APSIPA Transactions on Signal and Information Processing, 2023, 12, e12

Original Paper

Deep Learning for Human Action Recognition: A Comprehensive Review

Duc-Quang Vu[1,2], Trang Phung Thi Thu[3], Ngan Le[4] and Jia-Ching Wang[1]*

[1] *Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan*
[2] *Thai Nguyen University of Education, Thai Nguyen, Vietnam*
[3] *Thai Nguyen University, Thai Nguyen, Vietnam*
[4] *Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, USA*

ABSTRACT

Over the past several years, we have witnessed remarkable progress in numerous computer vision applications, particularly in human activity analysis. Human action recognition, which aims to automatically examine and recognize the actions taking place in the video, has been widely applied in many applications. This paper presents a comprehensive survey of approaches and techniques in deep learning-based human activity analysis. First, we introduce the problem definition in action recognition together with its challenges. Second, we provide a comprehensive survey of feature representation methods. Third, we categorize human activity methodologies and discuss their advantages and limitations. In particular, we divide human action recognition into three main categories according to training mechanisms, i.e., supervised learning, semi-supervised learning, and self-supervised learning. We further analyze the existing network architectures, their performance, and source code availability for each main category. Fourth, we provide

*Corresponding author: Jia-Ching Wang, jcw@csie.ncu.edu.tw. This work is supported in part by the Thai Nguyen University of Education under Grant TNUE-2022-02.

Received 23 September 2022; Revised 20 December 2022
ISSN 2048-7703; DOI 10.1561/116.00000068

a detailed analysis of the existing, publicly available datasets, including small-scale and large-scale datasets for human action recognition. Finally, we discuss some open issues and future research directions.

Keywords: Action recognition, supervised learning, self-supervised learning, deep learning, deep neural networks.

1 Introduction

Human action recognition, which aims to automatically examine and recognize the actions taking place in video, has been widely applied in many applications such as identity recognition [81], video surveillance, environmental home monitoring [2], human-machine interfaces [83], etc. Human action recognition covers many main computer vision topics, including human detection in video, human pose estimation, human tracking, and temporal data analysis. Human activity in the real-world consists of simple limb movements to joint complex movements of multiple limbs and the entire human body. Every human action has a certain purpose; therefore, we can understand the action and purpose of the person taking action through the human visual system. However, using human labor to observe human actions in various real-world situations is too expensive, even impossible. So, human action recognition is one of the most fundamental research problems in computer vision and machine learning. It has been studied for decades and is widely used in many applications. Therefore, accurate and efficient human action recognition remains a challenging research area in computer vision. This is due to their prevalence in normal life, and recognized actions can be used for many other tasks such as security surveillance, abnormalities detection, video retrieval, etc. The goal of action recognition is to identify many different actions from different data types. In the early days, most methods focused on using RGB or optical-flow videos as input for action recognition. This is due to their popularity and easy access. In recent years, many works have been proposed using other data modalities such as skeleton, depth, audio, acceleration, etc. That depends on the application scenarios and the distinct advantages of different data types for action recognition.

There are many subtasks in human action recognition. For example, action classification (classifying action from predefined categories), action detection (determining the starting and end positions of actions), and action prediction (predicting the future state of actions). However, the major difference between action classification and action prediction lies in when to make a decision. Specifically, action classification is to predict the action label after observing the entire action execution. This task aims to focus on non-urgent scenarios, such as video retrieval, entertainment, etc. The study of this paper focuses on action classification. This is one of the most fundamental research problems in machine learning and computer vision and has attracted many researchers in recent years. Many deep learning models have been built to solve this problem, with various architectures like Conv2D network [50], Conv3D network [104], and LSTM combined with Conv2D [50, 94]. In addition, some models used more than one network (two streams) with two inputs to increase the model's learning ability. For example, the input is an image, and in Simonyan and Zisserman [94], and in Joao and Andrew [48] the input is an RGB video clip, and an optical follow clip, etc.

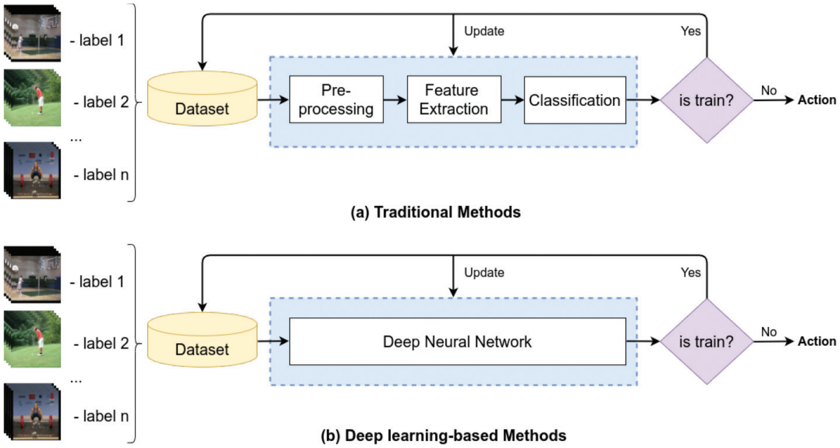
This work differs from several existing surveys for action recognition. For example, [14] provided a review for human activity recognition based on sensors such as accelerometer, gyroscope, magnetometer, electrocardiography, etc. Sun *et al.* [99] used the approach based on data modalities to present the review for action recognition such as RGB modality, skeleton modality, depth modality, infrared modality, point cloud, event stream, etc, and [99] also surveyed the action recognition problem via each stage such as preprocessing technics, models building & training. Our objective in this paper is to discuss state-of-the-art action recognition methods, especially with the modern deep neural network (DNN) approaches. In this work, we summarize many recent works and present a new survey of research on human action recognition techniques. We divide the human action recognition techniques into three groups based on training mechanisms, i.e., supervised learning semi-supervised learning, and self-supervised learning. For each group, we discuss network architectures, their advantages and limitations, and their performance. We further provide the recent datasets that have been commonly used to evaluate action recognition performance.

2 Human Action Recognition: Problem Definition and Challenges

2.1 Problem Definition

The goal of action recognition is to identify different actions from given videos (a sequence of 2D frames) where the action may or may not be performed throughout the entire duration of the video. The videos can be in RGB data, depth data, optical flow data, skeleton data, etc. However, essentially a video has spatial and temporal aspects. The spatial aspect is the individual frames that contain the context of the video, and the temporal aspect is the ordering of the frames, which contain the motion of the objects in the video. Sometimes, with just one frame, we can easily identify the action from the given video (e.g., applying lipstick, playing guitar, etc.). However, with more complex actions (e.g., walking vs. running, high jumping vs. long jumping, etc.), we require more than one frame to identify it correctly. Therefore, information on the temporal aspect is essential to differentiate between actions. Moreover, we sometimes need long-duration temporal information, or even whole frames from the video to correctly identify the action.

The action recognition problem in videos can be described as the following: Given a set of N samples of the form $\{(X_i, y_i)\}_{i=1}^{N}$. In which, X_i is a video clip where $X_i = (x_1, x_2, \ldots, x_T)$ being the input of length T with $x_j \in \mathbb{R}^{H \times W \times C}$ represents the jth frame. H, W and C are the height, width, and channel numbers, respectively. y_i is its corresponding label. We train a deep neural network $\mathcal{F}(X_i|\theta)$ by predicting y_i, and θ is the set of trainable parameters. An overview of the action recognition system is shown in Figure 1. The traditional system usually contains three steps that include pre-processing, feature extraction and classification (see Figure 1(a)). However, there are several limitations to the traditional methods. First, these systems are built based on many different components, e.g., pre-processing, feature extraction, and classification, so the performance of these systems depends on the performance of each component and the relationship among components. Second, the next component's input is from the previous one's output, so it is tough to train the model in parallel. Finally, the cohesion of independent components often does not perform well compared to end-to-end models.



(a) Traditional Methods

(b) Deep learning-based Methods

Figure 1: An overview diagram of the action recognition system. There are two phases in this system, including the training phase and the testing phase. Training phase aims to learn a recognition model that is able to distinguish various human actions defined in the training dataset. Testing phase utilizes the trained recognition model to recognize an action given a video.

To address these above limitations of the traditional systems, deep learning-based methods have been proposed as an end-to-end trainable model. Moreover, these deep models can train in parallel on GPUs via several python libraries such as Tensorflow, Pytorch, Keras, MXNet, etc. As a result, deep learning-based methods have quickly become state-of-the-art techniques for machine learning and artificial intelligence. An action recognition system based on deep learning is illustrated in Figure 1(b).

2.2 Challenges

In this section, we list some of the difficulties in deep learning-based action recognition. At least one of the challenges outlined below can significantly affect the performance of the whole recognition model.

Lack of Long-range Temporal Information. The frame rate of a video, denoted by FPS (frame per second), is the number of frames appearing per second of one video. Frame rate refers to how fast successive images make a video movement. The frame rates for a normal video are in the 25–30 FPS. Hence, a 10-second video has around 250-300 frames. However, we cannot put all frames of a video into a model for training. Instead, we select a small part, including continuous frames (e.g. 16 frames), to represent the entire video. This is also suitable for recognition systems in real-time. There is an issue here of whether a video clip with 16 frames is good enough to represent the entire video? There is an issue here of whether a video clip with 16 frames is good enough to represent the entire video. For example, with the "long jump" action performed by a human, we can see that the human performs various continuous sub-actions such as running, jumping, and landing. Therefore, if we choose the first 16 frames in the video, then the network model may confuse the "running" action. This is a huge challenge for deep models during the training because, in many cases, the actions appear only at a certain point in the video instead of always being repeated over and over again in the entire video (see Figure 2(a)). A simple solution for this challenge is to calculate averaging predictions over sampled clips. However, the long-range temporal information was still missing in learned features.

Computational Cost. Computational cost and complexity of spatio-temporal inputs are the main challenges in video understanding. With the skip connection technique in [35], ResNet has avoided the vanishing gradient problem without sacrificing network performance. Specifically, it helps upper layers in the network achieve features not worse than the lower layers. Moreover, with this architecture, the upper layers get more information directly from the lower layers, so they will adjust the weight more effectively. After the ResNet architecture, many variations of networks were introduced. Experiments show these CNN models with a depth of up to thousands of layers. ResNet has quickly become the most popular architecture in computer vision. However, the models with thousands of layers mean the computational cost in the network

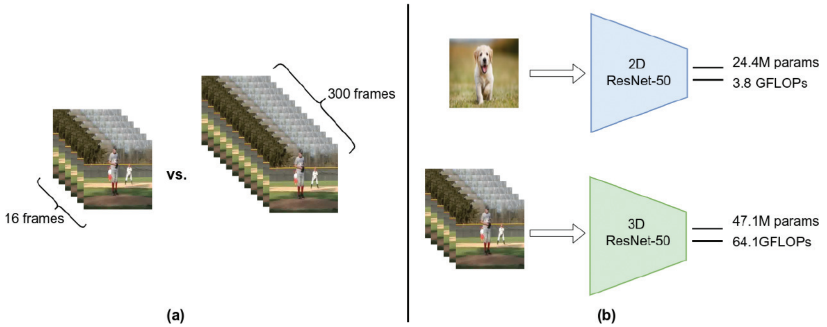


Figure 2: The two main challenges in the action recognition problem. (a) is the lack of temporal information when a clip with few frames represents the entire video. (b) are the comparison of the number of parameters and computational cost between the 2D ResNet-50 network usually applied for images and the 3D ResNet-50 network usually adopted for videos?

is huge. Besides, video input is larger than image input (due to videos having extra time dimension), so Conv3D often has more parameters and computational costs than Conv2D. For example, ResNet2D-50 [35] with the input of $(224, 224, 3)$ has 24.3M parameters and 3.8 GFLOPs, but ResNet3D-50 [34] with the input of $(16, 224, 224, 3)$ needs 46.9M parameters and 64.1 GFLOPs (in Figure 2(b)). The $R(2+1)D$-152 in [30] uses 118M parameters and 252 GFLOPs. As a result, model size and computational cost are one of the biggest challenges when we train a deep neural network for action recognition. For example, to train a 3D CNN model on the UCF101 dataset, we need three to four days and at least two months to train on the Sports-1M dataset with the same network architecture.

Except for these above challenges, the performance of an action recognition system is also affected by several common challenges of this task such as environmental conditions, video quality, camera motion, etc. [43].

2.3 Data Modalities

In the early days, most of the video understanding research focused on using RGB or gray-scale videos because of their popularity and easy access. Recent years have witnessed the use of other data modalities, such as infrared, point cloud, event stream, skeleton, depth, radar, etc. as follows:

RGB/Grayscale Videos: RGB or grayscale videos, providing rich appearance information, are the most popular data type. It has been used in most computer vision tasks. However, it is captured in a daytime environment and sensitive to viewpoint together with illumination.

Infrared: Infrared is a common data type for night-time environments; however, it lacks color and texture information.

Depth: While RGB provides rich appearance information, depth provides geometric shape information. A combination of both RGB and depth has been widely used in videos analysis recently.

Point Cloud: Point cloud includes both RGB and depth data and captures the 3D structure and distance information. This data kind is robust to viewpoint and has been popularly used in robot navigation and autonomous driving applications. However, this data is high complexity and sparse.

Event Stream: Event stream is specific data that contains both different changing and RGB. It is acquired by event cameras when object moving with high speed. Although this data kind is high-range dynamic and motion blur free, it is sparse and its capturing devices are expensive.

Skeleton: Skeleton data is defined on body joint, thus providing structural information of subject pose. Even though it does not provide any texture or shape information, it is robust to viewpoint and background.

3 Background

Based on the learning paradigm, we split action recognition approaches into two groups corresponding to traditional methods and modern methods.

3.1 Traditional Methods

The traditional methods are based on efficient spatio-temporal feature representations and motion propagation across frames in videos such as HOG3D [54], SIFT3D [91], ESURF [116], MBH [18], iDTs [111]. STIP-based [13, 61] is one of the most common methods widely used for action recognition. STIP methods extend the local feature detection technology from images to the 3D spatio-temporal domain. The main advantage of spatio-temporal-based methods is that they do not require preprocessing such as background segmentation or human detection. However, the features are sensitive to changes in camera views. To eliminate the background motion and overcome differences in the viewing angle, iDTs [111] uses key points or the joints in the human skeleton to represent actions. However, this approach requires an accurate human skeleton model, and accurate tracking of key points which are challenging problems in computer vision. These traditional features are mainly used in classic machine learning methods such as Boost, support vector machines, and probability map models to recognize the action.

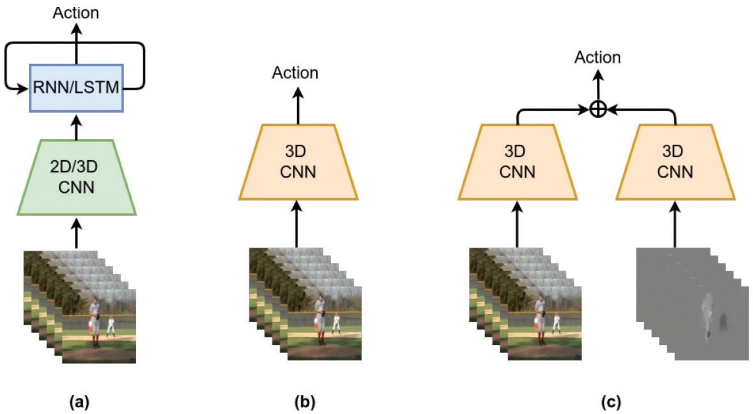


Figure 3: Comparison between three different deep network architectures for action recognition. (a): Recurrent Neural Networks (e.g., LSTM); (b): convolutional networks Networks (e.g., 3D CNN); (c): two-stream convolutional networks (e.g., RGB - optical flow).

3.2 *Modern Methods*

In recent years, DNNs have been successfully applied to computer vision. Various DNN-based feature extractions have been proposed to address human action recognition. Depending on network architecture, DNN-based feature representations can be Recurrent Neural Networks [23], 3D Convolutional Networks Networks (3D-CNNs) [44], two-stream convolutional networks [94], etc.

Recurrent Neural Networks (RNNs): RNNs with Long Short-Term Memory (LSTM) implementation are believed to cope with sequential information better, and thus many proposed methods [23, 79] attempted to incorporate LSTM to deal with action recognition. This approach aims to utilize the networks that have high performance in image classification to extract features from independent frames. And then, add a recurrent layer such as an LSTM to capture temporal ordering. Finally, a fully connected layer is added on top for the model to classify (see Figure 3(a)). However, [23] concluded that the LSTM is not as effective as the temporal pooling with feature maps from convolution layers.

Convolutional Networks Networks (CNNs): 3D CNN was first introduced by [44] to extract features from both the spatial and the temporal dimensions by performing 3D convolutions, thereby capturing the motion information encoded in multiple adjacent frames. Later on, C3D features, 3D CNN architectures and their improvements [22, 34, 48, 85, 104, 107] have been proposed. Recently, various 3D CNN-based networks have been proposed to address this task and they are also demonstrated to be more efficient than

LSTM networks [34, 48]. An overview of the 3D CNN architecture is shown in Figure 3(b).

Two-stream Network: The two-stream network was first introduced by [94], and then they have been improved in [27]. Two-stream networks explore video appearance and motion clues with two separate networks. One network exploits spatial information from individual frames, while the other uses temporal information from optical flow. The two outputs of the network are then combined via a latent fusion layer (see Figure 3(c)). With this approach, we can significantly boost the performance of CNN models compared to one-stream CNN i.e. conventional CNNs. However, this approach also has several drawbacks. For example, the input of the spatial network is usually an individual frame; therefore, it suffers from the problem of false label assignment. Each frame's ground truth is assumed the same as the video's ground truth, which may not be the case if the action happens for a small duration within the entire video. Besides, training with a two-stream network requires a lot of training time compared to a stand-alone network. Various two-stream approaches have been proposed such as RGB - OF [48] or RGB - RGB [16] or RGB - Audio [5, 52], etc.

4 Action Recognition Techniques

DNNs are typically trained under a supervised learning framework where a model learns a single task using labeled data. Instead of relying solely on labeled data, one can make use of unlabeled or related data to improve model performance, which is often more accessible and ubiquitous. In this section, we divide human action recognition techniques into three categories based on the training paradigm including supervised learning, semi-supervised learning, and self-supervised learning. Specifically, Section 4.1 presents the detail of state-of-the-art supervised learning-based methods. The approaches based on semi-supervised are discussed in Section 4.2. Next, the self-supervised-based methods are introduced in Section 4.3. Finally, we survey several other approaches for action recognition such as knowledge distillation in Section 4.4.

4.1 Supervised Learning

Supervised learning is a common machine learning technique to construct a function from training data. The training data usually consists of pairs of an input object (i.e., image, text, speech, etc.) and the ground truth output (i.e., label, image, vector, etc.). The supervised learning-based methods aim to predict a valid input object's value after considering some training examples (i.e., input and output pairs corresponding). To this end, methods have to generalize from sample data to predict the unresolved situations in a "reasonable" way. Figure 4 shows the visual introduction to the supervised learning strategy on action recognition.

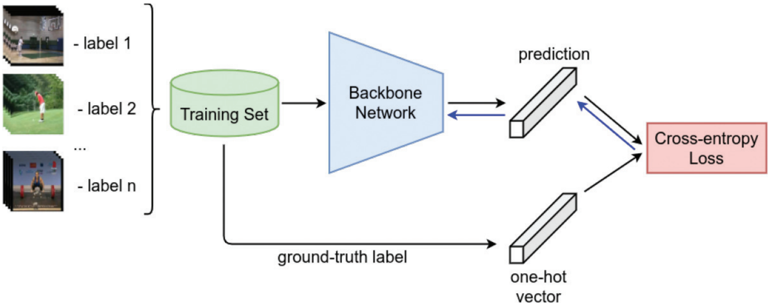


Figure 4: Overview of the supervised learning methods for action recognition. In which, the black line denotes the forward path and the blue line is the backward i.e., the backpropagation step.

In the action recognition problem, given a set of N samples of the form $(X_1, y_1), (X_2, y_2), \ldots, (X_N, y_N)$ such that X_i is a video clip and y_i is its label (i.e., class). A model is built to learn a function F that maps $F : X_i \to y_i$. The output of the F is corresponding to the probability distribution p_i over labels, where $p_i = F(X_i, \theta')$ and θ' is the set of trainable parameters. The correctness of the prediction was measured using cross-entropy as follows:

$$\mathcal{L}(y_i, p_i) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(p_i). \tag{1}$$

Following the success of C3D, I3D networks, many new model architectures have been proposed in recent years. The objective of the proposed models is to achieve better performance and/or reduce computational costs. As mentioned above, the computational cost is one of the biggest challenges in the action recognition task; therefore, many researchers focus on reducing the computational cost task for 3D CNNs. A new method was proposed to explicitly factorize 3D convolution into two separate and successive operations, a 2D spatial convolution and a 1D temporal convolution, and called $(2+1)$D convolution [85, 106, 108]. The authors demonstrated that $(2+1)$D convolution has more advantages than 3D convolution on action recognition. Inspired by group convolution [58] and channel separation [37], [105] present a model architecture named irCSN-152 and ipCSN-152 that based on ResNet-152. In the ip-CSN (interaction-preserved channel-separated network), a $3 \times 3 \times 3$ convolution is replaced by a $1 \times 1 \times 1$ traditional convolution and a $3 \times$

3×3 depthwise convolution. This block reduces significantly the number of parameters and FLOPs compared to the traditional $3 \times 3 \times 3$ convolution. In the ir-CSN network (interaction-reduced channel-separated network), the authors remove the extra $1 \times 1 \times 1$ convolution. This yields the depthwise bottleneck block. The experiments show that both ir-CSN and the ip-CSN outperform deep 3D ResNet while significantly reducing parameters and FLOPs. Several other methods are proposed to reduce the FLOPs, such as in [126], [60], etc.

Achieving state-of-the-art performance is the most important task in action recognition. Most methods adopt the ResNet backbone as a standard architecture to modify. For example, the STM network was introduced by [45]. In this network, the authors presented a method to enhance the ability to learn Spatio-temporal and motion features from a video. To do that, the authors proposed encoding these two features in a unified 2D framework. The Channel-wise spatio-temporal Module (CSTM) is used to learn spatio-temporal features and the Channel-wise Motion Module (CMM) is for encoding motion features. These two modules are added to the original residual blocks in the ResNet architecture. The experiment shows that STM performs a little better than major proposed 3D CNN and 2D CNN-based methods. Inspired by the evolutionary algorithms in the optimization field, such as the genetic algorithm, [4] proposed a new method for finding video CNN architectures. In their work, a novel evolutionary search algorithm is developed to automatically explores different types of models and combine layers based on mutation operations. And then, they obtain new architectures superior to manually designed architectures. There are three mutation operations proposed in this paper, including "Change Layer", "Change Temporal Size" and "Add Layer". However, the crossover operator is not mentioned in this paper. To find good architectures with state-of-the-art performance, the authors built a population with 2000 different CNN architectures. Each newly generated child architecture (from their parent) is trained for 1000 iterations.

The SlowFast network [16] is proposed as a variation of the 3D CNN networks category. Two parallel pathways are utilized to capture a video scene's appearances and object motion in each pathway. Instead of using two streams (one stream is RGB, and the other is optical flow), the SlowFast network utilizes RGB for all streams. A slow pathway operates at a low frame rate, capturing spatial semantics, and a fast pathway captures motion at fine temporal resolution at a high frame rate. SlowFast network has been proposed to tackle the action recognition and action spatial localization tasks and got the highest scores in many benchmark datasets, e.g., Kinetics, Charades, AVA, etc. Recently, various Transformer-based models have been proposed for computer vision tasks such as A-ViT [120], Action Transformer [75], etc. With their flexible attention mechanism, transformer-based models have achieved impressive performance on many data types and they have quickly become a promising approach in recent years.

Unlike the aforementioned methods, knowledge distillation approaches aim to learn a lightweight network, i.e., a student, such that it can mimic the behaviors of the heavy network i.e., a teacher with high performance. With useful information from the teacher, the student can learn more efficiently and be more "intelligent". Inspired by this motivation, one of the first knowledge distillation works was introduced by [7] suggesting minimizing the ℓ_2 distance between the last layers of these two networks. Hinton *et al.* [36] later pointed out that the hidden relationships between the teacher's predicted class probabilities are also significant and informative for the student. Then, the soft labels generated by the teacher model are adopted as the supervision signal in addition to the regular labeled training data during the training phase. In addition to the soft labels as in [36]. Romero *et al.* [89] proposed the bridge among the middle layers of the student and teacher networks and adopted the ℓ_2 loss to supervise the output of the student further. Several other aspects and knowledge of the teacher network are also exploited later.

Diba *et al.* [19] proposed a new model named Temporal 3D ConvNet (T3D). In this model, 3D dense blocks and Temporal Transition Layers (TTL) are arranged alternately. The TTL layers use kernels with different sizes for temporal dimensions to increase the ability to learn temporal features. Additionally, the T3D model uses knowledge transferred from a pre-trained 2D ConvNet (DenseNet-169) on ImageNet. Like T3D, [20] proposed Spatio-Temporal Channel Correlation (STC) model based on ResNet architecture, and the authors also used the teacher models are 2D ResNet and ResNext pre-trained on ImageNet. The main contribution of this method is to propose STC blocks alternating 3D Residual blocks. The STC block behaves similarly to the squeeze-and-excitation block in [38].

Crasto *et al.* [17] proposed a new approach named MARS. The authors found that most state-of-the-art methods consist of a two-stream architecture with 3D convolutions for action recognition. However, the cost of computing optical flow and the cost of two-stream is huge. Therefore, it increases action recognition latency. The authors introduced two learning approaches. The first approach is Motion-Emulated RGB Stream (MERS). In MERS, a 3D teacher network takes optical flow as input, and the other 3D CNN with RGB input is the student network. The training phase is done in two steps. In step 1, the authors train the teacher to classify actions using optical flow clips and freeze the network's weights. The distillation progress from the teacher to the student was performed using the MSE function through all the layers of the student. In step 2, all the student layers have frozen their weights, and the last layer is added to the top of the network for training with a cross-entropy loss. The second approach is Motion Augmented RGB Stream (MARS). This approach is nearly the same as MERS. In step 1, the authors also train the teacher to classify actions using optical flow clips and freeze the network's weights. However, in step 2, to effectively leverage both appearance and motion information, the authors combine the standard cross-entropy loss and MSE loss and backpropagate through all the network layers. The problem of precomputing optical flow has still a problem at large. To avoid flow computation at the test phase, their main contribution is the knowledge distillation from the flow stream (as the teacher) to the RGB stream (as the student). The experiments show that this approach outperforms RGB or Flow alone and preserves the performance of two-stream approaches.

Girdhar *et al.* [31] proposed a distillation model based on ResNet architecture. In their model, ResNet50 pre-trained on image datasets as "teachers" to train video models in a distillation framework without using labeled data. This is an interesting approach to learning spatio-temporal representations from unlabeled video data.

Self-knowledge distillation is a promising approach to replace the conventional knowledge distillation approach. There is no teacher network in self-knowledge distillation, therefore we can save a lot of training time due to without training the teacher network. Moreover, we also avoid the problem of the capacity gap between teacher and student networks [109]. For action recognition, various self-knowledge distillation methods have been proposed such as TY [109], SKD-SRL [110], SKD [25], etc.

We provide a summary of state-of-the-art methods based on supervised learning for action recognition in Table 1. For each method, we briefly describe its characteristics and performance in terms of accuracy.

4.2 Semi-supervised Learning

Semi-supervised learning is a kind of machine learning that uses both labeled and unlabeled data for training - typically a small amount of labeled data along with a large amount of unlabeled data. Semi-supervised learning is the combination of unsupervised learning (without any labeled data) and supervised (all data is labeled) (see Figure 5). Many researchers have found that unlabeled data, when used in conjunction with a bit of labeled data, can significantly improve the performance of the model. Besides, semi-supervised learning helps the models reduce the dependence on labeled datasets. Furthermore, unlabeled data can be collected automatically without human labor, so semi-supervised learning-based methods are always low-cost approaches. While various semi-supervised learning-based approaches in the image domain have been promising performances, the semi-supervised learning-based video domain is still quite novel.

Iosifidis *et al.* [42] introduced traditional Action Bank for action representation. The authors then proposed an extreme learning machine algorithm by combining geometric properties and discrimination criteria of the training data representation in the ELM space. Inspired by FixMatch [97] in the image domain, various SSL methods have recently been presented e.g., TCL [95], VideoSSL [46]. In TCL, Singh *et al.* [95] proposed to maximize the similarity among encoded representations of an input clip with different speeds and

Table 1: A summary of supervised learning methods for Action Recognition. The column "Performance" presents the top-1 accuracy of the best model in each method. The column "Model size" shows the number of parameters and FLOPs of each model. In case the authors didn't provide information about model size in their paper, we denote by —. Moments denotes the Moments in Time dataset and SS is the Something Something dataset.

Method	Description	Network	Model size	Performance	Code
BQN [39]	- Focusing on busy motion in the input videos. - Separating busy features from quiet features. - Two networks have been proposed for two features types.	BQN	92M 241GFLOPs	77.3 (Kinetics400) 97.6 (UCF101) 77.6 (HMDB51)	Link
STAM [92]	- Proposing two types of transformer including temporal transformer and spacial transformer.	Transformer	96M 270GFLOPs	79.3 (Kinetics400) 97.0 (UCF101) 39.7 (Charades)	Link
En-VidTr [124]	- Proposing two types of transformer including temporal transformer and spacial transformer.	VidTr-M	98.1M 220GFLOPs	79.7 (Kinetics400) 96.7 (UCF101) 74.4 (HMDB51)	None
Omni-sourced [24]	- Leveraging crawled data. - Adopt pre-trained models as a teacher. - Training students with teacher's labels.	irCSN-152	—	83.6 (Kinetics400) 96.0 (UCF101) 71.1 (HMDB51)	Link
G-Blend [114]	- Identifying causes for performance drop on multi-modal networks. - Proposing a technique to avoid overfitting on these networks.	ipCSN-152	32.8M 110.1GFLOPs	83.3 (Kinetics400)	Link
irCSN-152 [105]	- Design an architecture named Channel-Separated Convolutional Network. - Utilize Group convolution to offer computational savings.	irCSN-152	29.6M 96.7GFLOPs	82.6 (Kinetics400)	Link
ipCSN-152 [105]	- Design an architecture named Channel-Separated Convolutional Network. - Utilize Group convolution to offer computational savings.	ipCSN-152	32.8M 108.8GFLOPs	79.2 (Kinetics400)	Link

Table 1: Continued.

Method	**Description**	**Network**	**Model size**	**Performance**	**Code**
GB+DF+LB [73]	- Focusing on improving the last layers. - Propose 3 classification branches instead of using the global average pooling alone.	ResNet-152	—	53.4(SS V1) 78.8 (Kinetics400)	None
HATNet [21]	- Fusing 2D and 3D architectures into one. - Training on HVU dataset.	ResNet-50	—	77.6 (Kinetics400) 97.8 (UCF101) 76.5 (HMDB51)	None
CoST [63]	- Proposing a novel operation to learn features using 2D Conv with a weight-sharing constraint.	ResNet-101	—	31.5 (Moments) 77.5 (Kinetics400)	None
RNL-TSM [40]	- Present region-based non-local operations as a self-attention.	ResNet-50	35.95M 41.16GFLOPs	49.47 (SS V1) 77.2 (Kinetics400)	Link
MSNet [60]	- Learn correspondences across frames and convert them into motion features.	ResNet-50	49.2M 67.6GFLOPs	55.1 (SSV1) 67.1 (SS V2) 76.4 (Kinetics400) 77.4(HMDB51)	Link
CMA [15]	Propose a cross-modality attention operation.	ResNet-152	—	75.98 (Kinetics400) 96.5(UCF101)	None

FASTER32 [126]	- Leverages the video's redundancy to reduce FLOPs.
	- Combine an expensive model that captures actions, and a lightweight model that captures scene changes.
		ResNet-50	—
			67.7GFLOPs
				75.3 (Kinetics400)
				96.9(UCF101)
				75.7(HMDB51)
					None

MARS [17]	- Knowledge distillation from the flow network to the RGB network.	ResNeXt-101	—	74.9 (Kinetics400)
				53(SS V1)
				98.1(UCF101)
				80.9(HMDB51)
					None

STM [45]	- Encode features in a 2D framework.
	- The Channel-wise Spatio Temporal Module presents the spatio-temporal features.
	- The Channel-wise Motion Module efficiently encodes motion features.
		ResNet-50	23.88M
			32.93GFLOPs
				73.7 (Kinetics400)
				50.5(SS V1)
				64.2(SS V2)
				96.7(Jester)
				96.2(UCF101)
				72.2(HMDB51)
					None

Table 1: Continued.

Method	Description	Network	Model size	Performance	Code
SlowFastNet [16]	- Two streams with one a low frame rate and the other a high frame rate.	ResNet-101	— 234GFLOPs	79.8 (Kinetics400) 81.8(Kinetics600)	Link

EvaNet [4]	- Finding video CNN architectures based on an evolutionary algorithm.	Inception Net	—	77.4 (Kinetics400)
				82.3(HMDB51)
				31.8(Moments)
					None

R(2+1)D [106]	- Explicitly factorize 3D Conv into two operations, a 2D Conv and a 1D Conv.	ResNet-34	—	75.4 (Kinetics400)
				73.3(Sports1M)
				97.3(UCF101)
				78.7(HMDB51)
					Link

P3D [85]	- (2+1)D Conv uses ReLU between the 2D and 1D Conv in each block.
	- Using separate spatial and temporal components renders the optimization easier.
		ResNet-152	—	77.4 (Kinetics400)
				93.7(UCF101)
				66.4(Sports1M)
				75.12(ActivityNet)
				80.8(ASLAN)
					Link

I3D [48]	- Repeat 2D filters in the pre-trained Inception Net.	Inception-V1	25M
			—
				74.2 (Kinetics400)
				93.4(UCF101)
				66.4(HMDB51)
					Link

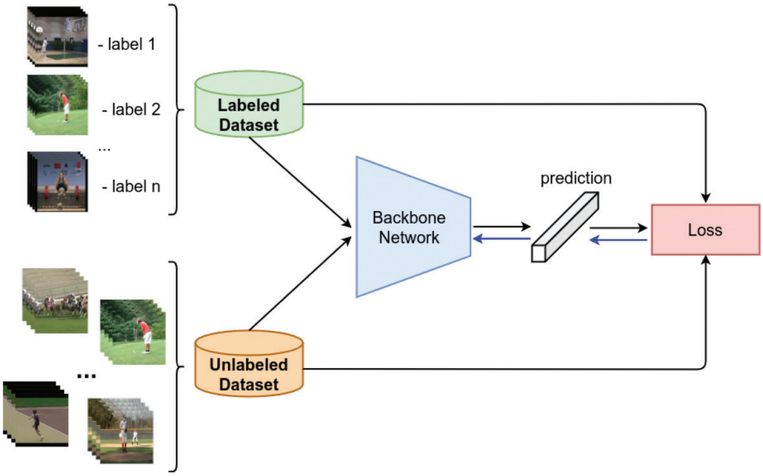


Figure 5: Overview of the semi-supervised learning methods for action recognition. In which, the black line denotes the forward path and the blue line is the backward i.e., backpropagation.

otherwise. Moreover, the authors introduced an efficient group-contrastive loss to distinguish a couple of motion representations with pace-invariance that extremely boosts action recognition performance. In VideoSSL, [46] utilized a pre-trained model on ImageNet to guide the training of the 3D CNN model via pseudo-labels of unlabeled examples.

We provide a summary of state-of-the-art methods based on semi-supervised learning for action recognition in Table 2. For each method, we briefly describe its characteristics and performance in terms of accuracy and the percentage of labeled data used for the training stage.

Table 2: A summary of semi-supervised learning methods for Action Recognition. The column "Performance" presents the top-1 accuracy of the best model in each method. The percent (%) after each dataset denotes the percent of labeled data used for training. * denotes that these methods were re-implement for video domain by [46].

Method	**Description**	**Network**	**Performance**	**Code**
VideoSSL [46]	Utilizing a pre-trained network on ImageNet to guide the training of the 3D CNN.	3D ResNet-18	47.6 (Kinetics100 - 5%) 32.4 (UCF101 - 5%) 32.7 (HMDB51 - 40%)	None
TCL [95]	Proposing two types of loss including Maximize Instance Agreement and Maximize Group Agreement.	TSM ResNet-18	29.81 (SS-V2 - 5%) 30.28 (Kinetics400 - 5%) 93.29 (Jester - 5%)	Link
FitMach* [97]	The pseudo-labels from weakly-augmented data are utilized to guide the training for a strongly-augmented version of the same data.	3D ResNet-18	40.5 (Kinetics100 - 5%) 27.1 (UCF101 - 5%) 32.9 (HMDB51 - 40%)	None
S4L* [121]	The combination of the self-supervised and semi-supervised learning method.	3D ResNet-18	33.0 (Kinetics100 - 5%) 22.7 (UCF101 - 5%) 29.8 (HMDB51 - 40%)	None
MT* [6]	Calculating the average of model weights over training steps that helps to generate a more robust model compared to using the final weights.	3D ResNet-18	27.8 (Kinetics100 - 5%) 17.5 (UCF101 - 5%) 27.2 (HMDB51 - 40%)	None
PL* [62]	The prediction from a sample is reused to guide itself.	3D ResNet-18	27.8 (Kinetics100 - 5%) 17.6 (UCF101 - 5%) 27.3 (HMDB51 - 40%)	None

4.3 Self-Supervised Learning

Unlike supervised learning, in self-supervised learning, most methods require a data pair x_i, z_i where z_i is automatically generated for a pre-defined pretext

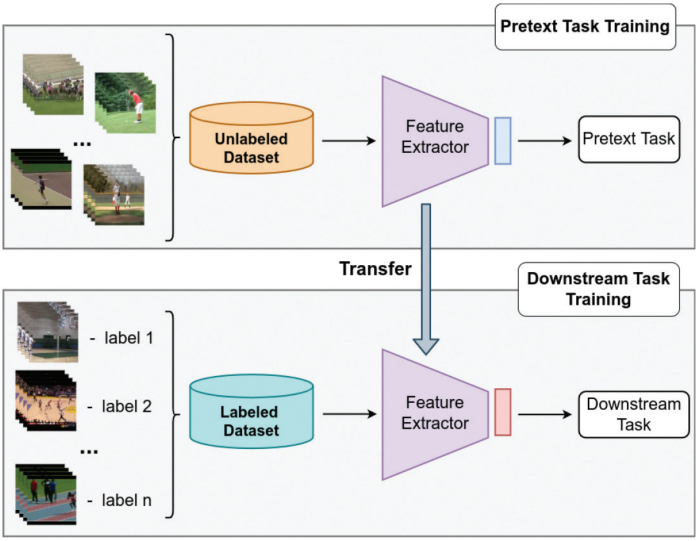


Figure 6: Overview of self-supervised learning-based methods for action recognition. There are two tasks in self-supervised learning including the pretext task and the downstream task. For the pretext task, the network is trained with pseudo-labels generated without human labor. For the downstream task, the network is transferred to address this task with labeled data.

task without involving any human annotation. Figure 6 shows an overview of self-supervised learning-based methods. In which, a deep network as a feature extractor is used to learn spatio-temporal features from the input video via pretext tasks. After the self-supervised training finished, the learned visual features can be further transferred to the downstream tasks i.e., the target task (in this case is action recognition).

There are many pretext tasks have proposed such as video rotation prediction [47], frame order verification [76], solving video jigsaw [3], video clip order prediction [118], motion and appearance statistics prediction [113], video playback rate perception [119], contrastive learning [84], etc.

Misra *et al.* [76] proposed a model that allows verifying temporal order. With the input being a tuple of frames' order, their model predicts whether the frames' order is correct or not. The authors proposed a ConvNet model to perform this pretext task. The objective of the model is not only to solve the temporal order verification task but also to learn spatio-temporal features from input videos. Fernando *et al.* [28] presented a self-supervised CNN called O3N based on odd-one-out learning. The input of the network is a tuple of related videos where one of the videos has the wrong temporal order while the other ones have the correct temporal order. The goal of O3N is to predict an odd video, i.e., the video with the wrong temporal order from these input videos.

A model based on deep reinforcement learning is introduced by [8]. The authors observed that there has been unused potential in self-supervision based on ordering. The diverse permutations will affect CNN differently. How can we find permutations that have higher utility for improving a CNN representation than the random set? The authors presented a reinforcement learning algorithm that helps to create permutations in the training phase. To learn the function for proposing permutations, the authors simultaneously train a policy and self-supervised network by utilizing the improvement over time of the CNN network as a reward signal.

Kim *et al.* [53] have shown ambiguity in time direction when we hardly distinguish between a “catch” or a “throw” action from given shuffled frames. The authors introduced a self-supervised task called Space-Time cubic puzzles. Given a randomly permuted sequence of 3D spatio-temporal pieces cropped from a video clip. The 3D CNN is used to learn both spatial and temporal relations from frames of the input video and predict their original arrangement. Through performing Space-Time cubic puzzles, the 3D CNN increase significantly the video representation and achieve state-of-the-art performance compare to other self-supervised methods of action recognition.

Far apart from the previous methods, [113] presented a self-supervised spatio-temporal representation learning for videos. Inspired by the success of two-stream approaches in video classification, based on regressing both motion and appearance statistics along spatial and temporal dimensions, the authors proposed to learn visual features from given only the input video data. A C3D network was introduced to learn visual features along spatial and temporal dimensions by predicting several numerical labels generated through the characteristics of video such as the region with the largest motion and its direction, the most diverged region in appearance and its dominant color, and the most stable region in appearance and its dominant color.

Caron *et al.* [10] found that a randomly initialized AlexNet achieves 12% in terms of accuracy on ImageNet while the chance is at 0.1%. This means that a randomly initialized network is intimately tied to its convolutional structure, which gives a strong prior to the input signal. The authors presented a new approach for self-supervised learning named Deep Cluster. Thus, we can use labels obtained from a randomly initialized network to kick-start the process, which can be refined later. Inspired by Deep Cluster, [5] propose a novel self-supervised method that leverages unsupervised clustering in one modality (e.g., audio) as a supervisory signal for the other modality (e.g., video). The authors presented three approaches for training video models from self-supervised audio-visual information including Multi-Head Deep Clustering (MDC), Concatenation Deep Clustering (CDC), and Cross-Modal Deep Clustering (XDC). For the two first methods, the pseudo-labels from the second modality are complemented by the pseudo-labels generated in the first modality. In the third approach i.e., XDC, the audio clusters drive the learning of the video representation and vice versa. The authors showed that XDC outperforms large-scale fully-supervised pretraining for action recognition on the same architecture.

Contrastive learning is an approach to formulate the task of finding similar and dissimilar things for a CNN model. Using this approach, we can train a deep neural network to classify between similar and dissimilar images or videos. Inspired by contrastive learning and the success of contrastive learning methods like SimCLR in the image field, [84] presented a self-supervised Contrastive Video Representation Learning (CVRL) method to learn spatio-temporal visual representations from unlabeled videos. The CVRL model is pre-trained on the Kinetics600 and the Kinetics400 datasets. The authors studied data augmentations involving spatial and temporal cues and proposed a spatial and temporal augmentation method to impose strong data augmentation for video. The experiments show that the CVRL achieves state-of-the-art performance on the downstream task and semi-supervised learning. Especially, the performance of CVRL achieves 72.6% in terms of accuracy approximates with supervised learning models. This significantly closes the gap between unsupervised and supervised video representation learning. Several other contrastive learning-based methods have been proposed by [78], [100], [101], etc. A list of the video feature self-supervised learning methods can be found in Table 3.

4.4 Other Approaches

Far apart from the aforementioned categories, various approaches have focused on action recognition via weakly supervised learning. Weakly supervised learning aims to train the models on huge volumes of samples. However, different from fully-supervised video datasets that are labeled by humans, the labels in datasets used in weakly supervised learning are usually generated from hashtags, and noise labels of social media without human fine-tuning. With this approach, our training dataset may be expanded to billion samples without incurring high expensive annotation costs. The drawback of this approach is the noise labels in many cases, not ground-truth labels, therefore it increases the confusion of the models during training on these weakly supervised datasets.

IG-65M [30] is one of the most popular weakly supervised datasets for action recognition. This dataset contains 65M videos collected from Instagram with many different hashtags. Ghadiyaram *et al.* [30] proposed using IG-65M to pre-training the networks and then these networks will be fine-turned on fully-supervised datasets such as Kinetics, Sports-1M, Epic-Kitchens, etc. The authors demonstrated that their approach has improved the state-of-the-art of these action recognition datasets compared to the independent training i.e., only training on fully-supervised datasets.

To recognize the fine-grained actions, [65] proposed the new method namely Hierarchical Atomic Action Network to conduct weakly-supervised fine-grained

Table 3: A summary of methods on Self-Supervised Learning for Action Recognition as the downstream task. We record the results on two standard datasets, including UCF101 and HMDB51. All results are the top-1 accuracy, which corresponds to backbone architectures in the column "Network."

Method	**Description**	**Network**	**Pre-training Dataset**	**Performance**	**Code**
VideoMAE [103]	Proposing data-efficient learning via video reconstruction using autoencoders as the pretext task.	ViT-L	Kinetics700	96.1 (UCF101) 61.1 (HMDB51)	Link
BraVe [87]	Training the network to learn features from a narrow view to the general content of the input clip.	TSM-50x2	Kinetics600	93.1 (UCF101) 77.8 (HMDB51)	Link
MCN [66]	Proposing multi-task process between contrastive learning and meta-learning.	3D ResNet-18	UCF101	84.8 (UCF101) 54.8 (HMDB51)	Link
CVRL [84]	Contrastive learning based on the SimCLR method.	ResNet-50	Kinetics400	92.1 (UCF101) 65.4 (HMDB51)	None
AVID+CMA [78]	Contrastive learning for cross-modal discrimination of video from audio and vice versa.	R2+1D-18	Audioset	91.5 (UCF101) 64.7 (HMDB51)	Link
XDC [5]	- Based on Deep Clustering. - Leverages unsupervised clustering in audio as a supervisory signal for video and vice versa. - The first self-supervised method outperforms large-scale fully-supervised pretraining.	R2+1D-18	Kinetics Audioset IG-65M	91.5 (UCF101) 63.1 (HMDB51)	None
PCL [101]	- Combine Pretext tasks with contrastive learning, referred to as Pretext-Contrastive Learning.	ResNet-18	UCF101	82.3 (UCF101) 43.2 (HMDB51)	None
PRP [119]	- Capture temporal resolution characteristics within the video domain in a self-supervised manner. - Introduce a motion attention mechanism to focus on meaningful foreground regions.	R2+1D-18	UCF101	72.1 (UCF101) 35.0 (HMDB51)	Link

Table 3: Continued.

Method	Description	Network	Pre-training Dataset	Performance	Code
DPC [33]	- Learning spatio-temporal features by recurrently predicting future representations. - Predicting further into the future with progressively less temporal context.	ResNet-34	Kinetics400	75.7 (UCF101) 35.7 (HMDB51)	Link
IIC [100]	- Uses positive-negative pairs to train with contrastive learning. - Different modalities of the same video are treated as positives and breaking temporal relations in the video or other videos are treated as negatives.	ResNet-18	UCF101	74.4 (UCF101) 38.8 (HMDB51)	Link
TCE [56]	- Encoding videos such that adjacent frames exist close to each other and videos are separated from one another.	ResNet-50	Kinetics400	71.2 (UCF101) 36.6 (HMDB51)	Link
VCP [67]	- Randomly choose one from 4 transformations or keeping the original. - Predict which is transform applied to the input clip.	ResNet-18	UCF101	66.0 (UCF101) 31.5 (HMDB51)	None
3D Cubic Puzzles [53]	- Ambiguity in time direction when hardly distinguishing between a “catch” or a “throw” action from shuffled frames. - Introducing a pretext task based on solving Space-Time Cubic Puzzles.	ResNet-18	Kinetics400	65.8 (UCF101) 33.7 (HMDB51)	None
Video Clip Ordering [118]	Learning the spatio-temporal representation of the video by predicting the order of shuffled clips from the video.	ResNet-18	UCF101	64.9 (UCF101) 29.5 (HMDB51)	None
Skip-Clip [26]	- Training a deep model for future clip order ranking based on a context clip.	ResNet-18	UCF101	64.4 (UCF101)	None

Table 3: Continued.

Method	**Description**	**Network**	**Pre-training Dataset**	**Performance**	**Code**
3D RotNet [47]	- A set of rotations are applied to all videos as a pretext task and a model is defined to predict these rotations.	ResNet-18	Kinetics400	62.9 (UCF101) 33.7 (HMDB51)	None
CMC [102]	- Presenting a set of sensory views of a video clip. - Based on contrastive learning, A model is built to maximize the mutual information between different views of the same scene.	CaffeNet	UCF101	59.1 (UCF101) 26.7 (HMDB51)	Link
M&A [113]	- Based on regressing both motion and appearance statistics along spatial and temporal dimensions. - Predicting several numerical labels generated through the characteristics of video such as the region with the largest motion and its direction, etc.	C3D	UCF101	58.8 (UCF101) 20.3 (HMDB51)	Link
Arrow of Time [115]	- Learning to see the arrow of time – to tell whether a video sequence is playing forward or backward. - Focusing on the motion cues in videos and using the arrow of time to pretrain action recognition models.	AlexNet	UCF101	55.3 (UCF101)	None
Cross & Learn [90]	- Information shared across modalities has a much higher semantic meaning compared to modality-specific information. - Present a self-supervised method for representation learning utilizing two different modalities (RGB and flow).	CaffeNet	UCF101	58.7 (UCF101) 27.2 (HMDB51)	Link

Table 3: Continued.

Method	Description	Network	Pre-training Dataset	Performance	Code
Geometry [29]	- Extracting pixel-wise geometry information as flow fields and disparity maps from synthetic imagery and real 3D movies. - Introducing a new type of auxiliary supervision based on exploring geometry.	CaffeNet	UCF101	55.1 (UCF101) 23.3 (HMDB51)	None

temporal action recognition. There are four hierarchy levels in their method including clip level, atomic action level, fine action class level, and coarse action class level. In which, the author proposed a self-supervised learning approach to discover visual concepts. After completing learned atomic actions by visual concepts, the authors further mapped to coarse and fine action labels via the semantic label hierarchy. The experiment results have shown that the Hierarchical Atomic Action Network achieved state-of-the-art performance on several standard datasets.

In fully-supervised learning methods, all the action classes are known a priori and available during both training and testing. However, these methods are not suitable for many real-world applications, where several action classes are not seen during training. Zero-shot learning (ZSL) has been proposed to address this issue. Specifically, ZSL aims to recognize videos in new classes that are unavailable during the training phase [70]. Generalized zero-shot learning (GZSL) introduced in [117] becomes harder than ZSL because the test videos can belong to the seen or unseen classes. To address the action recognition problem with GZSL, [70] introduced the out-of-distribution detector. Specifically, the authors split the problem into two partway separations i.e., seen and unseen action classes. The authors proposed an adversarial network that trained on seen action classes to classify videos in unseen action classes. Their approach has been conducted on several popular datasets and the results have shown that their method achieved state-of-the-art performance compared to other existing methods.

5 Datasets and Metrics

5.1 Datasets

There have more than 20 datasets that are used in action recognition. We provide a summary of these datasets and their characteristics in Tables 4

Table 4: A summary of common small-scale datasets from 2011 to now used for action recognition.

Dataset	**Description**	**#classes**	**Samples**	**Download**
HMDB51 [59]	- At least 1s / video. - Single activity / video.	51	6,849	Link
UCF50 [88]	- Realistic videos from Youtube. - Single activity / video.	50	6,676	Link
UCF101 [98]	- At least 1.06s/video. - Single activity / video.	101	13,320	Link
ActivityNet [9]	- Large-scale video. - 1.41 activity instance / video.	203	27,811	Link
Hollywood2 [72]	- 19.7s/video on average action videos and scene videos.	22	3,669	Link
MSR-Action3D [64]	An action dataset of depth sequences captured by a depth camera.	20	—	Link
MSR-Daily Activity 3D [112]	- A daily activity dataset captured by a Kinect device camera. - An activity is performed in either "sitting on sofa" or "standing" pose.	12	320	Link
ASLAN [55]	- Focus on action similarity.	432	3,697	Link
RGBD-HuDaAct [80]	- Synchronized color-depth video streams 30s-150s/video.	16	1,189	Link
Charades [93]	- Video action classification performance 6.8 actions/video.	157	9,848	Link

and 5. We categorize all datasets into two types. The first one is a group of small-scale datasets that include less than 100K sample videos in each dataset (see Table 4), and the other type is the group of large-scale datasets that includes greater than 100K videos (see Table 5).

Table 5: A summary of common large-scale datasets from 2011 to now used for action recognition.

Dataset	Description	#classes	Samples	Download
Kinetics400 [51]	- Last around 10s /video. - Single activity / video.	400	273K	Link
Kinetics600 [11]	- Last around 10s /video. - Single activity / video.	600	435K	Link
Kinetics700 [12]	- Last around 10s /video. - Single activity / video.	700	643K	Link
Kinetics700-2020 [96]	- Last 10s around /video. - Single activity / video.	700	648K	Link
Human3.6M Dataset [41]	- 3D human poses.	17	3.6M	Link
Sports-1M [50]	- Single action/video. - YouTube videos contain 6 different types of bowling, 7 different types of American football, and 23 types of billiards.	487	1.1M	Link
Youtube-8M [1]	- Provide precomputed and compressed features based on a Deep CNN pretrained on ImageNet.	3862	6.1M	Link
Something-Something [32]	- Video prediction tasks. - 6.8 actions/video.	174	220K	Link
HACS [125]	- 2-second clip annotations.	200	890K	Link
Moments in Time [77]	- 3s/video.	339	1M	Link
HVU-Dataset [21]	- Holistic video understanding (multi-label & multi-task video).	3,142	572K	Link

Table 5: Continued.

Dataset	**Description**	**#classes**	**Samples**	**Download**
Jester [74]	- 3s/video on average.	27	148K	Link
IG65M [49]	- Weakly supervised dataset.	400	65M	None
VideoLT [123]	- Large-scale long-tailed video recognition.	1,004	256K	Link

Small-scale Datasets: Most of the datasets are published before 2016 and in RGB format. Several datasets with other formats such as depth sequences captured by a depth camera in MSR-Action3D and RGBD-HuDaAct datasets. In small-scale datasets, the most common datasets are HMDB51, UCF50, UCF101, and ActivityNet. The HMDB51 dataset [59] is collected from various sources, mostly from movies and a small proportion from public databases such as the Prelinger archive, YouTube, and Google videos. The dataset contains 6,849 clips divided into 51 action categories, each containing a minimum of 101 clips. The actions categories can be grouped into five types: general facial actions (laugh, chew, talk, etc.), facial actions with object manipulation (smoke, eat, drink, etc.), general body movements (climb, backhand flip, handstand, jump, stand up, etc.), body movements with object interaction (kickball, ride a bike, shoot a gun, sword exercise, etc.), and body movements for human interaction (kiss, shake hands, punch, etc.). The Two-stream model by [94] has the best performance with 88% in terms of accuracy by using architectures of discriminatively trained ConvNets for action recognition in video.

The UCF101 [98] is an action recognition dataset, including 101 action categories. All videos from this dataset are real action videos, collected from YouTube. UCF101 gives diversity in terms of actions, with 13,320 videos containing large variations in camera motion, object appearance, pose, object scale, viewpoint, cluttered background, illumination conditions, etc. The videos in 101 action categories are grouped into 25 groups, where each group can consist of 4-7 videos of an action. The same group videos may share some common features, such as a similar background, similar viewpoints, etc. R2+1D-BERT [49] has been the best method for UCF101 with an average accuracy of up to 98.69%. In R2+1D-BERT, the authors combined 3D convolution with late temporal modeling for action recognition by replacing the conventional Temporal Global Average Pooling layer at the end of the 3D convolutional architecture with the Bidirectional Encoder Representations from Transformers (BERT) layer to better utilize the temporal information with BERT's attention mechanism.

The ActivityNet aims at covering a wide range of complex human activities that are of interest to people in their daily lives. In version 2015, ActivityNet provides samples from 203 activity classes with an average of 137 untrimmed videos per class and 1.41 activity instances per video for 849 video hours. Videos in ActivityNet are divided into five main groups, including eating and drinking activities (549 videos); sports, exercise, and recreation (3485 videos); socializing, relaxing, and leisure (1249 videos); personal care (844 videos); and household activities (1075 videos). The model ranked first in performance ActivityNet, is now W-TALC, a Weakly-supervised Temporal Activity Localization and Classification framework using only video-level labels, [82] with a map of 93.2. This method can detect fine granularity activities and achieve better performance than current state-of-the-art methods on ActivityNet.

Large-scale Datasets: Nowadays, social networks are increasingly popular, with millions of images and videos uploaded every day. Therefore, collecting images or videos from the internet isn't effortless. But building huge annotated datasets is extremely expensive in terms of time-consuming and labor-intensive. One of the most common large-scale datasets is Kinetics with three versions: Kinetics400 [51], Kinetics600 [11], and Kinetics700 [96]. The videos were temporally trimmed and lasted around 10 s and 200–1000 clips for each action. The total has 306,245 videos in Kinetics400 and 650,317 videos in Kinetics700. Currently, OmniSource irCSN-152 [24] is known as the best model for Kinetics400 with an accuracy of 83.6% is 1.7% better than that of irCSN-152 [30] whose performance keeps the second rank and is 21.34% higher than the least performance method by [17] for this dataset. Similarly, the LGD-3D Two-stream model [86] has given the best performance for Kinetics600 with the top-1 accuracy and top-5 accuracy of 82.7% and 96%, respectively. For Kinetics700, the best performing model on this dataset is I3D [12]. It gave an accuracy of over 81%.

Sport-1M is a large-scale annotation with 1,133,158 video URLs from Youtube covering 487 sports labels. Despite only holding the second rank in performing Kinetics400, irCSN-152 [30] has been the best method on Sport-1M; the ratios of video top-1 accuracy and video top-5 accuracy are 75.5% and 92.7%, respectively. Following the Sport-1M, YouTube-8M [1] has published and become the largest multi-label video classification dataset. There are composed of more than 6 million videos and 3800 classes in this dataset. Each class has at least 200 corresponding video examples, with an average of 3,552 training videos per class. To solve storage and computational bottlenecks problems, the authors provide pre-computed and compressed features based on a Deep CNN pre-trained on ImageNet to extract the hidden representation immediately before the classification layer. The DCGN, a deep convolutional graph neural network [71], gave the best performance on this dataset with 87.7% top-1 accuracy.

Something-Something [32] is another large-scale dataset with two versions. Something-Something v1 includes 108,499 videos, where the training set is 86,017 videos, the validation set is 11,522 videos, and the other is the test set without labels. In version two, the number of videos significantly increased, with 220,847 videos in total. All versions include 174 classes, defined as caption templates for videos. Whilst PAN ResNet101 model [122] has performed best on Something-Something V1 with 55.3 and 82.8 of top-1 and top-5 accuracy. Komkov *et al.* [57] has provided the mutual modality learning (MML) method for version 2 with the accuracy of 69.02% (top-1) and 92.7% (top-5). One large-scale dataset with weak labels is introduced by [49] named IG65M, which includes more than 65M videos from Instagram. To harness millions of public videos from Instagram, the authors adopted the associated hashtags as labels to train video classification models. Aside from those, Human3.6M Dataset [41], Jester [74], HVU-Dataset [21], HACS [125], Moments in Time [77] are the common large scale datasets that have been used in recent years.

5.2 Metrics

Action recognition is about predicting action classes from videos; hence the best performance is achieved when the disparities between the labels prediction and ground-truth labels are minimal. The most straightforward way of computing the disparity is to measure top-1 accuracy. Besides, model size and computational cost are also considered, especially when implementing the model on embedded or mobile devices with limited memory and speed. The standard evaluation metrics used for action recognition are presented in the following:

- **Accuracy:** Top-1 accuracy is a standard performance measure for multi-class classification in action recognition. This measure is calculated as the ratio between the number of correctly predicted scores per the total number of points in the test set.

- **#Params:** The number of parameters or model size is the total of parameters that are used in the model. This measure affects the ability to save models in memory. Typically, the larger the number of parameters, the more memory it takes.

- **Computational cost:** The complexity or computational cost, or the number of float-point operations (FLOPs) is a measure of multiply-adds in the model. It is an indirect metric and an approximation [69]. Typically, a deep learning model requires computation at millions of FLOPs (MFLOPs) or billions of FLOPs (GLOPs). This measurement is usually directly proportional to the running time.

- **Frame rate (FPS):** is the frequency (rate) at which consecutive images called frames are processed within 1 second and it is expressed by the number of frames per second. Typically, a video with a higher FPS keeps the motion smooth and the details crisp. In computer vision, FPS is used to measure processing speed. Far apart from FLOPs, which is the indirect metric of computation complexity, the frame rate is the direct metric that includes speed and other factors such as memory access cost and platform characteristics.

6 Discussion

Overall, most supervised learning-based methods toward becoming more and more deep and complex. However, the performance of these methods depends mainly on the availability of large-scale datasets. This is sometimes not suitable in the case of amounts of labeled data being very small. Moreover, labeling for a large annotation dataset usually takes extremely expensive in terms of time-consuming and labor-intensive. Various semi-supervised and/or self-supervised learning methods have been proposed recently to minimize dependence on large-annotation datasets and avoid these limitations. Because there are millions of images and videos uploaded every day. So, collecting these unlabeled data is very simple and much less expensive than annotation data. Through the above survey, we can see that semi-supervised and self-supervised learning-based approaches are the two main state-of-the-art strategies that are increasingly being improved. Besides, reducing the model's complexity is also a promising future approach, due to the final objective of an action recognition system is to deploy it in real-time on edge devices. Therefore, how to train a lightweight model that can run in real-time for action recognition on embedded or mobile devices is also a novel approach that has recently gained interest. To address this issue, several possible research directions may be of interest such as knowledge distillation, self-knowledge distillation, few-shot/zero-shot learning, contrastive learning, etc. For the problem of lack of long-range temporal information, various promising approaches have been proposed and improved such as using the slow pathway in SlowFastNet [16], combining other data types like audio, optical follow [48], pose estimation [68], etc.

7 Conclusion

This paper presents a survey of literature on deep learning approaches for action recognition. Although there have been many excellent studies on human action recognition, there are many challenges existing such as lack of long-range temporal information, computational cost, etc. In this work, we have reviewed human action recognition methods and provided comprehensive feature representation from hand-designed-based to deep neural network-based. As for the learning paradigm, we have reviewed three main strategies, including supervised learning, semi-supervised learning, and self-supervised learning together with the recent knowledge distillation. Besides the survey of new techniques, we have also provided a summarized the existing datasets at both large and small scales.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A Large-scale Video Classification Benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[2] J. K. Aggarwal and M. S. Ryoo, "Human Activity Analysis: A Review," *ACM Computing Surveys (CSUR)*, 43(3), 2011, 1–43.

[3] U. Ahsan, R. Madhok, and I. Essa, "Video Jigsaw: Unsupervised Learning of Spatiotemporal Context for Video Action Recognition," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2019, 179–89.

[4] P. AJ, A. Angelova, A. Toshev, and M. S. Ryoo, "Evolving Space-time Neural Architectures for Videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, 1793–802.

[5] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran, "Self-supervised Learning by Cross-Modal Audio-Video Clustering," *Advances in Neural Information Processing Systems*, 33, 2020.

[6] T. Antti *et al.*, "Mean Teachers are Better Role Models: Weight Averaged Consistency Targets Improve Semi-supervised Deep Learning Results," in *NeurIPS*, 2017, 1195–204.

[7] J. Ba and R. Caruana, "Do Deep Nets Really Need to be Deep?" In *NIPS*, 2013.

[8] U. Buchler, B. Brattoli, and B. Ommer, "Improving Spatio-temporal Self-supervision by Deep Reinforcement Learning," in *ECCV*, 2018, 770–86.

[9] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A Large-scale Video Benchmark for Human Activity Understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, 961–70.

[10] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep Clustering for Unsupervised Learning of Visual Features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, 132–49.

[11] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, “A Short Note about Kinetics-600,” *arXiv preprint arXiv:1808.01340*, 2018.

[12] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, “A Short Note on the Kinetics-700 Human Action Dataset,” *arXiv preprint arXiv:1907.06987*, 2019.

[13] B. Chakraborty, M. B. Holte, T. B. Moeslund, and J. Gonzàlez, “Selective Spatio-temporal Interest Points,” *Computer Vision and Image Understanding*, 116(3), 2012, 396–410.

[14] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, “Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges, and Opportunities,” *ACM Computing Surveys (CSUR)*, 54(4), 2021, 1–40.

[15] L. Chi, G. Tian, Y. Mu, and Q. Tian, “Two-stream Video Classification with Cross-modality Attention,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, 4511–20.

[16] F. Christoph, H. Fan, J. Malik, and K. He, “SlowFast Networks for Video Recognition,” in *The IEEE 2019 International Conference on Computer Vision (ICCV)*, IEEE, 2019, 6201–10.

[17] N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid, “Mars: Motion-Augmented RGB Stream for Action Recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, 7882–91.

[18] N. Dalal, B. Triggs, and C. Schmid, “Human Detection Using Oriented Histograms of Flow and Appearance,” in *ECCV*, Springer, 2006, 428–41.

[19] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. Van Gool, “Temporal 3D Convnets: New Architecture and Transfer Learning for Video Classification,” *arXiv preprint arXiv:1711.08200*, 2017.

[20] A. Diba, M. Fayyaz, V. Sharma, M. Mahdi Arzani, R. Yousefzadeh, J. Gall, and L. Van Gool, “Spatio-temporal Channel Correlation Networks for Action Classification,” in *ECCV*, 2018, 284–99.

[21] A. Diba, M. Fayyaz, V. Sharma, M. Paluri, J. Gall, R. Stiefelhagen, and L. Van Gool, “Holistic Large Scale Video Understanding,” *arXiv preprint arXiv:1904.11451*, 2019.

[22] A. Diba, V. Sharma, and L. V. Gool, “Deep Temporal Linear Encoding Networks,” in *The 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, 2329–38.

[23] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, “Long-Term Recurrent Convolutional Networks for Visual Recognition and Description,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4), 2017, 677–91.

[24] H. Duan, Y. Zhao, Y. Xiong, W. Liu, and D. Lin, “Omni-sourced Webly-supervised Learning for Video Recognition,” *arXiv preprint arXiv:2003.13042*, 2020.

[25] Q. V. Duc, T. Phung, M. Nguyen, B. Y. Nguyen, and T. H. Nguyen, “Self-knowledge Distillation: An Efficient Approach for Falling Detection,” in *International Conference on Artificial Intelligence and Big Data in Digital Era*, Springer, 2022, 369–80.

[26] A. El, S. Zhai, G. W. Taylor, and J. M. Susskind, “Skip-Clip: Self-Supervised Spatiotemporal Representation Learning by Future Clip Order Ranking,” *arXiv preprint arXiv:1910.12770*, 2019.

[27] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional Two-Stream Network Fusion for Video Action Recognition,” in *The 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, 1933–41.

[28] B. Fernando, H. Bilen, E. Gavves, and S. Gould, “Self-supervised Video Representation Learning with Odd-One-Out Networks,” in *CVPR*, 2017, 3636–45.

[29] C. Gan, B. Gong, K. Liu, H. Su, and L. J. Guibas, “Geometry Guided Convolutional Neural Networks for Self-supervised Video Representation Learning,” in *CVPR*, 2018, 5589–97.

[30] D. Ghadiyaram, D. Tran, and D. Mahajan, “Large-scale Weakly Supervised Pre-training for Video Action Recognition,” in *CVPR*, 2019, 12046–55.

[31] R. Girdhar, D. Tran, L. Torresani, and D. Ramanan, “Distinit: Learning Video Representations without a Single Labeled Video,” in *ICCV*, 2019, 852–61.

[32] R. Goyal, S. E. 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 *ICCV*, Vol. 1, No. 4, 2017, 5.

[33] T. Han, W. Xie, and A. Zisserman, “Video Representation Learning by Dense Predictive Coding,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, 1483–92.

[34] K. Hara, H. Kataoka, and Y. Satoh, “Can Spatio-temporal 3D CNNs Retrace the History of 2D CNNs and Imagenet?” In *CVPR*, 2018, 6546–55.

[35] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, 770–8.

[36] G. Hinton, O. Vinyals, and J. Dean, “Distilling the Knowledge in a Neural Network,” *arXiv preprint arXiv:1503.02531*, 2015.

[37] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.

[38] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, 7132–41.

[39] G. Huang and A. G. Bors, "Busy-Quiet Video Disentangling for Video Classification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, 1341–50.

[40] G. Huang and A. G. Bors, "Region-based Non-local Operation for Video Classification," *arXiv preprint arXiv:2007.09033*, 2020.

[41] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6m: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7), 2013, 1325–39.

[42] A. Iosifidis, A. Tefas, and I. Pitas, "Semi-supervised Classification of Human Actions based on Neural Networks," in *2014 22nd International Conference on Pattern Recognition*, IEEE, 2014, 1336–41.

[43] I. Jegham, A. B. Khalifa, I. Alouani, and M. A. Mahjoub, "Visionbased Human Action Recognition: An Overview and Real World Challenges," *Forensic Science International: Digital Investigation*, 32, 2020, 200901.

[44] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1), 2013, 221–31.

[45] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan, "STM: Spatiotemporal and Motion Encoding for Action Recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, 2000–9.

[46] L. Jing, T. Parag, Z. Wu, Y. Tian, and H. Wang, "Videossl: Semi-supervised Learning for Video Classification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, 1110–9.

[47] L. Jing, X. Yang, J. Liu, and Y. Tian, "Self-supervised Spatiotemporal Feature Learning via Video Rotation Prediction," *arXiv preprint arXiv:1811.11387*, 2018.

[48] C. Joao and Z. Andrew, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," in *CVPR*, 2017, 6299–308.

[49] M. Kalfaoglu, S. Kalkan, and A. A. Alatan, "Late Temporal Modeling in 3D CNN Architectures with Bert for Action Recognition," *arXiv preprint arXiv:2008.01232*, 2020.

[50] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks," in *CVPR*, 2014, 1725–32.

[51] 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," *arXiv preprint arXiv:1705.06950*, 2017.

[52] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "Epicfusion: Audio-Visual Temporal Binding for Egocentric Action Recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, 5492–501.

[53] D. Kim, D. Cho, and I. S. Kweon, "Self-supervised Video Representation Learning with Space-Time Cubic Puzzles," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 2019, 8545–52.

[54] A. Klaser, M. Marszałek, and C. Schmid, "A Spatio-temporal Descriptor based on 3D-gradients," in *BMVC 2008-19th British Machine Vision Conference*, British Machine Vision Association, 2008, 275–1.

[55] O. Kliper-Gross, T. Hassner, and L. Wolf, "The Action Similarity Labeling Challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3), 2011, 615–21.

[56] J. Knights, A. Vanderkop, D. Ward, O. Mackenzie-Ross, and P. Moghadam, "Temporally Coherent Embeddings for Self-Supervised Video Representation Learning," *arXiv preprint arXiv:2004.02753*, 2020.

[57] S. Komkov, M. Dzabraev, and A. Petiushko, "Mutual Modality Learning for Video Action Classification," *arXiv preprint arXiv:2011.02543*, 2020.

[58] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, 60(6), 2017, 84–90.

[59] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A Large Video Database for Human Motion Recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[60] H. Kwon, M. Kim, S. Kwak, and M. Cho, "MotionSqueeze: Neural Motion Feature Learning for Video Understanding," in *European Conference on Computer Vision*, Springer, 2020, 345–62.

[61] I. Laptev, "On Space-time Interest Points," *International Journal of Computer Vision*, 64(2-3), 2005, 107–23.

[62] D.-H. Lee *et al.*, "Pseudo-label: The Simple and Efficient Semisupervised Learning Method for Deep Neural Networks," in *Workshop on Challenges in Representation Learning, ICML*, Vol. 3, No. 2, 2013.

[63] C. Li, Q. Zhong, D. Xie, and S. Pu, "Collaborative Spatiotemporal Feature Learning for Video Action Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, 7872–81.

[64] W. Li, Z. Zhang, and Z. Liu, "Action Recognition Based on a Bag of 3D Points," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, IEEE, 2010, 9–14.

[65] Z. Li, L. He, and H. Xu, “Weakly-Supervised Temporal Action Detection for Fine-Grained Videos with Hierarchical Atomic Actions,” in *European Conference on Computer Vision*, Springer, 2022.

[66] Y. Lin, X. Guo, and Y. Lu, “Self-supervised Video Representation Learning with Meta-contrastive Network,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 8239–49.

[67] D. Luo, C. Liu, Y. Zhou, D. Yang, C. Ma, Q. Ye, and W. Wang, “Video Cloze Procedure for Self-supervised Spatio-Temporal Learning,” *arXiv preprint arXiv:2001.00294*, 2020.

[68] D. C. Luvizon, D. Picard, and H. Tabia, “2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, 5137–46.

[69] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, “Shufflenet v2: Practical Guidelines for Efficient CNN Architecture Design,” in *European conference on computer vision*, 2018, 116–31.

[70] D. Mandal, S. Narayan, S. K. Dwivedi, V. Gupta, S. Ahmed, F. S. Khan, and L. Shao, “Out-of-Distribution Detection for Generalized Zero-shot Action Recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 9985–93.

[71] F. Mao, X. Wu, H. Xue, and R. Zhang, “Hierarchical Video Frame Sequence Representation with Deep Convolutional Graph Network,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, 262–70.

[72] M. Marszalek, I. Laptev, and C. Schmid, “Actions in Context,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, 2929–36.

[73] B. Martinez, D. Modolo, Y. Xiong, and J. Tighe, “Action Recognition with Spatial-temporal Discriminative Filter Banks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, 5482–91.

[74] J. Materzynska, G. Berger, I. Bax, and R. Memisevic, “The Jester Dataset: A Large-scale Video Dataset of Human Gestures,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, 2874–82.

[75] V. Mazzia, S. Angarano, F. Salvetti, F. Angelini, and M. Chiaberge, “Action Transformer: A Self-attention Model for Short-time Pose-based Human Action Recognition,” *Pattern Recognition*, 124, 2022, 108487.

[76] I. Misra, C. L. Zitnick, and M. Hebert, “Shuffle and Learn: Unsupervised Learning using Temporal Order Verification,” in *ECCV*, Springer, 2016, 527–44.

[77] 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 Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 2019, 502–8.

[78] P. Morgado, N. Vasconcelos, and I. Misra, "Audio-visual Instance Discrimination with Cross-modal Agreement," *arXiv preprint arXiv:2004.12943*, 2020.

[79] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond Short Snippets: Deep Networks for Video Classification," in *Computer Vision and Pattern Recognition*, 2015.

[80] B. Ni, G. Wang, and P. Moulin, "RGBD-HuDaAct: A Color-depth Video Database for Human Daily Activity Recognition," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, IEEE, 2011, 1147–53.

[81] S. N. Paul and Y. J. Singh, "Survey on Video Analysis of Human Walking Motion," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 7(3), 2014, 99–122.

[82] S. Paul, S. Roy, and A. K. Roy-Chowdhury, "W-talc: Weakly supervised Temporal Activity Localization and Classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, 563–79.

[83] L. L. Presti and M. La Cascia, "3D Skeleton-based Human Action Classification: A Survey," *Pattern Recognition*, 53, 2016, 130–47.

[84] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui, "Spatiotemporal Contrastive Video Representation Learning," *arXiv preprint arXiv:2008.03800*, 2020.

[85] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 5533–41.

[86] Z. Qiu, T. Yao, C.-W. Ngo, X. Tian, and T. Mei, "Learning Spatio-temporal Representation with Pseudo-3d Residual Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, 12056–65.

[87] A. Recasens, P. Luc, J.-B. Alayrac, L. Wang, F. Strub, C. Tallec, M. Malinowski, V. Pătrăucean, F. Altché, M. Valko, *et al.*, "Broaden your Views for Self-supervised Video Learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 1255–65.

[88] K. K. Reddy and M. Shah, "Recognizing 50 Human Action Categories of Web Videos," *Machine Vision and Applications*, 24(5), 2013, 971–81.

[89] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for Thin Deep Nets," in *ICLR*, 2015.

[90] N. Sayed, B. Brattoli, and B. Ommer, “Cross and Learn: Cross-Modal Self-supervision,” in *Pattern Recognition*, Cham: Springer International Publishing, 2019, 228–43.

[91] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional Sift Descriptor and Its Application to Action Recognition,” in *Proceedings of the 15th ACM International Conference on Multimedia*, 2007, 357–60.

[92] G. Sharir, A. Noy, and L. Zelnik-Manor, “An Image is Worth 16 × 16 Words, What is a Video Worth?” *arXiv preprint arXiv:2103.13915*, 2021.

[93] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, “Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding,” in *European Conference on Computer Vision*, Springer, 2016, 510–26.

[94] K. Simonyan and A. Zisserman, “Two-stream Convolutional Networks for Action Recognition in Videos,” in *NIPS*, 2014, 568–76.

[95] A. Singh, O. Chakraborty, A. Varshney, R. Panda, R. Feris, K. Saenko, and A. Das, “Semi-Supervised Action Recognition with Temporal Contrastive Learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 10389–99.

[96] L. Smaira, J. Carreira, E. Noland, E. Clancy, A. Wu, and A. Zisserman, “A Short Note on the Kinetics-700-2020 Human Action Dataset,” *arXiv preprint arXiv:2010. 10864*, 2020.

[97] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, “FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence,” *Advances in Neural Information Processing Systems*, 33, 2020.

[98] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild,” *arXiv preprint arXiv:1212.0402*, 2012.

[99] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, “Human Action Recognition from Various Data Modalities: A Review,” *IEEE transactions on pattern analysis and machine intelligence*, 2022.

[100] L. Tao, X. Wang, and T. Yamasaki, “Self-supervised Video Representation Learning using Inter-Intra Contrastive Framework,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, 2193–201.

[101] L. Tao, X. Wang, and T. Yamasaki, “Self-Supervised Video Representation Using Pretext-Contrastive Learning,” *arXiv preprint arXiv:2010. 15464*, 2020.

[102] Y. Tian, D. Krishnan, and P. Isola, “Contrastive Multiview Coding,” *arXiv preprint arXiv:1906.05849*, 2019.

[103] Z. Tong, Y. Song, J. Wang, and L. Wang, “Videomae: Masked Autoencoders are Data-efficient Learners for Self-supervised Video Pretraining,” *arXiv preprint arXiv:2203.12602*, 2022.

[104] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning Spatiotemporal Features with 3D Convolutional Networks,” in *The 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, 4489–97.

[105] D. Tran, H. Wang, L. Torresani, and M. Feiszli, “Video Classification with Channel-separated Convolutional Networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, 5552–61.

[106] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A Closer Look at Spatiotemporal Convolutions for Action Recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, 6450–9.

[107] G. Varol, I. Laptev, and C. Schmid, “Long-Term Temporal Convolutions for Action Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6), 2018, 1510–7.

[108] D.-Q. Vu, N. Le, and J.-C. Wang, “(2+1)D Distilled ShuffleNet: A Lightweight Unsupervised Distillation Network for Human Action Recognition,” in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, 2022.

[109] D.-Q. Vu, N. Le, and J.-C. Wang, “Teaching Yourself: A Self Knowledge Distillation Approach to Action Recognition,” *IEEE Access*, 9, 2021, 105711–23.

[110] D.-Q. Vu, J.-C. Wang, *et al.*, “A Novel Self-knowledge Distillation Approach with Siamese Representation Learning for Action Recognition,” in *2021 International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, 2021, 1–5.

[111] H. Wang and C. Schmid, “Action Recognition with Improved Trajectories,” in *ICCV*, 2013, 3551–8.

[112] J. Wang, Z. Liu, Y. Wu, and J. Yuan, “Mining Action Let Ensemble for Action Recognition with Depth Cameras,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, 1290–7.

[113] J. Wang, J. Jiao, L. Bao, S. He, Y. Liu, and W. Liu, “Self-supervised Spatio-temporal Representation Learning for Videos by Predicting Motion and Appearance Statistics,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, 4006–15.

[114] W. Wang, D. Tran, and M. Feiszli, “What Makes Training Multi-Modal Classification Networks Hard?” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 12695–705.

[115] D. Wei, J. J. Lim, A. Zisserman, and W. T. Freeman, “Learning and Using the Arrow of Time,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, 8052–60.

[116] G. Willems, T. Tuytelaars, and L. Van Gool, “An Efficient Dense and Scale-invariant Spatio-temporal Interest Point Detector,” in *ECCV*, Springer, 2008, 650–63.

[117] Y. Xian, B. Schiele, and Z. Akata, “Zero-shot Learning-the Good, the Bad and the Ugly,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, 4582–91.

[118] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang, “Self-supervised spatiotemporal Learning via Video Clip Order Prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, 10334–43.

[119] Y. Yao, C. Liu, D. Luo, Y. Zhou, and Q. Ye, “Video Playback Rate Perception for Self-Supervised Spatio-Temporal Representation Learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 6548–57.

[120] H. Yin, A. Vahdat, J. M. Alvarez, A. Mallya, J. Kautz, and P. Molchanov, “A-ViT: Adaptive Tokens for Efficient Vision Transformer,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 10809–18.

[121] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, “S4l: Self-supervised semi-supervised Learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 1476–85.

[122] C. Zhang, Y. Zou, G. Chen, and L. Gan, “PAN: Towards Fast Action Recognition via Learning Persistence of Appearance,” *arXiv preprint arXiv:2008.03462*, 2020.

[123] X. Zhang, Z. Wu, Z. Weng, H. Fu, J. Chen, Y.-G. Jiang, and L. S. Davis, “Videolt: Large-scale Long-tailed Video Recognition,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 7960–9.

[124] Y. Zhang, X. Li, C. Liu, B. Shuai, Y. Zhu, B. Brattoli, H. Chen, I. Marsic, and J. Tighe, “Vidtr: Video Transformer without Convolutions,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 13577–87.

[125] H. Zhao, Z. Yan, L. Torresani, and A. Torralba, “HACS: Human Action Clips and Segments Dataset for Recognition and Temporal Localization,” *arXiv preprint arXiv:1712.09374*, 2019.

[126] L. Zhu, L. Sevilla-Lara, D. Tran, M. Feiszli, Y. Yang, and H. Wang, “Faster Recurrent Networks for Video Classification,” *arXiv preprint arXiv:1906.04226*, 2, 2019.