3D Causal ShuffleNet in Dual Paths for Efficient Action Recognition

Siyuan Zhao
University of Toronto
syuan.zhao@mail.utoronto.ca

Kaitian Zheng
University of Toronto
kaitian.zheng@mail.utoronto.ca

Abstract

This paper presents a novel approach to video action recognition using a 3D Causal ShuffleNet model. We address significant challenges in computational demand and memory usage encountered by conventional 3D Convolutional Neural Networks (CNNs) in this domain, especially when deployed on mobile devices or for real-time video analysis. Our proposed solution amalgamates several key techniques: the stream buffer strategy from MoViNet, which optimizes memory usage by storing only essential video segments; the efficient group convolution and channel shuffle layers from ShuffleNet, enhancing processing speed; and the dual-pathway architecture from the SlowFast network, separately analyzing temporal and spatial aspects of video for improved accuracy. The model is trained using the UCF101 dataset, known for its broad spectrum of realistic action videos. Anticipated outcomes include maintaining robust accuracy while significantly boosting resource efficiency, paving the way for more viable real-time and resource-constrained application deployments.

1 Introduction

Video action recognition involves classifying actions based on video segments which essentially is a form of video classification. It revolves around Spatio-temporal feature learning - learning spatial features (what people or objects are in the video) and temporal features (how these people or objects move in the video). The applications of action recognition range from surveillance systems to interactive gaming.

The field has seen substantial advancements with the advent of deep learning models, particularly in the realm of 3D Convolutional Neural Networks (CNNs). These models treat video as images with an additional dimension (time). However, the computational intensity of these models, especially in terms of memory and processing power, limits their deployment in resource-constrained environments like mobile devices or real-time video streaming.

Notably, models like I3D[1] and 3D ResNets[2] have achieved impressive accuracy in action recognition. They leverage deep 3D convolutional networks to extract rich spatiotemporal features from video frames, providing an in-depth understanding of action sequences. While effective, their high computational cost poses a challenge for real-time applications. This necessitates a model that maintains high accuracy while being lightweight enough for deployment on less powerful devices.

Our approach focuses on addressing this challenge by integrating the merits of three innovative models: MoViNet[3], ShuffleNet[4], and the SlowFast[5] network.

In particular, our project proposes to develop a novel 3D CNN model that amalgamates the stream buffer technique from MoViNet that can reduce memory requirements, the group convolution and channel shuffle technique of ShuffleNet that can increase computational efficiency, and the dual-pathway concept from the SlowFast network that can maintain high accuracy. This combination aims

to create a lightweight yet highly accurate model for action recognition, suitable for deployment in real-time applications on mobile devices and for processing live-stream videos.

2 Background and Related Work

The field of action recognition has witnessed various models which can be categorized into CNN-based and non-CNN-based models.

CNN-Based Methods. CNNs have achieved significant success in 2D convolution, especially post-2015 with the introduction of ResNet in image recognition and object detection. The idea of using CNN for video recognition is simple, i.e., treating videos as 3D images. However, the direct application of 3D convolution like C3D [6] presented challenges including a large number of parameters and high computation costs. Improvements in CNN methods led to three main types: 1) Adding temporal modeling capabilities to 2D convolution, such as Two-Stream CNNs [7] (typically using optical flow to model temporal motion features). 2) Simplifying 3D convolution, like P3D [8] and R(2+1)D [9] (both split the 3D convolution into separate spatial and temporal components), or I3D [1] (using pre-trained 2D convolutional kernels to initialize 3D kernels). 3) Exploration of stronger Spatio-temporal modeling capabilities like the SlowFast networks [5] mentioned above.

Other Methods. There are other models which have shown promising results in video action recognition. For example, TimeSformer [10] (Vision Transformer-Based Methods), Video MAE [11, 12] (Masked Autoencoders, one of the state-of-the-art models), and Omnivore [13] (Multimodal Methods).

While non-CNN models show promising results in specific scenarios, their high computational demands for training and inference present significant challenges. Given these constraints, we redirect our efforts toward refining the SlowFast network, a CNN-based model noted for its strong performance. The SlowFast network stands out due to its unique dual-pathway architecture, which processes video frames at slow and fast rates separately, adept at capturing both spatial and temporal features. This architecture offers a solid foundation for further enhancements.

Our project enhances the SlowFast network, focusing on making it more suitable for environments with limited computational resources. By adopting the group convolution and channel shuffling techniques from ShuffleNet [4], we aim to reduce the computational overhead of the convolution operations while maintaining a robust feature extraction capability. Additionally, incorporating the stream buffer technique from MoViNet [3] reduces memory usage as this technique allows the model to only process a video clip at a time without the need for storing the whole video in memory. These modifications are designed to boost the overall efficiency of the SlowFast network, making it a more resource-friendly option for video action recognition tasks.

3 Data

3.1 Dataset Description

The UCF101 dataset, an extensive collection of realistic action videos sourced from YouTube, forms the foundation of our model's training process. With a total of 13,320 videos spanning 101 diverse action categories, this dataset presents an opportunity to develop a robust action recognition model capable of handling a wide array of real-world scenarios.

The dataset's richness comes from its variety; it includes a vast range of actions and the inherent complexities of natural settings, such as camera motion, varied object appearances, changes in scale, multiple viewpoints, background clutter, and fluctuating lighting conditions. This complexity makes UCF101 an ideal benchmark for an action recognition model.

UCF101's videos are further classified into 25 groups per category, with 4-7 videos each. These groups are designed to have commonalities such as similar backgrounds or viewpoints, which introduces the challenge of intra-class variation to our model. The actions encapsulated in the dataset are categorized into five major types, namely Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, and Sports. This categorization aids in the comprehensive coverage of possible actions and ensures that our model can generalize across various domains.

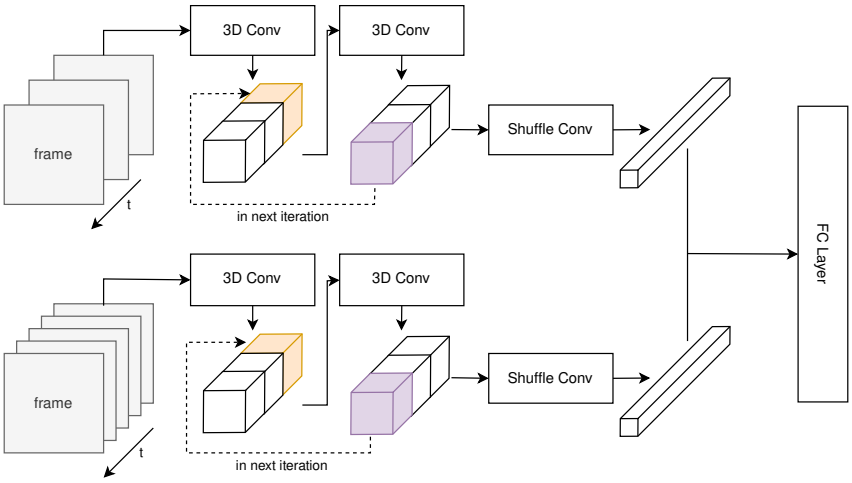

Figure 1: **Model Architecture**. The top pathway processes spatial information at a lower frame rate, employing a larger temporal stride, while the bottom pathway captures temporal information with a higher frame rate, using a smaller stride. Both pathways comment with a 3D convolution layer. A stream buffer is used between two convolution layers to maintain temporal context across subclips. The yellow cube represents the buffer that stores previous information, while the purple one represents the information to be passed to the next iteration. Subsequently, a shuffle convolution layer (Shuffle Conv) that performs grouped convolutions and channel shuffling is employed. Finally, the concatenated outputs of both pathways are fed into a fully connected layer (FC Layer) for classification.

The total length of these video clips is over 27 hours. All the videos have a fixed resolution of 320 × 240 but the frame rate varies. The average duration of the videos is around 10 sec. The shortest video has 28 frames (around 1.1 sec). Therefore, the longest clip of the video should not exceed 28 frames.

3.2 Data Preprocessing

Data Split. The dataset was divided into a training set (9537 videos) and a test set (3783 videos) according to the split annotation file available on the UCF101 official website. Subsequently, the training set was further partitioned in an 80:20 ratio for training and validation purposes.

Frame Extraction. Each video is broken down into its constituent frames.

Spatial Segmentation. To enhance the robustness of our model through data augmentation, each frame undergoes a random cropping process. Specifically, frames are cropped from four different corners and subsequently scaled to a uniform size of 112×112 pixels. Subsequently, pixel values are normalized to 0 mean and a standard deviation of 1.

Temporal Segmentation. Videos are segmented temporally into non-overlapping subclips such that each subclip contains $T = 16$ frames where T is a hyperparameter. This segmentation allows us to capture the full scope of actions while also considering the computational constraints. For instance, longer videos may be divided into multiple clips, ensuring that no video is too long to be processed in memory.

4 Model Architecture

Our proposed model integrates key elements from SlowFast, MoViNet, and ShuffleNet architectures, optimized for video processing that separately captures both spatial and temporal data aspects, as illustrated in Figure 1.

4.1 Dual Pathway Structure

The architecture employs a dual-pathway approach:

	# of parameters	# of computations
Ordinary Convolution	$k \times I \times O$	$k \times I \times O \times K$
Group Convolution	$k \times I \times O/g$	$k \times I \times O \times K/g$

Table 1: **A comparison between the group and ordinary convolution.** Suppose the kernel size is k which is the product of the kernel's height, width, and temporal depth. Let K represent the output size which is the product of the output's height, width, and temporal depth. Let I, O represent the number of input and output channels before splitting into groups respectively. Let g represent the number of groups. The number of parameters required for group convolution is obtained by $k \times (I/g) \times (O/g) \times g$. This is similar for calculating the number of computations.

Spatial Pathway: Focuses on capturing spatial details, such as objects and scenes within video frames. It utilizes a larger temporal stride, α, to emphasize spatial content over temporal changes. If $\alpha = 4$, for example, the spatial pathway essentially samples 1 frame out of every 4 frames. In other words, the spatial pathway processes T/α frames out of a subclip. The rationale behind employing a larger stride is to reduce the temporal resolution, thereby focusing more on individual frame contents and less on their temporal sequence. This strategy is effective for identifying static or slow-moving elements.

Temporal Pathway: Features a smaller temporal stride, β (we use $\alpha/\beta = 2$ during training), to capture rapid temporal changes. the spatial pathway processes T/β frames out of a subclip. This fine-grained temporal resolution is crucial for analyzing dynamic actions and rapid events.

The sampled frames are inputted into a regular 3D convolution layer (kernel: $12 \times (3, 3, 3)$, stride: (1, 2, 2), padding: 1), followed by batch normalization and ReLU activation.

4.2 Stream Buffer Integration

The output of the first convolution layer only contains the information from currently sampled frames. We adopt a stream buffer concept from MoViNet to maintain the temporal context learned from previous frames.

Buffer Concatenation: For each subclip's output z_i from the first convolution where i represents the index of the current subclip (the i-th subclip), a buffer B_i storing previous frames' information is concatenated along the temporal dimension, forming $z_i \oplus B_i$ as the input of the following Shuffle Convolution Layer, using $\oplus$ to represent the concatenation operation. Note that B_i has the same dimensions along the height and width axes so that the concatenation is feasible. The buffer mechanism is crucial for maintaining temporal context across frames, allowing the model to understand and interpret actions that unfold over time.

Buffer Update: The subsequent subclip's buffer B_{i+1} is updated with a portion of the following 3D convolution layer output. This Layer does not change the dimension along the height and width axes. Formally,

$$B_{i+1} = \text{3DConv}(z_i \oplus B_i)[: b]$$

where $[: b]$ represents slicing of the first b unit along the temporal dimension, and b represents the the sampled buffer. We use $b = 1$ in our model.

This approach facilitates memory-efficient training by requiring only a fixed-size subclip of the video in memory, rather than the entire video. It's particularly advantageous for devices with limited memory and for processing live-streaming videos, in which case, the video data arrives in sequence and future information is not available.

4.3 Shuffle Convolution Layer

The Shuffle Convolution Layer is central to the model's computational efficiency. A conventional convolution layer aggregates across all input channels, a process that can be resource-intensive when dealing with a high number of input channels. To mitigate this issue, a shuffle convolution layer implements the following steps.

Group Convolution: Input channels are divided into $g = 3$ groups (g is a hyperparameter) and 3D convolution is applied inside each group individually while the weights are shared. This method

significantly reduces computational load compared to traditional convolution methods. Table 1 shows a detailed comparison between these two convolution approaches. Specifically, group convolution requires fewer parameters and computations.

Channel Shuffle: A drawback of group convolution is that information is not shared between groups, potentially limiting the representational power of the network. This shuffle step rearranges channel orders to facilitate inter-group information flow, overcoming the isolation inherent in group convolutions.

Second Group Convolution: A subsequent group convolution follows the shuffle step. This convolution integrates the newly shared information, ensuring a comprehensive processing of the input data.

Activation: A ReLU activation immediately follows each group convolution.

4.4 Final Prediction Process

Pathway Integration and Prediction: The outputs of the last Shuffle Convolution Layer from both the spatial and temporal pathways are merged and fed into a fully connected layer. This layer outputs a prediction vector of length 101, representing the probability scores across different classes. The integration of both pathways ensures a holistic understanding of the video, capturing both the spatial and temporal nuances.

Prediction Mechanism: The prediction is made based on the portion of the video up to the latest sampled frames. We use the prediction after the last subclip is processed as the final prediction for the video.

4.5 Other Hyper-parameters

The details of kernel size and paddings are stated on our GitHub repository (end of this article). We use SGD optimization with a batch size of 16, an initial learning rate of 0.1, and a momentum of 0.9. The learning rate follows the `ReduceLROnPlateau` scheduler which divides the learning rate by 10 when there is no improvement in the loss for 10 epochs. The model is trained for 100 epochs.

5 Results

5.1 Performance Metrics

Our model's performance was evaluated using several key metrics: Top-1 Accuracy, Top-5 Accuracy, and a Confusion Matrix. These metrics provided a comprehensive understanding of the model's ability to accurately classify video actions.

Top-1 Accuracy: This metric reflects the percentage of test video clips for which the model's highest-probability prediction matches the true label. Our model achieved a Top-1 Accuracy of 70.12%, indicating a high level of precision in its predictions.

Top-5 Accuracy: Given the extensive variety of classes and the subtle nuances distinguishing similar actions, Top-5 Accuracy emerged as a crucial metric for our model. In datasets with many classes, particularly when some actions are markedly similar, such as "apply eye makeup" versus "apply lipstick" or "rafting" versus "ski jet", the likelihood of the model confusing these similar categories increases. Top-5 Accuracy, by considering a prediction accurate if the true label is among the model's top five predictions, offers a more lenient yet informative measure of the model's capability to correctly identify the action category, even if it is not the model's top prediction. The Top-5 Accuracy for our model was 91.51%, showcasing its proficiency in correctly categorizing actions, even in challenging scenarios involving closely related activities.

Confusion Matrix: To further dissect the model's performance across different categories, we utilized a confusion matrix, as illustrated in Figure 2. This detailed visualization helped us identify specific categories where the model excelled or struggled. Notably, the classes for "Horse Riding" (index 41), "Soccer Penalty" (84), and "Ice Dancing" (43) are predicted with exceptional accuracy. Conversely, certain classes such as "Yoyo" (100), "Archery" (2), and "Pizza Tossing" (57) show lighter shades on diagonal or off-diagonal clustering, indicating frequent misclassifications. "Yoyo"

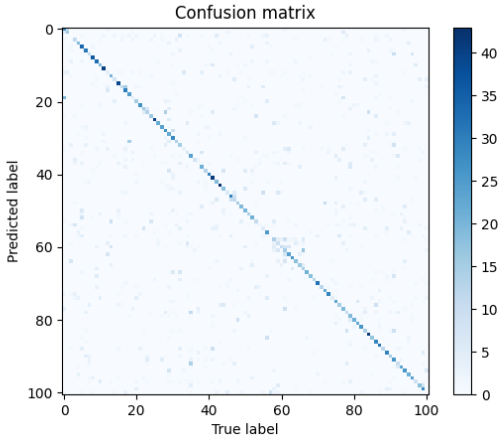

Figure 2: **Confusion Matrix of Model Predictions**. This confusion matrix visualizes the performance of our model across the 101 different action classes (0-indexed). The x-axis represents the true labels, while the y-axis corresponds to the predicted labels by the model.

is often confused with "Nunchucks" (55), "Archery" with "Bench Press" (9) or "Playing Sitar" (64), and "Pizza Tossing" with "Soccer Juggling" (83), suggesting that the model struggles to distinguish between these specific activities.

5.2 Benchmark

In evaluating the effectiveness of our SlowFastCausalShuffleNet model, we established a comprehensive benchmark, comparing it against a variety of both lightweight and state-of-the-art models, as shown in Table2.

Lightweight Models Comparison:

3D-ShuffleNetV1 (0.5x and 1.0x) and 3D-MobileNetV1 (1.0x) [14] serve as direct competitors to our architecture, due to similar design goals: achieving high accuracy while maintaining a small footprint in terms of the number of parameters. These models offer varying trade-offs between parameter count and accuracy (the number inside the parameters indicates a scale of network size).

3D-ShuffleNetV1 (0.5x) additionally serves as our baseline model, due to its direct lineage to our model. Our model inherits the efficient design principles of ShuffleNet and extends them. By modifying the original architecture, we aimed to enhance its performance while maintaining its lightweight nature. By comparing to 3D-ShuffleNetV1 (0.5x), we illustrate the incremental improvements our modifications provide.

State-of-the-Art Models Comparison:

In addition to these lightweight models, we benchmarked against more complex, state-of-the-art models such as ResNet-50 [14], TwoStream I3D [1], and R(2+1)D-TwoStream [9] (pre-trained on another action recognition dataset Kinetics). The purpose of comparing our model with these advanced architectures is twofold: 1) To demonstrate how our model performs in relation to the current best-performing models in the field. 2) To provide a perspective on the trade-offs between model complexity and performance, particularly highlighting how a significantly smaller model like ours compares in accuracy to these larger, more complex systems.

6 Discussion

Interpretation of Performance. Our SlowFastCausalShuffleNet, an evolution of the 3D ShuffleNetV1 0.5x, achieves a commendable balance between efficiency and accuracy, with a Top-1 Accuracy of 70.12% and a Top-5 Accuracy of 91.51%. These performance indicators are notable, especially considering the model's lightweight architecture with only 0.61 million parameters.

Model	Params (Millions)	Accuracy (%)	Input Size ($T \times H \times W$)
SlowFastCausalShuffleNet	0.61	70.12	$16 \times 112 \times 112$
Baseline model			
3D-ShuffleNetV1 0.5x	0.55	64.39	$16 \times 112 \times 112$
Other lightweight models			
3D-ShuffleNetV1 1.0x	1.52	76.00	$16 \times 112 \times 112$
3D-MobileNetV1 1.0x	3.91	70.95	$16 \times 112 \times 112$
State-of-the-art models			
ResNet-50	44.54	88.92	$16 \times 112 \times 112$
TwoStream I3D	25	93.4	$64 \times 224 \times 224$
R(2+1)D-TwoStream (pretrained)	33.3	97.3	$32 \times 112 \times 112$

Table 2: **Comparative Performance Benchmarking**. This table presents a comparison of different action recognition models, illustrating each model's complexity via the parameter count (in millions), the accuracy achieved in percentage, and the input size used during training and testing (denoted as T for the number of frames in time, H for height, and W for width of the frame).

The presence of a clear diagonal line in the confusion matrix also indicates that the model frequently predicts the correct labels for each class. It becomes clear that our model excels in classifying distinct activities such as "horse riding" "Soccer Penalty" and "ice dancing" which are actions with characteristic movements that our model can effectively capture and distinguish. However, the confusion matrix also provides critical insights into the model's limitations which will be discussed in the next section.

The dual-pathway architecture of our model is pivotal to its success. While the slow path introduces additional parameters, its primary focus is on capturing spatial details over longer temporal spans, thereby enriching the model's spatial acuity. The fast path, on the other hand, is streamlined for quick temporal processing. This division allows our model to robustly discern spatial and temporal features, contributing to a noteworthy accuracy improvement on the test set.

By comparing our model to similar lightweight models, we have shown that our model improvements are substantive. This is evident from the achieved accuracy, which is higher than the baseline 3D-ShuffleNetV1 (0.5x) model while barely increasing the complexity. Our model remains less complex than the heavier 3D-ShuffleNetV1 (1.0x) and 3D-MobileNetV1 (1.0x) models but achieves similar accuracy.

When positioned against state-of-the-art models like ResNet-50, TwoStream 3D, and the pre-trained R(2+1)D TwoStream, our model exhibits a notable divergence. These leading models, especially the R(2+1)D TwoStream, achieve significantly higher accuracy. These models often benefit from deeper layers, larger input sizes, and more sophisticated temporal feature processing, which collectively enable them to set the benchmark for accuracy.

Reflections of the model. While the SlowFastCausalShuffleNet does not outperform the state-of-the-art models in accuracy, it serves a distinct niche within the action recognition domain. It is particularly well-suited for applications where computational efficiency is paramount, offering a viable solution that is both practical and effective. The model's design, which prioritizes a balance between parameter count and performance, makes it an attractive option for real-time applications or devices with limited computational power.

In the broader context, our model's performance invites a discussion on the trade-offs between the depth and complexity of a neural network and its applicability in various scenarios. For cutting-edge research and applications where accuracy is the sole criterion, state-of-the-art models with larger parameter counts and deeper architectures may be preferred. However, for deployment in resource-constrained environments, the efficiency of a model like SlowFastCausalShuffleNet could be more desirable, especially when considering the diminishing returns on accuracy past a certain threshold.

7 Limitations

Our model's performance is closely tied to the diversity and representativeness of the training dataset. The UCF101 dataset, while a standard benchmark, does not encompass a wide array of scenarios. It is particularly limited in capturing the complexity of actions that are intrinsically tied to variable backgrounds. This limitation could potentially reduce the model's ability to generalize to real-world scenarios where the background context plays a crucial role in action recognition.

Moreover, the discrepancy between training and test accuracies suggests a degree of overfitting. This indicates that while the model has learned to identify patterns within the training data, it may struggle to maintain this performance on previously unseen data. The dataset also lacks actions that have temporal asymmetry — such as distinguishing between ascending and descending motions in jumping, or the nuances between catching and passing a ball. Since our causal component is designed to understand temporal sequences, the absence of such asymmetric actions in the dataset means this feature is not fully leveraged.

The confusion matrix shows our model has some tendencies to predict certain classes and to misclassify certain classes. This bias could lead to problems if it happens to be the action we particularly need to recognize.

To address these issues, we could opt for a larger and more diverse dataset if computation and storage resources allow. We could also employ techniques such as Gradient-weighted Class Activation Mapping (Grad-CAM) to visualize which regions of the input our model deems important. This could offer insights into whether the model is focusing on relevant features or if it is unduly influenced by background cues. For important classes, we could feed into more data from the class or implement an active learning schema.

8 Ethical Considerations

Privacy Concerns: Our models are designed to identify various human activities, many of which may represent individuals' hobbies. If the model is employed for monitoring purposes, there is a risk that data detailing people's hobbies could be harvested and commodified, potentially being sold to corporations that specialize in targeted marketing or recommendation algorithms.

Security Risks: The model can be adapted to recognize violations or suspicious activities (maybe through transfer learning). If the model lacks robustness against adversarial attacks, some deliberate manipulations could cause the model to either overlook or incorrectly interpret specific actions, potentially leading to significant security lapses.

Discrimination and Bias: If the UCF101 dataset has inherent biases, such as underrepresentation of certain groups or overemphasis on specific types of actions, the model could potentially reinforce these biases when deployed, leading to discriminatory outcomes, especially in sensitive applications like surveillance.

9 Conclusion

In this work, we introduced the SlowFastCausalShuffleNet, an innovative approach to action recognition that refines the architecture of 3D ShuffleNetV1 0.5x by integrating a causal temporal component and optimizing the network's layers. This model eschews the traditional reliance on deeper networks in favor of a dual-pathway structure that emphasizes efficiency and causal sequence processing.

We are optimistic about the prospects of efficient, causally-aware models in the domain of video understanding and anticipate applying these principles to a broader array of tasks. Future work will aim to expand the capabilities of our model to address the challenges presented by more complex and temporally asymmetric actions and to diversify the scenarios and backgrounds represented in training datasets.

The code we used to train and evaluate our models is available at `https://github.com/Hewitt6/3D-Causal-ShuffleNet-in-Dual-Paths-for-Efficient-Action-Recognition`

References

[1] J. Carreira and A. Zisserman, *Quo vadis, action recognition? a new model and the kinetics dataset*," 2018.

[2] C. Feichtenhofer, A. Pinz, and R. P. Wildes, *Spatiotemporal residual networks for video action recognition*," 2016.

[3] D. Kondratyuk, L. Yuan, Y. Li, L. Zhang, M. Tan, M. Brown, and B. Gong, *Movinets: Mobile video networks for efficient video recognition*," 2021.

[4] X. Zhang, X. Zhou, M. Lin, and J. Sun, *Shufflenet: An extremely efficient convolutional neural network for mobile devices*," 2017.

[5] C. Feichtenhofer, H. Fan, J. Malik, and K. He, *Slowfast networks for video recognition*," 2019.

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, *Learning spatiotemporal features with 3d convolutional networks*," 2014.

[7] K. Simonyan and A. Zisserman, *Two-stream convolutional networks for action recognition in videos*," 2014.

[8] Z. Qiu, T. Yao, and T. Mei, *Learning spatio-temporal representation with pseudo-3d residual networks*," 2017.

[9] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, *A closer look at spatiotemporal convolutions for action recognition*," 2018.

[10] G. Bertasius, H. Wang, and L. Torresani, *Is space-time attention all you need for video understanding?*" 2021.

[11] Z. Tong, Y. Song, J. Wang, and L. Wang, *Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training*," 2022.

[12] C. Feichtenhofer, H. Fan, Y. Li, and K. He, *Masked autoencoders as spatiotemporal learners*," 2022.

[13] R. Girdhar, M. Singh, N. Ravi, L. van der Maaten, A. Joulin, and I. Misra, *Omnivore: A single model for many visual modalities*," 2022.

[14] O. Köpüklü, N. Kose, A. Gunduz, and G. Rigoll, *Resource efficient 3d convolutional neural networks*," 2021.