Enhancing Skeleton-Based Action Recognition in Real-World Scenarios through Realistic Data Augmentation

Mickael Cormier[1,2,3] Yannik Schmid[1] Jürgen Beyerer[1,2,3]
[1]Fraunhofer IOSB [2]Karlsruhe Institute of Technology [3]Fraunhofer Center for Machine Learning
{firstname.lastname}@iosb.fraunhofer.de

Abstract

Skeleton-based action recognition is a prominent research area that provides a concise representation of human motion. However, real-world scenarios pose challenges to the reliability of human pose estimation, which is fundamental to such recognition. The existing literature mainly focuses on laboratory experiments with near-perfect skeletons, and fails to address the complexities of the real world. To address this, we propose simple yet highly effective data augmentation techniques based on the observation of erroneous human pose estimation, which enhance state-of-the-art methods for real-world skeleton-based action recognition. These techniques yield significant improvements (up to +4.63 *accuracy) on the widely used UAV Human Dataset, a benchmark for evaluating real-world action recognition. Experimental results demonstrate the effectiveness of our augmentation techniques in compensating for erroneous and noisy pose estimation, leading to significant improvements in action recognition accuracy. By bridging the gap between laboratory experiments and real-world scenarios, our work paves the way for more reliable and practical skeleton-based action recognition systems. To facilitate reproducibility and further development, the Skelbumentations library is released at* https://github.com/MickaelCormier/Skelbumentations. *This library provides the code implementation of our augmentation techniques, enabling researchers and practitioners to easily augment skeleton sequences and improve the performance of skeleton-based action recognition models in real-world applications.*

1. Introduction

Action recognition is the process of classifying different actions in videos and is used in various applications such as human-computer interaction and surveillance. Deep learning techniques have led to the development of different methods for action recognition, including RGB-based and skeleton-based approaches. However, RGB-based methods are sensitive to environmental factors like background color, lighting conditions, and clothing, which can result in recognition errors. In contrast, skeleton-based methods have gained popularity due to their resilience to these factors. These methods analyze the 2D or 3D coordinates of major human joints over time to accurately recognize actions.

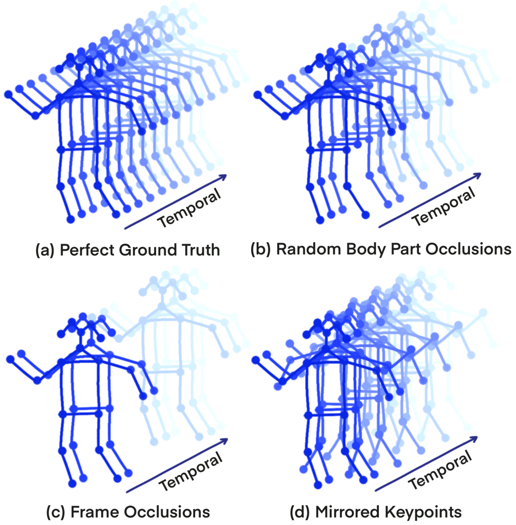

Figure 1. **Realistic data augmentation –** Illustration of augmented skeleton sequences. A skeleton sequence is shown as ground truth in a) where all keypoints of the skeletons are perfectly placed for each frame. Missing keypoints due to low-confidence are represented in b) where body parts such as a leg or an arm are missing on single frames. The case of a person totally occluded for a few seconds is shown in c), where a block of skeletons is missing for several frames. The case of mirror swapping for which keypoints detector fails to correctly differentiate between left and right is shown in d).

However, in real-world scenarios, human pose estimation, which is essential for skeleton-based action recognition, faces significant challenges due to environmental factors and noise. Existing literature primarily focuses on laboratory experiments with near-perfect skeletons, neglecting the complexities of real-world conditions. This creates a gap between high recognition rates in controlled environments and lower rates in more challenging datasets.

There are several challenges in skeleton-based action recognition. Missing detections during tracking can be addressed through interpolation, but this can lead to imprecise bounding boxes and inaccurate pose estimations. Fragmented tracks, where the subject disappears or is occluded for a significant duration, require zero-padding to maintain a fixed track length. Additionally, applying Human Pose Estimation models on a single-frame basis may introduce artifacts like jittering, random keypoints or sudden body part swaps. Furthermore, in real-world scenarios, when applying a confidence threshold to the keypoints detector, keypoints with low confidence are often missing. Finally, the use of temporal upsampling in order to improve the speed of processing and consistency over time may also produce interpolation artifacts if the chosen key frames are erroneous.

To address these limitations and enhance the reliability of skeleton-based action recognition models in the presence of environmental noise, we propose dedicated augmentation techniques. These techniques aim to improve the robustness of skeleton-based action recognition models by mitigating the impact of environmental factors and artifacts that arise throughout the action recognition pipeline. Since this pipeline is required to operate in real-time while delivering reliable predictions, the use of lightweight models is an imperative. Thus the use of larger models for Human Pose Estimation or larger and more complex Graph Convolutional Networks (GCNs) do not present a realistic option. In this case we chose a data-driven solution in augmenting the training with multiple example of realistic noise sources. Specifically, as illustrated in Fig. 1 we propose data-augmentation techniques for modelling keypoints as well as body parts occlusion in the case of body parts not being visible as well as for the case of missing keypoints due to low confidence. We use frame occlusion for cases of a person occlusion for several seconds. We also cover the case a keypoints detector failing to correctly identifying left and right by modelling keypoints mirroring and swapping. Furthermore, we simulate noise generated by temporal upsampling.

We conduct extensive experiments on both sets of the UAV-Human action recognition dataset [23] and additionaly validate our results on the JHMDB dataset [17]. Our main contribution are summarized as follows:

- We propose realistic data augmentation techniques enhancing skeleton-based action recognition human action recognition in real-world scenarios with noisy skeleton input.
- We release Skelbumentation, a library with the implementation of our augmentation techniques, to easily augment skeleton sequences and improve the performance of skeleton-based action recognition models in real-world applications.
- All state-of-the-arts methods trained with our method on both benchmarks for skeleton-based action recognition largely outperform their baseline.

2. Related Work

2.1. Skeleton-based Action Recognition

There is a long series of works in action recognition from videos using deep learning, starting from [36] which relied on RGB and optical flow, leading to several improvements using spatio-temporal representations [7, 14, 19, 35, 44, 45, 47, 51] and better performance in standard action recognition benchmarks. However, RGB and flow-based models are often biased [24, 25, 46] and challenging for real-world scenarios in term of privacy. Using an abstract representation of a person through a sequence of skeletons solves most privacy issues and offers through its representation of human joints and motions is less susceptible to dataset biases. The use of skeleton features was thus early recognized as an efficient alternative for action recognition [17] and also recently for salient behavior recognition [15]. Nowadays, with the fast developement of deep learning for human pose estimation, the extraction of skeletons from in-the-wild videos for skeleton-based action recognition have made tremendous progess [3, 6, 10, 21, 26, 43, 48, 50] and offer different strategies from bottom-up models extracting multiple poses at the same time [3, 6, 21] to top-down models relying on the performance of person detectors [10, 26, 43, 48, 50]. However [8] highlights the important challenges faced by pose detector in real world surveillance scenarios and show that it is currently almost impossible in such scenarios to obtain a skeleton quality similar as dataset collected in cooperative situations.

2.2. Datasets

The NTU RGB+D Dataset [28, 32], is probably one of the most popular example of large-scale dataset for action recognition recorded in laboratory. It consists of RGB and depth videos capturing various actions performed by multiple subjects in different environments. The dataset provides synchronized RGB and depth modalities, along with annotations of 3D skeletal joint positions for each action sequence. The dataset is divided into cross-subject (CS) and cross-view (CV) splits to evaluate generalization abilities across subjects and camera views, respectively.

Skeletics 152 [16] is a large dataset which aims at benchmarking skeleton based human action recognition in-the-wild. It is based on the much larger Kinetics-700 dataset [4, 20] composed of youtube videos of 700 actions. Skeletics 152 provides a total of 152 curated actions with 3D skeletons estimated from RGB Videos. Posetics [49] is a similar dataset build from Kinectics-400 [20] with refined and filtered poses in 2D and 3D. This dataset is used for large-scale pre-training real-world skeleton-based action recognition.

Recent datasets tackle the challenge of real-world scenarios for surveillance and autonomous robots. The JRDB dataset [12, 30, 41] is a large-scale benchmark for egocentric robot visual perception captured on an university campus. It provides annotations for spation-temporal action, social group and activity detection as well as for multi-person pose estimation and tracking in challenging crowded indoor and outdoor locations. The Human in Events dataset [27] is a large-scale dataset for understanding human motions, poses and actions in complex and realistic events. The dataset is manually annotated and covers a wide range of human-centric annotations including tracking, pose estimation and action recognition and focuses on challenging scenes which are crowded and complex.

2.3. Graph Convolutional Neural Networks

The majority of these skeleton-based action recognition datasets is dominated by GCNNs which contributed an important performance boost [5, 29, 33, 34, 40]. Most of these approaches profit from an improved representation of skeleton toplogy to process long-rang dependencies.

The 2s-AGCN [33] introduced an adaptive graph convolutional network that utilizes self-attention to dynamically learn the graph topology, resulting in improved action recognition performance. Building upon this, the MS-AAGCN [34] extended the approach by incorporating multi-stream adaptive graph convolutional networks with attention modules and a 4-stream ensemble based on 2s-AGCN [33]. These methods primarily focus on spatial modeling of actions. In contrast, the MS-G3D Net [29] proposed a unified approach for capturing complex joint correlations across both spatial and temporal dimensions. A Channel-wise Topology Refinement Graph Convolution (ctr-gcn) is introduced in [5] with a depth-wise and dynamic graph convolution approach. Their model learns a channel-shared topology during training, which models the generic correlations of the joints. The novelty of their approach is that this topology gets refined with a dynamic channel-wise topology. The channel-wise topology is dynamically inferred for every sample, and models subtle correlations between keypoints within certain channels. For temporal modeling, four parallel branches of convolutions with different kernel sizes and dilation are used to model different temporal scales. Trivedi et al. [40] introduced in *psumnet* a unified modality part-based streaming approach, which makes use of four different skeleton modalities: joint, bone, joint-velocity and bone-velocity. Instead of having an own stream for each modality, they are concatenated on the channel dimension, resulting in input data with 12 channels for three-dimensional coordinates. This way, unified modality streams are also able to model correlations across different modalities.

3. Method

3.1. Challenges in Real-World Surveillance

Skeleton-based action recognition datasets are mostly collected in the laboratory in a controlled environment with near-perfect skeletons as input [11, 28, 32, 42]. Unfortunately, when the task is transferred to the real world, for example, in the area of surveillance using either UAV or static cameras, there are many challenges whether indoors or outdoors regarding generalization [9, 38]. In this context, pose estimators face important challenges [8], which can also be felt in more realistic AR datasets, as shown in Fig. 2.

Typically, the input to a skeleton-based action recognition model is a sequence of 2D skeletons generated by a detector, a tracker, and a top-down pose estimator. Since models for human pose estimation are very time-consuming, temporal upsampling with either interpolation or models can be used additionally [18, 52]. However, temporal upsampling can also introduce artifacts or amplify existing artifacts.

There are various types of errors that can occur in skeleton input. For example, imprecise person recognition may cut a person, resulting in missing body parts. A person may be missed, i.e. not recognized, or incorrectly tracked, resulting in repeated identity changes. In the wild, people are often partially or completely obscured by objects or other people. In such cases, the temporal upsampler may produce interpolation artifacts that let the movement appear less natural or realistic. In addition, individual keypoints may be swapped, or the pose estimator may even incorrectly infer the right and left sides of body parts.

Despite numerous inaccuracies and errors, this type of input is fed to action recognition models and is often expected to yield reliable results. However, state-of-the-art models in the literature are not at all prepared for this kind of data, and few works actually propose strategies to deal with it. In this work, we propose several data augmentation techniques for skeleton-based action recognition to remedy this situation.

3.2. Data Augmentation

We propose various augmentation techniques specifically for skeleton sequences. The idea of these augmentations is to introduce errors into the training data that the model may encounter in real-world applications. On the one hand, these augmentations aim to increase the robustness of the model to these errors. On the other hand, the augmentations should help against overfitting and improve generalization. In the following, several intuitive augmentation techniques are introduced and combined.

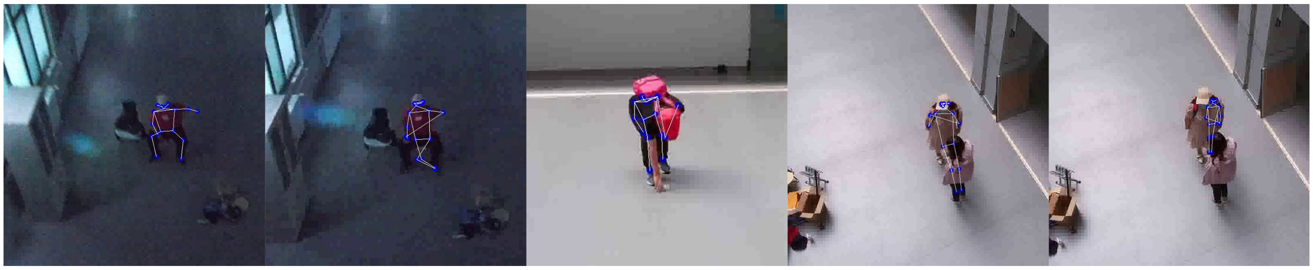

Figure 2. **Wrong skeleton annotation in UAVHuman [23] –** From left to right: a hand keypoint is placed in the air while the hand is on the body of the person; the knee and feet of the person are placed between his legs while the right hand is placed on the left; the predictor seems to detect the head of the person on the backpack; two persons are merged into a single skeleton; a skeleton is attached to one side of the person.

3.2.1 Occlusions

As previously proposed by Angelini et al. [1], the training data can be augmented by adding artificial occlusions to the skeleton sequences. However, instead of randomly occluding the same keypoints throughout a sequence, this work aims at an approach that better reflects the occlusions encountered in real-world scenarios. Inspired by Song et al. [37], multiple occlusion cases are used to augment the data. In contrast, the cases in this work are not mutually exclusive, but are applied simultaneously. While a pose-aware data augmentation method is proposed in [31], which is composed of a random global jitter on the whole skeleton followed by part-based local jitter and add noise to the training data, we aim to learn different invariances.

Instead of occluding selected keypoints throughout the entire sample, the approach of this work is to occlude the keypoints on a randomly selected subsequence only. This approach is closer to mimic real-world situations, as keypoints do not have to be occluded through the entire sequence. The minimum duration of artificial occlusions introduced into the sequence is selected to be 25 frames. This choice is based on the fact that short-time occlusions in the keypoint sequence can be reasonably reconstructed using interpolation [1]. As in previous works, occluded joints will be set to the origin [37].

The following occlusion augmentation cases are applied on a random subsequence with a random length between 25 and 100 frames:

1. Frame Occlusions: All keypoints in the subsequence are set to zero as in Algorithm 1. This simulates the loss of certain keyframes.

2. Random body part occlusions: Four groups of keypoints are created: left leg, right leg, left arm, and right arm. One of these body parts is then randomly selected and the keypoints of the selected body part are set to zero in the subsequence as in Algorithm 2. Since occlusions usually occur in a local area and not over the entire skeleton, this case aims to simulate this with the help of the body parts.

3. Random Keypoints Occlusion: Random keypoints according to $\mathcal{B}(p)$ are selected to be occluded in the subsequence as shown in Algorithm 3.

Algorithm 1: Frame Occlusions

```
Input: The skeleton sequence $P \in \mathbb{R}^{C\times T\times V}$ with C
       channels, T frames and V keypoints
size ← Random integer in $[25, 100]$
start ← Random integer in $[1, T - \text{size}]$
end ← start + size
for $i \in [start, end) \subseteq \mathbb{N}$ do
    $P[:, i, :] \leftarrow 0$
return $P$
```

Algorithm 2: Random Body Part Occlusions

```
Input: The skeleton sequence $P \in \mathbb{R}^{C\times T\times V}$ with C
       channels, T frames and V keypoints, body
       parts B
size ← Random integer in $[25, 100]$
start ← Random integer in $[1, T - \text{size}]$
end ← start + size
part ← Choose random $b \in B$
for $i \in [start, end) \subseteq \mathbb{N}$ do
    for $j \in part$ do
        $P[:, i, j] \leftarrow 0$
return $P$
```

In summary, three occlusion cases are constructed: Frame Occlusions, Body Part Occlusions, and Random Occlusions. Each of these cases can be assigned a probability, which determines how likely the given case is to be applied to the current skeleton sequence sample. When multiple cases are

Algorithm 3: Random Keypoint Occlusions

```
Input: The skeleton sequence $P \in \mathbb{R}^{C \times T \times V}$ with C
       channels, T frames and V keypoints, keypoint
       occlusion probability w
size ← Random integer in $[25, 100]$
start ← Random integer in $[1, T - \text{size}]$
end ← start + size
for $j \in [1, V] \subseteq \mathbb{N}$ do
    chance ← Random in $[0, 1) \subset \mathbb{R}$
    if $chance < w$ then
        for $i \in [start, end) \subseteq \mathbb{N}$ do
            $P[:, i, j] \leftarrow 0$
return $P$
```

applied simultaneously, each case is applied to its independently selected subsequence.

3.2.2 Interpolation

Short-term occlusions can be reasonably reconstructed using a simple interpolation [1]. However, this interpolation is not able to reconstruct the missing data exactly and differs from the original data. The idea in this section is to use short-term occlusion interpolation as another data augmentation strategy. On the one hand, the interpolation of short occlusions increases the data variety, and on the other hand, it aims to improve the robustness of the model to these interpolations. For each sample, two subsequences between 3 and 27 frames in length are randomly selected. The first and last frames are used for interpolation, while the rest of the inner frames are occluded. In the first subsequence, the entire skeleton is occluded, and in the second subsequence, only a random body part is occluded. The motivation here is that occlusions and their potential erroneous interpolation are not likely to occur on the entire skeleton, but only on parts of it. The body parts used are: left leg, right leg, left arm, right arm, and back. The occluded keypoints are then reconstructed by linear interpolation between the two outer frames (Algo. 4).

3.2.3 Keypoint swapping

Since most human pose estimation models provide predictions from a single image section, they may confuse keypoints with each other. To provide some robustness, we propose a simple augmentation case that aims to artificially reproduce this effect by randomly swapping two keypoints.

Based on this simple idea, this perturbation can be used for further data augmentation techniques. Here we propose another case in Algorithm 5 that is often encountered in real-world scenarios: randomly swapping all keypoints of legs or arms, e.g. mirroring the body parts.

Algorithm 4: Interpolation for COCO Topology

```
Input: The skeleton sequence $P \in \mathbb{R}^{C \times T \times V}$ with C
       channels, T frames and V keypoints, body
       parts B
$B = \{\{5, 7, 9\}, \{6, 8, 10\}, \{6, 8, 10\}, \{11, 13, 15\},$
$\{12, 14, 16\}, \{5, 6, 11, 12\}\}$
// Interpolate entire skeleton
size ← Random integer in $[3, 27]$
start ← Random integer in $[1, T - \text{size}]$
end ← start + size
for $i \in [start, end) \subseteq \mathbb{N}$ do
    weight ← $\frac{i - \text{start}}{\text{size}}$
    $P[:, i, :] \leftarrow P[:, start, :] * (1 - \text{weight}) + P[:$
      $, end, :] * \text{weight}$
// Interpolate only one body part
size ← Random integer in $[3, 27]$
start ← Random integer in $[1, T - \text{size}]$
end ← start + size
part ← Choose random $b \in B$
for $i \in [start, end) \subseteq \mathbb{N}$ do
    for $j \in part$ do
        weight ← $\frac{i - \text{start}}{\text{size}}$
        $P[:, i, j] \leftarrow P[:, start, j] * (1 - \text{weight}) + P[:$
          $, end, j] * \text{weight}$
return $P$
```

3.3. Skelbumentations

For implementation and further study of useful skeleton-based data augmentation techniques, we develop a Python library called *Skelbumentations* for skeleton sequence augmentation. Its principle is based on the popular image augmentation library Albumentations [2]. While Albumentations also offers the possibility to augment keypoints in addition to images, the operations are limited to image-related tasks such as cropping and blurring the image. Furthermore, Albumentations does not support the ability to augment sequences of images with keypoints. Therefore, we borrow concepts such as *select* and *compose* functions and implement our augmentation cases. To apply an operation to only a part of the skeleton sequence, different select operations can be used. An example of the simplicity of the augmentation pipeline is shown in Listing 1.

Behind the scenes, Skelbumentations keeps track of occluded keypoints with an invalid map. This invalid map can also be passed to the pipeline before augmentation in case the data already contains some occluded keypoints. By keeping track of occluded keypoints, Skelbumentations can block perturbations from changing occluded keypoints or ignore them when calculating keypoint velocities for high motion selection. The setting of occluded keypoints to the origin is done at the end of the pipeline and can be turned off. In

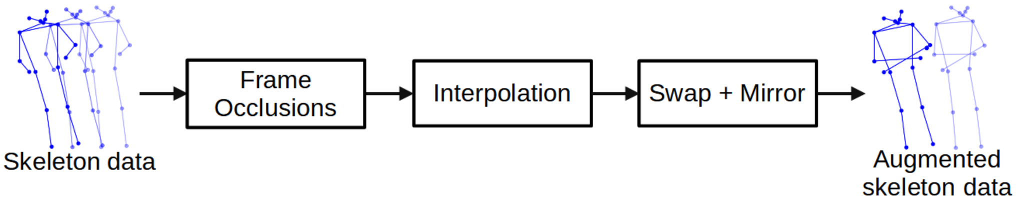

Figure 3. **Augmentation pipeline overview –** Schematic illustration of our agumentation pipeline. First a random block of 25 to 100 Frames is set as occluded. Then the interpolation augmentation is applied: first a block of whole frames is deleted and reconstructed by interpolation, second the same procedure is repeated on an other random block, this time only for a single body part. Then random keypoints are swapped for random frames in the sequence. Finally, for small random blocks of frames, mirror augmentation is applied.

```
Algorithm 5: Mirror keypoints
  Input: The skeleton sequence $P \in \mathbb{R}^{C \times T \times V}$ with C
         channels, T frames and V keypoints, opposite
         keypoints O
  Coco:
  $O_{\text{legs}} = \{(11, 12), (13, 14), (15, 16)\}, O_{\text{arms}} =$
  $\{(5, 6), (7, 8), (9, 10)\}$
  size ← Random integer in $[1, 4]$
  start ← Random integer in $[1, T - \text{size}]$
  end ← start + size
  for $i \in [start, end) \subseteq \mathbb{N}$ do
      for $(v1, v2) \in O_{legs}$ do
          temp ← $P[:, i, v1]$
          $P[:, i, v1] \leftarrow P[:, i, v2]$
          $P[:, i, v2] \leftarrow$ temp
  size ← Random integer in $[1, 4]$
  start ← Random integer in $[1, T - \text{size}]$
  end ← start + size
  for $i \in [start, end) \subseteq \mathbb{N}$ do
      for $(v1, v2) \in O_{arms}$ do
          temp ← $P[:, i, v1]$
          $P[:, i, v1] \leftarrow P[:, i, v2]$
          $P[:, i, v2] \leftarrow$ temp
```

this case, the user can decide how to deal with occluded keypoints using the invalid map, which is also returned by Skelbumentations next to the keypoints.

Finally, a possible augmentation pipeline combining the different techniques is illustrated in Fig. 3. First, occlusion and interpolation of a random body part and entire frames are performed, then swap and mirror augmentation are applied. The code of the library will be released upon publication on Github.

```python
import skelbumentations as S

pipeline = S.Compose([

    # Frame Occlusion
    S.SelectRandomFrames([
        S.WholeOcclusion()
    ], min_num=25, max_num=100),

    # Interpolation
    S.SelectRandomWithBorder([
        S.OneOf([
            S.SpecificOcclusion(arm_left),
            S.SpecificOcclusion(arm_right),
            # [...]
        ])
    ],[
        S.InterpolateOcclusions()
    ], min_num=1, max_num=25,),
    S.SelectRandomWithBorder([
        S.WholeOcclusion()
    ],[
        S.InterpolateOcclusions()
    ], min_num=1, max_num=25),

    # Swapping
    S.SelectRandomFrames([
        S.SwapPerturbation()
    ], contiguous=False, p=0.8, min_num=1,
    max_num=30),
    # [ ... ]

    # Mirroring
    S.SelectRandomFrames([
        S.MirrorPerturbation(opposite_points=
    opposite_legs)
    ], p=0.8, min_num=1, max_num=4),
    # [...]
])
```

Listing 1. An Augmentation pipeline of frame occlusions created with Skelbumentations. Every pipeline starts with the Compose class. In this pipeline a randomly positioned frame block of randomly 25 to 100 frames is selected and occluded of the skeleton sequence. Aftwerward interpolation swapping and mirroring augmentation are applied.

4. Experiments

4.1. Datasets and Experimental Settings

UAV-Human is a large-scale benchmark for UAV-based human behavior understanding. It contains 67,428 annotated video sequences with 155 action classes, 20 of which are gestures used to control the unmanned aerial vehicle. The videos were recorded during both day and night, indoors and outdoors. Unlike other datasets that were captured with fixed ground cameras, UAV-Human was captured with a flying drone at different speeds, altitudes, and trajectories. In addition to RGB video, other data modalities were collected such as night vision video, infrared sequences, and depth maps. 2D skeletons obtained with the RMPE pose estimator [13] are also provided. This is currently the largest dataset for action recognition with UAVs [23].

We implement the baselines using the PyTorch deep learning framework. All experiments are run on a single GPU, either an RTX 3090, A6000, or RTX Quadro 6000 GPU, depending on availability. The Stochastic Gradient Descent (SGD) optimizer is used with 0.9 momentum and 0.0005 weight decay for model training. The initial learning rate is set to 0.1 and decreases by a factor of 0.5 after a plateau. We train all models with 300 epochs and select the best performance. We use the 2D skeletons provided by the dataset and convert them to pseudo-3D with $z = 0$. All sequences are resized to 300 frames. We use the default weighting of each original method for the stream ensemble.

Similarly, we conduct experiments on the JHMDB dataset [17] which is a subset of HMDB [22] with 928 short videos with 21 action classes. With around 40 frames per video, all frames provide an approximative 2D skeleton ground-truth annotated with a puppet model that is fitted to the actor. In this case, we train the models on sequences of 40 frames for 600 epochs and adjusts the augmentations to fit the shorter sequences.

4.2. Ablation Study

We perform ablation studies using 2s-agcn [33] and evaluate the different augmentations on the CSV1 benchmark of the UAVHuman dataset to verify the effectiveness of our proposed methods. The results of the ablation studies are presented in Table 1. Empirical studies using Nesterov or AdamW did not provide better results and are not listed here.

Since research has shown that using the Exponential Moving Average (EMA) of a model's trainable parameters for evaluation can significantly improve the results [38, 39], we perform a first experiment with EMA. The results are improved by $+0.73\%$ and $+1.03\%$ for each stream, respectively, and by $+0.41\%$ to 44.16 for the ensemble. We then apply the frame augmentation and show an improvement for each stream of $+1.49\%$ and $+1.31\%$ respectively and $+1.40\%$ to 45.55 for the ensemble. The interpolation augmentation further improves the bone stream by $+0.92\%$, but the joint stream shows a loss of -0.29%. The ensemble results remain unchanged. Finally, we apply the Swapping and Mirroring augmentations, which provide another significant improvement of $+2.77\%$ and $+1.72\%$, respectively, and $+2.81\%$ to 48.36 for the ensemble.

Method	Joints Acc (%)	Bones Acc (%)	2-Streams Acc (%)
2s-agcn [33]	40.80	40.64	43.75
+ EMA	41.53(+0.73)	41.67(+1.03)	44.16(+0.41)
+ Frame Aug	43.02(+1.49)	42.98(+1.31)	45.55(+1.40)
+ Interpolation	42.73(−0.29)	43.90(+0.92)	45.55(+0.00)
+ Swap & Mirror	**45.5**(+2.77)	**45.62**(+1.72)	**48.36**(+2.81)

Table 1. **UAVHuman CSv1** – Top-1 Accuracy. The proposed augmentations applied to the baseline approach increase the performance significantly.

Method	CSv1 Acc (%)	CSv2 Acc (%)	Params (M)	MACS
2s-agcn [33]	43.75	71.31	7.56	25.38
4s-agcn	44.97	72.57	–	–
4s-ctr-gcn [5]	45.93	72.19	5.80	21.64
psumnet [40]	45.09	72.16	2.50	7.15
2s-agcn (ours)	48.36(+4.61)	73.53(+2.21)	7.56	25.38
4s-agcn (ours)	49.60(+4.63)	76.04(+3.47)	–	–
4s-ctr-gcn (ours)	**50.09**(+4.15)	**76.47**(+4.27)	5.80	21.64
psumnet (ours)	48.11(+3.02)	73.25(+1.09)	2.50	7.15

Table 2. **Benchmarking GCN skeleton-based action recognition algorithms on the UAVHuman benchmark.** Inputs are 2D (zero-padded to 3D) skeletons with 17 joints. We set the input length to 300, input person number to 2, and apply all augmentations introduced in Sec. 3.2.

This is an impressive total improvement of $+4.61\%$ over the baseline. This is probably mainly due to the rather poor quality of the skeletons provided by the dataset. However, such variable skeleton quality is a property of real-world action recognition and thus needs to be accounted for.

4.3. Benchmarking GCN Algorithms

We compare four representative GCN approaches: 2s-AGCN [33] and four stream variant for better comparibility, ctr-gcn [5], and psumnet [40]. We report the top-1 accuracy for the different methods with and without augmentations for both the CSV1 and CSV2 benchmarks in Table 2. The four methods show similiar improvement of more than $+4\%$ for the former. For the latter 2s-agcn and psumnet show an improvement of $+2.21\%$ and $+1.09\%$respectively, while 4s-agcn and 4s-ctr-gcn show improvements of $+3.47\%$ and $+4.27\%$ respectively. This is particularly noticeable, since both these models perform well without augmentations and seem to profite more from these.

Finally, we validate our observations with the 4s-agcn model on the JHMDB dataset. The dataset is composed of three splits which are then averaged. As shown in Table 3,

Method	Split1 Acc (%)	Split2 Acc (%)	Split3 Acc (%)	Average Acc (%)
4s-agcn	76.87	71.11	74.72	74.23
4s-agcn (ours)	**80.22**(+3.35)	**78.15**(+7.04)	**75.09**(+0.37)	**77.82**(+3.59)

Table 3. **Benchmarking 4s-agcn on the JHMDB benchmark.** Inputs are 2D (zero-padded to 3D) skeletons with 15 joints. We set the input length to 40, input person number to 1, and apply all augmentations introduced in Sec. 3.2 adapted to the sequence length.

our augmentations also improve the baseline considerably. It is to notice, that we did not search for optimal parameter for neither dataset, thus the results may improve with optimized hyperparameters.

While the proposed data augmentation pipeline provides significant improvement, the overall results are still insufficient compared to the performance of GCNs on lab data. Further work is needed on realistic uncontrolled scenarios.

5. Discussion of Potential Societal Implications

The field of skeleton-based action recognition, which includes human pose estimation as well as tracking, is related to visual surveillance and has several potential applications in real-world scenarios. Intended scenarios may include the use of retrograde action systems by law enforcement to detect violent crime. However, it is currently unclear how well human pose estimation models discriminate between clothing and skin color in low-resolution surveillance images, i.e., whether the model could provide less reliable skeletons leading to systematic misclassification. We believe that mitigation strategies should include robustness against such unreliable skeletons to reduce the impact of bias in datasets.

6. Conclusion

In this work, we have proposed Skelbumentations, a Python library that allows data augmentation for skeletal sequences. Furthermore, we propose several data augmentation techniques for skeleton-based action recognition. Our experiments on the largest UAV action recognition dataset show an impressive improvement over baselines without augmentation. Based on the skeleton augmentation results, we believe that the research community will develop new robust models for real-world skeleton-based action recognition.

References

[1] Federico Angelini, Zeyu Fu, Yang Long, Ling Shao, and Syed Mohsen Naqvi. 2d pose-based real-time human action recognition with occlusion-handling. *IEEE Transactions on Multimedia*, 22(6):1433–1446, 2020. 4, 5

[2] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020. 5

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[4] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 2

[5] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13359–13368, October 2021. 3, 7

[6] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5386–5395, 2020. 2

[7] Anoop Cherian, Basura Fernando, Mehrtash Harandi, and Stephen Gould. Generalized rank pooling for activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3222–3231, 2017. 2

[8] Mickael Cormier, Aris Clepe, Andreas Specker, and Jürgen Beyerer. Where are we with human pose estimation in real-world surveillance? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 591–601, January 2022. 2, 3

[9] Mickael Cormier, Andreas Specker, Julio C. S. Jacques Junior, Lucas Florin, Jürgen Metzler, Thomas B. Moeslund, Kamal Nasrollahi, Sergio Escalera, and Jürgen Beyerer. Upar challenge: Pedestrian attribute recognition and attribute-based person retrieval – dataset, design, and results. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 166–175, January 2023. 3

[10] Yan Dai, Xuanhan Wang, Lianli Gao, Jingkuan Song, and Heng Tao Shen. Rsgnet: Relation based skeleton graph network for crowded scenes pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1193–1200, 2021. 2

[11] Srijan Das, Rui Dai, Michal Koperski, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome: Real-world activities of daily living. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[12] Mahsa Ehsanpour, Fatemeh Saleh, Silvio Savarese, Ian Reid, and Hamid Rezatofighi. Jrdb-act: A large-scale dataset for spatio-temporal action, social group and activity detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[13] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 7

[14] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019. 2

[15] Thomas Golda, Johanna Thiemich, Mickael Cormier, and Jürgen Beyerer. For the sake of privacy: Skeleton-based salient behavior recognition. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3983–3987. IEEE, 2022. 2

[16] Pranay Gupta, Anirudh Thatipelli, Aditya Aggarwal, Shubh Maheshwari, Neel Trivedi, Sourav Das, and Ravi Kiran Sarvadevabhatla. Quo vadis, skeleton action recognition? *International Journal of Computer Vision*, May 2021. 2

[17] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *International Conf. on Computer Vision (ICCV)*, pages 3192–3199, Dec. 2013. 2, 7

[18] Kyung-Min Jin, Byoung-Sung Lim, Gun-Hee Lee, Tae-Kyung Kang, and Seong-Whan Lee. Kinematic-aware hierarchical attention network for human pose estimation in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5725–5734, 2023. 3

[19] M. E. Kalfaoglu, Sinan Kalkan, and Aydin Alatan. Late temporal modeling in 3D CNN architectures with bert for action recognition. *ArXiv*, abs/2008.01232, 2020. 2

[20] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 3

[21] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11977–11986, 2019. 2

[22] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011. 7

[23] Tianjiao Li, Jun Liu, Wei Zhang, Yun Ni, Wenqian Wang, and Zhiheng Li. Uav-human: A large benchmark for human behavior understanding with unmanned aerial vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16266–16275, June 2021. 2, 4, 7

[24] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *ECCV*, 2018. 2

[25] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[26] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. Tokenpose: Learning keypoint tokens for human pose estimation. *arXiv preprint arXiv:2104.03516*, 2021. 2

[27] Weiyao Lin, Huabin Liu, Shizhan Liu, Yuxi Li, Rui Qian, Tao Wang, Ning Xu, Hongkai Xiong, Guo-Jun Qi, and Nicu Sebe. Human in events: A large-scale benchmark for human-centric video analysis in complex events. *arXiv preprint arXiv:2005.04490*, 2020. 3

[28] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2684–2701, 2020. 2, 3

[29] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 143–152, 2020. 3

[30] Roberto Martin-Martin, Mihir Patel, Hamid Rezatofighi, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments. *IEEE transactions on pattern analysis and machine intelligence*, 2021. 3

[31] Anshul Shah, Shlok Mishra, Ankan Bansal, Jun-Cheng Chen, Rama Chellappa, and Abhinav Shrivastava. Pose and joint-aware action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3850–3860, January 2022. 4

[32] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016. 2, 3

[33] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3, 7

[34] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Transactions on Image Processing*, 29:9532–9545, 2020. 3

[35] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[36] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. 2

[37] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Richly activated graph convolutional network for robust skeleton-based action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5):1915–1925, 2021. 4

[38] Andreas Specker, Mickael Cormier, and Jürgen Beyerer. Upar: Unified pedestrian attribute recognition and person retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 981–990, January 2023. 3, 7

[39] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 7

[40] Neel Trivedi and Ravi Kiran Sarvadevabhatla. Psumnet: Unified modality part streams are all you need for efficient pose-based action recognition. In *European Conference on Computer Vision*, pages 211–227. Springer, 2022. 3, 7

[41] Edward Vendrow, Duy Tho Le, Jianfei Cai, and Hamid Rezatofighi. Jrdb-pose: A large-scale dataset for multi-person pose estimation and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3

[42] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning, and recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014. 3

[43] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2

[44] 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. In *European conference on computer vision*, 2016. 2

[45] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 2

[46] Philippe Weinzaepfel and Grégory Rogez. Mimetics: Towards understanding human actions out of context. *International Journal of Computer Vision*, 129(5):1675–1690, 2021. 2

[47] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[48] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018. 2

[49] Di Yang, Yaohui Wang, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, and Francois Bremond. Unik: A unified framework for real-world skeleton-based action recognition. *BMVC*, 2021. 2

[50] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11802–11812, 2021. 2

[51] Kaiyu Yue, Ming Sun, Yuchen Yuan, Feng Zhou, Errui Ding, and Fuxin Xu. Compact generalized non-local network. In *Advances in Neural Information Processing Systems*, 2018. 2

[52] Ailing Zeng, Xuan Ju, Lei Yang, Ruiyuan Gao, Xizhou Zhu, Bo Dai, and Qiang Xu. Deciwatch: A simple baseline for 10x efficient 2d and 3d pose estimation. In *European Conference on Computer Vision*. Springer, 2022. 3