Regional Attention with Architecture-Rebuilt 3D Network for RGB-D Gesture Recognition

Benjia Zhou, Yunan Li, Jun Wan

1 Macau University of Science and Technology, Macau SAR, China
2 National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
3 School of Computer Science and Technology, Xidian University, China
4 Xi’an Key Laboratory of Big Data and Intelligent Vision, China
5 School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

19098536ii20001@student.must.edu.mo, yunanli@xidian.edu.cn, jun.wan@ia.ac.cn

Abstract

Human gesture recognition has drawn much attention in the area of computer vision. However, the performance of gesture recognition is always influenced by some gesture-irrelevant factors like the background and the clothes of performers. Therefore, focusing on the regions of hand/arm is important to the gesture recognition. Meanwhile, a more adaptive architecture-searched network structure can also perform better than the block-fixed ones like Resnet since it increases the diversity of features in different stages of the network better. In this paper, we propose a regional attention with architecture-rebuilt 3D network (RAAR3DNet) for gesture recognition. We replace the fixed Inception modules with the automatically rebuilt structure through the network via Neural Architecture Search (NAS), owing to the different shape and representation ability of features in the early, middle, and late stage of the network. It enables the network to capture different levels of feature representations at different layers more adaptively. Meanwhile, we also design a stackable regional attention module called dynamic-static Attention (DSA), which derives a Gaussian guidance heatmap and dynamic motion map to highlight the hand/arm regions and the motion information in the spatial and temporal domains, respectively. Extensive experiments on two recent large-scale RGB-D gesture datasets validate the effectiveness of the proposed method and show it outperforms state-of-the-art methods. The codes of our method are available at: https://github.com/zhoubenjia/RAAR3DNet.

Introduction

Gesture is produced as part of deliberate action and signs, involving the motion of the up body, especially the arms, hands, and fingers. Video-based classification makes an essential component in gesture recognition. It has been applied to many human-centred tasks, such as apparent personality analysis (Li et al. 2020), sign language recognition (Cui, Liu, and Zhang 2019) and human-computer interaction (HCI) (Wang et al. 2016b). The handcrafted features (Wan, Guo, and Li 2015) are always used for gesture recognition in the early years. The powerful feature representation ability of deep learning also promotes the application of neural networks in the field of gesture recognition (Karpathy et al. 2014; Li et al. 2016; Miao et al. 2017; Simonyan and Zisserman 2014; Narayana, Beveridge, and Draper 2018).

For most of the deep learning-based gesture recognition methods, some popular networks like ResNet (He et al. 2016), SENet (Hu, Shen, and Sun 2018) and Inflated 3D Network (I3D) (Carreira and Zisserman 2017) are usually employed as the backbone for gesture recognition. Although these networks have achieved great success in many tasks, it still worth pointing that the same modules are shared from shallow to deep layers in these networks. Even the modules in networks like I3D that employ multi-branch structure to improve the width and diversity are fixed and all the same through the network. However, features in the early stage and late stage are quite different. Features in the early stage are low-level features, which show the visual texture in each frame, whereas the high-level features in the late stage are abstracted and more related to the class of gestures. Therefore, it is not suitable to use the same structure to learn different features, and then we need to make the network more adaptive and automatically determine what the shape is for different parts of it.

Meanwhile, one of the most significant challenges hindering the improvement of recognition accuracy is the influence of gesture-irrelevant factors, such as backgrounds, different clothes of performers, and so on. The various textures and appearances could mislead the network to learn inconsequential or less important features. For dynamic gesture recognition from a video sequence, we believe it is vital to focus on gesture movements, such as hands, arms or elbows of the performers. Many researchers notice that it is critical to make the network focusing on the gesture regions both spatially and temporally. Modules such as hand detector (Wang et al. 2016c; Liu et al. 2017), and additional modalities of data like optical flow (Li et al. 2017) or saliency (Li et al. 2018) are widely used via combining with the raw RGB (and depth) data to design different algorithms. However, most of them require extra offline oper-
Related Work

Evolution of Approaches for Gesture Recognition

The study on gesture taxonomies and representations has been continued for many years. Early methods are often based on handcrafted features (Klaser, Marszalek, and Schmid 2008; Wan et al. 2014). Recently, the rapid progress of deep learning boosts many deep neural network-based approaches (e.g., hand detection, optical flow calculation) in advance. It would increase time complexity because of using hand detector network in the testing stage. Therefore, it may be more reasonable if the attention maps of gesture regions are learned along with the task of gesture recognition in the same network.

Inspired by the above discussions, we propose a regional attention with architecture-rebuilt 3D network for dynamic gesture recognition based on RGB-D data, which is illustrated in Fig. 1. We take the I3D network as the backbone and employ the theory of NAS to find the optimal combination of different operations in each module of the network. To make the network focus on the gesture regions, we propose a regional attention module DSA, which includes a static attention sub-module (SAtt) and dynamic attention sub-module (DAtt). For static attention, we learn a heatmap of hands or body for each frame with the supervision of the Gaussian map of skeleton keypoints. It indicates where the hands/arms are and highlights those regions. For dynamic attention, we present a fast approximate rank pooling algorithm to learn the accumulated dynamic images, which reduces the time complexity a lot when compared with the traditional rank pooling techniques (Bilen et al. 2016; 2017) and thus can give a real-time dynamic image computation. Then with the DSA structure applied, the network can pay attention to the gesture regions spatiotemporally. Our contributions can be summarized as three-fold:

1. We replace the structure-fixed modules in the general network with automatically reconstructed cells via NAS. The cells in the early, middle, and late stages of the network can have different structures and learn the low-level and high-level features more adaptively.

2. We propose a stackable attention structure, called DSA, to generate attention map in both spatial and temporal space. DSA consists of the SAtt and DAtt sub-modules. SAtt highlights the hands/arms features via an online learnable Gaussian skeleton heatmap while DAtt captures the gesture motions via the proposed fast approximate rank pooling algorithm with decreasing the time complexity to a large extent.

3. Extensive experiments that prove the integration of our designs can ultimately improve the performance of gesture recognition. Experiments demonstrate that our method can strike the balance between good performance and low computation burden, and outperform those top techniques on two large-scale gesture datasets.
Attention mechanism has been wildly used in both low-level and high-level tasks like pose estimation (Chu et al. 2017), object detection (Li et al. 2019a) and image restoration (Li et al. 2019b). As the interference of backgrounds, the clothes of performers and the diversity of presentation for the same gesture are still the barrier for improving the recognition accuracy, many researchers also employ the attention mechanism to guide the network to focus on the gesture itself through the video. Some methods concentrate on the regions of gesture in each frame. Liu et al. (Liu et al. 2017) leverage the faster R-CNN (Ren et al. 2015) as the hand detector to highlight the corresponding regions. Lin et al. (Lin et al. 2018) utilize both detected hands and skeleton information to further focus on the gesture. The work (Narayana, Beveridge, and Draper 2018) uses a focus of attention network (FOANet) to extract global raw data, local left and right hand regions via different networks. The other ones mainly concern about the motion information among frames. In (Li et al. 2018), Miao et al. 2017), the additional modality of optical flow data is used to capture the movements in videos. Zhang et al. (Zhang et al. 2018) explore attention modules used in different gates of the LSTM when incorporating it and the 3D CNNs (Tran et al. 2015). Although with these techniques the network can focus on the gestures, extra modalities of data served as an input of the network or offline training process for models like hand detector are always inevitable. On the contrary, the attention module of DSA in our method does not require any other data except for the RGB-D ones from the original dataset. The module can also be learned end-to-end within the recognition network, and it makes the network easier to apply to different situations with less time complexity. Meanwhile, since the DSA highlights both body parts in each frame and the movements through the adjacent frames, it facilitates the advantage of the temporal and spatial attention concurrently.

Proposed Method

Overview of the network

The pipeline of RAAR3DNet is shown in Fig.1. The network takes the I3D network as the backbone, and leverages NAS to automatically find the optimal structure for early, middle and late stage of the network to replace the original shape-fixed Inception Modules. Meanwhile, to better focus on the location and movement information of gesture-related parts like hands and arms, as show in Fig.2, a stackable regional attention module DSA is embedded in the network.

Figure 2: The detail of DSA module. It has two sub-modules of dynamic attention (DAtt) and static attention (SAtt), which are sequentially combined together.

Local Network Structure Search in 3D

We use PC-darts (Xu et al. 2019), which based on gradient descent and more efficient than darts (Liu, Simonyan, and Yang 2019), to search more efficient architecture for gesture
recognition. In the search stage, we search and rebuild more adaptively structures and replace the Inception Modules in the I3D network with them. As shown in Fig.1 because the features in different stages of the backbone have different solutions, we design three kinds of cells to learn the different levels of features. Specifically, we replace the first two Inception Modules in the I3D with cell1, the middle five Inception Modules with cell2, and the last two Inception Models with cell3. Each cell represents a directed acyclic graph (DAG) with k nodes \{m\}_{i=0}^{k-1}. Each node indicates an output of a network layer, and each edge \((i, j)\) of the DAG indicates the information flow from node \(m_i\) to \(m_j\), which consists of the candidate operations weighted by the architecture parameter \(o^{(i,j)}\). Different from the I3D network which only employ three operation ‘Conv, 1 × 1 × 1’, ‘Conv, 3 × 3 × 3’ and ‘Max-pooling, 3 × 3 × 3’ in the original Inception Module, we add two extra operations: ‘Conv, 1 × 3 × 3’ and ‘Conv, 3 × 1 × 1’ to perform convolution either in spatial or temporal domain only. That is because the size of features in the spatial domain (the height and width of frame) and temporal domain (the number of frames) differ a lot. Consequently, it is not necessary to perform the spatial and temporal convolution together all the time. Besides, a novel operator, dilated convolution ‘dlConv, 3 × 3 × 3’ is introduced into the search space for searching more powerful architecture. So the final search space \(O\) we defined includes seven candidate operations: ‘Zero’, ‘Identity’, ‘dlConv, 3 × 3 × 3’, ‘Conv, 1 × 1 × 1’, ‘Conv, 3 × 3 × 3’, ‘Conv, 1 × 3 × 3’, ‘Conv, 3 × 1 × 1’, where ‘Zero’ and ‘Identity’ mean no feature flow connection, direct feature flow without any convolution/pooling operations, respectively. ‘Conv, x × y × z’ and ‘dlConv, x × y × z’ represent 3D vanilla and dilated convolution that kernel with the size of \(x \times y \times z\), respectively. Similar to the work of (Xu et al. 2019), the architecture parameters \(o^{(i,j)}\) is optimized via the stochastic mini-batch gradient descent algorithm. Specially, for each edge \((i, j)\), we can formulate it by a function \(o^{(i,j)}(\cdot)\) where \(o^{(i,j)}(m_i) = \sum_{o \in O} \exp(o^{(i,j)} / \eta_i) \cdot o(m_i)\). Softmax \(\eta_o^{(i,j)} = \sum_{r \in O} \exp(o^{(i,j)} / \eta_i) / \sum_{r \in O} \exp(o^{(i,j)} / \eta_i)\) is utilized to relax architecture parameter \(o^{(i,j)}\) into operation weight \(o \in O\). The intermediate node can be denoted as \(m_j = \sum_{i < j} o^{(i,j)}(m_i)\). And the output node \(m_{k-1}\) concat all the intermediate nodes. The cross-entropy loss is utilized for the training loss \(L_{train}\) and validation loss \(L_{val}\). Then the network parameters \(w\) and architecture parameters \(o^{(i,j)}\) are learned via solving the bi-level optimization problem:

\[
\begin{align*}
\min_{\alpha} \quad & L_{val}(w^*(\alpha), \alpha), \\
\text{s.t.} \quad & w^*(\alpha) = \arg \min_{w} L_{train}(w, \alpha) \quad (1)
\end{align*}
\]

When the search converging, the optimal operation between the pair of node \((i, j)\) can be obtained by replacing each mixed operation \(o^{(i,j)}\) with the most likely operation:

\[o^{(i,j)} = \max_{o \in O, o \neq \text{zero}} o_o^{(i,j)}\]

**Dynamic-static Attention**

In the DSA module, two kinds of attention are learned by two sub-modules of dynamic attention and static attention, respectively. In the dynamic attention module, we use our proposed fast approximate rank pooling to learn the motion of gestures, whereas a Gaussian guidance heatmap is learned in the static attention module under the train-phase-only supervision of skeleton data.

**Dynamic attention sub-module**

The dynamic attention sub-module concerns the effective motion information among frames. We aggregate the intermediate spatiotemporal-structural information into a dynamic image instead of using the 3D manipulation directly like (Tran et al. 2015) to avoid time-consuming processing. To improve efficiency, we propose a fast approximate rank pooling algorithm based on (Bilen et al. 2016), and it can reduces time complexity drastically.

**Dynamic Image via Fast Approximate Rank Pooling**

According to (Smola and Schölkopf 2004), the rank pooling map can be obtained by solving a convex optimization problem using the objective function of RankSVM. This process is known as rank pooling (Fernando et al. 2017). According to the work (Bilen et al. 2017), the approximation of rank pooling can be defined as:

\[
d^* \propto \sum_{t_1 > t_2} V_{t_1} - V_{t_2} = \sum_{i = 1}^{T} \beta_t V_t, \quad (2)
\]

where \(T\) is the length of the video clip and \(\beta_t = 2t - T - 1\). \(V_t\) is the feature map of time step \(t\). The result of Eq.2 is the dynamic image \(DI\). In this computation, the time complexity is related to the number of frames, for a T-frame video clip, the time consumption can be up to \(T(T - 1)...1 = T!\), and it is still high when processing a long-term video.

To simplify the processing of approximating rank pooling, we further study Eq.2, and achieve the fast approximate rank pooling via computing the relation between the dynamic image of frame \(n\) to \(m(m > n)\) and that of frame \(n + 1\) to \(m + 1\) as:

\[
DI(n + m + 1) = DI(n, m) + DI(m, m) \times V(I_n) + V(I_{m+1}) - 2 \sum_{I_{n+1}} V(I_l) \quad (3)
\]

where \(I_n\) is the first frame of last dynamic image, and \(I_{m+1}\) is the last frame of the current dynamic image. The last term is the overlapped part between two dynamic images. In this way, the computation of dynamic image can be irrelevant to the number of frames.

After obtaining the dynamic image, at each time \(t\), we do the normalization for the rank parameters as:

\[
DI_{(c, x, y)} = \frac{DI_{(c, x, y)} - DI_{min}}{DI_{max} - DI_{min}}, \quad (4)
\]

where \(DI_{(c, x, y)}\) represents the dynamic image at \(c\)-th channel and the coordinate of \((x, y)\). \(DI_{min}\) and \(DI_{max}\) are the minimum and maximum of the dynamic image, respectively.
Structure of dynamic attention sub-module. Considering the value of dynamic images varies along with the amplitude of movement, which may be out of the range $[0, 1]$. Therefore, we first apply the normalization to the dynamic images. The batch normalization \cite{ioffe2015batch} is conducted to eliminate the influence of distribution inside a mini-batch:

$$ DI_{\text{norm}}(t) = \left( \frac{DI(t) - E[DI(t)]}{\sqrt{\text{Var}[DI(t)]}} \right) \times \gamma + \beta, \quad (5) $$

where the expectation $E$ and variance $\text{Var}$ are computed over the training data set so that this normalization can consider the data distribution in the mini-batch, $\gamma$ and $\beta$ are the scale and shift parameter of the Batch Normalization layer.

We add 1 to each pixel of the guidance map to avoid the zero-value attention map, which may lead to a vanishing of feature maps. Then we can get the DAmm guided feature map $O_{D,t}$ as:

$$ O_{D,t}^l = [(DI_{\text{norm}} + 1) \odot f^l_t] \odot f^l_t, \quad (6) $$

where $O_{D,t}^l$ is the guided feature map of layer $l$ at time step $t$ and $DI_{\text{norm}}$ is dynamic gesture feature map of layer $l$ at time step $t$. Then, on each channel of the feature map $f^l_t$, we use the element-wise product operation $\odot$ to generate the attention map. After that, $O_{D,t}^l$ can be derived by the $1 \times 1$ convolution of the corresponding attention map and the feature map $f^l_t$, which we describe with $\odot$ in Eq. (6).

Finally, the shape of $O_{D,t}^l$ is the same as $f^l_t$.

Structure of the static attention sub-module. Being aware of the location of hands and arms is important to avoid the interference by gesture-irrelevant factors. Therefore, we try to highlight the location of hands/arms in each frame via the static attention sub-module. It is guided by a heatmap, which is related to the location of keypoints in hands/arms regions. The heatmap is derived from a lightweight gesture region heatmap generation network – HeatmapNet.

input image

| HMBlock | multi-scale heatmaps |
|---------|----------------------|
|         |                      |
|         |                      |
|         |                      |
|         |                      |
|         |                      |

Figure 3: The framework of our HeatmapNet used for guidance heatmap generation. With the Gaussian skeleton map derived via OpenPose \cite{cao2017openpose}, the ground truth, we learn a heatmap to indicate the hand/arm regions via a lightweight sub-network, which is composed of several cascaded Heatmap Blocks. These blocks can learn a increasingly clear heatmap stage-by-stage.

HeatmapNet. As shown in Fig. 3 to derive the ground truth of hands/arms’ location, we first employ the OpenPose \cite{cao2017openpose} to generate skeleton data from raw RGB videos. Then we obtain the Gaussian skeleton map according to Eq. (7):

$$ H_t = G(I_t, P_t, \sigma), \quad (7) $$

where $I_t$ and $P_t$ are the input frame and the skeleton points at time $t$, and its corresponding Gaussian map is represented as $H_t$, which is generated via Gaussian function $G$ with standard deviation $\sigma$. After having the Gaussian map as the ground truth, the HeatmapNet is used to learn the static guidance heatmap. Considering the cascaded network is always used in high-level vision tasks \cite{newell2016stacked} to learn both global and local cues and refine the details of feature map, we also use a cascaded structure to predict the static guidance map. In each stage of the network, we have a heatmap block (HMBlock) for prediction. The prediction of static guidance map for time step $t$ at stage $s$ can be formulated as:

$$ \tilde{h}_t^s = \begin{cases} F_\theta(I_t) & s = 1, \\ F_\theta(I_t, \tilde{h}_t^{s-1}) & s > 1, \end{cases} \quad (8) $$

where $F_\theta$ is the residual block with parameter $\theta$. $I_t$ is the input frame at time step $t$ and $\tilde{h}_t^{s-1}$ is the predicted map at the previous stage $s - 1$. With the stage-wise learning of the heatmap, the regions related to the gesture can be highlighted more apparently. Finally, the guidance maps in all the stages are combined by averaging via Eq. (9):

$$ h_t = \frac{1}{S} \sum_{s=1}^{S} \tilde{h}_t^s, \quad (9) $$

where $S$ is the number of stages. To better guide the features of gesture in both low-level and high-level, we then generate multi-scale guidance maps by max pooling. The size of these heatmaps is in accord with those of corresponding feature maps. Then the static guidance maps are fed into different layers of the network to highlight the gesture-relevant regions. Here we use MSE loss $\mathcal{L}_m^m$ to learn the guidance heatmap. It needs to be emphasized that unlike the previous methods \cite{liu2017hmr, wang2017structured} based on the detection techniques (i.e., faster R-CNN \cite{ren2015faster}) through the training and test phase, the HeatmapNet is only learned in the training phase, and avoids employing extra input data or complex computation when recognizing gestures. We only use the original skeleton data generated via OpenPose algorithm \cite{cao2017openpose} in the training phase, and in the inference phase, we only need to input the original image without additional skeleton information. And our network can automatically focus on the performer’s hands and arms. Meanwhile, The FLOPs of our HMBlock is about $5.51 \times 10^8$, whereas that of faster R-CNN is about 36 times of ours at $2.02 \times 10^{10}$. Therefore, the addition of HMBlock in the training phase will not increase the burden of learning too much.

Through the SAmm sub-module, we can obtain the guided feature map $O_{S,t}^l$ as Eq. (10):

$$ O_{S,t}^l = [(h_t^l + 1) \odot f^l_t] \odot f^l_t, \quad (10) $$

where $O_{S,t}^l$ is the guided feature map of layer $l$ at time step $t$ and $h_t^l$ is static gesture heatmaps of layer $l$ at time step $t$. $h_t^l$ is generated via the HeatmapNet. Similar to the dynamic attention sub-module, the processing of “adding 1” to each pixel and the process for shape consistency are also conducted before the element-wise multiplication.
Network Training

The entire network can be trained in an end-to-end manner after overall network architecture is searched out. The main branch of gesture recognition with RGB/depth data is trained together with the branch of HeatmapNet. The parameter of the main branch is learned by minimizing the cross-entropy loss $L_{cls}$. Meanwhile, the online heatmap sub-network is trained together to learn a guidance heatmap via the MSE loss $L_{hm}$. Then we jointly optimize the entire network with a multi-task loss function. The Loss function can be expressed as:

$$
L^m = L_{cls}^m + \gamma L_{hm}^m
$$

$$
= - \sum_{k=1}^{K} p_k^m \log(p_k^m) + \gamma \|F(I^m) - \mathcal{H}\|^2, \tag{11}
$$

where $m$ indicates the modality of data, which can be either RGB or depth. $p_k = \{p_1, p_2, \ldots, p_K\}$ is the ground-truth probability distribution of the $k$-th class of the gesture, and $\hat{p}_k$ is its estimation. $\mathcal{H}$ is the ground truth Gaussian skeleton map, and $F(\cdot)$ is the mapping function the sub-network learning for HMBlocks as mentioned in Section . $\gamma$ is the balancing parameter and we have $\gamma = 100$ in this paper.

Experiments

Datasets

We evaluate our method and compare it with other state-of-the-art methods on two RGB-D gesture datasets: Chalearn IsoGD dataset (Wan et al. 2016) and NvGesture dataset (Molchanov et al. 2016). Meanwhile, we also conduct the ablation studies on a hand-centred action dataset, THU-READ dataset (Tang et al. 2017, 2018) to show the generality of our network.

Experimental setup

Our experiments are all conducted with Pytorch on the NVIDIA RTX 2080 Ti GPU. During the training process, the inputs are spatially resized to $256 \times 256$ and then cropped into $224 \times 224$ randomly in the training stage, and are center cropped into $224 \times 224$ in the test stage. The data is fed into the network with a mini-batch of 64 samples. For optimization, We use the SGD optimizer to train our network with the weight decay of 0.0003 and the momentum of 0.9. The learning rate is initially fixed as 0.01 and if the accuracy on the validation set not improved every 3 epochs, it is reduced by 10 times. The training work is stopped after 80 epochs or when the learning rate is under 1e-5.

Comparison with state-of-the-art methods

Our method is compared with recent state-of-the-art methods on IsoGD and NvGesture dataset. Table 1 shows the comparison on IsoGD dataset. Since most of methods release their result on the validation subset, we also conduct experiments on it for a fair comparison.

As can be seen in Table 1, our proposed method achieves the best performance on all conditions of using single RGB/depth data and using the fusion of them. For RGB data, our method outperforms the second-best one, Zhu et al. (2019) at about 5%, even though their methods are trained with a complex composition of Res3D, Gated convLSTM and 2D CNNs. Ours is also 17% higher than Res3D (Li et al. 2017), which achieves 1st place in the 2nd round of Chalearn LAP large-scale isolated gesture recognition challenge. The performance on the depth data is a little lower than that on RGB data. That may because the texture showing the details of fingers is not available in depth data. However, the recognition result on depth data is similar to that on RGB data. Ours also outperform the second-best one, 3DDSN (Duan et al. 2018) about 6%. The performance on the fusion RGB-D data also shows the effectiveness of our method, which outperforms Zhu et al.’s method at about 5.6%.

The comparison on NvGesture dataset is shown in Table 2. As can be seen, our method can still achieve a competitive result on this dataset. Compared with the method of MTUT et al. (Molchanov et al. 2016), which uses a combination of time-consuming models of 3D and LSTM, our network achieves about 4.5% improvement on RGB data. Meanwhile, it outperforms GPM (Gupta et al. 2019), the second-best result on depth data at about 1.17%. The gap between the performance of ours and GPM becomes more significant on the fusion of RGB-D data at 2.49%. Noticing that our result is also better than the human recognition accuracy even without auxiliary data like IR as provided in that dataset. It shows the effectiveness of our network architecture searching strategy and the attention module for improving the recognition performance.

Ablation studies

In this section, we perform several groups of experiments to verify the effect of each component of our network, including the idea of automatically searching the architecture of network and the spatiotemporal DSA module.
Effect of network architecture searching. Here we first compare the performance of the baseline raw I3D network and our proposed network on three different datasets - IsoGD, NvGesture and THU-READ dataset.

| Method                      | Modality | Accuracy (%) |
|-----------------------------|----------|--------------|
| HOG+HOG\(^2\) (Ohn-Bar and Trivedi 2014) | RGB      | 24.50        |
| SNN (Yang and Tran 2018)   | Depth    | 36.30        |
| R3DCNN (Molchanov et al. 2016) | Depth   | 70.70        |
| DTA (Yang, Molchanov, and Kautz 2018) | RGB      | 85.00        |
| MTU (Abavisani, Joze, and Patel 2019) | RGB-D   | 86.67        |
| RAAR3DNet(Ours)             | RGB-D    | 88.59        |

Table 2: Results on the NvGesture dataset.

Effect of attention mechanism. Table 4 shows the influence of different attention schemes. For a fair comparison, we remove other modules and only take the original I3D network as the baseline. We give the result of the original I3D, the network only with dynamic attention or static attention, and finally the entire DSA module.

| Strategy   | Recognition Rate |
|------------|------------------|
| baseline(I3D) | 81.67% / 83.12% |
| DAtt       | 83.54% / 84.58% |
| SAtt       | 83.75% / 85.21% |
| DSA(DAtt+SAtt) | 84.79% / 85.83% |

Table 4: Performance of attention modules on the NvGesture dataset.

Conclusion

In this paper, we propose a regional attention with searched architecture 3D network for gesture recognition basing on RGB-D data. We take the I3D network as the backbone, and employ NAS to search the optimal connection among features in different stages of it. In this way, the structure of the network can fit the low-level and high-level features better and improves the recognition result. Meanwhile, we also design a stackable attention module of DSA to guide the network to pay more attention to the hand/arm regions in each frame and the motion trajectory among video sequence. Finally, comparisons with state-of-the-art methods on two gesture datasets prove the effectiveness of the proposed method.
Acknowledgements

This work was supported by the National Key R&D Program of China under Grant #2018YFC0807500, the Chinese National Natural Science Foundation Projects #61961160704, #61876179, #62002271, #61772396, #61772392, #61902296, the Fundamental Research Funds for the Central Universities #JB180301, Xi’an Key Laboratory of Big Data and Intelligent Vision #2018OS053ZD4CG37, the External cooperation key project of Chinese Academy Sciences #173211KYSB2020002, the Key Project of the General Logistics Department Grant No.ASW17C001, Science and Technology Development Fund of Macau (No. 0019/2018/ASC, 0010/2019/AFJ, 0025/2019/AKP).

References

Abavisani, M.; Joze, H. R. V.; and Patel, V. M. 2019. Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training. In CVPR, 1165–1174.
Bilen, H.; Fernando, B.; Gavves, E.; and Vedaldi, A. 2017. Action recognition with dynamic image networks. TPAMI.
Bilen, H.; Fernando, B.; Gavves, E.; Vedaldi, A.; and Gould, S. 2016. Dynamic image networks for action recognition. In CVPR, 3034–3042.
Cao, Z.; Simon, T.; Wei, S.-E.; and Sheikh, Y. 2017. Real-time Multi-Person 2D Pose Estimation using Part Affinity Fields. In CVPR.
Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In CVPR, 4724–4733. IEEE.
Chu, X.; Yang, W.; Ouyang, W.; Ma, C.; Yuille, A. L.; and Wang, X. 2017. Multi-context attention for human pose estimation. In CVPR, 1831–1840.
Cui, R.; Liu, H.; and Zhang, C. 2019. A deep neural framework for continuous sign language recognition by iterative training. IEEE TMM 21(7): 1880–1891.
Duan, J.; Wan, J.; Zhou, S.; Guo, X.; and Li, S. Z. 2018. A unified framework for multi-modal isolated gesture recognition. TOMM 14(1): 21.
Fernando, B.; Gavves, E.; Oramas, J.; Ghodrati, A.; and Tuylaeaars, T. 2017. Rank pooling for action recognition. TPAMI 39(4): 773–787.
Gupta, V.; Dwivedi, S. K.; Dabral, R.; and Jain, A. 2019. Progression Modelling for Online and Early Gesture Detection. In 3DV, 289–297. IEEE.
He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In CVPR, 770–778.
Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In CVPR, 7132–7141.
Hu, T.-K.; Lin, Y.-Y.; and Hsu, P.-C. 2018. Learning Adaptive Hidden Layers for Mobile Gesture Recognition. In AAAI, 6934–6942.
Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 448–456.
Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Fei-Fei, L. 2014. Large-scale video classification with convolutional neural networks. In CVPR, 1725–1732.
Klaser, A.; Marszalek, M.; and Schmid, C. 2008. A spatio-temporal descriptor based on 3d-gradients. In BMVC, 1–10.
Köpïkliü, O.; Gunduz, A.; Kose, N.; and Rigoll, G. 2019. Real-time hand gesture detection and classification using convolutional neural networks. In FG, 1–8. IEEE.
Li, H.; Liu, Y.; Ouyang, W.; and Wang, X. 2019a. Zoom out-and-in network with map attention decision for region proposal and object detection. IJCV 127(3): 225–238.
Li, Y.; Miao, Q.; Ouyang, W.; Ma, Z.; Fang, H.; Dong, C.; and Quan, Y. 2019b. LAP-Net: Level-Aware Progressive Network for Image Dehazing. In ICCV, 3276–3285.
Li, Y.; Miao, Q.; Tian, K.; Fan, Y.; Xu, X.; Li, R.; and Song, J. 2016. Large-scale gesture recognition with a fusion of RGB-D data based on the C3D model. In ICPR, 25–30. IEEE.
Li, Y.; Miao, Q.; Tian, K.; Fan, Y.; Xu, X.; Li, R.; and Song, J. 2018. Large-scale gesture recognition with a fusion of RGB-D data based on saliency theory and C3D model. TCSVT 28(10): 2956–2964.
Li, Y.; Miao, Q.; Tian, K.; Fan, Y.; Xu, X.; Ma, Z.; and Song, J. 2017. Large-scale Gesture Recognition with a Fusion of RGB-D Data Based on Optical Flow and the C3D Model. PRL.
Li, Y.; Wan, J.; Miao, Q.; Escalera, S.; Fang, H.; Chen, H.; Qi, X.; and Guo, G. 2020. CR-Net: A Deep Classification-Regression Network for Multimodal Apparent Personality Analysis. IJCV 1–18.
Lin, C.; Wan, J.; Liang, Y.; and Li, S. Z. 2018. Large-Scale Isolated Gesture Recognition Using a Refined Fused Model Based on Masked Res-C3D Network and Skeleton LSTM. In FG, 52–58. IEEE.
Liu, H.; Simonyan, K.; and Yang, Y. 2019. Darts: Differentiable architecture search. ICLR.
Liu, Z.; Chai, X.; Liu, Z.; and Chen, Y. 2017. Continuous gesture recognition with hand-oriented spatiotemporal feature. In Proceedings of the IEEE International Conference on Computer Vision Workshops, 3056–3064.
Miao, Q.; Li, Y.; Ouyang, W.; Ma, Z.; Xu, X.; Shi, W.; and Cao, X. 2017. Multimodal Gesture Recognition Based on the ResC3D Network. In ICCV Workshops, 3047–3055.
Molchanov, P.; Yang, X.; Gupta, S.; Kim, K.; Tyree, S.; and Kautz, J. 2016. Online Detection and Classification of Dynamic Hand Gestures With Recurrent 3D Convolutional Neural Network. In CVPR, 4207–4215. IEEE.
Narayana, P.; Beveridge, R.; and Draper, B. A. 2018. Gesture recognition: Focus on the hands. In CVPR, 5235–5244.
Newell, A.; Yang, K.; and Deng, J. 2016. Stacked hourglass networks for human pose estimation. In ECCV, 483–499.
