An isolated Vietnamese Sign Language Recognition method using a fusion of Heatmap and Depth information based on Convolutional Neural Networks.

Xuan-Phuoc NGUYEN*, Thi-Huong NGUYEN†, Duc-Tan Tran‡, Tien-Son Bui §, Van-Toi NGUYEN‡

* Faculty of Computer Science, PHENIKAA University, Yen Nghia, Ha Dong, Hanoi 12116, Vietnam
† Hanoi College of High Technology, Hanoi, Vietnam
‡ Faculty of Electrical and Electronic Engineering, PHENIKAA University, Yen Nghia, Ha Dong, Hanoi 12116, Vietnam.
§ Hanoi University of Industry, Vietnam.

***Abstract*—In recent years, interest in sign language recognition has continuously increased. However, recognition methods for exploiting the combination of RGB and depth data are limited, especially applied to Vietnamese sign language. This paper presents an isolated Vietnamese sign language recognition method using a novel streams-enhanced 3D ConvNet. The experimental results demonstrate the superiority of the proposed method over other methods using variations from RGB, depth, and RGB-D data. The speed and accuracy of our method are better than those of previous methods.**

***Index Terms*—Keywords: sign language recognition**

I. Introduction

As we know, deaf and hard-of-hearing people use sign language to communicate. They use their upper body to express speech, including hand gestures, facial expressions, and body language. The need for research into sign language recognition stems from its potential to empower the deaf and hard-of-hearing community, facilitating seamless communication with technological interfaces. The development of accurate and efficient algorithms in this field promises to bridge the communication gap, promote communication accessibility for the deaf, and promote inclusivity in technological development.

Sign language recognition presents challenges within computer vision. Existing studies have explored various modalities for feature representation, such as RGB frames [1], [2], [3], optical flows [4], audio waves [5], and human skeletons [6], [7]. Among these modalities, skeleton-based action recognition has received increasing attention in recent years due to its action-focusing nature and compactness.

In practice, human skeletons in a video are mainly represented as a sequence of joint coordinate lists, where pose estimators extract the coordinates. Since only the pose information is included, skeleton sequences capture only action information while being immune to contextual noises, such as background variation and lighting changes. Among all the methods for skeleton-based action recognition [8], [9], [10], graph convolutional networks (GCN) [7] have been one of the most popular approaches. However, these methods are limited in aspects such as: A small perturbation in the coordinates often leads to completely different predictions [11]; difficult to combine with varying modality datasets such as RGB, depth, RGB, optical flows, and skeletons.

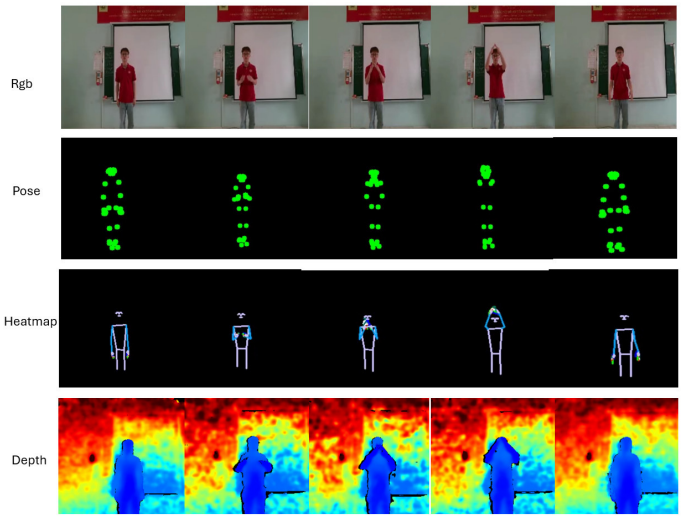

Fig. 1. Pose estimators produce 2D poses that are represented by stacks of heatmaps of skeleton joints.

In this paper, we propose a novel framework, Ad2C (Advantage 2 Convolution), which can solve the difficulties mentioned above. Specifically, Figure 1's present pose estimators produce 2D poses that Ad2C accepts as input. Rather than operating on coordinates on a human skeleton graph, 2D poses are represented by stacks of heatmaps of skeleton joints. A 3D heatmap volume will be created by stacking the heatmaps at various timesteps along the temporal dimension. To identify actions, Ad2C based on I3D architecture and advandge it, uses two flow 3D-CNN for the 3D heatmap volume and depth data.

Table 1 summarizes the key distinctions between Ad2C and the most popular skeleton-based recognition method.

TABLE 1
SUMMARIZES THE KEY DISTINCTIONS BETWEEN AD2C AND GCN-BASED TECHNIQUES.

	skeleton-based recognition	Ad2C
Input	Skeleton	Varied
Format	Coordinates	Heatmap Volumes Depth Volumes
Architecture	GCN	3D-CNN

The weaknesses of some of the methods can be solved with Ad2C. Initially, applying 3D heatmap volumes is kinder to the original posture estimate; our experience indicates that Ad2C performs brilliantly across input acquired by various methods. Furthermore, Ad2C benefits from the latest developments in 3D convolutional neural networks and is simpler to combine other modalities in multi-stream convolutional networks. This feature creates a lot of chances for improvement in terms of recognition performance through advantages architecture. We carry out extensive experiences over a number of our datasets, including RGB, Depth, RGB-D, and heatmap-depth, to confirm the efficacy and efficiency of Ad2C.

The main contributions of our study are folders.

- We propose a new model, called Ad2C, which combines the generated Heatmap-Depth data and gives good performance.
- We experimentally the performance of fusion methods in our dataset, including the evaluation of RGB, Depth, RGB-D, and Heatmap- Depth discrete data.

The remainder of this article is organized as follows. Part II discusses our related work. Section III presents details of our proposed method. Part IV describes the experiment result. Finally, there are conclusions and future development directions in Part V.

II. RELATED WORK

I3D. [41] The video classification model, Two-Stream 3D ConvNets, is a published basic method. This method combines two I3D networks, one trained on RGB inputs and another on optical flow inputs. The two streams capture complementary information about the appearance and motion of the video frames. The I3D networks are based on inflating the filters and pooling kernels of the Inception v1 model into 3D, allowing them to leverage the ImageNet pre-trained weights and architectures. The two-stream I3D model achieves state-of-the-art performance on several action recognition benchmarks, such as UCF-101 and HMDB-51, after pre-training on the Kinetics dataset [1]. I3D was initially applied to Kinetics data about activities. However, I3D was later applied to other datasets on sign language recognition [38], [39], [40]

S3D. [42] S3D (Separable 3D Convolutional Neural Networks) is a variant of I3D that aims to improve computational efficiency while maintaining high performance in video understanding tasks. S3D is designed to reduce the computational cost of 3D convolutions by decomposing them into separate spatial and temporal convolutions. This reduces the number of parameters and increases efficiency without significantly sacrificing performance. By leveraging separable convolutions, S3D models can process videos faster and with less computational resources compared to traditional 3D ConvNets.

Applications in Sign Language Recognition: S3D's efficiency makes it suitable for applications where real-time processing is crucial. Its ability to handle spatio-temporal data effectively helps in capturing the dynamic nature of sign language gestures. S3D has been used in various research works involving large-scale sign language datasets, helping to achieve high accuracy in recognizing complex gestures and movements.

3D-CNN for RGB-based action recognition. In applied to learning spatial features to be included in videos, 3D-CNN is a natural expansion of 2D-CNN. Action recognition has continuously made use of it [12] [2]. Due to its numerous parameters, 3D-CNN requires an enormous number of videos to learn suitable representation. After I3D [1], 3D-CNN has emerged as the mainstream method for action recognition. Subsequently, the action recognition community presented various advanced 3D-CNN architectures [13], [14], [15], [16], which exceed I3D with regard to accuracy and efficiency.

GCN for skeleton-based action recognition. In skeleton-based action recognition, graph convolutional networks are frequently used [17], [18], [19], [20], [21], [7]. It represents sequences of the human skeleton as spatiotemporal graphs. A popular baseline for GCN-based techniques is ST-GCN [7], which combines interleaving temporal convolutions with spatial graph convolutions for spatiotemporal modeling. While self-attention mechanisms enhance the modeling capacity [24], [25], adjacency powering is used for multiscale modeling upon the baseline [23], [22]. Although GCN has achieved considerable success in skeleton-based action recognition, its scalability and reliability are restricted [11]. Furthermore, careful design may be required for GCN-based systems that fuse features from skeletons and other modalities [26].

CNN for skeleton-based action recognition. Convolutional neural networks are used in another line of research for skeleton-based action recognition. 2D-CNN-based methods start by modeling skeleton sequences using manually produced modifications to create a pseudo-image. Heatmaps are aggregated along the time dimension in one line of work [27], into a 2D input with color encodings, or learned modules [28], [29]. Despite meticulous design, information loss still happens during aggregation, resulting in subpar recognition performance. By using transformations, the coordinates in a skeleton sequence can be immediately converted to a pseudo image in other works [30], [31], [32], [33], [34]. This usually results in a 2D input with the shape K ×T, where K represents the number of joints and T is its temporal length.

These approaches are not as competitive as GCN on widely used benchmarks because such input cannot make use of the locality nature of convolution networks [30]. 3D-CNNs have only been used in a small number of earlier studies for skeleton-based action recognition. They either directly sum up

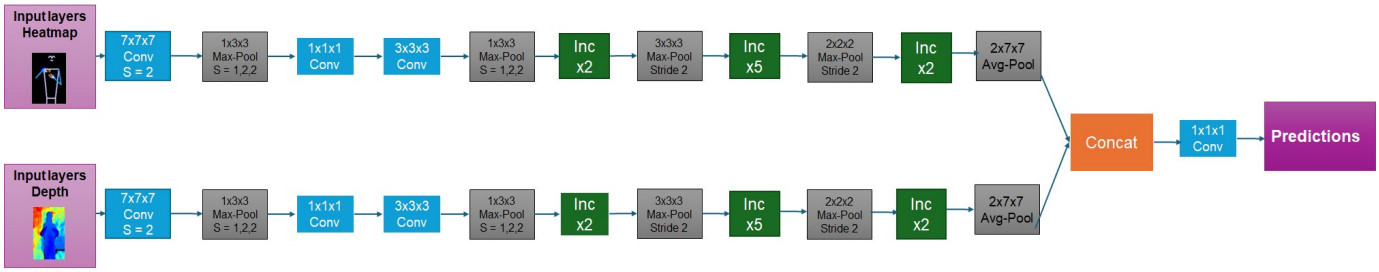

Fig. 2. Ad2C framework

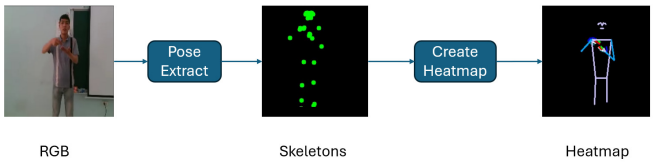

Fig. 3. The process of creating heat maps from RGB images.

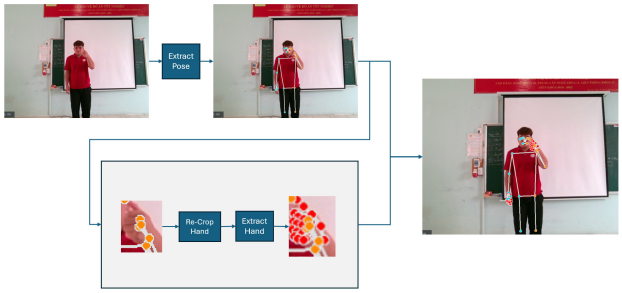

Fig. 4. Overlay joint or limb heatmaps along the time dimension and apply preprocessing that creates the heatmap volume.

the 3D skeletons into cuboids [37] or stack the pseudo images of distance matrices [36], [35] to create the 3D input. These methods likewise experience significant information loss and achieve performance that is much below the state-of-the-art.

III. PROPOSED METHOD

A. Ad2C architecture

We propose Ad2C structural designs, an overview of the Ad2C architecture is depicted in (Figure 2). Ad2C is the approach of a 3D-CNN for action recognition, which involves improving and combining two CNN networks based on the I3D structure. We further improved the model by splitting the Heat Map and Depth data into two separate training streams. Each stream is passed through Conv3D layers to extract features before combining. The combination of information can then be fed into another Conv3D layer for further processing to enhance the feature combination from the previous layer or no processing may be required. Additionally, we also train a 3D-CNN enhancement that applies a combination of Heatmap and Depth data without going through two streams. However, this test gives results that are not as feasible as the previous method. The variant method combines two CNN networks with Heatmap and Depth input layers to demonstrate the flexible interaction capabilities of Ad2C.

B. Training Objective

We start by looking at skeleton extraction, which is the basis of skeleton-based action recognition. We point out some

aspects that need to be considered such as choosing a skeleton extraction tool and promoting using 2D skeletons. Therefore, we introduce the Heatmap Volume (Figure 3), representing the 2D skeleton sequences used in the training model. In terms of storing estimated heatmaps, they are typically saved as a set of three coordinates (x;y;c), where c marks the maximum point of the heatmap and (x;y) is the coordinate corresponding to c. In our experiment, we found that the triple coordinates (x; y; c) save a large amount of storage space without sacrificing performance.

The training objective is a multi-task optimization problem. Let s_i be the maximum IoU score between the i-th default span and any ground truth span, and let $x_{ij}^k = \{1, 0\}$ be an indicator for matching the i-th default span to the j-th ground truth span of category $k \in [1, K]$. The total objective loss function consists of three weighted components: the localization loss (loc), the class confidence loss (conf), and the activity confidence loss (act):

$$\text{Loss} = L_{loc}(x, t, g) + \alpha L_{conf}(x, c) + \beta L_{act}(s, c) \quad (1)$$

where α and β are the weight terms used to balance each component of the loss function, the localization loss is a Smooth L1 loss between the predicted temporal offsets (t) and the ground truth span parameters (g). The class confidence loss (c) is a softmax loss over multiple class confidences. The activity confidence loss is a binary classification loss using sigmoid cross-entropy.

C. Proposed setting

- *Data splitting and preprocessing*

Data splitting: We present an analysis of a dataset from our collection, reflecting real-world scenarios. This dataset includes 120 isolation signs, each performed by 20 volunteers, each volunteer performing 10 times per sign. For the Ad2C model, we split the dataset into three sets: training, validation, and testing, based on the 20 signers for a sign. This process ends with a total of 16x120 signers for training, 2x120 signers

for validation, and 2x120 signers for testing; this corresponds to 80% for training, 10% for validation, and 10% for testing. The number of videos per set is as follows: the training set has 18948 videos, the validation set has 2369 videos, and the testing set has 2369 videos. The dataset is divided in this manner for both RGB and depth data.

Preprocessing:

Heatmap inputs: Use RGB data to convert into Heatmaps through skeleton extraction. In this way, we have eliminated environmental factors such as lighting, background, or blind spots in the image.

First, We resize the frame to 250 x 250 pixels and apply a center crop to 224 x 224 pixels. We resize the frame to 250 x 250 pixels and apply a center crop to 224 x 224 pixels. For data augmentation, we apply random rotations between -15 degrees and 15 degrees and adjust brightness by up to 10%. Additionally, the data undergoes horizontal and vertical translation. This augmentation is suitable because, in real-world scenarios, people may stand off center, and the camera setup may experience rotations within this range.

Then extracting the skeleton, we use the Holistic Pose method of media pipe. The two extraction stages are shown in Figure 4:

- Stage 1: preliminary extraction of the body and face skeleton.
- Stage 2: focus on each area of the hands.

Because the size of the body and face is large, we can detect it directly from large images. On the contrary, the fingers in the hand are much smaller but the information about the movements is very huge. Therefore, we will approximately cut the hand based on the previously detected wrist points and include them in the hand detection model for more accurate extraction. There will be 52 points in total

- 32 face and body points
- 20 finger points

Heatmap is created from line segments with skeleton points corresponding to accompanying colors. Heatmaps help the model learn better and more naturally with spatial information like RGB and the same skeleton- avoiding being influenced by the environment's same skeletons. Sign language recognition is greatly influenced by the fingers and wrist. Therefore, we use the colors of different wrist and finger segments, including 5 creating colors: green, white, dark blue, red, and light green. To better represent the data about the relative position of the hand and body parts, we used light purple for the body and blue for the arms. The head is quite far from the body, so we chose a color similar to the body color. The foot part carries no information, so it can be omitted.

Depth inputs: To enable the integration of Depth data with Heatmap data, resizing to 224 x 224 is necessary. Given that Depth data values range up to 1300 (equivalent to 130 cm), the process involves normalization and resizing. Initially, the Depth data is normalized to the range of 0-1. Subsequently, to fit within the 0-255 range typical for images, the values are scaled by multiplying by 255. Following this normalization step, standard resizing techniques are applied. Once the Depth data has been resized to the desired dimensions, normalization to the range of 0-1 is necessary. This method ensures that the depth values retain meaningful depth information throughout the process.

Time augmentation: In practical scenarios, sign language practitioners do not consistently execute gestures at standardized speeds. Thus, the input data necessitates speed variation achieved through random frame removal or repetition, while maintaining chronological order. This approach ensures the dataset's realism and enhances its robustness for model training. Furthermore, in practical scenarios, gestures are sometimes not fully executed to swiftly express language concepts. To equip the model with the ability to detect signs even in such instances, we initially segmented the video into seven parts, randomly selecting start indices from the first three segments and end indices from the last three. The rationale behind this data augmentation strategy stems from the observation that signers often abbreviate the beginning or end of gestures to increase communication speed while ensuring that the intended meaning remains comprehensible to the interlocutor. Before temporal data augmentation, Heatmap and Depth data are combined to ensure that each frame corresponds to its respective pair.

IV. EXPERIMENTAL RESULTS

A. Experiment

As illustrated in the proposed settings, different colors are assigned to various wrist and finger segments for clarity. These include green, white, dark blue, red, and light blue. To accurately depict the relative positions of hands and body parts, light purple is used for the body and blue for the arms. Given that the head is positioned relatively far from the body, a color similar to the body color is chosen. The footer part does not convey any significant information and can be disregarded. The outcomes of these settings are shown in Figure 5.

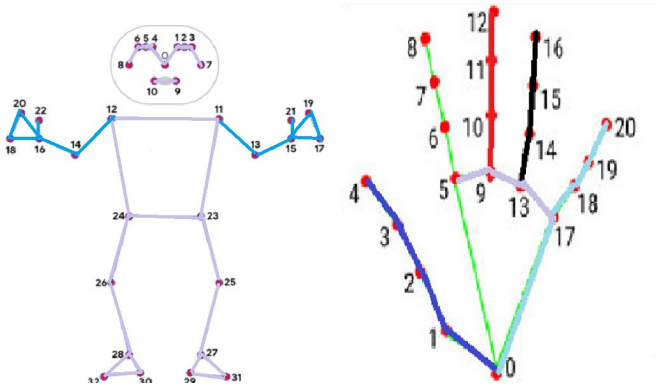

Fig. 5. Heatmap line segments with multiple colors

To achieve effective model training, we employed the RTX 3090 24GB hardware platform, utilizing a batch size of 80 and executing the process for 300 epochs. We also implemented Weight Decay with a rate of 0.00005 and Dropout with a

value of 0.3 in the final Conv3D layer to mitigate overfitting. Additionally, each training sample comprised 72 frames. Due to the significantly larger parameter count, we initially disabled the Heatmap branch to train the Depth branch with a learning rate of 0.001 over 100 epochs. Subsequently, we enabled all layers of the model and continued training with a reduced learning rate below 0.0005.

When testing improvements in two streams of I3D for our Ad2C model, we tested the use of an Early Fusion model (which combines Heatmap and Depth data immediately after the input layers). With this early merging, training only takes place on one 3D-Conv stream. We used this model to pre-train the Kinetics dataset and fine-tune all its layers. For both Mid-Fusion (Ad2C architecture) and Late-Fusion (combines two streams at the end), we also used the Kinetics dataset to conduct training for both branches of 3D-Conv and fine-tune its inner layers. For these experiments, we proposed the improved model Ad2C as Mid-Fusion and started the experiments with our Heatmap-Depth dataset.

Finally, we start testing Ad2C and compare the training results of Ad2C on different input data types of our dataset. We tested Ad2C with Heatmap-Depth and RGB-D inputs. The results show that Heatmap-Depth input has better accuracy than RGB-D on the Ad2C model. To further demonstrate the superiority of Ad2C, we continue to compare the accuracy of the Ad2C model with the original I3D on the same RGB-D input data type. Because the original I3D only has 1 stream, to be able to use RGB-D, we must use the early fusion method (after the input layer). We also tested I3D with discrete data types including RGB and Heatmap. We presented and analyzed the test results in detail below.

B. Results

Table 2 presents the performance comparison of our dataset. From the table, it is clear that the model performance varies significantly depending on the input data type and the model type.

TABLE 2
MODEL PERFORMANCE

Model	Input	Accuracy	mAP	Validation Loss
I3D	RGB	59.35%	0.581	1.65
	Heatmap	53.33%	0.548	1.85
	RGB-D	64.21%	0.633	1.52
Ad2C	RGB-D	79.80%	0.804	0.77
	Heatmap- Depth	**80.15%**	**0.825**	**0.74**

When the I3D model is trained with discrete data (RGB or Heatmap), accuracy is higher with RGB data (59.53%) than with Heatmap (53.33%). This shows that using RGB can higher model performance in terms of accuracy. Additionally, mAP is also quite large in RGB (0.581) compared to Heatmap (0.548), indicating better performance when used only on RGB data. However, in these cases, Validation Loss is still large, with RGB and Heatmap data being 1.65 and 1.85 respectively.

Interestingly, when the I3D model is trained with RGB-D data, combining both RGB and Depth data, the accuracy is higher (64.21%). This shows that combining different types of data can improve model performance. mAP is approximately equivalent (0.633) to that in discrete RGB (0.581), suggesting a trade-off between precision and mAP. Validation loss is still large, although it is slightly reduced compared to RGB or Heatmap data but still at the threshold of 1.52

With the advanced model Ad2C, using a combination of data types is necessary. Therefore, Ad2C only uses input layers of RGB-D or variants such as Heatmap-Depth without using separate data. Comparing Ad2C and I3D on the same RGB-D input data set, it is easy to see that Ad2C's accuracy is much better (79.80%) than (64.21%). Ad2C continues to improve when using the Heatmap-Depth input layer, and this model's accuracy continues to increase from (79.80%) to (80.15%). More specifically, in the case of using the Ad2C model, the Validation Loss of the model has decreased significantly. When Ad2C trains on the RGB-D dataset, the Loss is at 0.77 and this number continues to decrease to 0.74 when using Heatmap-Depth as the input layers

Table 3 compares different data-trained Ad2C networks regarding the number of parameters, input frame sizes, and inference time on a machine with GPU RTX 3090 24GB. From the table, we can observe that the inference time on the Ad2C network increases significantly with 72 input frames sized 224 x 224.

TABLE 3
NUMBER OF PARAMETERS, INPUT SIZES OF THE MODELS, AND INFERENCE TIME.

Network	Input data	Number of parameters	GPU inference time
I3D	RGB	12.06M	12.36(ms)
I3D	RGB-D	24.77M	51.39(ms)
Ad2C	Heatmap-Depth	25.09M	51.11(ms)

The results presented in Table 3 show that input data type and model structure can significantly impact the performance of networks on the same datasets we collected. However, there seems to be a trade-off between accuracy and mAP on RGBD and Heatmap-Depth data that results in improved performance. Further research is needed to explore these relationships in more detail and optimize model performance under different conditions.

V. CONCLUSION AND FUTURE WORK

In our research, we improve current modern recognition methods to produce better performance and quality than previously published methods based on the heatmap-depth dataset. The recognition method is suitable for device conditions, does not consume resources, and runs well on GPU and in real-time. Our experimental implementation yields good, scalable results on larger datasets. The effectiveness of the dataset and training method has been confirmed by multiple experiments.

Our next research work in the future is to continue improving training methods to achieve higher performance. This requires meticulous methodological research to deliver our model suite of the highest quality.

ACKNOWLEDGMENT

We extend our gratitude to the volunteers for their contributions to data collection and to Phenikaa University's High-Performance Computing System (PHPC) for processing the data.

REFERENCES

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In CVPR, pages 6299–6308, 2017. 1, 2, 4, 5, 9, 12, 13

[2] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In ICCV, 2015. 1, 2, 4, 10

[3] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In ECCV, 2016. 1, 3

[4] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. arXiv:1406.2199, 2014. 1

[5] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. arXiv:2001.08740, 2020. 1

[6] Philippe Weinzaepfel and Gregory Rogez. Mimetics: To- ´ wards understanding human actions out of context. IJCV, 129(5):1675–1690, 2021. 1

[7] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In AAAI, volume 32, 2018. 1, 2, 3, 5, 7, 14

[8] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In CVPR, pages 1110–1118, 2015. 1

[9] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In CVPR, pages 588–595, 2014. 1

[10] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In CVPR, pages 1290–1297. IEEE, 2012. 1

[11] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In KDD, pages 1399–1407, 2019. 2

[12] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. TPAMI, 35(1):221–231, 2012. 2

[13] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In CVPR, pages 203–213, 2020. 2, 4, 10

[14] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In ICCV, pages 6202–6211, 2019. 2, 3, 4, 5, 10, 13

[15] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In ICCV, pages 5552–5561, 2019. 2

[16] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In CVPR, pages 6450– 6459, 2018. 2

[17] Jinmiao Cai, Nianjuan Jiang, Xiaoguang Han, Kui Jia, and Jiangbo Lu. Jolo-gcn: mining joint-centered light-weight information for skeleton-based action recognition. In WACV, pages 2735–2744, 2021. 2

[18] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In ICCV, pages 13359–13368, 2021. 2

[19] Pranay Gupta, Anirudh Thatipelli, Aditya Aggarwal, Shubh Maheshwari, Neel Trivedi, Sourav Das, and Ravi Kiran Sarvadevabhatla. Quo vadis, skeleton action recognition? IJCV, 129(7):2097–2112, 2021. 2

[20] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In MM, pages 1625–1633, 2020. 2

[21] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Constructing stronger and faster baselines for skeleton-based action recognition. arXiv:2106.15125, 2021. 2

[22] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In CVPR, 2019. 2, 7

[23] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In CVPR, 2020. 2, 5, 7, 11, 13

[24] Bin Li, Xi Li, Zhongfei Zhang, and Fei Wu. Spatio-temporal graph routing for skeleton-based action recognition. In AAAI, volume 33, pages 8561–8568, 2019. 2

[25] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Twostream adaptive graph convolutional networks for skeletonbased action recognition. In CVPR, pages 12026–12035, 2019. 2, 7, 8

[26] Srijan Das, Saurav Sharma, Rui Dai, Francois Bremond, and Monique Thonnat. Vpn: Learning video-pose embedding for activities of daily living. In ECCV, pages 72–90. Springer, 2020. 2

[27] Vasileios Choutas, Philippe Weinzaepfel, Jer ´ ome Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In CVPR, pages 7024–7033, 2018. 2, 8, 12, 13

[28] Sadjad Asghari-Esfeden, Mario Sznaier, and Octavia Camps. Dynamic motion representation for human action recognition. In WACV, pages 557–566, 2020. 2

[29] An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. Pa3d: Poseaction 3d machine for video recognition. In CVPR, pages 7922–7931, 2019. 2, 8, 12, 13

[30] Carlos Caetano, Jessica Sena, Franc¸ois Bremond, Jefers- ´ son A Dos Santos, and William Robson Schwartz. Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. In AVSS, pages 1–8. IEEE, 2019. 2

[31] Hamid Reza Vaezi Joze, Amirreza Shaban, Michael L Iuzzolino, and Kazuhito Koishida. Mmtm: Multimodal transfer module for cnn fusion. In CVPR, pages 13289–13299, 2020. 2

[32] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In CVPR, 2017. 2

[33] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Cooccurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. arXiv:1804.06055, 2018. 2

[34] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In CVPR, pages 5137–5146, 2018. 2

[35] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861, 2017. 12

[36] Zeyi Lin, Wei Zhang, Xiaoming Deng, Cuixia Ma, and Hongan Wang. Image-based pose representation for action recognition and hand gesture recognition. In FG, pages 532– 539. IEEE, 2020. 2

[37] Hong Liu, Juanhui Tu, and Mengyuan Liu. Two-stream 3d convolutional neural network for skeleton-based action recognition. arXiv:1705.08106, 2017. 2

[38] Wong, Ryan, Necati Cihan Camgöz, and Richard Bowden. "Hierarchical i3d for sign spotting." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022.

[39] Maruyama, M., et al. "Word-level sign language recognition with multi-stream neural networks focusing on local regions. arXiv 2021." arXiv preprint arXiv:2106.15989.

[40] Zhu, Yu, and Tiantian Yuan. "Continuous Sign Language Recognition Based on I3D-TCP Network Structure." 2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE). IEEE, 2023.

[41] Carreira, Joao, and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset." proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[42] Zhang, Da, et al. "S3d: single shot multi-span detector via fully 3d convolutional networks." arXiv preprint arXiv:1807.08069 (2018).