sensors

MDPI

Review

Convolutional Neural Networks or Vision Transformers: Who Will Win the Race for Action Recognitions in Visual Data?

Oumaima Moutik [1], Hiba Sekkat [1], Smail Tigani [1], Abdellah Chehri [2,*], Rachid Saadane [3], Taha Ait Tchakoucht [1] and Anand Paul [4]

[1] Engineering Unit, Euromed Research Center, Euro-Mediterranean University, Fes 30030, Morocco
[2] Department of Mathematics and Computer Science, Royal Military College of Canada, Kingston, ON 11 K7K 7B4, Canada
[3] SIRC-LaGeS, Hassania School of Public Works, Casablanca 8108, Morocco
[4] School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, Republic of Korea
* Correspondence: chehri@rmc.ca

Abstract: Understanding actions in videos remains a significant challenge in computer vision, which has been the subject of several pieces of research in the last decades. Convolutional neural networks (CNN) are a significant component of this topic and play a crucial role in the renown of Deep Learning. Inspired by the human vision system, CNN has been applied to visual data exploitation and has solved various challenges in various computer vision tasks and video/image analysis, including action recognition (AR). However, not long ago, along with the achievement of the transformer in natural language processing (NLP), it began to set new trends in vision tasks, which has created a discussion around whether the Vision Transformer models (ViT) will replace CNN in action recognition in video clips. This paper conducts this trending topic in detail, the study of CNN and Transformer for Action Recognition separately and a comparative study of the accuracy-complexity trade-off. Finally, based on the performance analysis's outcome, the question of whether CNN or Vision Transformers will win the race will be discussed.

Keywords: convolutional neural networks; vision transformers; recurrent neural networks; conversational systems; action recognition; natural language understanding; action recognitions

Citation: Moutik, O.; Sekkat, H.; Tigani, S.; Chehri, A.; Saadane, R.; Tchakoucht, T.A.; Paul, A. Convolutional Neural Networks or Vision Transformers: Who Will Win the Race for Action Recognitions in Visual Data? *Sensors* **2023**, *23*, 734. https://doi.org/10.3390/s23020734

Academic Editor: Loris Nanni

Received: 10 December 2022
Revised: 1 January 2023
Accepted: 4 January 2023
Published: 9 January 2023

1. Introduction

With the emergence of deep learning, computer vision (CV) pushed the limits of what was possible in the digital world [1–3]. Over recent years, problems that were assumed unsolvable are now being solved with super-human accuracy. The main reason for this success is the great diversity of the market and needs. New tasks such as medical imaging, Industry, Object Recognition [4], Autonomous Vehicle Navigation [5], Face Detection [6], Fingerprint Recognition [7], Fast Image Processing [8], and Robotic Navigation [9] have been tested at high accuracy. Furthermore, integrating artificial intelligence in image recognition is the subject of many uses.

NLP, or natural language processing, is a revolution in how computers and other technological devices are used. It is a processing system that translates human instructions into computer language and the other way around. As a result, the user interface is significantly more convenient and accessible. One of the most significant breakthroughs in the NLP field during 2022 has been creating machine learning models that create texts from scratch, with the GPT-3 (Generative Pre-Trained Transformer 3) [10] leading the way. The peculiarity of Transformers is that they can understand the context of words in a way that was not possible before. Remarkably, however, recent work demonstrated that Vision Transformers could also have equal or higher performance on large-scale image classification tasks [11].

Video understating like Action Recognition (AR), which we specifically deal with in this review, involves two famous techniques: Convolutional Neural Networks and Vision Transformers.

In point of fact, multiple different approaches to deep learning are still being utilized in several applications (Figure 1). Convolutional neural networks, on the other hand, have been shown to be the most successful model for dealing in computer vision with image and video data, and they are the ones that are used the most.

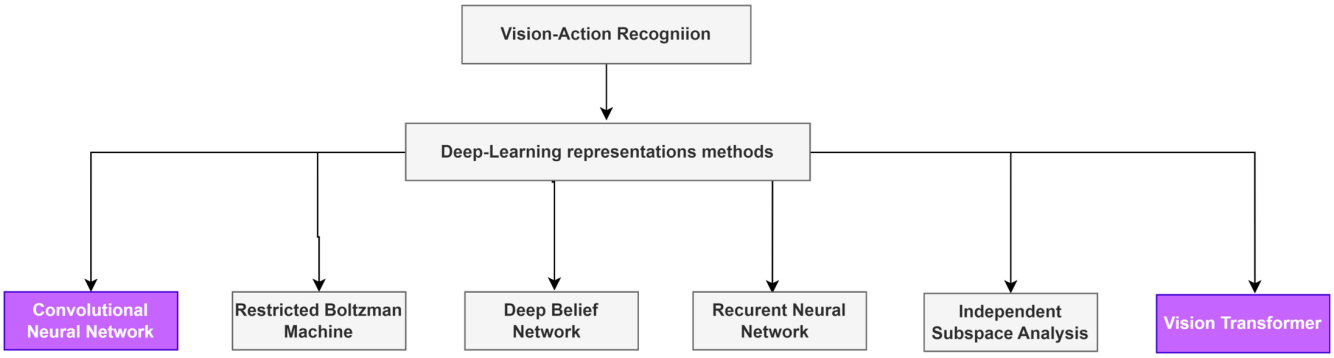

Figure 1. Different Deep Learning methods for Action Recognition.

During this time, Transformer is the trendiest. In light of these considerations, the purpose of this paper is to conduct an investigation of both of these techniques on a more in-depth level.

Action recognition is core work in video understanding [12], which refers to recognizing an action in a video based on the complete execution of the action, reconciling the characteristics of video image data to achieve high-level understanding. It has been studied for decades, solving many problems such as abnormal event detection [13], video retrieval [14], intelligent robots [15], and visual surveillance [16].

The next Section describes the mechanism of CNN and its evolution over time, Section 2 represents some of CNN's applications, and Action Recognition using CNN is summarized in Section 3. A description of Transformer models and architecture is given in Section 4. In the same way, as in Section 4, Action Recognition using this time transformer is given in Section 6. Section 7 compares these two models of the Deep Learning approach in terms of performance and complexity. Section 8 concludes the paper.

2. Convolutional Neural Networks, Review

Convolutional neural networks are inspired by the human visual system. The CNN model mimics the cortical area's structure by going back to 1962. Hubel et al. [17] introduced the hierarchical model of the visual system based on an experimental study. The study shows that the cortex has very tiny areas of cells that are sensitive to specific parts of the field of vision, a primary area that specifically detects dark and bright spots, as well as the edges of the visual scene, and a secondary area that interprets the visual information.

Inspired by this discovery, in 1980, the first version of the CNN was reported by Fukushima [18], which introduced a neural network model for a visual recognition technology to recognize patterns named Neocognitron. The network is comprised of two layers of cells linked, similar to the biological human visual system.

Stimulated by that, LeCun et al. [19], in 1989, gave a multi-layer network containing seven learned layers for handwriting digit recognition by applying a backpropagation method learning to handwrite digit images without the complex prepossessing stage. The architecture includes four convolutional layers, a pooling layer, and followed by three fully connected layers. The method revealed good results compared to the existing ones. However, as a consequence of the lack of trainable information and computing power, this architecture failed to work well under technical issues.

After that, the same authors introduced a new class of neural architecture in 1998 named the LeNet [20]. One of the most common structures of this class is the LeNet-5. This class has seven layers of neural network architecture, without inputs. It is composed of two alternate convolutions and pooling layers followed by three fully connected layers at the end. Convolutions to hold the spatial direction of features, the downsampling of the feature space is performed by the average pooling along with a sigmoid activation between layers.

At the time, it was assumed that a better algorithm would always give better results regardless of the data.

In 2012, Krizhevsky et al. [20] proved this theory wrong by announcing a deep CNN model named AlexNet for the image classification task. The model was record-breaking in the image recognition task. It successfully classifies 1.2 million high-res images from the Large-Scale Visual Recognition Challenge (LSVRC) challenge into 1000 different categories using purely supervised learning.

The most remarkable algorithm would only work correctly if the data learned represented the physical world—consequently, the project aimed to map the entire world of objects.

The architecture contains five convolutional layers, each one followed by max-pooling layers, and three fully connected layers with a final 1000-way Softmax. Due to the size of the network, the researchers applied data augmentation on the image data to expand the dataset using label-preserving transformations and dropout to reduce the overfitting problem. Figure 2 illustrates some of the famous architectures of CNN over time).

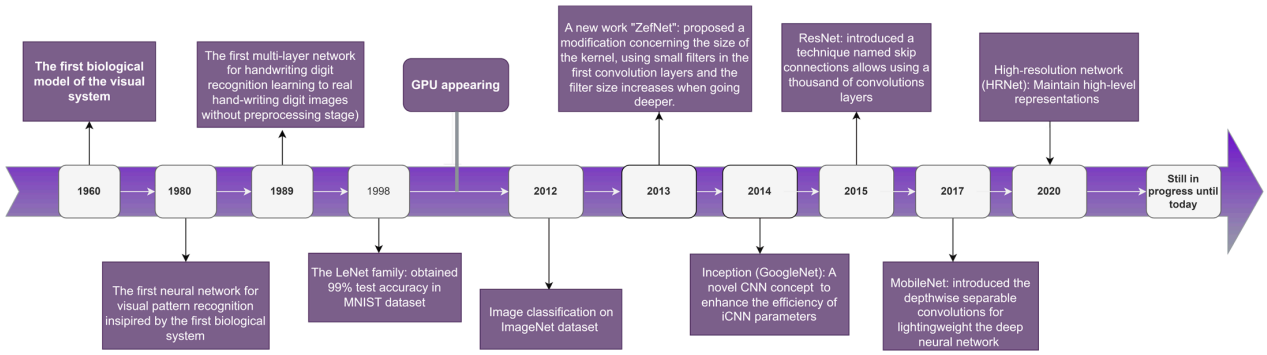

Figure 2. The evolution of CNN over time (Inception [21], ZefNet [22], ResNet [23], MobileNet [24], and HRNet [25]).

2.1. CNN Mechanism

CNN is a supervised deep learning model composed of a series of layers, each with specific functionality, which typically takes an image as input, then several hidden layers and an output.

CNN can handle complex images, unlike the multi-layer perceptron (MLP), which flattens image input, a 3×3 matrix of values separated by its three-color planes—red, green, and blue—into a vector and sends it to a multi-level perceptron. This causes a medium accuracy score when predicting classes but would have limited or no accuracy when it comes to complicated images with pixel dependencies everywhere or otherwise.

ConvNets capture an image's spatial and temporal dependencies, and the architecture better adapts to the image data set by minimizing the number of parameters implied and reusing the weights. Technically, the network is trained to understand image details better and scalable to massive data.

The CNN method consists of applying different hidden layers, respectively. It generally has three major neural layers: convolution layers, pooling layers, and fully connected ones. Each type of layer performs a specific role and converts the input volume into an output neural activation volume. The utilization of a convolution layer on an RGB (red, green, and blue) image is depicted in Figure 3.

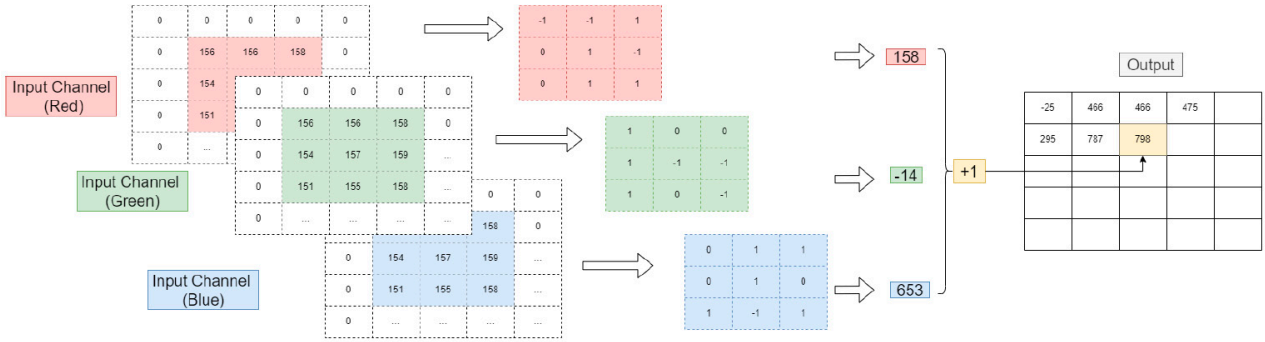

Figure 3. The application of convolution layer on RGB image.

2.2. Convolution Layer

The major component of CNN and where most of the computation is involved. It is the layer that takes advantage of the inherent properties of the image and the layer to apply to the input image. Its parameters are centered on using kernels/filters/feature detectors.

These learning filters are relatively tiny relative to the size of the image in the spatial dimension. Therefore, the size determines the characteristics that the filters can select. Usually, these filters in the first layers can only extract features that cover up to 0.24% of the screen.

The filters are square matrices that map onto the image data by applying the dot product to the subregion of the input data and obtaining the output as a dot product matrix; that is to say, a 2D activation map. For greater accuracy in image analysis, it is recommended to add padding to the image, which is a margin of zero values placed on the edge of the image, and a practical technique to develop the depth so that the output of the current convolutional layer does not evolve small in dimension.

As presented in GoogleNet, the convolution steps can be customized by specifying the number of pixels shifted on the input matrix. This number is in the stride. The following formula is used to determine the spatial dimension of the output of the convolution layers:

$$\frac{(\mathrm{V}-\mathrm{K})+2Z}{S+1} \tag{1}$$

where V is the input size (Input height + padding height top + padding height bottom), K represents the kernel size, Z is the quantity of zero padding set, and S is the stride.

2.3. Nonlinearity Layer

The nonlinearity Layer makes the precedent layer nonlinear by applying an activation function, like Rectified Linear Unit (ReLU), the Sigmoid function, or the hyperbolic tangent function in any of its layers or even in more than one layer.

ReLU function is simple and fast and helps the training phase to converge reliably by outputting the input directly y is positive. Otherwise, it will output 0 (Equation (2)), while the **SOFTMAX** activation function is typically applied at the end of the final layer to convert the network's output into a probability distribution. The formula is given in Equation (3):

$$\mathrm{Relu}(\mathrm{x}) = \mathrm{f}(\mathrm{x}) = \begin{cases} 0, & \text{for } y < 0 \\ y, & \text{for } y > 0 \end{cases} \tag{2}$$

$$\mathrm{Softmax}(\mathrm{z}) = \frac{e^{z_i}}{\sum_{j=0}^{K} e^{z_j}} \tag{3}$$

where z represents the input vector, e^{z_i} is the standard exponential function for the input, k is the number of classes and e^{z_j} refers to the standard exponential function for the output.

2.4. Pooling Layer

This layer takes the convolution layer's output and reduces the feature maps' dimensions by summarizing the subregions, i.e., taking the common of the maximum value. The purpose of using this layer is to optimize the complexity of the model procedure and control the overfitting problem. The most popular pooling operation is max pooling, which involves taking the maximal value of each sub-region to reduce the dimensional scale.

The max-pooling layer is a 2 × 2 kernel dimensional with a stride of 2 on the spatial size of the input while keeping 25% of the original dimension and the depth volume at its initial size. Another well-known pooling layer, average pooling, consists of calculating the average of each sub-region of the activation map from any 2 × 2 square in the activation map.

2.5. Fully Connected Layer

The objective of this layer is to flatter all the high-level features learned by all the convolution layers and mix all the elements for learning the non-linear combinations of the features. The layer is a feed-forward neural network that forms the last layer of the CNN network. Each neuron has connections to all neurons in the previous layer, and its connection has its weight. The activation is calculated by a matrix operation followed by a bias gap.

3. CNN-Related Work

In the past seven years, numerous studies and applications based on CNN have been developed in various domains, including healthcare [26] and autonomous vehicles [27].

Based on Web of Science databases, Figure 4 depicts the number of publications per year that mention CNN. Object detection is one of these applications, with the goal of identifying the object with a bounding box and determining its classification.

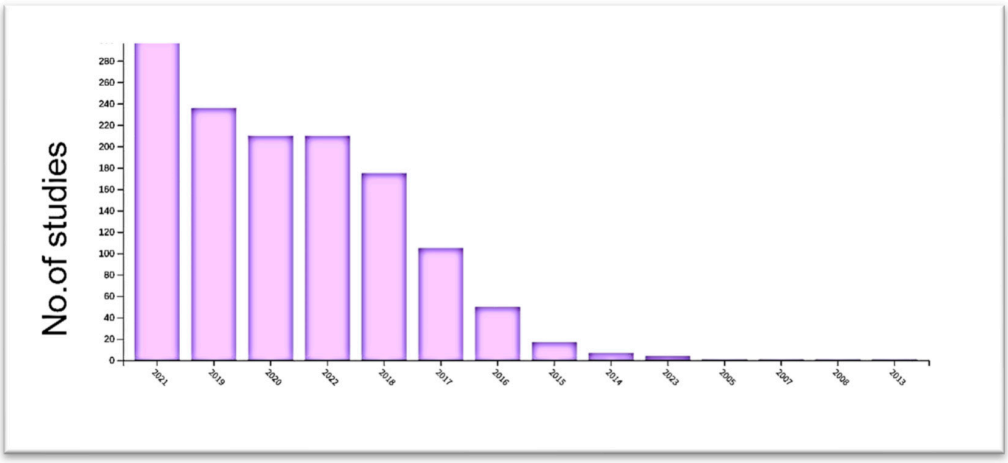

Figure 4. The distribution of CNN techniques in WOS on an annual basis. Since 2015, the number of studies on CNN applications in WOS has increased significantly.

One-stage processes, such as YOLO [28,29], SSD [30], and CornerNet [31,32], can be distinguished from two-stage techniques, such as R-CNN [33], Fast R-CNN [34], and Faster R-CNN [35], which made a breakthrough in object detection.

In the two-stage object detection process, region proposals are picked beforehand, and then CNN classifies the items. In a single step, the model simultaneously returns the category probability and position coordinates of the objects.

CNN [36], a biometric identification approach based on characteristics of the human face, is currently employed in the real world for face detection, a significant application.

Deepface [37] and DeepID [38] performed better than humans in unconstrained circumstances for the first time using the LFW dataset [39].

Detection, alignment, extraction, and classification constitute the DeepFace procedure. After detecting the face as the input to the CNN, Taigman et al. [37] trained the model using the Siamese network, achieving state-of-the-art performance. While DeepID directly enters two face pictures into CNN to extract feature vectors for classification, CNN is used by DeepID to extract feature vectors. Recent advancements in face recognition have primarily concentrated on the loss function.

FaceNet [38], which was proposed by Google in 2015, employs a 22-layer CNN to train a model with 200 million photos, including eight million humans. FaceNet substitutes softmax with triplet loss to discover more effective embeddings.

4. CNN for Action Recognition

CNN's models are now dominating action recognition (AR) in video and visual data. The objective of AR is to extract spatial information and motion over time, making video processing more complex than images. However, understanding human actions remains challenging due to the lack of equity concerning performance evaluation related to datasets, backbone choices, and experimental techniques [40].

Unlike image recognition, where ImageNet [41] has been the ideal benchmark for evaluation, the Kinetics dataset [42] is now the most popular reference for action recognition. Although, the kinetics dataset is highly biased in favor of spatial modeling, which ill-judged the validation of a spatiotemporal modeling capability model.

CNN-based action recognition generally offers three ways: 2D convolution [43–45] or 3D convolution [46–48] or both [49,50]. However, 2D and 3D methods are different in terms of feature extraction. Hence in the case of 2D convolution, a feature map extracts only spatial (two-dimensional) information, which involves adding another model to capture temporal information (fusions). In contrast, 3D convolution methods offer both spatial and temporal knowledge for a set of continuous images simultaneously.

Figure 5 illustrates the difference between 2D convolutions and 3D convolutions processing. The question is how they perform in contrast to each other concerning the Spatiotemporal modeling of video data. The 3D CNN is an extension of the success of 2D models in image recognition [51] to recognize actions in videos.

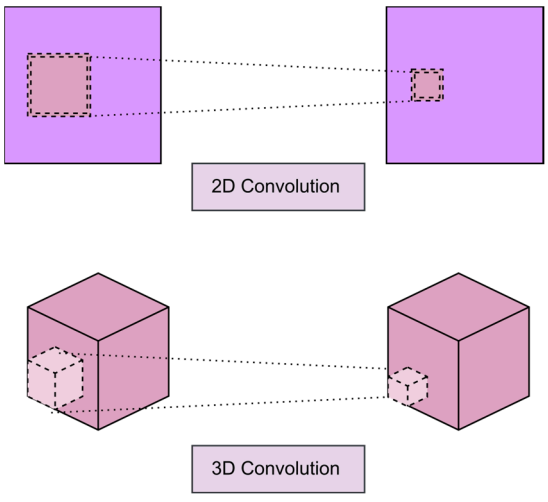

Figure 5. 2D vs. 3D convolution layer.

The objective is to extract spatiotemporal features directly from several video frames by applying a 3D filter on several adjacent video frames. Thus, motion information is caught. The operation at the position in the feature map in the layer is formalized as follows:

$$v_{il}^{xy} = \varnothing\left(b_{i,j} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1} W_{i,j,m}^{p,q,r} \cdot V_{i-1,\ m}^{x+p,\ y+q,\ z+r}\right) \quad (4)$$

where ϕ is the non-linear activation function, and w is the 3D weight matrix. P, Q, and Rare, respectively, the filter's height, width, and temporal length. One of the essential works on 3D architectures for action recognition dates back to 2012. The authors designed a 3D model for action recognition for the first time.

The model consists of a wired layer that generates the gray, gradient, and optical flow channels, followed by 3D convolution and subsampling applied to each channel and finally producing the final action by aggregating the information from all channels.

In 2015, Tran et al. presented a study [52] to find the most appropriate temporal length for 3D CNN and developed a VGG-style 3D CNN architecture. The same authors performed a search for 3D CNN in a deep residual learning framework and employed a ResNet18 style 3D CNN architecture named Res3D; this surpasses C3D by a considerable margin in terms of accuracy and recognition time.

The 2D approach is reached with two-stream models [53], processing the RGB images and the optical flow separately in two CNN models with late fusion in the higher layers [54].

In that respect, a 2D CNN model for image-level feature extraction and an additional model for temporal information capture. For example, the TRN [55] method relies on many features to structure images' relationships. The TSM [45] approach shifts some channels across the temporal dimension, allowing the exchange of information between neighboring images, or the TAM [40] method, which relies on 1×1 convolution in depth to capture temporal dependencies between images efficiently.

Various methods of temporal aggregation of feature descriptors have also been proposed. There are also more complex approaches that have been studied on how to capture the long term. These models have achieved SOTA performance on multiple large-scale benchmarks for instance, Kinetics [42], Something-Something [56], Sports1M [54], and Youtube-8M [57]. However, there is no winner between these approaches; 3D models perform more sufficiently than 2D standards on the Kinetics dataset, while 2D methods perform better on Something-Something.

5. Vision Transformer

In the beginning, transformer models [58] were only applied to Natural Language Processing (NLP). The transformers were used in text classification [59], language translation [60], and question answering [61].

For example, Vaswani et al. [62] developed a Transformer technique that was built on paying attention to activities related to machine translation. In 2008, a research group from Google led by Devin et al. [63] introduced BERT (Bidirectional Encoder Representations from Transformers), which consisted of 340 million parameters that had a huge impact on the future. On the basis of the BERT mechanism of self-attention, the obtained results have powerful representational capabilities that enable them to extract intrinsic characteristics.

The model pre-trains a Transformer on an unlabeled text that considers each word's context (it is bidirectional). The recent mixture of transformer approaches can reach a huge 1.6 trillion parameters and contains multiple FPNs [64]. The Transformer is achieving the SOTA performance in different NLP datasets such as Glue [65], SQuAD 2.0 [66], and Swag [66].

All this proves that Transformer has already dominated in NLP applications by showing better performance and speed than RNNs models, thus raising the question of the possibility of leading the computer vision community and overtaking CNN.

By all accounts, Transformer is already contributing to the computer vision domain (Figure 6), showing excellent results in different applications. For instance, object detection [67], segmentation [68], video understanding [51], and the like, as well as achieving the top performance on different image recognition benchmarks.

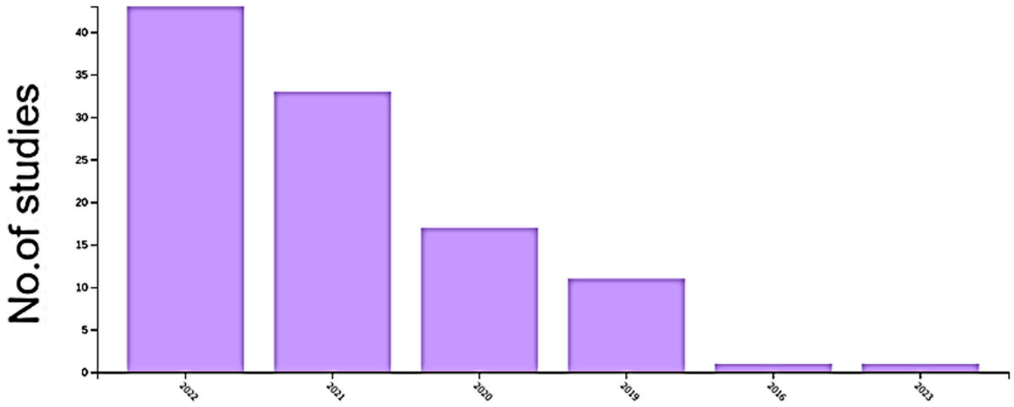

Figure 6. The Web-of-Science annual distribution of Vision Transformer techniques. Since 2020, the number of studies on Vision Transformer applications in WOS has expanded considerably.

5.1. The standard Vision Transformer

We present in this sub-section the general components of a Vision Transformer [11]. It is based typically on two major elements: a linear projection of an image and an encoder transformer that contains several MLP neural network models and a self-attention mechanism.

5.1.1. Patch Embedding

The standard method of Vision Transformer consists of partitioning the input image into separate patches of the same shape as a sequence of embedded words used when applying transformers to natural language. In other words, the Visual Transformer splits the image into visual tokens $(x_1, x_2, \ldots x_n)$ by $X \in R^{n \times d}$. While CNN uses pixel arrays, it is required to specify the patch size n. This stage consists of flattening the image patches returned, which means the vectorization of the patches into vectors, projecting the flattened patches into a lower-dimensional space by applying the linear function to each vector X_n.

The output is $Z_n = W_{xn} + b$ referred to as the patch embeddings. While W and b are two shared parameters learned from the training data, they also add a position embedding learned from patches $p \in 1, 2, \ldots, n$ to the Z vectors so that the Z vector captures content and position at the same time [11,69]. As a result, closer patches tend to have comparable similarity position embedding than others.

In classification tasks, another point to consider is adding another embedding learnable vector z_0 to the sequence X, which is the CLS token, to accumulate and store information extracted from the other tokens with the same shape as the other Z vectors [11].

5.1.2. Transformer Encoder

At this level, the approach consists of applying the encoder transformer [59]. The Multi-Head Self Attention layer is the major component of this procedure, applied to the sequence $z_1, z_2, \ldots$ besides the MLP model, Layer Norm (LN) is integrated before each block (Figure 7), and a residual connection is added to Multi-Head Attention.

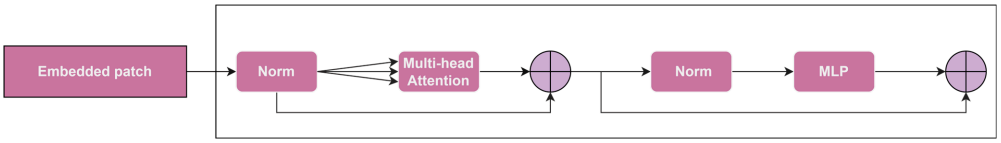

Figure 7. Transformer Encoder.

That is to say, if the major element of CNN is convolution, then Self-attention is the major element of the Transformer. The self-Attention layer captures long-term dependencies between all inputs and aims at transforming one element into another, in contrast to the short memory of RNN models that usually forget the content of the distant position and

mixes the contents of adjacent positions. It receives n entities without context and outputs n entities with contextual information.

In other words, the Self-attention layer takes the inputs in the form of $(x_1, x_2, \ldots x_n)$ by $X \in R^{n \times d}$, and applies three learnable weight matrices (Queries $W^Q \in R^d * d_q$, Keys $W^k \in d * d_k$ and Values $W^V \in d * d_k$).

Technically, comparing the query with all keys, re-weight, and aggregating the values with weights. The output of the self-attention layer is in the formula below:

$$\text{Attention}(\text{Q}, \text{K}, \text{V}) = z = \text{softmax}\left(\frac{Q.K^T}{\sqrt{d_q}}\right)V \tag{5}$$

where $z \in R^{n \times d}$ and **softmax** to get the attention score, with $\text{Q} = \text{XW}^{\text{Q}}$, $\text{K} = \text{XW}^{\text{K}}$ and $\text{V} = \text{XW}^{\text{V}}$ and the computation is the dot product.

Vision transformer uses Multi-Head Self Attention instead of a Self-Attention Layer, where the number of heads is generally eight for longer-term dependencies and to compress multiple complex relationships between different elements in the sequence, that is, the combined independent multiple self-attention that have the same input and do not share parameters $\text{W}_\text{i}^{\text{Q}}$, $\text{W}_\text{i}^{\text{K}}$, $\text{W}_\text{i}^{\text{V}}$ where $i = 0, \ldots (h-1)$ and h is the number of the attention blocks, $\text{W}_\text{i}^{\text{Q}} \in R^{d \times d_k}$, $\text{W}_\text{i}^{\text{K}} \in R^{d \times d_k}$ and $\text{W}_\text{i}^{\text{V}} \in R^{d \times d_k}$.

$$\text{MultiHead}(\text{Q}, \text{K}, \text{V}) = Concat(head_1, \ldots, head_h)\, W \tag{6}$$

where $head_i = Attention\left(Q\text{W}_\text{i}^{\text{Q}}, K\text{W}_\text{i}^{\text{K}}, V\text{W}_\text{i}^{\text{V}}\right)$, the results are then concatenated into one matrix $[\text{C}_0, \text{C}_1, \ldots \text{C}_{\text{h}-1}] \in R^{h.d \times d_k}$

5.1.3. Pure Transformer

Image processing is more challenging than text processing, given the high dimensions, noise, and redundant modality. The researchers proposed several innovative architectures in a very short time to address these challenges, such as positional coding and normalization strategy.

The first Vision Transformer approach was purely named ViT, realized by Google Research and Brain Team [11] for the classification task applied directly to image patches, as explained in the previous section.

ViT generally needs to be pre-trained on large datasets and fine-tuned to slighter tasks. If not, when trained in medium-sized datasets, it produces weak accuracy, such as the ImageNet dataset. The authors state that it is advantageous to use higher resolutions in the fine-tuning part than in the pre-training part.

Although ViT can capture long-range dependencies between patches, it does not consider regional feature extraction because the 2D patch is projected onto a vector with a single linear layer. Many works have proposed to solve the problem of localizing visual information.

For instance, TNT [70] splits the original patch into several sub-patches, and it introduces a new architecture, "Transformer in Transformer", which uses an internal transformer block to map the relationship of the sub-patches and an external transformer block for the exchange of information at the patch location.

Chu et al. [71] introduced a method, "Twins", which is a shifted window-splitting approach for cross-window connections to perform alternately local and global attention layer by layer.

Afterward, Z. Huang et al. introduced another method called Shuffle [72], which consists of using the spatial shuffle operation instead of the staggered window partitioning to allow cross-window connections. In contrast, the method of RegionViT [73] generates both regional and local tokens from the image. Hence, the local tokens get global information via attention to the regional tokens. T2T presents the aggregation of local features to enhance local information [74].

On the other hand, many approaches have been proposed to enhance the computation of self-attention, such as DeepViT [75], which proposed a method to increase diversity at different layers through inter-head communication to regenerate attention maps.

KVT [76] implements K-NN attention to use only the locality of image patches and compute only attention with the most similar tokens. At the same time, XCIT [77] performs self-attention computation on feature channels instead of tokens for effectively processing high-resolution images.

6. Transformers for Action Recognition

Action Recognition (AR) for Vision Transformers is a suitable target. In the same way as language modeling, in which words or input characters are represented as a set of tokens, videos are defined as a set of successive frames [51]. The Transformer encoder not only makes training and inference more powerful by not involving costly 3D convolutions, but also allows a complete video to be processed in a single pass.

Although CNN is still the most widely used, it is limited concerning long-term dependencies, either in space or time. Thereby, long-range dependencies can only be captured when these operations are repeatedly applied during the repetition of local operations has several limitations, is computationally inefficient, and creates optimization problems that must be handled carefully.

Neimark et al. [51] introduced a method named VTN, which discarded the standard approach of video action recognition based on 3D ConvNets, and set up a transformer-based method that takes into account all the information in the video clip, applying first the SOTA of 2D architectures to learn spatial features, add temporal information in the data stream using attention mechanisms on the resulting features by only inputting the RGB frames.

The temporal part of the VTN method follows the Longformer method presented in [78], which was addressed first for text processing to handle long sequences. Longformer deals with all the tokens in the input sequences with a within-reach complexity (O(n)) using sliding window attention. In contrast, the BERT classification token [63] is given via a fully connected layer to identify events or actions.

It showed competitive results in terms of accuracy, while the training and running were 16.1 and 5.1 faster during inference compared to the SOTA with different backbones.

Girdhar et al. [79] proposed an Action Transformer approach that gathers all human-specific contextual cues in the video clip to capture only the semantic context of others' actions.

For instance, focus on hands and face, two essential elements to identify an action. The method stands on two networks, the base, and the head. In the base network architecture, a 3D CNN architecture is applied to produce features transmitted to the Region Proposal Network (RPN) to get object proposals.

The Action Transformer Head applies self-attention layers on the person Box as a query (Q), while the features from the neighboring video clip are used as key (K) and value (V).

The self-attention layer is applied to add the context of other present people and objects to the query vector to facilitate subsequent classification. The key and the value features are calculated as a linear projection of the original feature map from the base network and are tensors of size $16 \times 25 \times 25 \times 128$, while the query is a 128-dimensional vector. Following the Transformer encoder mechanism (Equation (6)), the operation is represented as:

$$Z^{(r)} = \sum_{x,y,t} softmax\left(\frac{Q^r K_x^T}{\sqrt{D}}\right) V_{xyt} \tag{7}$$

$Q(r)$ corresponds to the features extracted by RPN by applying the scalar product on K features, normalized by $\sqrt{D}$. The resulting query is of the following form (Equation (8)).

The authors in [80] utilized a dropout to Z^r and appended it to the original query feature after it is passed through a residual branch consisting of a LayerNorm operation, followed

by a Feed-Forward Network (FFN). It is implemented as a 2-layer MLP and a dropout. The final feature is passed across another LayerNorm to get the updated query (Q'').

$$Q(r)' = \text{LayerNorm}(Q(r)) + Dropout(A(r))$$
$$Q(r)'' = \text{LayerNorm}\big(Q(r)'\big) + Dropout\big(FPN\big(Q(r)'\big)\big) \quad (8)$$

The authors used only RGB data as input and obtained good results at the semantic level. Additional modalities, such as motion/flow, are likely to improve efficiency and, therefore, increase computational cost.

Arnab et al. [81] proposed multiple pure transformer models for video classification called ViVit. They considered two methods for video embedding: Uniform frame sampling, which follows the mechanism, and Tubelet embedding, which is an extension of ViT embedding that corresponds to 3D convolution.

The results of the embedding are passed to the sequence of the video $V \in RT \times H \times W \times C$ to obtain a set of tokens which are the input of the transformer $z \in R^{n_t \times h_t \times w_t \times d}$. The authors proposed several variants that factorize the Spatiotemporal dimensions for the long token sequences encountered in videos.

The first model presents a straightforward extension of ViT that forwards all Spatiotemporal tokens through the encoder; each transformer layer takes all pairwise interactions between each Spatiotemporal token, which poses the problem of complexity $\big(O(n^2)\big)$.

The second model called a factorized encoder involves two separate transforming encoders, a spatial/temporal encoder; the first is used to model tokens extracted from the same temporal cue, and the second is used to model interactions between tokens from different temporal lines, while the output token from this encoder is ultimately classified (requires fewer FLOPs).

The third model, called Factorized Self-Attention, differs from the first model in that instead of calculating self-attention on all token pairs, a factorization aims to calculate self-attention first spatially and secondary temporally. Hence, each block of self-attention of the transformer modelizes the Spatiotemporal interactions. This model is better than model 1 in terms of efficiency and has the same complexity as Model 2.

The last model, named Factorized dot-product attention, with the same computational complexity as Models 2 and 3 while keeping the same number of parameters as the non-factored Model 1, consists of factorizing the multi-head dot-product attention operation, i.e., modifying the keys and values of each query to look only at tokens of the same spatial or temporal index by building K_s, $V_s \in R^{n_h \cdot n_w \times d}$ and K_t, $V_t \in R^{n_t \times d}$, while half of the head's attention is on the token spatial dimension Y_s, and the other half is for the token's temporal dimension Y_t , then merge the output of several heads by concatenating them and applying the linear projection $Y = concat(Y_s\ ,\ y_T\ t)$.

Plizzari et al. [82] suggested a two-stream Transformer network, as shown in Figure 8. On the one hand, A spatial-Self Attention (SSA) stream is applied in each frame to extract low-level features for embedding the relations between body joints [83], according to the self-attention formula, in a given frame at time t, for each node of the skeleton i^t, a query vector $q_t^i \in R^{dq}$, a key vector $k_t^i \in R^{dk}$ and a value vector $v_i^t \in R^{dv}$ are obtained by applying trainable linear transformations to the features of the node $n_i^t \in R^{C_{in}}$, shared by all nodes, of parameters $W_q \in R^{C_{in} \times d_q}$, $W_k \in R^{C_{in} \times d_k}$ and $W_v \in R^{C_{in} \times d_v}$.

Afterward, for each pair of body nodes (i^t, j^t), they applied a query-dot scalar product to get a weight $\alpha_{ij}^t \in R$ which denotes the power of the correlations among the two nodes. The outcome's grade α_{ij}^t is used to weigh each joint value v_i^t and a weighted sum is calculated to get a new embedding for the node i^t.

The spatial formula of this approach is like the following:

$$\alpha_{ij}^t = q_i^t \times k_j^t, z_i^t = \sum_i softmax\left(\frac{\alpha_{ij}^t}{\sqrt{d_k}}\right) v_j^t \quad (9)$$

where $W_q \in R^{Cha}$ (Cha is the number of output channels that form the new embedding of node i^t). The authors applied multi-head attention, which means repeating this operation H times, every single time with a distinct set of learnable parameters $(z_i 1^t, \ldots, z_i H^t)$ referring to the same node being merged using a learnable transformation (Equation (7)) to get the output of SSA.

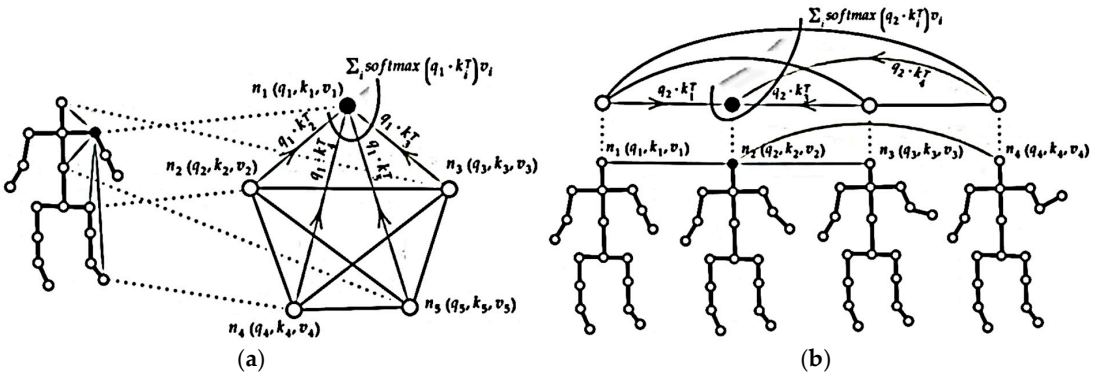

Figure 8. Two-stream Transformer network: (**a**) Spatial Self attention; and (**b**) Temporal self-attention Modified [82].

On the other hand, to capture the change in some joints over time, they proposed a temporal self-attention stream (TSA), in which the dynamics of each joint are studied separately with all images. The formula is the same as (Equation (9)), where i^v, j^v indicate the same articulation at different times.

Figure 8 depicts spatial/temporal attention for the representation of skeletal data. The NTU-RGB+D 60 [84] and NTURGB+D 120 [85] datasets produced SOTA results for both models. They merged the two streams. In addition, they used a 2s-STTR architecture, similar to those presented in [86,87].

7. Comparative Study between CNN, Vision Transformer, and Hybrid Models

To ensure a fair comparison, we use a multi-criteria analogy term in this section. We first present the advantages and disadvantages of the considered approaches, and then outline the state of the art of each method in terms of their modality, regarding the most widely used academic action recognition datasets.

7.1. Pros and cons of CNN and Transformer

One of the most significant advantages of CNN is the sparse connectivity that is provided by the Convolutional layer. This sparse connectivity saves memory because it only needs a small number of weights and connections [52], which makes it memory effective. Because there are no weights allocated between two neurons of neighboring layers in the CNN, and the set of weights operates with one and all pixels of the input matrix [52], weight sharing is another essential feature of the convolutional layer. This feature helps reduce the time needed for training and the costs associated with training.

In their evaluation of previous work, Koot et al. [88] discovered that CNN performs better than Transformer when it comes to latency accuracy on lightweight datasets. CNN is also described to capture inductive bias, which is also known as prior knowledge, such as translation equivariance and localization while having pooling operation give partial scale invariance [89].

CNN, on the other hand, has a few weaknesses, including a slowness that is brought on by the max pooling operation, and, in contrast to Transformer, it does not take into consideration the several perspectives that can be gained by learning [90], which leads to a disregard for global knowledge.

Due to the fact that it offers solutions to CNN's numerous weaknesses, the Transformer has quickly become CNN's most formidable opponent. The capability of Trans-

former to prioritize relevant content while minimizing the repetition of content that is not important is one of the program's capabilities [89]. In addition, because less demand is placed on the processing power, the visual characteristics are dynamically re-weighted based on the context [90]. This ability to mimic long-distance interactions in space and time, an essential requirement for visual movies [91], is another reason the Transformer stands out in front of CNN.

The transformer effectively encodes temporal data, an additional crucial component for action recognition. Lastly, multi-head attention, which is the booster part of the performance of vanilla self-attention and the essential component of the visual transformer process, affords the opportunity to learn many representations from various perspectives [92].

In spite of this, Dosovitskiy et al. [11] found in their research that transformer has a significantly lower level of image-specific inductive bias compared to CNNs. In order to overcome inductive bias, the model needs to be trained on very large and sufficient datasets so that it can figure out these image-specific features on its own based on the training examples. Therefore, it is important for the self-attention mechanism to automatically recognize the temporal relationships between video frames by searching through a huge database of video sequences. This is done in order to fulfill the requirements of the self-attention mechanism. The end consequence is longer training timelines, a significant increase in the demands placed on computer resources, and enormous datasets that need to be processed [90].

In light of what has been said, researchers are progressively merging these two models into a single model in order to leverage the complementary strengths of the two models and offset the flaws of the two models.

The findings of Zhao et al. [93] demonstrate that Transformer and CNN are mutually supportive of one another and could both be integrated into a single predictive model. They developed a hybrid model by employing multi-stage convolutional layers as the backbone of the model and exchanging a few particular layers for transformer layers. This offers the hybrid model with the global and local modelling capabilities of transformers and convolutions, respectively.

Dosovitskiy et al. [11] also acknowledged it. Experiments were undertaken to demonstrate that the transformer model excels after training on the CNN model. In light of CNN's ability to add location data to the Transformer model, it is important to note that the network is still in existence. Thus, a substantial amount of effort is required to add convolution to a typical transformer block. Hybrid models are approaches that mix CNN and Transformer.

Table 1 depicts the complementarity of the two models. A summary of the most pertinent works is offered in the subsequent section.

Table 1. CNN, Transformer, and hybrid model advantages, "-"indicates that the property is invalid in the method.

Properties	CNN	Transformer	Hybrid Model
Sparse connectivity	✔	-	✔
Weight sharing	✔	-	✔
Best at Latency accuracy on small datasets	✔	-	✔
Inductive bias	✔	-	✔
Capture local information	✔	-	✔
Dynamic weight	-	✔	✔
Capture global information	-	✔	✔
Learn from different angles	-	✔	✔
Best at Spatio-temporal Model long-distance interactions	-	✔	✔

Table 2 presents a comparison of the accuracy and complexity of the CNN, Transformer, and Hybrid model techniques for the recognition of actions.

Table 2. RGB comparison of CNN, Transformer, and Hybrid model approaches for Action recognition.

	Model + (Citation)	The Idea of the Model	Parameters	Flops	Year	Datasets (Accuracy)		
						UCF	HDM	Kin
CNN	Omi et al. [94]	Present a 3D CNN multi-domain-learning using adapters between layers. The results showed with ResNet backbone.	183.34		2022	63.33	93.29	67.80
	TEINet [95]	A technique for temporal modeling that enhances motion-related properties and adjusts the temporal contextual information channel-wise (backbone ResNet-50).		0.06	2020	96.7	72.1	76.2
	Xinyu Li [96]	Introduced a 3D CNN network that learns video clip-level temporal features from different spatial and temporal scales.	103	0.12	2020	97.9	75.2	
	SlowFast Networks [48]	A single-stream design that operates at two separate frame rates. SlowPath captures spatial semantics, but FastPath combines temporal semantics via the side connection. We displayed the outcomes using a 3D Resnet backbone.	32.88	0.36	2019			75.6
	Du Tran et al. [97]	Suggested factorizing 3D convolutions by separating channel interactions and spatiotemporal interactions in order to obtain greater precision at a lower computing cost.	32.8	1.08	2019			82.5
Transformer	Jue Wang et al. [98]	Dynamically predicts a subset of video patches to attend for each query location based on motion information.	73.9M	1.28	2022			79.0
	Arnab et al. [81]	Presented multiple models which factorize different components of the spatial-temporal transformer encoder. A solution to regularize the transformer model during training small datasets.		4.77	2021			84.9
	Liu et al. [99]	A pure transformer backbone model addressed the inductive bias of locality by utilizing the advantage of the intrinsic spatiotemporal locality in videos.			2022			84.9
	Yang et al. [100]	Fix the issue with videos' needed set length. A strategy that uses the attention date to repeatedly construct interaction between the current frame input and the prior hidden state.	107.7	0.11	2022			81.5
	Xiong et al. [101]	A multi-view transformer is composed of multiple individuals, each of which focuses on a certain depiction. Through a lateral link between individual encoders, information from several visual representations is successfully merged.			2022			89.1
	Zha et al. [102]	The components of the Shifted Chunk Transformer are a frame encoder and a clip encoder. The frame encoder uses the picture chunk and the shifting multi-head self-attention elements to capture intra-frame representation and inter-frame motion, respectively.	59.89	0.34	2021	98.7	84.6	83.0
	Zhang et al. [103]	Using the proposed standard deviation, an approach that aggregates Spatial-temporal data with stacking attention and an attention-pooling strategy to reduce processing costs.	0.392	0.39	2021	96.7	74.7	80.5
Hybrid- Model	Kalfaoglu et al. [104]	Combining 3D convolution with late temporal modeling is a great way to improve the performance of 3D convolution designs.	94.74	0.07	2020	98.69	85.10	
	Bonan Li et al. [105]	The issue of time-length videos was solved by implementing two attention modules: a short-term attention module and a long-term attention module, each of which provided a distinct temporal token attention.		2.17	2022			81.3

The accuracy and complexity of CNN, Transformer, and Hybrid model techniques for Action identification on the UCF101 (UCF), HMDB51 (HDM), and Kinetics-400 (Kin) datasets are compared in Table 2 [106–108]. The datasets [UCF] [106], [HMDB51 (HDM) [107], and [Kinetics-400 (Kin) [108] are [UCF] [106], [HMDB51 (HDM) [107], and [Kinetics-400 (FLops).

7.2. Modality Comparative Literature

Visual data may originate from a range of modalities and may be utilised singly or in combination to describe a visual activity [109]. This review concentrates on the RGB data and the skeletal data. RGB data are commonly employed in real-world application scenarios since they are simple to acquire and provide a wealth of information [109].

Using the following RGB datasets, we evaluate the efficacy of the techniques under consideration: UCF101 [106], which contains 27 h of video data, over 13,000 clips, and 101 action classes, containing video data totaling over 13,000 min.

HMDB51 [107] consists of 51 action categories and 7000 manually annotated footage extracted from various sources, including digital movies and YouTube.

Kinetics-400 [42] includes 400 classes of human motion with at least 400 video clips for each class. Each clip lasts about 10 s. The accuracy metric is used to evaluate the models since all classes are similarly important.

In support of what has already been stated, Table 2 illustrates how performance and complexity vary from one model to the next by highlighting a variety of methodologies utilized for each RGB-based model.

In the Kinetics dataset, Xiong et al. [101] achieved the highest level of accuracy with 89.1% within the Google Research lab, exceeding the findings that have been considered to be state-of-the-art thus far. This in no way negates the fact that all the models discussed produce findings that are fairly encouraging, independent of the datasets and metrics that are taken into consideration. Each model tries to fix a distinct issue by considering the existing issues that have been found.

For example, Bonan Li et al. [105], based on a CNN architecture, addressed the problem of time-length movies with two attention mechanisms. This confirms the hybrid model's viability as a method because it demonstrates that the problem can be solved using multiple attention mechanisms. Skeleton data are an additional acceptable modality for action recognition because they encode the trajectories of human body joints. These trajectories reflect meaningful human movements, simplicity, and informative representation, which are the primary characteristics of skeleton data.

The NTU-RGB+D 60 [84] and the NTU-RGB+D 120 [85] datasets are the most well-known examples of 3D action recognition. NTU-RGB+D 6O is comprised of 56,880 video clips and 60 activities that were carried out by 40 different individuals. Each individual human skeleton is made up of 25 joints, each of which has a unique set of three-dimensional coordinates. The NTU-RGB+D 120 database is an expansion of the NTU-RGB+D 60 database, and it contains 114,480 skeletal sequences of 120 action classes performed by 106 unique subjects. Cross-Subject Evaluation (CS) and Cross-View Evaluation are the metrics that are utilized in order to evaluate the reported outcomes in relation to these two benchmarks (CV).

CS evaluates the model according to the subjects who participated in the data set, while CV analyses the model according to the camera view. The evaluation results are presented for each of them as the classification accuracy is expressed as a percentage. CNN [110] served as the inspiration for the creation of a graph convolutional network (GCN), which was proposed to capture the structural relations among the data.

Because skeleton data occur naturally in graphs, numerous GCN techniques have been developed to represent skeleton data as graph structures consisting of edges and nodes. This is possible because skeleton data occur naturally in the form of graphs. Since GCN uses convolutions, works that employ this technology have been placed in the same area as CNN.

Table 3 is quite similar to Table 2. Still, instead of comparing models using the same datasets and metrics as Table 2, it compares models using various datasets and metrics described earlier in this section. Recent research has shown the effectiveness of Transformer and self-attention in resolving the same challenge as skeleton-based action recognition. This is despite the fact that CNN and GCN have made significant strides in solving this problem.

Qin et al. [111] used the same graph model that was used in GCN; however, they added a technique that gave the Transformer model with both the joint and bone representations of the skeleton as a single input.

Table 3. Skeleton modality comparison of CNN, Transformer, and Hybrid model techniques for action recognition on the datasets NTU-RGB+D 60 [84] and NTU-RGB+D 120 [85] in terms of the accuracy according to the Cross-Subject Evaluation (CS) and Cross-View Evaluation (CV).

	Model	Year	The Idea of the Model	Datasets (Accuracy)			
				NTU RGB+D		NTU RGB+D 120	
				CS	CV	CS	CV
CNN	Yan et al. [112]	2018	The authors developed the first strategy for collecting spatial and temporal data from skeleton data by encoding the skeleton data with GCN.	85.7	92.7		
	Banerjee et al. [113]	2020	The author developed a CNN-based classifier for each feature vector, combining Choquet fuzzy integral, Kullback–Leibler, and Jensen–Shannon divergences to verify that the feature vectors are complementary.	84.2	89.7	74.8	76.9
	Chen et al. [114]	2021	The authors employed a GCN-based method to model dynamic channel-by-channel topologies employing a refining technique.	93.0	97.1	89.8	91.2
	Chi et al. [115]	2022	The authors developed a novel method that combines a learning objective and an encoding strategy. A learning objective based on the information bottleneck instructs the model to acquire informative, yet condensed, latent representations. To provide discriminative information, a multi-modal representation of the skeleton based on the relative positions of joints, an attention-based graph convolution that captures the context-dependent underlying topology of human activity and complementing spatial information for joints.	92.1	96.1	88.7	88.9
	Song et al. [116]	2022	The authors developed a strategy based on a collection of GCN baselines to synchronously extend the width and depth of the model in order to extract discriminative features from all skeletal joints using a minimal number of trainable parameters.	92.1	96.1	88.7	88.9
Transformer	Shi et al. [117]	2021	The authors designed a pure transformer model for peripheral platforms or real-time applications. Segmented linear self-attention module that records temporal correlations of dynamic joint motions and sparse self-attention module that performs sparse matrix multiplications to record spatial correlations between human skeletal joints.	83.4	84.2	78.3	78.5
	Plizzari et al. [82]	2021	The authors proposed a novel method to the modelling challenge posed by joint dependencies. A spatial self-attention (SSA) module is used to comprehend intra-frame interactions between various body parts, while a temporal self-attention (TSA) module is used to describe inter-frame correlations.	89.9	96.1	81.9	84.1
	Helei et al. [80]	2022	The authors propose two different modules. The first module records the interaction between multiple joints in consecutive frames, while the second module integrates the characteristics of various sub-action segments to record information about multiple joints between frames.	92.3	96.5	88.3	89.2
Hybrid models	Wang et al. [111]	2021	The authors investigated employing a Transformer method to decrease the noise caused by operating joints. They suggested simultaneously encoding joint and body part interactions.	92.3	96.4	88.4	89.7
	Qin et al. [118]	2022	The authors proposed a strategy for concatenating the representation of joints and bones to the input layer using a single flow network in order to lower the computational cost.	90.5	96.1	85.7	86.8

8. Conclusions

This work examines CNN and Transformer for Action Recognition individually, as well as the trade-off between accuracy and complexity. In addition, this paper evaluates the majority of pertinent research emphasizing the benefits of each of the aforementioned tactics and their corresponding outcomes.

The challenge of visual action recognition is fraught with obstacles and limits. Since the quality of research has improved over time, it is evident that solutions are on the horizon for addressing these issues, whether by employing CNN or Transformer approach. Transformer, which is fairly new to the field of computer vision, has been quite competitive with CNN, which is ten years more established up to this point.

As for the primary question, and in light of this study, it should be mentioned that although both algorithms (i.e., CNN and Transformers) work in their way and have their own shortcomings and benefits, it is still difficult to determine who will win this race. Nevertheless, the hybrid method that is more efficient and cost-effective. It combines CNN with transformers to provide a reliable model. After all, the old adage asserts that working together is the key to success!

This hybrid model is the most attractive formula because it enables us to take advantage of a model's strengths while simultaneously reducing the effects of that model's downsides. Additionally, it has been demonstrated that hybrid models are highly useful for bridging the gaps generated by the deficiencies of specific models.

Therefore, we believe that this hybrid model might win the race. Furthermore, we anticipate a greater emphasis on testing this hybrid approach in action recognitions in visual data.

Author Contributions: Conceptualization, O.M., H.S. and S.T.; methodology, O.M., H.S. and S.T.; validation, A.C., R.S. and A.P.; formal analysis, O.M, H.S. and S.T.; investigation, O.M., H.S. and S.T.; writing—original draft preparation, O.M. and H.S.; writing—review and editing, A.C., R.S., A.P. and T.A.T.; visualization, A.C. and R.S.; funding acquisition, A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Noy applicable.

Informed Consent Statement: Noy applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep Learning for Computer Vision: A Brief Review. *Comput. Intell. Neurosci.* **2018**, *2018*, 1–13. [CrossRef] [PubMed]
2. Freeman, W.T.; Anderson, D.B.; Beardsley, P.A.; Dodge, C.N.; Roth, M.; Weissman, C.D.; Yerazunis, W.S.; Tanaka, K. Computer Vision for Interactive. *IEEE Comput. Graph. Appl.* **1998**, *18*, 42–53. [CrossRef]
3. Ayache, N. Medical computer vision, virtual reality and robotics. *Image Vis. Comput.* **1995**, *13*, 295–313. [CrossRef]
4. Che, E.; Jung, J.; Olsen, M. Object Recognition, Segmentation, and Classification of Mobile Laser Scanning Point Clouds: A State of the Art Review. *Sensors* **2019**, *19*, 810. [CrossRef]
5. Volden, Ø.; Stahl, A.; Fossen, T.I. Vision-based positioning system for auto-docking of unmanned surface vehicles (USVs). *Int. J. Intell. Robot. Appl.* **2022**, *6*, 86–103. [CrossRef]
6. Minaee, S.; Luo, P.; Lin, Z.; Bowyer, K. Going Deeper into Face Detection: A Survey. *arXiv* **2021**, arXiv:2103.14983.
7. Militello, C.; Rundo, L.; Vitabile, S.; Conti, V. Fingerprint Classification Based on Deep Learning Approaches: Experimental Findings and Comparisons. *Symmetry* **2021**, *13*, 750. [CrossRef]
8. Hou, Y.; Li, Q.; Zhang, C.; Lu, G.; Ye, Z.; Chen, Y.; Wang, L.; Cao, D. The State-of-the-Art Review on Applications of Intrusive Sensing, Image Processing Techniques, and Machine Learning Methods in Pavement Monitoring and Analysis. *Engineering* **2021**, *7*, 845–856. [CrossRef]
9. Deng, G.; Luo, J.; Sun, C.; Pan, D.; Peng, L.; Ding, N.; Zhang, A. Vision-based Navigation for a Small-scale Quadruped Robot Pegasus-Mini. In Proceedings of the 2021 IEEE International Conference on Robotics and Biomimetics (ROBIO), Sanya, China, 27–31 December 2021; pp. 893–900. [CrossRef]

10. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 6 January 2023).
11. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929.
12. Degardin, B.; Proença, H. Human Behavior Analysis: A Survey on Action Recognition. *Appl. Sci.* **2021**, *11*, 8324. [CrossRef]
13. Ravanbakhsh, M.; Nabi, M.; Mousavi, H.; Sangineto, E.; Sebe, N. Plug-and-Play CNN for Crowd Motion Analysis: An Application in Abnormal Event Detection. *arXiv* **2018**, arXiv:1610.00307.
14. Gabeur, V.; Sun, C.; Alahari, K.; Schmid, C. Multi-modal Transformer for Video Retrieval. *arXiv* **2020**, arXiv:2007.10639.
15. James, S.; Davison, A.J. Q-attention: Enabling Efficient Learning for Vision-based Robotic Manipulation. *arXiv* **2022**, arXiv:2105.14829. [CrossRef]
16. Sharma, R.; Sungheetha, A. An Efficient Dimension Reduction based Fusion of CNN and SVM Model for Detection of Abnormal Incident in Video Surveillance. *J. Soft Comput. Paradig.* **2021**, *3*, 55–69. [CrossRef]
17. Hubel, D.H.; Wiesel, T.N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* **1962**, *160*, 106–154. [CrossRef]
18. Huang, T.S. *Computer Vision: Evolution and Promise*; CERN School of Computing: Geneva, Switzerland, 1996; pp. 21–25. [CrossRef]
19. LeCun, Y.; Boser, B.E.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.E.; Jackel, L.D. Handwritten Digit Recognition with a Back-Propagation Network. *Adv. Neural Inf. Process. Syst.* **1990**, *2*, 396–404.
20. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **1980**, *36*, 193–202. [CrossRef]
21. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. *arXiv* **2014**, arXiv:1409.4842.
22. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In *Computer Vision–ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2014; Volume 8689, pp. 818–833. ISBN 978-3-319-10589-5.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
24. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
25. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep High-Resolution Representation Learning for Visual Recognition. *arXiv* **2020**, arXiv:1908.07919. [CrossRef]
26. Anwar, S.M.; Majid, M.; Qayyum, A.; Awais, M.; Alnowami, M.; Khan, M.K. Medical Image Analysis using Convolutional Neural Networks: A Review. *J. Med. Syst.* **2018**, *42*, 226. [CrossRef] [PubMed]
27. Valiente, R.; Zaman, M.; Ozer, S.; Fallah, Y.P. Controlling Steering Angle for Cooperative Self-driving Vehicles utilizing CNN and LSTM-based Deep Networks. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 2423–2428.
28. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2016**, arXiv:1506.02640.
29. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. *arXiv* **2016**, arXiv:1612.08242.
30. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2016**, arXiv:1512.02325.
31. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. *arXiv* **2019**, arXiv:1808.01244.
32. Law, H.; Teng, Y.; Russakovsky, O.; Deng, J. CornerNet-Lite: Efficient Keypoint Based Object Detection. *arXiv* **2020**, arXiv:1904.08900.
33. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv* **2014**, arXiv:1311.2524.
34. Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083.
35. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:1506.01497. [CrossRef]
36. Du, H.; Shi, H.; Zeng, D.; Zhang, X.-P.; Mei, T. The Elements of End-to-end Deep Face Recognition: A Survey of Recent Advances. *arXiv* **2021**, arXiv:2009.13290. [CrossRef]
37. Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1701–1708.
38. Sun, Y.; Wang, X.; Tang, X. Deep Learning Face Representation from Predicting 10,000 Classes. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1891–1898.
39. Liu, W.; Wen, Y.; Yu, Z.; Yang, M. Large-Margin Softmax Loss for Convolutional Neural Networks. *arXiv* **2017**, arXiv:1612.02295.
40. Chen, C.-F.; Panda, R.; Ramakrishnan, K.; Feris, R.; Cohn, J.; Oliva, A.; Fan, Q. Deep Analysis of CNN-based Spatio-temporal Representations for Action Recognition. *arXiv* **2021**, arXiv:2010.11757.

41. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009.
42. Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. The Kinetics Human Action Video Dataset. *arXiv* **2017**, arXiv:1705.06950.
43. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *Computer Vision–ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 9912, pp. 20–36. ISBN 978-3-319-46483-1.
44. Fan, Q. More Is Less: Learning Efficient Video Representations by Big-Little Network and Depthwise Temporal Aggregation. *arXiv* **2019**, arXiv:1912.00869.
45. Lin, J.; Gan, C.; Han, S. TSM: Temporal Shift Module for Efficient Video Understanding. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7082–7092.
46. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? In A New Model and the Kinetics Dataset. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 7–11 July 2017; pp. 4724–4733.
47. Hara, K.; Kataoka, H.; Satoh, Y. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 10–14 June 2018; pp. 6546–6555.
48. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. SlowFast Networks for Video Recognition. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6201–6210.
49. Luo, C.; Yuille, A. Grouped Spatial-Temporal Aggregation for Efficient Action Recognition. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5511–5520.
50. Sudhakaran, S.; Escalera, S.; Lanz, O. Gate-Shift Networks for Video Action Recognition. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1099–1108.
51. Neimark, D.; Bar, O.; Zohar, M.; Asselmann, D. Video Transformer Network. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11 October–17 October 2021; pp. 3156–3165.
52. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53. [CrossRef]
53. Simonyan, K.; Zisserman, A. Two-Stream Convolutional Networks for Action Recognition in Videos. *arXiv* **2014**, arXiv:1406.2199v2.
54. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale Video Classification with Convolutional Neural Networks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Columbus, OH), Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
55. Zhou, B.; Andonian, A.; Oliva, A.; Torralba, A. Temporal Relational Reasoning in Videos. In *Computer Vision–ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11205, pp. 831–846. ISBN 978-3-030-01245-8.
56. Goyal, R.; Kahou, S.E.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Fruend, I.; Yianilos, P.; Mueller-Freitag, M.; et al. The "Something Something" Video Database for Learning and Evaluating Visual Common Sense. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22 October–29 October 2017; pp. 5843–5851.
57. Abu-El-Haija, S.; Kothari, N.; Lee, J.; Natsev, P.; Toderici, G.; Varadarajan, B.; Vijayanarasimhan, S. YouTube-8M: A Large-Scale Video Classification Benchmark. *arXiv* **2016**, arXiv:1609.08675.
58. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv* **2020**, arXiv:1910.03771.
59. Fang, W.; Luo, H.; Xu, S.; Love, P.E.D.; Lu, Z.; Ye, C. Automated text classification of near-misses from safety reports: An improved deep learning approach. *Adv. Eng. Inform.* **2020**, *44*, 101060. [CrossRef]
60. Zhu, J.; Xia, Y.; Wu, L.; He, D.; Qin, T.; Zhou, W.; Li, H.; Liu, T.-Y. Incorporating BERT into Neural Machine Translation. *arXiv* **2020**, arXiv:2002.06823.
61. Wang, Z.; Ng, P.; Ma, X.; Nallapati, R.; Xiang, B. Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering. *arXiv* **2019**, arXiv:1908.08167.
62. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All you Need. *arXiv* **2017**, arXiv:1706.03762.
63. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
64. Fedus, W.; Zoph, B.; Shazeer, N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *arXiv* **2022**, arXiv:2101.03961.
65. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *arXiv* **2019**, arXiv:1804.07461.
66. Rajpurkar, P.; Jia, R.; Liang, P. Know What You Don't Know: Unanswerable Questions for SQuAD. *arXiv* **2018**, arXiv:1806.03822.
67. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. *arXiv* **2020**, arXiv:2005.12872.

68. Ye, L.; Rochan, M.; Liu, Z.; Wang, Y. Cross-Modal Self-Attention Network for Referring Image Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 10494–10503.
69. Zellers, R.; Bisk, Y.; Schwartz, R.; Choi, Y. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. *arXiv* **2018**, arXiv:1808.05326.
70. Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; Wang, Y. Transformer in Transformer. *arXiv* **2021**, arXiv:2103.00112.
71. Chu, X.; Tian, Z.; Wang, Y.; Zhang, B.; Ren, H.; Wei, X.; Xia, H.; Shen, C. Twins: Revisiting the Design of Spatial Attention in Vision Transformers. *arXiv* **2021**, arXiv:2104.13840.
72. Huang, Z.; Ben, Y.; Luo, G.; Cheng, P.; Yu, G.; Fu, B. Shuffle Transformer: Rethinking Spatial Shuffle for Vision Transformer. *arXiv* **2021**, arXiv:2106.03650.
73. Chen, C.-F.; Panda, R.; Fan, Q. RegionViT: Regional-to-Local Attention for Vision Transformers. *arXiv* **2022**, arXiv:2106.02689.
74. Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.; Tay, F.E.H.; Feng, J.; Yan, S. Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11 October–17 October 2021; pp. 538–547.
75. Zhou, D.; Kang, B.; Jin, X.; Yang, L.; Lian, X.; Jiang, Z.; Hou, Q.; Feng, J. DeepViT: Towards Deeper Vision Transformer. *arXiv* **2021**, arXiv:2103.11886.
76. Wang, P.; Wang, X.; Wang, F.; Lin, M.; Chang, S.; Li, H.; Jin, R. KVT: K-NN Attention for Boosting Vision Transformers. *arXiv* **2022**, arXiv:2106.00515.
77. El-Nouby, A.; Touvron, H.; Caron, M.; Bojanowski, P.; Douze, M.; Joulin, A.; Laptev, I.; Neverova, N.; Synnaeve, G.; Verbeek, J.; et al. XCiT: Cross-Covariance Image Transformers. *arXiv* **2021**, arXiv:2106.09681.
78. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *arXiv* **2020**, arXiv:2004.05150.
79. Girdhar, R.; Joao Carreira, J.; Doersch, C.; Zisserman, A. Video Action Transformer Network. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 244–253.
80. Zhang, Y.; Wu, B.; Li, W.; Duan, L.; Gan, C. STST: Spatial-Temporal Specialized Transformer for Skeleton-based Action Recognition. In Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China; 2021; pp. 3229–3237.
81. Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lucic, M.; Schmid, C. ViViT: A Video Vision Transformer. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11 October–17 October 2021; pp. 6816–6826.
82. Plizzari, C.; Cannici, M.; Matteucci, M. Spatial Temporal Transformer Network for Skeleton-based Action Recognition. In *International Conference on Pattern Recognition*; Springer: Cham, Switzerland, 2021; Volume 12663, pp. 694–701.
83. Manipur, I.; Manzo, M.; Granata, I.; Giordano, M.; Maddalena, L.; Guarracino, M.R. Netpro2vec: A Graph Embedding Framework for Biomedical Applications. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2022**, *19*, 729–740. [CrossRef]
84. Shahroudy, A.; Liu, J.; Ng, T.-T.; Wang, G. NTU RGB+D: A Large-Scale Dataset for 3D Human Activity Analysis. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1010–1019. [CrossRef]
85. Liu, J.; Shahroudy, A.; Perez, M.; Wang, G.; Duan, L.-Y.; Kot, A.C. NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2684–2701. [CrossRef]
86. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Skeleton-Based Action Recognition with Directed Graph Neural Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16-20 June 2019; pp. 7904–7913. [CrossRef]
87. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 12018–12027. [CrossRef]
88. Koot, R.; Hennerbichler, M.; Lu, H. Evaluating Transformers for Lightweight Action Recognition. *arXiv* **2021**, arXiv:2111.09641.
89. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *ACM Comput. Surv.* **2022**, *54*, 1–41. [CrossRef]
90. Ulhaq, A.; Akhtar, N.; Pogrebna, G.; Mian, A. Vision Transformers for Action Recognition: A Survey. *arXiv* **2022**, arXiv:2209.05700.
91. Xu, Y.; Wei, H.; Lin, M.; Deng, Y.; Sheng, K.; Zhang, M.; Tang, F.; Dong, W.; Huang, F.; Xu, C. Transformers in computational visual media: A survey. *Comput. Vis. Media* **2022**, *8*, 33–62. [CrossRef]
92. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A Survey on Vision Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 87–110. [CrossRef] [PubMed]
93. Zhao, Y.; Wang, G.; Tang, C.; Luo, C.; Zeng, W.; Zha, Z.-J. A Battle of Network Structures: An Empirical Study of CNN, Transformer, and MLP. *arXiv* **2021**, arXiv:2108.13002.
94. Omi, K.; Kimata, J.; Tamaki, T. Model-agnostic Multi-Domain Learning with Domain-Specific Adapters for Action Recognition. *IEICE Trans. Inf. Syst.* **2022**, *E105.D*, 2119–2126. [CrossRef]
95. Liu, Z.; Luo, D.; Wang, Y.; Wang, L.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; Lu, T. TEINet: Towards an Efficient Architecture for Video Recognition. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 11669–11676. [CrossRef]
96. Li, X.; Shuai, B.; Tighe, J. Directional Temporal Modeling for Action Recognition. *arXiv* **2020**, arXiv:2007.11040.

97. Tran, D.; Wang, H.; Feiszli, M.; Torresani, L. Video Classification with Channel-Separated Convolutional Networks. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5551–5560.
98. Wang, J.; Torresani, L. Deformable Video Transformer. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 14033–14042.
99. Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; Hu, H. Video Swin Transformer. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 3192–3201.
100. Yang, J.; Dong, X.; Liu, L.; Zhang, C.; Shen, J.; Yu, D. Recurring the Transformer for Video Action Recognition. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 14043–14053.
101. Yan, S.; Xiong, X.; Arnab, A.; Lu, Z.; Zhang, M.; Sun, C.; Schmid, C. Multiview Transformers for Video Recognition. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA; 2022; pp. 3323–3333.
102. Zha, X.; Zhu, W.; Lv, T.; Yang, S.; Liu, J. Shifted Chunk Transformer for Spatio-Temporal Representational Learning. *arXiv* **2021**, arXiv:2108.11575.
103. Zhang, Y.; Li, X.; Liu, C.; Shuai, B.; Zhu, Y.; Brattoli, B.; Chen, H.; Marsic, I.; Tighe, J. VidTr: Video Transformer Without Convolutions. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 13557–13567.
104. Kalfaoglu, M.E.; Kalkan, S.; Alatan, A.A. Late Temporal Modeling in 3D CNN Architectures with BERT for Action Recognition. *arXiv* **2020**, arXiv:2008.01232.
105. Li, B.; Xiong, P.; Han, C.; Guo, T. Shrinking Temporal Attention in Transformers for Video Action Recognition. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 1263–1271. [CrossRef]
106. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv* **2012**, arXiv:1212.0402.
107. Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB: A large video database for human motion recognition. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2556–2563. [CrossRef]
108. Imran, J.; Raman, B. Evaluating fusion of RGB-D and inertial sensors for multimodal human action recognition. *J. Ambient Intell. Humaniz. Comput.* **2020**, *11*, 189–208. [CrossRef]
109. Sun, Z.; Ke, Q.; Rahmani, H.; Bennamoun, M.; Wang, G.; Liu, J. Human Action Recognition from Various Data Modalities: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, 1–20. [CrossRef]
110. Zhang, S.; Tong, H.; Xu, J.; Maciejewski, R. Graph convolutional networks: A comprehensive review. *Comput. Soc. Netw.* **2019**, *6*, 11. [CrossRef]
111. Wang, Q.; Peng, J.; Shi, S.; Liu, T.; He, J.; Weng, R. IIP-Transformer: Intra-Inter-Part Transformer for Skeleton-Based Action Recognition. *arXiv* **2021**, arXiv:2110.13385.
112. Yan, S.; Xiong, Y.; Lin, D. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *arXiv* **2018**, arXiv:1212.0402. [CrossRef]
113. Banerjee, A.; Singh, P.K.; Sarkar, R. Fuzzy Integral-Based CNN Classifier Fusion for 3D Skeleton Action Recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 2206–2216. [CrossRef]
114. Chen, Y.; Zhang, Z.; Yuan, C.; Li, B.; Deng, Y.; Hu, W. Channel-wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition. *arXiv* **2021**, arXiv:2107.12213.
115. Chi, H.-G.; Ha, M.H.; Chi, S.; Lee, S.W.; Huang, Q.; Ramani, K. InfoGCN: Representation Learning for Human Skeleton-based Action Recognition. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 20154–20164.
116. Song, Y.-F.; Zhang, Z.; Shan, C.; Wang, L. Constructing Stronger and Faster Baselines for Skeleton-based Action Recognition. *arXiv* **2022**, arXiv:2106.15125. [CrossRef]
117. Shi, F.; Lee, C.; Qiu, L.; Zhao, Y.; Shen, T.; Muralidhar, S.; Han, T.; Zhu, S.-C.; Narayanan, V. STAR: Sparse Transformer-based Action Recognition. *arXiv* **2021**, arXiv:2107.07089.
118. Qin, X.; Cai, R.; Yu, J.; He, C.; Zhang, X. An efficient self-attention network for skeleton-based action recognition. *Sci. Rep.* **2022**, *12*, 4111. [CrossRef] [PubMed]