Fluxformer: Flow-Guided Duplex Attention Transformer via Spatio-Temporal Clustering for Action Recognition

Younggi Hong, Min Ju Kim, Isack Lee, and Seok Bong Yoo

***Abstract*—Vision transformers have demonstrated impressive performance in various robotics and automation applications, such as classification automation and action recognition. However, the drawback of transformers is their quadratic increase in computing resources with larger inputs and dependence on considerable data for training. Most action recognition models using the transformer structure rely on a few frames from the original video to reduce computation, so temporal information is compromised by low frame rates. Spatial information is also compromised by reducing the number of embeddings as the transformer layer iterates. The letter proposes a robust model for action recognition that overcomes the limitations of most action recognition models with the transformer structure using the duplex attention function, flow-guided information, RGB information, and spatial support tokens. The proposed duplex attention mechanism leverages optical flow and RGB to address the lack of temporal information. The method employs spatial interest clustering to convert input data into tokens, improving the preservation of spatial information. Finally, meaningful action event frames are extracted by analyzing the flow and clustering to distinguish scenes. The experimental results reveal that the proposed model outperforms state-of-the-art methods in action recognition accuracy.**

***Index Terms*—Computer vision for automation, recognition, visual learning.**

I. Introduction

HUMAN action recognition models are crucial in enabling robots to interact effectively with humans and meet their needs as they become more prevalent in various industries and our daily lives. They allow robots to gain valuable insight into human tasks and improve communication, workflow, safety, and task automation. However, robots have limited computing resources, so processing video data can be challenging. It is computationally infeasible to train a deep network on every video frame when the action unfolds over hundreds of frames.

Manuscript received 27 April 2023; accepted 4 August 2023. Date of publication 21 August 2023; date of current version 28 August 2023. This letter was recommended for publication by Associate Editor Y. Xiang and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported in part by the Industrial Fundamental Technology Development Program under Grant 20018699 and in part by the MOTIE of Korea and the IITP grant funded by the Korea Government (MSIT) under Grants 2021-0-02068 and RS-2023-00256629. *(Corresponding author: Seok Bong Yoo.)*

The authors are with the Department of Artificial Intelligence Convergence, Chonnam National University, Gwangju 61186, South Korea (e-mail: 216141@jnu.ac.kr; 182577@jnu.ac.kr; sackda24@jnu.ac.kr; sbyoo@jnu.ac.kr).

Our source codes with pretrained models are available at https://github.com/YGspace/Fluxformer.

Digital Object Identifier 10.1109/LRA.2023.3307285

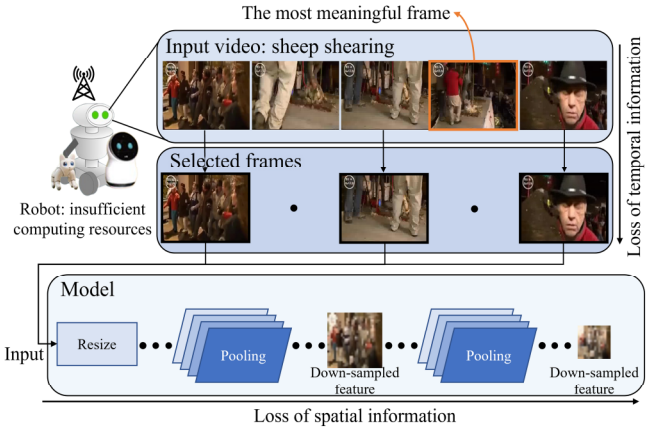

Fig. 1. Traditional frame selection-based action recognition model with the loss of spatio-temporal information.

Transformer based action recognition models demonstrate a quadratic increase in time complexity compared to traditional convolutional based models. Some models [1], [2], [3], [30], [34], [35], [36] sample only 16 frames out of a total of hundreds of frames (i.e., 300 frames), uniformly select frames, and resize input for training to mitigate this computational complexity.

However, this approach reduces the frame rate of the input dataset, which comes at a cost. This reduction can lead to a decline in temporal resolution, motion blur, and incomplete motion, hindering the accuracy of computer vision algorithms in identifying motion. In addition, the accuracy of the model may decrease due to the compromise in spatial resolution. For example, in Fig. 1 of the videos in the Kinetics 400 dataset, the mid-course, a meaningful frame for identifying action, is absent from the selected sheep shearing video frame. Additionally, low frame rates and scene changes make frames less temporally and semantically connected. Low frame rate or abrupt and large motion videos provide limited temporal information for motion pattern analysis, and the loss of temporal information can significantly affect action recognition. Furthermore, in managing the complexity of transformer based models, pooling methods and resizing data are often used to decrease their size with each pass through a transformer. The downsampling that occurs for this reason corrupts spatial information and can make the model unable to distinguish between features.

In this letter, we propose a novel model, the flow-guided duplex attention transformer (Fluxformer), designed to recognize actions by extracting meaningful action event frames

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

while effectively supplementing spatio-temporal information. We summarize the contributions as follows:

- We propose incorporating temporal information from flow into the model using the suggested duplex attention method.
- We propose a meaningful action event extractor (MAEE) method to enhance the accuracy of action recognition in the model.
- To maintain the spatial information of the input data, we propose tokenizing the input data through clustering and incorporating the data into the final transformer layer.

II. Related Works

A. Convolutional Neural Network Based Action Recognition

Convolutional neural network (CNN) based models incorporate downsampling, shift invariance, and weight sharing, which are de facto standard backbones for computer vision tasks for images [4], [5], [6], [7], [8], [9], [10], [11] and videos [12], [13], [14], [15], [16], [17], [18]. The video task focuses on feature extraction for spatial dimensions. Using a two dimensional (2D) CNN in action recognition, Stergiou et al. [19] used a pooling layer to reduce the size of the activation map. Liu et al. [20] proposed a temporal adaptive module to learn temporal information through a two level adaptive scheme. In this case of using the 2D CNN, there is strength in the spatial dimensions, but the temporal dimensions are weak.

To complement this 2D CNN based approach, optical flow models and 3D CNN have been developed. Carreira et al. [1] proposed the new two-stream inflated 3D CNN (I3D), which builds on the inflation of 2D CNN. Tran et al. [21] proposed R(2+1)D, which increased accuracy by factoring the 3D convolutional filter into separate spatial and temporal components. Piergiovanni et al. [22] proposed capturing flows within the convolutional network to learn the optical flow and weight of the CNN end to end. Khalid et al. [23] suggested a three stream method to use spatio-temporal information with a 2D and 3D CNN. Moreover, Duan et al. [24] used a 2D human skeleton as input to extract a spatial feature for action and proposed the PoseC3D model using 3D CNN architecture. However, the CNN based model has extensive computation for temporal dimensions in video tasks and limited reception field problems.

B. Vision Transformer Based Action Recognition

The study has recently commenced by changing from a CNN based backbone to a vision transformer (ViT) [25] based backbone [26], [27], [28], [29], [30], [31], [32], [33]. The pure transformer based model, ViViT [34], is a video transformer model using a pretrained ViT backbone. ViViT extracts the spatio-temporal token and adds it to the transformer layer. The MViT [35] model has a multiscale structure that uses the hierarchical structure in the base transformer with the pooling layer. The video swin transformer [36] applies SwinTransformer [37] to videos and extends its 2D window mechanism to 3D version to obtain the temporal information for the video task. MViT and video swin transformer are hierarchical based models that pretrain large scale images for spatial dimensions, but temporal information is vital in video tasks. Bidirectional encoder representations from transformers (BERT) [38] is a deep interactive representable model for unlabeled text by jointly conditioning on both the left and right contexts of all layers. BERT pretraining of video transformers (BEVT) [39] uses BERT and decouples video representation learning into spatial representation learning and temporal dynamics learning. Further, BERT masks some words in a sentence and predicts tokens using unmasked and masked tokens as inputs to the model. Based on this, BEVT uses masked tokens with two inputs: divided video and images. It balances spatio-temporal feature extraction and sharing weights. Yan et al. [40] proposed multiview transformers for video recognition using the cross view fusion of encoders separated into each connected view. For the action recognition task, information on the movement of objects is critical, and for this purpose, a study combined clustering and the transformer. Li et al. [41] proposed GroupFormer as a clustered attention mechanism. It obtains spatio-temporal information by grouping clustered spatial transformer blocks from the input video clip and a nonclustered temporal transformer block. Chen et al. [42] proposed a multimodal video transformer (MM-ViT) for video action recognition by utilizing various block based compressed features and audio waveforms. Sun et al. [43] proposed a windowed and linear transformer (WLiT) for efficient video action recognition by combining spatial windowed attention and linear attention. Li et al. [44] proposed a shrinking temporal attention transducer (STAT) that efficiently builds a spatiotemporal attention map by taking into account the decay of spatial attention in short and long temporal sequences. Existing models [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41] have faced challenges in balancing computational complexity and preserving temporal and spatial information. MM-ViT [42], WLiT [43], and STAT [44] is proposed to reduce computational complexity using multimodality compression, spatial windows attention, and temporal attention. Similarly, our proposed method uses real-time flow estimation, attention-based pooling, and attention-based clustering to reduce complexity and supplement spatio-temporal information.

C. Frame Selection Method

Real world videos are often too large for deep learning models to handle due to their massive number of frames. As a result, current training methods typically use subsampled video frames to overcome this limitation. However, fixed frame selection strategies are not optimal because they may not capture the essential information in the video.

More advanced approaches [45], [46], [47], such as adaptive frame selection, learned frame selection, and clip selection, have been proposed in the literature to address this challenge. However, these methods necessitate pre-training to calculate and sample probability values on a frame-by-frame basis. Such methods are specialized to the training domain, resulting in poor generalization performance. To overcome this limitation, we propose a novel approach that allows the network to extract meaningful frames from the video. Our proposed MAEE does not require pre-training and instead employs flow-based scene change detection and the k-means algorithm to extract meaningful frames from a temporal perspective.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

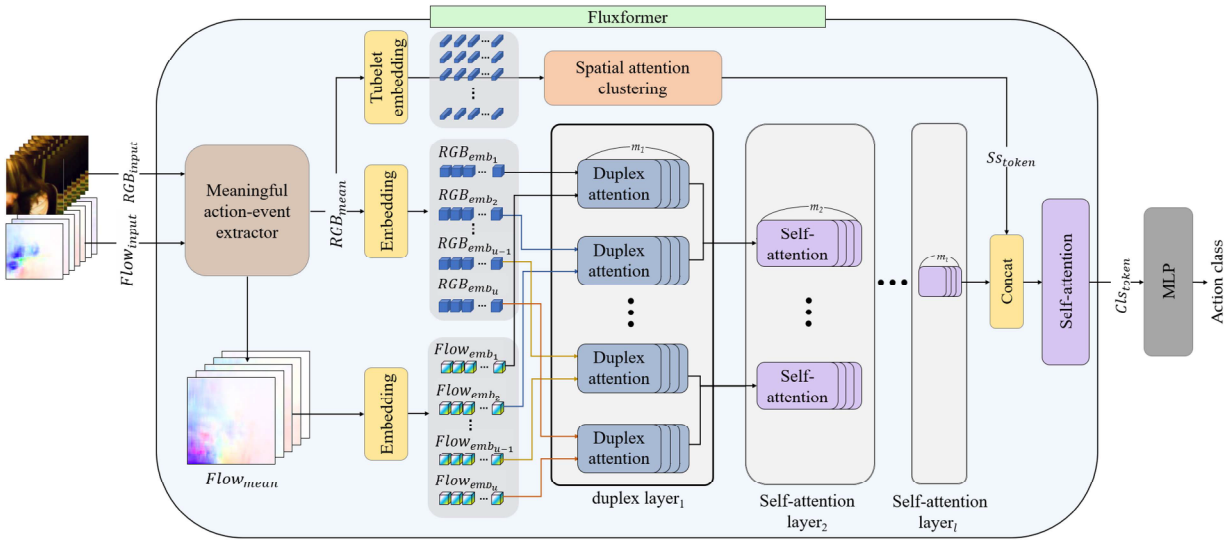

Fig. 2. Overall architecture of Fluxformer.

III. Method

A. Overview

Fig. 2 illustrates the comprehensive block diagram of the proposed Fluxformer. The MAEE block receives RGB and estimated flow as input, extracts meaningful action event frames, and forwards RGB_{mean} and $Flow_{mean}$ to their corresponding embedding blocks. The embedding block transforms RGB_{emb} and $Flow_{emb}$, which are input into the duplex attention block for each embedding order. The duplex attention operation is executed on RGB_{emb} and $Flow_{emb}$ concurrently. The duplex attention uses flow to supplement and incorporate temporal information critical to action recognition into the model. The pooling function to decrease the number of embeddings is present at the end of each attention block. This process is repeated for l layers. The top RGB_{mean} is embedded through tubelet embedding and input into the spatial attention clustering (SAC) block. It outputs spatial support tokens (Ss_{token}), which are the same size as the model embedding depth. The Ss_{token} is concatenated to the last position of the final layer output. Subsequently, it passes through the standard self-attention layer and produces a Cls_{token}. Finally, the Cls_{token} is used as an input of the multilayer perceptron (MLP) to recognize the action class.

B. Meaningful Action Event Extractor

In many action recognition applications, some video frames are uncorrelated to classes, and general action classes recognition models can be negatively influenced by these frames. In addition, traditional heuristic frame selection can lead to the loss of temporal information about the action in the original data. Therefore, we propose an MAEE to refine the data entering the action recognition model to improve accuracy. As illustrated in Fig. 3, the MAEE model consists of two main components: scene change detection and temporal clustering.

1) Flow Based Scene Change Detection: We used optical flow to calculate the flow value between two RGB frames to identify meaningful action events in a video frame. The flow value is used to identify the regions in the video frame where a scene change has occurred, with larger flow values indicating more significant changes. We selected the region with the highest flow value as the part of the frame where the scene change occurred, focusing on the most crucial part of the video frame for action recognition.

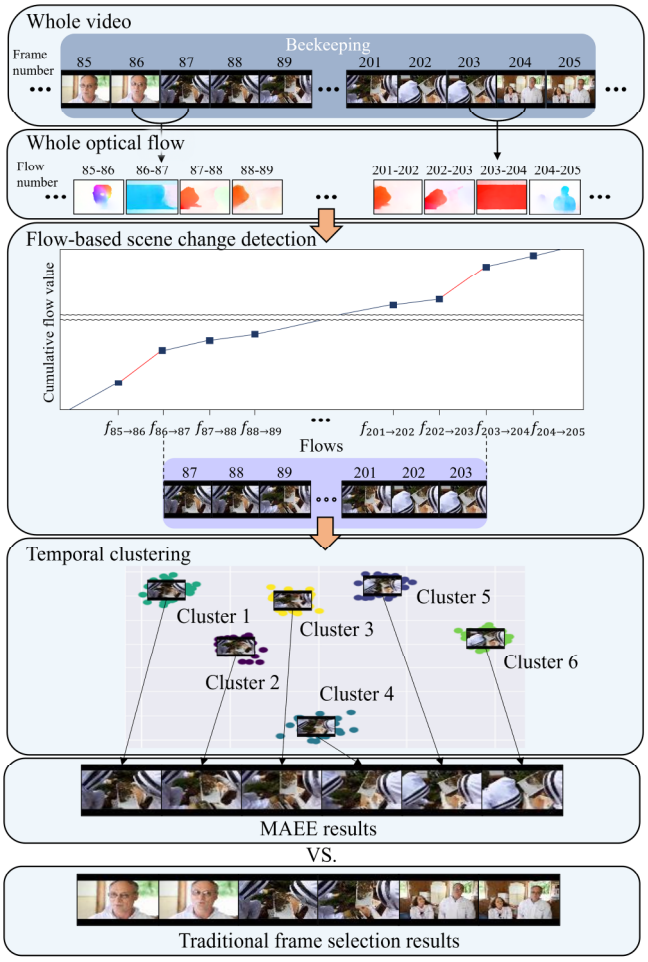

Fig. 3. Architecture of the meaningful action event extractor (MAEE) block.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Using this approach, we can extract the most meaningful information from the video frame and use it to improve the accuracy of the action recognition model. The optical flow technique accurately identifies scene changes, and selecting the region with the highest flow value ensures that we focus on the most relevant part of the video frame.

2) Temporal Clustering: To preserve the temporal information in video frames, we applied a temporal clustering technique to select frames that capture the order of actions. We reduced the dimensionality of the high dimensional video frames using the principal component analysis technique. This technique allows the most critical frame features to be retained while reducing the overall size of the data. Next, we used K-means [48] clustering to group similar frames. The K-means algorithm can be mathematically expressed as follows:

1) Initialization: Randomly select K cluster centers $m_1, m_2, \ldots, m_k$.

2) Assign each data point x_i to the closest cluster center m_j: $argmin_{j=1\ldots k}||x_i - m_j||^2$.

3) Update each cluster center m_j as the mean of all data points assigned to it: $m_j = \frac{1}{|S_j|}\sum_{x_i \in S_j} x_i$. S_j represents the set of data points assigned to the j-th cluster. $|\cdot|$ denotes the number of data points.

4) Repeat Steps 2-3 until cluster assignments no longer change.

This clustering approach helps identify similar frames in terms of the underlying features that are likely to capture the same action or movement. Then, we chose an equal number of frames from each cluster as input for the action recognition model. By selecting frames this way, we retain the temporal information in the video and ensure that the action recognition models are trained on frames that accurately capture the order of actions. This approach represents a significant improvement over traditional sampling methods, which may miss crucial temporal information, resulting in lower accuracy for action recognition.

C. Duplex Attention

The proposed model used transformers incorporating the pooling capabilities of MViT to reduce spatial dimensionality, retain important information, and minimize computational complexity and overfitting. According to [49], most ViT-based models such as [30], [34], [35], [36] using traditional frame selection methods typically sample frames individually, prioritizing per-frame importance prediction without considering frame interaction. These methods frequently encounter difficulties in accurately discerning the relationships between frames due to the possible loss of temporal data, so additional temporal information needs to be included. To address this challenge, we applied a novel method called the duplex attention method, which uses optical flow information to identify frame correlations and constrain the transformer. The experiments also demonstrate that integrating flow data into the proposed model can enhance the performance of the spatial attention mechanism. Fig. 4 depicts the architecture of the duplex attention. Both RGB_{emb} and $Flow_{emb}$ are processed through a normalization layer. Subsequently, RGB_{emb} is separated into the query (Q), key (K), and value (V), which are directed at the self-attention mechanism. The multihead self-attention is the equation detailed as follows:

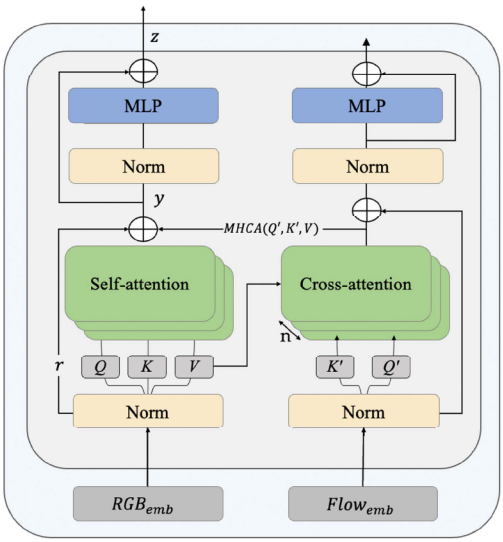

Fig. 4. Architecture of duplex attention.

$$MHSA(Q,K,V) = Concat(head_1 \ldots, head_n)W^O, \quad (1)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V), \quad (2)$$

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{D_h}}\right)V, \quad (3)$$

where W^O, W_i^Q, W_i^K, and W_i^V denote in the projection parameter matrices, n denotes the number of heads, D_h signifies the specific inner dimension for each layer, and $head_i$ represents the self-attention value of each head. Adding them together results in MHSA(Q,K,V), the multihead attention value. In addition, $Flow_{emb}$ is partitioned into Q', K'. In contrast, Q', K', and V are funneled into the cross-attention mechanism. The multihead cross-attention equation follows:

$$MHCA(Q',K',V) = Concat(Chead_1 \ldots, Chead_n)W^O, \quad (4)$$

$$Chead_i = Attention(Q'W_i^{Q'}, K'W_i^{K'}, VW_i^V), \quad (5)$$

where $Chead_i$ represents the cross-attention value of each head. The self-attention data are subsequently integrated with the original input as a residual r, combining the cross-attention information. The combined equation follows:

$$y = MHSA(Q,K,V) + MHCA(Q',K',V) + r. \quad (6)$$

Finally, y is normalized using a normalization layer and output through an MLP.

$$z = MLP(Norm(y)) + y. \quad (7)$$

As illustrated in Fig. 2, the procedure is executed m times on a single layer, using pooling in the final iteration to minimize the number of output embeddings. The condensed embeddings are subsequently passed to the following layer, reiterating this process for the designated number of layers (l). By combining two attentions: self-attention and cross-attention, duplex attention can supplement the model with the missing temporal information. It creates a more comprehensive representation of actions, improving action recognition accuracy.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

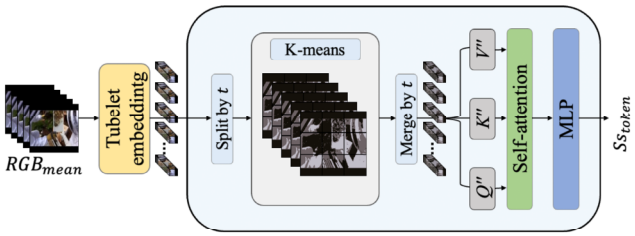

Fig. 5. Architecture of the spatial attention clustering (SAC) block.

D. Spatial Attention Clustering

In the action recognition model, pooling reduces the data size and retains temporal information but may lose crucial spatial details, affecting model accuracy. To address this, we used skip connections, tubelet embedding, and K-means clustering to preserve spatial and temporal information. Self-attention and MLP layers further refine and extract relevant features. The SAC process is visually illustrated in Fig. 5. To perform tubelet embedding, we processed the frame into a tubelet using a 3D CNN to capture both spatial and temporal information. The numbers of token's dimentions for the tubelet embedding is detailed as follows:

$$N_t = \left\lfloor \frac{T}{t} \right\rfloor, \; N_h = \left\lfloor \frac{H}{h} \right\rfloor, \; N_w = \left\lfloor \frac{W}{w} \right\rfloor. \tag{8}$$

The tube dimensions are $t \times h \times w$, where T, H, and W represent the number of frame, height, and width. We applied the floor operator $\lfloor \cdot \rfloor$ to obtain an integer outcome. The output embeddings are flattened and projected onto a lower dimensional space. After converting the original dimensions to a compact format, we performed clustering based on parameter k. The clustered information is concatenated with zero dimensional locations and undergoes multihead self-attention processing for further refinement. The equation for the self-attention follows:

$$softmax\left(\frac{Q''K''^{T}}{\sqrt{D_h}}\right)V''. \tag{9}$$

Finally, we altered the dimensions using an MLP process, generating an Ss_{token} with D_h dimensions. This Ss_{token} is concatenated with the last position of the output embedding of the final layer, preserving and integrating the spatial information into the model processing pipeline. It ensures the preservation of spatial and temporal information, whereas the self-attention and MLP layers refine and extract essential features.

IV. Experiment

This section briefly describes the benchmark dataset used to evaluate the model and report the experimental results demonstrating its superior performance, especially on low frame rate videos. Furthermore, we conducted ablation studies to analyze the contribution of each model component to the overall performance.

A. Datasets

The HMDB-51 [50] dataset consists of 6766 video clips from various sources, such as movies and web videos, covering 51 action categories, such as jump, kiss, and laugh. Each category contains at least 101 clips, and the original evaluation scheme divided the dataset into three parts with 70 training and 30 testing clips per action class in each split. The final performance of the model was determined by its average accuracy across these three splits.

Kinetics 400/600 [51], [52] is a collection of large, high quality datasets with links to up to 650,000 video clips that span 400 or 600 human activity classes depending on the dataset version. The videos feature human-human and human-object interactions, such as handshakes, hugs, and people performing musical instruments. At least 400 to 600 video clips were included in each activity class. Each clip is roughly 10 s long and was manually labeled with one action class.

Something-Something V2 (SSV2) [53] comprises an extensive collection of labeled video clips depicting people performing predefined basic actions using everyday objects created by numerous crowd workers. The dataset enables machine learning models to gain a detailed understanding of fundamental physical world actions. There are 220,847 videos, including 168,913 for training, 24,777 for validation, and 27,157 for testing, for 174 labels. To evaluate the performance of the proposed model, we used lightweight versions [54] of the Kinetics 400 and Kinetics 600 datasets. In the SSV2 dataset, we focused on classes with large motions, where the sampled frames are more likely to lose temporal information. In this case, large motions denote abrupt motions with significant changes in the temporal domain, and we selected the class based on its characteristic of having high optical flow values.

B. Implementation Details

We used a video clip length of 16 frames for this analysis. We applied the random sample function in the training phase for the sampling method, randomly selecting a starting point and extracting a fixed length segment from that point. In order to more simplify the process of optical flow extraction, we made modifications to GMflow [55] so that it operates at 20 gigaFLOPs and 3.63 million parameters. This adjustment allows us to extract an optical flow in approximately 7 milliseconds (ms). In the validation phase, we used the uniform sample function to select a uniformly distributed starting point and extract a constant length segment. Furthermore, we resized the input images to 224×224 pixels. We used MViT as our base architecture. The input data is projected to the embedding dimension of 96 channels using a patch kernel$(3 \times 7 \times 7)$, stride$(2 \times 4 \times 4)$, padding$(1 \times 3 \times 3)$ and overlap$(3 \times 7 \times 7)$. In Fig. 2, the parameter u indicates the temporal dimension of the embedded input data, which was set to 8. We set l, which represents the number of layer iterations, to 3. The parameter m_i indicates the number of iterations of the attention block in the i-th layer. m_1 was set to 3, m_2 was set to 7, and m_3 was set to 6. Finally, n, which represents the number of heads, was set to 2. The K-means method for MAEE used a parameter K of 16, and SAC used the number of 100 clusters. We implemented the random crop function for the training data and the center crop function for the validation data. We applied a cosine warm up scheduler and a soft cross entropy loss function during the training process. We set the batch size to 2 and continued training until the loss satisfactorily converged. The experiments were conducted on a system equipped with an Intel I7-10700 K CPU and an Nvidia GeForce RTX 3090 GPU.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

TABLE I
ACCURACY COMPARISON WITH VARIOUS ACTION RECOGNITION MODELS ON DIFFERENT DATASETS

Dataset	N^c	I3D	R(2+1)D	ViViT	MViT	Ours
		RGB + Flow	RGB + Flow	RGB	RGB	RGB + Flow
		top-1 / top-5	top-1 / top-5	top-1 / top-5	top-1 / top-5	top-1 / top-5
HMDB-51	1	64.5 / 81.3	66.1 / 83.9	58.1 / 84.0	75.1 / 84.3	**83.6 / 88.6**
	2	64.8 / 81.4	66.3 / 83.5	58.2 / 83.8	76.2 / 84.6	**82.6 / 88.7**
	3	65.1 / 82.3	66.4 / 83.8	58.4 / 83.8	77.1 / 84.7	**82.6 / 88.7**
	4	65.7 / 83.1	65.8 / 83.5	58.3 / 84.1	77.7 / 84.6	**81.1 / 88.5**
	5	66.5 / 83.1	67.1 / 84.2	58.5 / 83.7	77.0 / 84.7	**83.7 / 89.1**
Kinetics 400 (5%)	1	55.4 / 77.6	51.6 / 72.2	54.3 / 78.0	51.8 / 77.4	**75.9 / 83.3**
	2	55.1 / 77.2	52.1 / 72.6	53.5 / 78.6	53.4 / 79.4	**73.8 / 83.0**
	3	55.8 / 77.5	51.8 / 72.4	55.5 / 78.1	53.9 / 79.5	**74.2 / 83.6**
	4	56.3 / 78.8	52.6 / 73.8	55.4 / 78.1	54.2 / 79.5	**74.0 / 83.6**
	5	56.9 / 78.9	52.6 / 73.9	55.8 / 78.2	54.2 / 79.5	**74.2 / 83.7**
Kinetics 600 (5%)	1	49.1 / 71.9	44.3 / 68.2	49.3 / 76.5	52.0 / 74.2	**59.7 / 77.7**
	2	48.9 / 71.6	45.8 / 69.8	50.2 / 76.5	51.8 / 74.0	**56.9 / 76.4**
	3	49.6 / 72.2	44.7 / 69.0	49.6 / 76.3	52.9 / 74.8	**56.9 / 76.5**
	4	49.8 / 72.2	45.9 / 68.9	51.8 / 76.3	53.1 / 75.0	**56.8 / 76.2**
	5	50.5 / 72.8	45.9 / 69.7	51.0 / 76.4	53.2 / 74.9	**56.9 / 77.1**
SSV2 (lightweight)	1	42.5 / 67.6	48.0 / 72.4	44.2 / 69.5	59.1 / 78.1	**82.9 / 93.1**
	2	41.1 / 66.8	47.9 / 73.4	43.5 / 68.8	58.0 / 76.6	**80.2 / 90.0**
	3	42.8 / 67.2	48.2 / 72.9	45.4 / 70.4	63.5 / 83.7	**80.1 / 90.0**
	4	42.9 / 67.5	48.2 / 73.4	45.9 / 71.0	64.9 / 84.7	**79.0 / 88.8**
	5	43.7 / 68.2	48.4 / 73.6	46.4 / 71.1	65.7 / 86.9	**79.4 / 89.2**

TABLE II
COMPUTATION COMPLEXITY WITH VARIOUS ACTION RECOGNITION MODELS ON KINETICS 400 (5%)

Model	FLOPs (G)	Param (M)
I3D	107.1	25.3
R(2+1)D	325.6	31.3
ViViT	3992	310.8
MViT	225	51.2
Ours (flow estimation + the others)	243 (20 + 223)	86.4 (3.6 + 82.8)

TABLE III
EVALUATION OF PROPOSED MODULES ON KINETICS 400 (5%)

Duplex attention	SAC	MAEE	Accuracy
✓	✓	✓	**75.9**
✗	✓	✓	68.7
✓	✗	✓	71.4
✓	✓	✗	66.1
✓	✗	✗	58.9
✗	✗	✗	54.2

The bold values denote the best scores.

C. Experimental Results and Analysis

This section analyzes the experimental results of the proposed action recognition model, incorporating temporal information from the optical flow using the duplex attention method, implementing an MAEE, and maintaining spatial information through SAC input data. In detail, the I3D, R(2+1)D, ViViT, and MViT models used as compared methods were experimented with using their open source codes with the default hyperparameters presented in the papers. The experiments were conducted on four widely used benchmark datasets: HMDB-51, Kinetics 400 (5%), Kinetics 600 (5%), and SSV2 (lightweight). We conducted experiments varying the number of clips per video (N^c), with quantities ranging from one to five. The performance of each model is measured in terms of top-1 accuracy and top-5 accuracy.

The results in Table I on the various datasets provide an evaluation of the performance of our proposed model in comparison to the I3D, R(2+1)D, ViViT, and MViT models against various numbers of clips per video (from one to five). The datasets used for this evaluation encompass a diverse range of action recognition tasks. Our analysis demonstrates the model's robustness and generalization capability across various tasks. The proposed Fluxformer consistently outperforms I3D, R(2+1)D, ViViT, and MViT across all datasets, achieving higher accuracy rates. This superior performance can be attributed to the incorporation of temporal information through duplex attention, along with an MAEE, and the preservation of spatial information by SAC. These techniques result in enhanced accuracy for action recognition models.

Upon analyzing the impact of the N^c, it is observed that the proposed model achieves the highest accuracy with just one clip per video. The performance remains relatively stable when using two to five clips per video, with only minor fluctuations in accuracy values. In contrast, the MViT model displays the highest accuracy when using five clips per video, and the performance is relatively consistent across various numbers of clips, with the lowest accuracy observed when processing two clips per video. Overall, the proposed model demonstrates robustness and adaptability across a range of lightweight datasets and varying numbers of clips per video, especially in abrupt and large motions. It indicates the potential for widespread application in action recognition tasks.

Table II compares the proposed model's computation complexity with I3D, R(2+1)D, ViViT and MViT models in terms of floating point operations (FLOPs) and the number of parameters (Param) on the Kinetics 400 dataset. The FLOPs and Param are essential factors in determining the computational efficiency of a model. In terms of computational requirements, the ViViT model is the most resource intensive, as evidenced by its 3992 gigaFLOPs and 310.8 million parameters. While ViViT has shown impressive performance in various computer vision applications, its substantial resource requirements may be a limiting factor in resource constrained environments. On the other hand, the MViT model offers a significant reduction in computational complexity compared to ViViT, using 225 gigaFLOPs and 51.2 million parameters. However, it should be noted that the performance of the MViT model in action recognition tasks may not be as exceptional as that of our proposed model. Our proposed action recognition model shows an optimal balance between computational efficiency and performance, with 243 gigaFLOPs and 86.4 million parameters. The optical flow estimation has 20 gigaFLOPs and 3.6 million parameters, and the other action recognition has 223 gigaFLOPs and 82.8 million parameters. Despite having slightly higher FLOPs compared to the MViT model, our model provides superior action recognition accuracy while maintaining reasonable computational complexity. Our experimental data indicate that the proposed model provides strong performance in terms of both action recognition accuracy and resource efficiency. The experimental results demonstrate the advantages of our proposed model in terms of action recognition accuracy and resource efficiency. The proposed model requires a total of 31 ms per frame for inference, with 7 ms for optical flow estimation and 24 ms for the others. Hence, the proposed model can support real-time tasks with limited computing resources. This indicates that the duplex attention mechanism using optical flow and RGB, spatial attention clustering, and flow-guided clustering to extract meaningful action events effectively addresses the limitations of most action recognition models with the transformer structure.

D. Ablation Study

This section presents the experiments investigating the effect of several proposed modules on action recognition performance. As listed in Table III, the base model we used, MViT, achieves an accuracy of 54.2% on the Kinetics 400 dataset. When duplex

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

TABLE IV
EVALUATION OF PROPOSED MODULES ON KINETICS 600 (5%)

Duplex attention	SAC	MAEE	Accuracy
✓	✓	✓	**59.7**
✗	✓	✓	58.3
✓	✗	✓	58.7
✓	✓	✗	58.2
✓	✗	✗	56.8
✗	✗	✗	53.2

The bold values denote the best scores.

TABLE V
EVALUATION OF PROPOSED MODULES ON SSV2 (LIGHTWEIGHT)

Duplex attention	SAC	MAEE	Accuracy
✓	✓	✓	**82.9**
✗	✓	✓	77.1
✓	✗	✓	78.7
✓	✓	✗	79.3
✓	✗	✗	75.1
✗	✗	✗	65.7

The bold values denote the best scores.

attention is added to the base model, the accuracy increases to 4.7%. This result suggests that duplex attention is an effective enhancement for supplementing temporal information in video data.

Next, the SAC technique is added to the model with duplex attention, significantly improving accuracy, increasing to 7.2%. This finding indicates that Ss_{token} is effective at supplementing spatial information and improving the ability to capture spatial relationships between features. Finally, the MAEE technique is added to the model with duplex and SAC, further improving accuracy to 9.8%. This outcome suggests that the MAEE is effective at extracting meaningful action events from the entire video and improving the ability to capture the overall context of the video. Table IV presents the results of a similar ablation study on the Kinetics 600 dataset, demonstrating that each proposed module significantly improves accuracy when added to the base model. The accuracy increases from 53.2% for the base model to 56.8% withduplex attention, to 1.4% with SAC, and, finally, to 1.5% with the MAEE.

As listed in Table V, the inclusion of each proposed module leads to a noteworthy enhancement in accuracy when compared to the base model. Starting from 65.7% accuracy for the base model, adding duplex attention increases it to 9.4%, followed by SAC at 4.2%, and, finally, MAEE at 3.6%.

In summary, the ablation studies demonstrate that each of the proposed enhancements (duplex attention, SAC, and MAEE) significantly improves accuracy when added to the base model and that their combined use can result in even greater improvements. These findings suggest that these enhancements effectively supplement temporal and spatial information and extract meaningful action events from video data, leading to more accurate and robust deep learning models for video processing tasks.

In addition, the proposed MAEE has different action recognition performance depending on the number of clusters when sampling images with different number of clusters. The number of clusters, K, is used to maintain temporal information and helps to avoid sampling temporally similar images. As shown in Table VI, regardless of the dataset, the ideal number of clusters is the same: 16. Therefore, most datasets performed well with

TABLE VI
ACCURACY VARIATION ACCORDING TO K

	$K = 4$	$K = 8$	$K = 16$	$K = 32$
HMDB-51	65.2	74.8	**83.6**	81.3
Kinetics 400 (5%)	58.1	69.3	**75.9**	74.6
Kinetics 600 (5%)	52.0	53.2	**59.7**	55.4
SSV2 (lightweight)	65.1	73.9	**82.9**	81.9

The bold values denote the best scores.

16 clusters, which is consistent with our experimental results where we set the number of clusters to 16 for most datasets.

V. CONCLUSION

In conclusion, we presented the Fluxformer, an action recognition model that effectively extracts meaningful frames while incorporating spatio-temporal information. This approach addresses the challenges faced by robots with limited computing resources and surpasses the limitations of conventional models. The Fluxformer integrates temporal information from the optical flow, improves accuracy through significant action extraction, and preserves spatial information by tokenizing the input data. Consequently, this model offers a significant advancement in human action recognition, enhancing communication, safety, and automation in human-robot interactions while overcoming the challenges of computational complexity and resource limitations.

REFERENCES

[1] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6299–6308.

[2] K. Bruno, D. Tran, and L. Torresani, "SCSampler: Sampling salient clips from video for efficient action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6232–6242.

[3] R. Jian and M. Radomir, "Best frame selection in a short video," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 3212–3221.

[4] A. Krizhevsky et al., "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, pp. 84–90, 2017.

[5] K. Simonyan et al., "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[6] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[7] H. Zhang et al., "ResNeSt: Split-attention networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2736–2746.

[8] L. Radosavovic, R. P Kosaraju, R. Girshick, K. He, and P. Dollar, "Designing network design spaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10428–10436.

[9] M. Seo et al., "A self-supervised sampler for efficient action recognition: Real-world applications in surveillance systems," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1752–1759, Apr. 2022.

[10] G. Goletto, M. Planamente, B. Caputo, and G. Averta, "Bringing online egocentric action recognition into the wild," *IEEE Robot. Automat. Lett.*, vol. 8, no. 4, pp. 2333–2340, Apr. 2023.

[11] S. Li, J. Yi, Y. A Farha, and J. Gall, "Pose refinement graph convolutional network for skeleton-based action recognition," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1028–1035, Apr. 2021.

[12] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 5533–5541.

[13] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[14] B. Zhou et al., "Temporal relational reasoning in videos," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 803–818.

[15] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

[16] S. Xie et al., "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 305–321.

[17] R. Christoph et al., "Spatiotemporal residual networks for video action recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3476–3484.

[18] C. Feichtenhofer, "X3D: Expanding architectures for efficient video recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 203–213.

[19] A. Stergiou, R. Poppe, and G. Kalliatakis, "Refining activation downsampling with SoftPool," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10337–10346.

[20] Z. Liu, L. Wang, W. Wu, C. Qian, and T. Lu, "TAM: Temporal adaptive module for video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 13688–13698.

[21] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 6450–6459, 2018.

[22] A. Piergiovanni and M. S. Ryoo, "Representation flow for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9937–9945.

[23] M. U. Khalid and J. Yu, "Multi-modal three-stream network for action recognition," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2018, pp. 3210–3215.

[24] H. Duan, Y. Zhao, K. Chen, D. Lin, and B. Dai, "Revisiting skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2959–2968.

[25] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[26] X. Wang et al., "OadTR: Online action detection with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7545–7555.

[27] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L. -C. Chen, "Max-DeepLab: End-to-end panoptic segmentation with mask transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5463–5474.

[28] H. Zhao, V. Belagiannis, and K. Dietmayer, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16259–16268.

[29] H. Touvron et al., "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[30] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 3163–3172.

[31] G. Bertasius et al., "Is space-time attention all you need for video understanding?," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 813–824.

[32] S. Wang et al., "Linformer: Self-attention with linear complexity," 2022, *arXiv:2006.04768*.

[33] N. Kitaev et al., "Reformer: The efficient transformer," 2020, *arXiv:2001.04451*.

[34] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6816–6826.

[35] L. Yanghao et al., "MViTv2: Improved multiscale vision transformers for classification and detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4804–4814.

[36] Z. Liu et al., "Video swin transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3192–3201.

[37] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9992–10002.

[38] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[39] R. Wang et al., "BEVT: BERT pretraining of video transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14713–14723.

[40] S. Yan et al., "Multiview transformers for video recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3323–3333.

[41] S. Li et al., "GroupFormer: Group activity recognition with clustered spatial-temporal transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 13648–13657.

[42] J. Chen et al., "MM-ViT: Multi-modal video transformer for compressed video action recognition," in *Proc. Winter Conf. Appl. Comput. Vis.*, pp. 786–797, 2022.

[43] R. Sun et al., "Wlit: Windows and linear transformer for video action recognition," *Sensors*, vol. 23, no. 3, pp 1616–1616, 2023.

[44] B. Li et al., "Shrinking temporal attention in transformers for video action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 1263–1271.

[45] W. Wenhao, D. He, X. Tan, S. Chen, and S. Wen, "Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6222–6231.

[46] Z. Wu, C. Xiong, C. -Y. Ma, R. Socher, and L. S. Davis, "AdaFrame: Adaptive frame selection for fast video recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1278–1287.

[47] Y. Serena, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2678–2687.

[48] T. Kanungo, D. M Mount, N. S Netanyahu, C. D Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.

[49] M. Zhao et al., "Search-Map-Search: A frame selection paradigm for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 10627–10636.

[50] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2011, pp. 2556–2563.

[51] W. Kay et al., "The kinetics human action video dataset," 2017, *arXiv:1705.06950*.

[52] J. Carreira et al., "A short note about kinetics-600," 2018, *arXiv:1808.01340*.

[53] R. Goyal et al., "The 'something something' video database for learning and evaluating visual common sense," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 5843–5851.

[54] "Kinetics dataset (5%)," 2022. [Online]. Available: https://www.kaggle.com/datasets/rohanmallick/kinetics-train-5per

[55] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao, "GMFlow: Learning optical flow via global matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8121–8130.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.