# Learning Higher-order Object Interactions for Keypoint-based Video Understanding

Yi Huang*, Asim Kadav, Farley Lai, Deep Patel, Hans Peter Graf
NEC Labs America
yihuang, asim, farleylai, dpatel, hpg@nec-labs.com

October 4, 2021

## Abstract

Action recognition is an important problem that requires identifying actions in video by learning complex interactions across scene actors and objects. However, modern deep-learning based networks often require significant computation, and may capture scene context using various modalities that further increases compute costs. Efficient methods such as those used for AR/VR often only use human-keypoint information but suffer from a loss of scene context that hurts accuracy.

In this paper, we describe an action-localization method, KeyNet, that uses only the keypoint data for tracking and action recognition. Specifically, KeyNet introduces the use of *object based keypoint information* to capture context in the scene. Our method illustrates how to build a structured intermediate representation that allows modeling higher-order interactions in the scene from object and human keypoints without using any RGB information. We find that KeyNet is able to track and classify human actions at just 5 FPS. More importantly, we demonstrate that object keypoints can be modeled to recover any loss in context from using keypoint information over AVA action and Kinetics datasets.

## 1 Introduction

Video understanding tasks such as action recognition have shown tremendous progress in the recent years towards various applications in video organization, behavioral analyses etc. Recent methods use parameter intensive 3D convolution or transformer-based networks over RGB data to reach state of the art results[12, 24, 32, 20, 22, 10, 23, 19, 11]. These methods are often expensive due to the large amount of computation involved in processing videos. However, many applications especially those in AR/VR often function in resource-constrained settings and often limit to using keypoint based human pose information that can be captured efficiently using hardware sensors. A major draw back of keypoint based methods is that they miss contextual information reducing the overall accuracy.

Figure 1: Left: RGB based action recognition Right: Proposed Human and object keypoint based action recognition

To address this problem, we develop a method that uses object and human keypoints to understand videos. By integrating object keypoints in our action recognition pipeline, we can recover the scene context information that is lost by using human keypoints. We propose capturing object keypoint information using the Pavlidis[1] algorithm over an existing real-time segmen-

*Work done as a NEC Labs Intern

tation method[15]. This information can also be alternatively obtained from hardware sensors such as WiFi or radar based sensors [29, 26]. This generates significant keypoint data over multiple frames that can be difficult to learn. Hence, we structure the joint and keypoint information in intermediate space using a transformer based architecture with joint and positional embeddings that allow KeyNet to recover contextual information and train for the action recognition and localization tasks.

In this setting, our method is not only is capable of preserving the advantage of the computation efficiency of keypoints-based methods but is also able to compensate for the loss of context information with our proposed context-aware structure representation. The primary contributions of our work can be summarized as three aspects: 1) We propose a context-aware structure representation using human and object keypoints in videos. To the best of our knowledge, it is the first work that utilizes the sub-sampled keypoints to provide context features of objects. 2) We propose KeyNet, a transformer-based network that successfully model the higher-order interaction of various actors and object in videos. 3) On various datasets, we demonstrate that our KeyNet architecture achieves superior performance as compared to the prior convolution-based methods and is an efficient video understanding method for real-world applications.

## 2 Related Work

We discuss and compare against other video understanding methods.

**RGB and multi-modal video understanding** Recent work on action recognition often use 2D/3D convolutions, optical flow and transformer based methods to learn relationships over spatio-temporal elements. For example, a large body of existing work uses the output from convolution blocks and aggregates the intermediate features. This representation is then pooled, along with LSTM or other building blocks to learn the temporal information. In contrast, 3D convolution methods, learn the temporal information with the spatial information. For example, some proposed methods, [17, 22, 10, 23] use a short video snippet as an input and use a series of deep convolution networks to capture the spatial-temporal features. Other methods such as I3D networks [4], and their generated features have been used for variety of video understanding tasks. For example, SlowFast networks combines the knowledge between fast frame rate and slow frame rate video to obtain high accuracy[2]. Multi-stream-based methods[8, 28, 31, 4, 19, 11] combine information from video frames and other modality, such as optical flow, human pose, and audio. They use multiple streams of deep convolution networks to model the knowledge from different modalities and leverage fusion techniques[9] to integrate the knowledge for action recognition. There are several methods that capture human-object interactions in the RGB space, often explicitly using the object information in the scene by using an object detector or convolutional feature maps to capture extract objects in the scene [13, 20].

**Keypoint-based Methods** Existing work over Keypoint-based action recognition uses the skeleton-based action recognition. Existing work follows the classification by detection approach or a top-down approach. Here, the first step is to estimate the keypoints and then use this information to create "video tracklets" of human skeletons, learning classification or localization tasks over this intermediate representation. For example, ST-GCN[32] uses graph convolution networks to jointly learn the relation of each human body part across each actor. Other work [18, 21] extend this work with addition edges to reasonably aggregate the spatial temporal information in videos. Early work in this area follows the RGB methods, extracting the pose features and then using RNN/LSTMs to learn the temporal information [7, 6, 25]. These methods do not capture any object information, and often limit to basic human pose-based action classes such as "walking", "dancing" etc. Another work, captures the object interactions but uses a separate RGB stream to learn objects and fuses it using a relational network [30].

**Our Work.** Our work intends to design a context-aware structure representation for videos that is aware of not only actors but also the interactive objects. Distinct from Non-keypoints-based action recognition, our method only uses sparse information in videos as input and models the knowledge with a lightweight model, therefore, make it more computationally efficient. Different from the skeleton-based methods, we build our structure representation by using the human and objects key-

points, early in our video representations. This allows the network to learn human-object interaction in the keypoint space, but introduces additional complexity in the intermediate space, where a large amount of keypoint information is introduced. In the next section, we introduce, how we structure this intermediate space to allow transformer networks learn from this information.

# 3 KeyNet

In this section, we describe the overall design of our proposed Keynet as shown in Figure 2. Our primary goal is to validate the hypothesis that using sparse keypoints can generate a representation that is sufficient to learn the interactions between each actor and the background context information.

The model consists of three stages and establishes a tubelet based action recognition pipeline. First, we estimate a set of human and object keypoints for $T$ frames video clip. Second, the Keypoints Embedding Network projects the keypoints to more representative features by introducing positional embeddings that introduces position, segment and temporal information to the keypoints. Finally, an Action Tagger Network learns the higher-order interactive features and assigns action tags for each actor or predict the action label for the video, depending on the dataset. We introduce the proposed Action Representation in Section 3.1, the Keypoints Embedding Network in Section 3.2, and the Action Tagger Network is described in Section 3.3.

## 3.1 Action Representation

**Scene Sequence.** We designed the keypoints-based action representation in KeyNet as a scene sequence $D$ where $H_i$ denotes the set of $k_h$ keypoints in the $i_{th}$ human tracklets and $O_j$ denotes the set of $k_o$ keypoints from the $j_{th}$ objects.

$$D = (H_1, H_2 ... H_N, O_1, O_2 ..., O_K)$$

$$H_i = (P_1, P_2, ..., P_{k_h})$$

$$O_j = (P_1, P_2, ... P_{k_o})$$

To obtain a scene sequence $D$ for action representation, we proposed a keypoints sampling method to extract $N$ human tracklets as $H_i$ for actor features and $M$ objects keypoints as $O_j$ for contextual features.

The object keypoints are introduced to compensate the context information loss in the scene information, often observed in keypoints-based methods.

**Human Tracklet.** To get $N$ human tracklets, we combine a person detector with simple IOU-based tracker, to build a person tubelets over $T$ frames. Then, we use the HRNet keypoints estimator is used to extract $P$ human joints information for each detected person over $T$ frames [27]. More precisely, for our person detector, we follow previous works [2] to apply Faster R-CNN with ResNeXt-101-FPN backbone. This detector is pretrained on COCO and fine-tune on AVA with mAP 93.9AP@50 on the AVA validation set. Regarding keypoints Estimator, we use HRNet[27] pretrained on PoseTrack with 81.6% AP on PoseTrack18 validation set. By selecting the top $N$ person based on the detection confidence score, we can form a human tracklet $S$ sequence with $N * P * T$ keypoints.

**Object Keypoint.** We extract object keypoints is to provide contextual features in scenes to enhance the performance for those object interactive actions. We proposed that human-object interactive action can be modeled by a set of class-agnostic keypoints with only the shape and spatial information about the object. Therefore, we extract the object keypoints by performing a subsampling along the contour of the mask detected by Mask R-CNN[15]. The flowchart for extracting keypoints is shown in Figure 3

More specifically, for each video clip, we apply Mask R-CNN on its keyframe to collect the class-agnostic masks and for each object mask. For contour tracing, we leverage the Theo Pavlidis' Algorithm [1] to obtain a set of keypoints around each detected object. Finally, by applying an equal distance sampling on the contour, we extract the keypoints that have the same interval along the contour of the detected mask. Hence, by selecting the top $M$ object with the highest confidence scores, we can obtain $B$ with $K * P$ keypoints for each $T$ frames video clips.

## 3.2 Keypoints Embedding Network

In this section, we describe how to build an intermediate structured information using no RGB data, and with

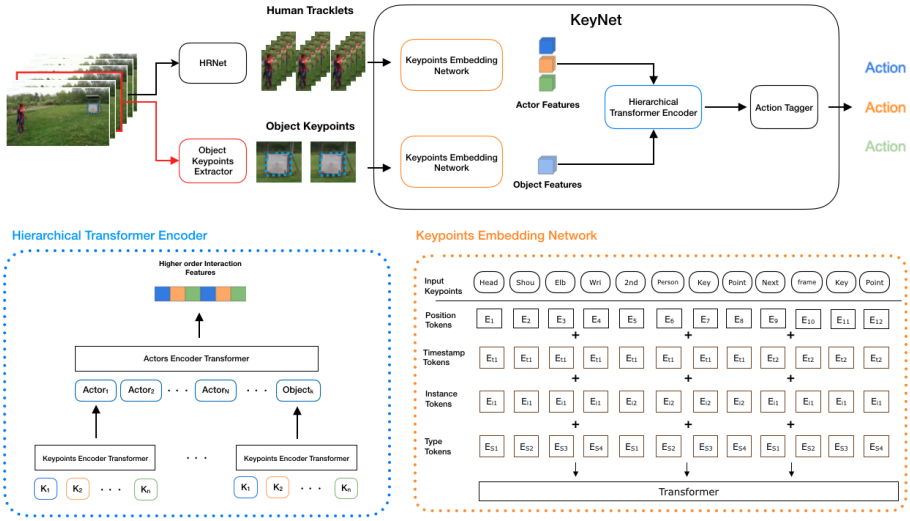


Figure 2: Flowchart for our proposed KeyNet.


Figure 3: We extract object keypoints over masks using Pavidilis algorithm

just person and object keypoints to perform action classification. To effectively learn the knowledge of action in keypoints representation, we need the information of the spatial correlation between each joint as well as how these joints may evolve through time. Therefore, we embed this information into the scene sequence by first converting each keypoint in a scene sequence to a sequence of Token and linearly projecting each Token into an embedding $E$, a learnable lookup table to model the relationship of each keypoint.

**Tokenization:** The goal of tokenization is to address extra spatial temporal information and convert it into a more representative information for learning the interaction between keypoints. To achieve this goal, we extend the prior tokenization techniques [24] by adding an additional instance token in the embedding representation for our experiments. For Position Token, Type Token and Segment Token, we follow previous work [24] to provide each keypoints with representations of spatial location, temporal location index, and the unique body type information (e.g. Head, Shoulder, and Wrist.) respectively. Our addition of extending the Segment Token to $T$ time frames and addressing the idea of Instance Token to indicate the id of tracklets that keypoints belong to in the current scene allow the network to learn localization information in the scene. We generalize the application of previous tokenization methods from pair-wise matching to jointly provide information of the spatial-temporal correlation of multiple instances at the same time. For the equation below, we described how to convert a scene sequence to 4 types of tokens:

**Position Token**[24]: The down-sampled spatial location of original image and gives the unique representation of each pixel coordinate. For a keypoints $P$, we write its Position Token as $\rho$, whose range lies in $[1, W']$, $[1, H']$. It reduce the computation cost while preserving the spatial correlation of each keypoints in image. The general expression of Position Token is below, where $\rho_n^{p_k^t}$ indicates the Position Token of the $k_{th}$ keypoint for the $n_{th}$ person in timestamp $t$.

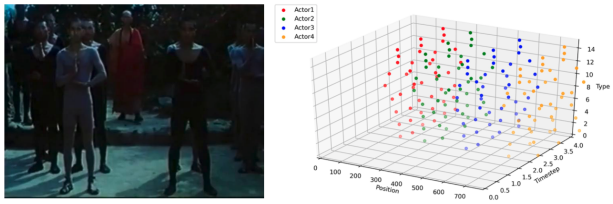


Figure 4: The visualization for our proposed Position Token, Type Token, Segment Token and Instance Token. The x, y and z axis represents the value of Position Token, Segment Token and Type token respectively and the color denotes the value of Instance Token.

$$\{\rho_1^{p_1^t}, \rho_1^{p_2^t} \cdots \rho_2^{p_1^t}, \rho_2^{p_2^t} ... \rho_{N-1}^{p_K^{t-T}} \cdots \rho_N^{p_K^{t-T}}\} \quad (1)$$

**Type Token**[24] The Type Token represents the characteristic of human body(i.e. Head, Right Shoulder and Left Wrist). It is range from $[1, K]$ where $K$ is the number of keypoints. It provides the knowledge of how each part of human body evolves in the keypoint sequence, which is an essential to achieve high accuracy at low resolutions. We assign the Type Token of a keypoint $P_n^{p_k^t}$ as $k$ and the Type Token for the $n_{th}$ person in timestamp $t$ can be written as $k_n^{p^t}$ . A general expression for Type Token is shown below

$$\{1_1^{p^t}, 2_1^{p^t} \cdots 1_2^{p^t}, 2_2^{p^t} \cdots (K-1)_N^{p^{t-T}} \cdots K_N^{p^{t-T}}\} \quad (2)$$

**Segment Token** The Segment token provides the difference between the timestamp of keypoints $p^t$ and the timestamp of key-frames. In our modelling of the video scene sequence, the range of Segment token is from [1, T] where T is the total number of frames in a video clip. We assign the Segment Token of a keypoint $P_n^{p_k^t}$ as $t$ and the Segment Token for the $k_{th}$ keypoint from the $n_{th}$ person can be written as $t_n^{p_k}$. The general expression of the Segment token is shown in Equation 3

$$\{1_1^{p_1^t}, 1_1^{p_2^t} \cdots 1_2^{p_1^t}, 1_2^{p_2^t} \cdots T_{N-1}^{p_K^{t-T}} \cdots T_N^{p_K^{t-T}}\} \quad (3)$$

**Instance Token** The Instance Token provides the spatial correlation for a keypoint $P^t$ that provides instance correlation within a frame. It serves a similar role as the Segment Token, providing spatial instead of temporal information. We assign the Instance Token of a keypoint $P_n^{p_k^t}$ as $n$ and the Instance Token for the $k_{th}$ keypoint in timestamp $t$ can be written as $n^{p_k^t}$. The general expression of the Segment token is shown in Equation 4

$$\{1^{p_1^t}, 1^{p_2^t} \cdots 2^{p_1^t}, 2^{p_2^t} ... (N-1)^{p_K^{t-T}} \cdots N^{p_K^{t-T}}\} \quad (4)$$

Here we define $P_n^{p_k^t}$ as the $k_{th}$ keypoint for the $n_{th}$ person in timestamp $t$. The visualization of our proposed tokenization methods in demonstrated in Figure 4. After tokenizing the scene sequence as the four types of the aforementioned tokens, we linearly projected each token to four types of embedding metrics and the output can be obtained by summing information of each type of token. That is $E = E_position + E_Type + E_Segment + E_Instance$. Finally, the Action Tagger Network takes the embedding $E$ as input to make the actor-level action recognition for each token.

## 3.3 Action Tagger Network

The goal of the Action Tagger Network is to learn the spatial-temporal correlation of each keypoints $P^t$ in scene sequence $D$ and make the prediction for given downstream tasks (e.g. action recognition and action localization) To achieve this, similar to make the prediction in sentence-level and token-level classification sub-task in BERT, we feed embedding vector $E$ to a series of self-attention blocks to model the higher-order-interaction for every keypoints embedding vectors. Then, we feed this representation to a fully-connected layers which is a learnable linear projection to make either sentence-level or token-level predictions.

**Transformer** In our implementation of Transformers, we use the Transformers create three vectors from each of the input vectors (in our case the embedding of each keypoints). Hence, for each of the keypoint embedding, we create a projection for the Query vector (Q), a Key vector (K), and a Value vector (V). Next, we score for every keypoints of scene sequence $S$ against other keypoints by taking the dot product of the query vector(Q) with the key vector(K) the respective keypoints. Finally, it normalizes

the score by $\sqrt{D}$ and a softmax operation. By multiplying each value vector(V) by the softmax score, the result can be obtained by summing up the weighted value vectors. The self-attention equation is as follow:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{D}})$$

**Hierarchical Transformer Encoder** In our experiments, we find that as the length of input sequence increases, the computation complexity slows down the learning efficiency for the transformer due to its quadratic processing time, i.e., $O(n^2)$ for a sequence with n elements. Hence, to address this quadratic inefficiency, for long sequences, instead of learning the self-attention weight of all keypoints in a single Transformer, we replace it with our proposed Hierarchical Transformer Encoder that learn the action representation in a hierarchical manner. Hence, given a keypoints embedding features $E_{\rho_n^t}$, a Keypoints Encode Transformer will first encode it into a list of action-level representation. We follow [5] we take the first representation $h^{\rho_n^t}$ as the feature for an actor.

$$E^{\rho_n^t} = (e_1^{\rho_n^t}, e_2^{\rho_n^t}, ...e_K^{\rho_n^t})$$

$$e_K^{\rho_n^t} = \rho_n^{p_k^t} + k_n^{p^t}$$

$$h^{\rho_n^t} = Transformer(E^{\rho_n^t})$$

where $\rho_n^{p_k^t}$ is the Position Token and $k_n^{p^t}$ is the Type token.

Then, an Actor Encode Transformer will encode the actor-level representation $(h^{\rho_n^1}, h^{\rho_n^2}...h^{\rho_N^{t-T}})$ to obtain context sensitive actor-level representations. $(d^{\rho_1}, d^{\rho_2}, ...d^{\rho_N})$ Finally, the actor level action classification is performed by linearly project $d^{\rho^n}$ to the number of total action classes in the given dataset.

$$R^{\rho_n^t} = (r_1^{\rho_n^t}, r_2^{\rho_n^t}, ...r_K^{\rho_n^t})$$

$$r^{\rho_n^t} = h^{\rho_n^t} + P_n^{p_k^t} + T_n^{p_k}$$

$$d^{\rho_n} = Transformer(R^{\rho_n})$$

where $P_n^{p_k^t}$ is the Instance Token and $T_n^{p_k}$ is the Segment Token.

# 4 Experiments

We evaluate the effectiveness of our approach on two tasks, action recognition, and action detection. For action recognition, we report the performance on JHMDB and Kinetics datasets, reporting the Top-1 accuracy score. For action localization, we report the performance on the AVA dataset, evaluating the mean average precision (mAP). The content of this section is organized as follow: First, we introduce the subsets of datasets used in our experiment in section 4.1. Then, we describe the implementation details in section 4.2. Finally, we report the performance of action recognition and action detection in 4.3 and 4.4, respectively.

## 4.1 Dataset

**JHMDB Dataset[16].** JHMDB dataset is a pose-action recognition dataset that consists of 659 training videos and 267 testing videos. It provides rich annotations including 15 joint positions, puppet mask, and puppet flow, which make it a good fit to evaluate KeyNet utilizing the evolution of human joints as the major information to recognize human action. In our experiments, we use this dataset as a starting point to validate if using only Keypoints as input modality is feasible for transformer-based architectures to recognize simple person movement actions. For evaluation, we report the performance of action recognition in terms of accuracy on the first split of the JHMDB dataset.

**Kinetics-skeleton Dataset[32].** Kinetics-skeleton dataset is collected by providing extra annotation of human skeleton keypoints on the Kinetics[4] dataset. Originally, the Kinetics dataset only provide coarse-grained action labels over the entire sequenece. Yan et al. [32] use publicly available human pose estimator, Openpose[3], to extract 18 keypoints for the top two persons, in every scene, with the highest confidence scores, in terms of the summation of joint confidence scores. In our experiments, we use this to validate if the proposed KeyNet can recognize action with keypoints annotation on different human body parts. For evaluation, we manually select 16 action categories and report the performance in terms of accuracy.

**AVA Dataset[14].** The Atomic visual Action (AVA) v2.1 consists of 211K, 57K, and 117K video clips for training, validation, and test sets. The center frame or keyframe is taken at 1 FPS from 430 15-minute movie clips with dense actor level annotation of all the bounding boxes and one or more among the 80 action classes. For evaluation, our goal is to focus on validating the effectiveness and feasibility of keypoint based approach on multiple actors. We sub-sample this dataset for two reasons. First, this dataset is heavily imbalanced, and even though RGB data can be augmented to handle class imbalance, improving class imbalances for pose information is rather difficult. Second, we identify the classes, where scene information provides the largest utility and test our methods specifically for those classes. Hence, to ease the high imbalance nature of AVA dataset, we manually select the 20 action classes that have more than 2000 samples including 8 classes of person movement actions (**P**), 4 classes of person-person interactive actions (**PP**) and 8 classes of person-object manipulation actions(**PO**). For evaluation, we follow the official method of using frame-level mean average precision (frame-AP) at IOU threshold 0.5 as described in [14]

## 4.2 Experiment Details

In this subsection, we provide our experiment details, including our hyperparameter settings and the data preprocessing procedure used in evaluating KeyNet. We use Adam as the optimizer and design a learning rate schedule with a linear warmup. The learning rate will warm up to the initial learning rate $\eta$ for a fraction of $0.01$ of total training iterations and then linear decay to 0 as reaching the total training iteration. For the action localization task on AVA dataset, we choose $N = 5$ human tracklets and $M = 3$ object masks to form the scene sequence and optimize our KeyNet model with batch size 32. For the action recognition task, we choose $N = 5$ human tracklets and $M = 1$ object mask to form the scene sequence and optimize our KeyNet model with batch size 64

**Data Augmentation** In our experiments, we found that data augmentation is a critical component to optimize the performance of our KeyNet. Without the augmentation techniques, KeyNet tends to easily overfit on those majority classes. (e.g. stand, sit, talk to and watch actions in AVA dataset). To solve this problem, we augment the training data with random flips, crops, and expand and further address the problem of the data imbalance with the $WeightedRandomSampler$ provided by Pytorch to equally sampled action categories in each training iteration before the estimation step. As shown in Table 1, adding data augmentation and re-sampling techniques can lead to a $+17.16\%$ performance gain in terms of mean average precision.

| Action Type | Data Aug. | Weighted Sampler | mAP |
|---|---|---|---|
| P | | | 14.25 |
| P | ✓ | | 20.28 |
| P | | ✓ | 16.42 |
| P | ✓ | ✓ | **31.41** |

Table 1: Ablation Study of techniques for the data imbalance in AVA dataset.

## 4.3 Performance on Action Recognition

Since recognizing action categories requires the awareness of both spatial and temporal domains, we first conduct experiments on small scale JHMDB dataset to determine the best spatial-temporal configuration for our proposed KeyNet. Then we generalize the task to kinetics dataset with more complex action and validate the effectiveness of using object keypoints to provide context features in videos.

**Spatial Resolution** Spatial resolution is a key factor for recognizing human action on small scales. Decreasing the spatial resolution caused the network to lose the fine-grained information but also reduce the computation cost. To determine the trade-off between the recognition performance and the computation cost, we variate the resolution of Position Token and report the performance and the computation cost for KeyNet. According to the statistic information in table 2, the optimal resolution for position token is $32x24$.

**Temporal Sequence length.** Temporal sequence length indicates the tokens along the temporal dimension which maps to the total number frames that the network processes from the input. Especially for those actions with slow motion (e.g. taichi), it is necessary to increase the temporal sequence lengths to let our model

| Token Resolution | Accuracy |
|---|---|
| 32∗24 | **55.81** |
| 64∗48 | 53.55 |
| 96∗72 | 50.41 |
| 128∗96 | 37.99 |

Table 2: Experiments for token resolution on the JHMDB dataset

| N Frames | Sequence Length | Token Size | Accuracy |
|---|---|---|---|
| 10 | 150 | 32∗24 | **55.81** |
| 15 | 225 | 32∗24 | 50.54 |

Table 3: Experiments of temporal footprints on the JHMDB dataset

fully capture the features of the entire action; however, this will cause the increases the computation. In table 3, we compare different configurations of temporal sequence lengths for our proposed KeyNet and find that the one with a longer temporal sequence tends to have worse performance indicating the difficulty of transformers in modelling longer sequences. The longer sequence length prevents the self-attention layers in the Transformer unit from learning the representative attention vectors for each type of action. Therefore, in our following experiments, we fix the number of frames in our input as 10 for a lower sequence length for the best performance.

**Effectiveness of Object Keypoints** To demonstrate the effectiveness of our strategy that using object keypoints to compensate the context information, we conduct experiments on JHMDB and Kinetics-16 dataset shown in Table. 4. We use the Kinetics-16 dataset to evaluate object based action recognition while JHMDB is collected to evaluate the action of human body part motion or single-person action. Our result show that the proposed methods improve the performance on the kinetics-16 dataset ($+4.5\%$) but also hurt the performance over the JHMDB dataset for a small margin ($-0.46\%$). This occurs because JHMDB dataset has been designed for single person action, often with little to no correlation with objects in the scene. As a result for majority of the classes, this additional information, adds complexity to the input space and makes learning difficult.

| Dataset | Object Keypoints | Accuracy |
|---|---|---|
| JHMDB | ✗ | **55.81** |
| JHMDB | ✓ | 55.35 |
| Kinetics-16 | ✗ | 45.40 |
| Kinetics-16 | ✓ | **49.90** |

Table 4: Experiments of using object keypoints to provide context features in action recognition tasks

## 4.4 Performance on Action Detection

For this subsection, we describe the details about how to generalize our proposed methods to the action localization scenario to predict action for each actor in scene sequence $D$. This is analogous to the correlation between sentiment analysis (sentence-level predictions) and Part of Speech Tagging (token-level predictions) in the Natural Language Process field. The implementation can easily be done by replacing the last fully connected layer with a multi-label prediction layer. However, we discover this poses two challenges: First, how to provide sufficient information to learn the complex interaction across each tracklet. Second, how to boost the learning efficiency of a long sequence of keypoints extracted from multi-person and multi-object data annotations.

**1 Frames Per Second.** For the first challenge, the most intuitive way to provide additional information is to increase the temporal footprint. As a result, we must address the issue of learning efficiency mentioned in 4.3. Therefore, instead of collecting more frames in a scene sequence, we decrease the sampling rate in videos. Our proposed workflow is described as followed: First, we detect and estimate keypoints for human instances in all video frames. Then, we run a tracking algorithm for each of the detected bounding box starting from the key-frames. Finally, we acquire the tracklets with different temporal footprints by sub-sampling the frames with specific intervals. We report and analyze the performance of KeyNet with different temporal footprint settings to demonstrate the effectiveness of our proposed method. As shown in in Table 5, decreasing FPS from 5 to 1 can lead to $+3.24\%$ for person movement actions (P) and $+1.8\%$ for person movement and person-person interaction actions (PP).

**Hierarchical Self-Attention Layer.**. In table 6, we have shown that by learning the person-level and actor-

| Input Modality | FPS | Temporal Footprints | P | P + PP |
|---|---|---|---|---|
| Keypoints | 5 | 2s | 28.17 | 14.94 |
| Keypoints | 1 | 5s | 31.41 | 16.73 |

Table 5: Comparison of temporal footprint and input modality. $P$ denotes the Person Movement actions and $PP$ denotes Person-Person interactive actions in terms of mean average precision (mAP)

| Keypoints | Action Type | mAP |
|---|---|---|
| Transformer | P | 26.85 |
| Hierarchical Self-attention | P | 31.41 |

Table 6: Effectiveness of our proposed hierarchical self-attention layer. Noted $P$ denotes the person-movement actions in the AVA dataset.

level knowledge hierarchically, our proposed hierarchical self-attention layer can improve the learning efficiency on the AVA dataset and lead to a $4.56\%$ performance gain. We also provide the performance of different configurations for transformer architecture. According to table 7, the best configuration is using 6 heads, 4 hidden layers and layers with 128 hidden unit. We follow this optimal setting for the following experiments.

**Object Keypoints.** We evaluate the effectiveness of object keypoints on all of the selected actions in the AVA dataset, including person movement(P), person-person interaction(PP), and person-object manipulation (PO) action categories. Based on the statistical information in table 8, the model with object keypoints to compensate context information loss has superior performance to the one without object keypoints.

**Context Information Recovery** To demonstrate the effectiveness of using the object keypoints to recover context information in videos, we design a transformer-based RGB baseline to compare the recognition performance with Keypoints-based methods. For the RGB baseline, we directly takes the image-level features from HRNet[27] and Mask-RCNN[15] as actor and context features. Then we feed the actor and context features to the same Action Tagger Network with our KeyNet. In Table 9, it clearly shows that KeyNet, using only keypoints, can achieve better performance than the RGB-baseline and based on our results, we believe that using human and object keypoints as a structure representation has the potential to fully recover the essential context information for action recognition.

| N Heads | N Hidden | Hidden Size | Int.Size | Param. | mAP |
|---|---|---|---|---|---|
| 2 | 4 | 128 | 128 | 0.91 M | 29.47 |
| 4 | 4 | 128 | 128 | 0.91 M | 29.45 |
| 6 | 4 | 128 | 128 | 1.78 M | **30.56** |
| 4 | 4 | 64 | 256 | 1.99 M | 24.09 |
| 4 | 4 | 128 | 128 | 0.91 M | 29.45 |
| 4 | 6 | 128 | 128 | 5.97 M | 30.41 |

Table 7: Experiments for the architecture searching for the proposed transformer-based architecture.

| Object Keypoints | Action Type | mAP |
|---|---|---|
| ✗ | P + PP + PO | 11.23 |
| ✓ | P + PP + PO | 11.45 |

Table 8: Effectiveness of addressing object keypoints to provide contextual features.

# 5 Conclusion

In this work, we demonstrate that using the object-based keypoints informatio can compensate for accuracy loss due to the missing context information in keypoint-based methods. We also show a method to extract object keypoints from segmentation information and build a structure representation with human keypoints from videos. According to our experimental results, we have validated

| Input Modality | Action Type | mAP |
|---|---|---|
| RGB | P | 18.12 |
| Keypoints | P | 31.41 |
| RGB | P + PP | 15.85 |
| Keypoints | P + PP | 16.73 |
| RGB | P + PP + PO | 9.28 |
| Keypoints | P + PP + PO | 11.45 |

Table 9: The demonstration of context information recovery by comparing the performance of using full images and keypoints as input modality. $P$ denotes the Person Movement actions and $PP$ denotes Person-Person interactive actions in terms of mean average precision (mAP)

our proposed KeyNet has superior performance to the RGB baseline, a method based on image-level information, and shows the potential of using only keypoints to recover essential context information for action recognition.

# References


[1] Theo Pavlidis' Algorithm. *Algorithms for Graphics and Image Processing in 1982, chapter 7 (section 5).*, 1982.

[2] Slowfast networks for video recognition. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October, 2019.

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih En Wei, and Yaser Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, 2021.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, 2017.

[5] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1(Mlm):4171–4186, 2019.

[6] Wenbin Du, Yali Wang, and Yu Qiao. Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3725–3734, 2017.

[7] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.

[8] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional Two-Stream Network Fusion for Video Action Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December(i):1933–1941, 2016.

[9] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional Two-Stream Network Fusion for Video Action Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December(i):1933–1941, 2016.

[10] Yutong Feng, Jianwen Jiang, Ziyuan Huang, Zhiwu Qing, Xiang Wang, Shiwei Zhang, Mingqian Tang, and Yue Gao. Relation Modeling in Spatio-Temporal Action Localization. 2021.

[11] Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees G.M. Snoek. Actor-transformers for group activity recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 836–845, 2020.

[12] Rohit Girdhar, Joao Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:244–253, 2019.

[13] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8359–8367, 2018.

[14] Chunhui Gu, Chen Sun, David A Ross, George Toderici, Caroline Pantofaru, and Susanna Ricco. AVA_A Video Dataset of Atomic Visual Actions.pdf. pages 6047–6056, 2018.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397, 2020.

[16] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J. Black. Towards understanding action recognition. *Proceedings of the IEEE International Conference on Computer Vision*, pages 3192–3199, 2013.

[17] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:7082–7092, 2019.

[18] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 140–149, 2020.

[19] Chih-yao Ma, Min-hung Chen, Zsolt Kira, Ghassan Alregib, and C V Mar. Exploiting Spatiotemporal Dynamics for Activity Recognition.

[20] Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan Alregib, and Hans Peter Graf. Attend and Interact: Higher-Order Object Interactions for Video Understanding. Technical report.

[21] Yuya Obinata and Takuma Yamamoto. Temporal Extension Module for Skeleton-Based Action Recognition. pages 534–540, 2021.

[22] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-Context-Actor Relation Network for Spatio-Temporal Action Localization. 2020.

[23] Hao Shao, Shengju Qian, and Yu Liu. Temporal interlacing network. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 1(c):11966–11973, 2020.

[24] Michael Snower, Asim Kadav, Farley Lai, and Hans Peter Graf. 15 Keypoints Is All You Need. 2019.

[25] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[26] Fei Wang, Stanislav Panev, Ziyi Dai, Jinsong Han, and Dong Huang. Can wifi estimate person pose? *arXiv preprint arXiv:1904.00277*, 2019.

[27] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *arXiv*, (March):1–23, 2019.

[28] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc van Gool. Temporal segment networks: Towards good practices for deep action recognition. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9912 LNCS:20–36, 2016.

[29] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 851–860, 2016.

[30] Wei Wang, Jinjin Zhang, Chenyang Si, and Liang Wang. Pose-based two-stream relational networks for action recognition in videos. *arXiv preprint arXiv:1805.08484*, 2018.

[31] Zuxuan Wu, Yu Gang Jiang, Xi Wang, Hao Ye, and Xiangyang Xue. Multi-stream multi-class fusion of deep networks for video classification. *MM 2016 - Proceedings of the 2016 ACM Multimedia Conference*, pages 791–800, 2016.

[32] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 7444–7452, 2018.