Training samples are divided into a training set containing 16950 samples and a validation set containing 3308 samples.

The Scratch dataset was proposed to develop scratch detection models for healthcare services. Live cameras recorded videos at 30fps, mounted at 2-meter height. Hands areas were localized by applying the mediapipe library [16] and cropped to form sample clips, sorted into two classes: scratch and no-scratch. Two separate groups of participants were managed to provide a training set and testing set, composed of 2064 and 451 samples, respectively. Cropped clips have a resolution range from 10x10 pixels to 226x226 pixels with an average of 80x80 pixels.

The AtomicA dataset was proposed to develop action recognition models for analyzing customers' behaviour while interacting with retail store products. The dataset focuses on six atomic actions: Put-Items, Fetch-Items, Touch-Items, Try-Items, Carry-Items, and Others. To conduct this dataset, we first recorded videos fed by a surveillance system setup at retail clothing stores at a frame rate of 2fps. Human detection was then applied to localize humans with ROIs, which were used to crop videos and provide sample clips. Because Put-Items and Fetch-Items are fleeting actions, appearing only in two to five consecutive frames, all sample clips were configured to be short at this length. In the validation set, each action has 100 samples. In the training set, respectively, Put-Items, Fetch-Items, Touch-Items, Try-Items, Carry-Items, and Others have 420, 587, 1367, 356, 5569, and 2563 samples. Cropped clips have a resolution range from 78x78 pixels to 670x670 pixels with an average of 298x298 pixels.

B. Train and evaluation configuration

The implementation in this research adopted the mmaction2 library [17], which is a part of the open lab project. Recent SOTA models and training frameworks, including data preprocessing, model optimisation, and model evaluation process, are well integrated. The VideoMAEV2-base and UniformerV2-base were used as video feature extractor backbones. The VideoMAEV2-base pre-trained weights were optimized by distilling knowledge from the VideoMAEV2-giant model on dataset Kinetic710. Before that, the VideoMAEV2-giant model was trained on several datasets such as somethingV2, kinetic400, kinetic600, kinetic700, DIVING48, MIT, UCF101, and HMDB51 using a mask-based self-supervise learning framework. The UniformerV2-base pre-trained weights were optimized on Kinetic710 using the CLIP framework and tuned for action recognition on Kinetic400.

To have a fair comparison among different classifier heads, we fixed the training configuration, detailed as follows: learning rate of 1e-3, five warmup epochs using the step schedule, and 95 main epochs using the Cosine Annealing schedule. The learning rate is auto-scaled along with batch size (the base batch size is 256). The number of frames is 16. The input size is 224x224. Due to data imbalance among classes in the training sets, a weighted sampler was applied according to the number of samples per class. The binary-cross-entropy-with-logit loss was applied to optimize the models on the TinyVIRAT2 dataset, and the cross-entropy loss was applied to optimize the models on the Scratch and the AtomicA datasets. The difference in the applied loss functions is because the TinyVIRAT2 dataset has multiple targeted classes per video, while the Scratch and AtomicA datasets have a single targeted class per video. To evaluate the models' performances on the TinyVIRAT2 dataset, we used the F1 score, which was officially adopted for this dataset. We used the accuracy metric for the Scratch and AtomicA datasets.

C. Evaluation results

First, we compared the effectiveness of the proposed STE_head with the common classifier meanFC. Because the STE_head contains STE blocks that might be deemed more complex than the meanFC and might provide the STE_head an unfair advantage in the comparison, we inserted into the meanFC classifier spatiotemporal self-attention blocks to form another competitive classifier, named transformer classifier (Tr_head). A number of these additional blocks define the depth of Tr_head. Table I indicates the performance of these three classifiers with two backbones, VideoMAEV2-base and UniformerV2-base, on the three datasets. It is worth noting that a combination of the VideoMAEV2-base feature extractor and the meanFC classifier is the original structure of the VideoMAEV2 base model. At a depth of 1, the proposed STE_head on VideoMAEV2-base backbone improves the performance of the entire model on the two datasets TinyVIRAT2 and Scratch at 0.67 (F1 score) and 0.66% (Acc@1), respectively. Similarly, the proposed STE_head on the UniformerV2-base backbone improves the performance of the entire model on the dataset TinyVIRAT2 at 10.87 (F1 score). It also outperforms the Tr_head. There is a giant leap of performance improvement on the dataset AtomicA, more than 10% compared to the Tr_head and the meanFC, even with the VideoMAEV2-base or the UniformerV2-base backbone. This giant leap might be caused by the diversity of local spatiotemporal representation of encoded deep features from sample clips in the dataset AtomicA. Applying mean among these features in the classifier heads, as the meanFC and the Tr_head, minimizes these essential representations. In the proposed STE_head, local spatiotemporal representation is magnified in STEb blocks and the patch_base_classifier. For future work, we will investigate this outcome and provide comprehensive insight.

Second, we conducted ablation studies to understand the effectiveness of the two proposed components in the proposed STE_head separately, the STEb and the patch_base_classifier,

TABLE I: Performances of classifier heads with two different video feature extractors VideoMAEV2-base (VMAE2bb) and UniformerV2-base (UniF2bb) on datasets TinyVIRAT2, Scratch, and AtomicA

	TinyVIRAT2	Scratch	AtomicA
	F1-score	Acc@1	Acc@1
VMAE2bb + meanFC_head	78.17	96.9	50.17
VMAE2bb + Tr_head - depth1	78.28	97.65	55.17
VMAE2bb + STE_head - depth1	78.84	**97.65**	70.33
UniF2bb+ meanFC_head	74.48	90.91	58.17
UniF2bb + Tr_head - depth1	72.44	89.14	60.5
UniF2bb + STE_head - depth1	**85.35**	89.58	**71.67**

and investigated whether it is worth building a deep STE_head. In these studies, we only used the VideoMAEV2-base backbone.

Table II shows performances of different combinations among the STEb, the meanFC, and the patch_base_classifier. It is noted that a combination between the STEb and the patch_base_classifier is the STE_head with a depth of 1. We observed that the STEb provides critical value to extract meaningful spatiotemporal representation from encoded deep features. Given that the meanFC or the patch_base_classifier is applied for classification, a single STEb can boost the entire model performance.

Table III shows the performance of the entire model with different depths of the STE_head by varying the number of STEb blocks from 0 to 4. The results indicate that the current STEb block design can extract essential local spatiotemporal representation among encoded features. It provides the best performance at a depth of 1. However, this design is not scalable because the entire model performance worsens when the depth gets deeper. We observed that when the head architecture gets deeper, more neighbouring encoded features in both spatial and temporal axises are included to extract local spatiotemporal representation. It suggests that too many representations from local neighbour features embedded into encoded features may degrade encoded features' representation, leading to worse performance. In future research, we will investigate this issue to control the balance between features' representation and magnified local spatiotemporal representation so that the proposed design can be scaled deeper to improve further recognition performance.

TABLE II: Effectiveness of STEb and patch_base classifier to the entire model performance, evaluated on the dataset TinyVIRAT2

VMAE2bb	***F1-score***	***recall***	***precision***
meanFC_head	78.17	73.85	88.06
patch_base_classifier	78.01	73.68	87.48
STEb+meanFC_head	78.64	74.38	88.4
STEb+patch_base_classifier	**78.84**	**74.48**	**88.83**

TABLE III: Performance of the entire model with different depth of STE_head on the dataset TinyVIRAT2

VMAE2bb+STE_Head	***F1-score***	***recall***	***precision***
Depth = 0	78.01	73.68	87.48
Depth = 1	**78.84**	**74.48**	**88.83**
Depth = 2	78.27	74	88.09
Depth = 3	77.9	73.61	87.75
Depth = 4	77.29	72.95	87.23

IV. Conclusion

In this research, we found that even though encoded deep features were adequately optimized to be location-invariant, they still maintain their local representation that can be exploited further for performance improvement. It could be a room to build models that leverage pre-trained deep learning models for generalization capacity while adapting to custom datasets. In action recognition application, we proposed the spatiotemporal excitation classifier head consisting of four main components: temporal inception, spatial extractor, multilayer perceptron, and patch_base_classifier. This head can plug on top of any pre-trained image or video feature extractor backbones to perform action recognition. One current limitation is that although the proposed classifier head can improve the entire model performance, it lacks scalability, which we are going to address in future research.

References

[1] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, Y. Qiao, "VideoMAEV2: Scaling video masked autoencoders with dual masking", Inter. Conf. CVPR, pp. 14549-14560, 2023.

[2] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X.Tang, L.V. Gool, "Temporal segment networks for action recognition in videos," IEEE Transl. J. on PAMI, vol. 41, pp. 2740-2755, November 2019.

[3] J. Carreira, A. Zisserman, "Quo Vadis, action recognition? A new model and the kinetics dataset", Inter. Conf. CVPR, pp. 6299-6308, 2017.

[4] C. Zhang, Y. Zou, G. Chen L. Gan ,"PAN: Persistent appearance network with an efficient motion cue for fast action recognition", Inter. Conf. ACM 27th, pp. 500-509, 2019.

[5] Y. Li, B. Ji, X. Shi, J. Zhang, B. Kang, L. Wang, "TEA: Temporal excitation and aggregation for action recognition", Inter. Conf. CVPR, pp. 909-918, 2020.

[6] L. Wang, Z. Tong, B. Ji, G. Wu, "TDN: Temporal difference networks for efficient action recognition", Inter. Conf. CVPR, pp. 1895-1904, 2021.

[7] Y. Tian, Y. Yan, G. Zhai, G. Guo, Z. Gao ,"EAN: Event adaptive network for enhanced action recognition", Inter. J. Comp. Vis. vol. 130, pp. 2453-2471, August 2022.

[8] K. Li, Y. Wang, Y. He, Y. Li, Y. Wang, L. Wang, Y. Qiao, "UniFormerV2: Unlocking the potential of image ViTs for video understanding", Inter. Conf. ICCV, pp. 1632-1643, 2023.

[9] C. Ryali, Y.T. Hu, D. Bolya, C. Wei, H. Fan, P.Y. Huang, V. Aggarwal, A. Chowdhury, O. Poursaeed, J. Hoffman, J. Malik, Y. Li, C. Feichtenhofer, "Hiera: A hierarchical vision transformer without the bells-and-whistles", Inter. Conf. ML, pp. 29441-29454, 2023.

[10] G. Bertasius, H. Wang, L. Torresani, "Is space-time attention all you need for video understanding?", Inter. Conf. ICML 2021

[11] T.T. Dat, B.Q. Huy, D.C. Nhan, S.H. Seok, P.S. Lam, L. Xin, L. Khoa, DirecFormer: A directed attention in transformer approach to robust action recognition, Inter. Conf CVPR, pp. 20030-20040, 2022

[12] A.J. Piergiovanni, W. Kuo, A. Angelova, "Rethinking video ViTs: Sparse video tubes for joint image and video learning", Inter. Conf. CVPR, pp. 2214-2224, 2023

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions", Inter. Conf. CVPR, pp. 1-9, 2015.

[14] VideoMAEV model zoo, https://github.com/OpenGVLab/VideoMAEv2/blob/master/docs/MODEL_ZOO.md.

[15] TinyVIRAT2 dataset, https://www.crcv.ucf.edu/tiny-actions-challenge-cvpr2021/#tabtwo.

[16] Mediapipe package, https://developers.google.com/mediapipe/solutions/vision/pose_landmarker.

[17] Mmaction2 package, https://github.com/open-mmlab/mmaction2.