Real-time HGR using Handcrafted Descriptors and Late Fusion Transformers for Computer Interaction

Christian Dave T. Navesis
University of the Philippines Diliman
Quezon City, Philippines
christian.dave.navesis@eee.upd.edu.ph

Kier R. Quiambao
University of the Philippines Diliman
Quezon City, Philippines
kier.quiambao@eee.upd.edu.ph

Rhandley D. Cajote
University of the Philippines Diliman
Quezon City, Philippines
rhandley.cajote@eee.upd.edu.ph

Abstract—**In this paper we develop a vision-based Hand Gesture Recognition (HGR) system for real-time computer mouse control by leveraging alternate deep learning techniques, post-processing methods, and dataset refinements. The goal is to enhance classification accuracy, reliability, and system usability for real-time applications. By integrating Google's Mediapipe Hands for hand tracking and incorporating dynamic gesture recognition techniques, the system extracts raw skeleton data for two main classifiers: a static hand pose MLP classifier for precise frame-by-frame classification and a dynamic gestures classifier using late fusion temporal transformers window prediction of dynamic gestures mapped to computer mouse macro actions. The system achieved an overall accuracy 85.86% and a Levenshtein accuracy of 73.86%, indicating an optimized, more efficient, and reliable HGR system that offers improved functionality and enhanced usability in practical applications.**

Index Terms—**vision-based hand gesture recognition, multilayer perceptron, temporal transformers**

I. Introduction

Recognizing hand gestures in real time is crucial for creating intuitive and natural interfaces—especially in contexts where gestures may be the most practical or even the only viable form of interaction. This is particularly relevant in scenarios such as sign language for the deaf [1], communication for the elderly [2], and interaction in sterile medical environments, such as for surgeons needing to signal or access digital information while performing operations [3]. These examples underscore the diverse applications and significant motivations for developing a real-time dynamic hand gesture recognition (HGR) system. While sensor-based approaches using accelerometers, gyroscopes, and ECG sensors [4] have shown promise, the need to attach hardware to the user's body makes them cumbersome and less intuitive. In contrast, vision-based HGR systems—which use only a standard camera input—enable more natural, contactless interaction without requiring any wearable devices.

Beyond medical and accessibility use cases, real-time gesture recognition has broad applicability in professional environments where traditional touch- or voice-based input may be limited or impractical. For instance, mechanics and technicians with gloved or oil-covered hands may need to consult diagrams or control systems without contaminating surfaces. In cleanrooms, biosafety labs, or manufacturing plants, maintaining sterility or avoiding cross-contamination similarly precludes physical interaction with input devices. In emergency response, construction, or field operations, protective equipment and noisy environments make voice commands unreliable. In all these settings, gesture-based interfaces offer a hygienic, silent, and intuitive interaction modality that preserves workflow continuity.

Despite the promise shown by deep learning-based HGR systems, developing robust, real-time gesture recognition remains a significant challenge. Many existing models are designed for offline classification and rely on large annotated datasets and computationally intensive architectures, such as CNNs and RNNs. These approaches often struggle to meet the low-latency and efficiency requirements of real-time deployment, particularly on resource-constrained or edge devices. Real-time HGR systems must contend with lighting variations, occlusions, diverse hand orientations, and environmental noise, all while delivering fast and accurate performance. Achieving this requires lightweight and responsive models, robust hand tracking, and system-level resilience to misclassifications [5]. When properly implemented, such systems can enable inclusive, hands-free interaction that improves productivity, accessibility, and safety across a wide range of application domains.

In this work, we propose a hybrid framework that combines hand-crafted features—which are lightweight and interpretable—with a late-fusion Transformer architecture that models spatiotemporal dependencies across feature streams. The system enables hands-free interaction through generic mouse actions and intuitive macros. Our design prioritizes real-time performance, emphasizing fast inference, lightweight classifiers, and responsiveness, while maintaining tolerance to misclassifications. We show that such a system can serve as a practical alternative to conventional input methods across diverse professional domains.

A. Related Work

Hand gesture recognition (HGR) has long been explored as a promising interaction modality for human-computer interaction (HCI), particularly in environments where traditional inputs like touch, voice, or gaze are limited. Early systems relied on rule-based or vision-based techniques using colored gloves or fiducial markers, but these approaches were quickly superseded by deep learning models that provided superior accuracy and generalization. Convolutional neural networks (CNNs), recurrent neural networks (RNNs), and 3D CNNs have been widely used to classify gestures from RGB, depth,

979-8-3315-6946-4/25/$31.00 ©2025 IEEE

and infrared video streams. Some systems incorporate pose estimation models such as MediaPipe Hands or OpenPose to extract skeletal representations prior to classification [9].

Recent studies have explored various modalities to improve gesture recognition performance, including optical flow [6], depth [7], and infrared imaging [8]. While these modalities provide rich spatiotemporal information, they are computationally expensive and require large datasets, making them difficult to deploy on resource-constrained devices. To address this, some works have turned to skeletal data, using lightweight tracking models to extract joint coordinates, resulting in a more compact and robust representation [10]. These skeletal features are suitable for training with deep learning models such as 1D-CNNs, 3D-CNNs, and Transformer-based architectures.

1D-CNNs are effective for modeling temporal sequences of skeletal features, while 3D-CNNs capture both spatial and temporal information from gesture evolution. For example, Liu et al. [11] used 3D-CNNs to process embedded skeletal hand postures across time. However, these approaches often fall short in modeling long-term dependencies between frames. Transformers, which employ self-attention mechanisms, have recently gained traction for skeleton-based gesture and action recognition due to their ability to model temporal relationships across frames more effectively.

Plizzari et al. [10] proposed a spatial-temporal transformer that captures both intra-frame and inter-frame dependencies efficiently with fewer parameters—making them better suited for real-time applications. The network uses self-attention to understand how different body parts interact within a single frame and how these interactions evolve over time across multiple frames. Moreover, transformers can effectively achieve good performance with fewer parameters. Transformer-based systems can be lightweight, efficient, and more suitable for real-time applications with limited computational resources.

Using HGR for human-computer interaction (HCI) systems that utilize hand gestures for mouse control was done in [9]. In this study a sliding window approach based on Kopuklu's [5] with two deep learning models as a gesture detector and a gesture classifier. This simultaneous hand gesture detection and classification is implemented for mouse control. The architecture specifically used an RGB detector, MediaPipe Hands, and skeleton classifier for continuous HGR. Mouse actions are mapped to corresponding gestures from the IPN Hand dataset that are executed when the algorithm accurately detects a gesture.

Despite the high accuracy achieved by these classifiers, they are still room for improvement despite the high accuracy and any small misclassification of a gesture will affect the fluidity and performance of the HGR. The existing HGR system architecture can be further improved to be tolerant to misclassifications and is able to compromise with it. In this paper we propose an approach that utilize hand-crafted spatial features and an architecture that separates the static pose classifier and a more complex dynamic gesture classifier with late temporal fusion transformers with post-processing for real-time mouse control.

II. Methods

A. System Architecture

Figure 1 shows the system architecture of our proposed HGR system for mouse control. MediaPipe Hands tracking is utilized for detecting and tracking the user's hand and its landmarks in the video. A sliding window approach is adopted where consecutive frames of hand gestures are captured and the window updates with each new frame. The hand landmarks are extracted and represented as a series of skeleton frames which are input into a Static Pose MLP Classifier for frame-wise predictions. The handcrafted descriptors module generates the descriptors such as limb angles and joint velocities. We then determine the temporal dynamics of the hand movements and predict gestures over the window of frames. For post-processing, we apply fuzzy logic and gesture activation rules to the output of the late fusion temporal transformer. The mouse control is implemented using macros that execute basic mouse functions based on recognized gestures and poses.

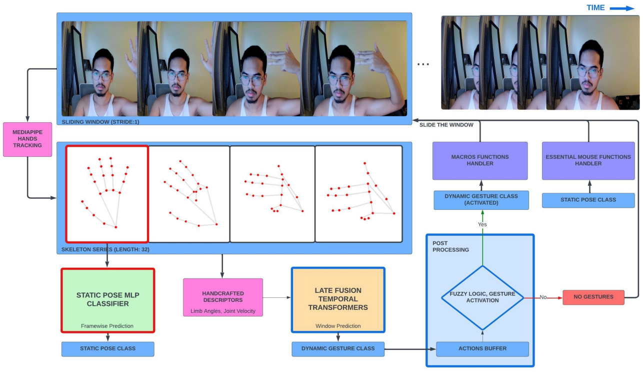
TIME
SLIDING WINDOW (STRIDE:1)
SLIDE THE WINDOW
MEDIAPIPE HANDS TRACKING
MACROS FUNCTIONS HANDLER
ESSENTIAL MOUSE FUNCTIONS HANDLER
DYNAMIC GESTURE CLASS (ACTIVATED)
STATIC POSE CLASS
Yes
SKELETON SERIES (LENGTH: 32)
POST PROCESSING
STATIC POSE MLP CLASSIFIER
Framewise Prediction
HANDCRAFTED DESCRIPTORS
Limb Angles, Joint Velocity
LATE FUSION TEMPORAL TRANSFORMERS
Window Prediction
FUZZY LOGIC, GESTURE ACTIVATION
No
NO GESTURES
STATIC POSE CLASS
DYNAMIC GESTURE CLASS
ACTIONS BUFFER

Fig. 1. System Architecture

Overall, the system processes video frame inputs to identify hand gestures and translate them into mouse control functionalities using a strategic pipeline of two classifiers. The Static Pose MLP Classifier allows the system to quickly recognize static poses. The Dynamic Gestures Classifier uses Late Fusion Temporal Transformers, which is tasked with recognizing dynamic hand gestures by analyzing sequences of hand movements over the window length. These dynamic gestures involve intricate details of the hand and long, complex motion patterns that unfold over multiple frames. Temporal transformers excel in capturing these patterns as they can model the temporal dependencies and variations in the gesture sequence. Separating this task from the static pose classifier allows the system to more accurately and reliably recognize dynamic gestures that are mapped to computer macros. By strategically separating the classifiers, the system leverages the strengths of each approaches ensuring a functional gesture-based mouse control system.

B. Static Pose MLP Classifier

The Static Pose Multilayer Perceptron (MLP) model shown in Figure 2 classifies the static hand poses from

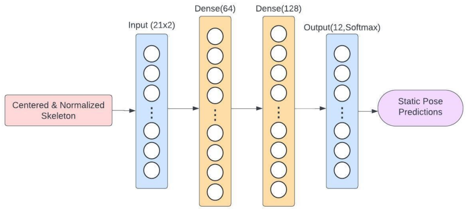

Fig. 2. Static Pose MLP Model

skeletal data. The input consists of 21 key points, each with an x- and y-coordinates giving a total of 42 features which are then centered and normalized to remove variations due to hand position in the frame and ensuring hand gestures are on a comparable scale. The center of mass translation involves adjusting the x- and y- coordinates so that the hand's centroid, is at the origin. Following translation, normalization is applied to scale the data to a standard range, [-1, 1], using mean and standard deviation normalization. This ensures that the hand posture data is consistent regardless of hand size. These preprocessing steps ensures the input to the MLP are spatially invariant. This model is also designed to be simple with a limited number of layers and inherently ensures that the model is computationally efficient with very few parameters and can execute with minimal delay during inference.

C. Handcrafted Descriptors

In this implementation, we utilize handcrafted descriptors as input to temporal transformers, capturing the essential dynamics of hand gestures while ensuring computational efficiency. The primary descriptors used are limb angles and joint velocities, which provide a detailed representation of hand movements.

1) Limb Angles: Given the position vectors of joints Pi and Pj, the limb vector L_{ij} is calculated as: $L_{ij} = P_j - P_i$. The angle θ between two limb vectors L_{ij} and L_{kl}can then be calculated by.

$$\cos\theta = \frac{\mathbf{L}_{ij} \cdot \mathbf{L}_{kl}}{|\mathbf{L}_{ij}||\mathbf{L}_{kl}|} \tag{1}$$

This descriptor also utilizes the depth-estimated z-position by MediaPipe to further enhance the recognition of dynamic actions. It is designed to be spatially invariant and effectively describes the spatial features of gestures with evolving limb angles.

2) Joint Velocity: Joint velocity calculates the speed and direction of joint movements from the given position vectors. It is calculated by subtracting the position of the same joint in the next frame (skipping the adjacent frame), effectively capturing the direction of movement.

Given the position vectors $P_i(t)$ and $P_i(t+2\Delta t)$, the joint velocity V_i is calculated as:

$$V_i = \frac{P_i(t+2\Delta t) - P_i(t)}{2\Delta t} \tag{2}$$

This descriptor is particularly useful for gestures that involve translation in space as they progress within the duration of the gesture.

Using these handcrafted descriptors, we ensure that our system captures the essential features of hand gestures while maintaining computational efficiency.

D. Dynamic Gesture Late Fusion Temporal Transformers

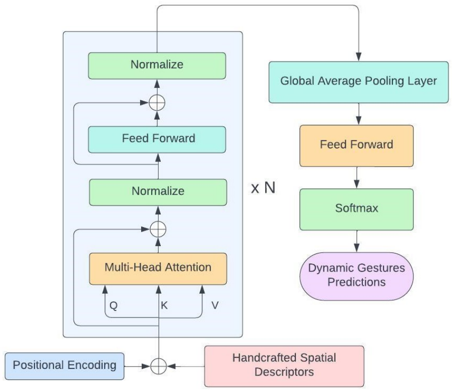

Fig. 3. Late Fusion Temporal Transformer Model

Figure 3 illustrates the Dynamic Gestures Classification Model, consisting of N-cascaded Temporal Transformer Encoder Blocks, followed by a global average pooling layer, and a classification head. The input to this model includes the handcrafted spatial descriptors discussed in the previous section.

The temporal transformer encoder block captures the temporal relationships between frames in a gesture sequence. It starts with a positional encoder that encodes the order of frames, allowing the block to process the entire sequence in parallel. Multi-head attention is then applied to learn the temporal dynamics of the handcrafted spatial descriptors, enabling the model to focus on different parts of the sequence simultaneously. To stabilize the training process and mitigate issues like vanishing or exploding gradients, normalization layers and skip connections are incorporated.

Following the transformer block, a Global Average Pooling Layer aggregates the encoded temporal features, reducing dimensionality by averaging each feature over the time dimension. This process generates a compact representation of the gesture sequence.

Lastly, the classification head, consisting of fully connected layers, maps the pooled features to the gesture classes. These layers effectively integrate the learned temporal and spatial features to accurately predict the performed gesture. By employing preprocessed descriptors as input and allowing an end-to-end transformer network to work out the temporal dynamics, the model ensures it learns the essential complex features. The use of preprocessed descriptors also reduces the required complexity of the transformer blocks, resulting in fewer parameters. This makes the model more desirable for real-time applications where low latency is essential.

E. Fuzzy Logic and Gesture Activation

Following the temporal transformers, the window predictions are then stored in an action buffer which maintains a queue of recent predictions for post-processing. Fuzzy logic is then applied where the buffer is analyzed to identify the most frequently occurring gesture class which is referred to as the dominant class. After determining the dominant class from the buffer, the logic checks the last two window predictions to ensure that they match the dominant class. If both conditions are met, the dominant class is returned as the detected gesture. In essence, this post-processing technique helps prevent misclassifications and ensures that transient frames do not trigger incorrect gesture activations.

Furthermore, the post-processing logic includes a rising edge activation mechanism for distinguishing between single-time and continuous-time gestures. For single-time gestures, the system looks for a transition from non-activation to activation or the rising edge of the dominant class. This ensures that the gesture is only recognized once per occurrence. For continuous-time gestures, the system allows sustained activation of the dominant class, enabling gestures that require prolonged input to be correctly identified and processed. Over-all the post-processing refines the gesture predictions from the temporal transformers to improve accuracy through dominant class detection in the action buffer, the fuzzy logic helps minimize misclassification of incorrect gesture.

F. Mouse Implementation

Once a gesture is successfully identified and activated after post-processing, the system translates it into corresponding mouse functionalities. This mapping process establishes how specific gestures control the mouse cursor and interact with the user interface. This approach allows for intuitive and hands-free control of the mouse cursor. Noise filtering of mouse cursor position is done to eliminate jittery cursor movements.

III. Results and Discussion

A. Metric Tests

For testing the Static Pose MLP Classifier, its performance is assessed using the confusion matrix and accuracy metrics. The best accuracy of 99.55% is achieved, his indicates that the model is highly effective in recognizing static gestures.

For the temporal transformer, the input is converted into fixed length sequences by concatenation and reshaping and used a fixed-size sliding window. Gestures of the same class are concatenated and reshaped sequences of fixed length based on the average length of all the gestures. When the fixed length exceeds the duration of the actual gesture, parts of that gesture may be repeated which causes confusion between actual repeated gestures and concatenated ones. This is evident in both the CLICK1 and CLICK2 classes which are the single and double left click mouse actions, respectively. Figure 5 shows the confusion matrix of this approach and an accuracy of 76.266% is achieved. This lower accuracy can be attributed to potential gesture repetitions. However, the approach is still essential and more suited for real-time applications to ensure more consistent and reliable predictions. Overall, the achieved accuracy suggest that the temporal transformer performs well on the dataset.

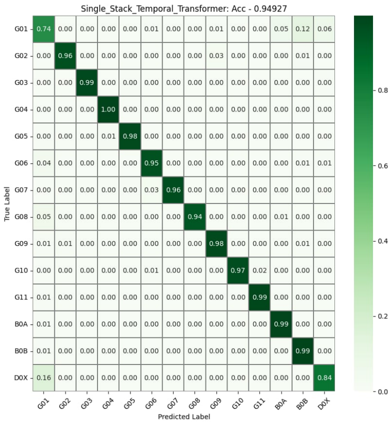

Fig. 4. Confusion Matrix for Temporal Transformer on the IPN Dataset (Dynamic Length)

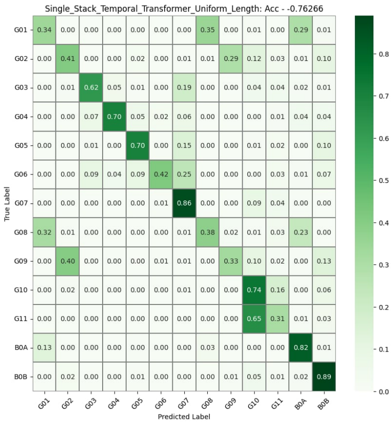

Fig. 5. Confusion Matrix of the Temporal Transformer on the IPN Dataset (Uniform Length)

It is also important to note that the dataset was also modified by applying uniform length adjustment, concatenation, and normalization, which enhanced the compactness, robustness, and real-time responsiveness of our architecture. Figure 6 shows the confusion matrix of the final system evaluated on the modified IPN hand dataset. The Levenshtein accuracy of the system is 73.86% on the modified IPN dataset. The Table summarize the precision, recall, and F1-score of the final system.

The final system achieved a high overall accuracy of 85.86%, indicating good model performance. There is some confusion in handling the gestures with misclassifications which are the 3_SWIPE_LEFT and 3_SWIPE_RIGHT classes. The rest of the gestures are recognized and accurately classified.

IV. Conclusion

In this paper, we proposed a vision-based real-time HGR system for mouse control. The developed HGR system

TABLE I
PRECISION, RECALL AND F1-SCORE ON THE MODIFIED DATASET

Class	Precision	Recall	F1-Score
OPEN	1.0	1.0	1.0
CLOSE	1.0	1.0	1.0
CLICK1	1.0	0.97	0.98
CLICK2	0.82	1.0	0.9
3_SWIPE_LEFT	0.42	0.59	0.49
3_SWIPE_RIGHT	0.41	0.18	0.25
TWIRL_OK	1.0	0.9	0.94
TWIRL_OK_HOR	0.91	1.0	0.95
FINGER_HEART	0.96	0.93	0.94
THUMBS_UP_DIAG	1.0	1.0	1.0

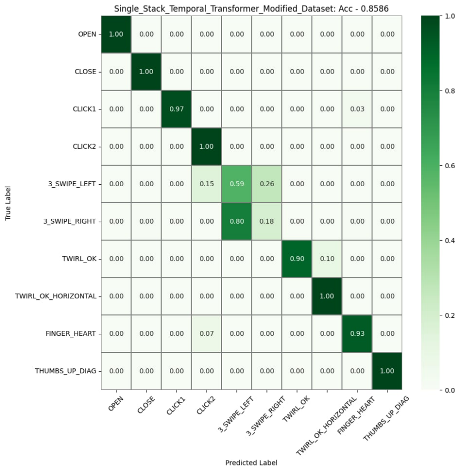

Fig. 6. Confusion Matrix of the Final System on the Modified Dataset

achieved high accuracy suitable for real-time classification of both static poses and dynamic gestures with minimal delay. The system is also robust against misclassifications within the framework, thus enhancing reliability and can be extended beyond mouse control to other command-mapping or interaction tasks across domains where non-contact interaction is essential and cover many practical applications.

Using deep learning techniques, dataset modifications, and post-processing techniques, the system achieved an overall accuracy of 85.86%. The implementation of the static pose MLP classifier with late fusion temporal transformers has enhanced the system's performance for real-time applications.

For future work, we recommend extending the gesture-based interaction beyond mouse control. The system's modular architecture and lightweight components make it well-suited for broader integration into general-purpose computer interfaces, such as mapping gestures to keyboard macros, system-level commands, or complex UI interactions. These enhancements could support richer, non-contact interaction frameworks applicable in productivity, accessibility, and assistive computing contexts.

ACKNOWLEDGMENT

The authors would like to acknowledge the Department of Science and Technology Science Education Institute (DOST-SEI) and the Engineering Research and Development for Technology (DOST-ERDT) for partially supporting the research activity.

REFERENCES

[1] G. H. Samaan et al., "Mediapipe landmarks with RNN for dynamic sign language recognition", Electronics, vol. 11, no. 19, 2022, ISSN: 2079-9292. DOI: 10.3390/electronics11193228

[2] M. Oudah, A. A;-Naji, and J. Chahl. "Elderly Care Based on Hand Gestures Using Kinect Sensor" Computers, vol. 10, no. 1, p. 5, Jan. 2021. doi: 10.3390/computers10010005.

[3] A.-R. Lee, Y. Cho, S. Jin, and N. Kim, "Enhancement of surgical hand gesture recognition using a capsule network for a contactless interface in the operating room," Comput. Methods Programs Biomed., vol. 190, p. 105385, Jul. 2020. doi: 10.1016/j.cmpb.2020.105385.

[4] S. Berman and H. Stern, "Sensors for gesture recognition systems", IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 3, pp. 277–290, 2012. DOI: 10.1109/TSMCC. 2011.2161077.

[5] O. Kopuklu, A. Gunduz, N. Kose, and G. Rigoll, "Real-time hand gesture detection and classification using convolutional neural networks", CoRR, vol. abs/1901.10323, 2019. arXiv: 1901.10323.

[6] Ortega-Avila, S., Rakova, B., Sadi, S., & Mistry, P. (2016, May). Non-invasive optical detection of hand gestures. In Proceedings of the 29th Annual ACM Symposium on User Interface Software and Technology, pp. 797-806. DOI: 10.1145/2735711.2735801.

[7] Z. Ju, Yuehui Wang, Wei Zeng, Shengyong Chen and H. Liu, "Depth and RGB image alignment for hand gesture segmentation using Kinect," 2013 International Conference on Machine Learning and Cybernetics, Tianjin, China, 2013, pp. 913-919, doi: 10.1109/ICMLC.2013.6890413.

[8] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel RealSense Stereoscopic Depth Cameras," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017, pp. 1-10. Available: https://arxiv.org/abs/1705.05548.

[9] P. G. Estavillo, D. J. R. Del Carmen and R. D. Cajote, "Vision-Based Gesture Recognition for Mouse Control," TENCON 2023 - 2023 IEEE Region 10 Conference (TENCON), Chiang Mai, Thailand, 2023, pp. 1145-1150, doi: 10.1109/TENCON58879.2023.10322367.

[10] C. Plizzari, M. Cannici, and M. Matteucci, "Skeleton-based action recognition via spatial and temporal transformer networks", Computer Vision and Image Understanding, vol. 208-209, p. 103 219, 2021, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2021.103219.

[11] J. Liu, Y. Liu, Y. Wang, V. Prinet, S. Xiang and C. Pan, "Decoupled Representation Learning for Skeleton-Based Gesture Recognition," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 5750-5759, doi: 10.1109/CVPR42600.2020.00579.

[12] Z. Qiu, T. Yao and T. Mei, "Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks," in 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017 pp. 5534-5542. doi: 10.1109/ICCV.2017.590