RNAS-MER: A Refined Neural Architecture Search with Hybrid Spatiotemporal Operations for Micro-Expression Recognition

Monu Verma[1] Priyanka Lubal[2], Santosh Kumar Vipparthi[3], Mohamed Abdel-Mottaleb[1]
[1]Electrical and Computer Engineering, University of Miami, USA
[2]Vision Intelligence Lab, Malaviya National Institute of Technology, Jaipur, India
[3]CVPR Lab, Indian Institute of Technology, Ropar, India
monuverma.cv@gmail.com

1. Introduction

This supplementary represents the detailed implementation settings, computational complexity analysis in terms of searching time, and analysis of generated CNN architecture by proposed RNAS and state-of-the-art NAS approaches AutoDeepLab and AutoMER.

1.1. Implementation Settings

To design the RNAS-MER, we use six layers with five hidden nodes in the inner-level search space. The down-sampling is done in the entire search space using stride 2. Further, to search for the best possible architecture, we used 30 epochs and a batch size of four due to GPU memory constraints. The main goal of the search is to learn the best hyper-parameters α, β, and η. We used stochastic gradient descent (SGD) optimizer with a momentum of 0.9, a cosine learning rate of 0.007, and a weight decay of 0.0003. The batch size is set to 4 due to GPU memory constraints. We used the reduction ratio of 16 for channel attention and average pooling in spatial attention for spatiotemporal attention. The entire search stage is accomplished in an end-to-end manner. For training a model, similar settings of searching like SGD optimizer with an initial learning rate of 0.007, weight decay $3e^{-4}$, and momentum 0.9 are initialized. The batch size is set to 12, and epochs are set to 70 for training a model. The cross-entropy loss function is used for loss optimization. We implement our model with Pytorch 1.1.0 and run all experiments on an NVIDIA RTX 2080Ti GPU. We have normalized CASME-II (152), SMIC (90), and SAMM (102) image sequences, respectively. For the composite dataset, the image sequence length is fixed to 120 for all samples. While we fixed the image resolution to 80 × 80 for all datasets. Moreover, to evaluate the performance of the proposed RNAS-MER with state-of-the-art MER approaches, we adopted recognition accuracy, unweighted average recall (UAR), and unweighted average F1-score as evaluation metrics.

Method	Pub-Year	Search Time
AutoDeepLab (3D)	CVPR-19	29:19:00
AutoMER (3D)	TNNLS-21	18:56:18
RNAS-MER-1	Ablation	13:33:00
RNAS-MER-2	Ablation	07:05:30
RNAS-MER-3	Ablation	14:54:00
RNAS-MER-4	Ablation	16:13:30
RNAS-MER	**Proposed**	**16:04:30**

Table 1: Computational Complexity Analysis of state-of-the-art MER approaches and proposed RNAS-MER with its four variants.

1.2. Computation Complexity

The computation complexity of the proposed RNAS-MER, RNAS variants and state-of-the-art approaches are compared in terms of the number of parameters, number of flops, and memory needed for trained MER models in the main draft. Here, we are providing the computational complexity in terms of search time for the proposed RNAS, variants of RNAS, and existing NAS-based methods: AutoDeepLab and AutoMER, over the Composite dataset with four RTX 2080 GPUs. The search time (H:M:S) is presented in Table 1.

1.3. Architecture Analysis

The importance of the proposed RNAS inner- and outer-level search space compared to existing NAS methods in terms of generated architecture after NAS is demonstrated in Figure 3. From Fig. 3, we observed some essential aspects of the proposed RNAS as follows:

1). In Figure 3(b), we can see that the resultant cell architecture of the AutoDeepLab is working with the previous to previous cell's resultant features only (None operation from I-1). Similarly, in Figure 3(d), for AutoMER, the previous to previous cell's resultant features are not considered in the cell structure (skip operation from I-2). However, from Figure 3(f), it is clear that the cell structure generated after RNAS uses both previous and previous to previous cells' resultant features, which is one of the benefits of NAS algorithms.

2). The cell structure of the RNAS utilizes most of the hybrid operations instead of simple convolution operations, which proved our hypothesis regarding the role of the complementary hybrid features in MER.

3). From Figure 3, it is evident that the proposed RNAS chooses the small-scaled filter size $1 \times 1 \times 1$ along with other convolutional with $3 \times 3 \times 3$ and $5 \times 5 \times 5$ filter sizes to extract the minute but effective micro-expressive variations. Therefore, based on the experimental results and generated architecture, as shown in Figure 3, we can conclude that our proposed RNAS algorithm gives the best optimum architecture for MER.

We firmly confirm the release of all our codes and searched architectures to ensure the re-producibility of our results.