Research

Towards efficient video-based action recognition: context-aware memory attention network

Thean Chun Koh[1] · Chai Kiat Yeo[1] · Xuan Jing[1,2] · Sunil Sivadas[2]

Received: 16 July 2023 / Accepted: 1 November 2023
Published online: 13 November 2023

Abstract
Given the prevalence of surveillance cameras in our daily lives, human action recognition from videos holds significant practical applications. A persistent challenge in this field is to develop more efficient models capable of real-time recognition with high accuracy for widespread implementation. In this research paper, we introduce a novel human action recognition model named Context-Aware Memory Attention Network (CAMA-Net), which eliminates the need for optical flow extraction and 3D convolution which are computationally intensive. By removing these components, CAMA-Net achieves superior efficiency compared to many existing approaches in terms of computation efficiency. A pivotal component of CAMA-Net is the Context-Aware Memory Attention Module, an attention module that computes the relevance score between key-value pairs obtained from the 2D ResNet backbone. This process establishes correspondences between video frames. To validate our method, we conduct experiments on four well-known action recognition datasets: ActivityNet, Diving48, HMDB51 and UCF101. The experimental results convincingly demonstrate the effectiveness of our proposed model, surpassing the performance of existing 2D-CNN based baseline models.

Article Highlights

- Recent human action recognition models are not yet ready for practical applications due to high computation needs.
- We propose a 2D CNN-based human action recognition method to reduce the computation load.
- The proposed method achieves competitive performance compared to most SOTA 2D CNN-based methods on public datasets.

Keywords Action recognition · Deep learning · Convolutional neural network · Attention

1 Introduction

Human action recognition is a computer vision task to identify some human actions from a series of observations. Every human action, no matter how trivial, is done for some purpose. Due to its wide range of applications in intelligent video surveillance [1, 2], robotics [3], video storage retrieval, smart home monitoring, entertainment and autonomous driving vehicles, human action recognition (HAR) has gained significant popularity in the field of

✉ Thean Chun Koh, KOHT0034@e.ntu.edu.sg; Chai Kiat Yeo, ASCKYEO@ntu.edu.sg; Xuan Jing, xuanjing@gmail.com; Sunil Sivadas, sunil.sivadas@ncs.com.sg | [1]School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore 639798, Singapore. [2]NCS Pte Ltd, Ang Mo Kio Street 62, Singapore 569141, Singapore.

video analytics. HAR relies on computational algorithms to identify and understand human actions [4].

With the advance in computational technologies, deep learning has replaced traditional machine learning in many computer vision tasks, employing multiple layers of artificial neural networks to achieve state-of-the-art (SOTA) accuracy in tasks such as facial recognition, object detection etc.

Despite the extensive research conducted in the field of HAR, numerous challenges still remain unaddressed. HAR from raw videos poses a significant challenge as the model must essentially identify actions based on a series of observations. To achieve accurate predictions, spatial and temporal information are essential, resulting in a higher computational demand compared to other computer vision tasks [5, 6], which only require spatial information. Consequently, HAR models tend to be complex in nature. In the past, researchers relied on designing hand-crafted feature extractors to encode the necessary features for obtaining precise motion representations from video sequences, aiming to enhance the accuracy of HAR models [7–9]. Nevertheless, methods based on hand-crafted feature extraction have limitations as they heavily rely on human insight and lack the ability to automatically adapt to new data. Consequently, their applicability in real-world scenarios which are often dynamic and ever-changing, is very limited.

Convolutional Neural Networks (CNNs) play a crucial role in deep learning and find extensive use in various HAR models. They have the ability to directly learn human action features from video data without the need for any hand-crafted feature pre-processing [10]. Currently, the most popular HAR methods include two-stream networks based on 3D CNN and Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM). These methods achieve commendable performance, but their computational requirements are high, especially when dealing with long untrimmed videos. Consequently, researchers have shifted their focus towards developing efficient HAR models using 2D CNN-based approaches.

This paper expands our initial work [11] to showcase the comprehensive performance of our proposed 2D CNN-based model, Context-Aware Memory Attention Network (CAMA-Net) which is specifically designed for HAR. CAMA-Net eliminates the need for optical flow computation and 3D convolution. We conduct additional extensive experiments on different public datasets, namely ActivityNet [12], Diving48 [13], HMDB-51 [14] and UCF-101 [15] to prove that our model is robust enough to work in datasets with many different activities. In all the datasets, the proposed model outperforms the SOTA baselines. In addition, we perform more ablation studies to showcase the contributions of the various entities in CAMA-Net and also provide an insight of the inference speed gap between 2D CNN, 3D CNN and two-stream based HAR models. In this paper, we also provide a detailed survey of the related work.

The contributions of our paper can be summarized as follows:

- We introduce a novel HAR model, named Context Aware Memory Attention Network (CAMA-Net), which does not rely on optical flow computation or 3D convolution which are computationally intensive.
- The Context Aware Memory Attention (CAMA) module in CAMA-Net accurately computes the relevance scores between key and value pairs obtained from the backbone output for the proposed model to learn a more discriminative spatio-temporal representation for action recognition.
- We comprehensively evaluate the performance and robustness of CAMA-Net across four widely-used datasets: ActivityNet [12], Diving48 [13], HMDB-51 [14] and UCF-101 [15]. These datasets have different video lengths and different action classes.
- The experimental results validate its competitive performance when compared to state-of-the-art methods in the field of HAR and demonstrate the robustness of our proposed model across various datasets.

2 Related works

2.1 Deep learning based action recognition

Over the past few years, deep learning models have emerged as the preferred approach for action recognition tasks. This is primarily due to their ability to extract high-level features from input data, which is in stark contrast to the comparatively rigid and less adaptable nature of hand-crafted feature methods.

At present, the predominant approaches in HAR utilize two-stream networks [16–18]. In these networks, one stream takes RGB frames as input, extracting appearance information, while the other stream employs optical flow as input, capturing motion information. Optical flow, which recovers pixel-level motion from variations in brightness patterns within spatial-temporal images [19–21], is used to effectively track the movement of objects.

Motion representation is thus one of the most important components for action recognition task. [16–18] use optical flow to represent short-term motion and many works use it as an additional input source, resulting in significant improvement in action recognition performance compared to using only the raw data. Current

popular optical flow computation approaches [22–24] pre-compute the optical flow out-of-band and store the information which is inefficient. To address this inefficiency of estimating optical flow, some recent works accelerate optical flow estimation by the judicious construction of CNN models, such as FlowNet family [25, 26], PWC-Net [27] and SpyNet [28] etc. Nonetheless, these models focus on improving the accuracy of the optical flow estimation which is not directly related to the deep learning models for HAR. Other works [18, 29] propose an encoder-decoder network, where the encoder network aims to regenerate the optical flow and the decoder network is the action recognition network. However, the encoder-decoder architecture also entails high computational resource. Hence, it remains challenging to have the best motion representation which is efficient and effective for HAR [30, 31]. To this end, we decide to drop optical flow for fast HAR.

Another category of HAR approaches frequently proposed is 3D CNNs due to their well-defined architectures for temporal modelling [32–34]. 3D convolutional operators are built such that they combine the information in both the spatial and temporal dimensions within the local receptive fields [35, 36]. 3D convolutions and 3D pooling are used in 3D CNNs for propagating temporal information across all the layers in the network, so it can learn features that encode temporal information efficiently. The C3D model [33] is first pre-trained on a large-scale public video dataset to learn the spatio-temporal features which are then used as the input to the linear Support Vector Machine (SVM) classifier for action class prediction. I3D [37] uses a deep Inflated 3D CNN model by expanding the popular Inception model [38] to 3D so it can learn the spatiotemporal features in videos for HAR application. T3D [39] proposes a temporal 3D CNN model by extending the original idea of DenseNet [40], while DTPP [41] modifies the temporal pyramid pooling function which originally only works for spatial dimension to three space-time dimensions and use the 3D structure in a two-stream CNN in lieu of the common two-stream 2D CNN.

However, these 3D convolution-based models are typically trained and learned using short video snippets instead of considering the entire videos. As a result, they struggle to accurately capture actions that extend beyond their limited temporal context. To address this limitation, Slowfast Networks [42] incorporates two pathways operating at different frame rates. The slow and fast frame rates allow for the capture of both spatial semantics and fine resolution temporal motion respectively, with lateral connections employed to integrate information from both pathways. It is worth noting that, similar to other deep learning models, the performance of HAR significantly improves when 3D CNN models are trained on large-scale video datasets. However, the computational cost associated with 3D CNN-based methods increases considerably due to the extensive number of parameters involved in stacked 3D convolutions.

Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM) [43–45], originally popular in natural language processing, have also found application in HAR. RNNs are deep learning models that possess a memory state, denoted as "h", which summarizes past information to predict future outcomes. Through backpropagation, the RNN learns to capture the history or memory vector. In HAR, RNNs utilize the input (e.g., frames) and memory state (h) to predict the subsequent action. The incorporation of RNNs in HAR offers the advantage of preserving temporal information throughout the entire training process, thereby enhancing the accuracy of action recognition.

In general, HAR is like video understanding and can be treated as sequence modeling. LRCN [46] connects LSTM directly to SOTA CNN models to learn both spatial information and temporal dynamics. Thus, it can be perceived as a direct extension of the encoder-decoder architecture being applied for video representations. One notable advantage of LRCN is its capability to effectively handle sequences of varying lengths. To further enhance the processing of video data, a novel approach called DB-LSTM [47] has been introduced. DB-LSTM combines CNN with deep bidirectional LSTM networks [48]. These LSTM networks are stacked with multiple layers in both the forward and backward passes, thereby increasing the network's depth, enabling it to recognize actions in long videos which has been a challenge for most of the common sequence models.

In contrast to these approaches, our proposed model, CAMA-Net, does not rely on pre-computed optical flow. Instead, it directly takes raw RGB video frames as input for action recognition. This is accomplished via 2D CNN based methods together with temporal modelling. We provide details on how we integrate 2D ResNet with a memory attention network to find the correspondences between video frames in the later section.

2.2 Attention mechanism

Recently, attention model is becoming very popular as it can focus on the interesting regions in the target videos [49–52]. Attention mechanism has first been applied for sequence-to-sequence learning in machine translation [53].

The two common types of visual attention [54] are hard and soft attention. The hard attention uses binary choices to choose spatial regions. Several works such as [55, 56] use the idea of hard attention in object recognition to

extract the most important features in the images. On the other hand, in soft attention mechanisms, the spatial region of interest is chosen by the weighted averages. [57] designs a teacher-student learning-based model by utilizing an activation-based attention map and a gradient-based attention map. These attentions are propagated from a strong network to a weak CNN to improve the image recognition. Non-local Networks [58] learn long-range temporal relationship by using self-attention mechanism.

Wang et al. [59] introduces a channel attention block that employs 1D convolution to evaluate channel interactions while preserving dimensionality. Misra et al. [60] proposes employing triplet attention to determine attention weights via a three-branch structure, enabling the capture of cross-dimension interactions. Wang et al. [61] designs a self-attention mechanism that dynamically incorporates long-term temporal connections across the video sequence by capturing the relationship between the current frame and adjacent frame. The Stand-alone Inter-Frame Attention [62] is an attention mechanism that operates across multiple frames, computing local self-attention for every spatial position. Hao et al. [63] proposes an effective attention-in-attention technique for enhancing element-wise features, exploring the possibility of integrating channel context into the spatio-temporal attention learning module. Visual attention network [64] uses a large kernel attention to support the establishment of self-adaptive and extended-range correlations of self-attention.

Due to the advancements in applying attention mechanism in different computer vision tasks, we propose a novel approach that incorporates self-attention modules differently into a CNN-based method. Our way of integration is simply to find the correspondences between selected features using attention mechanism, without passing the entire set of features to the CNN model, thus reducing the number of learning parameters compared to pure CNN model. This integration aims to reduce computational complexity in action recognition tasks while maintaining competitive performance.

2.3 2D CNN-based methods for action recognition

As previously mentioned, the well-defined architectures in 3D CNNs make them popular in the field of HAR for temporal modeling. While these networks can achieve impressive performance, their widespread adoption is hindered by high computational requirements and significant GPU memory usage. To address these concerns and develop efficient HAR algorithms, researchers have turned their attention to 2D CNN-based methods. However, these methods do have their limitations. 2D convolutional operators operate within individual image frames, limiting their ability to capture spatial information across adjacent frames. If a 2D CNN model is used directly, it will only have partial observation, thereby compromising the accuracy of action prediction, particularly for longer duration actions. Therefore, to overcome this challenge and improve the performance of 2D CNN-based action recognition algorithms, it is crucial to incorporate temporal modeling techniques.

To address the limitation of sequence length during training, the Temporal Segment Network (TSN) [65] introduces a temporal sampling approach for video clips. TSN aggregates the features to generate video-level representations using an average pooling consensus function. Building upon TSN, the Temporal Relation Networks (TRN) [66] further enhances the temporal modeling capability by leveraging the relationships among video frames in the temporal domain.

In recent times, there has been a rise in the popularity of feature-level inter-frame difference methods for encoding short-term motion information between neighboring frames. For instance, the STM (Spatio-Temporal Motion) approach [67] models the motion representation of spatio-temporal features by utilizing the feature difference between adjacent frames. Another method called Temporal Shift Module (TSM) [68] employs a temporal shift operation to efficiently exchange temporal information among features through the channel dimension, thereby enhancing the performance of 2D CNN techniques. TANet (Temporal Adaptive Network) [69] improves the efficiency of action recognition tasks by stacking multiple Temporal Adaptive Modules (TAM) that encompass both global and local branches, enabling the learning of long-range temporal information. Furthermore, the Temporal Pyramid Network (TPN) [70] introduces feature hierarchy modules to aggregate diverse visual information from different feature levels.

3 Methods

3.1 Model architecture

To achieve faster action recognition, our proposed model, CAMA-Net, eliminates the need for pre-computed optical flow and solely relies on raw RGB video frames as input.

Figure 1 shows the CAMA-Net architecture. The video input is divided into L video clips, and from each video clip, a random short snippet is selected. Each snippet consists of a set of RGB frames. The snippet is then fed into the CNN backbone, followed by adaptive average pooling. The resulting outputs are passed through two separate channels: the Sequence CNN and the Segmental Consensus. The former produces a pair of memory features of (B, C, H,

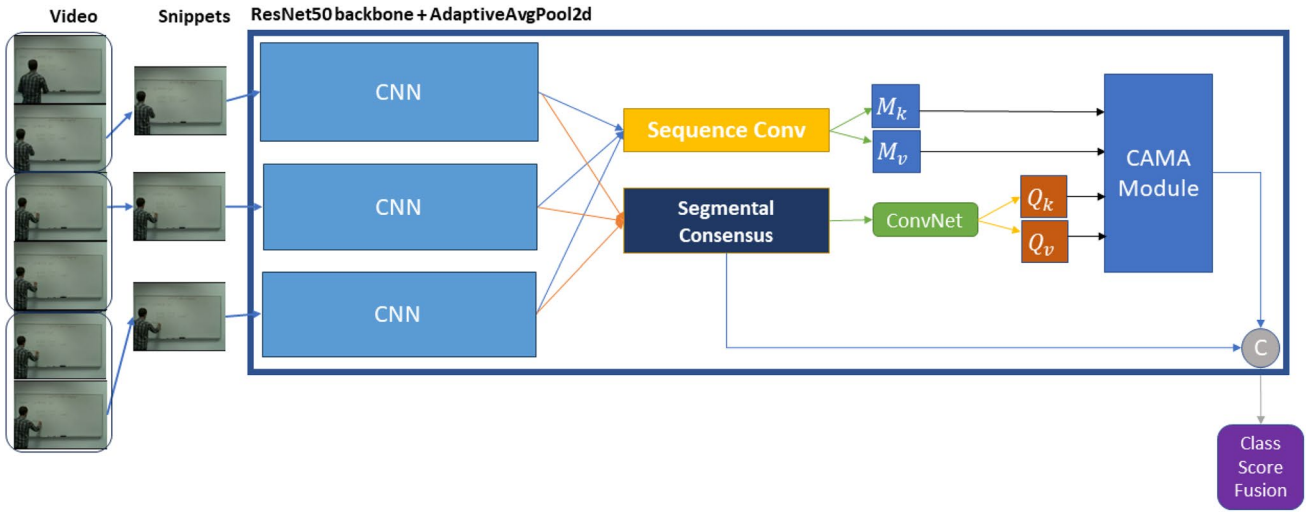

Fig. 1 Overview of CAMA-Net architecture. The video input is divided into L video clips, and from each video clip, a random short snippet is selected. Each snippet consists of a set of RGB frames. The snippet is then fed into the ConvNet backbone, followed by adaptive average pooling. The resulting outputs are passed through two separate channels: the Sequence ConvNet and the Segmental Consensus. The Sequence ConvNet produces a pair of memory features, while the Segmental Consensus generates a pair of query features. These memory and query features are input into the CAMA module to compute the relevance scores between them. Thereafter, the outputs from the CAMA module and the Segmental Consensus are concatenated. This concatenated output provides the action class scores for the different snippets

W, T) dimension, while the latter generates a pair of query features of (B, C, H, W) dimension. Here B is the batch size, C is the channel size, H and W are the height and weight respectively and T is the sequence length. Sequence CNN based channel is basically a sequence of 1×1convolution module while Segmental Consensus channel is basically an average pooling aggregation function in the temporal dimension.

The memory and query features are specifically designed to serve different purposes. The memory features are analogous to source features or base features, encompassing the majority of the content. On the other hand, the query features can be considered as summarized or filtered features, capturing the most important aspects. Both these features play a crucial role as inputs to the CAMA module, where the relevance scores between them are computed. These relevance scores serve to allow the proposed model to learn a more discriminative spatio-temporal representation for action recognition. Subsequently, the outputs of the CAMA module and the Segmental Consensus are concatenated boosting the prediction performance. This concatenated output provides the action class scores for the different snippets.

3.2 CAMA module

Figure 2 shows the details of the CAMA module. Its primary role is to determine the relevance between the memory key features (M_k) and the query key features (Q_k) through the utilization of three relevance functions. Both the

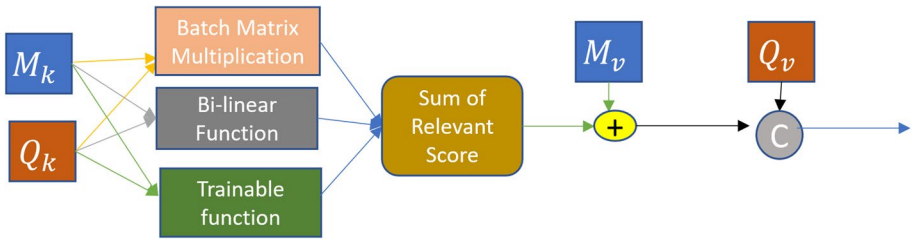

Fig. 2 CAMA module. The vital role of CAMA module is to determine the relevance between the memory key features (M_k) and the query key features (Q_k) through the utilization of three relevance functions. These computed relevance scores are then summed with the memory value features (M_v) and concatenated with the query value features (Q_v). The resulting output from the CAMA module is subsequently concatenated with the output of the Segmental Consensus module. This concatenated output provides the action class scores for the different snippets

memory and query features possess distinct key features (M_k, Q_k) and value features (M_v, Q_v) as shown in Fig. 2. The CAMA module employs three distinct functions to calculate the relevance score between the memory key features and the query key features. These scores are combined with the memory value and then concatenated with the query value. The resulting information is subsequently fed into a Fully Connected Network to predict the action class.

Our proposed relevance functions are unique, unlike others such as that proposed in [71] which calculates the relevance scores between the current features (query key) and all the features together (memory key). Our design comprises three functions. The first relevance equation takes a direct approach to determine the relevance between the memory key (M_k) and query key (Q_k) by comparing their affinity. We used batch matrix multiplication for the memory key (M_k) and query key (Q_k) which does not involve any learnable parameters. Before relevance score computation, we change the way the features are organized without changing their contents, as shown as in Fig. 3.The first relevance function $R(M_k, Q_k)$ is shown below:

$$R(M_k, Q_k) = M_k Q_k \tag{1}$$

In order to improve the accuracy of the relevance score calculation, we introduce a second relevance equation based on a bi-linear form. This additional equation is necessary because the first relevance equation alone is insufficient. To enable the bi-linear form, we utilize a new metric $W \in R^{c_k \times c_v}$, which facilitates the computation of the relevance equation:

$$R(M_k, Q_k) = M_k W Q_k \tag{2}$$

We define a third relevance function, which incorporates trainable relevance scores ($r|M_k, Q_k$), allowing the network to explicitly learn these scores. The outputs from these relevance functions are passed through the Softmax function. Softmax function is a function that turns a vector into a vector where its values summed to 1. Here we use Softmax function for the last dimension of each output. The relevance scores are then summed to generate a context term, denoted as c_k. This context term is added to the memory value, M_v, and concatenated with the query value, Q_v. Finally, this combined information is fed into the Fully Connected Network for predicting the action class. The equation for the context term is as follows:

$$\begin{aligned} c_k &= \sum (R(M_k, Q_k)) \\ &= softmax(M_k Q_k) + softmax(M_k W Q_k) + softmax(r) \end{aligned} \tag{3}$$

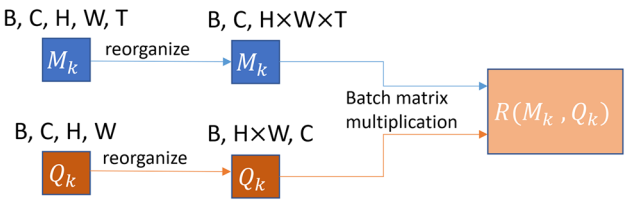

Fig. 3 First relevance score computation. Before any relevance score computation, memory key (M_k) and query key (Q_k) are organized without changing their contents. The first relevance equation takes a direct approach to determine the relevance between the memory key (M_k) and query key (Q_k) by comparing their affinity. Multiplication has been performed considering that the batch size of the two features are the same. We used batch matrix multiplication for the memory key (M_k) and query key (Q_k) which does not involve any learnable parameters

4 Experiments

4.1 Datasets

We evaluate the performance of the proposed CAMA-Net on popular benchmark datasets, ActivityNet, Diving48, HMDB-51 and UCF-101. ActivityNet [12] contains 200 different types of activities and Version 1.3 contains around 20,000 untrimmed videos. Diving48 [13] is a fine-grained video dataset on competitive diving, consisting of around 18,000 trimmed video clips of 48 unambiguous dive sequences. HMDB51 [14] contains about 7000 videos comprising 51 categories. UCF-101 [15] contains 101 action classes with around 13,000 videos.

4.2 Implementation details

We implement the CAMA-Net framework using ResNet50 and ResNet101 as the backbones. The video sampling frame, denoted as T, is set to 24. The shorter size of the input video frames is resized to 256 and common data augmentation techniques such as random horizontal flipping, multi-scale cropping are applied before training [65]. The optimal settings for model training are as follows: the batch size is set to 6, and the initial learning rate is set to 0.00008. The total number of training epochs is 80 for HMDB51 and Diving48, 120 for UCF101 and ActivityNet. The weight decay is set to 0.0002.

During testing, the video input from the test dataset is resized to 256 on the shorter side to maintain consistency with the training process. We initialize the model with a pre-trained ImageNet model when training on all the datasets. Both the model training and testing are conducted on two NVIDIA Tesla V100 Tensor Core GPUs.

Table 1 Performance comparison against SOTA baselines on ActivityNet dataset. Higher values are better

Method	Backbone	Accuracies	
		Top-1	Top-5
TSN [65]	ResNet 50	64.29	87.92
TRN [66]	ResNet 50	64.36	87.61
TSM [68]	ResNet 50	66.13	88.04
TANet [69]	ResNet 50	66.39	87.15
TPN [70]	ResNet 50	67.99	88.06
CAMA-Net	ResNet 50	**68.49**	**89.06**
CAMA-Net	ResNet 101	**69.49**	**90.09**

Table 2 Performance comparison against SOTA baselines on Diving48 dataset. Higher values are better

Method	Backbone	Accuracies	
		Top-1	Top-5
TSN [65]	ResNet 50	71.27	95.74
TRN [66]	ResNet 50	72.08	96.14
TSM [68]	ResNet 50	72.54	96.09
TANet [69]	ResNet 50	73.35	95.69
TPN [70]	ResNet 50	73.60	96.85
CAMA-Net	ResNet 50	**74.42**	**96.45**
CAMA-Net	ResNet 101	**76.85**	**96.50**

Table 3 Performance comparison against SOTA baselines on HMDB-51 dataset. Higher values are better

Method	Backbone	Accuracies	
		Top-1	Top-5
TSN [65]	ResNet 50	47.78	76.60
TANet [69]	ResNet 50	49.74	79.67
TRN [66]	ResNet 50	52.42	80.72
TPN [70]	ResNet 50	52.94	79.22
TSM [68]	ResNet 50	54.58	80.00
CAMA-Net	ResNet 50	**54.84**	**83.27**
CAMA-Net	ResNet 101	**57.45**	**84.25**

4.3 Performance comparison

The performance of the proposed CAMA-Net is compared with state-of-the-art (SOTA) baselines on the four well-known action recognition datasets, namely ActivityNet, Diving48, HMDB51 and UCF101. The performance metric used is on top-1 and top-5 accuracies. The results are shown in Tables 1, 2, 3 and 4 for the respective datasets. It is to be noted that all the models included in the comparison solely rely on the pre-trained ImageNet model for initialization and do not undergo any additional pre-training on other large-scale video datasets.

Table 4 Performance comparison against SOTA baselines on UCF-101 dataset. Higher values are better

Method	Backbone	Accuracies	
		Top-1	Top-5
TSN [65]	ResNet 50	83.48	96.91
TRN [66]	ResNet 50	83.66	95.82
TANet [69]	ResNet 50	83.85	96.35
TSM [68]	ResNet 50	84.56	96.99
TPN [70]	ResNet 50	85.65	96.83
CAMA-Net	ResNet 50	**86.28**	**97.36**
CAMA-Net	ResNet 101	**87.18**	**98.02**

Table 5 Performance (model inference speed) comparison between 2D CNN based method vs 3D CNN based method vs two-stream (optical flow+RGB) method on UCF-101 dataset. Higher values are better

Method	2D CNN (CAMA-Net)	3D CNN [33]	Two-stream [16]
Inference speed (number of video frames/s)	**26.6**	11.3	3.8

The SOTA baselines being compared are 2D CNN based action recognition methods with late fusion of temporal information such as TRN [66] and TSN [65], 2D CNN with built-in temporal modules such as TANet [69], TPN [70] and TSM [68]. The tables show that the proposed CAMA-Net outperforms all the SOTA baselines on all four datasets, testifying to its effectiveness of its action recognition ability with unique temporal information learning techniques.

2D CNN based action recognition methods possess the advantage of faster model inference speed when compared to 3D CNN based methods and two-stream methods (optical flow and RGB frame fusion). To provide some insights on the speed difference, we record the inference speed of CAMA-Net on UCF-101 dataset in number of video frames processed per second and the result is shown in Table 5. The inference speed of a seminal 3D CNN based method, C3D [33] which is the first 3D CNN model for action recognition task and the inference speed of a two-stream network [16] are also shown. As can be seen from Table 5, CAMA-Net is more than

Table 6 Study on the relevance functions used in CAMA module. Performance of the different combinations of relevance functions on the UCF101 dataset is shown. Higher values are better

Batch matrix multiplication	Bi-linear function	Trainable function	Accuracies	
			Top-1	Top-5
✓		✓	85.80	97.62
	✓	✓	86.04	97.49
✓	✓	✓	**86.28**	**97.36**

Table 7 Study on adaptive average pooling used in CAMA-Net. Performance with and without adaptive average pooling for CAMA-Net on the UCF101 dataset is shown. Higher values are better

Adaptive average pooling	Accuracies	
	Top-1	Top-5
✗	82.84	95.40
✓	**86.28**	**97.36**

Table 8 Study on concatenation of outputs of segmental consensus and CAMA module. Performance with and without output concatenation of the two modules in CAMA-Net on the UCF101 dataset is shown. Higher values are better

Concatenation	Accuracies	
	Top-1	Top-5
✗	85.78	97.07
✓	**86.28**	**97.36**

Table 9 Study on batch normalization used in CAMA-Net. Performance with and without batch normalization for CAMA-Net on the UCF101 dataset is shown. Higher values are better

Batch normalization	Accuracies	
	Top-1	Top-5
✗	81.23	94.85
✓	**86.28**	**97.36**

twice faster compared to C3D and ten times faster than the two-stream network during inference. For practical deployment especially at edge devices, a lightweight model with fast inference speed with a little tradeoff in recognition accuracy will be most desirable and feasible.

4.4 Ablation study

Similar to the performance comparison in the previous section, the performance metrics used in the ablation studies are top-1 and top-5 accuracies. The experiments are carried out on the UCF101 dataset with ResNet50 as the backbone.

Relevance functions used for CAMA module Three different relevance functions have been designed for the CAMA module to calculate the relevance score between the memory and query features. They are the batch matrix multiplication, the bi-linear function and the trainable function. To obtain insight on the effectiveness of these relevance functions in improving the action recognition performance, an experiment is conducted using different combinations of the relevance functions. Table 6 shows the performance of the different combinations of relevance functions. The best combination using all three relevance functions is thus adopted in CAMA module to yield the best performance.

Adaptive average pooling An experiment is also carried out to validate that adaptive pooling plays a significant role in improving the effectiveness of CAMA-Net. As shown in Fig. 1, raw RGB video frames are passed through the ResNet50 backbone to extract the encoded features. Adaptive average pooling is then applied to these encoded features before they are being passed to two separate channels. Adaptive average pooling is the process that applies a 2D adaptive average pooling over an input signal composed of several input planes. Table 7 shows the performance comparison with and without adaptive average pooling after the ResNet backbone. The result shows that CAMA-Net can achieve the best performance with adaptive average pooling.

Concatenation of output of Segmental Consensus and output of CAMA module A study is also carried out to show the effectiveness of concatenation of the outputs of Segmental Consensus and CAMA module. The concatenation of both outputs can reduce bias that result in poor action recognition performance and can be considered as a type of regularization for CAMA-Net. The performance with and without concatenation of these two outputs are shown in Table 8. The result shows that CAMA-Net can achieve the best performance with the concatenation of these two outputs.

Batch normalization An experiment is also carried out to ensure that batch normalization is useful to improve the effectiveness of CAMA-Net. Batch normalization is a common method to standardize the inputs of deep learning model to a single layer for each mini batch during training process. In theory, the learning process can be stabilized, and the number of epochs required for the deep learning model to train can be reduced. Table 9 shows the performance comparison between CAMA-Net with and without batch normalization. The result shows that CAMA-Net can achieve the best performance with batch normalization.

4.5 Other experiments

Channel sizes of input features for CAMA module To recap, the inputs to the CAMA module include memory key (M_k), memory value (M_v), query key (Q_k) and query value (Q_v). These input features are passed to the CNN module to obtain the key value pairs of memory features and query features. The filter size of each CNN module is fixed at 1×1.

Table 10 Study on varying channel size of input features for CAMA module. When varying channel sizes of the input features for CAMA module on the UCF-101 dataset is shown. Higher values are better

Key channel size	Value channel size	Accuracies	
		Top-1	Top-5
256	1024	85.06	97.09
512	1024	85.75	96.75
256	2048	85.91	97.99
512	**2048**	**86.28**	**97.36**

Table 11 Study on batch size used in CAMA-Net for model training and testing. Performance of the different batch sizes for CAMA-Net on the UCF101 dataset is shown. Higher values are better

Batch size	Accuracies	
	Top-1	Top-5
12	84.85	97.30
9	85.73	97.33
6	**86.28**	**97.36**
3	83.58	96.19

Table 12 Study on varying sequence length of input videos in CAMA-Net model training and testing. Performance of the different sequence lengths of video input in CAMA-Net on the UCF101 dataset is shown. Higher values are better

Sequence length	Accuracies	
	Top-1	Top-5
6	83.19	95.51
12	86.12	96.72
18	85.88	97.20
24	**86.28**	**97.36**

Table 13 Study on initial learning rate used in CAMA-Net model training and testing. Performance of the different initial learning rates in CAMA-Net on the UCF101 dataset is shown. Higher values are better

Initial learning rate	Accuracies	
	Top-1	Top-5
0.00064	82.84	96.11
0.00032	85.01	96.80
0.00016	85.73	96.38
0.00008	**86.28**	**97.36**

Since all the above features jointly contribute to the performance of our proposed model, an experiment is conducted to vary the channel size of each feature to find the optimum. Key channel size is used for memory and query key generation while value channel number is used for memory and query values generation. Table 10 shows the performance result when varying the channel sizes of the key and value pairs. The result shows that CAMA-Net can achieve the best performance with key channel size of 512 and value channel size of 2048.

Batch size, sequence length and learning rate Hyperparameter tuning is very important for a model to achieve the best performance. Therefore, extensive experiments have been carried out to explore the possible range of values to narrow down to the most optimum ones. In action recognition, batch size, sequence length and learning rate are important hyperparameters to achieve good action recognition accuracy. Batch size denotes the number of videos that will be propagated through the network while sequence length denotes the length of the sequence for video snippets. The learning rate is a hyperparameter that controls how fast the model changes in response to the estimated error each time the model weights are updated. The performance of varying the batch size, sequence length and initial learning rate is shown in Tables 11, 12 and 13 respectively. The results show that CAMA-Net can achieve the best performance with a batch size of 6, sequence length of 24 and initial learning rate for the model training of 0.00008.

5 Conclusion

In this paper, we introduce the Context Aware Memory Attention Network (CAMA-Net) for video action recognition, eliminating the requirement for optical flow extraction. CAMA-Net offers enhanced efficiency by avoiding the computationally intensive 3D convolution. Instead, we design a Context Aware Memory Attention (CAMA) module, an attention mechanism used to compute the relevance score between key-value pairs derived from the backbone network outputs. Through extensive experiments conducted on four widely-used benchmark datasets, our proposed model demonstrates remarkable performance improvements while maintaining competitive efficiency compared to SOTA 2D CNN-based models. Our model maintains its performance amid the many different action classes regardless of video length.

Vision Transformers (ViT) [72] are actively used in the research community to replace Convolutional Neural Networks to solve the computer vision tasks, including human action recognition. Recently, there are several lightweight ViT based models, such as MobileViT [73] and EfficientFormer [74] which are able to overcome the computational intensive problem in computer vision task. For future work, we will focus on exploring and improving lightweight ViT based models in human action recognition.

Acknowledgements This study is supported by RIE2020 Industry Alignment Fund—Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contributions from Singapore Telecommunications Limited (Singtel), through Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU).

Author contributions Thean Chun Koh, Chai Kiat Yeo and Xuan Jing contributed to the conception of the technique. Material preparation, data collection, analysis, design and implementation have been performed by Thean Chun Koh who also wrote the first draft of the manuscript. The first draft and subsequent versions have been revised by Chai Kiat Yeo and commented by Xuan Jing and the final manuscript has been approved. Sunil Sivadas provided the initial research direction.

Availability of data and materials The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Consent for publication This submitted manuscript is the expansion of the conference paper publication listed as below Koh et al. [11].

References

1. Ziaeefard M, Bergevin R (2015) Semantic human activity recognition: A literature review. Pattern Recognition 48(8):2329–2345
2. Aggarwal JK, Ryoo MS (2011) Human activity analysis: A review. Acm Computing Surveys (Csur) 43(3):1–43
3. Papadopoulos GT, Axenopoulos A, Daras P (2014) Real-time skeleton-tracking-based human action recognition using kinect data. In: International Conference on Multimedia Modeling, pp. 473–483. Springer
4. Kong Y, Fu Y (2022) Human action recognition and prediction: A survey. International Journal of Computer Vision 130(5):1366–1401
5. Zhang S, Wei Z, Nie J, Huang L, Wang S, Li Z (2017) A review on human activity recognition using vision-based method. Journal of healthcare engineering **2017**
6. Rodríguez-Moreno I, Martínez-Otzeta JM, Sierra B, Rodriguez I, Jauregi E (2019) Video activity recognition: State-of-the-art. Sensors 19(14):3160
7. Ke S-R, Thuc HLU, Lee Y-J, Hwang J-N, Yoo J-H, Choi K-H (2013) A review on video-based human activity recognition. Computers 2(2):88–131
8. Zhen X, Shao L (2016) Action recognition via spatio-temporal local features: A comprehensive study. Image and Vision Computing 50:1–13
9. Das Dawn D, Shaikh SH (2016) A comprehensive survey of human action recognition with spatio-temporal interest point (stip) detector. The Visual Computer 32(3):289–306
10. Zhang H-B, Zhang Y-X, Zhong B, Lei Q, Yang L, Du J-X, Chen D-S (2019) A comprehensive survey of vision-based human action recognition methods. Sensors 19(5):1005
11. Koh TC, Yeo CK, S VU, Jing X (2022) Context-aware memory attention network for video-based action recognition. In: 2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), pp. 1–5. https://doi.org/10.1109/IVMSP54334.2022.9816216
12. Heilbron FC, Escorcia V, Ghanem B, Niebles JC (2015) Activitynet: A large-scale video benchmark for human activity understanding. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 961–970
13. Li Y, Li Y, Vasconcelos N (2018) Resound: Towards action recognition without representation bias. In: Proceedings of the European Conference on Computer Vision (ECCV)
14. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T (2011) Hmdb: a large video database for human motion recognition. In: 2011 International Conference on Computer Vision, pp. 2556–2563. IEEE
15. Soomro K, Zamir AR, Shah M (2012) Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402
16. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems **27**
17. Wan Y, Yu Z, Wang Y, Li X (2020) Action recognition based on two-stream convolutional networks with long-short-term spatiotemporal features. IEEE Access 8:85284–85293
18. Zhu Y, Lan Z, Newsam S, Hauptmann A (2018) Hidden two-stream convolutional networks for action recognition. In: Asian Conference on Computer Vision, pp. 363–378. Springer
19. Piergiovanni A, Ryoo MS (2019) Representation flow for action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9945–9953 (2019)
20. Sun D, Roth S, Lewis JP, Black MJ (2008) Learning optical flow. In: European Conference on Computer Vision, pp. 83–97. Springer
21. Sevilla-Lara L, Liao Y, Güney F, Jampani V, Geiger A, Black MJ (2018) On the integration of optical flow and action recognition. In: German Conference on Pattern Recognition, pp. 281–297. Springer
22. Horn BK, Schunck BG (1981) Determining optical flow. Artificial intelligence 17(1–3):185–203
23. Zach C, Pock T, Bischof H(2007) A duality based approach for realtime tv-l 1 optical flow. In: Joint Pattern Recognition Symposium, pp. 214–223. Springer
24. Sun D, Roth S, Black MJ (2010) Secrets of optical flow estimation and their principles. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2432–2439. IEEE
25. Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, Van Der Smagt P, Cremers D, Brox T (2015) Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2758–2766
26. Ilg E, Mayer N, Saikia T, Keuper M, Dosovitskiy A, Brox T (2017) Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2462–2470
27. Sun D, Yang X, Liu M-Y, Kautz J (2018) Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8934–8943 (2018)

28. Ranjan A, Black MJ (2017) Optical flow estimation using a spatial pyramid network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4161–4170
29. Ng JY-H, Choi J, Neumann J, Davis LS (2018) Actionflownet: Learning motion representation for action recognition. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1616–1624. IEEE
30. Jhuang H, Gall J, Zuffi S, Schmid C, Black MJ (2013) Towards understanding action recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3192–3199
31. Sun S, Kuang Z, Sheng L, Ouyang W, Zhang W (2018) Optical flow guided feature: A fast and robust motion representation for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1390–1399
32. Ji S, Xu W, Yang M, Yu K (2012) 3d convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence 35(1):221–231
33. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4489–4497
34. Yang H, Yuan C, Li B, Du Y, Xing J, Hu W, Maybank SJ (2019) Asymmetric 3d convolutional neural networks for action recognition. Pattern recognition 85:1–12
35. Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M (2018) A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6450–6459
36. Feichtenhofer C (2020) X3d: Expanding architectures for efficient video recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 203–213
37. Carreira J, Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308
38. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI Conference on Artificial Intelligence
39. Diba A, Fayyaz M, Sharma V, Karami AH, Arzani MM, Yousefzadeh R, Van Gool L (2017) Temporal 3d convnets: New architecture and transfer learning for video classification. arXiv preprint arXiv:1711.08200
40. Huang G, Liu Z, Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
41. Zhu J, Zhu Z, Zou W (2018) End-to-end video-level representation learning for action recognition. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 645–650. IEEE
42. Feichtenhofer C, Fan H, Malik J, He K (2019) Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6202–6211
43. Graves A (2012) Long short-term memory. Supervised sequence labelling with recurrent neural networks, 37–45
44. Lee D, Lim M, Park H, Kang Y, Park J-S, Jang G-J, Kim J-H (2017) Long short-term memory recurrent neural network-based acoustic model using connectionist temporal classification on a large-scale training corpus. China Communications 14(9):23–31
45. Cheng J, Dong L, Lapata M (2016) Long short-term memory-networks for machine reading. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 551–561
46. Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2625–2634
47. Ullah A, Ahmad J, Muhammad K, Sajjad M, Baik SW (2017) Action recognition in video sequences using deep bi-directional lstm with cnn features. IEEE access 6:1155–1166
48. Graves A, Fernández S, Schmidhuber J (2005) Bidirectional lstm networks for improved phoneme classification and recognition. In: International Conference on Artificial Neural Networks, pp. 799–804. Springer
49. Sydorov V, Alahari K, Schmid C (2019) Focused attention for action recognition. In: BMVC 2019-British Machine Vision Conference, pp. 1–13
50. Jiang M, Pan N, Kong J (2020) Spatial-temporal saliency action mask attention network for action recognition. Journal of Visual Communication and Image Representation 71:102846
51. Meng L, Zhao B, Chang B, Huang G, Sun W, Tung F, Sigal L (2019) Interpretable spatio-temporal attention for video action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 0–0
52. Zheng Z, An G, Wu D, Ruan Q (2020) Global and local knowledge-aware attention network for action recognition. IEEE Transactions on Neural Networks and Learning Systems 32(1):334–347
53. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473
54. Guo M-H, Xu T-X, Liu J-J, Liu Z-N, Jiang P-T, Mu T-J, Zhang S-H, Martin RR, Cheng M-M, Hu S-M (2022) Attention mechanisms in computer vision: A survey. Computational visual media 8(3):331–368
55. Mnih V, Heess N, Graves A, et al (2014) Recurrent models of visual attention. Advances in neural information processing systems **27**
56. Ba J, Mnih V, Kavukcuoglu K (2014) Multiple object recognition with visual attention. arXiv preprint arXiv:1412.7755
57. Zagoruyko S, Komodakis N (2016) Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928
58. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7794–7803
59. Wang Q, Wu B, Zhu P, Li P, Zuo W, Hu Q (2020) ECA-Net: Efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11534–11542
60. Misra D, Nalamada T, Arasanipalai AU, Hou Q (2021) Rotate to attend: Convolutional triplet attention module. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3139–3148
61. Wang H, Wang W, Liu J (2021) Temporal memory attention for video semantic segmentation. In: 2021 IEEE International Conference on Image Processing (ICIP), pp. 2254–2258. IEEE
62. Long F, Qiu Z, Pan Y, Yao T, Luo J, Mei T (2022) Stand-alone inter-frame attention in video models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3192–3201
63. Hao Y, Wang S, Cao P, Gao X, Xu T, Wu J, He X (2022) Attention in attention: Modeling context correlation for efficient video classification. IEEE Transactions on Circuits and Systems for Video Technology 32(10):7120–7132
64. Guo M-H, Lu C-Z, Liu Z-N, Cheng M-M, Hu S-M (2023) Visual attention network. Computational Visual. Media 9(4):733–752
65. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Gool LV (2016) Temporal segment networks: Towards good practices for deep

action recognition. In: European Conference on Computer Vision, pp. 20–36. Springer
66. Zhou B, Andonian A, Oliva A, Torralba A (2018) Temporal relational reasoning in videos. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 803–818
67. Jiang B, Wang M, Gan W, Wu W, Yan J (2019) Stm: Spatiotemporal and motion encoding for action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2000–2009
68. Lin J, Gan C, Han S (2019) Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7083–7093
69. Liu Z, Wang L, Wu W, Qian C, Lu T (2021) Tam: Temporal adaptive module for video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13708–13718
70. Yang C, Xu Y, Shi J, Dai B, Zhou B (2020) Temporal pyramid network for action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 591–600
71. Zhang X, Xu C, Tao D (2020) Context aware graph convolution for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14333–14342
72. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al (2020) An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
73. Mehta S, Rastegari M (2021) Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. arXiv preprint arXiv:2110.02178
74. Li Y, Yuan G, Wen Y, Hu J, Evangelidis G, Tulyakov S, Wang Y, Ren J (2022) Efficientformer: Vision transformers at mobilenet speed. Advances in Neural Information Processing Systems 35:12934–12949