Knowledge-Based Systems 320 (2025) 113594

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A Survey of Video Action Recognition Based on Deep Learning

Ping Gong, Xudong Luo *

School of Computer Science and Engineering, Guangxi Normal University, Guilin, 541004, China
Guangxi Key Lab of Multi-source Information Mining, Guangxi Normal University, Guilin, 541004, China
Education Ministry Key Lab of Education Blockchain and Intelligent Technology, Guangxi Normal University, Guilin, 541004, China

ARTICLE INFO

Keywords:
Video action recognition
Deep learning
Multi-modal learning
AI-powered human behaviour analysis
Action recognition benchmark dataset

ABSTRACT

Video Action Recognition (VAR) involves identifying and classifying human actions from video data. Deep Learning (DL) has revolutionised VAR, significantly enhancing its accuracy and efficiency. However, large-scale practical applications of VAR using DL remain limited, underscoring the need for further research and innovation. Thus, this survey provides a comprehensive overview of recent advancements in DL-based VAR. Specifically, we summarise the key DL architectures for VAR, including two-stream networks, 3D-CNNs, RNNs, LSTMs, and Attention Mechanisms, and analyse their strengths, limitations, and benchmark performances. The survey also explores the diverse applications of DL-based VAR, such as surveillance, human–computer interaction, sports analytics, healthcare, and education, while presenting a detailed summary of commonly used datasets and evaluation metrics. Moreover, critical challenges, such as computational demands and the need for robust temporal modelling, are identified, along with potential future directions. This paper is a valuable resource for researchers and practitioners striving to advance VAR using DL techniques by systematically presenting concepts, methodologies, and trends.

1. Introduction

Action Recognition (AR) integrates computer vision and Machine Learning (ML) [1,2] within Artificial Intelligence (AI) to identify and classify human actions, activities, or behaviours [3–5]. AR systems are broadly categorised as sensor-based or vision-based [6]. Sensor-based AR involves recognising actions using data captured from various sensors (wearable or embedded) rather than relying on visual input. In contrast, vision-based AR processes visual input such as images, skeletons, and videos. As a subset of vision-based AR, Video AR (VAR) focuses on analysing video data to recognise ongoing actions [7–9]. The main types of VAR methods are:

- *RGB-based methods*: Process raw RGB (Red, Green, and Blue) video frames (or sometimes optical flow) as input to deep neural networks.
- *Pose-based methods*: Use human pose estimation techniques to extract skeleton data and classify actions.
- *Hybrid methods*: Combine pose-based and RGB-based approaches to improve performance.

In this paper, we focus on RGB-based methods for VAR, while also discussing pose-based methods.

VAR systems classify actions into several categories varying in complexity, context, and interaction [10–12]. (1) *Gestures*: Simple, repetitive movements such as hand waves or nods, essential for sign language interpretation and Human–Computer Interaction (HCI). (2) *Atomic actions*: Basic movements like jumping, running, or sitting, characterised by short durations. (3) *Object interactions*: Actions involving objects, such as typing or driving, requiring the integration of human motion and object recognition. (4) *Person-to-person interactions*: Activities involving two or more individuals, such as shaking hands or playing sports. (5) *Group activities*: Complex actions involving multiple participants, such as sports events or crowd dynamics. (6) *Daily activities*: Routine behaviours like cooking or shopping. (7) *Anomalous Actions*: Unusual or dangerous actions, such as fighting or falling, critical in surveillance and security contexts. Each category presents unique challenges due to variations in complexity, duration, and contextual factors.

The main components of VAR systems are follows: (1) *Motion analysis*: Detecting movement patterns and trajectories. (2) *Feature extraction*: Identifying relevant visual features such as shape, colour, and motion. (3) *Temporal analysis*: Understanding how features evolve over time. (4) *Classification*: Using ML or Deep Learning (DL) [13] to assign action labels. (5) *Contextual understanding*: Considering environmental

* Corresponding author at: School of Computer Science and Engineering, Guangxi Normal University, Guilin, 541004, China.
E-mail address: luoxd@mailbox.gxnu.edu.cn (X. Luo).

https://doi.org/10.1016/j.knosys.2025.113594
Received 30 September 2024; Received in revised form 18 March 2025; Accepted 18 April 2025
Available online 30 April 2025

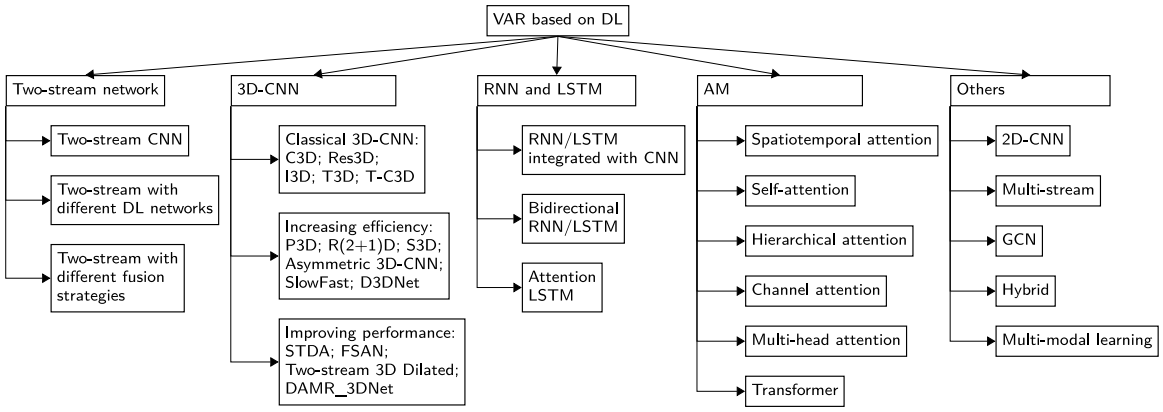

Fig. 1. The taxonomy of DL-based VAR.

factors and multi-entity interactions. (6) *Object and scene recognition*: Using scene elements to enhance action understanding. (7) *Human pose estimation*: Tracking human body parts for actions involving humans. (8) *Real-time processing*: Enabling live-AR. (9) *Data fusion*: Integrating multi-modal data, such as video with audio.

VAR techniques range from traditional handcrafted feature-based methods (*e.g.*, [14–16]) to advanced DL approaches (*e.g.*, [17–19]). Conventional methods often rely on predefined spatiotemporal patterns in video sequences, limiting their adaptability to complex scenarios. DL, by contrast, automatically learns hierarchical and discriminative features directly from raw data, enabling superior performance and robustness.

Applying DL to VAR has become increasingly popular due to several compelling reasons. (1) *Hierarchical feature learning*: DL models, particularly Convolution Neural Networks (CNNs) [2,20,21], can automatically learn hierarchical representations from raw data, eliminating the need for handcrafted features. This ability enables the models to capture more complex and discriminative features, leading to better VAR performance. (2) *Temporal modelling*: DL architectures (*e.g.*, Recurrent Neural Networks (RNNs) [22,23] and Long Short-Term Memory (LSTM) [24,25]) can model temporal dependencies in video sequences. This is crucial for VAR as actions often involve a series of movements over time. (3) *End-to-end learning*: DL pipelines process raw video inputs directly to output action classifications, minimising errors from intermediate steps. (4) *Scalability*: DL efficiently handles large-scale datasets, improving generalisation and resilience across diverse data conditions. (5) *Transfer learning*: Pre-trained DL models can be fine-tuned for VAR tasks using domain-specific datasets, reducing the need for extensive labelled data. (6) *Joint spatiotemporal learning*: Architectures like 3D-CNNs and two-stream networks combine spatial and temporal information, which is crucial for AR. (7) *Benchmark performance*: DL consistently outperforms conventional methods in VAR benchmarks, proving its utility in practical applications. These reasons, among others, have driven the widespread adoption of DL in VAR, leading to significant advancements and more robust, accurate VAR systems.

VAR's influence extends across industries:

- *Surveillance*: Detects suspicious behaviour or anomalies, improving safety and security.
- *HCI*: Enables intuitive gesture-based interfaces.
- *Sports analytics*: Facilitates performance tracking and strategic decision-making.
- *Healthcare*: Assists in patient monitoring and rehabilitation progress tracking.
- *Content management*: Automates video categorisation, improving retrieval and organisation.

Despite its advantages, DL-based VAR faces challenges, such as computational demands, limited interpretability, and the need for large labelled datasets. Addressing these limitations will drive further advancements in the field.

The field's vastness and rapid growth can be overwhelming for both experienced researchers and newcomers. Therefore, a survey of VAR based on DL is an essential tool for combining current knowledge, comparing different DL models, and identifying challenges and future directions to explore in this dynamic and important area of study. The survey is a valuable resource for those currently working in the field and those looking to join it, offering a solid base for developing efficient and effective VAR systems.

To this end, this paper systematically overviews key concepts, methodologies, and challenges associated with DL-based VAR. Specifically, we discuss the role of DL in advancing this field and explore various DL architectures, including two-stream networks, 3D-CNNs, RNNs & LSTM, and Attention Mechanisms (AMs). Comparative analyses highlight their advantages, limitations, and performance on benchmark datasets. Moreover, we analyse widely used benchmark datasets, evaluation metrics, and performance comparisons of different methods, emphasising the pros and cons of each approach. We also identify and discuss challenges and open research problems in DL-based VAR, offering insights into potential future research directions. This survey is valuable for researchers, practitioners, and enthusiasts seeking to understand VAR's current state and prospects with DL. For convenience, Table 1 summarises the abbreviations used in this survey.

Fig. 1 shows our taxonomy of DL-based VAR, which differs from related ones. Specifically, we classify mainstream DL-based VAR methods from the perspective of network structure into models based on two-stream networks, 3D-CNNs, RNNs and LSTMs, and AMs. Methods that do not fit these categories are grouped under "Others". Table 2 compares their strengths and weaknesses. In contrast, in 2021, Özyer, Ak, and Alhajj [26] classified existing AR methods into five categories: network-based, motion-based, multiple-instance learning-based, dictionary-based, and histogram-based methods. Additionally, in 2022, Pham et al. [27] categorised main AR using DL into four classes: CNN-based, RNN-LSTM-based, Deep Belief Network (DBN) based, and Stacked Denoising Autoencoder (SDA) based.

There are some surveys on DL for VAR, but ours in this paper is different from theirs.

- In 2020, Zhu et al. [29] presented a survey on DL for VAR, highlighting three key trends. The first introduces a second path for learning temporal information using a convolutional network on the optical flow stream, inspired by two-stream networks. The second trend utilises 3D convolution kernels to capture video dynamics, while the third focuses on computational efficiency for scaling to larger datasets. They also proposed benchmark datasets and evaluated the performance of various VAR methods.

Table 1
Abbreviations used in this paper and their expansions.

Abbreviation	Meaning	Abbreviation	Meaning
AI	Artificial Intelligence	ML	Machine Learning
AIA	Asynchronous Interaction Aggregation network	MLENet	Multi-Level Extraction Network
AM	Attention Mechanism	MM-ViT	Multi-Modal ViT
AP	Average Precision	MS-AAGCN	Multi-Stream Attention-enhanced Adaptive GCN
AR	Action Recognition	NAS	Neural Architecture Search
ARID	Action Recognition in the Dark	NLP	Natural Language Processing
AUC	Area Under the Curve	PCA	Principal Component Analysis
AUC-ROC	Area under the ROC curve	PCANet	Principal Component Analysis Network
AVA	Atomic Visual Actions	RAFT	Recurrent All-pairs Field Transforms
BiGRU	Bidirectional Gated Recurrent Unit	ReLU	Rectified Linear Unit
BiLSTM	Bidirectional LSTM	ResNet	Residual Network
BiRNN	Bidirectional RNN	RGB	Red, Green, and Blue
CATNet	Cascade multi-head Attention Network	RGB-D	RGB-Depth
CDAD	Common Daily Action Dataset	RNN	Recurrent Neural Network
CMA	Cross-Modality Augment	RHM	Robot House Multi-View
CNN	Convolution Neural Network	ROC	Receiver Operating Characteristic
CSAM	Channel and Spatiotemporal interest points Attention Model	SD	Standard Deviation
CSL	Chinese Sign Language	SDA	Stacked Denoising Autoencoder
CVPR	Computer Vision and Pattern Recognition	SimAM	Simple and parameter-free AM
DB-LSTM	Deep Bidirectional LSTM	SMC	Selective Motion Complement
DC-BiLSTM	Densely-connected BiLSTM	SOTA	State-Of-The-Art
DBN	Deep Belief Network	SP-LTN	Self-attention Pooling based Long-term Temporal Network
DL	Deep Learning	STA-CNN	Spatial-Temporal Attentive CNN
DSMHA	Dual Stream Multi-Head Attention	STAN	Spatiotemporal Attention Network
DWT	Discrete Wavelet Transform	STAR-3D	Student–Teacher Activity Recognition model based on 3D-CNN
FSAN	Frame and Spatial Attention Network	STA-TSN	Spatial-Temporal Attention TSN
FPR	False Positive Rate	STDA	Spatial Temporal Deformable 3D-CNN with Attention mechanism
FPV	First-Person Vision	STDAN	Spatial-Temporal Dual-Attention Network
GAN	Generative Adversarial Network	ST-D LSTM	Spatiotemporal Differential LSTM
GCN	Graph Convolutional Network	ST-GCN	Spatial-Temporal GCN
GRU	Gated Recurrent Unit	STILT	Spatial-Temporal Interaction Learning Two-stream
HAA	Human-centric Atomic Action	STIPs	Spatiotemporal Interest Points
HAN	Hierarchical Attention Network	STPN	Spatiotemporal Pyramid Network
HA-SSD	Hierarchical Attention Single-Shot Detector	SVM	Support Vector Machine
HCI	Human–Computer Interaction	S3D RAN	Spurious-3D Residual Attention Network
HM-AN	Hierarchical Multi-scale Attention Network	TAMNet	Two-level Attention Model based Network
HMM	Hidden Markov Model	TDN	Temporal Difference Network
HOG	Histogram of Oriented Gradients	TDS	Temporal Dense Sampling
HRC	Human-Robot Collaboration	TFREM	Temporal Feature Refinement Extraction Module
IoU	Intersection over Union	TPR	True Positive Rate
LKA	Large Kernel Attention	TS-LSTM	Temporal Segment LSTM
LRCN	Long-term Recurrent Convolutional Network	TSN	Temporal Segment Network
LSF	Long-short-term Spatiotemporal Feature	VAR	Video Action Recognition
LSTM	Long Short-Term Memory	VGG	Visual Geometry Group
mAP	Mean Average Precision	VLAD	Vector of Locally Aggregated Descriptor
MAT-EffNet	Multi-head Attention-based Two-stream EfficientNet	ViT	Vision Transformer
MEACI-Net	Attentive Cross-modal Interaction Network with Motion Enhancement	YOLO	You Only Look Once
MIL	Multiple Instance Learning		

- In 2021, Pareek and Thakkar [30] surveyed ML and DL techniques for AR from 2011 to 2019, discussing public datasets, method advances, applications, challenges, and future directions. They reviewed DL methods like CNNs, RNNs, LSTMs, DBNs, and Generative Adversarial Networks (GANs). Human actions can be represented using modalities such as RGB, skeleton, depth, infrared, point cloud, audio, acceleration, radar, and WiFi, each suited for different applications.
- In 2022, Sun et al. [31] presented a survey of recent progress of DL for AR based on the type of input data modality. They reviewed the current mainstream DL models for single and multiple data modalities, including the fusion-based and the co-learning-based frameworks, and also presented comparative results on several benchmark datasets for AR, together with future research directions.
- In 2023, Wang and Yan [32] surveyed RGB and skeleton-based AR models, focusing on DL models, feature extraction, public datasets, challenges, and future research directions. They also reviewed popular 2D and 3D pose estimation algorithms for skeleton-based AR models.
- In 2024, Karim et al. [33] reviewed ML, DL, and hybrid methods for sensor- and vision-based systems, focusing on healthcare, surveillance, sports, and HCI. They highlighted benchmark datasets, real-time processing, privacy, and the importance of multi-modal data fusion (*e.g.*, RGB, depth, skeleton). They also explored integrating human AR with augmented and virtual reality.

While these previous surveys on DL for VAR have explored trends, datasets, and broad methodologies, this paper distinguishes itself by:

- Providing an exhaustive analysis of DL architectures, including their strengths and weaknesses.
- Emphasising real-world applications and future research directions.
- Extending coverage to the latest studies up to 2025, offering a contemporary view of the field.

The rest of this paper is organised as follows. Section 2 reviews VAR models based on two-steam networks. Section 3 explores models based on 3D-CNNs. Section 4 summarises models based on RNNs and LSTM. Section 5 discusses models based on AMs. Section 6 examines VAR models based on other DL models. Section 7 discuss human pose estimation based VAR models. Section 8 outlines the applications of DL-based VAR. Section 9 briefs the commonly used datasets for training and testing DL-based VAR models. Section 10 summarises main metrics

Table 2
Comparison of strengths and weaknesses of different VAR methods.

Method	Strengths	Weaknesses
Two-stream networks	• Able to separately process spatial features from RGB frames and temporal features from optical flow [28], capturing a comprehensive view of video content. • Generally performs well on standard datasets due to the combination of spatial and temporal features.	• Needs substantial computational power for optical flow features, which may hinder real-time applications. • The two streams (spatial and temporal) are often not well integrated, possibly leading to a lack of cohesion between the extracted features.
3D-CNNs	• Efficiently learns spatiotemporal features in a unified framework, enhancing the representation of short-term motions. • Generally outperforms two-stream networks in capturing short-term motion patterns.	• Difficult to train due to a large number of parameters. • mainly captures short-term temporal information, potentially missing out on longer-term patterns and relationships.
RNNs/LSTM	• Effectively models long-range temporal dependencies in video sequences. • LSTM networks avoid the vanishing gradient problem in RNNs.	• Struggles to model spatial relationships effectively compared to CNN-based methods. • Tend to overfit with long sequences. • Generally outperformed by CNN-based methods when it comes to VAR.
AM	• Learns to focus on informative parts of the video. • Improves feature representations.	• The AM can be computationally expensive, particularly for complex video data. • Requires large training data for training to effectively learn which regions to focus on. • Performance depends on other base models.
Transformer	• Excels at modelling long-range dependencies due to its AM, making it ideal for tasks requiring the capture of complex temporal or sequential patterns. • Can handle variable-length input sequences and is a dominant architecture in both NLP and vision tasks.	• Transformer models can be very computationally intensive, especially when applied to large datasets or high-dimensional inputs. • Lack of inductive bias for sequential data, which might make Transformers less intuitive in some time-series tasks.
2D-CNNs	• Well-established architecture with good performance for image classification. • Easier to train than 3D counterparts.	• Incapable of modelling temporal information. • Often need to be combined with other methods.
Multi-stream Networks	• Combines multiple modalities like RGB, optical flow, audio, etc. • Provides complementary information.	• The architectures of multi-stream networks can be complex, making them challenging to design and train. • Difficult to fuse different streams effectively. • Redundancy across streams.
GCNs	• Capable of effectively handling graph-based data and capturing spatial relationships in a non-Euclidean structure. • GCNs can be used to model relationships between entities in a video or AR task, making them useful for representing spatial interactions between objects.	• The training of GCNs can be computationally expensive, especially for large-scale graphs. • GCNs may struggle to capture long-range dependencies due to their local receptive field. • GCNs are less effective in capturing sequential or temporal information compared to methods like RNNs or CNNs.
Hybrid	• Hybrid models can integrate the strengths of different architectures (*e.g.*, CNNs for spatial features and RNNs for temporal features). They can learn from diverse feature representations, improving the model's performance. • Useful in complex tasks where both short-term and long-term dependencies, as well as spatial and temporal features, are important.	• Hybrid models can be difficult to design, requiring careful consideration of how to fuse different models effectively. • Training hybrid models can be computationally expensive and time-consuming, especially if they require large-scale datasets. • The complexity of hybrid models can lead to overfitting, especially when combining many different types of architectures.
Multi-modal learning	• Integrating information from multiple modalities (*e.g.*, visual, audio, and text) can provide complementarity or reinforcement for a more comprehensive understanding of input data. • Multi-modal learning can enhance the robustness of models, making them less prone to errors when one modality fails or is noisy.	• Fusing multiple modalities effectively is challenging. • Multi-modal models often require significantly more data and computational resources to train, making them resource-intensive. • Need to align data from different modalities (*e.g.*, synchronising audio with video) can introduce additional complexity and errors.

for evaluating VAR models. Section 11 identifies challenges for future research. Finally, Section 12 concludes this paper.

2. Models based on two-stream networks

This section will brief the basics of VAR models based on two-stream networks and typical models of this kind, compare them, and discuss their limitations.

2.1. Seminal work

In 2014, Simonyan and Zisserman [34] proposed Two-stream CNN for the first time and applied it to VAR. This was inspired by the two-streams hypothesis [35], which posits that the human visual cortex contains two pathways:

- the dorsal stream, which recognises motion, and
- the ventral stream, responsible for object recognition.

The Two-stream CNN was thus designed to incorporate separate spatial and temporal networks. The spatial stream carries out AR using video frames, while the temporal stream is trained to recognise action based on motion, represented in the form of dense optical flow [28]. In the softmax layer, the prediction results from the two networks are combined through a process (known as late fusion), which uses either averaging or a linear Support Vector Machine (SVM) [36] to fuse the softmax scores. On the standard video actions benchmark datasets UCF101 [37] and HMDB51 [38], the model's accuracy shows that it exceeds a large margin previous attempts to use deep networks for VAR.

However, like any other research work, it has its limitations, such as:

1. The prediction results of the spatial stream network and the temporal stream network classifiers are simply fused.
2. Dense calculation and storage of optical flow are required before training, which incurs high computational and storage costs

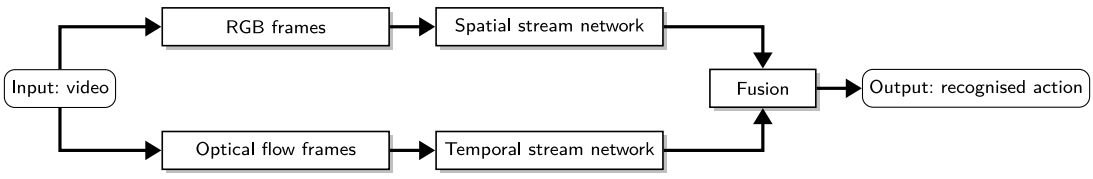

Fig. 2. General architecture of two-stream networks model.

and so hard for end-to-end training. This is not friendly for large-scale training or real-time deployment.

3. The two-stream architecture processes spatial and temporal information separately, and as such, it might not capture long-term temporal dependencies effectively. This limitation makes it challenging to recognise actions that require a more extended context for accurate recognition.

Since this work was published, more advanced models have been proposed, such as the Residual Network (ResNet) [39] and Inception [40,41]. These models outperform the original VGG (Visual Geometry Group) [42] used in the Two-stream CNN, in terms of accuracy and generalisation.

2.2. General architecture of two-stream networks

Video can naturally be decomposed into spatial and temporal components. The basic idea behind two-stream networks is to use two separate networks to extract spatial and temporal features from a video. Fig. 2 shows the general architecture of VAR models based on two-stream networks.

2.2.1. Working principle

The architecture of two-stream networks consists of two main components:

1. *Spatial stream network*: This component takes individual frames from the video and extracts spatial features that describe the appearance of objects and people in the frames. Typically, the network is a pre-trained CNN, such as VGG, ResNet, or Inception, which extracts high-level visual features. The output of the spatial network is a feature map representing spatial appearance information. This feature map is flattened and connected to fully connected layers to further capture higher-level semantic information.
2. *Temporal stream network*: This component takes sequences of frames and extracts temporal features that describe the motion of objects and people over time. It analyses the motion information between consecutive frames, often using optical flow. Optical flow represents the displacement of pixels between consecutive frames and captures the temporal dynamics in videos. Specifically, it provides a clear motion pattern of objects, surfaces, and edges in a visual scene caused by the relative motion between the observer and the scene. Optical flow can accurately describe the motion mode of each action. Compared to raw RGB images as input, optical flow effectively removes non-moving backgrounds, resulting in simpler learning problems.

To make a final prediction, the outputs of both streams are fused, or their features are combined at a higher convolutional layer in a process referred to as spatiotemporal fusion. Various fusion strategies can be employed, such as:

- *Early fusion* [43–45]: This strategy fuses the spatial and temporal streams at the input level. The RGB frames (spatial stream) and optical flow frames (temporal stream) are concatenated or stacked together and fed as input to the network.
- *Late fusion* [43–45]: In this approach, predictions or probability distributions obtained from both streams are concatenated and fed into another classifier for the final decision-making.
- *Attention-based fusion* [46]: This method applies AMs to selectively weight the contributions from each stream. The fusion process aims to leverage the complementary nature of spatial and temporal cues to improve overall recognition performance.

From a computational perspective, specific fusion methods include Average fusion, Max fusion, Concatenation fusion, Conv fusion, and Bilinear fusion [47].

2.2.2. Key characteristics

The two-stream network model uniquely captures complementary aspects of video data:

- *Spatial features*: Static details such as objects, backgrounds, and scene context, which are essential for understanding the setting.
- *Temporal features*: Dynamic cues from sequential frames, which are crucial for recognising motions and action flows.

The modularity of the two-stream network architecture enables separate processing of spatial and temporal data streams. This allows:

- specialising each stream to focus on its respective feature set, and
- jointly refining spatial and temporal features during end-to-end training.

Moreover, the architecture supports adaptability through advanced variants, such as:

- *Efficient network designs*: Integration of lightweight architectures (*e.g.*, MobileNet [48], EfficientNet [49]) reduces computational costs.
- *Temporal augmentations*: Using additional features like hidden motion cues or replacing optical flow for more efficient computations.
- *Advanced fusion techniques*: Dynamic or attention-based fusion strategies enhance the interplay between streams.

These characteristics make the two-stream network model a robust and scalable framework for VAR.

2.2.3. Applications in current research

The two-stream network model has been widely adopted in VAR tasks across various domains:

- *Surveillance*: Identifying anomalous or suspicious activities in real time, enhancing security and safety systems.
- *Sports analytics*: Analysing player movements, strategies, and performance in competitive sports.
- *Healthcare*: Monitoring patients during rehabilitation, detecting falls, or tracking physical therapy progress.
- *HCI*: Facilitating gesture-based control systems and enhancing interaction modalities.

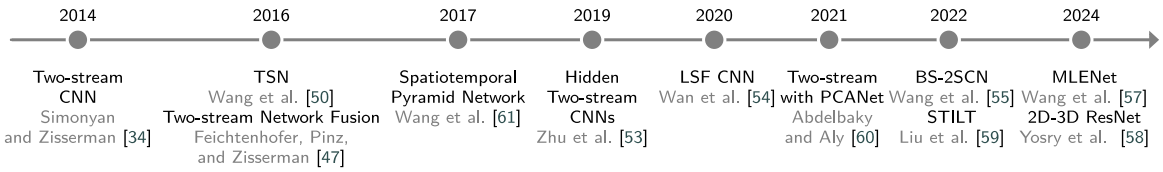

Fig. 3. Development timeline of two-stream networks.

2.3. Some variants of two-stream networks

To address the limitations of the seminal work, some variants of two-stream networks have been proposed. We will discuss some of them in this subsection.

2.3.1. Two-stream with different DL networks

Two-stream CNNs struggle with long-range temporal dependencies due to their limited temporal context. To address this, Wang et al. [50] proposed the Temporal Segment Network (TSN), which operates on sparsely sampled snippets from the entire video, improving upon the two-stream architecture. TSN achieves State-Of-The-Art (SOTA) results on HMDB51 (69.4%) and UCF101 (94.2%), further improved in their extended version [51] to 71.0%, 94.9%, and 89.6% on HMDB51, UCF101, and ActivityNet [52], respectively.

Dense calculation and storage of optical flow incur high computational and storage costs. Thus, some researchers tried to improve the calculation of optical flow, or abandoning optical flow and adopting other features. For example, in 2019, Zhu et al. [53] proposed a CNN architecture, called Hidden Two-stream CNNs, that implicitly captures motion information between adjacent frames. The network only takes raw video frames as input and directly predicts action classes without explicitly computing optical flow.

In 2020, Wan et al. [54] proposed a two-stream CNN with Long-short-term Spatiotemporal Features (LSF) for VAR tasks. This model consists of two sub-networks: the long-term spatiotemporal feature extraction network, which uses 3D-CNN to process RGB image sequences, and the short-term spatiotemporal feature extraction network, which employs 2D-CNN to process optical flow and estimates from two adjacent frames. Their method improves the accuracy of VAR by using long-short-term spatiotemporal information on standard benchmark datasets: UCF101 and HMDB51.

To improve the perception ability of appearance coherence features of the classical Two-stream CNN, in 2022, Wang et al. [55] proposed an improved Two-stream CNN with a Bidirectional Gated Recurrent Unit (BiGRU) and SimAM [56] (Simple and parameter-free AM), called BS-2SCN. The recognition mode of a single frame of a spatial stream is changed to multi-frame image recognition using BiGRU, which solves the shortcomings of a classical two-stream network in the perception of action appearance coherence features. The accuracy and stability of BS-2SCN had been improved on datasets UCF101 and HMDB51.

Two-stream networks often process spatial and temporal information separately, missing crucial correlations. In 2022, Liu et al. [57] addressed this with their Spatial-Temporal Interaction Learning Two-stream (STILT) network for VAR. STILT uses a spatiotemporal interaction learning with alternating AMs between its spatial and temporal streams. It outperformed previous VAR models on UCF101, HMDB51, and Kinetics [58].

To enhance the efficiency of temporal modelling in videos, in 2024, Wang et al. [59] proposed the Multi-Level Extraction Network (MLENet) for VAR. MLENet employs a two-stream architecture and a Temporal Feature Refinement Extraction Module (TFREM) to optimise spatiotemporal motion feature extraction. Unlike other VAR models that rely on optical flow, MLENet achieves superior performance with fewer inputs, consistently surpassing previous models on benchmark datasets, including Something-Something V1 & V2 [60], UCF101, and HMDB51.

2.3.2. Two-stream with different fusion strategies

A better fusion of the two streams enables a comprehensive understanding of human actions and enhances the discriminative power of the recognition model. Specifically, in 2016, Feichtenhofer, Pinz, and Zisserman [47] pointed out that rather than fusing at the softmax layer, a spatial and temporal network can be fused at a convolution layer without loss of performance but with a substantial saving in parameters. Thus, they proposed a model of two-stream network fusion for VAR, which does not significantly increase the number of parameters over previous models yet exceeds the SOTA on two standard benchmark datasets.

To create an efficient, end-to-end trainable VAR model with spatial and temporal streams, Wang et al. [64] proposed the Spatiotemporal Pyramid Network (STPN) in 2017. STPN fuses spatial (2D-CNN) and temporal (1D-CNN on optical flow) features in a pyramid structure, capturing both fine-grained and high-level information. Based on squeeze-and-excitation blocks, a composite gating mechanism fuses the learned features. By avoiding 3D convolutions, STPN achieves SOTA accuracy on UCF101 and HMDB51, outperforming previous handcrafted methods.

To fuse spatial and temporal information with CNN, in 2021, Abdelbaky and Aly [63] proposed a VAR model to fuse spatial and temporal features learned from a simple unsupervised CNN, called Principal Component Analysis Network (PCANet) in combination with bag-of-features and Vector of Locally Aggregated Descriptors (VLAD) encoding schemes. Feature fusion is used to obtain the spatiotemporal features and SVM classifier is employed for AR. On the datasets of KTH [65] and UCF sports [66], their model presents satisfactory and competitive results.

In 2024, Yosry et al. [62] examined the impact of various fusion techniques on recognition model accuracy for VAR. They proposed two frameworks: the first utilises 2D ResNet-101 [39] and LSTM for long-term spatiotemporal features from keyframe images, while the second employs 3D ResNet-101 [67] for short-term features from video clips. The early-fusion achieves 95.5% accuracy on dataset UCF101 and 70.1% for dataset HMDB51, while the late-fusion reached 95.5% on UCF101 and 77.7% on HMDB51, demonstrating competitiveness with previous models.

2.3.3. Comparison of the variants

The models reviewed in this section follow the timeline shown in Fig. 3. As shown in Table 3, their similarities and differences are mainly in spatial stream, temporal stream, and fusion strategies. For the spatial stream, the input mainly includes a single frame, video frames, and RGB images. For the temporal stream, the input mainly includes optical flow frames, video frames, and image sequences. In particular, the two streams in [54,59] both are spatiotemporal streams, unlike other two-stream models where one is a spatial stream and the other is a temporal stream.

2.4. Limitations and future work

The limitations of two-stream networks for VAR can be addressed via various strategies and advancements. These solutions span across the main key challenges in the domain.

Table 3
Comparison of the models based on two-stream networks.

Methods	Spatial stream		Temporal stream		Fusion strategies
	Input	Network	Input	Network	
Two-stream CNN [34]	Single frame	CNN-M-2048 architecture [61]	Optical flow frames	CNN-M-2048 architecture	Late fusion (averaging or a linear SVM)
Two-stream Network Fusion [47]	Single frame	VGG-16	Optical flow frames	VGG-16 or VGG-M	Two stream networks with a convolutional fusion layer between the networks, and a temporal fusion layer (incorporating 3D convolutions & pooling)
TSN [50]	RGB images or RGB difference	BN-Inception	Optical flow fields or Warped optical flow fields	BN-Inception	Late fusion (weighted average)
Hidden Two-stream CNNs [53]	Video frames	VGG-16	Video frames	Stacked temporal stream CNN (MotionNet+ traditional temporal CNN)	Late fusion (a ratio of 1:1.5 for spatial to temporal stream)
LSF CNN [54]	Stacked RGB images	A long-term *spatiotemporal* feature extraction network, C3D	Optical flow frames	A short-term *spatiotemporal* features extraction network, VGG-16	Fused in the fully-connected layer and feed the fusion features into an SVM
BS-2SCN [55]	Single frame	ResNet+SimAM +BiGRU	Multi-frame Optical flow	ResNet+SimAM	Late fusion (classical weighted fusion method)
MLENet [59]	RGB frames with various sampling rates	ResNet50	RGB frames with various sampling rates	ResNet50+TFREM	Late fusion (the traits from two paths are combined to create fusion features)
2D-3D ResNet [62]	Keyframe images	2D ResNet-101+LSTM	Video clips	3D ResNet-101	Late fusion/early fusion
STILT [57]	Video frames	A spatial network with a spatiotemporal interaction learning	Optical flow frames	A temporal network with a spatiotemporal interaction learning	Late fusion (adaptively weighted fusion)
Two-stream with PCANet [63]	Short-time motion energy image templates	PCANet	Image Sequence	PCANet	Feature and score fusion (SVM classifier)
Spatiotemporal Pyramid Network [64]	Single frame	CNN (such as BN-Inception, ResNets and VGG-16) with AM	Multiple optical flow frames with interval sampling	CNN (such as BN-Inception, ResNets and VGG-16) with AM	Pyramid fusion (spatiotemporal compact blinear operator)

2.4.1. Computational cost

Two-stream networks process both appearance and motion streams separately, which increases computational load. The need for optical flow estimation and subsequent feature fusion further amplifies this burden. The limitation can be addressed in the following ways:

- *Lightweight networks:* Use architectures like MobileNet or EfficientNet, which are designed to be computationally efficient while maintaining performance. These architectures reduce the number of parameters, thus lowering computational costs.
- *Distributed networks:* Using distributed processing across multiple machines or using cloud–edge systems can offload some of the computational tasks, especially for real-time VAR.

2.4.2. Temporal modelling

Optical flow is often noisy and can lead to errors, particularly in complex scenes with occlusions or fast-moving objects. Some ways to address the limitation are:

- *Improved optical flow estimation*: Networks like FlowNet [68] and RAFT (Recurrent All-pairs Field Transforms) [69] have significantly improved the accuracy of optical flow estimation, making motion tracking more robust in complex environments.
- *Keypoint-based motion estimation:* Instead of using full-frame optical flow, focus on key points or regions of interest, which are more robust to occlusions and noisy flow.
- *Temporal AMs*: Temporal AMs, such as self-attention or Transformer-based models, can better capture relevant temporal dependencies in noisy environments.

2.4.3. Fusion strategies

Determining the most effective fusion strategy (early, late, or multimodal) and optimally combining appearance and motion features remains a challenge. This limitation could be addressed in the following ways:

- *Dynamic fusion:* Adaptive fusion mechanisms allow the network to decide when and how to combine appearance and motion features depending on the context of the action or quality of the streams.
- *Multi-scale fusion:* Multi-scale fusion integrates features from both streams at different levels (*e.g.*, pixel level, feature level), which improves recognition accuracy.
- *Attention networks:* Cross-attention or temporal AMs guide the network to learn which parts of the appearance and motion streams are most relevant for a given action.
- *End-to-end learned fusion:* Fusion networks that are jointly optimised with the appearance and motion streams allow the network to learn the most effective fusion strategy directly from data.

2.5. Summary

Two-stream networks, using separate spatial and temporal networks, have become a highly effective methodology for VAR. The

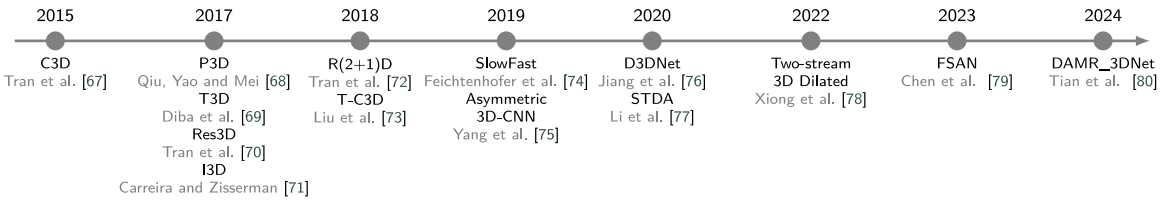

Fig. 4. Development timeline of 3D-CNNs.

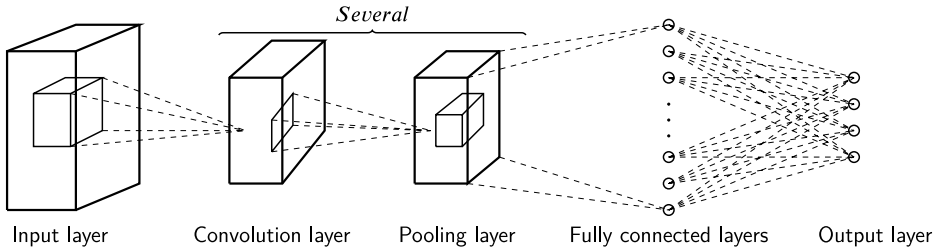

Fig. 5. General architecture of 3D-CNNs.

spatial network extracts visual features from individual video frames using 2D-CNN architectures like VGG or ResNet, capturing static scene information. Conversely, the temporal network uses a separate 1D-CNN to learn motion patterns across frames from a sequence of optical flow images, encoding the direction and magnitude of motion. Although the two streams are trained independently, their predictions are fused, enabling the networks to specialise and complement each other. Despite potential challenges, such as computational demand and complexities of end-to-end training, two-stream networks often outperform single-stream 3D-CNN, mainly due to their proficiency in independently learning appearance and motion features. As a result, they demonstrate strong performance on significant video action benchmarks, making them one of the most efficient techniques for VAR to date.

However, there are still several significant challenges. By addressing these challenges, researchers can further advance the capabilities of two-stream networks and contribute to more accurate, robust, and interpretable VAR systems.

3. Models based on 3D-CNNs

In this section, we will explore VAR models based on 3D-CNNs (another kind of the mainstream methods).

3.1. Seminal work

Conceptually, video data can be viewed as a 3D tensor containing two spatial and one temporal modes [70,71], so can 3D-CNNs be used for modelling spatiotemporal information in videos? To answer the question, in 2013, Ji et al. [72] applied the 3D convolution operation to extract spatial and temporal features from video data for VAR. This work is widely viewed as the seminal contribution to VAR using 3D-CNNs [73]. Since then, some classical models based on 3D-CNNs for VAR have emerged, *e.g.*, C3D [74], Res3D [75], T3D (Temporal 3D) [76], I3D [77], and T-C3D [78]. Fig. 4 shows the development timeline of the VAR models.

3.2. General architecture and components

Fig. 5 shows the general architecture of 3D-CNN-based models for VAR. Their general architecture and components can be summarised as follows:

1. *Input layer*: The input to a 3D-CNN is a three-dimensional tensor representing spatiotemporal or volumetric data (*e.g.*, consecutive video frames).
2. *Convolution layers*: Convolution layers are the core building blocks of 3D-CNN [79–81]. They consist of a set of filters/kernels that perform 3D convolutions across the spatial and temporal dimensions of the input tensor. 3D-CNNs directly extract spatiotemporal information from videos by using 3D convolutions [82,83]. After each convolution operation, an activation function can deal with non-linearity in the network. Common choices for activation functions are Rectified Linear Units (ReLU) [84,85] or variants such as Leaky ReLU [86] or Parametric ReLU [87].
3. *Pooling layers*: Pooling layers are used to downsample the feature maps, reducing the spatial and/or temporal dimensions of the data, and also help to reduce computational complexity [88–90]. Max pooling and Average pooling are typical operations.
4. *Fully connected layers*: One or more fully connected layers can be added following the convolution and pooling layers. These layers perform high-level feature aggregation and mapping to the output classes or regression targets [91,92]. The fully connected layers allow the network to learn complex relationships between features and make predictions. Dropout and regularisation techniques can be applied to prevent overfitting and improve generalisation. Dropout randomly sets a fraction of the input units to zero during training, while regularisation techniques like L_1 or L_2 regularisation impose penalties on the network's weights to control their magnitude.
5. *Output layer*: It is the final layer of the network responsible for producing the desired output. In classification tasks, the output layer often consists of softmax activation, providing probabilities for each class. The output layer can have a linear activation or another suitable activation function for regression tasks.

The architecture and depth of 3D-CNNs can vary depending on the specific task and dataset. Deeper architectures with multiple convolution layers and increased model complexity are commonly used to capture complex spatiotemporal patterns. Architectural variations, such as residual connections or AMs, can also be incorporated to enhance model performance.

3.3. Classical 3D-CNN based models for VAR

Classical 3D-CNNs (*e.g.*, C3D and Res3D) are based on the main network architectures (*e.g.*, VGG, ResNet, and DenseNet [93]), where 2D elements are replaced with 3D forms.

C3D is a general network. Its overall structure is quite similar to VGG-16 but replaces all the 2D convolution kernels with 3D convolution kernels [94]. Res3D is considered the first deep 3D ResNet in view of ResNet's ability to alleviate the degradation problem of network deepening. Res3D extends the 2D convolution of ResNet to full-3D convolutions with a depth of 18. T3D extended the DenseNet architecture with 3D filters and pooling kernels. T-C3D combined a residual 3D-CNN with a temporal encoding method to explore the temporal dynamics of the whole video. The basic idea of I3D is to "inflate" the 2D network Inception-V1 into a 3D network. And inspired by the basic idea of two-stream architecture, I3D has two CNNs based on pre-trained parameters, training video and optical flow, respectively.

These studies reveal that 3D-CNN can model complex spatiotemporal patterns, capture appearance and motion cues, and achieve SOTA performance on benchmark datasets.

3.4. Recent 3D-CNN based models for VAR

3.4.1. Increasing efficiency

Classical 3D-CNN models require complete videos as input and can be extended to large datasets. However, they are computationally expensive and require a lot of memory. For instance, it usually takes weeks to train C3D. Researchers are actively addressing these issues from various angles and developing new and improved models.

Some studies consider replacing 3D convolution with other methods to reduce computational complexity. In 2017, Qiu, Yao, and Mei [95] proposed the P3D (Pseudo-3D ResNet) network, which approximates 3D convolution with combined $1\times3\times3$ spatial and $3\times1\times1$ temporal convolutions, reducing computational cost and memory while maintaining similar performance. In 2018, Tran et al. [96] proposed the R(2+1)D network, factorised the 3D convolution filters into separate spatial and temporal components, and yielded significant gains in accuracy. In 2018, Xie et al. [97] replaced the I3D top layer with 2D convolutions and transformed the network with a separable spatiotemporal deconvolution to propose the S3D model. This model shows promising results in both speed and accuracy. In 2019, Yang et al. [98] proposed asymmetric one-directional 3D convolutions with MicroNets to approximate the conventional 3D convolution for VAR. The asymmetric 3D-CNN boasts fewer parameters and a reduced computational cost. It outperforms the most competitive 3D-CNN models and other previous models on UCF101 and HMDB51.

Some studies use multiple branches to process video data with different sampling rates. In 2019, Feichtenhofer et al. [99] proposed SlowFast, an efficient network with slow (low frame rate, semantic information) and fast (high frame rate, motion) paths, fused via lateral connections. The lightweight fast path improves efficiency. In 2020, Jiang et al. [82] proposed D3DNet, a dual 3D-CNN with two lightweight branches. A coarse branch (modified from C3D) uses fast temporal downsampling to maintain a large temporal receptive field. A fine branch progressively downsamples temporally and uses reduced-capacity 3D convolutions. The model balances speed and VAR performance.

3.4.2. Improving performance

3D-CNN models have shown promising performance at modelling the motion and appearance information. However, the fixed geometric structure of 3D convolution filters largely limits the learning capacity for VAR. Thus, some studies focus on improving the learning capacity.

In 2020, Li et al. [100] proposed a Spatial Temporal Deformable 3D-CNN with Attention mechanism (STDA) to capture the complex action variations. The AM uses long-range temporal dependencies across multiple frames and long-distance spatial dependencies inside each frame. The deformable 3D module captures temporal and spatial variations via flexible convolution filter offsets and exhibits fast computation and strong capacity for VAR feature extraction.

In 2022, Xiong et al. [101] proposed a VAR model using action sequence optimisation and a two-stream 3D dilated neural network. Optimised action sequences refine the video, increasing action feature ratios. The reconstructed video and pose flow then feed into the two-stream network. Class score fusion provides the final prediction. This approach combines two modalities, enlarges the action feature receptive field, and achieves competitive results on UCF101 and HMDB51.

In 2023, Chen et al. [102] proposed a Frame and Spatial Attention Network (FSAN). It first uses a spurious-3D-CNN to extract basic deep features and then use a two-level attention module to mine discriminative features between actions. Their model performs well on UCF101 and HMDB51.

In 2024, Tian et al. [103] proposed an AR method using keyframe selection and DAMR_3DNet to extract spatiotemporal features. Keyframe selection reduces computation, while DAMR_3DNet enhances C3D via 3D decoupled convolution and feature fusion. A 3D attention module emphasises action features, and a 3D residual structure prevents gradient vanishing. Their model outperforms previous models on UCF101, CSL [104], and HMDB51.

3.5. Comparison, limitations, and future work

Table 4 shows their similarities and differences mainly in input, basic architectures, and end-to-end trainability.

Some unique 3D-CNN models include two-stream networks: one input is video frames, and the other is optical flow. To highlight the difference, in this case, the video frames would be emphasised as RGB frames. In particular, In 2019, Yang et al. [98] defined an input type as RGBF frames, which fuse the valuable information in the RGB and Flow frames. The RGBF frame is generated by multiplying each channel of an RGB frame with the corresponding movement confidence map. Moreover, there are two parallel pathways in [101]. The input video is processed into RGB and skeleton sequences and then fed into these two pathways in parallel.

The above 3D-CNNs have some common disadvantages. (1) Large memory footprint, requiring much memory for training and inference. For resource-constrained devices, this can be a challenge. (2) The generalisation ability is limited and may not generalise well to different types of videos with different characteristics, such as different resolutions, frame rates, and lighting conditions. This may require retraining the model on a new dataset or using data augmentation techniques. (3) Most DL models lack interpretability, making their decisions difficult to be understood.

Some suggestions about how to improve 3D-CNNs for VAR are as follows:

1. Design more efficient 3D-CNN architectures to mitigate the high computational demand. For instance, mixed CNNs that use both 2D and 3D convolutions, or depth-wise separable convolutions, could help reduce the computational load without a significant compromise on performance.
2. Incorporate other models good at temporal modelling (like LSTM or Transformer) in conjunction with 3D-CNNs to capture long-term temporal dynamics.
3. Use pre-trained models on large datasets to initialise the network, which often leads to better performance and faster convergence.
4. Consider incorporating other forms of data (like audio or text) into the model. This can provide additional context and help improve the VAR performance.

These suggestions may provide starting points for future work. Innovative solutions often come from challenging existing paradigms and experimenting with new ideas.

Table 4
Comparison of the models based on 3D-CNNs.

Methods	Input	Basic architecture	End-to-end trainable
C3D [74]	Video frames	3D VGG-16-like or 3D AlexNet-like	✓
P3D [95]	Video frames	ResNet-like architecture	✓
T3D [76]	Video frames	Temporal Transition Layer+DenseNet3D architecture	✓
Res3D [75]	Video frames	3D-Resnet18 architecture	✓
I3D [77]	RGB frames and Optical flow	Two parallel pathway (The Inflated Inception-V1 architecture)	✗
R(2+1)D [96]	RGB frames and Optical flow	ResNets with (2+1)D convolutions	✗
T-C3D [78]	Video frames	3D ResNet+temporal encoding method	✗
SlowFast [99]	Video frames of high frame rate and low frame rate	Two parallel pathway (3D ResNet)	✓
Asymmetric 3D-CNN [98]	RGBF frames (fuse useful information of the RGB and Flow frames at the pre-processing stage)	MicroNets	✗
D3DNet [82]	Video frames	Two parallel pathway (Fine branch) +Coarse branch (C3D-like network)	✓
STDA [100]	Video frames	Temporal Deformable 3D-CNNs+Spatial Deformable 3D-CNNs (STDA-ResNeXt-101)	✓
Two-stream 3D Dilated [101]	RGB sequences+skeleton sequences	Two parallel pathway (3D Dilated CNN)	✗
FSAN [102]	Video frames	A spurious-3D-CNN and two-level attention module (Frame attention module and Spatial attention module)	✓
DAMR_3DNet [103]	Keyframe sequences	D3DNet+3D AM +3D Residual module	✗

Table 5
Architectural differences between 2D-CNN and 3D-CNN.

Feature	2D-CNN	3D-CNN
Convolutional filters	Operate on spatial dimensions (height, width)	Operate on spatial and temporal dimensions (height, width, depth)
Input data	Process single frames or images	Process sequences of frames or volumetric data
Output features	Capture spatial features	Capture both spatial and temporal features
Temporal Dependencies	Require additional mechanisms (*e.g.*, RNNs, LSTMs) for temporal modelling	Model temporal dynamics directly using 3D convolutions
Computational Complexity	Lower computational demand; faster to train.	Higher computational demand due to the temporal dimension
Memory Requirements	Require less memory	Require significantly more memory for training and inference

3.6. Comparative analysis of 2D- and 3D-CNN

Understanding the differences between 2D-CNNs and 3D-CNNs is essential for appreciating their respective roles in VAR. While 2D-CNNs are widely used for image-based tasks, 3D-CNNs extend their capabilities to spatiotemporal data, such as videos. This section provides a detailed comparison of their architectural differences, application scenarios, and performance characteristics.

3.6.1. Architectural differences

The main distinction between 2D-CNNs and 3D-CNNs lies in their architectural design and how they process data. A 2D-CNN operates on two-dimensional spatial data, such as individual video frames or static images, while a 3D-CNN extends this by adding a temporal dimension. This enables 3D-CNNs to process sequential data such as video clips or volumetric medical images. Table 5 summarises the key architectural differences between 2D-CNNs and 3D-CNNs.

The ability of 3D-CNNs to directly capture spatiotemporal patterns makes them particularly effective for tasks involving motion analysis, whereas the simpler structure of 2D-CNNs makes them more suitable for tasks where temporal modelling is less critical.

3.6.2. Application scenarios

The choice between 2D-CNNs and 3D-CNNs largely depends on the specific application and the nature of the data. Below, we discuss several key application scenarios where these models are employed:

- *Image classification*: 2D-CNNs are well-suited for static image classification tasks and are widely used in domains such as medical imaging, facial recognition, and general object detection. 3D-CNNs, however, are rarely used for single-image tasks due to their temporal focus.
- *VAR*: While 2D-CNNs process individual video frames, they require additional mechanisms, such as RNNs or attention models, to handle temporal dependencies. In contrast, 3D-CNNs can capture spatiotemporal features directly, making them the preferred choice for recognising complex actions in videos.
- *Medical imaging*: In tasks like tumour segmentation or volumetric brain analysis, 3D-CNNs are highly effective due to their ability to process volumetric data. 2D-CNNs are used when analysing 2D slices independently.
- *Surveillance*: For detecting events or anomalies, 3D-CNNs excel in analysing continuous motion, while 2D-CNNs can focus on individual frames for simpler anomaly detection tasks.
- *Sports analytics*: Dynamic sports actions, such as running or throwing, benefit from 3D-CNNs' spatiotemporal modelling, while static poses or frame-based activities are better handled by 2D-CNNs.

Each of these application areas demonstrates the complementary strengths of 2D-CNNs and 3D-CNNs, allowing practitioners to select the appropriate model based on the task requirements.

3.6.3. Performance comparison

The performance of 2D-CNNs and 3D-CNNs depends on multiple factors, including accuracy, computational efficiency, and suitability for real-time applications. A detailed analysis is follows:

- *Accuracy*: 3D-CNNs consistently outperform 2D-CNNs on tasks requiring spatiotemporal understanding, such as VAR, due to their ability to model motion directly. However, for static image tasks, both models perform equally well.
- *Speed*: 2D-CNNs are computationally efficient and faster to train due to their simpler architecture. This makes them suitable for real-time applications, particularly on hardware-constrained devices.

- *Scalability*: 2D-CNNs scale effectively with large datasets, requiring minimal computational resources. Conversely, 3D-CNNs demand high-end hardware (*e.g.*, GPUs with substantial memory) to handle the increased complexity of spatiotemporal data.
- *Suitability for real-time use*: Due to their lower resource requirements, 2D-CNNs are more practical for real-time use. In contrast, the computational overhead of 3D-CNNs makes real-time deployment challenging without significant optimisation.

3.6.4. Insights and recommendations

Based on the comparative analysis, the following insights can be drawn:

- *Advantages of 2D-CNNs*: Their simpler architecture and lower computational demand make them ideal for tasks involving static images or when computational efficiency is critical.
- *Advantages of 3D-CNNs*: The ability to directly model spatiotemporal features gives 3D-CNNs an edge in video analysis tasks, such as VAR and dynamic event detection.
- *Challenges of 2D-CNNs*: While effective for spatial feature extraction, they require additional mechanisms to capture temporal dependencies, which can complicate the model pipeline.
- *Challenges of 3D-CNNs*: The high computational cost and memory requirements of 3D-CNNs make them less suitable for real-time or resource-constrained applications.

To address these challenges, hybrid approaches combining 2D-CNNs with temporal models (*e.g.*, RNNs, Transformers) or using lightweight 3D-CNN architectures can balance accuracy and efficiency, offering a practical solution for many applications.

3.7. Summary

3D-CNNs are effective in various VAR tasks, including AR and event detection. They are good at capturing spatiotemporal dependencies and motion patterns, enabling accurate classification of actions and activities in videos. 3D-CNNs have shown promise in capturing both spatial and temporal dependencies in video data, enabling effective VAR. However, several limitations and open challenges persist. By addressing these limitations, researchers can unlock the full potential of 3D-CNNs and create more accurate, robust, and interpretable VAR systems.

4. Models based on RNN and LSTM

In this section, we will discuss VAR models based on RNN and LSTM.

4.1. Briefing RNN and LSTM

Among DL models, RNN and its variant LSTM have shown promising results in handling sequential data [105,106], making them particularly well-suited for VAR tasks where temporal dynamics play a critical role [107].

RNNs are *recurrent* because they have internal loops, can retain a state from one step to the next in their hidden layers. This property makes them ideal for time-series prediction, Natural Language Processing (NLP), and other sequential data types. While RNNs are good at modelling short-term dependencies, they need help in the case of long-term ones. Moreover, RNNs can suffer from vanishing and exploding gradient problems. When the sequences are long, the gradients computed during the update phase can either become very small (vanish) or very large (explode).

In 1997, Hochreiter and Schmidhuber [24] proposed LSTM networks, a special kind of RNN. LSTM was explicitly designed to avoid the long-term dependency problem (the *vanishing gradient* problem mentioned earlier) by maintaining a more constant error through *gates* (components of LSTM that manage the cell state).

The cell state functions like a conveyor belt, passing information across the data sequence without altering it. The gates determine which information is permitted onto this conveyor belt. With this arrangement, the LSTM can either retain or disregard the information in the cell state throughout lengthy sequences, making it ideal for tasks that involve learning from long data sequences, such as video.

LSTM networks have three types of gates [108]:

1. *Forget gate*: It decides which information should be removed from the cell state.
2. *Input gate*: It adds fresh information to the cell state.
3. *Output gate*: It determines the subsequent hidden state.

LSTM networks can modify the information in the cell state, which gives them their long-term memory abilities. This feature is useful in overcoming the vanishing gradient issue, making them ideal for tasks that involve sequences.

In addition to LSTM, there are other improved versions of RNNs, such as Bidirectional RNNs (BiRNNs) [109], and Gated Recurrent Unit (GRU) [110,111]. Some studies applied them to VAR, and some new effective models have emerged and achieved good results, such as the models in [112–115].

4.2. Reasons why RNN & LSTM useful for VAR

RNN and LSTM networks are useful for VAR for several reasons:

1. *Temporal modelling*: RNN and LSTM are designed to handle sequential data and can effectively model temporal dependencies in video sequences. They can capture the order and relationships between frames over time, allowing them to understand video dynamics and motion patterns.
2. *Long-term dependencies*: Compared to CNN, which can only handle short videos, RNN and LSTM have varying degrees of improvement in capturing long-term dependencies in video sequences. This is particularly important in AR tasks, where actions may span multiple frames and require a contextual understanding of the entire sequence.
3. *Memory and context*: LSTM networks have an explicit memory cell that enables them to store and access information over longer time spans. This memory cell helps retain important contextual information throughout the video sequence, allowing for a better understanding of actions and their context.
4. *Variable-length input*: RNN and LSTM can handle video sequences of variable lengths. They process videos with different numbers of frames without the need for fixed-size inputs, making them flexible for real-world video data (often with varying durations).

In short, RNN and LSTM are well-suited for VAR because they can model temporal dependencies, capture long-term context, and handle variable-length inputs.

4.3. General architecture and components

Fig. 6 shows the general architecture of models based on LSTM for VAR. The general architecture of RNN/LSTM-based models for VAR can be summarised as follows:

1. *Preprocessing*: In this stage, the video is broken down into individual frames, effectively treating the video as a sequence of images. This is important because RNNs and LSTM are designed to handle sequential data.

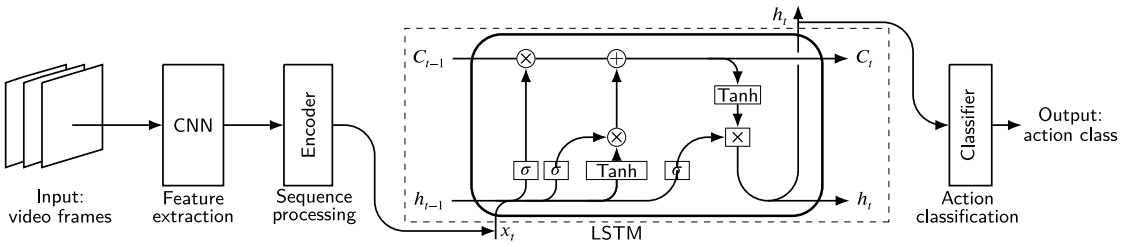

Fig. 6. General architecture of LSTM-based model.

2. *Feature extraction*: Before a video can be fed into an RNN, it first needs to be transformed into a suitable format. For each frame, a feature extraction process occurs, typically using a CNN (*e.g.*, [115,116]). The CNN transforms each frame into a dense vector that captures the relevant features (*e.g.*, edges, shapes, textures, and objects) in the image. To focus on the effective features, an AM is usually added to CNN, such as the work of [117]. It also uses classical algorithms to extract features, *e.g.*, Dharejo et al. [118] used 3D-DWT (Discrete Wavelet Transform) to extract relevant features.
3. *Sequence processing*: The sequence of feature vectors (one for each frame) is then fed into an RNN or LSTM, thereby capturing the temporal dynamics of the video.
4. *RNN/LSTM processing*: The RNN/LSTM reads the input sequence, and for each element, it computes the corresponding hidden state. This hidden state is then combined with the next input to produce a new state. This allows the model to implicitly "understand" the temporal dependencies between different frames.
5. *Action classification*: The output of the RNN or LSTM (usually the hidden state from the last time step) is then used as the basis for a final classification layer. This layer outputs the probabilities for each possible action category, and the one with the highest probability is chosen as the predicted action.

In practice, each of these steps would involve a lot more details, such as data cleaning, data augmentation, handling of imbalanced classes, and hyperparameter tuning.

4.4. Typical RNN/LSTM-based models

In general, RNN/LSTM-based models differ in their strategies for feature extraction, RNN/LSTM processing, and the use of multiple branch networks, resulting in different effects. Several typical models and methods have emerged, including RNN/LSTM+CNN, enhanced RNN/LSTM, and RNN/LSTM combined with the AM.

4.4.1. Models based on RNN/LSTM with CNN

As mentioned above, a video must be converted into an appropriate format (feature vector) before being fed into an RNN/LSTM. Some methods use a CNN for this transformation, resulting in a model based on an RNN/LSTM with CNN. For example, in 2018, Yuan, Zhao, and Wang [114] proposed a lightweight DL model based on the C3D network and the RNN for complicated VAR. They used a two-stream C3D network (a kind of CNNs) to extract the spatiotemporal features of single acts in spatial-optical data and RGB data, respectively. An RNN model based on two stacked LSTM layers is presented for mining patterns and semantics among the spatiotemporal features of single acts. The method performs well for complex VAR.

In 2019, Ma et al. [119] proposed a Temporal Segment LSTM (TS-LSTM) and a Temporal-Inception CNN for VAR. They used a two-stream ResNet-101 (pre-trained on ImageNet) to extract spatial and temporal features concatenated into feature matrices. These matrices were input to both TS-LSTM (which processes temporally segmented, pooled features via LSTM layers) and Temporal-Inception. Using RNNs/LSTMs and Temporal CNNs on spatiotemporal feature matrices leverages spatiotemporal dynamics for improved performance.

In 2019, Wang et al. [120] considered that 3D-CNN is better for low-level spatiotemporal feature extraction while RNN is better for modelling high-level temporal feature sequences. So, they proposed a model named I3D-LSTM. First, they adopted the Kinetics-pretrained I3D model to learn low-level features between adjacent frames and then used LSTM to model the high-level temporal features produced by the Kinetics-pretrained 3D-CNN model. The I3D-LSTM model achieves leading performance on dataset UCF101. However, the performance of the I3D-LSTM model needs to be improved on other benchmark datasets.

In 2021, Hu et al. [121] proposed an improved SpatioTemporal Differential LSTM (ST-D LSTM) network. An enhanced input differential feature module and a spatial memory state differential module were added to the network. In this architecture, the model first converts the video to frame images, then uses the pretrained CNN (an Inception-V3 network) to extract the spatial features of the frame images; next, it inputs the extracted features into the ST-D LSTM network to extract the temporal and spatial information. Finally, the output result is classified by Softmax. ST-D LSTM can effectively improve the accuracy of classical Long-term Recurrent Convolutional Networks (LRCN).

In 2022, Wang et al. [122] proposed a model, ResLNet (Deep Residual LSTM Network), to have convolutions collaborated with LSTM more effectively under the residual structure to learn better spatiotemporal representations. The superiority of ResLNet and its ablation study are shown on the three most popular benchmark datasets: HMDB51, UCF101, and Kinetics. ResLNet could be adopted for various features, such as RGB and optical flow.

In an attempt to obtain improved results in VAR, in 2023, Abdelrazik, Zekry, and Mohamed [123] presented an algorithm that combines CNN and RNN. In the first part, there is a preprocessing stage to make the video frame suitable for the input of both CNN networks, which consists of a fusion of Inception-ResNet-V2 and GoogleNet to obtain activations, with the previously trained wights in Inception-ResNet-V2 and GoogleNet and then passed to a deep GRU connected to a fully connected SoftMax layer to recognise and distinguish the human action in the video. Their model gives better accuracies of 97.97% on UCF101 and 73.12% on HMDB51 than those in the related literature.

In 2024, Cob-Parro et al. [124] proposed a real-time people detection and AR approach using edge computing. Their model has two main modules: one detecting and tracking individuals with a CNN (MobileNetV2 [48] with SSD [125]) and Kalman filters, and the other recognising actions through an LSTM network fed with lightweight feature vectors from bounding boxes. On five datasets: KTH, WEIZMANN [126], WVU [127], IXMAS [128], and their newly created GBA dataset, the model achieves over 99% accuracy in most cases while minimising computational overhead, making it suitable for real-time applications.

These models share some common considerations. Firstly, they all use a CNN to extract low-level features. Then, they employ RNNs, particularly LSTMs, to learn high-level features from the extracted low-level features. The variations between these methods are mainly due to the different types of CNNs used.

This type of models has two main limitations. Firstly, there are differences in feature extraction efficiency and effectiveness between various CNNs. More complex CNNs perform better, but have lower efficiency (*e.g.*, C3D, I3D, and T3D). However, using a simple CNN does not produce satisfactory results even though it improves efficiency. One way to improve this is by integrating AMs in the feature extraction process. Secondly, RNN/LSTM have inherent defects, but these can be improved by using bidirectional RNN/LSTM, Attention LSTM, and other DL models.

4.4.2. Models based on BiRNN/BiLSTM

BiRNNs and BiLSTM (Bidirectional LSTM) models extend the capabilities of their unidirectional counterparts by processing the input sequence in both forward and backward directions. By considering both past and future frames, these models can capture a more proper understanding of the temporal dynamics present in videos. This bidirectional processing is achieved by duplicating the hidden layers and processing the input sequence in both directions, then combining the outputs from both directions to make predictions.

In 2017, Ullah et al. [112] presented a VAR model using a CNN and a Deep Bidirectional LSTM (DB-LSTM). Deep features were extracted from every sixth frame to reduce redundancy, and the DB-LSTM learned sequential information. Multiple DB-LSTM layers (forward and backwards) increased depth, enabling the model to effectively learn long-term sequences and analyse features over time, processing lengthy videos.

In 2018, Li, Nie, and Su [129] used part selection within clips and considered the bidirectional temporal information when modelling the temporal pattern using multiple layers of an LSTM framework, which can learn compositional representations in space and time. On datasets UCF101 and HMDB51, the proposed architecture achieves SOTA results.

In 2019, Hanson et al. [130] proposed a spatiotemporal encoder for violence detection (a subset of VAR) in videos. This encoder is comprised of three parts: a VGG13 network spatial encoder, a BiLSTM, and a classifier. The BiLSTM is used as the temporal encoder, the input to which are the feature maps from the spatial encoder. The performance on relevant datasets is quite impressive.

In 2021, He et al. [131] proposed a DL model to capture the spatial and temporal patterns of human actions from videos. They use sample representation learner to extract the video-level temporal feature, and use a Densely-Connected BiLSTM (DC-BiLSTM) network to model the visual and temporal associations in both forward and backward directions. On datasets UCF101 and HMDB51, the DC-BiLSTM model achieves SOTA for VAR.

In 2022, Tan et al. [132] used a BiLSTM with Temporal Dense Sampling (TDS) and a fusion network for VAR. TDS segments the video and applies max pooling. A multi-stream BiLSTM encodes long-term spatiotemporal dependencies in both directions. A fusion network with a trained and fully-connected layer adaptively weights the spatial and temporal network outputs. Their model achieves 94.78% on UCF101 and 70.72% on HMDB51, outperforming previous models.

Despite VAR progress, accuracy and computational complexity remain challenges. Thus, in 2024, Hassan, Miah, and Shin [133] proposed a VAR model combining a DB-LSTM with transfer learning. They used MobileNetV2 for feature extraction and DB-LSTM for dependency identification. Iterative fine-tuning ensures adaptability. Their model achieves 99.20%, 93.30%, and 76.30% on UCF11 [134], UCF Sport [135], and JHMDB [136], respectively.

These BiRNN/BiLSTM based models have some commonalities, such as they all used BiRNN/BiLSTM as a key component to process temporal information. Meanwhile, their differences are also evident. Ullah et al. [112] processed video data using CNN and DB-LSTM network. Li, Nie, and Su [129] used a BiLSTM network for VAR. Hanson et al. [130] used a BiLSTM network for detecting violence in videos. He et al. [131] used sample representation learner to extract the video-level temporal feature. Tan et al. [132] used the TDS partitions video into segments and fusion network where a fully-connected layer is trained to adaptively to assign the weights. Hassan, Miah, and Shin [133] used a DB-LSTM to identify dependencies of features.

The limitations of these BiRNN/BiLSTM models mentioned above are as follows:

1. *Computational complexity*: BiRNN/BiLSTM involve significant computational resources because they need to process the video sequence in both forward and backward directions. This can be challenging when dealing with high-resolution or long-duration videos, where the number of frames can be quite large.
2. *Latency in real-time applications*: In the bidirectional model, the future context is required to generate predictions. This means the entire video sequence must be available before processing can start, which can introduce latency and make BiRNNs unsuitable for real-time VAR.
3. *Vanishing gradient problem*: Although LSTM networks are designed to mitigate the vanishing gradient problem, they can still suffer from it when processing very long sequences. This can make it difficult to learn long-range dependencies in the data.

A few general strategies that can be employed to address the limitations of these models are as follows:

1. *Transfer learning*: Transfer learning can be used to transfer knowledge from pre-trained models to new models and improve the performance of the VAR system, such as the model in [133].
2. *AMs*: AMs can be used to focus on important spatiotemporal features and improve the performance of the VAR system.
3. *Use of convolutional layers*: Convolutional layers can be used before the RNN/LSTM layers to reduce the dimensionality of the input. This can make the model more computationally efficient and help it extract useful spatial features from the video frames.
4. *Hybrid models*: Combining RNN/LSTM with other types of models can improve performance. For example, a 3D-CNN could be used to extract spatiotemporal features from the video, which are then processed by the RNN/LSTM.

The choice of model and improvement strategies depend on the specific requirements of the VAR task, such as the nature of the actions to be recognised, the available computational resources, and the need for real-time processing.

4.4.3. Models based on attention LSTM

Attention LSTM is an advanced LSTM-based method where the concept of AM is incorporated into the LSTM. It helps the model to focus on the most relevant parts of the input sequence at each time step, making it highly efficient in handling complex actions. In the context of VAR, Attention LSTM can automatically locate the most important regions in a video sequence, which contribute more to the VAR.

Attention LSTM can greatly aid in VAR due to its inherent capabilities in learning long-term dependencies and its ability to give more importance to relevant information. The reason is as follows:

1. *Handling temporal dependencies*: VAR requires understanding temporal dependencies between different frames of a video. LSTM networks are explicitly designed to deal with sequences and their long-term dependencies, making them suitable for this task.
2. *Attention to important frames*: However, not all frames in a video sequence are equally informative for recognising an action. The AM within an LSTM can be used to assign different weights to different frames based on their relevance, enabling the model to focus more on the key frames that are more indicative of the action.
3. *Reducing noise*: By focusing more on relevant frames, the AM also helps reduce the impact of noise or irrelevant frames in the video sequence, leading to more accurate VAR.

VAR can significantly benefit from Attention LSTM for the following reasons:

1. *Complex sequences*: The action depicted in a video is often composed of a complex sequence of movements spread over time. Attention LSTM can capture this complexity better than simpler models.
2. *Variability of actions*: The same action can be performed very differently by different people in different environments. Attention LSTM can learn to focus on the key elements of the action that remain consistent across these variations.
3. *Large volume of data*: Video data consists of a large number of frames, which can make the VAR task computationally expensive. Attention LSTM can alleviate this by focusing on the most informative frames, reducing the computational cost.
4. *Real-time applications*: In real-time VAR, it is important to make quick decisions based on the most recent frames of the video. Attention LSTM is well-suited because the AM can dynamically adjust to focus on the most relevant current information.

Attention LSTM is a robust and efficient approach for VAR, capable of handling complex sequences, reducing noise, and focusing on the most informative parts of the video.

Some recent typical studies are as follows. In 2020, Zhang et al. [137] presented the Spatial-Temporal Dual-Attention Network (STDAN) for VAR. STDAN uses convolutional and fully-connected LSTMs with distinct AMs, extracting features from both CNN layers. Two attention modules enhance spatiotemporal attention. Principal component analysis and feature fusion further boost its performance, surpassing previous models.

Some VAR models often treat visual and temporal cues equally, hindering feature distinction. To address this issue, Dai, Liu, and Lai [117] proposed an end-to-end two-stream attention-based LSTM network in 2020. This network selectively focuses on relevant image features, assigning varying attention levels to each feature map. A deep feature correlation layer adjusts network parameters based on correlations between the two feature streams. Their model achieves SOTA performance in everyday scenarios.

In 2021, Muhammada et al. [138] proposed a BiLSTM-based attention module with a dilated CNN for VAR. A CNN extracts high-level features, which are enhanced via skip connections. These features are then input to a BiLSTM network for temporal learning. An attention layer refines spatiotemporal information, improving performance. Their model showed improvements over previous models.

In 2022, Bayoudh, Hamdaoui, and Mtibaa [140] developed a hybrid 2D/3D-CNN, LSTM, and attention module model for VAR. CNN features are input to an LSTM to capture short- and long-term dependencies. An attention module focuses on salient visual features. Trained and evaluated on KTH, the model achieves promising results compared to previous models.

In 2024, Kumari and Anand [141] proposed a hybrid CNN-attention-based LSTM model for VAR. They used lightweight MobileNetV2 to extract spatial features, which are then fed to an attention-enhanced LSTM to focus on key gesture cues. Their model achieves 84.65% accuracy on dataset WLASL [142].

The five papers are similar in using LSTM networks combining AM for VAR. However, they are different.

1. Zhang et al. [137] used convolutional LSTM and fully-connected LSTM with different attentions for VAR.
2. Dai, Liu, and Lai [117] proposed a two-stream attention based LSTM network.
3. Muhammada et al. [138] used attention-based BiLSTM networks with dilated CNN features.
4. Bayoudh, Hamdaoui, and Mtibaa [140] used a hybrid 2D/3D-CNN to extract features before fed them into an LSTM network, and used AM to focus on relevant salient features.
5. Kumari and Anand [141] selected MobileNetV2 as the backbone to extract meaningful features, which were then fed to an AM-based LSTM.

Some limitations of these papers are as follows:

1. Zhang et al. [137] suffer the less combinational use of convolutional LSTM and fully-connected LSTM leads an inadequate useation towards the deep feature extracted from different-level layers. More seriously, the weak attention capability in CNN-LSTM based models may result in noise interference and even performance degradation.
2. Due to the calculation of optical flow, end-to-end training of Dai, Liu, and Lai [117] cannot be achieved. In addition, two-stream processing can also increase computational overhead.
3. Muhammad et al. [138] proposed a VAR model which used a single-stream learning strategy for AR. Still, its ability to learn discriminative features from video frames is limited, and its application ability on large-scale datasets needs to be strengthened.
4. The model efficiency of Bayoudh, Hamdaoui, and Mtibaa [140] needs to be improved, and its accuracy on other datasets also needs to be improved.
5. The model by Kumari and Anand [141] was only tested on a subset of the WLASL dataset with 100 classes, which limits its generalisability to a broader range of sign language variations.

Some ways to overcome the limitations of these papers are as follows:

1. The work of Zhang et al. [137]: Temporal and spatiotemporal attention modules combined with PCA (Principal Component Analysis) can encode key information and suppress irrelevant feature dimensions.
2. The work of Dai, Liu, and Lai [117]: Consider replacing the optical flow scheme and learning the features of other modes.
3. The work of Muhammad et al. [138]: Using a two-stream learning strategy can extend their work to more intelligently learn discriminative features from video frames to recognise complex actions in large-scale datasets.
4. The work of Bayoudh, Hamdaoui, and Mtibaa [140]: Need to improve the discriminatory performance of the proposed architecture in terms of accuracy and efficiency by better tuning the hyperparameters and enhancing the architectural design of the model.
5. The work of Kumari and Anand [141]: Testing the model on larger datasets with more classes, such as the full WLASL dataset, could help improve its robustness and generalisability.

4.5. Comparison, limitation, and future work

Table 6 shows the differences between the above RNNs and LSTM-based models in input, low-level features extraction, high-level features extraction, and classifier.

While RNNs and LSTM networks have achieved considerable success in the field of VAR, they still have some common limitations:

1. *Handling long sequences*: While LSTM networks are designed to handle the vanishing and exploding gradient problem in standard RNNs, they can still struggle with very long sequences. This is particularly relevant in VAR where videos can last for several minutes or even hours.
2. *Real-time processing*: The sequential nature of RNNs and LSTM makes real-time processing a challenge. This is because each timestep in the sequence depends on the previous ones, which can slow down computation, particularly for long sequences.

Table 6
Comparison of the models based on RNNs & LSTM.

Method	Input	Low-level features extraction	High-level features extraction	Classifier
Method in [112]	Video frames	A pretrained AlexNet model for feature extraction on large scale ImageNet dataset	DB-LSTM	Softmax
Method in [114]	Three input data including RGB videos, TV-L1 optical flow [139] videos, and spatial-optical videos	A two-stream C3D network to extract the spatiotemporal features	A RNN model based on two stacked LSTM layers uesd for mining patterns and semantics among the spatiotemporal features	A softmax layer
Method in [117]	RGB frames and optical flow	Convolution layer	Two-stream attention based LSTM	Use score fusion in the final classification
TS-LSTM+ temporal-inception [119]	RGB frames and optical flow	A Two-stream CNN using ResNet-101 pre-trained on ImageNet	Temporal segment LSTM or Inception-style Temporal-CNN	No details in the original paper
I3D-LSTM [120]	RGB frames and optical flow	A Kinetics-pretrained I3D model	LSTM network	A softmax classifier
ST-D LSTM [121]	Video frames	Pretrained CNN (a Inception-V3 network)	ST-D LSTM	Classified by softmax
ResLNet [122]	Video frames	Residual LSTM block	ConvLSTM with batch normalisation	Softmax
Method in [123]	Video frames	A fusion of Inception-ResNet-V2 and GoogleNet	A deep GRU	A fully connected softmax layer
Method in [124]	Video/images	MobileNetV2+SSD	LSTM is fed with a lightweight feature vector	A softmax layer
Method in [129]	Videos frames	C3D+Part Selection	Multiple layers of LSTM framework (three LSTM structures)	Softmax layer
Method in [130]	Video frames	A VGG13/AlexNet network spatial encoder	A BiLSTM used as the temporal encoder	No details in the original paper
DC-BiLSTM [131]	RGB frames and optical flow	Sampling stack and representation learner (Densely Connected CNNs)	A DC-BiLSTM network	Fusion layer
TDS-BiLSTM [132]	Video frames	A TDS strategy	A multi-stream BiLSTM network	Fusion network
Method in [133]	Videos	MobileNetV2	Deep BiLSTM	Softmax
STDAN [137]	Video frames	A CNN	FC-LSTM and Conv-LSTM+attention modules	Fusion module and softmax layer
Method in [138]	Video frames	A DCNN	Attention and BiLSTM	Combine the centre loss and the Softmax loss
Method in [140]	Video frames	Hybrid 2D/3D-CNN	LSTM+Attention	Softmax
Method in [141]	Frames sequence	MobileNetV2	LSTM+Attention	Softmax

3. *Model complexity and training difficulty*: Training LSTM can be computationally intensive, especially with high-dimensional inputs like videos. It can be difficult to train these networks optimally, leading to potential overfitting or underfitting problems.

The field of VAR is still evolving, with plenty of opportunities for new and improved methods. Researchers continue to explore innovative ways to leverage RNNs and LSTM for this task, addressing the existing challenges and opening up new possibilities.

To address these limitations, future research could explore several promising avenues:

- *Improved sequence modelling*: Transformers have shown significant success in handling long sequences, particularly in NLP. Adapting these architectures to VAR could help mitigate the struggles with very long video sequences. Combining RNNs/LSTMs with AMs or 3D-CNNs may help better capture long-range dependencies in videos while simplifying training.
- *Real-time processing enhancements*: (1) Developing more efficient models like *sparse LSTMs* or *lightweight RNNs* that can process sequences faster without compromising accuracy will be crucial for real-time VAR. (2) Leveraging parallel processing techniques and hardware accelerators (*e.g.*, GPUs, TPUs) can help reduce computational time, particularly for long videos.
- *Simplification of models*: (1) Research into methods like model pruning, quantisation, and distillation could help simplify complex LSTM-based models, making them more efficient and easier to train. (2) Instead of focusing solely on deep models, the research could also look into shallow architectures or hybrid models that balance performance with computational efficiency.
- *Transfer learning*: Transfer learning, where models pre-trained on large datasets are fine-tuned for specific VAR tasks, could reduce the computational cost associated with training LSTMs from scratch.
- *Self-supervised and semi-supervised learning*: Given the large amounts of unlabelled video data available, self-supervised and semi-supervised learning methods could be explored to reduce the need for vast labelled datasets, which would also help address overfitting and under-fitting issues during model training.

These future directions could pave the way for the development of more robust, efficient, and scalable models for VAR using DL.

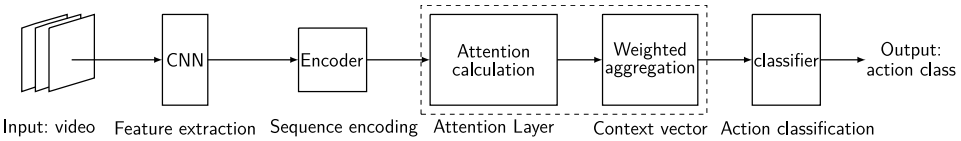

Fig. 7. General workflow of AM-based models for VAR.

4.6. Summary

RNN and LSTM provide a powerful way to process sequence data, making them an important tool for tasks such as VAR. However, they can be computationally intensive and require careful tuning to work well in practice. Hence, the emergence of different variants and optimisation techniques to improve their performance and stability.

These novel methods continue to push the boundaries of what is possible in VAR, by addressing challenges such as varying video lengths, multi-scale actions, and the need to focus on the most relevant parts of each video.

5. Models based on AMs

This section will review VAR models based on AMs.

5.1. Basic idea

VAR models based on AMs aim to selectively focus on certain regions or frames of a video that are important for recognising an action, while ignoring irrelevant regions or frames. Using AMs in DL models for VAR can help to improve the accuracy and robustness [143–145].

AMs are a powerful tool in DL that allow models to weight the importance of input data dynamically. They have shown to significantly improve the performance of various models, such as CNNs, specially the two-stream networks (*e.g.*, [55,143]) and 3D-CNN (*e.g.*, [146,147]), and RNN & LSTM (*e.g.*, [148–152]).

5.2. Reasons why AMs useful for VAR

AMs-based models are useful for VAR because they enable the models to focus on the most relevant parts or frames of a video sequence. Videos typically contain a large amount of visual information, and not all frames contribute equally to understanding the action being performed. AMs allow the model to selectively attend to the most informative frames or regions within each frame, improving the model's ability to recognise actions accurately.

AMs-based models have proven to be very effective for VAR due to several reasons:

1. *Handling temporal complexity*: Videos are essentially a sequence of images (frames) that contain temporal information. This sequence information is very important for understanding actions that are occurring in the video. Just like the role of attention in sequence-to-sequence models for NLP, AMs in VAR help the model to focus on the important frames at each point in time, thereby better capturing the temporal dynamics of the video.
2. *Selective focus*: Not all frames in a video are equally important for understanding the action. For example, in a video of someone shooting a basketball, the frames where the person is in the act of shooting are more important than the frames where the person is just dribbling the ball or standing still. AMs help the model to focus on the frames that are most relevant for understanding the action.
3. *Handling redundancy*: Videos often contain a lot of redundant information. For example, in a video of a person running, some frames look very similar. AMs can help the model to ignore this redundant information and focus on the unique frames that provide new information about the action.
4. *Contextual understanding*: Some actions can be best recognised when considering the surrounding context. AMs allow models to weigh the importance of local features in the context of the entire sequence, improving the performance on recognising complex actions that require understanding the broader context.
5. *Model interpretability*: Attention maps can provide a visualisation of what parts of the video the model is focusing on, which can help with understanding and interpreting the model's predictions.

These reasons illustrate why AMs can be highly beneficial for VAR. However, AMs might not always lead to the best performance, depending on the specific task and dataset. Other factors like the model architecture, the training procedure, and the quality and size of the training data can also have a big impact on the performance of the model.

5.3. General way of working

Fig. 7 shows the general workflow of AM-based models for VAR. AMs in VAR generally follow a similar principle as those in NLP [153,154] tasks. A general overview of how they might work is as follows:

1. *Feature extraction*: First, features are extracted from each frame in the video. This can be done using a CNN that has been pre-trained on an image classification task. These features serve as the input to the AM.
2. *Sequence encoding*: The sequence of features is then passed through some form of sequence model, such as a RNN or a Transformer model. This helps to encode the temporal information in the video.
3. *Attention calculation*: The AM then calculates an attention score for each frame in the video. This score indicates how important each frame is for understanding the action that is taking place. The attention scores are typically calculated using a softmax function to add up to 1 and can be interpreted as probabilities.
4. *Context vector*: The attention scores are then used to calculate a weighted sum of the features of all the frames, which results in a single context vector. This context vector should represent the most important information in the video for understanding the action.
5. *Action classification*: Finally, the context vector is passed through a fully connected layer to classify the action. The output layer typically uses a softmax function to produce a probability distribution over the possible actions.

This is a high-level overview and the specifics can vary depending on the exact model architecture. For instance, the AM could be applied at different layers of a model, or multiple attention heads might be used to focus on different parts of the input. Also, AM could be applied in both spatial and temporal domains in the video, helping the model to focus on the most relevant regions and frames for VAR.

AMs have been applied in various ways in the domain of VAR. Based on how researchers apply the AMs and the specific structure of video data, what we discuss include spatiotemporal attention, self-attention, hierarchical attention, channel attention, multi-head attention, and cross-modal attention.

5.4. Models based on spatiotemporal attention

Spatiotemporal AM based models have been increasingly leveraged for VAR tasks due to their ability to capture and attend to the most critical spatial and temporal features across a video sequence.

A spatiotemporal AM based model for VAR generally works as follows:

1. *Spatial attention*: This aspect of the model focuses on identifying which regions in individual frames are important for the AR being performed. It computes a spatial attention map for each frame to weigh the significance of different areas, effectively highlighting regions that contain the most relevant information for the action. These models use attention mechanisms to select informative regions within each frame. The aim is to concentrate on the parts of the frame most pertinent to the action being executed, thereby enhancing recognition accuracy. Examples of spatial attention methods include non-local CNNs, soft attention models, and spatial pyramid pooling.
2. *Temporal attention*: This component learns which frames in the video sequence are the most relevant for AR. Not all frames contribute equally to the understanding of the action; some frames are more informative than others. Temporal attention provides a mechanism to weigh the importance of each frame in the sequence. These models use AMs to focus on certain frames or sequences of frames over others. This is done to emphasise the moments in the video that are crucial for AR, while de-emphasising less relevant moments. Temporal attention can be achieved using techniques like RNN, LSTM, or self-AM. Some methods also employ optical flow or motion-based features to better capture temporal information.
3. The combination of spatial and temporal attention allows the model to focus on the most important parts of the video, both within individual frames and across the sequence of frames, thereby increasing the efficiency and effectiveness of VAR.

These models can be integrated into various architectures, including 3D-CNNs, RNNs, and more recently, Transformer-based models, which inherently include AMs.

In 2018, Li et al. [155] proposed a unified Spatio-Temporal Attention Network (STAN) for multi-modal VAR. Input videos are represented by multiple modalities and processed by CNNs. Spatial attention neural cells pool local descriptors into video segment representations. These representations are concatenated and input to an LSTM with temporal attention cells, providing global guidance across modalities. The resulting video representation is used for VAR. STAN achieves superior results on UCF101, CCV [156], THUMOS14 [157], and Sports-1M.

In 2019, Meng et al. [151] proposed an interpretable and easy plug-in spatiotemporal AM for VAR. For spatial attention, they learned a saliency mask to allow the model to focus on the most salient parts of the feature maps. For temporal attention, they employed a convolution LSTM based AM to identify the most relevant frames from an input video. This model shows superior or competitive accuracy with the previous models while increasing model interpretability.

In 2020, Yang et al. [158] proposed a Spatial-Temporal Attentive CNN (STA-CNN). STA-CNN incorporates temporal and spatial attention modules into a unified CNN for VAR. The temporal module mines discriminative temporal segments, while the spatial module uses optical flow to locate motion salient regions and, with an auxiliary loss, focuses on discriminative non-motion areas. STA-CNN achieves SOTA performance on UCF101 and HMDB51.

TSN enables the network to capture long-term information in the video. However, this approach can be affected by unrelated frames or areas, leading to inaccurate AR. To address this issue, in 2022, Yang et al. [159] proposed a Spatial-Temporal Attention TSN (STA-TSN). This approach enables the network to focus on key features in space and time, resulting in improved performance over TSN. STA-TSN outperforms TSN on datasets UCF101, HMDB51, JHMDB, and THUMOS14 and achieves SOTA results.

All the four papers above used some form of spatiotemporal AM to improve VAR performance. Some differences of them are as follows. Li et al. [155] proposed a unified spatiotemporal attention network for VAR. Meng et al. [151] focused on interpretability of the spatiotemporal AM. Yang et al. [158] proposed a convolutional spatiotemporal attention learning method for AR. Yang et al. [159] focused on spatiotemporal and TSN.

These methods have been validated through experiments from different perspectives that the spatiotemporal models can extract more critical and useful features. However, there are inherent limitations to these models.

1. *Computational complexity*: Spatiotemporal AM based models require the computation of attention maps for each frame in the sequence and across all spatial locations, which can be computationally intensive, especially for high-resolution videos or long video sequences. One way to mitigate this would be to employ efficient AMs, use sparse attention, or apply dimensionality reduction techniques. Moreover, computational costs could also be offset by advances in hardware acceleration.
2. *Long-term dependencies*: While these models can effectively capture short-term temporal dynamics, capturing long-term temporal dependencies across frames can be challenging. The integration of RNN components like LSTM or GRU units, or advanced Transformer models like the Longformer, could help to capture longer-term dependencies.
3. *Sensitivity to noise*: spatiotemporal attention models might be sensitive to noisy frames or regions within frames, which can lead to a deterioration in performance. Robust preprocessing techniques such as denoising or outlier detection could be used. Furthermore, training the model on a variety of data, including noisy examples, could increase its robustness.

While spatiotemporal AM based models have shown promise in VAR tasks, they do present challenges such as computational complexity, handling long-term dependencies, and sensitivity to noise. Addressing these limitations through various strategies can contribute to the advancement of spatiotemporal AM based models for VAR.

5.5. Models based on self-attention

Self-AM based models for VAR have emerged as a powerful tool for efficiently and effectively understanding the temporal dynamics and spatial relationships within video frames. This application of self-AM, mainly seen through the integration of Transformers within AR architectures, brings the ability to capture long-term dependencies and complex interactions between different parts of the input.

Self-AM was first proposed in the Transformer model by Vaswani et al. [160] in 2017 for the task of neural machine translation. The key idea behind self-attention, or intra-attention, is to compute a representation of each element in a sequence by considering its relationship with all other elements in the sequence.

In the context of VAR, each frame or group of frames can be treated as an element in a sequence, and the self-AM can be applied to compute a representation of each frame that is informed by its relationship with all other frames in the video. This way, the model can learn to pay attention to the most important parts of the video that are relevant for AR.

In 2019, Purwanto et al. [161] presented a three-stream network for AR in extremely low-resolution videos. The network is combined with the I3D model pre-trained on dataset Kinetics to produce more discriminative spatiotemporal features in blurred low-resolution videos. A bidirectional self-AM is also aggregated with the network to further manifest various temporal dependence among the spatiotemporal

features. Moreover, they devised a fusion strategy to integrate the information of the three modalities. Their model outperforms the main SOTA extreme low-resolution AR methods on datasets HMDB51 and IXMAS.

Traditional 3D-CNNs struggle with fast movements, long-term dependencies, and overfitting. To overcome these limitations, in 2020, Anvarov, Kim, and Song [162] integrated squeeze-and-excitation (SE) modules for enhanced feature selection and self-AM for better long-range dependencies. Their model achieves SOTA results with 95.6% (on UCF-101) and 74.1% (on HMDB51) accuracy. Their model improves accuracy while keeping computational costs low, making it a promising advancement for VAR.

Some VAR models struggle with dark videos. To address this, in 2021, Chen et al. [163] proposed DarkLight Network, comprising: (1) a dual-pathway structure using both dark videos and its brightened counterpart; and (2) self-AM to fuse and extracts features. Their model achieves SOTA results on dataset ARID [164].

In self-driving VAR, existing methods often struggle to capture long-term temporal relationships and effectively aggregate discriminative representations. To address this, Li et al. [165] proposed a Self-attention Pooling Long-term Temporal Network (SP-LTN) in 2022. SP-LTN uses self-attention pooling to weight representations based on their importance, highlighting discriminative contributions. Using only RGB frames, SP-LTN outperforms previous models on UCF101 and HMDB51.

In 2024, Xia and Wen [166] proposed a VAR model. First, a self-AM captures regions of interest and train a keyframe sampling module to select key action frames. Then a deep feature model captures appearance, motion, and difference features, which are fed into BiLSTMs, adaptively weighted, fused, and classified. The model outperforms previous ones on HMDB51, UCF101, Kinetics-400, and Something-Something V2.

Self-AM based models, particularly those derived from the Transformer architecture, have proven successful in various applications, including VAR. However, they do have their limitations. Some common challenges and potential suggestions for improvements are as follows:

1. *Computational complexity*: Self-AM based models, particularly those that calculate pairwise attention between all pairs of frames, can be computationally intensive and require a lot of memory. This becomes more challenging with the increase in video length. Approximate AMs or sparse attention methods can be used to reduce this complexity. Also, efficient Transformer variants and attention models designed specifically for sequential data, such as the Longformer, can be used.
2. *Lack of local feature capture*: Self-AM based models are great at capturing global dependencies but might overlook local spatiotemporal information that is typically captured by convolution operations. Hybrid models that combine CNNs and self-AMs can be an effective way to capture both local and global dependencies.
3. *Sensitivity to noise*: Self-AM based models can be sensitive to noise and irrelevant frames in the video, which can negatively impact their performance. The use of a more robust preprocessing stage, such as frame selection or noise reduction techniques, could help to address this. Also, methods for learning robust representations, such as contrastive learning, can also help to make models more resistant to noise.
4. *Long-term dependencies*: Despite the strength of self-AM in modelling dependencies, capturing long-term dependencies in lengthy videos still remains a challenge. This could be improved by incorporating additional architectural elements designed to better handle temporal information, such as recurrent layers or temporal convolution layers.

While self-AM based models have brought considerable advancements in the field of VAR, they do come with challenges related to computational complexity, local feature capture, and handling long-term dependencies. Ongoing research is needed to further improve these models and mitigate their limitations.

5.6. Models based on hierarchical attention

Hierarchical AM has emerged as a promising methodology for VAR. Hierarchical AM based models for VAR follow a multi-step process to analyse and classify the actions in a video. These steps generally include feature extraction, hierarchical processing, AM, and action classification.

After feature extraction, the hierarchical structure comes into play. Lower-level features extracted from frames or short clips are aggregated and processed to form higher-level features representing more extended periods of the video. This hierarchical processing allows the model to capture both fine-grained details and broader context, which is crucial for understanding some actions.

In 2016, Wang et al. [143] proposed the Hierarchical Attention Network (HAN) to incorporate static spatial information, short-term motion, and long-term temporal structures for action understanding. HAN efficiently captures long-range temporal structures, models temporal transitions between frame chunks, and uses a multi-step spatiotemporal attention module to learn important regions and segments. Trained and evaluated on UCF101 and HMDB51, it significantly outperformed previous models.

In 2018, Yan et al. [167] proposed a attention network, namely Hierarchical Multi-scale Attention Network (HM-AN), by incorporating the AM into the Hierarchical Multi-scale RNN and applying it to VAR. The experimental results demonstrated the improved effect of HM-AN over LSTM with attention on the vision task.

In 2019, Sang, Zhao, and He [168] proposed TAMNet, an end-to-end trainable network for VAR. TAMNet uses a recurrent region attention model to capture action-relevant spatial information and a video frame attention model to highlight important frames, reducing interference from similar sequences. On UCF101 and HMDB51, their model effectively focuses on important frames and captures action-relevant regional visual information.

Public gesture datasets often lead to astronaut gesture recognition. To address this, in 2020, Gu, Zhang, and Wang [169] proposed an astronaut gesture dataset (DSSL) and the Hierarchical Attention Single-Shot Detector (HA-SSD). HA-SSD uses lightweight MobileNet as a feature extractor and a hierarchical attention module to enrich features. The dataset is suitable for space station use, and the model effectively localises and recognises gestures with strong generality.

Hierarchical AM based models are designed to handle VAR tasks by using the hierarchical nature of videos, which are composed of a temporal sequence of frames, each containing spatial information. However, these models have certain limitations:

1. *Complexity and computation cost*: Hierarchical AM based models add an additional layer of complexity to the network structure. This results in a higher computational cost, particularly for high-resolution videos with some frames. Research into more efficient hierarchical AMs and computation-friendly model architectures could lead to significant improvements in efficiency and scalability. Moreover, the use of hardware acceleration techniques can help to mitigate computational costs.
2. *Temporal sequence length*: Hierarchical models require processing of long temporal sequences, which can be challenging. For example, modelling long-term temporal dependencies can be difficult due to issues like gradient vanishing. The use of architectures designed to handle long sequences, such as LSTM or GRU, can help with this problem.

While hierarchical AM based models have significantly improved VAR, they do present challenges such as high computational costs, and difficulty with long temporal sequences. Future research should aim to address these limitations to further improve the effectiveness of hierarchical AM based models in VAR tasks.

5.7. Models based on channel attention

Channel AM for VAR represent a significant evolution in DL architectures, aiming to better use the interdependencies among channels of convolutional feature maps. These models consider that not all channels (or features) contribute equally to the recognition of an action in a video sequence.

Conventional CNN architectures do not explicitly model the interdependencies between channels, which can lead to less optimal feature representation. Channel AM based models, on the other hand, learn to weigh the importance of each feature channel, focusing on the most relevant features for the AR task.

An overview of how channel AM based models work is as follows:

1. *Feature extraction*: For each video, spatial and temporal features are extracted from a sequence of frames using convolution layers. This results in a set of feature maps.
2. *Channel AM*: The channel AM computes a set of weights, with one weight for each feature channel. These weights determine the importance of each channel for the task at hand. This is typically done using operations like global average pooling and fully connected layers.
3. *Rescaling feature maps*: The weights rescale the feature maps, typically through multiplication. Important feature channels have a greater influence on the output, while less important channels have less.
4. *Classification*: The rescaled feature maps are then passed to the subsequent layers of the network for action classification.

In addition, channel AM based models can be integrated with other types of attention based models, such as spatial attention based models or temporal attention based models, to create more complex and powerful architectures for VAR.

In 2019, Wu, Ma, and Li [170] advanced VAR using CNNs, addressing limitations in capturing long-term temporal dependencies and focusing on relevant motion areas. They use dynamic image sequences and a Channel and STIPs (Spatiotemporal interest points) Attention Model (CSAM) to enhance focus on key channels and motion regions. Channel attention is implemented channel-wise, while STIPs attention projects the detected STIPs onto the feature map. Refined features are processed via global average pooling and an LSTM. Their model outperforms previous depth models on three RGB-D datasets.

In 2021, Wang et al. [171] presented a video architecture, termed as Temporal Difference Network (TDN), with a focus on capturing multi-scale temporal information for efficient AR. The core of the TDN is to devise an efficient temporal module by explicitly leveraging a temporal difference operator, and systematically assess its effect on short-term and long-term motion modelling. They compared different implementation forms of short-term temporal difference module and long-term temporal difference module, including spatiotemporal attention, channel attention, and residual connection combining attention. Their experiments show that channel attention is not the best.

In 2021, Ullah et al. [172] proposed an attention-based LSTM network for sports AR. DenseNet processes input frames, then refined by a channel and spatial attention module. The refined features are input to a BiLSTM, followed by a fully connected layer and softmax classifier. Promising results were achieved, with potential improvements suggested, such as incorporating temporal attention, using optical flow, and stacking LSTM cells.

To reduce 3D-CNN parameter counts, Chen et al. [173] proposed Spurious-3D Residual Attention Networks (S3D RANs) in 2022. S3D RANs use 2D-CNNs on single-view volumetric video frames to learn temporal motion features. View- and channel-wise attention modules within the residual units learn view importance and focus on relevant information. The model achieves higher accuracy and lower complexity than existing methods on UCF101 and HMDB51.

To minimise computation in VAR, models must reduce frame number, size, and resolution. In 2024, Dastbaravardeh et al. [174] proposed an approach optimising AR with CNNs, channel attention, and autoencoders, focusing on low-resolution videos. The model selects discriminative features and uses random frame sampling to prevent overfitting. On UCF50 [175], UCF101, and HMDB51, it achieves 77.29%, 98.87%, and 97.16% accuracy, respectively.

While channel AM based models for VAR have achieved significant success in various applications, they are not without their limitations. A few common challenges and potential suggestions to address them are as follows:

1. *Complexity and computation cost*: Channel AM based models add a new level of complexity to the network architecture. Computing attention weights for each channel can be computationally expensive, especially for videos with high resolutions and some frames. To mitigate this, you could employ more efficient AMs or approximations that can compute attention in a more computationally efficient way. Research into lightweight AMs is ongoing and may yield promising solutions.
2. *Difficulty in capturing long-term dependencies*: Channel AMs can focus on relevant features but struggle with long-term temporal dependencies, especially in long videos. This can be addressed by incorporating components designed for temporal dependency capture, *e.g.*, recurrent layers or temporal AM.
3. *Lack of spatial attention*: Channel AM based models focus on the interdependencies between channels, but they do not explicitly pay attention to different spatial regions in the video frames. Combining channel attention with spatial AMs can help the model focus on the most relevant features both across channels and within each channel. This can lead to a more accurate understanding of the video content.

5.8. Models based on multi-head attention

Multi-head AM for VAR have demonstrated the ability to capture various aspects of input video data by distributing the process of attention across different learned linear projections. This technique stems from the Transformer architecture proposed by Vaswani et al. [160] in 2017 for the task of machine translation.

In a multi-head AM based model, the AM is not just computed once, but multiple times in parallel, each time with different learned linear transformations of the input. This way, each attention head can capture different aspects of the input. For example, one head might learn to focus on spatial features, while another head might learn to focus on temporal features. This is particularly useful in VAR where spatial and temporal dependencies play a key role in accurately identifying actions.

A basic overview of how multi-head AM based models work in the context of VAR is as follows:

1. *Feature extraction*: Each video frame is passed through a CNN to extract spatial features. These spatial features are usually concatenated with temporal features to consider the sequence of frames.
2. *Multi-head AM*: The extracted features are then passed through a multi-head AM. Each “head” in this mechanism computes a distinct attention score by applying different learned linear transformations to the input features. This allows the model to capture different types of relationships between video frames.
3. *Concatenation and transformation*: The attention outputs from each head are then concatenated and passed through a linear transformation to produce a single output.
4. *Classification*: This output is then used to predict the action class in the video, typically via a fully connected layer and a softmax function.

Multi-head AM based models can be very effective for VAR because they can capture a richer set of features by paying attention to different parts of the input simultaneously. They have been used in various SOTA architectures for video understanding, providing significant improvements over models that do not use AMs.

In 2019, Girdhar et al. [176] proposed the Action Transformer for AR and localisation. They repurposed a Transformer architecture to aggregate spatiotemporal context features. Their multi-head/layer architecture learns to attend to relevant regions of the person and their context. On AVA [177], their model significantly outperformed the previous models of using only RGB frames.

In 2020, Wang, Peng, and Qiao [178] proposed the Cascade multi-head Attention Network (CATNet) for VAR. CATNet uses two-level attention: multi-head local self-attention and relation-based global attention. Starting with segment features from a backbone network (I3D), CATNet learns the importance of local features and integrates them into global representations. It then learns global information attention relationally. On Kinetics, HMDB51, and UCF101, CATNet significantly boosted the baseline network.

Two-stream VAR methods often struggle with similar actions. To address this, Zhou et al. [179] proposed a Multi-head Attention-based Two-stream EfficientNet (MAT-EffNet) in 2023. MAT-EffNet extracts spatial and temporal features using EfficientNet and uses multi-head attention to capture key action information. Late average fusion yields the final prediction. MAT-EffNet outperformed previous models on UCF101, HMDB51, and Kinetics-400.

In 2024, Hussain et al. [180] proposed an AR model with a two-stream parallel BiLSTM and Dual Stream Multi-Head Attention (DSMHA). One stream uses a frozen ViT [181] for contextual information. The other fuses ViT features with FlowNet2 [68] output for motion features. Parallel BiLSTMs capture global semantics. The model achieves 78.63%, 96.02%, and 98.88% accuracy on HMDB51, UCF101, and YouTube Actions [134], respectively.

Multi-head AM based models have gained significant attention for their ability to effectively capture complex relationships within video data for AR tasks. However, like all models, they come with their own set of limitations. A few common challenges associated with these models and potential suggestions for addressing them are as follows:

1. *Computational complexity*: Multi-head AM based models can be computationally expensive and memory-intensive. This is due to the fact that they calculate attention scores for each pair of frames in the video across multiple heads, which can quickly become computationally prohibitive for long videos. To address this issue, could use techniques like approximate attention computation or sparse attention methods that focus on a subset of the most important frames. In addition, reducing the number of attention heads or using efficient Transformer variants could help decrease computational costs.
2. *Difficulty in interpretability*: Despite their performance, multi-head AM based models can often act as a black box, making it difficult to interpret which aspects of the video the model is paying attention to. Visualisation techniques that illustrate attention weights can aid interpretability, showing which frames the model finds most relevant. Moreover, incorporating explainability in the model design could also improve its interpretability.

While multi-head AM based models have achieved significant success in VAR tasks, there are inherent challenges such as computational complexity, and difficulty in interpretability. Future research directions could focus on developing more efficient, robust, interpretable, and hybrid models to address these limitations.

5.9. Models based on transformer

In recent years, Transformer models have made significant progress in the field of computer vision, particularly in VAR tasks [182,183]. The core advantage of the Transformer architecture lies in its powerful self-AM, which enables it to capture long-range dependencies and handle large-scale data effectively. In the context of VAR, Transformer models not only model spatial information but also capture dynamic changes across the temporal dimension, making them highly effective in a range of complex tasks. Transformer-based models such as ViT [181], Swin Transformer [184,185], VideoSwin [186], and TimeSFormer [187] have emerged as powerful tools in the VAR domain.

5.9.1. Key spatial–temporal transformer designs for VAR

Recent advancements in Transformer architectures for VAR focus on leveraging their powerful self-AMs to model complex spatial and temporal dependencies, enabling significant improvements in accuracy and efficiency across various VAR tasks. This section explores how Transformers are adapted for VAR by incorporating spatial and temporal features.

The four key spatial–temporal Transformer designs are as follows:

1. *ViT (Vision Transformer)*: In 2020, Dosovitskiy et al. [181] proposed ViT, a foundational model adapting Transformers to vision. ViT divides images into patches, linearly embeds them, and processes them via a Transformer. For VAR, video frames are treated as patch sequences, with the Transformer's self-attention capturing temporal dependencies. This effectively captures long-range spatiotemporal dependencies, suitable for complex AR tasks.
2. *Swin Transformer*: In 2021, Liu et al. [184] introduced The hierarchical vision Transformer, called Swin Transformer. It uses a shifted window approach to capture local and global dependencies. It reduces computation by partitioning images into non-overlapping windows and applying self-attention within them. For VAR, Swin Transformer effectively models spatiotemporal information, capturing local temporal patterns at high resolution and global patterns in higher layers, making learning both fine-grained and broad temporal relationships efficient.
3. *VideoSwin Transformer*: In 2022, Liu et al. [186] proposed VideoSwin Transformer, extending Swin Transformer to VAR by incorporating a temporal dimension. It integrates spatial and temporal attention modules to capture long-range dependencies and local patterns. Using a 3D shifted window attention module, it extends windows into the temporal dimension. Demonstrating strong performance on Kinetics and Something-Something V2, VideoSwin effectively models temporal relationships. Combining 2D and 3D self-attention, it scales well and surpasses traditional 2D convolutions and earlier Transformers in tasks requiring understanding of motion and spatial context.
4. *TimeSFormer*: In 2021, Bertasius, Wang, and Torresani proposed TimeSFormer [187] for VAR, treating time and space as critical. Unlike traditional Transformers, TimeSFormer separates temporal and spatial attention. Spatial attention focuses on intra-frame relations, while temporal attention captures inter-frame dynamics. By decoupling these attentions, TimeSFormer efficiently captures both fine-grained spatial details and long-range temporal dependencies, making it suitable for VAR.

The essential characteristics of these Transformers are as follows:

- *ViT*: Uses global self-attention, extended to video by treating frames as sequences of image patches.
- *Swin Transformer*: Introduces hierarchical and local window-based attention, adapted for VAR by using spatio-temporal attention.
- *VideoSwin*: Extends the Swin Transformer by adding a 3D AM, incorporating both spatial and temporal relationships.
- *TimeSFormer*: Separates spatial and temporal AMs to better handle both spatial content and temporal dynamics in video sequences.

Table 7
Comparison of the models based on AM.

Type of AMs	Model	Input	Framework
Spatiotemporal	Method in [151]	Video frames	CNN+spatial attention+temporal attention+LSTM
	STAN [155]	Video frames	Multiple streams (CNN+attention+LSTM)
	STA-CNN [158]	RGB frames and optical flow	Action Classification Network+Spatial Attention Network+Temporal AM
	STA-TSN [159]	Video frames	CNN+attention+TSN
Self	Method in [161]	RGB frames/optical flow/trajectory maps	A three-stream network with the I3D model as the backbone+a bidirectional self-attention network
	Method in [162]	Video frames	A 3D-CNN+attention
	DarkLight [163]	Dark videos and their brightened counterpart (enhanced through Gamma Intense Correction)	Dual-pathway structure with weight-shared CNNs+ attention
	SP-LTN [165]	Video frames	Pipeline (a 2D-CNN based architecture for spatial feature learning, a 3D-CNN-based architecture for temporal feature learning, and self-attention pooling)
	MS-KFS [166]	Video frames (key RGB-based frames and key Flow-based frame)	Multi-stream network with Bi-LSTM
	Method in [182]	Skeleton data (from video dataset)	Two-stream spatial–temporal Transformer network
Hierarchical	HAN [143]	RGB frames and optical flow	Two stream of CNNs (VGG) and LSTM networks+attention
	HM-AN [167]	Video frames	CNN (Residue-152) +RNN+attention
	TAMNet [168]	Video frames	CNN+two-stream BiLSTM+attention
	HA-SSD [169]	Gesture images	The HA-SSD model consists of a lightweight backbone named MobileNet and a hierarchical AM
Channel	CSAM [170]	Video frames	CNN (VGG16)+attention+LSTM
	TDN [171]	Video frames	TDN with a ResNet backbone+attention
	Method in [172]	Video frames	Densenet+attention+LSTM
	S3D RANs [173]	Video frames	ResNet-50 [39] as backbone with view and channel-wise attention submodule
	Method in [174]	Video frames	CNN with channel AM and autoencoders
Multi-head	Action Transformer [176]	Video frames	I3D+Transformer
	CATNet [178]	Video frames	Use an I3D network as the backbone feature extraction module
	MAT-EffNet [179]	Video frames	The Two-stream CNN
	Method in [180]	Video frames	Two-stream+parallel BiLSTM+DSMHA
Spatial+self	TP-ViT [191]	Video (two pathways with different frame rates and resolutions)	Two-pathway ViT
Multi-head+self	ViT-ReT [193]	Video	ViT+ResNet50

5.9.2. Specialised transformer-based models for enhanced VAR

Some of the recent approaches have built upon well-established Transformer architectures, such as ViT, while others have introduced innovative ideas and methods to address the unique challenges of VAR. This subsection will discuss some of them.

In 2021, Plizzari, Cannici, and Matteucci [182] proposed a spatiotemporal Transformer network for skeleton-based AR. The model uses spatial and temporal self-attention modules to capture intra- and inter-frame body part interactions. These modules are combined in a two-stream network. It achieves SOTA performance with joint coordinates and competitive results with bone information on NTU RGB+D 60 [188], NTU RGB+D 120 [189], and Kinetics Skeleton 400 [58,190].

In 2022, Jing and Wang [191] proposed a Two-Pathway ViT (TP-ViT) for VAR, addressing the limited exploration of multiple pathways in Transformer models. TP-ViT uses two parallel spatial Transformer encoders at different frame rates and resolutions. The high-resolution pathway captures spatial details, while the high-framerate pathway focuses on temporal dynamics. A temporal Transformer processes fused pathway outputs. Skeleton features are also incorporated. The model achieves SOTA performance on Kinetics and FineGym [192].

In 2023, Wensel, Ullah, and Munir [193] proposed a AR framework (named ViT-ReT) using two Transformer-based models: a recurrent Transformer for sequence prediction and the ViT for feature extraction. The framework is compared with traditional CNN-RNN models across four public datasets. The model achieves a 2× speedup over the baseline ResNet50-LSTM approach while maintaining similar accuracy. Additionally, it outperforms previous models in both accuracy and runtime, demonstrating its suitability for real-time and resource-constrained environments.

5.9.3. Limitations and suggestions

These models based on Transformers offer powerful approaches for capturing both spatial and temporal dependencies in video data. However, they have some common limitations summarised as follows:

1. *Data efficiency*: Most of these models require large-scale annotated video datasets for training, which may not always be available or feasible to collect.
2. *Real-time constraints*: Despite some advances in runtime efficiency (*e.g.*, ViT-ReT), the models still face challenges when it comes to real-time applications, especially when processing high-resolution or high-frame-rate video streams.
3. *Memory consumption*: The large parameter space of Transformer-based models often leads to high memory consumption, making them challenging to deploy on memory-limited devices or edge devices.

To address the limitations of Transformer-based models for VAR, several strategies can be applied: using transfer learning to improve data efficiency and adopting hierarchical or sparse AMs to enhance scalability. Additionally, domain adaptation, multi-modal learning, and hybrid architectures (combining Transformers with CNNs or RNNs) can help improve generalisation and task-specific performance across diverse video datasets and environments.

Overall, Transformer-based methods have shown impressive results for VAR by leveraging their ability to capture complex temporal and spatial dependencies. As research continues, future Transformer-based models may focus on improving efficiency and handling larger video datasets in real-time scenarios.

Table 8
Comparative analysis of different AMs in VAR.

AM	Used by	Focus	Strength	Limitation	Performance boost
Spatiotemporal attention	Meng et al. [151]; Li et al. [155]; Yang et al. [158]; Yang et al. [159]	Regions/frames of interest	Captures local motion	Misses global context	High (localised actions)
Self-attention	Vaswani et al. [160]; Purwanto et al. [161]; Anvarov, Kim, and Song [162]; Chen et al. [163]; Xia and Wen [166]; Plizzari, Cannici, and Matteucci [182]; Jing and Wang [191]; Wensel, Ullah, and Munir [193]	Global temporal dependencies	Strong long-range understanding	Computationally expensive	High (extended actions)
Hierarchical attention	Wang et al. [143]; Yan et al. [167]; Sang, Zhao, and He [168]; Gu, Zhang, and Wang [169]	Hierarchical data structure	Extract global & local features across levels	Complex and harder to optimise	High (hierarchical tasks, *e.g.*, video analysis)
Channel attention	Wu, Ma, and Li [170]; Wang et al. [171]; Ullah et al. [172]; Chen et al. [173]	Feature map prioritisation	Lightweight enhancement	Lacks temporal focus	Moderate
Multi-head attention	Girdhar et al. [176]; Wang, Peng, and Qiao [178]; Zhou et al. [179]; Hussain et al. [180]; Wensel, Ullah, and Munir [193]	Multiple attention heads	Focuses on different data parts simultaneously	High computational cost	Very high (deep understanding in complex tasks)

5.10. Comparison and limitation

Table 7 compares the above AM-based models mainly in input, framework, and type of AMs. The impact of various inputs, frameworks, and types of AMs on model performance is summarised as follows:

- *Inputs*: Video frames are commonly used across models (STAN, STA-TSN, SP-LTN, etc.), with variations like RGB frames, optical flow, and trajectory maps in some methods. Input data choice (video frames vs. flow) impacts the model's ability to capture temporal and spatial dynamics in AR.
- *Frameworks*: A range of hybrid architectures is used, combining CNN with temporal structures like LSTM (*e.g.*, STAN, Method in [151], TAMNet), or 3D-CNN (*e.g.*, SP-LTN, CATNet, etc.). Some methods, such as DarkLight, use dual-pathway CNNs with attention to enhanced feature learning, especially in low-light conditions. Models like Method in [161] employ a three-stream network to capture diverse information from different inputs (RGB, optical flow, and trajectory).
- *Types of AMs*: (1) Spatiotemporal attention is a key mechanism applied across many models (STAN, STA-CNN, etc.), helping in focusing attention on the most informative spatial and temporal regions in the video. (2) Self-attention (used in models like Method in [161], MS-KFS) aids in identifying important features by dynamically re-weighting input data without relying on fixed feature maps. (3) Hierarchical AMs (like those in HAN and TAMNet) allow the model to focus on different levels of information, improving multi-level representation. (4) Channel attention (used in CSAM, TDN, and others) focuses on relevant channels, improving feature extraction for more complex actions. (5) Multi-head attention in models like Action Transformer allows the model to attend to different subspaces, which could enhance its ability to capture diverse action patterns across the frames.

The combination of different input types, model frameworks, and AMs plays a crucial role in improving the performance of VAR models. The choice of AM, in particular, helps to direct the model's focus to the most relevant spatial, temporal, or channel-specific features, thus enhancing the accuracy of AR tasks.

Despite the significant improvements that AMs have brought to VAR, these methods still have several limitations:

1. *Computational intensity*: AM-based models, especially those that use self-attention (like Transformers), require significant computational resources. This is because self-attention involves calculating the attention score for each pair of frames in a sequence, which can be computationally expensive for long videos.
2. *Memory usage*: Because AMs calculate the relationship between each pair of frames in a sequence, they can require significant memory, particularly for long sequences. This makes it challenging to apply these models to very long video sequences without substantial computational resources.
3. *Lack of temporal order sensitivity*: Although AMs can capture long-range dependencies between video frames, they may not be as sensitive to the temporal order of frames as models like RNNs or LSTM Networks, which explicitly model the temporal dynamics of sequences.

5.11. Summary

Table 8 shows which AMs are used by which papers reviewed in this section, the distinctions among them, and their impact on model performance. Interconnections among these AMs are as follows:

- *Shared objectives*: All AMs aim to enhance feature representations by focusing computational resources on the most informative parts of video sequences. This is achieved by directing attention spatially, temporally, or across modalities.
- *Common techniques*: Most AMs integrate seamlessly with CNNs, RNNs, or Transformers to enhance temporal and spatial feature extraction. They utilise techniques such as softmax-normalised weights, self-attention, and multi-head attention for improved feature extraction.
- *Hybrid usage*: Some models, like hierarchical attention, combine multiple AMs to capture intricate spatiotemporal relationships, such as integrating channel, spatial, and temporal attention.

It is clear that these AMs (*i.e.*, spatiotemporal attention, self-attention, hierarchical attention, channel attention, multi-head attention, and cross-modal attention) have dramatically improved the performance of VAR models by enabling them to focus on the most relevant aspects of the input data. AMs have provided a more dynamic and contextual approach to understanding videos, allowing models to recognise actions more accurately by considering both spatial and temporal aspects of the data.

Despite existing limitations, the evolution and adaption of AMs continue to demonstrate immense potential in improving VAR. The ongoing research and development in this area are expected to yield effective solutions to these challenges, paving the way for more sophisticated and efficient video understanding systems in the future.

6. VAR models based on other DL models

In this section, we will examine VAR models based on other DL models.

6.1. Models based on 2D-CNNs

Conventional 2D-CNNs have shown great effectiveness in various image analysis tasks. VAR models based on 2D-CNNs extract the features of each frame image separately, such as those in [34,194–196] before 2020.

3D-CNNs are computationally expensive, while 2D-CNNs, though efficient, often require optical flow calculation. To address this, Wu and Chiu [197] proposed a 2D ResNet-50 based VAR model in 2021. Instead of optical flow, they use a multi-scale temporal shift module combined with a temporal feature difference extraction module. The temporal shift module interacts with convolutions to extract spatiotemporal features. Their model achieves 96.25% and 72.83% accuracy on UCF101 and HMDB51, respectively.

3D-CNNs are computationally expensive, and (1+2)D CNNs can neglect motion. To address this, in 2022, Joefrie and Aono [198] proposed a model that can efficiently capture spatiotemporal and motion features via motion extraction, multi-view extraction, and densely connected temporal aggregation. Injected into 2D ResNet-50, their model outperformed previous CNN methods on Something-Something V1 and Jester [199], and achieves competitive results on Moment in Time [200].

However, when coming to VAR, 2D-CNNs have some limitations:

1. *Lack of temporal information*: 2D-CNNs are designed for static image analysis and do not consider the temporal dimension. Videos, unlike images, have a temporal aspect.VAR typically requires understanding of the changes and movements across frames, which is not directly possible with 2D-CNNs.
2. *Difficulty in capturing complex actions*: Simple actions may be discernible through a series of images, but more complex actions that involve complicated sequences of movements can be hard to capture and recognise using 2D-CNNs.
3. *Lack of spatiotemporal interaction*: The interaction between spatial and temporal information is essential for understanding the context and progression of actions in videos. 2D-CNNs are limited in capturing such spatiotemporal interactions.

2D-CNNs need to incorporate other technologies to compensate for their inability to learn temporal relationships, such as introducing AMs to the 2D-CNN, and combination with RNNs.

6.2. Models based on multi-stream/channel CNNs

The introduction of two-stream networks has inspired a lot of subsequent research, resulting in multi-stream networks that integrate multiple modals as input data types.

In 2018, Liu and Yang [201] proposed a multi-stream 2D/3D-CNN network for VAR, using RGB images, optical flows, and gradient maps for feature extraction. An attention block and category/network weights further enhanced performance. While achieving competitive results, the model is less efficient due to complexity and long training times. Further evaluation on more complex datasets is also needed.

Complex video backgrounds may hinder AR. To address this, in 2023, Zong et al. [202] proposed a four-stream network with multi-task learning using spatial and temporal saliency, called STSF. The network comprises appearance, motion, spatial, and temporal saliency streams. Multi-task learning-based LSTM shares knowledge between CNN features, capturing long-term dependencies. Their model outperforms previous ones on UCF101, HMDB51, and Kinetics.

Multi-stream CNNs have proven to be quite effective for VAR tasks. However, there are several limitations and challenges associated with multi-stream CNNs:

1. *Computational intensity*: Multi-stream CNNs require significant computational resources. They have to process several streams concurrently, increasing the computational load and memory usage. Training such models require high-end GPUs, making them less accessible for researchers with limited resources.
2. *Difficulty in handling variable-length videos*: Most DL models, including multi-stream CNNs, are designed to handle fixed-length input. However, videos can have variable lengths, which necessitates additional preprocessing steps to standardise the input.
3. *Difficulty in combining information across streams*: One of the main challenges in multi-stream architectures is effectively combining information from different streams. There is no universally optimal method to do this and it often requires careful tuning and experimentation.

Some suggestions to improve the performance of multi-stream CNNs for VAR are as follows:

1. *Efficient architectures*: Use more efficient neural network architectures that require fewer computational resources. For instance, MobileNet, EfficientNet, and ResNet offer good trade-offs between accuracy and computational cost.
2. *Hybrid models*: Use a hybrid model that leverages the strengths of 3D-CNNs, which can inherently learn spatial and temporal features, with the multi-stream CNN approach. For instance, a Two-stream 3D-CNN could potentially offer better performance.
3. *AMs*: AMs can help the model focus on the most informative parts of the video at each point in time, improving the model's ability to recognise actions.

6.3. Models based on graph convolutional networks

Graph Convolutional Networks (GCNs) is a type of DL model specifically designed to process graph-structured data [203]. Their application in VAR has gained increasing attention due to the graph-like relationships between different video parts. In the context of AR, the main challenge is accurately capturing the temporal and spatial relationships between various elements in a video, such as human body joints or objects in the scene. GCNs excel at modelling dependencies between connected nodes, making them particularly well-suited for this task, especially when the nodes are derived from human skeletons or scene graphs [190,204].

Graph-structured data in VAR mainly as follows:

- *Object-based data*: In human-object interactions (*e.g.*, "picking up a ball" or "kicking a soccer ball"), the objects can be treated as nodes in a graph, with edges representing the spatial or temporal relationships between them.
- *Human pose and keypoint data*: Pose based on the spatial arrangement of body parts (*e.g.*, head, arms, legs) can be modelled as a graph. Each body part becomes a node, and the edges represent the physical connections (*e.g.*, shoulder to elbow or wrist to hand).
- *Optical flow and dense trajectories*: Each flow vector is treated as a node in a graph, and edges are formed based on spatial and temporal proximity.
- *Frame-level data*: Each frame of the entire video can also be considered as a graph node, with edges connecting adjacent or similar frames.

6.3.1. Recent advancements in GCN-based VAR models

In 2020, Jang, Kim, and Lee [205] proposed a VAR model based on a Spatial-Temporal GCN (ST-GCN). The model takes optical flow and image gradients as input. Optical flow and gradients are expressed in the descriptor style of HoG. Each node in the network corresponds to a region where optical flow and gradients are measured. On the NTU RGB+D dataset, the model achieves an impressive Top-1 recognition accuracy of 93.88%.

In 2022, Yenduri, Chalavadi, and Mohan [206] proposed a graph-based VAR framework to model spatiotemporal interactions without object-level supervision. They used the Harris 3D detector to obtain salient space–time interest points (STIPs), built a graph with STIPs as nodes connected by spatial and temporal edges, and employed GCN for reasoning. On UCF101, HMDB51, and Something-Something V2, the model's efficacy is confirmed; its accuracies are 98.1%, 85.3%, and 68.2%, respectively.

In 2023, Wang et al. [207] proposed STIGPN for human-object interaction VAR. Their method extracts visual-spatial and spatial-semantic features, processed by two streams to capture spatiotemporal features. Results are fused for final recognition. An S-STGC constructs a spatiotemporal graph (human and object as nodes), and a "GCN+RNN" extracts spatiotemporal representations. On CAD-120 [208], Something-Else [209], and Charades [210], the model achieves competitive results.

In 2024, Liu et al. [211] proposed a GCN-based multi-modality fusion network that efficiently integrates RGB and skeleton data. They developed a cross-modality data mapping method to represent RGB frames as graph data and created a GCN fusion module to combine features from both modalities. Additionally, a spatio-temporal joint AM was proposed to enhance the learning of action-specific features. This model supports end-to-end training, simplifying the process and achieving strong performance on the NTU RGB+D 60 and 120 datasets.

6.3.2. Limitations and future directions

GCNs have become an increasingly popular approach for VAR, where the underlying structure of the data can be represented as a graph. However, there are some inherent limitations to GCN-based methods:

- *Limited temporal modelling*: GCNs typically focus on the spatial relationships between nodes (such as joints in skeleton-based models) and may not effectively capture long-range temporal dependencies. The temporal dimension, which is crucial for AR, might be inadequately modelled using conventional GCNs, leading to reduced performance in complex AR tasks where motion dynamics span across many frames.
- *Challenges in handling occlusions and missing data*: In real-world scenarios, skeleton-based data can suffer from occlusions or missing joint information, especially when the subject is partially obscured. GCNs may struggle to handle incomplete graphs effectively, leading to performance degradation.
- *Scalability Issues*: As videos can contain a large number of frames, the number of nodes in the graph increases. The computational complexity of GCNs can grow substantially with the size of the input, leading to scalability issues in VAR tasks. This is particularly challenging when dealing with long-duration sequences.

Future directions and suggestions for GCN-Based VAR are as follows:

- *Integrating temporal modelling*: Incorporating advanced temporal modelling techniques like temporal graph convolutions or RNNs/LSTMs can help capture long-range temporal dependencies. This can improve GCN's ability to model the dynamic nature of actions over time.
- *Multi-modal inputs*: Incorporating other modalities, such as RGB frames, depth maps, or optical flow, alongside skeleton data, could provide more information about the environment, enhancing the model's ability to recognise complex actions. This fusion of skeleton-based features and appearance-based features could help to compensate for the limitations of relying solely on skeletal representations.
- *Robustness to missing data and occlusions* : Research into robust graph neural networks that can handle missing or noisy data is essential. Methods like graph convolutional autoencoders (GCNs with unsupervised learning) could be used to reconstruct missing parts of the graph. Additionally, techniques such as spatial–temporal interpolation or graph AMs could help GCNs adapt to incomplete or occluded skeletons during video processing.
- *Transfer learning for generalisation*: Pretraining models on large-scale datasets with diverse actions and human poses could improve the generalisation ability of GCN-based models. Transfer learning from large datasets like COCO [212] or ImageNet, followed by fine-tuning on specialised AR datasets, could reduce the overfitting problem and improve cross-dataset performance.

6.4. Models based on hybrid methods

As highlighted above, recent advancements in VAR have been propelled by DL models, including two-stream networks, 3D-CNNs, RNN/LSTMs, AMs, GCNs, and others. While each of these models excels in certain areas, they also come with inherent limitations:

- Two-stream networks capture spatial and motion information but struggle with long-term temporal dependencies.
- RNN/LSTMs are effective for sequential data but depend heavily on the quality of input features.
- 3D-CNNs provide strong spatiotemporal feature extraction but are computationally expensive.
- AMs enhance the focus on relevant features but may introduce increased computational complexity and slower training times.
- GCNs effectively model spatial and temporal relationships in skeletal data but often rely on fixed graph structures, which limits their adaptability. They also underutilise second-order information, such as joint motion and bone directions, reducing their potential to recognise complex actions.

Hybrid models address these challenges by combining multiple architectures to leverage their complementary strengths.

1. *Improved accuracy*: By leveraging the complementary strengths of different models, hybrid architectures often outperform single-model systems.
2. *Better generalisation*: Hybrid models can better generalise diverse datasets by integrating different information types. While one model may excel at capturing certain features (*e.g.*, appearance or motion), another can focus on different aspects, such as temporal relationships, creating a more robust overall system.
3. *Enhanced temporal and spatial feature extraction*: Combining 3D-CNNs with LSTMs or two-stream networks enables hybrid models to capture spatial and temporal features simultaneously. This is especially useful for recognising actions that rely on motion dynamics and visual appearance across video sequences.
4. *Resilience to noise and variability*: Combining multiple models enhances the strength of hybrid architectures to noise, occlusions, and other video artefacts. For instance, if one model encounters difficulties with occluded objects, another model may still be able to focus on temporal cues or alternative visual information. This integration ultimately improves the overall robustness of the system.

Hybrid models combine the strengths of various architectures to offer significant improvements in accuracy, generalisation, feature extraction, and resilience to noise. By addressing the challenges faced by single-model systems, hybrid architectures provide a more robust and versatile solution, signalling a promising future for more sophisticated and reliable AR systems.

6.4.1. Typical hybrid models

Some common hybrid approaches and their applications are as follows:

- *Two-stream+3D-CNN hybrid models*: The Two-stream+3D-CNN hybrid architecture involves two parallel networks, each using 3D-CNNs to capture different aspects of video data. One network focuses on spatial features, while the other captures motion or temporal patterns. In parallel, these networks process distinct inputs, such as RGB frames, optical flow, or other representations, allowing for effective spatiotemporal feature extraction over the entire video sequence. The results from both networks are then fused to create a comprehensive video representation, enhancing AR performance. Notable models using this architecture include I3D [77] and SlowFast [99].
- *Two-stream+RNN/LSTM-based models*: The Two-stream+RNN/LSTM hybrid architecture consists of two parallel networks: one for extracting spatial features from RGB frames and the other for capturing motion or temporal information. The LSTM or RNN in one network models long-range temporal dependencies, allowing the capture of sequential patterns across video sequences. Results from both streams are fused to combine spatial and temporal information, improving AR by integrating appearance and motion cues over time. An example of this approach is the work by Dai, Liu, and Lai [117], which highlights the performance enhancement achieved through this combination.
- *LSTM+3D-CNN-based models*: Hybrid models that combine LSTM networks with 3D-CNNs are highly effective at capturing both spatial features and long-term temporal dependencies in videos. While 3D-CNNs excel at extracting spatiotemporal features by processing video sequences and analysing the spatial information within each frame and its temporal relationships, they often struggle to model long-range temporal dependencies. LSTM effectively addresses this limitation and is specifically designed to capture long-term dependencies in sequential data. This combination enhances the model's ability to recognise complex actions that develop over time. A notable example of this hybrid approach is I3D-LSTM [120].
- *AM+RNN/LSTM hybrid models*: Combining AMs with RNNs/LSTMs enhances VAR by enabling the model to focus on the most relevant parts of the input sequence. While LSTMs excel at learning temporal dependencies, they lack the ability to selectively highlight essential segments, which AMs address by dynamically selecting the most informative frames or periods. This hybrid approach improves efficiency and alleviates LSTMs' challenges with long-term dependencies by guiding them to focus on critical segments, preserving key information over time. Such as the works in [117,138,141], and [140].
- *AM+GCN hybrid models*: Combining AMs and GCNs create a hybrid model that enhances VAR by focusing on key regions while capturing structural dependencies. AMs dynamically highlight the most relevant parts of the video, improving recognition efficiency and accuracy. Meanwhile, GCNs model spatial relationships between body parts or objects as nodes and edges in a graph. By incorporating attention, the model can emphasise critical moments or body parts for AR, while GCNs provide a structured representation of interactions. This combination enables better performance, especially for actions with subtle details that require selective focus and understanding of body part dependencies, as seen in [213,214].
- *Transformer-based hybrid models*: Transformer-based hybrid models combine Transformer (, *e.g.*, ViT) with other approaches like 3D-CNNs or two-stream networks. These models use Transformers to model long-range dependencies while leveraging CNNs or RNNs for low-level feature extraction. Such as the works in [182,191], and [193].

These typical hybrid models discussed each brings unique strengths to the field of VAR. By combining different architectures, these models effectively capture both spatial and temporal features, enhancing the accuracy and robustness of AR systems.

6.4.2. Limitations and future directions

While hybrid models offer significant advantages in VAR, they also present several challenges and limitations that must be addressed to optimise their performance.

1. *Increased model complexity*: Hybrid models often combine multiple architectures, increasing the system's overall complexity. This can result in longer training times, higher computational costs, and more complex hyperparameter tuning. Integrating several models may also make the system more challenging to optimise and deploy effectively.
2. *Computational demands*: Hybrid models, particularly those involving 3D-CNNs or large-scale two-stream networks, can be computationally expensive. This can limit their use in real-time applications or on devices with limited processing power, such as mobile phones or edge devices. Optimising the models for efficiency without compromising performance remains a key challenge.
3. *Data requirements*: Hybrid models typically require large, diverse datasets to perform optimally. This can be a limitation in scenarios where annotated video data is scarce or difficult to obtain. Additionally, training multiple models on such datasets can be time-consuming and resource-intensive.
4. *Deployment and real-time performance*: Deploying hybrid models in real-world applications, particularly those requiring real-time processing, poses significant challenges. The complexity of the models can result in slower inference times, which may not meet the stringent latency requirements of specific applications, such as live surveillance or interactive systems.

As hybrid models continue to evolve in VAR, there are several promising avenues for further research and development.

1. *Incorporating AMs*: One promising avenue is the integration of AMs with hybrid models. Attention-based approaches allow models to focus on relevant parts of the video, improving their ability to differentiate between important and less important features. Combining attention with two-stream, 3D-CNN, or LSTM models could enhance the recognition of complex actions by prioritising critical spatial and temporal cues.
2. *Exploring Transformer-based hybrid models*: Transformers, with their ability to capture long-range dependencies and their parallelisation advantages, could be particularly beneficial when combined with other models to improve both efficiency and accuracy.
3. *Efficient model design*: Given the computational intensity of hybrid models, there is a growing need for more efficient designs. Future research could focus on reducing model complexity while maintaining or improving performance, through techniques such as model pruning, quantisation, or knowledge distillation. These methods could make hybrid models more viable for real-time and resource-constrained applications, such as edge computing or mobile devices.
4. *Cross-domain and multi-modal approaches*: The future of VAR may lie in models that integrate data from multiple sources or modalities. Research into cross-domain models that can transfer knowledge across different types of video data (*e.g.*, from one dataset to another) or combine different sensory inputs (*e.g.*, visual and audio information) could open up new frontiers in AR. Hybrid models could be critical in effectively integrating and processing multi-modal data.

5. *Real-time and edge deployment*: Optimising hybrid models for real-time applications and deployment on edge devices remains a significant challenge. Future directions should focus on developing lightweight hybrid architectures that balance performance with latency requirements. Techniques like network compression, edge-specific optimisations, and adaptive inference could enable hybrid models to perform efficiently in dynamic environments.
6. *Improved generalisation to diverse scenarios*: Future research could focus on improving the generalisation of hybrid models to diverse and unseen scenarios.

Hybrid models can adapt to a wide range of AR tasks. Addressing these limitations and challenges could help achieve more robust and generalisable models.

6.5. Models based on multi-modal learning

In recent years, VAR has witnessed significant advancements, mainly driven by the power of DL techniques. One of the most promising directions in this field is multi-modal learning, a paradigm that integrates different sources of information or modalities to enhance the understanding and recognition of complex actions in videos. DL models, which have demonstrated exceptional capabilities in feature extraction and pattern recognition, form the backbone of these multi-modal approaches.

Multi-modal learning refers to the integration of multiple types of data – such as visual, audio, and motion information – into a single unified model to provide a richer and more comprehensive representation of the scene.

In the context of VAR, these modalities often include the raw video frames (RGB) and additional cues like optical flow, depth information, audio, and skeletal data. Mmulti-modal learning for VAR is crucial because it enables models to leverage complementary information from various sources. It provides a complete understanding of actions and improves recognition performance in challenging environments. DL techniques, such as CNNs, RNNs, and GCNs, play a central role in learning the complex relationships between these various modalities and enabling the model to identify actions with greater accuracy and robustness.

6.5.1. Recent advancements in multi-modal VAR models

In 2021, Wu, Ma, and Li [215] proposed a multi-modal two-stream 3D-CNN for VAR, incorporating depth and pose data alongside RGB videos. They constructed two representations: Depth residual dynamic image sequences (for spatiotemporal motion) and pose estimation map sequences (for body part configurations). The network processes these streams separately and fuses their classification scores. Their model's superior performance is confirmed on SBU Kinect Interaction dataset [216], UTD-MHAD [217], and NTU RGB+D 60 & 120.

In 2022, Chen and Ho [218] proposed a Multi-Modal ViT (MM-ViT), a pure Transformer-based approach that leverages multiple modalities, including RGB frames, motion vectors, and audio waveforms. MM-ViT operates efficiently in the compressed video domain and uses scalable variants to handle large spatiotemporal token sets. Integrating cross-modal AMs captures inter-modal interactions effectively. On datasets UCF101 and Kinetics-600, their model shows its superiority in accuracy and efficiency compared to existing methods.

In 2024, Shaikh et al. [219] proposed a DL model for VAR, called MAiVAR. It combines audio and video modalities using CNNs for enhanced feature extraction. A high-level weight assignment algorithm optimises the interaction between audio and video features, resulting in superior performance on datasets such as UCF51 and Kinetics Sounds.

In 2024, Zhang and Yan [220] proposed a video-language GCN, incorporating fine-grained action parsing knowledge to enhance VAR. By leveraging vision-language models and GCNs, the model integrates coarse-grained and fine-grained knowledge for improved recognition, demonstrating its effectiveness on the Kinetics-TPS [221] dataset.

These studies have demonstrated the potential of integrating various data streams (such as RGB videos, depth maps, pose data, and audio signals) into DL frameworks to achieve more robust and accurate AR. While these methods have contributed valuable insights and innovations, they also share several common challenges and limitations that need to be addressed to further enhance their effectiveness.

1. *Fusion mechanisms*: Although multi-modal methods are powerful, some models rely on late fusion strategies, where the outputs of individual modality-specific networks are combined at a later stage. This may fail to capture intricate relationships and interactions between modalities during the feature extraction phase, potentially leading to suboptimal performance.
2. *Computational complexity and scalability*: DL models, particularly those that handle multi-modal data, tend to be computationally expensive, requiring substantial hardware resources and long training times. Models like the MM-ViT and others involving large-scale Transformers or 3D-CNNs often encounter issues with memory usage and inference speed, which may hinder their application in real-time settings or on resource-constrained devices.
3. *Generalisation and robustness*: Some approaches are trained on specific datasets (*e.g.*, UCF-101, Kinetics, NTU RGB+D) and may struggle to generalise to unseen datasets or more diverse real-world conditions. This issue is compounded by the reliance on large annotated datasets, which are costly and time-consuming to compile.
4. *Model interpretability*: DL models, particularly complex multi-modal networks, often lack interpretability. Understanding how and why a model makes certain predictions can be crucial for applications in safety-critical areas such as healthcare, self-driving, or surveillance.

Some suggestions to improve the performance of these models for VAR are as follows:

1. Early or joint fusion techniques, where different modalities are integrated at lower levels of the network, could help improve the interaction between modalities and allow the model to learn more integrated and coherent features. Additionally, incorporating AMs could further enhance the model's ability to focus on the most relevant aspects of each modality.
2. Adopting lightweight neural network architectures, such as compact Transformers or model pruning techniques, could reduce the computational burden while maintaining high performance. Moreover, using techniques like knowledge distillation could help transfer the learned capabilities of large models to smaller, more efficient versions suitable for deployment in practical applications.
3. One direction for future improvement is to explore methods for model optimisation. Techniques such as knowledge distillation, pruning, or quantisation could reduce the model size and speed up inference times without sacrificing accuracy. This would make it more feasible to deploy such models in real-time applications, such as interactive systems, surveillance, or autonomous vehicles.
4. Introducing data augmentation techniques, transfer learning, or few-shot learning methods could help enhance the model's ability to generalise across diverse scenarios. Moreover, self-supervised learning approaches could reduce the need for large annotated datasets, enabling models to learn from unlabelled or partially labelled data, thus improving generalisation to new environments.

5. Future research could focus on enhancing the interpretability of multi-modal AR models through methods like attention visualisation or saliency mapping, which provide insights into which parts of the input (across modalities) the model is focusing on for decision-making.

By addressing issues related to fusion strategies, computational complexity, generalisation, and interpretability, the field can move towards more efficient, adaptable, and robust models capable of operating in a wider range of real-world environments.

6.5.2. Joint visual-linguistic VAR model

Advancements in multi-modal learning have significantly improved the performance of VAR systems. Multi-modal learning allows for integrating diverse data sources, enabling models to understand complex video actions better. By leveraging complementary modalities, these models enhance AR, especially in challenging environments where individual modalities may not suffice. Despite the benefits of multi-modal integration, the complexity of processing and aligning multiple data streams presents challenges, such as the need for synchronised input data and the increased computational cost.

Joint visual-linguistic VAR model is a specialised type of multi-modal learning focused on visual and linguistic integration. Building on the principles of multi-modal learning, the visual-linguistic VAR model specifically focuses on the fusion of visual and linguistic modalities. While multi-modal models generally seek to integrate diverse types of data, the visual-linguistic VAR model uniquely combines visual cues (such as motion, shape, and spatial relationships) with linguistic information (including dialogue, captions, and environmental sounds converted to text). This specific focus on the interplay between vision and language offers a more tailored solution for VAR, where the understanding of context and subtle cues plays a critical role. In the context of VAR, the visual-linguistic model is particularly beneficial in scenarios where visual ambiguity exists—when the same visual action might have multiple meanings depending on the surrounding linguistic context. By incorporating linguistic data, this model provides a more accurate, context-aware interpretation of actions, enhancing the robustness of AR systems across various domains.

In 2019, Sun et al. [222] proposed VideoBERT, a joint visual-linguistic model for VAR using self-supervised learning. Adapting BERT, they jointly model video (converted to visual tokens) and text transcriptions, learning a bidirectional joint distribution. VideoBERT achieves SOTA performance in video captioning on YouCook II and demonstrated strong zero-shot classification, highlighting the importance of large-scale data and cross-modal learning for VAR.

Some potential applications of VideoBERT include: (1) *Video captioning*: By analysing visual content alongside the corresponding text, VideoBERT can generate automatic captions for videos, making them more accessible to people with hearing impairments and enabling multilingual subtitles. (2) *Human–robot interaction*: VideoBERT allows robots to recognise and understand human actions and gestures, which is crucial for developing robots capable of natural, intuitive interactions in fields such as healthcare, manufacturing, and service industries. (3) *Activity recognition in healthcare*: VideoBERT can track and recognise activities in healthcare settings, such as monitoring patients in hospitals or elderly individuals in nursing homes and assisting caregivers in real-time detection and response to potential health concerns. (4) *Video surveillance and security*: VideoBERT can analyse surveillance footage to classify actions such as loitering, fighting, or carrying suspicious objects, aiding in early threat detection and enhancing public safety. (5) *Sports analysis*: VideoBERT can track player movements, identify actions, and analyse team performance in sports, providing valuable insights for coaches, players, and analysts.

The integration of visual and linguistic data offers several notable advantages.

- *Enhanced contextual understanding*: Linguistic data provides crucial context, improving the model's ability to disambiguate complex actions. For example, linguistic data helps clarify ambiguous actions that may be difficult to interpret from visual data alone, such as distinguishing between different actions that share similar visual cues. This enhanced contextual understanding is particularly useful in complex AR scenarios, where traditional VAR models, relying solely on visual cues, may struggle.
- *Cross-modality learning*: Visual-linguistic integration creates richer feature representations that are essential for tasks like video captioning and AR in noisy environments.
- *Robustness to ambiguity*: Linguistic information complements visual data, reducing ambiguity in scenarios with occlusions or noisy visuals. This robustness to ambiguity makes the integrated model more reliable in real-world settings where occlusions or incomplete data are common.

Despite their advantages, integrated models face notable challenges:

- *High computational overhead*: Multi-modal models are computationally expensive due to the integration of large-scale language and vision components.
- *Dependence on textual quality*: Performance is sensitive to the quality of accompanying textual annotations. Poor or ambiguous textual descriptions can lead to degraded performance.
- *Limited dataset availability*: Scarcity of annotated video-text datasets constrains training and evaluation.
- *Interpretability issues*: Multi-modal models often lack transparency, making it difficult to understand their decision-making processes.

Looking ahead, several areas for future enhancement could improve the effectiveness of integrated visual-linguistic models.

- One key area is the development of more efficient fusion techniques that enable seamless integration of modalities without the significant computational cost.
- Another important direction is the advancement of data alignment techniques, where more robust methods for synchronising and aligning textual and visual data can be explored.
- Furthermore, handling modality imbalance through specialised weighting or AMs would allow the model to better leverage underrepresented modalities.
- Finally, unsupervised or semi-supervised learning could reduce the reliance on annotated datasets, allowing the model to learn from weaker supervision.

6.5.3. Models based on cross-modal attention

Some studies explore the integration of multiple modalities, such as combining visual and audio information for VAR. Cross-modal AMs are employed to align and selectively attend to relevant information from different modalities, improving the recognition performance. These models use attention to weigh the importance of different modalities in videos, such as RGB frames, optical flow, and audio.

Video data naturally comes in several modalities: mainly visual (the actual frames of the video) and auditory (the accompanying sound). Conventional VAR models mostly focus on the visual modality, using techniques such as 3D-CNNs to extract spatiotemporal features. However, such models often overlook the crucial complementary information provided by the audio stream, which can provide important context.

Cross-modal AM based models address this by integrating multiple modalities. A basic overview of how they work is as follows:

1. *Separate feature extraction*: Each modality has a distinct feature extractor, which can be a CNN for visual data and a RNN or CNN for audio data. This results in a set of features for each modality.

Table 9
Comparison between HPE and VAR.

Aspect	HPE	VAR
Goal	Detect and track human body keypoints (joints) in images or videos	Recognise and classify human actions over a sequence of video frames
Output	A set of keypoints (*e.g.*, shoulders, elbows, knees) representing the human skeleton	A predicted action label (*e.g.*, "walking", "jumping", "boxing")
Input Data	Single image or video frame	Full video sequence
Techniques Used	CNN-based models (*e.g.*, OpenPose, HRNet), Transformers, and Graph-based models	CNNs, RNNs (LSTMs/GRUs), GCNs, Transformers, and Two-Stream Networks
Main Challenges	Handling occlusions, viewpoint variations, and inaccurate joint detections	Capturing long-range dependencies, handling motion blur, and recognising fine-grained actions
Applications	HCI, motion capture, sports analysis, healthcare monitoring	Surveillance, sports analytics, self-driving, video content indexing

2. *Cross-modal AM*: The extracted features are then passed through a cross-modal AM. This mechanism allows the model to weigh the importance of different features from each modality. For instance, in a scene where someone is playing a piano, the sound of the piano may have a higher attention weight. Chen and Ho [218] developd and compared three distinct cross-modal AM to improve their Multi-Modal Video Transformer for VAR.
3. *Feature integration*: The weighted features from each modality are then integrated, often through methods like concatenation or fusion, to form a joint representation.
4. *Action classification*: This joint representation is fed into a classifier (like a softmax layer) to predict the action class.

One of the significant advantages of cross-modal AM based models is their ability to focus on the relevant aspects of each modality. For example, in a noisy video, the model can focus on learning visual features more than auditory ones.

However, there is not much work using cross-modal AM for VAR. The following is an interesting exception. To address the issues that compressed VAR severely suffers from the coarse and noisy dynamics and the insufficient fusion of the heterogeneous RGB and motion modalities, in 2022, Li et al. [223] proposed a VAR model, namely Attentive Cross-modal Interaction Network with Motion Enhancement (MEACI-Net). It follows the two-stream architecture, *i.e.*, one for the RGB modality and the other for the motion modality. The interaction between the two streams is strengthened by introducing the Selective Motion Complement (SMC) and Cross-Modality Augment (CMA) modules, where SMC complements the RGB modality with spatiotemporally attentive local motion features and CMA further combines the two modalities with selective feature augmentation. On UCF101, HMDB51, and Kinetics-400, the effectiveness and efficiency of their model are confirmed.

Cross-modal AM based models have shown promise in the task of VAR, by effectively leveraging both visual and audio modalities. However, they do come with certain limitations:

1. *Alignment of modalities*: Cross-modal models need to handle the alignment between different modalities, which can be challenging given that visual data in videos is typically represented as a sequence of frames, while audio data is typically represented as a waveform or spectrogram. Novel methods for aligning the different modalities could be developed. Furthermore, one could employ more complex architectures, like sequence-to-sequence models with attention, to learn the alignment automatically.
2. *Computational complexity*: Cross-modal models tend to be computationally intensive due to the need to process and align multiple modalities. Improvements in model efficiency could be sought, either through the use of more efficient architectures or through the use of hardware acceleration.
3. *Data availability*: These models require large datasets with both visual and auditory information for training. However, such datasets might not always be available or may be difficult to collect and annotate. Techniques such as data augmentation, synthetic data generation, or transfer learning could be used to alleviate the need for large cross-modal datasets.

Future research is needed to address these limitations and improve the effectiveness of these models.

6.6. Summary

This section reviews various deep learning models for VAR, highlighting their pros and cons. 2D-CNNs capture spatial features well but struggle with motion. Multi-stream networks enhance recognition at a higher computational cost, while GCNs use skeletal data for spatial–temporal modelling but are limited by structure. Hybrid models combining CNNs, LSTMs, and Transformers offer a balance of accuracy and efficiency yet face challenges in computational demands and data alignment. Multi-modal learning, which integrates visual, auditory, and textual data, improves accuracy but presents challenges such as data alignment and robustness to noise. Future research should focus on optimising architectures, reducing resource needs, and improving performance under challenging conditions.

7. Human pose estimation VAR models

In the introduction section, we mentioned there are main two categories of VAR methods: RGB-based methods and human pose estimation based methods. In Sections 2–6, we mainly reviewed RGB-based methods. Now in this section, we will discuss human pose estimation based models.

7.1. Basics

Human Pose Estimation (HPE) is a fundamental step in AR, but they are not the same task as VAR. Pose estimation focuses on detecting human keypoints (joints) in images or videos, whereas VAR classifies sequences of movements into predefined actions. Table 9 summarises their differences.

However, HPE methods have been widely applied for VAR, especially in skeleton-based AR. These methods leverage human body keypoints to understand motion patterns, making them effective in many real-world applications. Their usage depends on the specific action type, dataset, and computational constraints.

Using human skeletal representations instead of raw RGB images has several advantages:

- *Robust to background and lighting*: Works well even in varying environmental conditions.
- *Viewpoint invariance*: Skeleton-based methods generalise across different camera angles.

- *Efficient representation*: Requires fewer computational resources compared to RGB-based DL models.
- *Applicable to small datasets*: Unlike deep CNN-based VAR models that need massive datasets, HPE-based models can perform well with fewer samples.

However, HPE-based methods also have limitations:

- *Pose estimation errors*: If pose detection fails (*e.g.*, due to occlusions), AR accuracy drops.
- *Lack of contextual information*: Actions involving object interactions (*e.g.*, playing the piano, eating) may not be well-recognised using only skeletons.

There are mainly two kinds of HPE-based VAR methods:

- *Direct skeleton-based methods*: Once human pose keypoints are extracted, various DL models classify the action: (1) RNNs/LSTMs: Capture temporal dependencies in sequential pose data; (2)GCNs: Model the skeleton as a graph to learn relationships between joints (most popular); and (3) Transformers: Process long-range dependencies in pose sequences.
- *Hybrid methods of pose & RGB data*: Use pose features for motion understanding while RGB frames capture contextual details (*e.g.*, clothing, object interactions). Main methods are: (1) Two-stream models: One stream for skeleton data (pose-based) and another for appearance features (RGB-based). (2) Late fusion approaches: Combine skeleton-based predictions with CNN-based models.

While HPE-based methods are powerful, they are not always the best approach. It is the best for body-centric actions with large movements (*e.g.*, running, dancing, martial arts), but it is less effective for actions involving small movements or object interactions (*e.g.*, cooking, writing).

7.2. Skeleton-based models

Skeleton-based VAR methods typically involves the following key steps:

1. *Skeleton representation*: The human body is represented as a graph of key points (joints) and edges (bones). The skeleton data can be structured using 2D or 3D coordinates of joints (*e.g.*, shoulders, elbows, knees) and kinematic chains that define skeletal connectivity. Typically rely on pose estimation algorithms (*e.g.*, OpenPose [224], AlphaPose [225], and VNect [226]).
2. *Feature Extraction*: Once the skeleton data is obtained, features (*e.g.*, joint positions: (x, y, z) coordinates, velocity and acceleration of joints, and angles between different body parts) are extracted. Temporal information is also captured to track motion over time.
3. *Action Classification*: Various DL models, such as RNNs/LSTMs, CNNs, GCNs, and Transformers, can be used to classify actions based on skeleton data [227,228].

In the rest of this subsection, we will examine some recent studies of skeleton-based methods using various DL technologies. Table 10 compares these DL approaches for skeleton-based methods.

7.2.1. LSTM based model

In 2024, Truong, Hoang, and Le [229] presented a skeleton-based approach for real-world human AR. They used MediaPipe [230] and YOLOv8 (You Only Look Once) [231] pose estimation to extract joint coordinates, preprocessed the data, and trained an LSTM-based model. On the KTH dataset, they achieved 96%–97% accuracy. Models using MediaPipe poses performed slightly better than those using YOLOv8.

7.2.2. GCN based model

In 2019, Zheng, Jing, and Xu [232] proposed a GCN-based model for VAR. They used OpenPose to extract the human skeleton in the video and constructed the spatial and temporal graph of the skeleton. Then, an ST-GCN was used to extract the spatial and temporal features of the human skeleton on consecutive video frames. On the UCF101 dataset, their model obtains a 50.53% top-1 and 81.58% top-5 accuracy.

In 2022, Alsawadi and Rio [233] implemented the BlazePose [234] skeleton topology into the ST-GCN model for AR. Experiments on the UCF101 and HMDB51 datasets show that the model works well. They also proposed an enhanced BlazePose topology, which improved the ST-GCN model's accuracy performance on the UCF101 benchmark dataset by more than 13%.

In 2024, Yang et al. [235] presented an enhanced skeleton-based VAR approach using "Expressive Keypoints" for detailed hands and feet representation, along with a skeleton transformation strategy for importance-weighted down-sampling. They also proposed a plug-and-play instance pooling module, expanding the applicability of the GCN-based method and maintaining computational efficiency for multi-person scenarios. On seven datasets, their model outperformed previous skeleton-based VAR models, demonstrating high discriminative ability and efficiency.

7.2.3. CNN+LSTM based model

In 2022, Malik et al. [236] developed a multi-view interaction-level AR system using 2D skeleton data, aiming for higher accuracy and reduced complexity. OpenPose was used to extract 2D skeleton features, which were directly input to a CNN-LSTM architecture. To reduce complexity, only extracted features were used, eliminating additional feature extraction. The model achieves 94.4% accuracy on MCAD [237] and 91.67% on IXMAS.

In 2023, Sharma and Singh [238] proposed ConvST-LSTM-Net for skeleton-based AR. The model uses a spatiotemporal network consisting of CNNs, ST-LSTMs & fully connected dense layers. They identified key joints in each frame using skeletal tracking algorithms, followed by feature extraction from RGB frame data to enhance model efficiency. ConvST-LSTM-Net outperformed SOTA models on NTU RGB+D 60, UT Kinetics, UP-Fall Detection [239], UCF101, and HMDB51.

In 2025, Le et al. [240] proposed a two-stream model for skeleton-based VAR, integrating LSTM and Depthwise Separable CNNs to capture spatiotemporal motion features from 2D/3D skeleton data (extracted via MediaPipe and MoveNet [241]). A temporal LSTM models sequential joint movements, and a joint-motion module extracts spatial features. Its accuracies on JHMDB (73.31%), Florence-3D Action [242] (97.67%), SBU Kinect Interaction (95.2%), and Penn Action [243] (94.0%) show superior performance. The study highlights robotics, surveillance, and HCI applications, noting challenges in recognising similar fine-grained actions.

7.2.4. Attention+LSTM based model

In 2024, Bharathi et al. [244] presented a DL approach for real-time VAR using pose estimation and an attention-based LSTM. Motivated by gesture ambiguity, occlusions, and clutter challenges, they used skeleton-based pose estimation (OpenPose) and integrated attention within an LSTM for enhanced spatiotemporal learning. On Berkeley MHAD [245], the model achieves 95.94% accuracy, outperforming traditional LSTMs (73.18%) and previous methods. Real-time testing confirmed robustness across camera angles, clothing, and real-world conditions, making it suitable for intelligent surveillance and AI-assisted monitoring.

Table 10
Comparison of DL approaches for skeleton-based methods.

Method	Strength	Weakness
RNN/LSTM	Good for modelling sequential motion; Captures long-term dependencies	Cannot model spatial relations well; Struggles with vanishing gradients for long sequences
CNN	Efficient feature extraction from structured skeleton data; Can learn spatial features	Requires pose encoding as an image-like format; Does not handle temporal dependencies effectively
GCN	Captures skeleton topology naturally; Learns spatial dependencies efficiently	Computational overhead due to graph construction; Limited in handling fine-grained actions
Transformer	Captures long-range dependencies effectively; Self-AM enhances AR	Requires large-scale datasets for training; High computational cost
Hybrid	Achieves SOTA performance by integrating spatial and temporal learning	Computationally expensive; More complex model architecture

Table 11
Comparison of HPE-based models in VAR.

Method	HPE method	Input type	Network
Method in [229]	MediaPipe, YOLOv8	Skeleton data	LSTM
Method in [232]	OpenPose	Skeleton data	ST-GCN
Method in [233]	BlazePose	Skeleton data	ST-GCN
Method in [235]	Expressive Keypoints	Skeleton data	GCN + Skeleton Transformation
Method in [236]	OpenPose	Skeleton data	CNN + LSTM
ConvST-LSTM-Net [238]	Skeletal tracking algorithm	Skeleton data	CNN + LSTM
Method in [240]	MediaPipe and MoveNet	Skeleton data	LSTM and Depthwise Separable CNNs
Method in [244]	OpenPose	Skeleton data	Attention based LSTM
Method in [246]	No details in the original paper	Skeleton data	Two-branch stacked RNN featuring LSTM cells
Method in [248]	Pre-trained pose detector	Skeleton data	Transformer encoder + CNN
Method in [249]	Pose estimation algorithm	Skeleton data	Pre-trained CNN models and ViT
YogNet [250]	Time-distributed CNNs and LSTMs	RGB + skeleton data	Two-stream network (CNN+ LSTM and 3D CNN)
RGBSformer [251]	RGB heatmaps + pose extraction	RGB + skeleton data	Two-stream Transformer
Method in [252]	A pre-trained Faster R-CNN	RGB + skeleton data	Two-stream network (RepVGG-B0 ConvNet and attention-based Bi-LSTM)
Method in [253]	Pose estimation algorithm	RGB + skeleton data	Two-stream network

7.2.5. RNN+LSTM+CNN based model

In 2020, Avola et al. [246] proposed a model that uses 2D skeletons with a two-branch stacked RNN featuring LSTM cells. Unlike 3D skeletons from RGB-D cameras, 2D skeletons are derived from RGB video streams, making the approach suitable for indoor and outdoor environments. Using 3D-CNNs to address missing skeletal data. On KTH, Weizmann [126], UCF Sports, IXMAS, HMDB51, UCF101, Kinetics400, UT-Kinect [247], and NTU RGB+D [188], the model's effectiveness, robustness, and accuracy are confirmed.

7.2.6. Transformer+CNN based model

In 2023, Verma et al. [248] combined skeleton-based and Transformer-based VAR. Skeleton models enhance motion representation, while the Transformer models dynamic convolutions. They used a pre-trained skeleton detector for skeleton annotation from videos. They designed a Transformer Encoder Network that alternates between CNN and Transformer encoders with residual connections to prevent gradient vanishing. Their model achieves a 97.71% AUC-ROC score for some action classes, surpassing SOTA approaches. On UCF101, they achieved 88.5% accuracy.

In 2024, Shi and Liu [249] proposed a VAR model. It integrates the latest pose estimation algorithms, pre-trained CNN models, and ViT to create an efficient system. The process begins with utilising a pose estimation algorithm to accurately extract human pose information from RGB frames. Next, a pre-trained CNN model is employed to perform feature extraction on the extracted pose data. Finally, the ViT model is applied to fuse and classify the extracted features. Experiments on two benchmark datasets, UCF50 (accuracy 83.41%) and UCF101 (accuracy 87.50%), show the effectiveness and efficiency of the proposed framework.

7.3. Hybrid methods of pose & RGB data

In 2022, Yadav et al. [250] proposed YogNet, a multi-person yoga expert system recognising 20 asanas using a two-stream deep spatiotemporal network. One stream uses time-distributed CNNs and LSTMs for keypoint detection and temporal predictions, while the other uses 3D CNNs for spatiotemporal features from RGB videos. Scores are combined via fusion. Using their YAR dataset, YogNet achieves 77.29% accuracy (pose stream), 89.29% (RGB stream), and 96.31% with fusion, recognising and correcting multi-person yoga asanas in real-time.

In 2023, Shi et al. [251] proposed RGBSformer, a two-stream Transformer framework for human AR using RGB and skeleton modalities. Skeleton data and heatmaps are extracted from RGB videos. These heatmaps and RGB frames are input to a Transformer operating at different resolutions. The skeleton stream uses fewer attention layers. Two fusion methods combine stream information. RGBSformer achieves SOTA performance on Kinetics400, NTU RGB+D 60, NTU RGB+D 120, and FineGym99 [192].

In 2023, Huang, Gochoo, and Tan [252] proposed a simple two-stream VAR model. The spatial stream uses RepVGG-B0 ConvNet [254] with cropped RGB features, while the temporal stream uses an attention-based Bi-LSTM to learn posture vectors from pose data from a pre-trained Faster R-CNN [255]. The model achieves SOTA performance on MSR Daily Activity3D [256], with 99.01% precision and 98.91% recall.

In 2024, Rehman et al. [253] proposed a comprehensive VAR system integrating RGB imaging and pose estimation. Their two-stream network processes skeletal and RGB data in parallel, enhanced by pose estimation. A fusion algorithm combines these modalities, enhancing accuracy. Their model achieves 98.94% accuracy on UTD-MHAD, outperforming previous models.

7.4. Summary

Table 11 shows the differences between the above HPE-based models in pose estimation method, input type, and DL networks.

These models present several strengths but also suffer from notable limitations:

1. *Pose estimation errors affect accuracy*: The accuracy of VAR models heavily depends on the precision of the underlying pose

estimation system. Any errors in detecting human joints (due to occlusions, extreme poses, or rapid motion) propagate to the VAR model, reducing recognition accuracy.
2. *Lack of contextual information*: Skeleton-based approaches ignore the surrounding environment and object interactions, making them less effective for actions involving external objects (*e.g.*, playing instruments, cooking, or using tools).
3. *Computational complexity*: GCNs and Transformer-based models for skeleton data offer high accuracy but are computationally expensive. This makes real-time applications (*e.g.*, autonomous driving, surveillance) challenging due to high inference latency.
4. *Limited generalisation across datasets*: Many HPE-based VAR models perform well on specific datasets but fail to generalise across different domains due to variations in camera angles, subject demographics, and movement styles.
5. *Challenges in capturing small movements*:Fine-grained actions that involve subtle movements (*e.g.*, facial expressions, finger movements) are difficult to recognise using only skeletal representations.

Future research should focus on multi-modal data fusion (integrating RGB, depth, optical flow, and skeleton data), optimising computational efficiency with lightweight architectures, enhancing cross-dataset generalisation through self-supervised learning, and refining fine-grained action modelling with improved keypoint detection. Leveraging pre-trained models and adaptive learning strategies could also increase the robustness of HPE-based VAR models, making them more effective in complex scenarios. Despite its limitations, it is still expected to become more intelligent and adaptable with advancements in DL and computer vision, leading to better solutions for automated action analysis and interactive AI systems.

8. Application

Given the rapid advancements in the field of computer vision and DL, the applications of VAR continue to grow and permeate various sectors of industry and daily life. VAR can be applied in a wide range of domains. In this section, we will explore some of them.

8.1. Security and surveillance

VAR technology has become a key component in security and surveillance systems. The task of VAR in security and surveillance is to recognise suspicious or prohibited activities in real time, such as violence.

Some important studies of this kind in recent years are as follows. In 2018, Arunnehru, Chamundeeswari, and Bharathi [257] presented a 3D-CNN with a 3D motion cuboid for action detection and recognition in real-time surveillance videos to prevent crimes. The resulting difference frame stack is called a 3D motion cuboid. On benchmark KTH and Weizmann dataset, their model outperforms previously published results in terms of accuracy.

To detect fights from surveillance cameras in public areas, prisons, and so on in a fast and accurate way, in 2019, Aktı, Tataroğlu, and Ekenel [258] collected surveillance camera fight dataset, and proposed a LSTM-based approach to solve it. It integrates Xception [259] model, BiLSTM, and AM, and uses a pre-trained CNN for feature extraction. Their model improves the SOTA accuracy for fight scene classification.

In the research in surveillance (such as security and sports), in 2020, Mihanpour, Rashti, and Alavi [260] proposed a VAR model using CNN and DB-LSTM networks. The proposed model extracts deep features from raw video frames using a pre-trained CNN model, specifically ResNet-152, followed by learning sequential information about the frames with the DB-LSTM network. The DB-LSTM network uses multiple layers stacked forward and backward to increase depth. On dataset UCF101, the recognition accuracy of their model reaches 95%.

In 2022, Ali, Hussain, and Sadiq [261] proposed a real-time human fighting state recognition model. The YOLOv3 (You Only Look Once) [262] algorithm locates fight scenes, and a centre locator measures inter-person distance. If the distance is below a threshold, deep sorting tracks the individuals. OpenPose and a trained VGG-16 classify the filtered frames as walking, hugging, or fighting. On a custom dataset, the model achieves 95.0% accuracy for walking, 87.4% for hugging, and 90.1% for fighting.

In 2024, Gopalakrishnan et al. [263] applied transfer learning and fine-tuning the I3D and the SlowFast for VAR in surveillance videos. Their studies show that if the dataset is small, then the I3D may be a better choice because it is more likely to generalise new data well; in terms of accuracy, I3D may be a better choice; in terms of any computational resources available, the SlowFast is more computationally expensive than the I3D; and if computational resources are limited, then the I3D may be a better choice.

The similarities and differences between the above papers are as follows:

- Arunnehru, Chamundeeswari, and Bharathi [257] proposed a 3D-CNN architecture with 3D motion cuboids for VAR in surveillance. They apply the model to recognise human actions in the real-world environment of surveillance video.
- Aktı, Tataroğlu, and Ekenel [258] proposed a LSTM-based approach for detecting fights in surveillance videos. Their model integrates Xception, BiLSTM, and AM.
- Mihanpour, Rashti, and Alavi [260] proposed an approach based on DB-LSTM and ResNet for VAR. They evaluated their model on dataset UCF101.
- Hussain and Sadiq [261] proposed an approach based on YOLOv3 for detecting the real-time state of human fighting. They used OpenPose technology and VGG-16 to classify walking, hugging, or fighting.
- Gopalakrishnan et al. [263] selected SOTA (I3D and SlowFast) models as appropriate, given their performance in spatiotemporal AR tasks. They also proposed using transfer learning to address data scarcity as a practical approach.

Each of the above five papers has some limitations as follows:

1. Arunnehru, Chamundeeswari, and Bharathi [257] may suffer motion cuboid limitations. Using 3D motion cuboids to capture spatiotemporal information might have limitations. The model's accuracy can be impacted if the cuboids do not effectively encapsulate the relevant motion or capture too much irrelevant information.
2. Aktı, Tataroğlu, and Ekenel [258] may suffer computational complexity problems; they integrated several models which can be computationally intensive and could limit use in real-time applications or in situations where computational resources are constrained.
3. Mihanpour, Rashti, and Alavi [260] may suffer generalisation problem. Their model trained on specific dataset might not generalise well to different types of environments, lighting conditions, or camera angles.
4. Hussain and Sadiq [261] processed few action such as walking, hugging, and fighting. The accuracy of their model in distinguishing between normal and aggressive behaviours, or in correctly identifying specific actions, can be a challenge.
5. Gopalakrishnan et al. [263] used the SPHAR [264] dataset, a small (90 videos) compiled dataset with only three activity classes: sitting, walking, and running.

8.2. Sport analysis

VAR technology rapidly transforms sport analysis by offering sophisticated tools for coaches, athletes, and teams to enhance their performance, strategy, and training. By automating identifying and classifying actions within sport footage, this technology provides detailed and quantitative data previously difficult or impossible to obtain through conventional manual observation and analysis.

Some important studies of this kind in recent years are as follows. In 2020, Martin et al. [265] proposed a twin spatiotemporal CNN for table tennis stroke recognition. Trained on the TTStroke-21 [266] dataset, their two-stream model uses RGB image sequences and optical flow as input. Each stream comprises three spatiotemporal convolution layers, fused in a fully connected layer. The model achieves 91.4% accuracy on TTStroke-21, compared to I3D's 43.1%.

In 2020, Sanford et al. [267] proposed self-attention models for group activity detection in soccer, using trajectory and video data. Focusing on passes, shots, and receptions in a large-scale Sportlogiq dataset, they found that most events could be detected using either vision or trajectory methods. I3D models trained on full broadcast camera frames outperformed Transformers or GCNs [268] directly modelling player interactions.

The similarities and differences between the above papers are as follows:

- Martin et al. [265] proposed a two-stream-based model to recognise table tennis strokes, the input are RGB images and optical flow. They used the TTStroke-21 dataset to train their model.
- Sanford et al. [267] proposed a self-AM model to learn and extract relevant information from a group of soccer players for activity detection from both trajectory and video data. They used a large scale soccer dataset provided by Sportlogiq.[1]

Each of the above papers has some limitations.

1. Martin et al. [265] may suffer data dependency problem and risk of overfitting. Such studies are often heavily dependent on the quality and quantity of the data used for training the neural networks. If the dataset is not diverse or large enough, the model's ability to generalise to real-world scenarios may be limited. There is a risk of the model overfitting to the specific characteristics of the training data, especially in a highly specialised domain like table tennis. This can limit the model's effectiveness in varied or unexpected scenarios.
2. Sanford et al. [267] may suffer complex dynamics and data variability problem. Soccer is a dynamic and complex sport with many simultaneous actions, making it challenging to accurately detect and interpret group activities. Variations in camera angles, video quality, and playing styles across different matches and teams can impact the model's performance.

8.3. Smart home

With its implementation in smart homes, VAR technology presents many possibilities that advance both convenience and security. An excellent example of its utility is in the homes of elderly individual residents, where AR can be a lifesaver of them. It can monitor their daily activities, identify falls or unusual behaviour, and alert caregivers or medical professionals. This application not only guarantees safety but also provides reassurance to family members.

In 2019, Das et al. [269] proposed the Toyota Smarthome dataset, a large real-world dataset of 16,000 unscripted RGB-D clips depicting 31 daily living activities performed by seniors. They also presented an attention-based VAR model. Using I3D for spatiotemporal feature extraction and 3D skeleton poses for attention weights, they proposed a pose-driven spatiotemporal attention module. LSTM-derived feature vectors are input to the attention module, which computes spatial and temporal attention scores. Their model outperformed previous models on benchmark datasets and the Toyota Smarthome dataset.

In 2024, Su et al. [270] proposed a network for effective fall detection. It consists of a lightweight 3D-CNN and an LSTM. In their model, the 3D-CNN with five layers is presented to avoid the phenomenon of over-fitting. Channel- and spatial-wise attention modules are adopted in each layer to explore the discriminative features further and improve detection performance. The LSTM is presented to extract the long-term spatiotemporal features of 3D tensors. On UCF11, HMDB51, MCFD [271], and URFD [272] datasets, their model shows its superiority in performing fall detection.

The limitations of such applications are as follows:

1. *Variability and unpredictability of human behaviour*: Human activities, especially in a natural setting like a home, can be highly variable and unpredictable. This variability can make it difficult to create models that accurately represent and predict real-world behaviour.
2. *Environmental factors*: The diversity in home environments (different layouts, lighting conditions, furniture, etc.) can significantly impact the performance of models designed to recognise and interpret activities of daily living.
3. *Generalisability*: Models trained in specific smart home environments may not generalise well to other settings due to differences in layout, resident behaviour, or other environmental factors.

8.4. Automotive

VAR in the automotive industry is becoming increasingly important as vehicles is becoming more connected and intelligent. This technology contributes to the advancement of safety features, driver assistance systems, and self-driving capabilities by providing real-time analysis of the vehicle's surroundings, the driver's behaviour, and the actions of other road users.

Driver behaviour is crucial for safety. In 2019, Xing et al. [273] developed a driving-related VAR system using deep CNNs and transfer learning. The system identifies seven driving activities (four normal, three distracting). Tested in real-world partially automated vehicles, the CNN models achieves 91% accuracy as a binary distraction classifier.

The limitations of such applications are as follows:

1. *Data quality and quantity*: The accuracy of DL models is highly dependent on the quality and diversity of the data used for training. If the dataset lacks variety (in terms of different driving conditions, driver behaviours, vehicle types, and so on), the model may not perform well in real-world scenarios.
2. *Environmental variability*: Changes in environmental conditions (lighting, weather, road conditions) can significantly affect the model's performance. The model needs to be robust enough to handle these variations.
3. *Driver variability*: Individual differences in driver behaviour, posture, and interaction with vehicle controls can pose a challenge for accurate AR.

8.5. Robotics

VAR in robotics is an essential component that bridges the gap between dynamic environmental understanding and intelligent robotic response. It equips robots with the ability to detect and interpret human actions, as well as other moving objects, which is essential for

[1] The Sportlogiq is an AI-powered sports analytics company (https://www.sportlogiq.com/).

collaborative robots (cobots), autonomous vehicles, and service robots among others. For example, in the field of Human-Robot Collaboration (HRC), AR is an essential requirement, which allows industrial robots to comprehend human intentions and adaptively execute planning.

In 2020, Xiong et al. [274] proposed a transferable two-stream CNN for VAR in HRC within smart manufacturing. Addressing temporal motion capture (using optical flow) and limited data (using transfer learning), their model comprises spatial and temporal streams, pre-trained on KTH and UCF101 and fine-tuned on engine assembly data. The two-stream CNN achieves 100% accuracy in the target domain, demonstrating robustness. The study confirmed the complementarity of spatial and temporal information and the effectiveness of transfer learning.

In 2021, Li et al. [275] proposed a transfer learning-enabled AR approach to facilitate robot reactive control in HRC assembly. The HRC assembly process consists of two main parts: AR and robotic adaptive control. This process involves the following steps: (1) sensing and pre-processing of data, (2) distilling knowledge and recognising actions from sampled videos, and (3) making decisions and reacting robotically in response to learned semantic knowledge.

These work may provide insightful knowledge to today's industrial HRC research and implementations. The two papers above share a similarity in using transfer learning for VAR. Their differences are as follows:

- Xiong et al. [274] proposed a model using transfer learning and two-stream CNN. They focused on VAR in manufacturing scenarios.
- The work of Li et al. [275] is specifically applied to HRC in assembly tasks, which implies a more industrial and targeted application.

Each of the above papers has some limitations.

1. Xiong et al. [274] used two-stream CNN and optical flow which can be computationally intensive, requiring significant processing power and memory. This might limit the applicability in real-time AR.
2. Li et al. [275] may suffer adaptability and flexibility problem. The dynamic and possibly unpredictable nature of human behaviour in an assembly setting poses a challenge. The model might not adapt well to new, unseen actions or changes in the assembly process.

8.6. Healthcare

VAR technology is increasingly being integrated into the healthcare sector, offering transformative benefits in various applications. For patients undergoing physical therapy, VAR can track and analyse their movements to provide feedback on their progress. It can help in ensuring that exercises are performed correctly and can track improvements over time, aiding therapists in adjusting treatment plans. In surgical training, this technology can be used to analyse and evaluate the movements and techniques of trainees. By comparing their actions with established best practices, it can provide valuable feedback and contribute to skill development.

Some important studies of this kind in recent years are as follows. In 2023, Li and Yeow [276] proposed a VAR model, PoseAction, which combines the strengths of AlphAction [277] and OpenPose. They used OpenPose as an accurate subject detector to recognise the positions of individuals within video streams and employed the Asynchronous Interaction Aggregation network (AIA) from AlphAction to predict the actions of the detected subjects. Their model is trained to recognise 12 common actions in ward environments, such as staggering, chest pain, and falling down, using medical-related video clips from the NTU RGB+D and NTU RGB+D 120 datasets. It achieves an impressive mAP of 98.72% at an IoU of 0.5 on the sub-dataset.

In 2023, Sarapata et al. [278] proposed a model for classifying video and frame-level assessments of motor tasks performed by patients with Parkinson's. The model utilises ST-GCN and consists of 10 graph convolutional layers that operate in both spatial and temporal dimensions. The model is trained to classify patient activities based on estimated body joint locations obtained using OpenPose. It can distinguish 8 MDS-UPDRS items and corresponding 15 activity classes. The model achieves an accuracy of 96.51% in VAR.

Despite the two papers above both use DL model for VAR in healthcare field, they still have some differences:

- Li and Yeow [276] employed AIA, which uses SlowFast with ResNet-50 structure as its baseline model, so it uses 3D CNN to learn spatiotemporal information.
- Sarapata et al. [278] used ST-GCN as backbone for learning spatiotemporal relations.

Each of the above papers has some limitations.

1. In the work of Li and Yeow [276], the training data for PoseAction is currently limited, so the generalisation ability of the model needs to be improved.
2. In the work of Sarapata et al. [278], the model may struggle with accurately assessing subtle differences in motor task, resulting in misclassifications.
3. These two models use a pose estimation tool to identify body joint locations, but the quality of these estimations can suffer in real-world conditions, especially in low-light or occluded environments.

8.7. Human–computer interaction

VAR in HCI represents a dynamic research field where the goal is to create systems that can interpret the specific actions and behaviours of humans captured on video and enable computers to interact with users in a more natural and intuitive way. This involves the analysis of video streams in real time to recognise human gestures, movements, and activities, so that computers can respond appropriately to user commands or needs. For HCI, AR must often occur in real-time, so requiring highly efficient algorithms.

Gesture recognition is a crucial aspect of analysing HCI. Some important studies of this kind in recent years are as follows. In 2021, Cheng and Li [279] proposed a gesture recognition approach using a CNN visual model in HCI, which does not require sensor data. Built on VGG16, this model extracts sequence frames for each gesture, creates a feature vector through convolution, and predicts probabilities for various gesture actions.

In 2024, Rahim et al. [280] proposed a three-stream hybrid model for dynamic hand gesture recognition, integrating RGB pixel and skeleton-based features. The three streams use: (1) a pre-trained ImageNet module with GRU/LSTM, (2) ResNet with GRU/LSTM, and (3) MediaPipe-extracted hand pose key points refined by stacked LSTMs. On a new hand gesture dataset, the model achieved 98.27% precision, 98.35% recall, a 98.29% F-1 score, and 98.35% accuracy.

The similarities and differences between the above papers are as follows. Both papers focus on gesture recognition. The first paper proposed a CNN-based model that can identify similar gestures during interaction without sensors. The second paper presented a multi-modal model using three-stream networks with LSTM for gesture recognition.

Each of the above papers has some limitations.

1. Cheng and Li [279] may suffer gesture variability problem. The model might not effectively handle the wide variability in human gestures, such as differences in speed, scale, or style of gesture, especially if the training data is not sufficiently diverse.

Table 12
Download URL of different datasets.

Dataset	Download URL
ActivityNet [52]	http://activity-net.org/download.html
ARID [164]	https://xuyu0010.github.io/arid.html
AVA [177]	https://research.google.com/ava/download.html
BON [282]	https://ieee-dataport.org/open-access/bon-egocentric-vision-dataset-office-activity-recognition
Charades [210]	https://prior.allenai.org/projects/charades
HAA500 [283]	https://www.cse.ust.hk/haa/
HMDB51 [38]	https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/#dataset
Kinetics [58,284,285]	https://www.deepmind.com/open-source/kinetics
RHM [286]	https://robothouse-dev.herts.ac.uk/datasets/RHM/HAR-1/
Something-Something [60]	https://developer.qualcomm.com/software/ai-datasets/something-something
Sports-1M [287]	https://cs.stanford.edu/people/karpathy/deepvideo/
UCF101 [37]	https://www.crcv.ucf.edu/data/UCF101.php

2. Rahim et al. [280] Used multiple GRU, LSTM, and stacked LSTM, along with pre-trained models, may increase the computational load, posing challenges for real-time applications on devices with limited processing power. While they used a newly created dataset, the model's ability to generalise to other datasets or real-world scenarios is uncertain.

8.8. Education and training

VAR can make learning more interactive and engaging. For instance, it can be used to create immersive educational experiences where students interact with educational content through gestures and movements, making learning more dynamic and enjoyable. In vocational training or workshops, VAR can monitor and guide learners as they practice new skills. For example, it could assess how students use tools and provide suggestions for improvement.

In 2019, Sun et al. [281] proposed the BNU-LCSAD dataset, which is a large-scale classroom student action dataset. And they provided baseline of student VAR results based the database using C3D network.

In 2021, A. and Yin [288] proposed a feature fusion network for student behaviour recognition. Combining a spatial affine transformation network with a CNN extracts detailed features. Spatiotemporal features are fused, and a modified softmax classifier improves recognition. Their model outperformed previous VAR models on HMDB51, UCF101, and a student behaviour dataset.

Identifying teacher actions via video understanding is important for improving teaching quality. Thus, in 2023, Jia et al. [289] developed a teacher-teaching AR dataset and tested mainstream models. They proposed a cross-channel non-local module based on SlowFast to capture long-range spatiotemporal dependencies, achieving significantly better performance.

In 2024, Sharma et al. [290] developed STAR-3D, a student–teacher AR model based on 3D-CNNs. The two-step model first detects "student scenes" and "teacher scents" and then uses a 3D-CNN to capture spatiotemporal features. Trained and validated on their EduNet dataset (20 action categories), STAR-3D achieved 83.5% accuracy.

Despite the four papers above concentrate on employing some form of DL techniques for recognising actions or behaviours in educational settings such as classrooms, they still have some differences:

- Sun et al. [281] proposed the BNU-LCSAD dataset and focused on students' actions. They used C3D for the baseline.
- A. and Yin [288] proposed a fusion network to recognise student behaviour. They use datasets HMDB51 and UCF101 as well as real student behaviour data for their model.
- Jia et al. [289] developed a teacher-teaching actions dataset and focused on recognising teaching actions. They use a SlowFast-based model to capture spatiotemporal features.
- Sharma et al. [290] present a self-developed video dataset for classroom activities. Divided into two categories: student scene and teacher scene.

Each of the above papers has some limitations.

1. Sun et al. [281] may suffer data diversity and representation problem. If the database needs to be more diverse regarding the student population, classroom settings, and types of actions, it may only represent some classroom environments. This lack of diversity can limit the generalisability of any models trained on this database.
2. A. and Yin [288] may suffer dataset diversity and representativeness problem. Student actions and behaviours can vary widely, and datasets HMDB51 and UCF101 do not encompass this full range of behaviours, especially more subtle or less common actions; this can affect the generalisability of the model.
3. Jia et al. [289] may suffer complexity of classroom environments problem. Classrooms are dynamic environments with a lot of simultaneous activities. However, the dataset they used for training and testing their model may not capture the full complexity of these environments, which can limit the model's effectiveness in real-world applications.
4. Sharma et al. [290] proposed the algorithm but has only been tested and validated on their own EduNet dataset and not on other benchmark datasets. And their actions are currently limited to 20 categories. They may need to focus on more accurate and elaborate identification of student–teacher activities with more categories in future work.

8.9. Summary

DL-based VAR has proven transformative across various domains by enabling the understanding and classification of complex human actions from video data. The key applications include security and surveillance, where VAR detects suspicious or anomalous activities in real-time to enhance safety. In sports analytics, it facilitates the analysis of player performance and strategies. In smart homes, VAR enables gesture-based controls and intelligent activity recognition. In the automotive sector, it enhances situational awareness in advanced driver-assistance systems and autonomous vehicles. Robotics benefits from VAR through improved HRC, while healthcare applications include patient monitoring, fall detection, and rehabilitation tracking. HCI leverages VAR to improve gesture-based interfaces, and education and training utilise it to adapt teaching methods based on AR.

Despite challenges such as computational demands and the need for domain-specific datasets, ongoing innovations promise to further expand VAR's applications and societal impact.

9. Datasets

The first task of DL-based VAR is to build a dataset. People can create new datasets for their specific problems, or make use of those available publicly. Such a dataset should meet the characteristics of balanced categories, sufficient data, correct labelling, and task related. This section will brief some benchmark datasets and recent published datasets. Table 12 shows their download URL.

9.1. Benchmark datasets

Benchmark datasets play a crucial role in training and evaluating VAR models.

9.1.1. HMDB51

The HMDB51 dataset was published by Kuehne et al. [38] in 2011, which comprises 51 distinct action classes, each containing at least 101 clips. It is one of the most popular standard benchmark datasets for VAR. These classes of actions can be broadly grouped into five types:

1. *General facial actions*: This includes actions such as smiling, laughing, chewing, and talking.
2. *Facial actions with object manipulation*: These involve actions like smoking, eating, and drinking.
3. *General body movements*: This category includes diverse movements such as cartwheeling, clapping, climbing (stairs), diving, falling, flipping, hand-standing, jumping, pulling up, pushing up, running, sitting (down/up), somersaulting, standing up, turning, walking, and waving.
4. *Body movements with object interaction*: These actions involve interacting with objects, such as brushing hair, catching, drawing a sword, dribbling, golfing, hitting, kicking, picking, pouring, pushing, riding (bike/horse), shooting (ball/bow/gun), swinging a bat, sword exercises, and throwing.
5. *Body movements for human interaction*: This type includes fencing, hugging, kicking someone, kissing, punching, shaking hands, and sword fighting.

HMDB51 is particularly suitable for several aspects of VAR, owing to its unique characteristics:

1. *Variety of actions*: HMDB51 contains 51 action categories, which include a range of human activities from daily life, sports, and facial actions to interactions with objects. This diversity is essential for training models to recognise a wide spectrum of human behaviours.
2. *Real-world videos*: The videos in HMDB51 are from movies, public databases, and online platforms like YouTube. This provides a realistic and challenging setting for AR models, as the videos feature natural human actions in varied contexts and environments.
3. *Limited data size for efficient training*: With about 7000 video clips, HMDB51 is smaller than some other datasets like UCF101. This more manageable size can be advantageous for developing and testing models with limited computational resources.

HMDB51 is a well-recognised dataset in the research community, making it a standard benchmark for evaluating and comparing the performance of AR algorithms. It is downloadable with video source files, and it is widely used.

9.1.2. UCF101

Dataset UCF101 was published by Soomro, Zamir, and Shah [37] in 2012, which contains 101 action classes and 13,320 video clips. The video clips are collected from YouTube and have an average length of 7 s. Its action classes are divided into 5 general types:

1. *Human-object interaction*: This includes actions like Hula Hoop, juggling balls, mixing batter, skateboarding, jumping rope, brushing teeth, and shaving beard.
2. *Body-motion only*: This includes actions like jumping jack, rock climbing, rope climbing, swinging, TaiChi, baby crawling, blowing candles, and body weight squatting.
3. *Human-human interaction*: This includes actions such as military parade, salsa spins, band marching, hair cutting, and head massag.
4. *Playing musical instruments*: This type includes drumming, playing guitar, playing piano, playing tabla, playing violin, playing cello, playing daf, playing dhol, playing flute, and playing sitar.
5. *Sports*: This type includes actions such as baseball pitching, basketball shooting, bench pressing, biking, billiards shots, field hockey penalties, floor gymnastics, frisbee catching, and front crawling.

UCF101 is particularly suitable for several aspects of VAR, including:

1. *Action classification*: With its 101 action classes, UCF101 provides a diverse range of human activities, making it ideal for training and evaluating models that classify specific actions in videos.
2. *Algorithm development and comparison*: UCF101's popularity makes it a standard benchmark for developing and comparing VAR algorithms, often used to demonstrate new method effectiveness.
3. *Generalisation and robustness testing*: The variety of actions and backgrounds in UCF101 videos helps in testing the generalisation and robustness of VAR models across different scenarios and environments.
4. *Real-world scenario simulation*: Since UCF101 includes videos captured in real-world settings, it allows for the simulation and analysis of VAR models in scenarios that are close to real-life applications.
5. *Benchmarking computational efficiency*: UCF101's size and complexity make it a good benchmark for evaluating VAR model computational efficiency during training and inference.

UCF101's versatility and broad range of action categories make it a suitable choice for a wide array of VAR research areas, from fundamental algorithm development to applied research in real-world scenarios. So, it is a widely used benchmark dataset in VAR research and development.

9.1.3. Sports-1M

Published in 2014 by Karpathy et al. [287], the Sports-1M dataset is a large-scale video dataset for sports classification. Comprising 1 million YouTube videos annotated with 487 sports classes (1000–3000 videos per class, about 5% multi-labelled), it includes diverse categories like aquatic, team, winter, ball, combat, and animal sports. It also features fine-grained classes, such as different types of bowling (6), American football (7), and billiards (23).

Sports-1M is particularly suited for various aspects of VAR due to its unique characteristics:

1. *Large scale and diversity*: Sports-1M is one of the largest video datasets available. It contains 1 million YouTube videos spanning 487 sports categories. This large scale ensures various actions and movements, critical for training robust AR models.
2. *Real-world complexity*: The videos in Sports-1M are sourced from YouTube, which means they reflect real-world conditions. This includes variability in lighting, camera angles, background environments, and quality of video, which helps models trained on this dataset to be more adaptable to different real-world scenarios.
3. *Rich annotations*: Each Sports-1M video is labelled with one of 487 sports categories, providing rich information for supervised learning of sport-specific action patterns.

The Sports-1M dataset's large scale, real-world complexity, rich annotations and its role as a challenging benchmark make it particularly suitable for VAR tasks.

9.1.4. ActivityNet

The ActivityNet dataset was published by Heilbron et al. [52] in 2015, which is a large-scale benchmark for human activity understanding in videos. It contains 27,801 videos that belong to 203 activity classes, with an average of 137 untrimmed videos per class. The total video length is 849 h. The activities are naturally occurring in a variety of contexts, with a diverse set of actors, objects, scenes, and viewpoints. The activity classes belong to 7 different top level categories: personal care, eating and drinking, household, caring and helping, working, socialising and leisure, and sports and physical exercises.

ActivityNet is particularly suitable for certain aspects of VAR due to its distinctive features:

1. *Wide range of activities*: ActivityNet includes a broad spectrum of human activities, categorised into a diverse set of classes. This variety is crucial for developing algorithms that can recognise and classify a wide range of human actions, from simple to complex.
2. *Long duration videos*: Unlike short-clip datasets, ActivityNet features longer videos, vital for studying activities unfolding over extended periods, providing a more realistic action scenario.
3. *Temporal annotation for activities*: The dataset provides temporal annotations, delineating action start and end times, essential for models that recognise actions and understand their temporal dynamics.

In summary, the ActivityNet dataset's diverse and comprehensive activity classes, focus on longer-duration videos with temporal annotations make it particularly suitable for advancing VAR, especially in areas requiring understanding of complex and temporally extended human activities.

9.1.5. Charades

The Charades dataset was published by Sigurdsson et al. [210] in 2016, which is a large-scale dataset with a focus on common household activities collected using the "Hollywood in Homes" approach, where participants were given scripts and asked to act them out.

The first publicly released version of the Charades dataset contains 9848 videos of daily activities with an average length of 30 s. The dataset contains 157 action classes. It includes actions such as holding a laptop, taking a dish, watching TV, washing a window, putting shoes somewhere, throwing a pillow, ans so on. This dataset could be applied in developing action representations, learning object states, human object interactions, modelling context, object detection in videos, and video captioning.

Charades is particularly suitable for certain aspects of VAR due to its distinct features:

1. *Focus on daily life activities*: Charades mainly feature videos of everyday activities, often performed in home settings. This focus on everyday, realistic actions makes it especially relevant for applications in smart home technology, elder care, and HCI.
2. *Scripted actions by amateur actors*: Unlike datasets compiled from professional video sources, Charades consists of videos where amateur actors perform scripted actions. This approach results in more natural and varied renditions of activities, which is beneficial for developing algorithms that can recognise actions in their less stylised, more authentic forms.
3. *Detailed temporal annotations*: Charades provides temporal annotations for each action within the videos. These annotations are crucial for developing models capable of understanding the timing and sequence of activities, a key challenge in AR and temporal localisation.

The Charades dataset focuses on everyday human activities. Its distinctive data collection methodology, complicated temporal annotations, and applicability for multi-task learning and interaction-based AR render it valuable for VAR research and development.

9.1.6. Kinetics

The Kinetics datasets focus on human actions (rather than activities or events). There are several versions of Kinetics, including the Kinetics-400 dataset (published by Kay et al. [58] in 2017), the Kinetics-600 dataset (published by Carreira et al. [284] in 2018), and the Kinetics-700 dataset (published by Carreira et al. [285] in 2019), each with different numbers of action classes and video clips. The clips are from YouTube videos.

Kinetics-400 contains 400 human action classes, with at least 400 video clips for each action. Each clip lasts around 10 s and is taken from a different YouTube video. The list of action classes mainly covers:

1. *Person (singular) actions*, *e.g.*, drawing, drinking, laughing, and pumping fist;
2. *Person-person actions*, *e.g.*, hugging, kissing, and shaking hands; and
3. *Person-object actions*, *e.g.*, opening present, mowing lawn, and washing dishes.

Some actions are fine-grained and require temporal reasoning (*e.g.*, different swimming styles), while others require object emphasis (*e.g.*, playing different wind instruments).

Kinetics-600 contains 600 action classes, each with at least 600 video clips. And the Kinetics-700 dataset contains 700 action classes, each with at least 600 video clips.

The Kinetics family is particularly suitable for various aspects of VAR due to its distinctive features:

1. *Large volume of data*: With thousands of video clips across its different versions, the dataset provides a substantial amount of data. This large volume is crucial for training DL models that require extensive data to achieve high accuracy and robustness.
2. *Benchmarking and comparative analysis*: The Kinetics family is a widely recognised standard benchmark for evaluating AR model performance, facilitating comparison with existing work.
3. *Continuous dataset expansion*: Kinetics is expanded and updated over time (*e.g.*, from Kinetics-400 to Kinetics-700), reflecting a continuous effort to include more diverse and comprehensive action classes.

Kinetics has extensive range of action classes, large volume of high-quality, so it becomes a benchmarking tool. It particularly suitable for advancing VAR, especially in applications that require understanding a wide array of human activities in realistic settings.

9.1.7. Something-Something

The Something-Something dataset was published by Goyal et al. [60] in 2017, which contains 108,499 short video clips across 174 classes. The videos show objects and actions performed on them. Labels are in textual form and represent detailed information about the objects and actions as well as other relevant information.

Something-Something is particularly suitable for certain aspects of VAR due to its unique features:

1. *Focus on object interaction*: Unlike some datasets focusing on body movements, Something-Something emphasises human-object interactions. This is crucial for developing algorithms that understand human manipulation and interaction with objects, a key aspect of real-world applications.
2. *Crowdsourced, user-generated content*: The dataset's crowdsourced videos exhibit high variability in camera angles, lighting, and backgrounds, ensuring models trained on it are robust and adaptable to diverse real-world settings.
3. *Fine-grained AR*: The dataset is suited for fine-grained AR tasks, where subtle differences between actions (often involving similar objects or gestures) need to be discerned. This level of granularity is challenging and crucial for advanced recognition systems.

Something-Something emphases object interactions, it has large variety of simple actions, crowd-sourced content, and focus on temporal dynamics. Therefore, it particularly suitable for advancing VAR, especially in applications that require an accurate understanding of human-object interactions and fine-grained action differentiation.

9.1.8. AVA

Published in 2018 by Gu et al. [177], the AVA dataset contains spatiotemporally localised Atomic Visual Actions from YouTube. It densely annotates 80 actions (14 pose, 49 person-object, and 17 person-person interaction classes) in 437 15 min clips, including actions like writing, drinking, and throwing. People and their actions are localised with bounding boxes.

AVA is useful for developing models that can detect human actions due to its unique characteristics:

1. *Focus on atomic actions*: AVA emphasises *atomic* actions, which are basic, fundamental human actions that are not further decomposable. This focus is crucial for understanding the building blocks of more complex activities and interactions, making it ideal for fine-grained AR tasks.
2. *Rich annotations*: AVA is richly annotated, capturing a diverse range of actions. It labels human actions at a specific point in time, in a manner that is more granular than many other datasets. This allows for a detailed understanding of human activities and interactions in various contexts.

The AVA dataset's emphasis on atomic actions, its detailed spatial and temporal annotations, realistic scenarios, and large scale make it particularly suitable for advancing AR, especially in areas requiring fine-grained of human actions in complex environments.

9.1.9. Jester

The Jester dataset was published by Materzynska et al. [199] in 2019, which is a large video dataset focused on human hand gestures. The main objective of this dataset is to facilitate the development and benchmarking of ML models for hand gesture recognition. It includes 148,092 short clips of videos of 3 s length, divided into 27 action classes. The gestures include such drumming fingers, rolling hand away, shaking hand, stop sign, swiping down, swiping up, thumb down, turning hand clockwise, and so on.

Jester is suitable for certain aspects of VAR due to its specific features:

1. *Specialisation in hand gesture recognition*: The dataset is dedicated to hand gesture recognition, featuring a wide range of gestures. This specialisation makes it ideal for developing and refining models that focus on recognising and interpreting hand movements, a crucial aspect in many HCI scenarios.
2. *Diversity of participants and environments*: The dataset includes videos recorded by a variety of participants in different environments. This diversity ensures that models trained on this dataset can handle variations in backgrounds, lighting conditions, and individual hand shapes and sizes.
3. *Fine-grained AR*: The dataset is well-suited for fine-grained AR tasks requiring discernment of subtle gesture differences, crucial for precision-dependent applications.

The Jester dataset's focus on hand gestures, and its relevance to real-world applications make it particularly suitable for advancing VAR, especially in the domain of hand gesture recognition and fine-grained action analysis.

9.2. Recent datasets

New datasets are fundamental to advancing VAR. These crucial resources enable training and refining algorithms for diverse scenarios, from simple gestures to complex interactions, for robust and accurate recognition. Recent VAR datasets are reviewed below.

9.2.1. ARID

The dataset of Action Recognition in the Dark (ARID) was published by Xu et al. [164] in 2021, which is considered the first dataset focused on this kind of AR in videos. It consists of over 3780 video clips with 11 action categories. The list of action classes can be categorised into two types: singular person actions, which includes jumping, running, turning, walking, and waving; and person actions with objects, which includes drinking, picking, pouring, pushing, sitting, and standing. The training and testing sets are partitioned by splitting the clip groups, with 70% of the groups in the training partition, and the remaining 30% of the groups in the testing partition.

Such a dataset focused on dark or low-light video conditions is crucial for VAR and is likely to gain popularity for several reasons:

1. *Challenging real-world conditions*: Many real-world scenarios (*e.g.*, nighttime surveillance, emergency response operations, and nocturnal wildlife observation) occur in low-light or dark conditions. A dataset capturing these conditions is crucial for developing systems operating effectively in such environments.
2. *Enhanced robustness of recognition systems*: Training models on dark video datasets helps in building more robust AR systems. These systems become capable of performing accurately not only in well-lit conditions but also in scenarios with poor lighting, thus enhancing their usability.
3. *Advances in low-light vision technologies*: Such datasets can drive innovation in low-light imaging and vision technologies. They can lead to the development of advanced techniques for image enhancement, noise reduction, and feature extraction in dark conditions.
4. *Critical for security and surveillance applications*: Surveillance systems often need to operate effectively during nighttime or in poorly lit areas. A dataset with dark videos is essential for developing and testing systems tailored for these applications.

Given the importance of low-light performance, the ARID dataset, focusing on dark videos, is crucial and likely to be popular in computer vision, security, and autonomous systems, attracting further research into low-light VAR.

9.2.2. HAA500

Published in 2021 by Chung et al. [283], the HAA500 (Human-centric Atomic Action) dataset has fine-grained atomic action classes, each containing a single action type.

- Containing 501 atomic action classes (212 sports, 51 musical instruments, 82 games/hobbies, and 155 daily actions), HAA500 videos capture essential action elements without irrelevant frames.
- It includes 10,000 clips, and each class contains 20 clips, with an average length of 2.12 s. Each clip is annotated with meta-information which contains the following two fields: the number of dominant people in the video and the camera movement.

HAA500 is particularly important for AR and is poised to gain popularity for the following reasons:

1. *Granularity and precision*: Atomic actions are fundamental units of human behaviour. Focusing on these basic elements allows for a more accurate human AR, which is essential for developing sophisticated models that recognise complex behaviours.
2. *Building blocks for complex actions*: Just as words are formed from phonemes, complex actions are composed of atomic actions. A dataset that captures these atomic actions can be a foundational building block for recognising and analysing more complicated activities. This is particularly important in surveillance, healthcare (*e.g.*, monitoring patients), and HCI.

The importance and potential popularity of HAA500 is due to its ability to provide detailed, granular insights into human behaviour, which is essential for developing more accurate, versatile, and interpretable AI models in AR.

9.2.3. CDAD

Published in 2022 by Xiang et al. [291], CDAD (Common Daily Action Dataset) focuses on everyday daily actions, comprising 57,824 video clips of 23 well-defined actions with rich annotations. Diverse positive and hard negative samples (with minor similarities) were collected for each action. Spatial and temporal characteristics were defined for each action and its annotations. Target and negative actions were collected within the same group (same person/background), and various target actions were collected in the same scene, focusing on the actions themselves.

Such a dataset focusing on common actions in daily life is crucial for the advancement of VAR technology and is likely to gain widespread popularity due to:

1. *Relevance to everyday life*: By concentrating on actions that occur regularly in daily life, such datasets directly address the most frequent and pertinent human behaviours. This relevance makes the technology developed using these datasets immediately applicable to a wide range of real-world scenarios, from smart home automation to elderly care.
2. *Enhanced accuracy in practical applications*: AI models trained on datasets of common daily actions are more likely to recognise and interpret these actions accurately in practical settings. This is crucial for applications like surveillance, healthcare patient monitoring, or consumer electronics user interaction.
3. *Broad applicability*: Common daily actions are universal across various cultures and environments, making the datasets applicable globally. This universality is particularly important for developing technologies that are meant for a global market.
4. *Facilitating assistive technologies*: For assistive technologies, such as those used by individuals with disabilities or the elderly, recognising everyday actions is essential. These datasets can help create smarter assistive devices that can better understand and predict the needs of their users.

The importance and potential popularity of the CDAD dataset is rooted in their direct applicability to a wide range of practical, everyday scenarios. These focusing on common actions in daily life datasets are instrumental in developing AI technologies that are accurate, relevant, and capable of enhancing various aspects of daily life, from personal convenience to safety and healthcare.

9.2.4. BON

The BON dataset was published by Tadesse et al. [282] in 2022. It is a First-Person Vision (FPV) dataset of office activities collected in Barcelona (Spain), Oxford (UK) and Nairobi (Kenya) using a GoPro Hero wearable camera. It is a large and publicly available FPV dataset of office activities, which contains 18 common office activities that can be categorised into person-to-person interactions, person-to-object interactions, and proprioceptive activity. Annotation is provided for each segment of video with 5-s duration. BON contains 25 subjects and 2639 total segments.

Such an FPV dataset is particularly important for AR. It is poised to gain popularity for several reasons:

1. *Unique perspective*: FPV provides a unique viewpoint, capturing what the person wearing the camera sees. This perspective is crucial for understanding and analysing human interactions and activities from the individual's viewpoint, which differs from third-person observations.
2. *Applications in wearable technology*: With the rise of wearable cameras and augmented reality devices, FPV datasets are vital for developing applications in these areas. These include assistive technologies for the visually impaired, personal memory aids, and interactive AR systems.
3. *Enhanced interaction understanding*: FPV datasets allow for a better understanding of how people interact with their environment and with other individuals, providing insights that are not easily captured through third-person views.

Given these diverse applications and the unique insights provided by FPV, the BON dataset is not only important but also expected to become increasingly popular in various domains, including technology, social sciences, and healthcare.

9.2.5. RHM

The dataset of Robot House Multi-View (RHM) was published by Abadi et al. [286] in 2023, which is a multi-view RGB benchmark dataset for AR, which contains four views: front, back, ceiling, and robot-views. The list of its activities is as follows: walking, sitting down, standing up, lifting objects, carrying objects, drinking, stairs climbing up, stairs climbing down, stretching, putting objects down, reaching, opening can, closing can, and cleaning.

Such a multi-view dataset for VAR is important and likely to gain popularity for several reasons:

1. *Comprehensive understanding of human activities*: Multi-view datasets provide different angles and perspectives of human activities, which are crucial for creating more accurate and robust recognition models. This is particularly important in real-world scenarios, where a single viewpoint might not capture all relevant aspects of an activity.
2. *Facilitates complex AR*: Some activities are complex and involve movements that may not be fully observable from a single viewpoint. A multi-view dataset allows for a better understanding of these complex activities, leading to more accurate AR.
3. *Supports a variety of applications*: Such a multi-view dataset can more effectively cater to the needs of the diverse applications such as surveillance, sports analysis, healthcare monitoring, and HCI.

Given these advantages, the RHM dataset is likely to be widely used and popular among researchers and practitioners in fields like computer vision and robotics.

9.2.6. TikTokActions

The TikTokActions dataset was introduce by Qian et al. [292] in 2024. It is a large-scale video dataset designed specifically for advancing AR research. Derived from TikTok, aims to encapsulate a wide range of modern human behaviours. The dataset spans 386 unique action categories, comprising 283,582 unique video clips.

The TikTokActions dataset may become popular in future in AR field due to the following main reasons:

1. *Diverse applications*: Its extensive range of action categories and real-world relevance make it an excellent resource for developing models applicable to social media analysis, behaviour monitoring, and HCI.
2. *Multi-modal research opportunities*: The dataset's roots in TikTok videos, often accompanied by captions and audio, offer avenues for integrating video, text, and audio data, making it ideal for cutting-edge multi-modal AI research.
3. *Support for foundation models*: The dataset has already been used to pretrain previous models like VideoMAEv2 [293], demonstrating its ability to enhance performance on standard benchmarks. As foundation models grow in prominence, datasets like TikTokActions that support robust pretraining will become increasingly valuable.

Table 13
Comparison of different datasets (the bold ones are the most commonly used).

Dataset	Action class number	Clips of an action class	Total video clips	Length of a clip	Released year
ActivityNet [52]	203	Avg. 137	27,801	5–10 min	2015
ARID [164]	11	≥110	3780	≥1.2 s	2021
AVA [177]	80	–	437	15 min	2018
BON [282]	18	69–237	2639	5 s	2022
CDAD [291]	23	≥1577	57,824	8–21 s	2022
Charades [210]	157	–	9848	Avg. 30 s	2016
HAA500 [283]	500	20	10,000	Avg. 2.12 s	2021
HMDB51 [38]	51	≥101	6766	≥1 second (s)	2011
Jester [199]	27	–	148,092	3 s	2019
Kinetics-400 [58]	400	400–1150	306,245	around 10 s	2017
Kinetics-600 [284]	600	600–1150	495,547	10 s	2018
Kinetics-700 [285]	700	600–1150	650,317	10 s	2019
RHM [286]	14	407–700	26,804	1-5 s	2023
Something-Something [60]	174	Avg. 620	108,499	2-6 s	2017
Sports-1M [287]	487	1000–3000	1,000,000	Avg. 336 s	2014
TikTokActions [292]	386	325–938	283,582	3–10 s	2024
UCF101 [37]	101	≥100	13,320	Avg. 7.21 s	2012

Table 14
Different datasets used in the papers we surveyed.

Dataset	Used in
ActivityNet	[51,83]
AVA	[99,176]
Charades	[99,207]
HMDB51	[34,47,50,53–55,57,59,62,64,75–77,82,96,98,100–103,112,114,119,121–123,129,131,132,137,143,151,158,159,161,162,165–168,173,174,178–180,193,197,201,202,206,223,233,235,238,246]
Jester	[82]
Kinetics	[57,76,96,99,122,166,171,173,178,179,182,191,202,218,223]
Something-Something	[59,82,166,171,198,206,218]
Sports-1M	[75,96,155]
UCF101	[34,47,50,53–55,57,59,62,64,74–77,82,95,96,98,100–103,112,114,119–123,129,131,132,137,143,151,155,158,159,162,165,166,168,173,174,178–180,193,197,201,202,206,218,223,232,233,235,238,246,248,249]

The TikTokActions dataset has the potential to contribute meaningfully to the field of AR by offering a rich and diverse resource for both academic research and practical applications, while supporting advancements in AI-based video analysis and VAR.

9.3. Summary

These datasets cover various actions and scenarios, from daily activities to sports competitions, providing researchers with a wide selection of datasets for researching and comparing VAR algorithms. Table 13 shows the comparison of these datasets mainly in the number of action classes, number of clips for an action class, number of total video clips, length of a clip, and released or published year.

Table 14 shows dataset usage in the reviewed models. Standard benchmark datasets enable direct model comparison. Limited computational power leads some to use smaller datasets like HMDB51 and UCF101. Institutions with substantial computational power provide pre-trained parameters on large datasets. While recently published datasets are not yet widely used, these distinctive video action datasets may become more prominent over time.

Table 15 shows papers using multiple datasets. Datasets vary significantly; evaluating on multiple datasets demonstrates general applicability and tests robustness. Good performance across diverse datasets suggests robustness to data variations. Multiple dataset evaluation is crucial for generalisability, robustness, and thorough model evaluation.

10. Evaluation

In this section, we will summarise key metrics for evaluating DL-based VAR models and compare some models reviewed in this paper against these metrics.

10.1. Evaluation metrics

VAR models can be evaluated using various metrics, which help measure how well they can classify and recognise different actions within video sequences. In this subsection, we will cover the main metrics for evaluating VAR models.

10.1.1. Accuracy

Accuracy is the most straightforward and commonly used metric. It is the proportion of correctly identified actions out of the total number of actions, *i.e.*,

$$Accuracy = \frac{n_{correct}}{n_{total}}, \tag{1}$$

where $n_{correct}$ is the number of correctly classified samples in the predicted samples, and n_{total} is the number of all samples. This simplicity makes it an appealing choice for evaluating the performance of AR models.

10.1.2. Top-1 accuracy and top-5 accuracy

Top-k accuracy is typically used in model evaluations where the model's prediction is considered correct if the true label is among the top k (1 or 5) predicted classes.

Top-1 accuracy is the proportion of times the model's highest probability prediction (*i.e.*, its top choice) is correct. It is particularly useful for evaluating models in tasks where only the most likely prediction is of interest; in many applications, a single decision or prediction is required; in such scenarios, one is mainly concerned with whether the model's best guess is correct or not. Its calculation formula is:

$$Accuracy_{\text{top-1}} = \frac{n_{\text{correct top-1}}}{n_{total}}, \tag{2}$$

where $n_{\text{correct top-1}}$ is the count of instances where the model's highest probability prediction is exactly the same as the true label of the

Table 15
One model using multiple datasets for evaluation.

Model	Dataset	Model	Dataset
Two-stream CNN [34]	UCF101, HMDB51	HAN [143]	UCF101, HMDB51
Two-stream Network Fusion [47]	UCF101, HMDB51	Method in [151]	UCF101, HMDB51
TSN [50]	UCF101, HMDB51	STAN [155]	UCF101, Sports-1M, THUMOS14, CCV
Hidden Two-stream CNNs [53]	UCF101, HMDB51	STA-CNN [158]	UCF101, HMDB51
LSF CNN [54]	UCF101, HMDB51	STA-TSN [159]	UCF101, HMDB51, THUMOS14, JHMDB
BS-2SCN [55]	UCF101, HMDB51	Method in [161]	HMDB51, IXMAS
MLENet [59]	UCF101, HMDB51, Something-Something	Method in [162]	UCF101, HMDB51
2D-3D ResNet [62]	UCF101, HMDB51	SP-LTN [165]	UCF101, HMDB51
STILT [57]	UCF101, HMDB51, Kinetics	MS-KFS [166]	UCF101, HMDB51, Kinetics, Something-Something
Two-stream with PCANet [63]	UCF Sports, KTH	HM-AN [167]	UCF Sports, HMDB51
Spatiotemporal Pyramid Network [64]	UCF101, HMDB51	TAMNet [168]	UCF101, HMDB51
T3D [76]	UCF101, HMDB51, Kinetics	HA-SSD [169]	Oxford hand, DSSL
Res3D [75]	UCF101, HMDB51, Sports-1M, THUMOS14	TDN [171]	Something-Something, Kinetics
I3D [77]	UCF101, HMDB51	S3D RANs [173]	UCF101, HMDB51, Kinetics
R(2+1)D [96]	UCF101, HMDB51, Sports-1M, Kinetics	Method in [174]	UCF101, HMDB51, UCF50
T-C3D [78]	UCF101, HMDB51	CATNet [178]	UCF101, HMDB51, Kinetics
SlowFast [99]	Charades, Kinetics, AVA	Method in [179]	UCF101, HMDB51, Kinetics
Asymmetric 3D-CNN [98]	UCF101, HMDB51	Method in [180]	UCF101, HMDB51, YouTube Actions
D3DNet [82]	UCF101, HMDB51, Something-Something, Jester	Method in [182]	NTU RGB+D 60, NTU RGB+D 120, Kinetics Skeleton 400
STDA [100]	UCF101, HMDB51	TP-ViT [191]	Kinetics, FineGym
Two-stream 3D Dilated [101]	UCF101, HMDB51	ViT-ReT [193]	UCF101, HMDB51, UCF50, YouTube action
FSAN [102]	UCF101, HMDB51	Method in [197]	UCF101, HMDB51
DAMR_3DNet [103]	UCF101, HMDB51, CSL	Method in [201]	UCF101, HMDB51
Method in [112]	UCF101, HMDB51, YouTube 11 Actions [134]	STSF [202]	UCF101, HMDB51, Kinetics
Method in [114]	UCF101, HMDB51	STIP-GCN [206]	UCF101, HMDB51, Something-Something V2
Method in [117]	UCF11, UCF Sports, JHMDB	STIGPN	CAD-120, Something-Else, Charades
TS-LSTM+temporal-inception [119]	UCF101, HMDB51	Method in [211]	NTU RGB+D 60, NTU RGB+D 120
ST-D LSTM [121]	UCF101, HMDB51	MEACI-Net [223]	UCF101, HMDB51, Kinetics-400
ResLNet [122]	UCF101, HMDB51, Kinetics	Method in [233]	UCF101, HMDB51
Method in [123]	UCF101, HMDB51	Method in [235]	UCF101, HMDB51, NTU RGB+D 60, NTU RGB+D 120, Kinetics Skeleton 400, PKU-MMD [294], N-UCLA [295]
Method in [124]	KTH, IXMAS, WVU, WEIZMANN, GBA [124]	Method in [236]	MCAD, IXMAS
Method in [129]	UCF101, HMDB51	ConvST-LSTM-Net [238]	UCF101, HMDB51, NTU RGB+D 60, UT Kinetics, UP-Fall Detection
Method in [130]	Hockey Fights [296], Movies [296], and Violent Flows [297]	Method in [240]	JHMDB, Florence-3D Action, SBU Kinect Interaction, Penn Action
DC-BiLSTM [131]	UCF101, HMDB51	Method in [246]	UCF101, HMDB51, Kinetics400, KTH, Weizmann, UCF Sports, IXMAS, UT-Kinect, NTU RGB+D
TDS-BiLSTM [132]	UCF101, HMDB51	Method in [249]	UCF50, UCF101
Method in [133]	UCF11, UCF Sports, JHMDB	RGBSformer [251]	Kinetics400, NTU RGB+D 60, NTU RGB+D 120, FineGym99
STDAN [137]	UCF101, HMDB51, UCF11		
Method in [138]	UCF11, UCF Sports, JHMDB		

data point. This simplicity makes it easy to understand and interpret, allowing for clear comparisons between different models.

Top-5 accuracy considers a prediction correct if the true label is among the model's top 5 predictions. In datasets with fine-grained categories (where categories are very similar), it is challenging for a model to pinpoint the exact category, Top-5 accuracy offers a more realistic measure of performance. The formula for Top-5 accuracy is:

$$Accuracy_{\text{top-5}} = \frac{n_{\text{correct top-5}}}{n_{total}}, \tag{3}$$

where $n_{\text{correct top-5}}$ is the count of instances where the model's top five most probable predictions include the true label of the data point.

10.1.3. Precision

Precision is the proportion of true positive results (relevant instances that are correctly identified as such) among the total instances identified as relevant (true positives and false positives). This is mathematically expressed as:

$$Precision = \frac{n_{TP}}{n_{TP} + n_{FP}}, \tag{4}$$

where n_{TP} is the number of correctly predicted true positives, and n_{FP} is the number of false positives.

Precision is particularly important when the cost of making a false positive prediction (*i.e.*, predicting an action incorrectly) is high. For example, if a surveillance system incorrectly identifies a benign action as a threat, it could lead to unnecessary panic or even mobilisation of resources. Hence, in such cases, it is essential that the model has a high precision to minimise false alarms.

10.1.4. Recall

Recall (also called Sensitivity) is the proportion of actual positives that are correctly identified as such. In other words, it is the proportion of relevant instances that were retrieved successfully. Recall is calculated as follows:

$$Recall = \frac{n_{TP}}{n_{TP} + n_{FN}}, \tag{5}$$

where n_{TP} is the number of correctly predicted true positives, and n_{FN} is the number of false negatives.

10.1.5. F1 score

F1 score is the harmonic mean of precision and recall. F1 score offers a balance between these two metrics, making it more robust in cases where either precision or recall is particularly low. A higher F1

Table 16
Different evaluation metrics usage in various models.

Evaluation metric	Used in
Accuracy	[34,47,50,53–55,57,59,62–64,75–78,98,100–103,112,114,117,119–124,129,130,132,133,137,138,140,141,143, 155,158,159,161,165–168,170,173,174,179,180,193,197,201,202,206,211,215,219,220,229,233,235,236,238,240, 244,246,248–253]
Confusion matrix	[34,57,62,63,103,112,117,123,124,131,133,138,140,141,159,162,167,169,174,202,205,211,215,229,233,236,240, 244,249–253]
Top-1 accuracy	[59,74,82,95,96,99,131,141,151,162,163,171,173,178,179,182,191,198,205,207,218,222,223,232,235,249,251]
Top-5 accuracy	[59,64,74,82,95,96,99,163,171,178,179,182,191,198,205,207,218,222,232,249,251]
Recall	[34,62,99,124,141,172,180,229,236,238,240,244,250,252,253]
Precision	[62,124,141,172,180,193,229,236,238,240,244,250,252,253]
F1 score	[62,124,141,180,229,236,238,240,244,250,253]
mAP	[95,99,151,155,159,167,169,176,207]
IoU	[75,151,155]
AUC-ROC	[74,248]

score indicates better overall performance of the model in accurately recognising actions in videos. F1 score is calculated as follows:

$$F1 = \frac{2 \times (Precision \times Recall)}{Precision + Recall}. \tag{6}$$

F1 score provides a more balanced measure of a model's performance, especially in scenarios where there are imbalanced classes. Some action classes are significantly more common in many real-world datasets than others. For instance, in a dataset of sports videos, certain actions like running occur more frequently than actions like diving. In such cases, accuracy alone can be misleading, as a model could achieve high accuracy by predicting the majority class most of the time. By contrast, F1 score gives a more honest depiction of a model's performance across all classes.

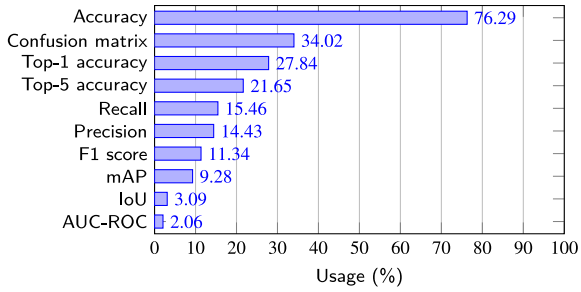

Fig. 8. Comparison of evaluation metrics usage in various models.

10.1.6. Mean average precision

This metric is beneficial when dealing with datasets with imbalanced class distribution. Average Precision (AP) calculates the average precision of a action classes in a dataset, and mean Average Precision (mAP) is the average of AP values across all action classes in the dataset, *i.e.*,

$$mAP = \frac{\sum_{i=1}^{n_{classes}} AP_i}{n_{classes}}, \tag{7}$$

where $n_{classes}$ is the number of all the action classes, and AP_i is the AP of the ith action class. This metric gives an overall performance measure of the model across different action categories, accounting for both accuracy and localisation ability. Its value ranges from 0 to 1, with higher values indicating better performance.

10.1.7. AUC-ROC

Area Under the ROC curve (AUC-ROC) is a performance measurement for classification problems at various threshold settings. ROC curve (Receiver Operating Characteristic curve) is a probability curve, and AUC represents the degree or measure of separability. In the context of VAR, AUC-ROC provides an extensive evaluation metric that assesses the model's ability to discriminate between positive and negative instances of actions. It takes into account both the true positive rate (correctly identifying positive instances) and the false positive rate (incorrectly classifying negative instances). A higher AUC-ROC score indicates better performance in terms of AR and discrimination.

The AUC-ROC curve is plotted with True Positive Rate (TPR) against the False Positive Rate (FPR) where TPR is on the y-axis and FPR is on the x-axis. The formulas for TPR and FPR are as follows:

$$TPR = Recall, \tag{8}$$

$$FPR = \frac{n_{FP}}{n_{TN} + n_{FP}}, \tag{9}$$

where n_{FP} is the number of false positives, and n_{TN} is the number of true negatives. To calculate the AUC, we need to plot TPR vs. FPR at different threshold settings, and the AUC is the area under the resulting curve. In practice, this is often done using statistical software or machine learning libraries, which have built-in functions to compute the AUC-ROC, as the calculation involves integrating the ROC curve, which is not straightforward in terms of a simple formula.

10.1.8. Jaccard index

The Jaccard index, also known as Intersection over Union (IoU), is a metric used to calculate the similarity between two sample sets. In the case of VAR, it is often used to measure the overlap between the predicted temporal action proposal and the ground truth. The formula for calculating Jaccard Index is

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{10}$$

where A is the set of actual positive instances (ground truth), B is the set of predicted positive instances, $|A \cap B|$ is the size of the intersection of sets A and B, and $|A \cup B|$ is the size of the union of the two sets.

10.1.9. Confusion matrix

The confusion matrix is not a metric but visualises a model's performance. For binary classification problems, it is a table with two rows and two columns that reports the number of false positives, false negatives, true positives, and true negatives. For multi-class problems, it extends to an $N \times N$ table, where N is the number of classes. Each cell (i, j) represents the number of instances where the true class is i and the predicted class is j. It allows more detailed analysis than a mere proportion of correct classifications (accuracy). A confusion matrix helps to understand how well the model performs across different classes, including less frequent ones. A confusion matrix helps pinpoint specific weaknesses in the model by identifying which classes are often confused. This information can guide targeted improvements in the model or data preprocessing.

10.2. Comparison of evaluation metrics usage

Table 16 shows evaluation metrics used in the VAR models reviewed. Fig. 8 shows over 75% of models use accuracy, about 35%

Table 17
Evaluation metrics used in each RGB-base VAR model.

Reference	Accuracy	Top-1 accuracy	Top-5 accuracy	Precision	Recall	F1 score	mAP	AUC-ROC	IoU	Confusion matrix
Two-stream CNN [34]	✓				✓					✓
Two-stream Network Fusion [47]	✓									
TSN [50]	✓									
Hidden Two-stream CNNs [53]	✓									
LSF CNN [54]	✓									
BS-2SCN [55]	✓									
MLENet [59]	✓	✓	✓							
2D-3D ResNet [62]	✓			✓	✓	✓				✓
STILT [57]	✓									✓
Two-stream with PCANet [63]	✓									✓
Spatiotemporal Pyramid Network [64]	✓		✓							
C3D [74]		✓	✓					✓		
P3D [95]		✓	✓				✓			
T3D [76]	✓									
Res3D [75]	✓								✓	
I3D [77]	✓									
R(2+1)D [96]		✓	✓							
T-C3D [78]	✓									
SlowFast [99]		✓	✓		✓		✓			
Asymmetric 3D-CNN [98]	✓									
D3DNet [82]		✓	✓							
STDA [100]	✓									
Two-stream 3D Dilated [101]	✓									
FSAN [102]	✓									
DAMR_3DNet [103]	✓									✓
Method in [112]	✓									✓
Method in [114]	✓									
Method in [117]	✓									✓
TS-LSTM+temporal-inception [119]	✓									
I3D-LSTM [120]	✓									
ST-D LSTM [121]	✓									
ResLNet [122]	✓									
Method in [123]	✓									✓
Method in [124]	✓			✓	✓	✓				✓
Method in [129]	✓									
Method in [130]	✓									
DC-BiLSTM [131]		✓								✓
TDS-BiLSTM [132]	✓									
Method in [133]	✓									✓
STDAN [137]	✓									
Method in [138]	✓									✓
Method in [140]	✓									✓
Method in [141]	✓	✓		✓	✓	✓				✓
HAN [143]	✓									
Method in [151]		✓					✓		✓	
STAN [155]	✓						✓		✓	
STA-CNN [158]	✓									
STA-TSN [159]	✓						✓			✓
Method in [161]	✓									
Method in [162]		✓								✓
DarkLight [163]		✓	✓							
SP-LTN [165]	✓									
MS-KFS [166]	✓									
HM-AN [167]	✓						✓			✓
TAMNet [168]	✓									
HA-SSD [169]							✓			✓
CSAM [170]	✓									
TDN [171]		✓	✓							
Method in [172]				✓	✓					
S3D RANs [173]	✓	✓								
Method in [174]	✓									✓
Action Transformer [176]							✓			
CATNet [178]		✓	✓							
MAT-EffNet [179]	✓	✓	✓							
Method in [180]	✓			✓	✓	✓				

(*continued on next page*)

confusion matrix, more than 25% Top-1 accuracy, about 20% Top-5 accuracy, about 15% recall/precision, about 10% F1-score/mAP, and less than 5% IoU/AUC-ROC.

Tables 17–18 show which evaluation metrics were used in each model among these 97 models. Some papers (*e.g.*, [34,138,159], and [167]) use multiple evaluation metrics in one paper and evaluate their models more objectively from different perspectives.

The dominance of accuracy over metrics like IoU can be explained by considering the characteristics and practicality of these metrics within the context of VAR:

Table 17 (*continued*).

Method in [182]			✓	✓							
TP-ViT [191]			✓	✓							
ViT-ReT [193]	✓				✓						
Method in [197]	✓										
Method in [198]			✓	✓							
Method in [201]	✓										
STSF [202]	✓										✓
Method in [205]			✓	✓							✓
STIP-GCN [206]	✓										
STIGPN [207]			✓	✓				✓			
Method [211]	✓										✓
Method in [215]	✓										✓
MM-ViT [218]			✓	✓							
MAiVAR [219]	✓										
Video-language GCN [220]	✓										
VideoBERT [222]			✓	✓							
MEACI-Net [223]			✓								

Table 18
Evaluation metrics used in each HPE-based VAR model.

Reference	Accuracy	Top-1 accuracy	Top-5 accuracy	Precision	Recall	F1 score	mAP	AUC-ROC	IoU	Confusion matrix
Method in [229]	✓			✓	✓	✓				✓
Method in [232]		✓	✓							
Method in [233]	✓									✓
Method in [235]	✓	✓								
Method in [236]	✓			✓	✓	✓				✓
ConvST-LSTM-Net [238]	✓			✓	✓	✓				
Method in [240]	✓			✓	✓	✓				✓
Method in [244]	✓			✓	✓	✓				✓
Method in [246]	✓									
Method in [248]	✓							✓		
Method in [249]	✓	✓	✓							✓
YogNet [250]	✓			✓	✓	✓				✓
RGBSformer [251]	✓	✓	✓							✓
Method in [252]	✓			✓	✓					✓
Method in [253]	✓			✓	✓	✓				✓

1. *Simplicity and general adoption of accuracy*: Accuracy is straightforward to calculate and interpret, making it a universal and widely accepted metric across diverse machine learning tasks. It provides a clear, single-number summary of a model's performance in terms of correct classifications out of total samples. Its simplicity makes it more accessible to a broader range of researchers and practitioners, especially in areas where subtle evaluation metrics like IoU may not be as widely understood or utilised.
2. *Dataset and task characteristics*: Many benchmark datasets used in VAR studies, such as UCF101 or HMDB51, focus on classifying entire video clips into discrete categories (*e.g.*, *jumping*, *dancing*). In such tasks, accuracy aligns naturally with the evaluation objectives, as the goal is to maximise correct predictions for complete sequences rather than evaluating partial overlaps or spatial nuances.
3. *Limited applicability of IoU in VAR*: IoU is typically used for spatial overlap evaluation (*e.g.*, object detection/segmentation). VAR often focuses on temporal patterns, not precise spatial localisation. While relevant for localised AR, this is less common than video-level classification.
4. *Computational and interpretational challenges*: IoU's complex overlap calculations can be computationally expensive, especially for high-resolution video or complex actions, making them less practical for large-scale evaluations or efficient models. They can also be harder to interpret in VAR, as they may not directly align with action recognition objectives, which often involve global video features.
5. *Historical precedence and benchmarking standards*: Accuracy's historical use in VAR studies has established it as the standard for model comparison, driving its continued adoption for consistency. Established benchmarks predominantly report accuracy, reinforcing its position.

Different models choose specific metrics based on their task characteristics, performance goals, and the nature of the data they are working with. Key factors influencing the selection of these metrics are as follows:

- Accuracy is straightforward and applicable to general classification tasks, especially when each video is expected to represent one clear action.
- Precision, Recall, and F1 score are critical for imbalanced datasets or tasks where false positives and false negatives are particularly problematic.
- mAP is favoured in action detection tasks where it is necessary to evaluate both the recognition of actions and the accuracy of their temporal localisation.
- AUC-ROC is particularly useful for evaluating the model's ability to distinguish between different actions, especially when there is class imbalance.
- IoU is essential for action detection tasks, where the location of the action is also a key part of the problem, not just the recognition of the action.
- Confusion Matrix helps understand misclassifications across multiple classes and is often used when fine-grained analysis is needed.

10.3. Models' evaluation comparison

From Tables 14–18, we can see that not every model we discussed in this paper is evaluated on the same datasets and using each metric. However, most of them were evaluated using the *accuracy* metrics on datasets UCF101 and HMDB51 (see Tables 14 and 15). In this subsection, we will compare the accuracies of these models on UCF101 and HMDB51.

Other evaluation metrics are used much less among the 97 reviewed models (see Fig. 8) and often on different datasets. So such an evaluation comparison makes little sense. Thus, we will not compare every model's performance against each metric on all the different datasets. However, for the sake of illustration, in this subsection, we compare some models against the metric of mAP on some different datasets.

10.3.1. Accuracy comparison of models based on two-stream networks

Most of the models we reviewed used datasets UCF101 and HMDB51, so we compared these two datasets accurately. Models that did not use these datasets were excluded from the comparison, as it would not be appropriate to compare them directly due to the use of different datasets. Fig. 9(a) shows the accuracy comparison of different models based on two-stream networks (the order of the models in Fig. 9 is according to their accuracies on UCF101 from high to low). From the figure, we can see:

- On UCF101, MLENet gets the highest accuracy among these models based on two-stream networks, 2D-3D ResNet gets the second, and STILT gets the third; and their average accuracy is 93.74%, and their sample Standard Deviation (SD) is 3.15.
- On HMDB51, MLENet gets the highest accuracy among these models based on two-stream networks, 2D-3D ResNet gets the second, and BS-2SCN gets the third; and their average accuracy is 70.02%, and their sample SD is 6.69.
- MLENet and 2D-3D ResNet perform outstandingly both on UCF101 and HMDB51.
- MLENet does the best in the 52 models (see Fig. 9) on UCF101.
- The average accuracy of these models on UCF101 reached 93.74%, indicating their overall good performance. The average accuracy on HMDB51 is only 70.02%, indicating that the performance on this dataset needs to be better.

The accuracy of the seminal work (Two-stream CNN, Simonyan and Zisserman [34]) is lower than other models (considered as subsequent improvements); this is a common phenomenon. Seminal work in any field is more about exploring new concepts and laying the groundwork. These initial models or methods are often more focused on proving the viability of a new approach or idea rather than optimising for the highest possible accuracy. Subsequent research builds upon the initial models, refining and improving them, leading to more robust and higher accuracy over time.

From the two value of the sample SD, we can see that the accuracy of these different models based on two-stream networks on UCF101 is more stable than that on HMDB51.

The average accuracy of a set of models on dataset UCF101 is much higher than that on dataset HMDB51. The possible reasons are as follows:

1. UCF101 is significantly larger than HMDB51, offering more varied and extensive data. This larger dataset size allows models to learn more generalised features and patterns, leading to higher accuracy.
2. The tasks or actions represented in UCF101 are more distinct from each other compared to those in HMDB51. This distinction makes it easier for models to differentiate between categories, leading to higher accuracy on UCF101.
3. HMDB51 is generally considered more challenging than UCF101. It contains more complex scenarios, or more variability in video quality, lighting, and camera angles. This complexity can make accurate AR more difficult, leading to lower average accuracy.

10.3.2. Accuracy comparison of 3D-CNN-based models

The accuracy comparison of these different 3D-CNN-based models on datasets UCF101 and HMDB51 is shown in Fig. 9(b). From the figure, we can see:

Table 19
Average accuracy and sample SD of four types of VAR models.

DL method	Average accuracy		Sample SD	
	on UCF101	on HMDB51	on UCF101	on HMDB51
Two-stream networks	93.74%	70.02%	3.15	6.69
3D-CNNs	91.16%	64.42%	4.06	7.36
RNNs/LSTM	92.44%	70.77%	5.60	12.62
AM	94.09%	70.37%	2.85	9.39

- On UCF101, FSAN gets the highest accuracy among these 3D-CNN-based models, Two-stream 3D Dilated gets the second, and STDA the third. Their average accuracy is 91.16%, and their sample SD is 4.06.
- On HMDB51, Two-stream 3D Dilated gets the highest accuracy among these 3D-CNN-based models, STDA gets the second, and FSAN gets the third. Their average accuracy is 64.42%, and their sample SD is 7.36.
- Two-stream 3D Dilated, FSAN, and STDA perform outstandingly on UCF101 and HMDB51.
- The average accuracy of these 3D-CNN-based models on UCF101 is higher than that on HMDB51, showing that there are more challenges on HMDB51.

C3D is the earliest simple solution with relatively the lowest accuracy; it provided new thinking at that time, and though it was not perfect, it gives a foundation for subsequent research, which stands on its shoulders to achieve better accuracy.

From the two value of sample SD, we can see that the accuracy fluctuates wildly among these different 3D-CNN-based models.

10.3.3. Accuracy comparison of models based on RNNs/LSTM

Fig. 9(c) shows the accuracy comparison of the models based on RNNs/LSTM on datasets UCF101 and HMBD51. From the figure, we can see:

- On UCF101, the method in [123] gets the highest accuracy among these models based on RNNs/LSTM, DC-BiLSTM gets the second, method in [246] the third. Their average accuracy is 92.44%, and their sample SD is 5.60.
- On HMDB51, the method in [238] gets the highest accuracy among these models based on RNNs/LSTM, the method in [112] gets the second, and the DC-BiLSTM gets the third. Their average accuracy is 70.77%, and their sample SD is 12.62.
- The method in [238] has the best performance among these 49 models on HMDB51 (see Fig. 9).
- ST-D LSTM gets the lowest accuracy among these models based on RNNs/LSTM both on UCF101 and HMDB51, which is also the lowest among these models reviewed.
- These models have better performance on UCF101; and there are significant differences in the performance of different models on HMDB51.

From the sample SD on HMDB51, we can see that some models perform exceptionally well, others lag significantly.

10.3.4. Accuracy comparison of AM-based models

Fig. 9(d) shows the accuracy comparison of these different AM-based models on datasets UCF101 and HMBD51. From the figure, we can see:

- On UCF101, MS-KFS gets the highest accuracy in these AM-based models, the method in [174] is the second, and the method in [180] is the third. Their average accuracy is 94.09%, and their sample SD is 2.85.

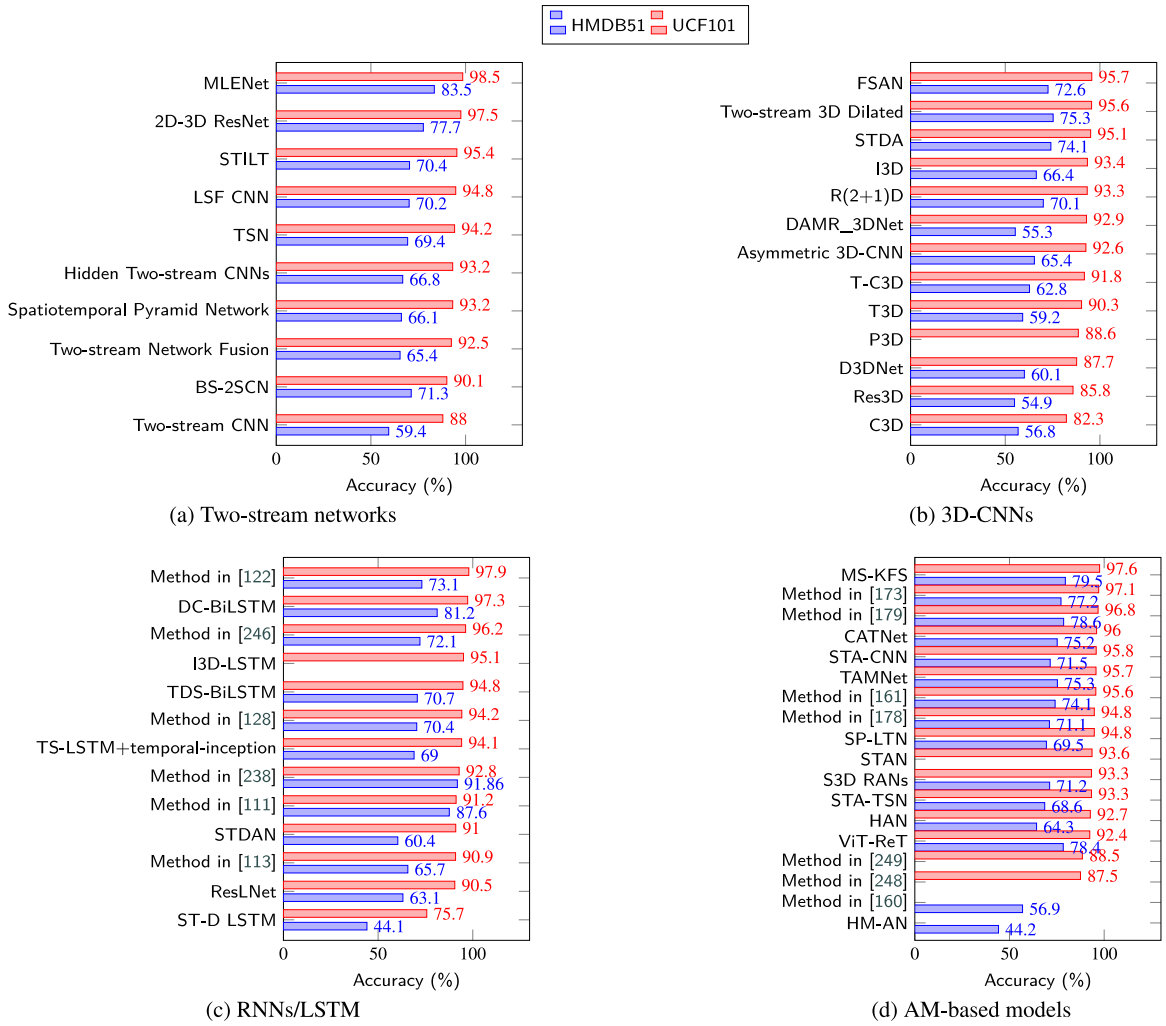

Fig. 9. Accuracy comparison of VAR models on UCF101 and HMDB51.

- On HMDB51, MS-KFS gets the highest accuracy in these AM-based models, and the method in [180] is the second, the method in [174] is the third. Their average accuracy is 70.37%, and their sample SD is 9.39.
- MS-KFS, the methods in [174,180], CATNet, and TAMNet all perform relatively well on the two datasets.
- These models perform better on UCF101; and there are significant differences in the performance of different models on HMDB51.

The sample SD shows the accuracy on UCF101 is stable, and fluctuates greatly on HMDB51. So future research needs to focus more on improving accuracy on HMDB51.

10.3.5. Discussion accuracy of the four types models

Table 19 compares the average accuracy and sample SD of models based on two-stream networks, 3D-CNN, RNNs/LSTM, and AM. From the table, we can see that the models' accuracy on dataset UCF101 is higher overall than that on dataset HMDB51. Therefore, improving the accuracy of algorithms or models on datasets such as HMDB51 is more exciting and challenging. In summary, on dataset UCF101, AM-based models perform best; the models based on two-stream networks get the second; models based on RNNs/LSTM get the third; and 3D-CNN-based models get the lowest average accuracy. On dataset HMDB51, models based on RNNs/LSTM perform best; AM-based models get the second; the models based on two-stream networks get the third; and 3D-CNN-based models get the lowest average accuracy.

Fluctuations in accuracy across different VAR models can be attributed to the main factors as follows:

1. *Model complexity and architecture*: Different models may be better suited to handle specific types of video data. Models with more complex architectures, such as deep CNNs or Transformers, typically have higher accuracy due to their ability to capture intricate features in the data. Models incorporating AMs can often perform better on complex tasks by selectively focusing on key regions of interest in the video.
2. *Model regularisation and hyperparameters*: Techniques like dropout, data augmentation, and weight decay prevent overfitting. However, improper application of regularisation techniques can cause performance issues, leading to fluctuations in accuracy. Fluctuations in accuracy can also result from improper hyperparameter choices, such as learning rate, batch size, and optimiser selection.
3. *Dataset characteristics*: A larger and more diverse dataset generally leads to better model performance and higher accuracy. Models may suffer from overfitting if the dataset is small or not sufficiently diverse. The diversity in video quality, occlusions, and environmental changes between different datasets could contribute to fluctuations in accuracy.
4. *Action complexity and data representation*: Some models may perform well on simple actions but struggle with complex ones. A model's ability to capture fine-grained motion and context is

Table 20
mAP of some VAR models.

Model	mAP	Dataset
Method in [176]	93.0%	AVA
HA-SSD [169]	89.68%	DSSL-Astronaut Gesture
HM-AN [167]	82.4%	Olympic Sports
P3D [95]	78.86%	ActivetyNet
STAN [155]	77.3%	THUMOUS14
Method in [151]	70.0%/67.0%	THUMOUS14/UCF101
STA-TSN [159]	68.4%	THUMOUS14
STIGCN [207]	59.7%	Charades
SlowFast [99]	45.2%/34.3%	Charades/AVA

Table 21
Performance by action complexity.

Action type	Example	Performance	
		DC-BiLSTM	Method in [112]
Simple (UCF101)	Jumping Jack	High	Moderate
Intermediate (HMDB51)	Brushing Hair	Moderate	High
Complex (HMDB51)	Hugging	Low	High

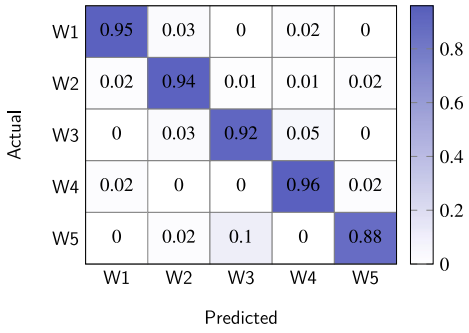

Fig. 10. Confusion matrix heatmap.

crucial for recognising complex actions. Models that rely solely on spatial features might struggle with actions that require long-term temporal context. If a model uses multi-modal input, the accuracy may fluctuate depending on how well the modalities are fused.

10.3.6. mAP comparison of some reviewed models

Among these 97 models examined in this paper, there are only 9 models which use mAP as the evaluation metric, shown in Table 20. Moreover, the datasets these 9 models used are not entirely the same.

From the local area, on dataset THUMOUS14, STAN has the highest mAP, the method in [151] gets the second, and STA-TSN gets the third. From the perspective of the mAP evaluation metric, STAN performs the best.

Relatively speaking, the method in [176] achieves a high mAP on dataset AVA, and there is significant room for improvement in the mAP of SlowFast on both datasets Charades and AVA. On Charades, STIGCN performs better than SlowFast.

10.3.7. Performance comparison of models on different types of actions

Now we compare the performance of models on different types of actions, focusing on factors such as action complexity, video complexity, and intra-class variation, with a specific comparison between the method in [112] and DC-BiLSTM.

The accuracies of the method in [112] on UCF101 and HMDB51 are 91.2% and 87.6%, respectively; while DC-BiLSTM achieves 97.3% and 81.2%. Whilst DC-BiLSTM performs significantly better on UCF101, its accuracy on HMDB51 is lower than that of the method in [112]. The key factors for the comparison are as follows:

- *Action complexity:* UCF101 predominantly features simple and distinct actions (*e.g.*, "jumping jack", "biking", and "mixing batter"), with clear visual cues and relatively static environments. These characteristics make it easier for models to extract discriminative features and achieve higher accuracy. On the other hand, HMDB51 contains more subtle and complex actions (*e.g.*, "brushing hair", "clapping hands", and "hugging"), often involving subtle movements, occlusions, and interactions with objects or other people. Such intricacies demand higher spatiotemporal reasoning, which can challenge even advanced models.
- *Video complexity:* Compared to UCF101, HMDB51 contains videos with greater scene diversity, dynamic backgrounds, and fast-moving objects. This increases the challenge for models, as they need to disentangle relevant motion and context from background noise.
- *Intra-class variation:* UCF101 exhibits less intra-class variation, where instances of the same action category are visually and temporally consistent. This consistency simplifies the task of classification. In contrast, HMDB51 presents more diverse instances within the same class, which adds difficulty for models to generalise across variations.

The two models exhibit distinct strengths and weaknesses based on the dataset characteristics as follows:

- *DC-BiLSTM*: Excels on UCF101 due to its ability to model sequential dependencies in simpler actions. However, its significant drop in accuracy on HMDB51 (81.2%) means potential limitations in spatial reasoning and handling complex or subtle actions. The model may overfit to straightforward temporal patterns in UCF101 but struggle with HMDB51's intricacies.
- *Method in [112]*: Maintains a more balanced performance on both datasets, excelling at HMDB51 (87.6%). This suggests a better balance between spatial and temporal feature extraction, making it more effective for complex and diverse datasets.

Table 21 shows a hypothetical breakdown of performance by action complexity, and the corresponding trends. The above comparison underscores that the method in [112] is better equipped to handle subtle and diverse datasets like HMDB51, while DC-BiLSTM is more optimised for simple datasets such as UCF101.

To further differentiate model capabilities, additional evaluation metrics (beyond *Accuracy*) should be considered:

- *Precision and recall:* Analysing these metrics can reveal whether a model is prone to false positives or false negatives, particularly for complex actions.
- *F1 score:* The F1 score provides a balanced measure of a model's ability to handle both precision and recall, offering deeper insights into performance.
- *Confusion matrices:* Examining confusion matrices for both models across UCF101 and HMDB51 can highlight which action categories are most frequently misclassified and provide clues about areas needing improvement.

Including visual comparisons, such as bar charts or confusion matrix heatmaps (*e.g.*, Fig. 10, where the labels $W1, \ldots, W5$ represent the different action categories or classes), can clarify the performance differences between models on simple versus complex actions. For example, a heatmap of HMDB51 categories could illustrate common misclassifications, guiding future optimisation efforts.

To address observed challenges, future work could explore incorporating AMs or multi-modal inputs to better capture subtle action patterns. These enhancements could improve performance, especially on datasets like HMDB51.

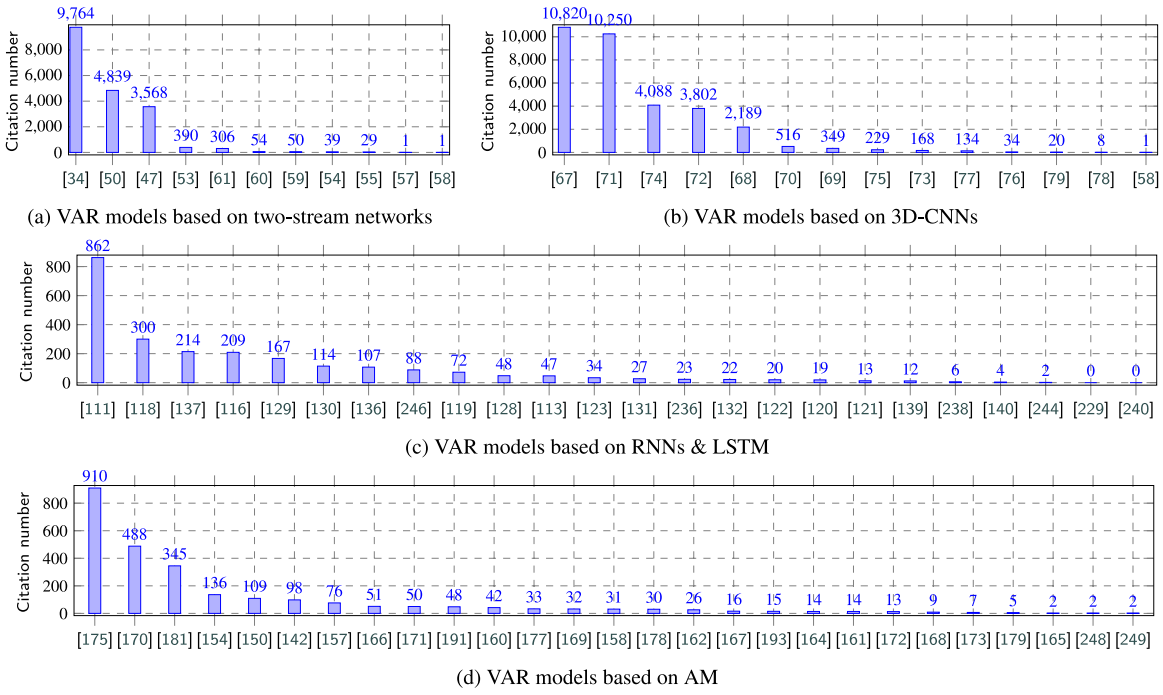

Fig. 11. Citation number comparison of the various models (data acquired from Google Scholar on 30 Dec 2024).

10.4. Citation comparison and discussion

Fig. 11(a) shows the citation rate on Google Scholar for various VAR models based on two-stream networks. Among these 11 papers, [34], published by Simonyan et al. in 2014, is the seminal work and inspired some subsequent studies, has the highest citation (more than 9700), and is still widely cited today. [47,50] were published in 2016, have gained recognition from some researchers with their unique, perspectives and have received 4839 and 3568 citations, respectively. The lowest citation rate is 1, which corresponds to the paper published in 2024, and over time, the number of citations may continue to increase. The average (excluding the highest and lowest) citation rate of these VAR models based on two-stream networks is 1031.

Fig. 11(b) shows the citation rate on Google Scholar for various 3D-CNN-based VAR models we reviewed. C3D [74] has the highest citation rate (more than 10,800), followed by I3D [77] (close to 10,300), both are classical work. The C3D network takes complete videos as input, uses 3D kernels to capture spatiotemporal features, and supports end-to-end learning. It is simple and effective, so it is frequently cited in later papers as a prime example of success in the early stages. The 3D-CNN in the early research stage did not surpass the two-stream networks based on optical flow until the I3D broke the dilemma, so it is often mentioned in subsequent papers on VAR as a shining work. R(2+1)D [96] was proposed by Tran et al. in 2018 received over 3800, and SlowFast [99] was proposed by Feichtenhofer et al. in 2019 received over 4000 citations, which is also remarkable. In contrast, other 3D-CNN-based VAR models of the same period have far less influence than these four papers. The average (excluding the highest and lowest) citation rate of these 3D-CNN-based VAR models is 1816.

Fig. 11(c) shows the citation rate on Google Scholar for various VAR models based on RNNs/LSTM. [112] has the highest citation rate (862), [119] is the second, and [138] is the third. ST-D LSTM [121] has relatively low accuracy (see Fig. 9(c)), so the paper may not get widely noticed by other researchers, and its citation rate is only 19. Ullah et al. [112] proposed a model combining CNN and DB-LSTM for VAR, the paper was published in 2017 when there was a growing interest in the combination of CNNs and LSTM networks for video analysis or AR, and the paper is easy to access, so it can get a citation rate of more than 860. Only 9 of these 24 papers have been cited 50 times or more. The average (excluding the highest and lowest) citation rate of these VAR models based on RNNs/LSTM is 70.

Fig. 11(d) shows the citation rate on Google Scholar for various VAR models based on AM. [176] has the highest citation rate (910), [171] is the second with a citation rate of 488, [182] is the third (345). Girdhar et al. [176] proposed the Action Transformer network for VAR in 2019. Transformer networks are known for their effectiveness in various domains, and the paper was published at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), which is a very influential conference, so the approach would naturally draw the attention of researchers and practitioners in the field and gets a high citation rate. Only 9 of these 27 papers have been cited 50 times or more. The average (excluding the highest and lowest) citation rate of these VAR models based on AM is 68.

Comparing the four types of models, the 3D-CNN-based model is the most popular. The reason for this may be due to its ease of implementation.

10.5. Summary

Evaluation metrics are essential for assessing the performance of DL-based VAR models and ensuring meaningful comparisons across methods. This section reviewed key metrics, including accuracy, top-1 and top-5 accuracy, precision, recall, F1 score, mean average precision (mAP), and AUC-ROC. These metrics capture various aspects of model performance, ranging from classification precision to recall balance and overall prediction reliability.

The section also compared the usage of these metrics across different VAR models, highlighting how certain models excel in specific tasks, such as short-term or long-term temporal AR. Furthermore, it presented performance comparisons of various methods based on benchmark datasets, offering insights into strengths and limitations across popular architectures, including two-stream networks, 3D-CNNs, RNNs, and attention-based models.

While these metrics provide valuable insights, challenges persist in effectively evaluating VAR systems under real-world conditions. Many existing metrics struggle to account for factors such as temporal consistency, dataset biases, and varying action complexities. Developing more

Table 22
Codes download URL of some reviewed models.

Model	Download URL
Action Transformer [176]	http://rohitgirdhar.github.io/ActionTransformer
C3D [74]	http://vlg.cs.dartmouth.edu/c3d
DarkLight [163]	https://github.com/Ticuby/Darklight-Pytorch
DC-BiLSTM [131]	https://github.com/zhiqicheng/DB-LSTM
Expressive Keypoints+GCN [235]	https://github.com/YijieYang23/SkeleT-GCN
Hidden Two-stream CNNs [53]	https://github.com/bryanyzhu/Hidden-Two-Stream
I3D [77]	https://github.com/manjotms10/activity-recognition-i3d
I3D-LSTM [120]	https://paperswithcode.com/paper/i3d-lstm-a-new-model-for-human-action
Method in [112]	https://github.com/aminullah6264/Pytorch-Action-Recognition
P3D [95]	https://github.com/ZhaofanQiu/pseudo-3d-residual-networks
R(2+1)D [96]	https://github.com/irhum/R2Plus1D-PyTorch
Res3D (R3D) [75]	https://github.com/jfzhang95/pytorch-video-recognition
SlowFast [99]	https://github.com/facebookresearch/SlowFast
T-C3D [78]	https://github.com/tc3d
T3D [76]	https://github.com/MohsenFayyaz89/T3D
TDN [171]	https://github.com/MCG-NJU/TDN
TS-LSTM+temporal-inception [119]	https://github.com/olivesgatech/TS-LSTM-and-Temporal-Inception
TSN [50]	https://github.com/yjxiong/temporal-segment-networks
Two-stream CNN [34]	https://github.com/jeffreyyihuang/two-stream-action-recognition
Two-stream Network Fusion [47]	https://github.com/feichtenhofer/twostreamfusion
STIGPN [207]	https://github.com/NingWang2049/STIGPN2

robust and comprehensive evaluation frameworks will be essential for driving future advancements in VAR.

11. Challenges and future works

Recently, DL-based VAR has significantly been improved. However, there are still numerous challenges to be addressed in future research. Some suggestions for tackling these challenges are as follows:

1. *Real-time processing*: Developing DL-based models for real-time VAR is essential for surveillance and autonomous vehicle applications. Current research focuses on lightweight architectures and optimising performance models, but challenges persist in handling high-resolution video streams and minimising latency without sacrificing accuracy. Additionally, resource constraints, such as limited computational power, highlight the need for advancements in model compression, hardware acceleration, and energy-efficient techniques. Future research should explore methods to balance speed and accuracy while addressing hardware challenges. There is also a pressing need for real-time evaluation frameworks that can robustly handle streaming video in uncontrolled environments.
2. *Robustness to variations*: Real-world videos exhibit a wide range of variations. Environmental factors like poor lighting, occlusion, and complex backgrounds could confuse VAR models. For example, in low-light conditions or with partially occluded objects, the model may misclassify or fail to recognise actions. Variations in how different individuals perform the same action (*e.g.*, differing postures, speeds, or movements) can also hinder the model's generalisation. A VAR model trained on one dataset may struggle with videos from different sources or contexts; for instance, models trained on indoor datasets may perform poorly on outdoor ones. Researchers should explore ways to enhance VAR models' robustness to these variations by incorporating invariance properties, data augmentation, and domain adaptation techniques.
3. *Multi-modal fusion*: The combination of information from different modalities (*e.g.*, audio, depth, and thermal data) has improved VAR performance by capturing complementary aspects of the scene. However, effectively fusing these modalities remains a complex challenge. Current research often focuses on early or late fusion strategies, but more advanced deep fusion techniques are needed to extract deeper semantic information. Additionally, handling noisy or incomplete data from one modality while relying on others for robustness is still an open problem. Future work should emphasise dynamic fusion methods that adapt to the content and context of the video.
4. *Fine-grained VAR*: Some existing models struggle to distinguish between similar actions or detect subtle movement differences. Developing models that accurately perform fine-grained VAR is a crucial challenge that warrants further investigation.
5. *Weakly supervised and unsupervised learning*: The reliance on large-scale annotated datasets is a significant limitation of current VAR models. Researchers should explore weakly supervised and unsupervised learning techniques to reduce the dependency on manual annotations, thus enabling the models to learn from a larger pool of unlabelled data.
6. *Explainability and interpretability*: Many VAR models, while accurate, operate as "black boxes", which makes it challenging to understand the reasoning behind their predictions. Future research should aim to develop techniques to interpret complex models, such as saliency mapping and attention visualisation, to enhance trustworthiness. Furthermore, the integration of explainability with real-time decision-making systems will be vital in ensuring that these systems can be used effectively and safely in practice.
7. *Long-term temporal modelling*: Some VAR models need to be improved in their ability to capture long-term temporal dependencies. Developing techniques to model long-range temporal information effectively could lead to significant improvements in performance, particularly for complex actions that unfold over extended periods.
8. *Applications*: VAR technology has advanced significantly and has been applied in various fields, but there are still areas where its application remains limited or is still in the early stages of exploration. For example, widespread implementation is still nascent in monitoring student engagement or assisting in physical education. Its application in monitoring and assisting older people, especially in detecting falls, is an area with significant potential that is not yet fully explored.

12. Conclusion

This survey serves as a valuable resource for researchers and practitioners, detailing SOTA techniques in VAR and guiding future advancements. It analyses the VAR field using DL techniques, focusing on

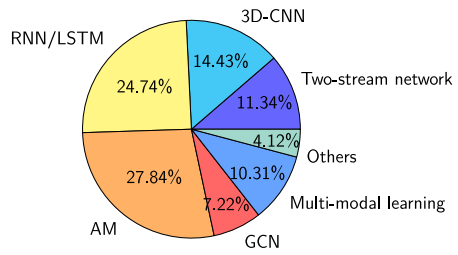

Fig. 12. Percentage of VAR models reviewed in this survey.

how various architectures have been applied to recognise and classify human actions in video data. It begins by defining VAR, distinguishing it from sensor-based AR, and outlining categories of actions, from simple gestures to complex group activities. Key processes in VAR, such as motion analysis, feature extraction, and temporal analysis, are explored, emphasising DL's role in enhancing these processes.

A significant portion discusses different DL models used in VAR, including two-stream networks for spatial and temporal information, 3D-CNNs for spatiotemporal features, and RNNs/LSTMs for modelling temporal dynamics. The paper also examines AMs that help models focus on informative video parts, analysing their complexities, strengths, and limitations.

Among a total of 97 RGB-based or HPE-based VAR models reviewed in these, 11 are based on two-stream networks (in Section 2), 14 on 3D-CNNs (in Section 3), 24 on RNN/LSTM (18 in Section 4 and 6 in Section 7.2), 27 on AMs (25 in Section 5 and 2 in Section 7.2.6), 7 on GCNs (4 in Section 6.3 and 3 in Section 7.2.2), 10 on multi-modal learning (6 in Section 6.5 and 4 in Section 7.3), and 4 on others (Sections 6.1 and 6.2). Fig. 12 displays the percentage of these VAR model types among all the types.[2] Table 22 lists code download links of some of these models.

Additionally, the survey reviews commonly used benchmark datasets and evaluation metrics, discussing how different models perform in real-world scenarios. It addresses challenges in current VAR methods, such as computational demands, capturing long-term patterns, and the need for large annotated datasets.

Finally, the paper identifies future research directions in DL-based VAR, including developing more efficient models and exploring unsupervised learning methods to minimise reliance on extensive labelled data.

CRediT authorship contribution statement

Ping Gong: Writing – original draft, Visualization, Investigation, Formal analysis, Data curation. **Xudong Luo:** Writing – review & editing, Visualization, Supervision, Software, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work we used ChatGPT and Grammarly in order to improve our English. After using this tool/service, we reviewed and edited the content very carefully as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

[2] For example, for the type of RNN/LSTM based models, its percentage among all the types is calculated as 24/97 = 24.74%.

Acknowledgements

We sincerely thank the anonymous reviewers for their time and great efforts in reviewing our paper. Their valuable comments, suggestions, and advices helped us significantly improve the quality of our manuscript. This work was partially supported by Guangxi Key Lab of Multi-source Information Mining & Security, China (24-A-01-01) and Education Ministry Key Lab of Education Blockchain and Intelligent Technology, China (EBME24-05).

Data availability

No data was used for the research described in the article.

References

[1] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. 79 (8) (1982) 2554–2558.
[2] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012).
[3] H. Wang, C. Schmid, Action recognition with improved trajectories, in: 2013 IEEE International Conference on Computer Vision, ICCV, 2013, pp. 3551–3558.
[4] T. Guo, H. Liu, Z. Chen, M. Liu, T. Wang, R. Ding, Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 762–770.
[5] M.G. Morshed, T. Sultana, A. Alam, Y.-K. Lee, Human action recognition: A taxonomy-based survey, updates, and opportunities, Sensors 23 (4) (2023) 2182.
[6] L.M. Dang, K. Min, H. Wang, M.J. Piran, C.H. Lee, H. Moon, Sensor-based and vision-based human activity recognition: A comprehensive survey, Pattern Recognit. 108 (2020) 107561.
[7] C. Feichtenhofer, A. Pinz, R.P. Wildes, Spatiotemporal multiplier networks for video action recognition, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 4768–4777.
[8] B. Li, P. Xiong, C. Han, T. Guo, Shrinking temporal attention in transformers for video action recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 1263–1271.
[9] G. Gao, Z. Liu, G. Zhang, J. Li, A. Qin, DANet: Semi-supervised differentiated auxiliaries guided network for video action recognition, Neural Netw. 158 (2023) 121–131.
[10] M.A. Khan, M. Sharif, T. Akram, M. Raza, T. Saba, A. Rehman, Hand-crafted and deep convolutional neural network features fusion and selection strategy: An application to intelligent human action recognition, Appl. Soft Comput. 87 (2020) 105986.
[11] M. Sharif, M.A. Khan, T. Akram, M.Y. Javed, T. Saba, A. Rehman, A framework of human detection and action recognition based on uniform segmentation and combination of Euclidean distance and joint entropy-based features selection, EURASIP J. Image Video Process. 2017 (2017) 1–18.
[12] M.A. Khan, T. Akram, M. Sharif, M.Y. Javed, N. Muhammad, M. Yasmin, An implementation of optimized framework for action classification using multilayers neural network on selected fused features, Pattern Anal. Appl. 22 (2019) 1377–1397.
[13] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[14] J. Gall, A. Yao, N. Razavi, L. Van Gool, V. Lempitsky, Hough forests for object detection, tracking, and action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 33 (11) (2011) 2188–2202.
[15] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, Int. J. Comput. Vis. 103 (2013) 60–79.
[16] C. Yuan, X. Li, W. Hu, H. Ling, S. Maybank, 3D R transform on spatio-temporal interest points for action recognition, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 724–730.
[17] S.C.B. Dash, S.R. Mishra, K. Srujan Raju, L. Narasimha Prasad, Human action recognition using a hybrid deep learning heuristic, Soft Comput. 25 (20) (2021) 13079–13092.
[18] B. Sun, D. Kong, S. Wang, J. Li, B. Yin, X. Luo, GAN for vision, KG for relation: A two-stage network for zero-shot action recognition, Pattern Recognit. 126 (2022) 108563.
[19] C. Yang, Y. Xu, J. Shi, B. Dai, B. Zhou, Temporal pyramid network for action recognition, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 591–600.
[20] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[21] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.
[22] V. Veeriah, N. Zhuang, G.-J. Qi, Differential recurrent neural networks for action recognition, in: 2015 IEEE International Conference on Computer Vision, ICCV, 2015, pp. 4041–4049.
[23] A. Montes, A. Salvador, S. Pascual, X. Giro-i Nieto, Temporal activity detection in untrimmed videos with recurrent neural networks, 2016, arXiv preprint arXiv:1608.08128.
[24] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[25] R.C. Staudemeyer, E.R. Morris, Understanding LSTM—a tutorial into long short-term memory recurrent neural networks, 2019, arXiv preprint arXiv:1909.09586.
[26] T. Özyer, D.S. Ak, R. Alhajj, Human action recognition approaches with video datasets—A survey, Knowl.-Based Syst. 222 (2021) 106995.
[27] H.H. Pham, L. Khoudour, A. Crouzil, P. Zegers, S.A. Velastin, Video-based human action recognition using deep learning: A review, 2022, arXiv preprint arXiv:2208.03775.
[28] B.K. Horn, B.G. Schunck, Determining optical flow, Artificial Intelligence 17 (1–3) (1981) 185–203.
[29] Y. Zhu, X. Li, C. Liu, M. Zolfaghari, Y. Xiong, C. Wu, Z. Zhang, J. Tighe, R. Manmatha, M. Li, A comprehensive study of deep video action recognition, 2020, arXiv preprint arXiv:2012.06567.
[30] P. Pareek, A. Thakkar, A survey on video-based human action recognition: Recent updates, datasets, challenges, and applications, Artif. Intell. Rev. 54 (2021) 2259–2322.
[31] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, J. Liu, Human action recognition from various data modalities: A review, IEEE Trans. Pattern Anal. Mach. Intell. 45 (3) (2022) 3200–3225.
[32] C. Wang, J. Yan, A comprehensive survey of RGB-based and skeleton-based human action recognition, IEEE Access 11 (2023) 53880–53898.
[33] M. Karim, S. Khalid, A. Aleryani, J. Khan, I. Ullah, Z. Ali, Human action recognition systems: A review of the trends and state-of-the-art, IEEE Access 12 (2024) 36372–36390.
[34] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, 2014, arXiv preprint arXiv:1406.2199.
[35] M.A. Goodale, A.D. Milner, Separate visual pathways for perception and action, Trends Neurosci. 15 (1) (1992) 20–25.
[36] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, IEEE Intell. Syst. Appl. 13 (4) (1998) 18–28.
[37] K. Soomro, A.R. Zamir, M. Shah, UCF101: A dataset of 101 human actions classes from videos in the wild, 2012, arXiv preprint arXiv:1212.0402.
[38] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: A large video database for human motion recognition, in: 2011 International Conference on Computer Vision, 2011, pp. 2556–2563.
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 1–9.
[41] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on Machine Learning, 2015, pp. 448–456.
[42] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
[43] S.Y. Boulahia, A. Amamra, M.R. Madi, S. Daikh, Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition, Mach. Vis. Appl. 32 (6) (2021) 121–139.
[44] K. Gadzicki, R. Khamsehashari, C. Zetzsche, Early vs late fusion in multimodal convolutional neural networks, in: 2020 IEEE 23rd International Conference on Information Fusion, FUSION, 2020, pp. 1–6.
[45] L.I. Kuncheva, A theoretical study on six classifier fusion strategies, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2) (2002) 281–286.
[46] C. Hori, T. Hori, T.-Y. Lee, Z. Zhang, B. Harsham, J.R. Hershey, T.K. Marks, K. Sumi, Attention-based multimodal fusion for video description, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, pp. 4193–4202.
[47] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 1933–1941.
[48] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
[49] M. Tan, Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, 2019, pp. 6105–6114.
[50] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: Towards good practices for deep action recognition, in: Computer Vision–ECCV 2016, in: Lecture Notes in Computer Science, vol. 9912, Springer, 2016, pp. 20–36.
[51] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks for action recognition in videos, IEEE Trans. Pattern Anal. Mach. Intell. 41 (11) (2018) 2740–2755.
[52] F.C. Heilbron, V. Escorcia, B. Ghanem, J.C. Niebles, ActivityNet: A large-scale video benchmark for human activity understanding, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 961–970.
[53] Y. Zhu, Z. Lan, S. Newsam, A. Hauptmann, Hidden two-stream convolutional networks for action recognition, in: Computer Vision–ACCV 2018, in: Lecture Notes in Computer Science, vol. 11363, Springer, 2019, pp. 363–378.
[54] Y. Wan, Z. Yu, Y. Wang, X. Li, Action recognition based on two-stream convolutional networks with long-short-term spatiotemporal features, IEEE Access 8 (2020) 85284–85293.
[55] Z. Wang, H. Lu, J. Jin, K. Hu, Human action recognition based on improved two-stream convolution network, Appl. Sci. 12 (12) (2022) 5784.
[56] L. Yang, R.-Y. Zhang, L. Li, X. Xie, SimAM: A simple, parameter-free attention module for convolutional neural networks, in: Proceedings of the 38th International Conference on Machine Learning, Vol. 139, 2021, pp. 11863–11874.
[57] T. Liu, Y. Ma, W. Yang, W. Ji, R. Wang, P. Jiang, Spatial-temporal interaction learning based two-stream network for action recognition, Inform. Sci. 606 (2022) 864–876.
[58] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al., The kinetics human action video dataset, 2017, arXiv preprint arXiv:1705.06950.
[59] F. Wang, X. Li, H. Xiong, H. Mo, Y. Li, MLENet: Multi-level extraction network for video action recognition, Pattern Recognit. 154 (2024) 110614.
[60] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al., The "something something" video database for learning and evaluating visual common sense, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, pp. 5842–5850.
[61] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets, 2014, arXiv preprint arXiv:1405.3531.
[62] S. Yosry, L. Elrefaei, R. ElKamaar, R.R. Ziedan, Various frameworks for integrating image and video streams for spatiotemporal information learning employing 2D–3D residual networks for human action recognition, Discov. Appl. Sci. 6 (4) (2024) 141.
[63] A. Abdelbaky, S. Aly, Two-stream spatiotemporal feature fusion for human action recognition, Vis. Comput. 37 (7) (2021) 1821–1835.
[64] Y. Wang, M. Long, J. Wang, P.S. Yu, Spatiotemporal pyramid network for video action recognition, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 1529–1538.
[65] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: A local SVM approach, in: Proceedings of the 17th International Conference on Pattern Recognition, Vol. 3, 2004, pp. 32–36.
[66] M.D. Rodriguez, J. Ahmed, M. Shah, Action mach a spatio-temporal maximum average correlation height filter for action recognition, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[67] K. Hara, H. Kataoka, Y. Satoh, Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and imagenet? in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6546–6555.
[68] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox, FlowNet 2.0: Evolution of optical flow estimation with deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2462–2470.
[69] Z. Teed, J. Deng, RAFT: Recurrent all-pairs field transforms for optical flow, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, Springer, 2020, pp. 402–419.
[70] B. Batalo, L.S. Souza, B.B. Gatto, N. Sogi, K. Fukui, Temporal-stochastic tensor features for action recognition, Mach. Learn. Appl. 10 (2022) 100407.
[71] B. Batalo, L.S. Souza, B.B. Gatto, N. Sogi, K. Fukui, Analysis of temporal tensor datasets on product grassmann manifold, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2022, pp. 4869–4877.
[72] S. Ji, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2012) 221–231.
[73] X. Huang, Z. Cai, A review of video action recognition based on 3D convolution, Comput. Electr. Eng. 108 (2023) 108713.
[74] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3D convolutional networks, in: 2015 IEEE International Conference on Computer Vision, ICCV, 2015, pp. 4489–4497.
[75] D. Tran, J. Ray, Z. Shou, S.-F. Chang, M. Paluri, ConvNet architecture search for spatiotemporal feature learning, 2017, arXiv preprint arXiv:1708.05038.
[76] A. Diba, M. Fayyaz, V. Sharma, A.H. Karami, M.M. Arzani, R. Yousefzadeh, L. Van Gool, Temporal 3D ConvNets: New architecture and transfer learning for video classification, 2017, arXiv preprint arXiv:1711.08200.
[77] J. Carreira, A. Zisserman, Quo vadis, action recognition? A new model and the kinetics dataset, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 6299–6308.

[78] K. Liu, W. Liu, C. Gan, M. Tan, H. Ma, T-C3D: Temporal convolutional 3D network for real-time action recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018, pp. 7138–7145.
[79] K. O'Shea, R. Nash, An introduction to convolutional neural networks, 2015, arXiv preprint arXiv:1511.08458.
[80] R. Yamashita, M. Nishio, R.K.G. Do, K. Togashi, Convolutional neural networks: An overview and application in radiology, Insights Imaging 9 (2018) 611–629.
[81] L. Alzubaidi, J. Zhang, A.J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M.A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions, J. Big Data 8 (2021) 1–74.
[82] S. Jiang, Y. Qi, H. Zhang, Z. Bai, X. Lu, P. Wang, D3D: Dual 3-D convolutional network for real-time action recognition, IEEE Trans. Ind. Inform. 17 (7) (2020) 4584–4593.
[83] K. Hara, H. Kataoka, Y. Satoh, Learning spatio-temporal features with 3D residual networks for action recognition, in: 2017 IEEE International Conference on Computer Vision Workshops, ICCVW, 2017, pp. 3154–3160.
[84] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: Proceedings of the 27th International Conference on Machine Learning, 2010, pp. 807–814.
[85] A.F. Agarap, Deep learning using rectified linear units (ReLU), 2018, arXiv preprint arXiv:1803.08375.
[86] A.L. Maas, A.Y. Hannun, A.Y. Ng, et al., Rectifier nonlinearities improve neural network acoustic models, in: Proceedings of the 30th International Conference on Machine Learning, 2013, p. 3.
[87] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: 2015 IEEE International Conference on Computer Vision, ICCV, 2015, pp. 1026–1034.
[88] H. Gholamalinezhad, H. Khosravi, Pooling methods in deep neural networks, a review, 2020, arXiv preprint arXiv:2009.07485.
[89] R. Riad, O. Teboul, D. Grangier, N. Zeghidour, Learning strides in convolutional neural networks, 2022, arXiv preprint arXiv:2202.01653.
[90] J. Yepez, S.-B. Ko, Stride 2 1-D, 2-D, and 3-D winograd for convolutional neural networks, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 28 (4) (2020) 853–863.
[91] S. Zhao, Y. Liu, Y. Han, R. Hong, Q. Hu, Q. Tian, Pooling the convolutional layers in deep ConvNets for video action recognition, IEEE Trans. Circuits Syst. Video Technol. 28 (8) (2017) 1839–1849.
[92] J. Zhang, L. Zi, Y. Hou, M. Wang, W. Jiang, D. Deng, A deep learning-based approach to enable action recognition for construction equipment, Adv. Civ. Eng. 2020 (2020) 1–14.
[93] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 4700–4708.
[94] X. Ouyang, S. Xu, C. Zhang, P. Zhou, Y. Yang, G. Liu, X. Li, A 3D-CNN and LSTM based multi-task learning architecture for action recognition, IEEE Access 7 (2019) 40757–40770.
[95] Z. Qiu, T. Yao, T. Mei, Learning spatio-temporal representation with pseudo-3D residual networks, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, pp. 5533–5541.
[96] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, A closer look at spatiotemporal convolutions for action recognition, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 6450–6459.
[97] S. Xie, C. Sun, J. Huang, Z. Tu, K. Murphy, Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification, in: Computer Vision – ECCV 2018, in: Lecture Notes in Computer Science, vol. 11219, Springer, 2018, pp. 305–321.
[98] H. Yang, C. Yuan, B. Li, Y. Du, J. Xing, W. Hu, S.J. Maybank, Asymmetric 3D convolutional neural networks for action recognition, Pattern Recognit. 85 (2019) 1–12.
[99] C. Feichtenhofer, H. Fan, J. Malik, K. He, Slowfast networks for video recognition, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV, 2019, pp. 6202–6211.
[100] J. Li, X. Liu, M. Zhang, D. Wang, Spatio-temporal deformable 3D ConvNets with attention for action recognition, Pattern Recognit. 98 (2020) 107037.
[101] X. Xiong, W. Min, Q. Han, Q. Wang, C. Zha, Action recognition using action sequences optimization and two-stream 3D dilated neural network, Comput. Intell. Neurosci. 2022 (2022) 6608448.
[102] B. Chen, F. Meng, H. Tang, G. Tong, Two-level attention module based on spurious-3D residual networks for human action recognition, Sensors 23 (3) (2023) 1707.
[103] Q. Tian, S. Li, Y. Zhang, H. Lu, H. Pan, Action recognition method based on a novel keyframe extraction method and enhanced 3D convolutional neural network, Int. J. Mach. Learn. Cybern. (2024) 1–17.
[104] J. Huang, W. Zhou, Q. Zhang, H. Li, W. Li, Video-based sign language recognition without temporal segmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018, pp. 2257–2264.
[105] Y. Ming, S. Cao, R. Zhang, Z. Li, Y. Chen, Y. Song, H. Qu, Understanding hidden memories of recurrent neural networks, in: 2017 IEEE Conference on Visual Analytics Science and Technology, VAST, 2017, pp. 13–24.
[106] W. Fang, Y. Chen, Q. Xue, Survey on research of RNN-based spatio-temporal sequence prediction algorithms, J. Big Data 3 (3) (2021) 97.
[107] J. Liu, A. Shahroudy, D. Xu, A.C. Kot, G. Wang, Skeleton-based action recognition using spatio-temporal LSTM network with trust gates, IEEE Trans. Pattern Anal. Mach. Intell. 40 (12) (2017) 3007–3021.
[108] D. Li, R. Wang, Context-LSTM: A robust classifier for video detection on UCF101, 2022, arXiv preprint arXiv:2203.06610.
[109] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (11) (1997) 2673–2681.
[110] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv preprint arXiv:1412.3555.
[111] R. Dey, F.M. Salem, Gate-variants of gated recurrent unit (GRU) neural networks, in: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems, MWSCAS, 2017, pp. 1597–1600.
[112] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, S.W. Baik, Action recognition in video sequences using deep bi-directional LSTM with CNN features, IEEE Access 6 (2017) 1155–1166.
[113] H. Yang, C. Yuan, J. Xing, W. Hu, SCNN: Sequential convolutional neural network for human action recognition in videos, in: 2017 IEEE International Conference on Image Processing, ICIP, 2017, pp. 355–359.
[114] Y. Yuan, Y. Zhao, Q. Wang, Action recognition using spatial-optical data organization and sequential learning framework, Neurocomputing 315 (2018) 221–233.
[115] C. Zhao, J.G. Han, X. Xu, CNN and RNN based neural networks for action recognition, J. Phys.: Conf. Ser. 1087 (6) (2018) 062013.
[116] A. Sarabu, A.K. Santra, Human action recognition in videos using convolution long short-term memory network with spatio-temporal networks, Emerg. Sci. J. 5 (1) (2021) 25–33.
[117] C. Dai, X. Liu, J. Lai, Human action recognition using two-stream attention based LSTM networks, Appl. Soft Comput. 86 (2020) 105820.
[118] F.A. Dharejo, M. Zawish, Y. Zhou, S. Davy, K. Dev, S.A. Khowaja, Y. Fu, N.M.F. Qureshi, FuzzyAct: A fuzzy-based framework for temporal activity recognition in IoT applications using RNN and 3D-DWT, IEEE Trans. Fuzzy Syst. 30 (11) (2022) 4578–4592.
[119] C.-Y. Ma, M.-H. Chen, Z. Kira, G. AlRegib, TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition, Signal Process., Image Commun. 71 (2019) 76–87.
[120] X. Wang, Z. Miao, R. Zhang, S. Hao, I3D-LSTM: A new model for human action recognition, in: IOP Conference Series: Materials Science and Engineering, 2019, 032035.
[121] K. Hu, F. Zheng, L. Weng, Y. Ding, J. Jin, Action recognition algorithm of spatio–temporal differential LSTM based on feature enhancement, Appl. Sci. 11 (17) (2021) 7876.
[122] T. Wang, J. Li, H.-N. Wu, C. Li, H. Snoussi, Y. Wu, ResLNet: deep residual LSTM network with longer input for action recognition, Front. Comput. Sci. 16 (6) (2022) 166334.
[123] M.A. Abdelrazik, A. Zekry, W.A. Mohamed, Efficient hybrid algorithm for human action recognition, J. Image Graph. 11 (1) (2023) 72–81.
[124] A.C. Cob-Parro, C. Losada-Gutiérrez, M. Marrón-Romera, A. Gardel-Vicente, I. Bravo-Muñoz, A new framework for deep learning video based human action recognition on the edge, Expert Syst. Appl. 238 (2024) 122220.
[125] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot multibox detector, in: Computer Vision–ECCV 2016, in: Lecture Notes in Computer Science, vol. 9905, Springer, 2016, pp. 21–37.
[126] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, IEEE Trans. Pattern Anal. Mach. Intell. 29 (12) (2007) 2247–2253.
[127] V. Kulathumani, R. Kavi, S. Ramagiri, WVU multi-view action recognition dataset, 2011, Available on: http://csee.wvu.edu/~vkkulathumani/wvu-action.html#download2.
[128] D. Weinland, R. Ronfard, E. Boyer, Free viewpoint action recognition using motion history volumes, Comput. Vis. Image Underst. 104 (2–3) (2006) 249–257.
[129] W. Li, W. Nie, Y. Su, Human action recognition based on selected spatio-temporal features via bidirectional LSTM, IEEE Access 6 (2018) 44211–44220.
[130] A. Hanson, K. PNVR, S. Krishnagopal, L. Davis, Bidirectional convolutional LSTM for the detection of violence in videos, in: Computer Vision–ECCV 2018 Workshops, in: Lecture Notes in Computer Science, vol. 11130, Springer, 2019, pp. 280–295.
[131] J.-Y. He, X. Wu, Z.-Q. Cheng, Z. Yuan, Y.-G. Jiang, DB-LSTM: Densely-connected bi-directional LSTM for human action recognition, Neurocomputing 444 (2021) 319–331.
[132] K.S. Tan, K.M. Lim, C.P. Lee, L.C. Kwek, Bidirectional long short-term memory with temporal dense sampling for human action recognition, Expert Syst. Appl. 210 (2022) 118484.
[133] N. Hassan, A.S.M. Miah, J. Shin, A deep bidirectional LSTM model enhanced by transfer-learning-based feature extraction for dynamic human activity recognition, Appl. Sci. 14 (2) (2024) 603.

[134] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos "in the wild", in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1996–2003.
[135] K. Soomro, A.R. Zamir, Action recognition in realistic sports videos, in: Computer Vision in Sports, in: Advances in Computer Vision and Pattern Recognition, Springer, 2015, pp. 181–208.
[136] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, M.J. Black, Towards understanding action recognition, in: 2013 IEEE International Conference on Computer Vision, 2013, pp. 3192–3199.
[137] Z. Zhang, Z. Lv, C. Gan, Q. Zhu, Human action recognition using convolutional LSTM and fully-connected LSTM with different attentions, Neurocomputing 410 (2020) 304–316.
[138] K. Muhammad, A. Ullah, A.S. Imran, M. Sajjad, M.S. Kiran, G. Sannino, V.H.C. de Albuquerque, et al., Human action recognition using attention based LSTM network with dilated CNN features, Future Gener. Comput. Syst. 125 (2021) 820–830.
[139] C. Zach, T. Pock, H. Bischof, A duality based approach for realtime TV-L1 optical flow, in: Pattern Recognition: DAGM 2007, in: Lecture Notes in Computer Science, vol. 4713, Springer, 2007, pp. 214–223.
[140] K. Bayoudh, F. Hamdaoui, A. Mtibaa, An attention-based hybrid 2D/3D CNN-LSTM for human action recognition, in: 2022 2nd International Conference on Computing and Information Technology, ICCIT, 2022, pp. 97–103.
[141] D. Kumari, R.S. Anand, Isolated video-based sign language recognition using a hybrid CNN-LSTM framework based on attention mechanism, Electronics 13 (7) (2024) 1229.
[142] D. Li, C. Rodriguez, X. Yu, H. Li, Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 1459–1469.
[143] Y. Wang, S. Wang, J. Tang, N. O'Hare, Y. Chang, B. Li, Hierarchical attention network for action recognition in videos, 2016, arXiv preprint arXiv:1607.06416.
[144] M.E. Kalfaoglu, S. Kalkan, A.A. Alatan, Late temporal modeling in 3D CNN architectures with BERT for action recognition, in: Computer Vision–ECCV 2020 Workshops, in: Lecture Notes in Computer Science, vol. 12539, Springer, 2020, pp. 731–747.
[145] S. Liu, X. Ma, Attention-driven appearance-motion fusion network for action recognition, IEEE Trans. Multimed. 25 (2022) 2573–2584.
[146] L. Shrestha, S. Dubey, F. Olimov, M.A. Rafique, M. Jeon, 3D convolutional with attention for action recognition, 2022, arXiv preprint arXiv:2206.02203.
[147] M. Dong, Z. Fang, Y. Li, S. Bi, J. Chen, AR3D: Attention residual 3D network for human action recognition, Sensors 21 (5) (2021) 1656.
[148] W. Du, Y. Wang, Y. Qiao, Recurrent spatial-temporal attention network for action recognition in videos, IEEE Trans. Image Process. 27 (3) (2017) 1347–1360.
[149] H. Ge, Z. Yan, W. Yu, L. Sun, An attention mechanism based convolutional LSTM network for video action recognition, Multimedia Tools Appl. 78 (2019) 20533–20556.
[150] S. Sudhakaran, S. Escalera, O. Lanz, LSTA: Long short-term attention for egocentric action recognition, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 9954–9963.
[151] L. Meng, B. Zhao, B. Chang, G. Huang, W. Sun, F. Tung, L. Sigal, Interpretable spatio-temporal attention for video action recognition, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop, ICCVW, 2019, pp. 1513–1522.
[152] H. Zhao, X. Jin, Human action recognition based on improved fusion attention CNN and RNN, in: 2020 5th International Conference on Computational Intelligence and Applications, ICCIA, 2020, pp. 108–112.
[153] G. Liu, J. Guo, Bidirectional LSTM with attention mechanism and convolutional layer for text classification, Neurocomputing 337 (2019) 325–338.
[154] D. Hu, An introductory survey on attention mechanisms in NLP problems, in: Intelligent Systems and Applications: IntelliSys 2019, in: Advances in Intelligent Systems and Computing, vol. 1038, Springer, 2020, pp. 432–448.
[155] D. Li, T. Yao, L.-Y. Duan, T. Mei, Y. Rui, Unified spatio-temporal attention networks for action recognition in videos, IEEE Trans. Multimed. 21 (2) (2018) 416–428.
[156] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, A.C. Loui, Consumer video understanding: A benchmark database and an evaluation of human and machine performance, in: Proceedings of the 1st ACM International Conference on Multimedia Retrieval, 2011, pp. 1–8.
[157] H. Idrees, A.R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, M. Shah, The THUMOS challenge on action recognition for videos "in the wild", Comput. Vis. Image Underst. 155 (2017) 1–23.
[158] H. Yang, C. Yuan, L. Zhang, Y. Sun, W. Hu, S.J. Maybank, STA-CNN: Convolutional spatial-temporal attention learning for action recognition, IEEE Trans. Image Process. 29 (2020) 5783–5793.
[159] G. Yang, Y. Yang, Z. Lu, J. Yang, D. Liu, C. Zhou, Z. Fan, STA-TSN: Spatial-temporal attention temporal segment network for action recognition in video, Plos One 17 (3) (2022) e0265115.
[160] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[161] D. Purwanto, R.R.A. Pramono, Y.-T. Chen, W.-H. Fang, Three-stream network with bidirectional self-attention for action recognition in extreme low resolution videos, IEEE Signal Process. Lett. 26 (8) (2019) 1187–1191.
[162] F. Anvarov, D.H. Kim, B.C. Song, Action recognition using deep 3D CNNs with sequential feature aggregation and attention, Electronics 9 (1) (2020) 147.
[163] R. Chen, J. Chen, Z. Liang, H. Gao, S. Lin, Darklight networks for action recognition in the dark, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2021, pp. 846–852.
[164] Y. Xu, J. Yang, H. Cao, K. Mao, J. Yin, S. See, ARID: A new dataset for recognizing action in the dark, in: Deep Learning for Human Activity Recognition: DL-HAR 2020, in: Communications in Computer and Information Science, vol. 1370, Springer, 2021, pp. 70–84.
[165] H. Li, J. Huang, M. Zhou, Q. Shi, Q. Fei, Self-attention pooling-based long-term temporal network for action recognition, IEEE Trans. Cogn. Dev. Syst. 15 (1) (2022) 65–77.
[166] L. Xia, X. Wen, Multi-stream network with key frame sampling for human action recognition, J. Supercomput. (2024) 1–31.
[167] S. Yan, J.S. Smith, W. Lu, B. Zhang, Hierarchical multi-scale attention networks for action recognition, Signal Process., Image Commun. 61 (2018) 73–84.
[168] H. Sang, Z. Zhao, D. He, Two-level attention model based video action recognition network, IEEE Access 7 (2019) 118388–118401.
[169] L. Gu, L. Zhang, Z. Wang, Hierarchical attention-based astronaut gesture recognition: A dataset and CNN model, IEEE Access 8 (2020) 68787–68798.
[170] H. Wu, X. Ma, Y. Li, Convolutional networks with channel and STIPs attention model for action recognition in videos, IEEE Trans. Multimed. 22 (9) (2019) 2293–2306.
[171] L. Wang, Z. Tong, B. Ji, G. Wu, TDN: Temporal difference networks for efficient action recognition, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 1895–1904.
[172] M. Ullah, M.M. Yamin, A. Mohammed, S.D. Khan, H. Ullah, F.A. Cheikh, Attention-based LSTM network for action recognition in sports, Electron. Imaging 33 (2021) 1–6.
[173] B. Chen, H. Tang, Z. Zhang, G. Tong, B. Li, Video-based action recognition using spurious-3D residual attention networks, IET Image Process. 16 (11) (2022) 3097–3111.
[174] E. Dastbaravardeh, S. Askarpour, M. Saberi Anari, K. Rezaee, Channel attention-based approach with autoencoder network for human action recognition in low-resolution frames, Int. J. Intell. Syst. 2024 (1) (2024) 1052344.
[175] K.K. Reddy, M. Shah, Recognizing 50 human action categories of web videos, Mach. Vis. Appl. 24 (5) (2013) 971–981.
[176] R. Girdhar, J. Carreira, C. Doersch, A. Zisserman, Video action transformer network, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 244–253.
[177] C. Gu, C. Sun, D.A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, et al., AVA: A video dataset of spatio-temporally localized atomic visual actions, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 6047–6056.
[178] J. Wang, X. Peng, Y. Qiao, Cascade multi-head attention networks for action recognition, Comput. Vis. Image Underst. 192 (2020) 102898.
[179] A. Zhou, Y. Ma, W. Ji, M. Zong, P. Yang, M. Wu, M. Liu, Multi-head attention-based two-stream EfficientNet for action recognition, Multimedia Syst. 29 (2) (2023) 487–498.
[180] A. Hussain, S.U. Khan, N. Khan, W. Ullah, A. Alkhayyat, M. Alharbi, S.W. Baik, Shots segmentation-based optimized dual-stream framework for robust human activity recognition in surveillance video, Alex. Eng. J. 91 (2024) 632–647.
[181] A. Dosovitskiy, An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.
[182] C. Plizzari, M. Cannici, M. Matteucci, Skeleton-based action recognition via spatial and temporal transformer networks, Comput. Vis. Image Underst. 208 (2021) 103219.
[183] V. Mazzia, S. Angarano, F. Salvetti, F. Angelini, M. Chiaberge, Action transformer: A self-attention model for short-time pose-based human action recognition, Pattern Recognit. 124 (2022) 108487.
[184] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.
[185] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, et al., Swin transformer V2: Scaling up capacity and resolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12009–12019.
[186] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, H. Hu, Video swin transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3202–3211.
[187] G. Bertasius, H. Wang, L. Torresani, Is space-time attention all you need for video understanding? in: Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 4–5.

[188] A. Shahroudy, J. Liu, T.-T. Ng, G. Wang, NTU RGB+D: A large scale dataset for 3d human activity analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1010–1019.
[189] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, A.C. Kot, NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding, IEEE Trans. Pattern Anal. Mach. Intell. 42 (10) (2019) 2684–2701.
[190] S. Yan, Y. Xiong, D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018, pp. 7444–7452.
[191] Y. Jing, F. Wang, TP-ViT: A two-pathway vision transformer for video action recognition, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2022, pp. 2185–2189.
[192] D. Shao, Y. Zhao, B. Dai, D. Lin, FineGym: A hierarchical video dataset for fine-grained action understanding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 2616–2625.
[193] J. Wensel, H. Ullah, A. Munir, ViT-ReT: Vision and recurrent transformer neural networks for human activity recognition in videos, IEEE Access 11 (2023) 72227–72249.
[194] Y. Zhou, X. Sun, Z.-J. Zha, W. Zeng, MiCT: Mixed 3D/2D convolutional tube for human action recognition, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 449–458.
[195] M. Lee, S. Lee, S. Son, G. Park, N. Kwak, Motion feature network: Fixed motion filter for action recognition, in: Computer Vision – ECCV 2018, in: Lecture Notes in Computer Science, vol. 11219, Springer, 2018, pp. 387–403.
[196] J. Li, P. Wei, Y. Zhang, N. Zheng, A Slow-I-Fast-P architecture for compressed video action recognition, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 2039–2047.
[197] K.-H. Wu, C.-T. Chiu, Action recognition using multi-scale temporal shift module and temporal feature difference extraction based on 2D CNN, J. Softw. Eng. Appl. 14 (5) (2021) 172–188.
[198] Y.Y. Joefrie, M. Aono, Video action recognition using motion and multi-view excitation with temporal aggregation, Entropy 24 (11) (2022) 1663.
[199] J. Materzynska, G. Berger, I. Bax, R. Memisevic, The Jester Dataset: A large-scale video dataset of human gestures, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop, ICCVW, 2019, pp. 2874–2882.
[200] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S.A. Bargal, T. Yan, L. Brown, Q. Fan, D. Gutfreund, C. Vondrick, et al., Moments in time dataset: One million videos for event understanding, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2) (2019) 502–508.
[201] X. Liu, X. Yang, Multi-stream with deep convolutional neural networks for human action recognition in videos, in: Neural Information Processing: ICONIP 2018, in: Lecture Notes in Computer Science, vol. 11301, Springer, 2018, pp. 251–262.
[202] M. Zong, R. Wang, Y. Ma, W. Ji, Spatial and temporal saliency based four-stream network with multi-task learning for action recognition, Appl. Soft Comput. 132 (2023) 109884.
[203] S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph convolutional networks: a comprehensive review, Comput. Soc. Netw. 6 (1) (2019) 1–23.
[204] T. Ahmad, L. Jin, X. Zhang, S. Lai, G. Tang, L. Lin, Graph convolutional neural network for human action recognition: A comprehensive survey, IEEE Trans. Artif. Intell. 2 (2) (2021) 128–145.
[205] H.-B. Jang, D.-J. Kim, C.-W. Lee, Human action recognition based on ST-GCN using opticalflow and image gradient, in: The 9th International Conference on Smart Media and Applications, 2020, pp. 207–213.
[206] S. Yenduri, V. Chalavadi, C.K. Mohan, STIP-GCN: Space-time interest points graph convolutional network for action recognition, in: 2022 International Joint Conference on Neural Networks, IJCNN, 2022, pp. 1–8.
[207] N. Wang, G. Zhu, H. Li, M. Feng, X. Zhao, L. Ni, P. Shen, L. Mei, L. Zhang, Exploring spatio–temporal graph convolution for video-based human–object interaction recognition, IEEE Trans. Circuits Syst. Video Technol. 33 (10) (2023) 5814–5827.
[208] H.S. Koppula, R. Gupta, A. Saxena, Learning human activities and object affordances from RGB-D videos, Int. J. Robot. Res. 32 (8) (2013) 951–970.
[209] J. Materzynska, T. Xiao, R. Herzig, H. Xu, X. Wang, T. Darrell, Something-Else: Compositional action recognition with spatial-temporal interaction networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1049–1059.
[210] G.A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, A. Gupta, Hollywood in homes: Crowdsourcing data collection for activity understanding, in: Computer Vision–ECCV 2016, in: Lecture Notes in Computer Science, vol. 9905, Springer, 2016, pp. 510–526.
[211] S. Liu, X. Wang, R. Xiong, X. Fan, GCN-based multi-modality fusion network for action recognition, IEEE Trans. Multimed. (2024) 1–3.
[212] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common objects in context, in: Computer Vision–ECCV 2014, in: Lecture Notes in Computer Science, vol. 8693, Springer, 2014, pp. 740–755.
[213] F. Liu, C. Wang, Z. Tian, S. Du, W. Zeng, Advancing skeleton-based human behavior recognition: Multi-stream fusion spatiotemporal graph convolutional networks, Complex Intell. Syst. 11 (1) (2025) 94.
[214] J. Lee, M. Lee, D. Lee, S. Lee, Hierarchically decomposed graph convolutional networks for skeleton-based action recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 10444–10453.
[215] H. Wu, X. Ma, Y. Li, Spatiotemporal multimodal learning with 3D CNNs for video action recognition, IEEE Trans. Circuits Syst. Video Technol. 32 (3) (2021) 1250–1261.
[216] K. Yun, J. Honorio, D. Chattopadhyay, T.L. Berg, D. Samaras, Two-person interaction detection using body-pose features and multiple instance learning, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 28–35.
[217] C. Chen, R. Jafari, N. Kehtarnavaz, UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor, in: 2015 IEEE International Conference on Image Processing, ICIP, 2015, pp. 168–172.
[218] J. Chen, C.M. Ho, MM-ViT: Multi-modal video transformer for compressed video action recognition, in: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV, 2022, pp. 1910–1921.
[219] M.B. Shaikh, D. Chai, S.M.S. Islam, N. Akhtar, Multimodal fusion for audio-image and video action recognition, Neural Comput. Appl. 36 (10) (2024) 5499–5513.
[220] R. Zhang, X. Yan, Video-language graph convolutional network for human action recognition, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2024, pp. 7995–7999.
[221] X. Chen, X. Liu, K. Liu, W. Liu, T. Mei, A baseline framework for part-level action parsing and action recognition, 2021, arXiv preprint arXiv:2110.03368.
[222] C. Sun, A. Myers, C. Vondrick, K. Murphy, C. Schmid, VideoBERT: A joint model for video and language representation learning, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV, 2019, pp. 7464–7473.
[223] B. Li, J. Chen, D. Zhang, X. Bao, D. Huang, Representation learning for compressed video action recognition via attentive cross-modal interaction with motion enhancement, 2022, arXiv preprint arXiv:2205.03569.
[224] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2D pose estimation using part affinity fields, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7291–7299.
[225] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, C. Lu, AlphaPose: Whole-body regional multi-person pose estimation and tracking in real-time, IEEE Trans. Pattern Anal. Mach. Intell. 45 (6) (2022) 7157–7173.
[226] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, C. Theobalt, VNect: Real-time 3D human pose estimation with a single RGB camera, ACM Trans. Graph. 36 (4) (2017) 1–14.
[227] K. Alomar, H.I. Aysel, X. Cai, RNNs, CNNs and transformers in human action recognition: A survey and a hybrid model, 2024, arXiv preprint arXiv:2407.06162.
[228] K.T. Le, H.H. Pham, N.A. Bui, W.N. Lie, N.D. Bui, A review on skeleton-based early action recognition, in: Proceedings of the International Conference on Intelligent Systems and Networks: ICISN 2024, in: Lecture Notes in Networks and Systems, vol. 1077, Springer, 2024, pp. 355–364.
[229] M.-T. Truong, V.-D. Hoang, T.-M.-C. Le, Skeleton-based posture estimation for human action recognition using deep learning, in: Computational Intelligence Methods for Green Technology and Sustainable Development: GTSD 2024, in: Lecture Notes in Networks and Systems, vol. 1195, Springer, 2024, pp. 85–98.
[230] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M.G. Yong, J. Lee, et al., MediaPipe: A framework for building perception pipelines, 2019, arXiv preprint arXiv:1906.08172.
[231] R. Varghese, M. Sambath, YOLOv8: A novel object detection algorithm with enhanced performance and robustness, in: 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems, ADICS, 2024, pp. 1–6.
[232] W. Zheng, P. Jing, Q. Xu, Action recognition based on spatial temporal graph convolutional networks, in: Proceedings of the 3rd International Conference on Computer Science and Application Engineering, 2019, pp. 1–5.
[233] M.S. Alsawadi, M. Rio, Human action recognition using BlazePose skeleton on spatial temporal graph convolutional neural networks, in: 2022 9th International Conference on Information Technology, Computer, and Electrical Engineering, ICITACEE, 2022, pp. 206–211.
[234] V. Bazarevsky, BlazePose: On-device real-time body pose tracking, 2020, arXiv preprint arXiv:2006.10204.
[235] Y. Yang, J. Zhang, J. Zhang, Z. Tu, Expressive keypoints for skeleton-based action recognition via skeleton transformation, 2024, arXiv preprint arXiv:2406.18011.
[236] N.u.R. Malik, S.A.R. Abu-Bakar, U.U. Sheikh, A. Channa, N. Popescu, Cascading pose features with CNN-LSTM for multiview human action recognition, Signals 4 (1) (2023) 40–55.
[237] W. Li, Y. Wong, A.-A. Liu, Y. Li, Y.-T. Su, M. Kankanhalli, Multi-camera action dataset for cross-camera action recognition benchmarking, in: 2017 IEEE Winter Conference on Applications of Computer Vision, WACV, 2017, pp. 187–196.
[238] A. Sharma, R. Singh, ConvST-LSTM-Net: Convolutional spatiotemporal LSTM networks for skeleton-based human action recognition, Int. J. Multimed. Inf. Retr. 12 (2) (2023) 34.

[239] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez Martínez, C. Peñafort-Asturiano, UP-Fall detection dataset: A multimodal approach, Sensors 19 (9) (2019) 1988.
[240] H. Le, C.-K. Lu, C.-C. Hsu, S.-K. Huang, Skeleton-based human action recognition using LSTM and depthwise separable convolutional neural network, Appl. Intell. 55 (4) (2025) 1–21.
[241] R. Votel, N. Li, et al., Next-generation pose detection with movenet and tensorflow. js, TensorFlow Blog 4 (2021) 4.
[242] L. Seidenari, V. Varano, S. Berretti, A. Bimbo, P. Pala, Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013, pp. 479–485.
[243] W. Zhang, M. Zhu, K.G. Derpanis, From actemes to action: A strongly-supervised representation for detailed action understanding, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2248–2255.
[244] A. Bharathi, R. Sanku, M. Sridevi, S. Manusubramanian, S.K. Chandar, Real-time human action prediction using pose estimation with attention-based LSTM network, Signal Image Video Process. 18 (4) (2024) 3255–3264.
[245] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Berkeley MHAD: A comprehensive multimodal human action database, in: 2013 IEEE Workshop on Applications of Computer Vision, WACV, IEEE, 2013, pp. 53–60.
[246] D. Avola, M. Cascio, L. Cinque, G.L. Foresti, C. Massaroni, E. Rodolà, 2-D skeleton-based action recognition via two-branch stacked LSTM-RNNs, IEEE Trans. Multimed. 22 (10) (2019) 2481–2496.
[247] L. Xia, C.-C. Chen, J.K. Aggarwal, View invariant human action recognition using histograms of 3d joints, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 20–27.
[248] A.K. Verma, A. Soren, A.K. Shrivastav, S. Kumar, Fusion of transformer model and skeleton detection model for abnormal human activity detection with transfer learning, in: 2023 IEEE World Conference on Applied Intelligence and Computing, AIC, 2023, pp. 556–561.
[249] C. Shi, S. Liu, Human action recognition with transformer based on convolutional features, Intell. Decis. Technol. 18 (2) (2024) 881–896.
[250] S.K. Yadav, A. Agarwal, A. Kumar, K. Tiwari, H.M. Pandey, S.A. Akbar, YogNet: A two-stream network for realtime multiperson yoga action recognition and posture correction, Knowl.-Based Syst. 250 (2022) 109097.
[251] J. Shi, Y. Zhang, W. Wang, B. Xing, D. Hu, L. Chen, A novel two-stream transformer-based framework for multi-modality human action recognition, Appl. Sci. 13 (4) (2023) 2058.
[252] C.-J. Huang, M. Gochoo, T.-H. Tan, Two-stream architecture using RGB-based ConvNet and pose-based LSTM for video action recognition, in: 2023 15th International Conference on Innovations in Information Technology, IIT, 2023, pp. 127–131.
[253] S.U. Rehman, A.U. Yasin, E. Ul Haq, M. Ali, J. Kim, A. Mehmood, Enhancing human activity recognition through integrated multimodal analysis: A focus on RGB imaging, skeletal tracking, and pose estimation, Sensors 24 (14) (2024) 4646.
[254] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, J. Sun, RepVGG: Making VGG-style convnets great again, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 13733–13742.
[255] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2016) 1137–1149.
[256] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2012, pp. 1290–1297.
[257] J. Arunnehru, G. Chamundeeswari, S.P. Bharathi, Human action recognition using 3D convolutional neural networks with 3D motion cuboids in surveillance videos, Procedia Comput. Sci. 133 (2018) 471–477.
[258] Ş. Aktı, G.A. Tataroğlu, H.K. Ekenel, Vision-based fight detection from surveillance cameras, in: 2019 9th International Conference on Image Processing Theory, Tools and Applications, IPTA, 2019, pp. 1–6.
[259] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 1251–1258.
[260] A. Mihanpour, M.J. Rashti, S.E. Alavi, Human action recognition in video using DB-LSTM and ResNet, in: 2020 6th International Conference on Web Research, ICWR, 2020, pp. 133–138.
[261] M.A. Ali, A.J. Hussain, A.T. Sadiq, Deep learning algorithms for human fighting action recognition, Int. J. Online Biomed. Eng. 18 (2) (2022) 71.
[262] J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, 2018, arXiv preprint arXiv:1804.02767.
[263] T. Gopalakrishnan, N. Wason, R.J. Krishna, N. Krishnaraj, Comparative analysis of fine-tuning I3D and SlowFast networks for action recognition in surveillance videos, Eng. Proc. 59 (1) (2024) 203.
[264] A. Manaf, S. Singh, A novel hybridization model for human activity recognition using stacked parallel LSTMs with 2D-CNN for feature extraction, in: 2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT, 2021, pp. 1–7.
[265] P.-E. Martin, J. Benois-Pineau, R. Péteri, J. Morlier, Fine grained sport action recognition with Twin spatio-temporal convolutional neural networks: Application to table tennis, Multimedia Tools Appl. 79 (2020) 20429–20447.
[266] L. Hacker, F. Bartels, P.-E. Martin, Fine-grained action detection with RGB and pose information using two stream convolutional networks, 2023, arXiv preprint arXiv:2302.02755.
[267] R. Sanford, S. Gorji, L.G. Hafemann, B. Pourbabaee, M. Javan, Group activity detection from trajectory and video data in soccer, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2020, pp. 3932–3940.
[268] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.
[269] S. Das, R. Dai, M. Koperski, L. Minciullo, L. Garattoni, F. Bremond, G. Francesca, Toyota smarthome: Real-world activities of daily living, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV, 2019, pp. 833–842.
[270] C. Su, J. Wei, D. Lin, L. Kong, Y.L. Guan, A novel model for fall detection and action recognition combined lightweight 3D-CNN and convolutional LSTM networks, Pattern Anal. Appl. 27 (1) (2024) 3.
[271] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, J. Rousseau, Multiple Cameras Fall Dataset, Tech. Rep 1350, DIRO- Université de Montréal, 2010, p. 24.
[272] B. Kwolek, M. Kepski, Human fall detection on embedded platform using depth maps and wireless accelerometer, Comput. Methods Programs Biomed. 117 (3) (2014) 489–501.
[273] Y. Xing, C. Lv, H. Wang, D. Cao, E. Velenis, F.-Y. Wang, Driver activity recognition for intelligent vehicles: A deep learning approach, IEEE Trans. Veh. Technol. 68 (6) (2019) 5379–5390.
[274] Q. Xiong, J. Zhang, P. Wang, D. Liu, R.X. Gao, Transferable two-stream convolutional neural network for human action recognition, J. Manuf. Syst. 56 (2020) 605–614.
[275] S. Li, J. Fan, P. Zheng, L. Wang, Transfer learning-enabled action recognition for human-robot collaborative assembly, Procedia CIRP 104 (2021) 1795–1800.
[276] Z. Li, R.C.-H. Yeow, PoseAction: Action recognition for patients in the ward using deep learning approaches, 2023, arXiv preprint arXiv:2310.03288.
[277] J. Tang, J. Xia, X. Mu, B. Pang, C. Lu, Asynchronous interaction aggregation for action detection, in: Computer Vision–ECCV 2020, in: Lecture Notes in Computer Science, vol. 12360, Springer, 2020, pp. 71–87.
[278] G. Sarapata, Y. Dushin, G. Morinan, J. Ong, S. Budhdeo, B. Kainz, J. O'Keeffe, Video-based activity recognition for automated motor assessment of Parkinson's disease, IEEE J. Biomed. Heal. Inform. (2023).
[279] J. Cheng, Z. Li, Gesture recognition for human-computer interaction based on CNN model, in: 2021 International Conference on Intelligent Computing, Automation and Applications, ICAA, 2021, pp. 241–244.
[280] M.A. Rahim, A.S.M. Miah, H.S. Akash, J. Shin, M.I. Hossain, M.N. Hossain, An advanced deep learning based three-stream hybrid model for dynamic hand gesture recognition, 2024, arXiv preprint arXiv:2408.08035.
[281] B. Sun, K. Zhao, Y. Xiao, J. He, L. Yu, Y. Wu, H. Yan, BNU-LCSAD: A video database for classroom student action recognition, in: Optoelectronic Imaging and Multimedia Technology VI, Vol. 11187, SPIE, 2019, pp. 417–424.
[282] G.A. Tadesse, O. Bent, K. Weldemariam, M. Istiak, T. Hasan, A. Cavallaro, et al., BON: An extended public domain dataset for human activity recognition, 2022, arXiv preprint arXiv:2209.05077.
[283] J. Chung, C.-h. Wuu, H.-r. Yang, Y.-W. Tai, C.-K. Tang, HAA500: Human-centric atomic action dataset with curated videos, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 13465–13474.
[284] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, A. Zisserman, A short note about kinetics-600, 2018, arXiv preprint arXiv:1808.01340.
[285] J. Carreira, E. Noland, C. Hillier, A. Zisserman, A short note on the Kinetics-700 human action dataset, 2019, arXiv preprint arXiv:1907.06987.
[286] M. Bamorovat Abadi, M.R. Shahabian Alashti, P. Holthaus, C. Menon, F. Amirabdollahian, RHM: Robot house multi-view human activity recognition dataset, in: ACHI 2023: The 16th International Conference on Advances in Computer-Human Interactions, 2023, pp. 159–166.
[287] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.
[288] A. Jisi, S. Yin, A new feature fusion network for student behavior recognition in education, J. Appl. Sci. Eng. 24 (2) (2021) 133–140.
[289] J. Jia, J. Song, Q. Hu, S. Tang, S. Xu, TAR: A dataset of teacher-teaching action recognition, in: 2023 8th International Conference on Image, Vision and Computing, ICIVC, 2023, pp. 676–681.
[290] V. Sharma, M. Gupta, A. Kumar, D. Mishra, STAR-3D: A holistic approach for human activity recognition in the classroom environment, Information 15 (4) (2024) 179.
[291] W. Xiang, C. Li, K. Li, B. Wang, X.-s. Hua, L. Zhang, CDAD: A common daily action dataset with collected hard negative samples, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2022, pp. 3921–3930.
[292] Y. Qian, Y. Sun, A. Kargarandehkordi, P. Azizian, O.C. Mutlu, S. Surabhi, Z. Jabbar, D.P. Wall, P. Washington, Advancing human action recognition with foundation models trained on unlabeled public videos, 2024, arXiv preprint arXiv:2402.08875.

[293] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, Y. Qiao, VideoMAE V2: Scaling video masked autoencoders with dual masking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 14549–14560.

[294] C. Liu, Y. Hu, Y. Li, S. Song, J. Liu, PKU-MMD: A large scale benchmark for continuous multi-modal human action understanding, 2017, arXiv preprint arXiv:1703.07475.

[295] J. Wang, X. Nie, Y. Xia, Y. Wu, S.-C. Zhu, Cross-view action modeling, learning and recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2014, pp. 2649–2656.

[296] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, R. Sukthankar, Violence detection in video using computer vision techniques, in: Computer Analysis of Images and Patterns: CAIP 2011, in: Lecture Notes in Computer Science, vol. 6855, Springer, 2011, pp. 332–339.

[297] T. Hassner, Y. Itcher, O. Kliper-Gross, Violent flows: Real-time detection of violent crowd behavior, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 1–6.