




## Article

# Body-Pose-Guided Action Recognition with Convolutional Long Short-Term Memory (LSTM) in Aerial Videos

Sohaib Mustafa Saeed<sup>1</sup>, Hassan Akbar<sup>1,2</sup>, Tahir Nawaz<sup>1,2</sup> , Hassan Elahi<sup>1,\*</sup>  and Umar Shahbaz Khan<sup>1,2</sup> 

<sup>1</sup> Department of Mechatronics Engineering, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan; smsaeed.mts20ceme@student.nust.edu.pk (S.M.S.); hassanakbar2013@yahoo.com (H.A.); tahir.nawaz@ceme.nust.edu.pk (T.N.); u.shahbaz@ceme.nust.edu.pk (U.S.K.)

<sup>2</sup> National Centre of Robotics and Automation, Islamabad 44000, Pakistan

\* Correspondence: hassan.elahi@ceme.nust.edu.pk; Tel.: +92-51-54444402

**Abstract:** The accurate detection and recognition of human actions play a pivotal role in aerial surveillance, enabling the identification of potential threats and suspicious behavior. Several approaches have been presented to address this problem, but the limitation still remains in devising an accurate and robust solution. To this end, this paper presents an effective action recognition framework for aerial surveillance, employing the YOLOv8-Pose keypoints extraction algorithm and a customized sequential ConvLSTM (Convolutional Long Short-Term Memory) model for classifying the action. We performed a detailed experimental evaluation and comparison on the publicly available Drone Action dataset. The evaluation and comparison of the proposed framework with several existing approaches on the publicly available Drone Action dataset demonstrate its effectiveness, achieving a very encouraging performance. The overall accuracy of the framework on three provided dataset splits is 74%, 80%, and 70%, with a mean accuracy of 74.67%. Indeed, the proposed system effectively captures the spatial and temporal dynamics of human actions, providing a robust solution for aerial action recognition.

**Keywords:** deep neural network; convolutional LSTM; action recognition; body pose keypoints; aerial surveillance



**Citation:** Saeed, S.M.; Akbar, H.; Nawaz, T.; Elahi, H.; Khan, U.S. Body-Pose-Guided Action Recognition with Convolutional Long Short-Term Memory (LSTM) in Aerial Videos. *Appl. Sci.* **2023**, *13*, 9384. <https://doi.org/10.3390/app13169384>

Academic Editors: Sheng Du, Xiongbo Wan, Wei Wang and Hao Fu

Received: 5 July 2023

Revised: 12 August 2023

Accepted: 16 August 2023

Published: 18 August 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Action recognition involves automatically identifying and categorizing human actions in video sequences, which is highly beneficial and needed for surveillance applications [1–3]. Action recognition is, indeed, a challenging task due to the presence of various challenges, particularly background clutter occlusions and camera viewpoint [4–6]. Conventional action recognition methods involved hand-crafted feature extraction [7,8], based on manually representing actions, such as motion, shape, or appearance descriptors. The limitations of this approach lie in the fact that hand-crafted features may not be able to effectively capture complex temporal relationships or variations in different action scenarios. Indeed, designing effective features could be challenging; plus, they may not generalize well to different datasets or action classes.

The 3D CNNs extend the concept of traditional 2D CNNs [9,10] to incorporate temporal information by processing video frames as 3D volumes. They, however, require a large amount of training data and computational resources. Additionally, they may struggle with long-term temporal dependencies or capturing fine-grained motion details. Moreover, the training of 3D CNNs from scratch can be challenging due to the limited availability of annotated video datasets.

Recurrent neural networks (RNNs), gated recurrent unit (GRU), or LSTM [11,12] model temporal dependencies by maintaining internal memory states. However, RNNs may struggle with modeling long-term dependencies or capturing complex spatial dynamics.

They could be sensitive to the order and timing of actions within sequences. RNNs are computationally intensive, especially for longer sequences.

Two-stream networks [13,14] consist of the spatial stream (CNN for appearance) as well as the temporal stream (CNN or RNN for motion). They require synchronized and aligned RGB and optical flow inputs, which could be challenging to obtain in practice. Combining the information from two streams can introduce additional complexity and potential performance degradation.

Graph convolutional networks (GCNs) [15,16] represent actions as graphs and exploit graph convolution operations to capture spatial and temporal relationships between body joints or keypoints. However, GCNs rely heavily on accurate and reliable detection and the tracking of skeletal keypoints and also have limitations when dealing with occlusions or missing keypoints in complex action scenarios. Designing appropriate graph structures and defining graph convolution operations are, inevitably, challenging.

The recent introduction of vision transformers has proved to be more efficient in accuracy. There are approaches that utilize transformers for action recognition [13,17,18] in complex scenarios; however, they are generally computationally more resource-consuming.

Aerial videos provide a comprehensive view [5] of the scene, enabling surveillance operators to monitor larger areas and detect events that may otherwise be overlooked. Action recognition from aerial scenarios, however, requires reliable detection of the target in complex backgrounds, with varying camera angle altitudes for an accurate classification of the action [19–22].

Malik et al. [23] proposed a method that relied on extracting 2D skeletal data using OpenPose that are then fed into LSTM for training and testing. Their framework was, however, validated in an indoor multi-view scenario and may not be directly deployable for aerial videos.

Another limitation in human action recognition is that the trained models generally misclassify when provided with unannotated data from new users [24], even after being trained on a large amount of data. This challenge arises as it is impractical to collect data for every new user. Yang et al. [25] aimed to address this problem by presenting a semi-supervised learning action recognition method for training on labeled as well as unlabeled data but not primarily for the aerial camera settings that are under consideration in this paper.

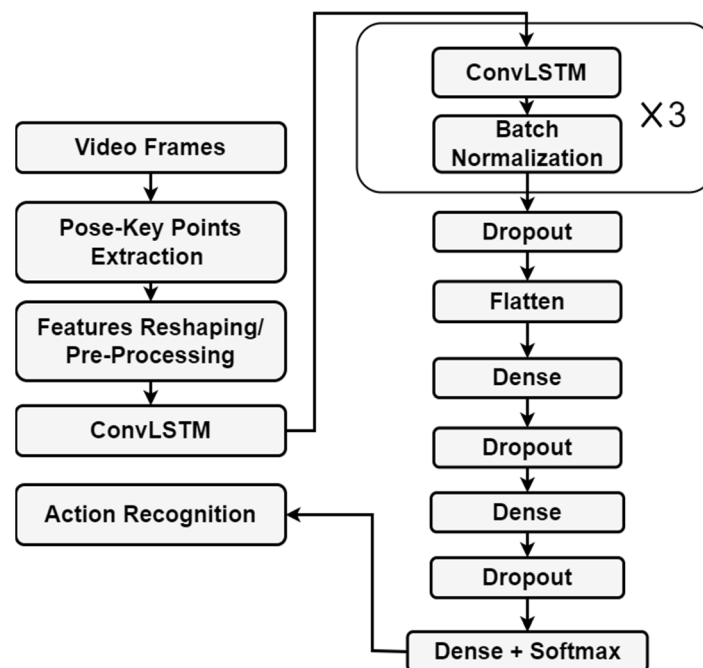
Dai et al. [26] introduced a dual-stream attention-based LSTM containing a visual attention mechanism that enables selectively focusing on key elements in the image frames by applying varying levels of attention to each individual deep feature map's output. The deep feature correlation layer embedded in their framework is, indeed, relevant to our work, and it contributes towards enhancing the robustness of the action recognition. The validation in [26] was, however, in experimental scenarios, different from that considered in this work.

Unlike the existing related methods reviewed above, the proposed research combines the robust pose detection ability of YOLOv8 with temporal sequencing ability of the ConvLSTM to propose an effective and efficient approach aimed specifically at aerial action recognition. In fact, the proposed framework offers a reliable recognition of human actions from an aerial perspective by utilizing the convolutional LSTM's capacity to parse temporal sequences. Specifically, the proposed method extracts the body pose keypoints from the frames and classifies actions at the frame level utilizing the customized convolutional LSTM network model. The reason behind relying on the extraction of the target body pose or keypoints is the lower computational cost as compared to the spatial features. Moreover, we use the LSTM network due to its demonstrated effectiveness for sequential data classification [23,24,27,28]; plus, it is not well explored in the literature for the problem under consideration. We showed the effectiveness of the proposed method in terms of encouraging performance accuracy and computational cost when compared on a public dataset (containing a wide range of action types) with several existing related approaches.

The organization of the paper is as follows. The proposed method for action recognition is described in Section 2. Section 3 provides details of the experimental results and analysis, which is followed by the conclusions in Section 4.

## 2. Proposed Action Recognition Method

We employed the YOLOv8 pose detection model for the extraction of 17 body keypoints. The extracted keypoints are then passed to the second stage, which is ConvLSTM, to extract spatiotemporal features across the sequence. The sequence length of 30 frames, chosen empirically, is set for the extraction of temporal information. The intuition behind incorporating the body pose with ConvLSTM is a selection of suitable features that are keypoints and performing the memory-based sequence classification using LSTM. Figure 1 illustrates the proposed human action recognition system.



**Figure 1.** Block diagram illustrating different steps involved in the proposed action system.

The architecture in Figure 1 is designed to process raw keypoints for the analysis of both spatial and temporal aspects. The ConvLSTM architecture shown in Figure 1 is made up of multiple hidden layers that work together to collect spatial and temporal features from frames. For an accurate classification of actions, this extracted feature set is essential. These characteristics ultimately influence how the recognized action is predicted, enabling the system to efficiently analyze actions occurring in successive frames.

Convolutional layers are used in the context indicated above to extract significant features from the body pose keypoints. Convolutional layers apply filters to the keypoints in order to capture significant spatial characteristics, such as the placement of body parts and their interactions. These filters help in finding patterns and correlations among the keypoints.

The network can automatically learn hierarchical representations of the body positions using convolutional layers. The network's capacity to recognize and accurately classify various activities within the video sequences is greatly aided by the extracted characteristics.

To accurately capture the temporal dynamics of activities throughout a series of frames, the use of LSTM is crucial. LSTMs effectively capture patterns and changes that emerge over time by processing the retrieved features or representations from each frame. LSTMs give the network the ability to comprehend how actions develop and classify by keeping track of past frames and taking into account how they affected the current frame.

### 2.1. Pose Extraction

The YOLOv8 pose extractor is a popular deep learning-based algorithm for keypoint detection. There are several other approaches that can be utilized for this purpose, but the latest YOLOv8 is known to be more efficient in accuracy as well as computationally [29]. Figure 2 shows the output of the pose extractor.



**Figure 2.** Results of YOLOv8 pose estimator on Drone Action dataset [22]: stabbing (top left), hitting stick (top right), waving hands (bottom left), and clapping (bottom right).

The keypoint coordinates for a given video can be represented as a  $(F, Kp)$ , where  $Kp$  represents the keypoints of the image and  $F$  represents the number of sequential frames or sequence length, which, in our case, is set to 30. The extracted keypoints are made to be aligned with the input of the next stage.

To extract spatiotemporal features from the video sequence, we stack the keypoint tensors for a given person over time. Let  $Kp_t$  be the keypoint tensor for the person at time  $t$  and let  $Kp_1, Kp_2, \dots, Kp_t$  be the keypoint tensors for the person over  $T$  frames of the video sequence. We stack these tensors along the time dimension to obtain a tensor  $P$  with dimensions  $(F, Kp_t)$ :

$$P = [Kp_1, Kp_2, Kp_3, \dots, Kp_T] \quad (1)$$

The YOLOv8 algorithm uses a fully convolutional neural network (FCN) to predict a heat map for each keypoint, which can be used to estimate the pose of the person in the video. The resulting output yields 17 keypoint coordinates for each detected person at the frame level across the video sequence.

### 2.2. Customized Convolutional LSTM Model

We used the LSTM model for action classification in aerial videos. The LSTM model is a type of RNN that can effectively encapsulate the dependencies of the sequential data. In the proposed approach, we first extract the temporal features from the aerial videos using the YOLOv8 pose extractor and then use the LSTM model to classify the actions based on these features. The tensor  $P$  in Equation (1) represents the spatiotemporal features of the person over time.

The LSTM model contains a memory cell and three gates, including an input gate, output gate, and forget gate [30–32], defined as follows:

*Input gate:*

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$

*Forget gate:*

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (3)$$

Output gate:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{4}$$

Memory cell:

$$C_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{5}$$

Output:

$$h_t = o_t \cdot \tanh(c_t) \tag{6}$$

where  $x_t$ ,  $h_t$ , and  $c_t$  denote the input, the output, and the cell state  $t$ , respectively.  $i_t$ ,  $f_t$ , and  $o_t$  are the input, the forget, and the output gates, respectively.  $W_i$ ,  $W_f$ ,  $W_o$ , and  $W_c$  refer to the collection of weight matrices used to transform the input data at each time step, whereas  $U_i$ ,  $U_f$ ,  $U_o$ , and  $U_c$  are the weight matrices to transform the hidden state from the previous time step.  $b_i$ ,  $b_f$ ,  $b_o$ , and  $b_c$  represent the bias terms.

The output  $H$  is a sequence of hidden states that captures the temporal dependencies in the spatiotemporal features. We can then use the final hidden state of the LSTM as input to a fully connected layer with softmax activation to obtain the probability distribution across the different action classes:

$$P = \text{Softmax}(W_h H + b) \tag{7}$$

We designed the custom sequential LSTM model by stacking three ConvLSTM 1D layers, such that each layer is followed by a batch normalization layer, with decreasing filter sizes of 128, 32, and 16, respectively. We added a dropout layer after the third ConvLSTM1D layer to prevent overfitting. Next, we flattened the output and added two fully connected layers with ReLU activation and a dropout layer after each. Finally, we added a dense output layer with the softmax activation function. The LSTM model applied in this research is convolutional LSTM (Figure 3), which combines convolutional layers with LSTM to model spatiotemporal data.

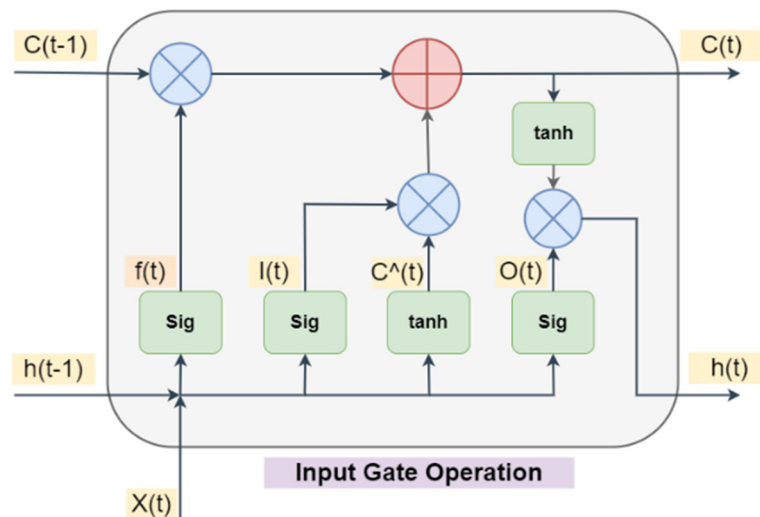


Figure 3. Structure of the convolutional LSTM.

### 3. Experimental Results and Analysis

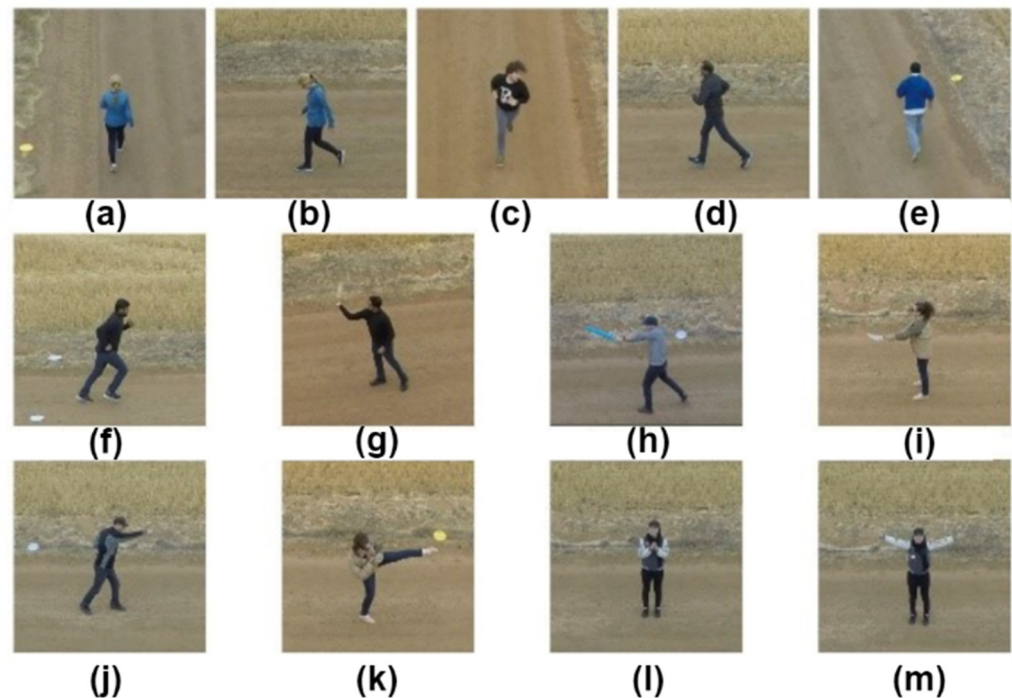
This section first describes the dataset in Section 3.1, which is followed by an evaluation of the results in Section 3.2 and performance comparisons with existing related approaches in Section 3.3.

#### 3.1. Dataset

We used the publicly available Drone Action dataset for evaluation [22]. This dataset comprises 240 videos that run for a total duration of approximately 44.6 min, embodying



66,919 frames and containing 13 distinct human action classes. The videos were captured from a low-altitude and slow-moving drone to ensure the details of body pose were reliably extracted. The complexity of this dataset is augmented by the diversity in body size, camera motion, varying target speed, and background clutter, making it a suitable benchmark for human action recognition studies. Figure 4 shows representative frames from the dataset for each action class [22].



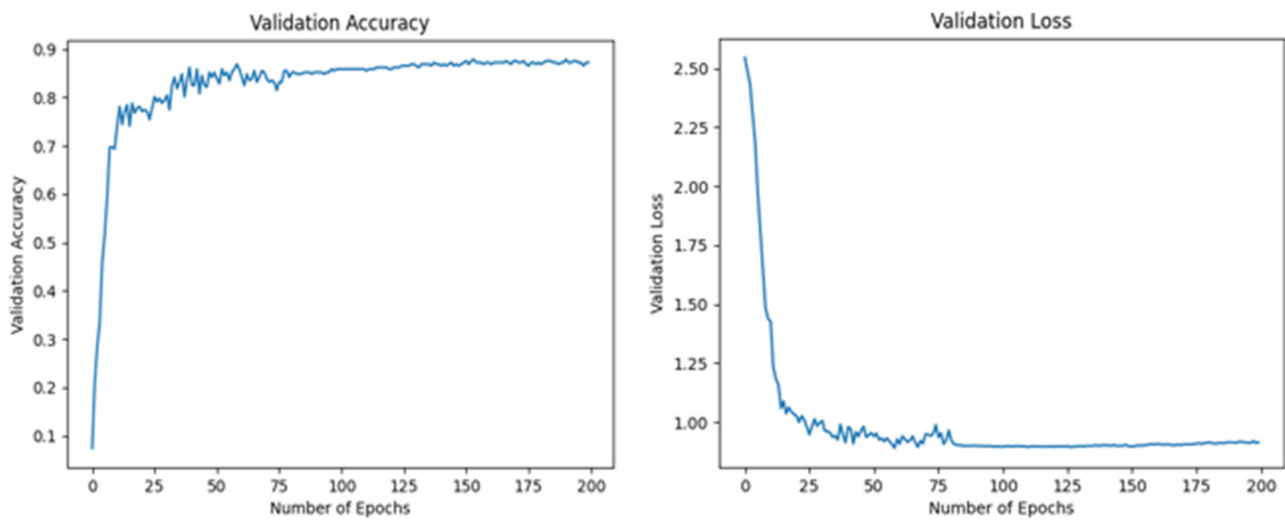
**Figure 4.** Representative frames from each class of the Drone Action dataset [22]: (a) walking front\_back, (b) walking side, (c) jogging front\_back, (d) jogging side, (e) running front\_back, (f) running side, (g) hitting with bottle, (h) hitting with stick, (i) stabbing, (j) punching, (k) kicking, (l) clapping, (m) waving hands.

### 3.2. Evaluation of Results

The proposed action recognition framework for aerial videos demonstrates an improved accuracy and robustness. Indeed, the combination of the YOLOv8-Pose algorithm and customized sequential convolutional LSTM model effectively captures the spatial and temporal information of actions, leading to an encouraging action recognition performance. The proposed model is trained and tested separately on the three dataset splits, as provided by the original paper [22]. In each split, 70% data are used for training and 30% for testing. The training was carried out for 200 epochs (chosen empirically), and network parameters were kept the same for training and testing for each split of the data. Table 1 lists the corresponding values for the validation loss and accuracy on all three splits. A representative graphical representation of the validation loss and validation accuracy is shown in Figure 5 for Split 1.

**Table 1.** Training details for the 3 splits.

Dataset	Epochs	Validation Loss	Validation Accuracy
Split 1	200	2.75–0.25	0.05–0.88
Split 2	200	2.55–1.00	0.13–0.83
Split 3	200	2.58–1.00	0.12–0.82



**Figure 5.** Plots for the validation accuracy (left) and validation loss (right) during the training for Split 1.

The overall accuracies achieved on Split 1, Split 2, and Split 3 are 74%, 80%, and 70%, respectively. The corresponding confusion matrices are provided in Figures 6–8, respectively. The class-wise results on each split are given in Tables 2–4, respectively, based on the standard well-known evaluation measures, which are precision, recall, and F1-score.

Analyzing the results in more detail, we observe that some actions had consistently high precision, recall, and F1-score values across all dataset splits. For instance, the actions “Clap”, “Kick”, “Walk\_fb”, “Walk\_side”, and “Wave\_hands” achieved high scores on all three splits. This suggests that the proposed framework is highly effective in recognizing these actions, even when presented with variations in the data. The high accuracy in these classes can be attributed to the combination of YOLO-Pose and the custom-designed ConvLSTM network, which allows for an efficient extraction of spatial and temporal information in video frames.

**Table 2.** Performance evaluation of the proposed method on all action types based on precision, recall, and F1-score on Split 1.

Action	Precision	Recall	F1-Score
Clap	1.00	1.00	1.00
Hit_botl	0.19	0.14	0.16
Hit_stick	0.65	0.64	0.65
Jogging	0.73	0.88	0.80
Jog_side	0.91	0.89	0.90
Kick	0.99	1.00	0.99
Punch	0.91	0.99	0.95
Run_fb	0.50	0.40	0.44
Run_side	0.86	0.89	0.87
Stab	0.29	0.40	0.34
Walk_fb	1.00	0.90	0.95
Walk_side	1.00	1.00	1.00
Wave_hands	0.98	1.00	0.99
<b>Average</b>	<b>0.77</b>	<b>0.78</b>	<b>0.77</b>

**Table 3.** Performance evaluation of the proposed method on all action types based on precision, recall, and F1-score on Split 2.

Action	Precision	Recall	F1-Score
Clap	1.00	1.00	1.00
Hit_botl	0.50	0.36	0.42
Hit_stick	0.72	0.78	0.75
Jog_fb	0.83	0.91	0.87
Jog_side	0.86	0.98	0.91

Table 3. Cont.

Action	Precision	Recall	F1-Score
Kick	0.99	0.92	1.00
Punch	0.76	0.99	0.83
Run_fb	0.73	0.53	0.62
Run_side	1.00	0.76	0.86
Stab	0.40	0.55	0.46
Walk_fb	1.00	1.00	1.00
Walk_side	0.98	0.98	0.98
Wave_hands	0.97	1.00	0.99
<b>Average</b>	<b>0.83</b>	<b>0.83</b>	<b>0.82</b>

Table 4. Performance evaluation of the proposed method on all action types based on precision, recall, and F1-score on Split 3.

Action	Precision	Recall	F1-Score
Clap	1.00	0.89	0.94
Hit_botl	0.33	0.29	0.31
Hit_stick	0.59	0.68	0.64
Jog_fb	0.67	0.61	0.63
Jog_side	0.85	0.58	0.69
Kick	0.99	0.85	0.92
Punch	0.83	0.95	0.84
Run_fb	0.28	0.33	0.30
Run_side	0.45	0.77	0.57
Stab	0.37	0.39	0.38
Walk_fb	0.91	0.95	0.93
Walk_side	0.96	1.00	0.98
Wave_hands	1.00	1.00	1.00
<b>Average</b>	<b>0.71</b>	<b>0.71</b>	<b>0.70</b>

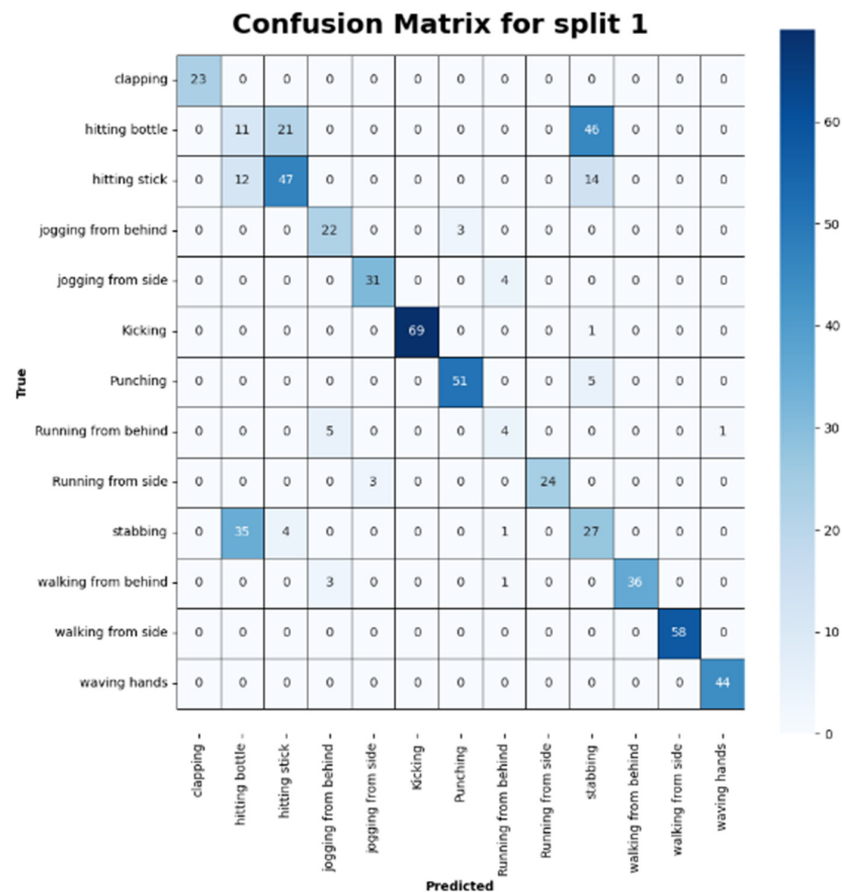


Figure 6. Confusion matrix for Split 1.



On the other hand, some actions demonstrated lower precision, recall, and F1-score values. For example, the “Hit\_botl” action achieved lower scores on all the three splits, with the lowest F1-score being 0.16 on Split 1. Similarly, the “Stab” action had an F1-score of 0.34 on Split 1, 0.46 on Split 2, and 0.38 on Split 3.

The lower performance for these actions (Hit\_botl, Stab) could be attributed to the higher complexity of the movements and the similarity of these actions with each other and some other classes, making it challenging for the proposed framework to differentiate them from others. Moreover, factors, such as background clutter and variation in viewpoint, could further hinder the recognition of these actions.

It is worth mentioning that there is performance variation for some actions across different splits. For instance, the “Hit\_stick” action had an F1-score of 0.65 on Split 1, which increased to 0.75 on Split 2 and then decreased slightly to 0.64 in Split 3. This observation suggests that the performance of the proposed framework is sensitive to the choice of training and testing data.

We also calculated the computational performance of the proposed method. The evaluation was performed in terms of the number of network parameters (in millions) and the number of floating-point operations (FLOPS) (in millions) and the classification time for the proposed customized ConvLSTM network. We practically implemented this model on Intel(R) Core(TM) i5-8250U CPU @ 1.80 GHz with 8.00 GB RAM. The total number of FLOPS was 36.79 million, with 1.03 million trainable parameters. The classification time for the 612 test sequences with 30 frames each on Split 1 was 3.58 s. The per sequence classification time was 5.457 milliseconds. This suggests that the proposed method is lightweight in terms of computational complexity and could be deployable in real-world applications.

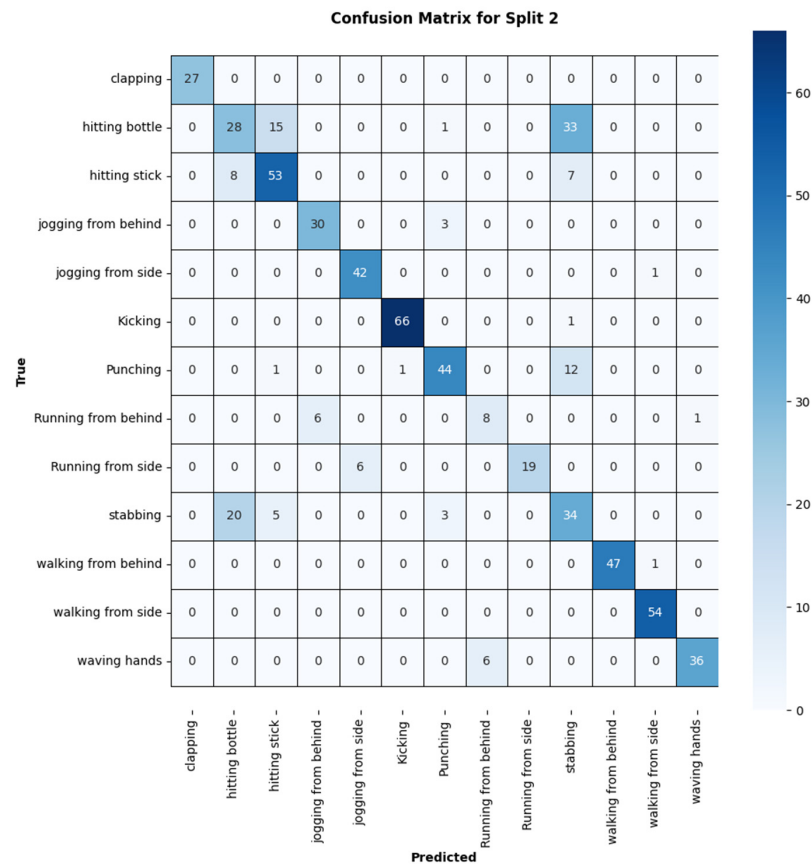


Figure 7. Confusion matrix for Split 2.

### 3.3. Performance Comparison with Related Approaches

We also compared the performance of the proposed action recognition framework with two existing approaches, as reported in the benchmark paper [22] (see Table 5). The

benchmark paper provides an analysis of the classification accuracy of two methods, including the high-level pose features (HLPFs) method and the pose-based convolutional neural networks (P-CNNs) method. The high-level pose features (HLPFs) method uses skeletal information from human poses to represent actions. In P-CNN, at each frame of a video, descriptors are extracted from the body regions. These descriptors encode relevant information, such as motion flow patterns and visual characteristics of the regions, leading to two-streamed information. Over time, these descriptors are aggregated, combining the information from multiple frames, to form a video descriptor. The proposed method shows better or comparable performance as compared to these existing methods (Table 5), owing to its capability to efficiently model temporal information and long-term dependencies in action sequences.

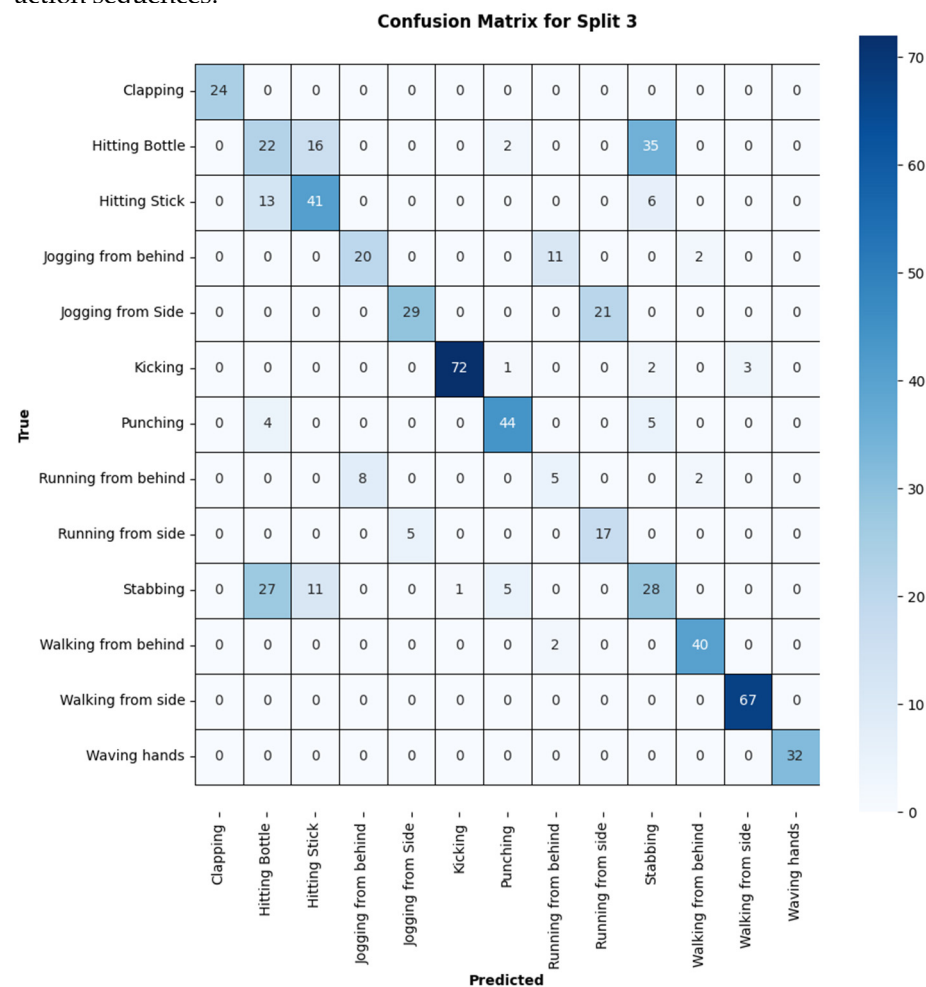


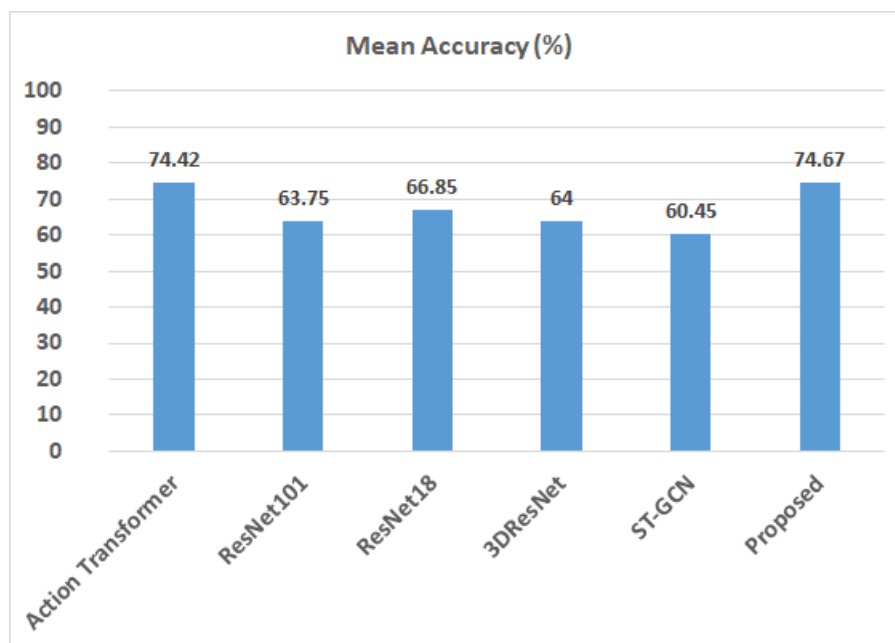
Figure 8. Confusion matrix for Split 3.

Table 5. Comparison of the proposed method with existing approaches on Drone Action dataset.

Method	Accuracy (Split 1)	Accuracy (Split 2)	Accuracy (Split 3)	Mean Accuracy
HLPF	63.89%	68.09%	61.11%	64.36%
P-CNN	72.22%	81.94%	73.61%	75.92%
Pose+ LSTM	74.00%	80.00%	70.00%	74.67%

For a more detailed performance comparison of the proposed approach with other models, we investigated several state-of-the-art deep learning models, such as Action Transformer, ResNet18, ResNet101, 3D ResNet, and ST-GCN. Action Transformer [33] has recently been employed for human action recognition. For evaluation, we set the

corresponding parameters as follows, heads: 1, layers: 4, embedding dimensions: 64, MLP: 256, and encoder layers: 5. The reason to keep the parameters at a minimum is to reduce the computational complexity of the model for the application at hand. ResNet is a specific configuration of the architecture that consists of 101 layers in the case of ResNet101 and 18 layers in ResNet18. The network includes residual blocks, which are designed to learn residual mappings that help mitigate the vanishing gradient issue. Each residual block contains multiple convolutional layers and shortcut connections that allow information to flow more effectively through the network. ResNet networks have been widely used for several computer vision tasks [34]. Further, 3D ResNet is an extension of the ResNet architecture designed to tackle video action recognition tasks by considering both spatial and temporal features in videos [28]. It adds a temporal dimension to the standard ResNet architecture, making it well suited for analyzing sequences of frames in videos. Thus, 3D ResNet takes advantage of this temporal aspect by incorporating 3D convolutional layers. These layers consider the spatial relationships within each frame as well as the temporal relationships between consecutive frames, enabling the network to capture motion patterns and changes over time. Finally, the Spatio-Temporal Graph Convolutional Network (ST-GCN) [35] is also a useful architecture used in video action detection applications, especially for addressing the spatial and temporal features present in films. In order to capture both spatial correlations within individual frames and temporal dependencies between successive frames, ST-GCN uses graph convolutional procedures. For evaluation, we replaced the proposed ConvLSTM with each of the above-mentioned models and accordingly trained and tested them on the same lines for all the three splits of the dataset. Figure 9 presents the performance comparison of the proposed approach with these models in terms of the mean accuracy across the three splits. It is clear that the proposed method outperforms all these related approaches, which further validates its effectiveness.



**Figure 9.** Performance comparison of the proposed method with existing related approaches in terms of mean accuracy across the three splits of the dataset.

#### 4. Conclusions

In this paper, we presented a convolutional LSTM-based model for human action recognition, which was built on the extracted target pose information using YOLOv8 to effectively encode the unique body movements for various action types. The proposed framework aimed to address the challenges associated with aerial action recognition, such

as varying viewpoints and background clutter. The study was inspired by the growing interest in drone applications and the need for robust and efficient action recognition methods for various applications, including security and surveillance. The comparisons with numerous existing methods show very encouraging performance through the proposed method. While the proposed framework can effectively classify the single person action in low-altitude aerial video sequences, in future work, the framework could be adapted to classify actions involving multiple objects.

**Author Contributions:** Conceptualization, S.M.S., H.A. and T.N.; methodology, S.M.S. and T.N.; software, S.M.S. and H.A.; validation, S.M.S. and H.A.; formal analysis, S.M.S. and H.A.; investigation, S.M.S. and H.A.; resources, T.N. and H.E.; writing—original draft preparation, S.M.S. and H.A.; writing—review and editing, T.N., H.E. and U.S.K.; supervision, T.N., H.E. and U.S.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially supported by the Higher Education Commission of Pakistan and the National Centre of Robotics and Automation under Grant DF 1009-0031.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

## References

1. Kumar, R.; Tripathi, R.; Marchang, N.; Srivastava, G.; Gadekallu, T.R.; Xiong, N.N. A secured distributed detection system based on IPFS and blockchain for industrial image and video data security. *J. Parallel Distrib. Comput.* **2021**, *152*, 128–143. [CrossRef]
2. Shorfuzzaman, M.; Hossain, M.S.; Alhamid, M.F. Towards the sustainable development of smart cities through mass video surveillance: A response to the COVID-19 pandemic. *Sustain. Cities Soc.* **2021**, *64*, 102582. [CrossRef] [PubMed]
3. Kashef, M.; Visvizi, A.; Troisi, O. Smart city as a smart service system: Human-computer interaction and smart city surveillance systems. *Comput. Hum. Behav.* **2021**, *124*, 106923. [CrossRef]
4. Özyer, T.; Ak, D.S.; Alhaji, R. Human action recognition approaches with video datasets—A survey. *Knowl.-Based Syst.* **2021**, *222*, 106995. [CrossRef]
5. Sultani, W.; Shah, M. Human Action Recognition in Drone Videos Using a Few Aerial Training Examples. *arXiv* **2021**, arXiv:1910.10027. Available online: <http://arxiv.org/abs/1910.10027> (accessed on 15 June 2023).
6. Wang, X.; Xian, R.; Guan, T.; de Melo, C.M.; Nogar, S.M.; Bera, A.; Manocha, D. AZTR: Aerial Video Action Recognition with Auto Zoom and Temporal Reasoning. *arXiv* **2023**, arXiv:2303.01589. Available online: <http://arxiv.org/abs/2303.01589> (accessed on 15 June 2023).
7. Hejazi, S.M.; Abhayaratne, C. Handcrafted localized phase features for human action recognition. *Image Vis. Comput.* **2022**, *123*, 104465. [CrossRef]
8. El-Ghaish, H.; Hussein, M.; Shoukry, A.; Onai, R. *Human Action Recognition Based on Integrating Body Pose, Part Shape, and Motion*; IEEE Access: Piscataway, NJ, USA, 2018; pp. 49040–49055. [CrossRef]
9. Arunehru, J.; Chamundeeswari, G.; Bharathi, S.P. Human Action Recognition using 3D Convolutional Neural Networks with 3D Motion Cuboids in Surveillance Videos. *Procedia Comput. Sci.* **2018**, *133*, 471–477. [CrossRef]
10. Sánchez-Caballero, A.; de López-Diz, S.; Fuentes-Jimenez, D.; Losada-Gutiérrez, C.; Marrón-Romera, M.; Casillas-Pérez, D.; Sarker, M.I. 3DFCNN: Real-time action recognition using 3D deep neural networks with raw depth information. *Multimed Tools Appl.* **2022**, *81*, 24119–24143. [CrossRef]
11. Sánchez-Caballero, A.; Fuentes-Jiménez, D.; Losada-Gutiérrez, C. Real-time human action recognition using raw depth video-based recurrent neural networks. *Multimed Tools Appl.* **2023**, *82*, 16213–16235. [CrossRef]
12. Muhammad, K.; Mustaqeem; Ullah, A.; Imran, A.S.; Sajjad, M.; Kiran, M.S.; Sannino, G.; de Albuquerque, V.H.C. Human action recognition using attention based LSTM network with dilated CNN features. *Future Gener. Comput. Syst.* **2021**, *125*, 820–830. [CrossRef]
13. Xiao, S.; Wang, S.; Huang, Z.; Wang, Y.; Jiang, H. Two-stream transformer network for sensor-based human activity recognition. *Neurocomputing* **2022**, *512*, 253–268. [CrossRef]
14. Zhao, Y.; Man, K.L.; Smith, J.; Siddique, K.; Guan, S.-U. Improved two-stream model for human action recognition. *EURASIP J. Image Video Process.* **2020**, *2020*, 24. [CrossRef]
15. Ahmad, T.; Jin, L.; Zhang, X.; Lai, S.; Tang, G.; Lin, L. Graph Convolutional Neural Network for Human Action Recognition: A Comprehensive Survey. *IEEE Trans. Artif. Intell.* **2021**, *2*, 128–145. [CrossRef]

16. Feng, L.; Zhao, Y.; Zhao, W.; Tang, J. A comparative review of graph convolutional networks for human skeleton-based action recognition. *Artif. Intell. Rev.* **2022**, *55*, 4275–4305. [[CrossRef](#)]
17. 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; IEEE: New Orleans, LA, USA, 2022; pp. 14043–14053. [[CrossRef](#)]
18. Wang, X.; Zhang, S.; Qing, Z.; Shao, Y.; Zuo, Z.; Gao, C.; Sang, N. OadTR: Online Action Detection with Transformers. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; IEEE: Montreal, QC, Canada, 2021; pp. 7545–7555. [[CrossRef](#)]
19. Barekattain, M.; Martí, M.; Shih, H.-F.; Murray, S.; Nakayama, K.; Matsuo, Y.; Prendinger, H. Okutama-Action: An Aerial View Video Dataset for Concurrent Human Action Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 2153–2160. [[CrossRef](#)]
20. Liu, C.; Szirányi, T. Real-Time Human Detection and Gesture Recognition for On-Board UAV Rescue. *Sensors* **2021**, *21*, 2180. [[CrossRef](#)]
21. Mliki, H.; Bouhleb, F.; Hammami, M. Human activity recognition from UAV-captured video sequences. *Pattern Recognit.* **2020**, *100*, 107140. [[CrossRef](#)]
22. Perera, A.G.; Law, Y.W.; Chahl, J. Drone-Action: An Outdoor Recorded Drone Video Dataset for Action Recognition. *Drones* **2019**, *3*, 82. [[CrossRef](#)]
23. Malik, N.U.R.; Abu-Bakar, S.A.R.; Sheikh, U.U.; Channa, A.; Popescu, N. Cascading Pose Features with CNN-LSTM for Multiview Human Action Recognition. *Signals* **2023**, *4*, 40–55. [[CrossRef](#)]
24. Yang, S.-H.; Baek, D.-G.; Thapa, K. Semi-Supervised Adversarial Learning Using LSTM for Human Activity Recognition. *Sensors* **2022**, *22*, 4755. [[CrossRef](#)]
25. Kumar, A.; Rawat, Y.S. End-to-End Semi-Supervised Learning for Video Action Detection. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; IEEE: New Orleans, LA, USA, 2022; pp. 14680–14690. [[CrossRef](#)]
26. Dai, C.; Liu, X.; Lai, J. Human action recognition using two-stream attention based LSTM networks. *Appl. Soft Comput.* **2020**, *86*, 105820. [[CrossRef](#)]
27. Mathew, S.; Subramanian, A.; Pooja, S. Human Activity Recognition Using Deep Learning Approaches: Single Frame CNN and Convolutional LSTM. *arXiv* **2023**, arXiv:2304.14499.
28. Zhang, J.; Bai, F.; Zhao, J.; Song, Z. Multi-views Action Recognition on 3D ResNet-LSTM Framework. In Proceedings of the 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Nanchang, China, 26–28 March 2021; pp. 289–293. [[CrossRef](#)]
29. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-Time Flying Object Detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972. Available online: <http://arxiv.org/abs/2305.09972> (accessed on 16 June 2023).
30. Arif, S.; Wang, J.; Ul Hassan, T.; Fei, Z. 3D-CNN-Based Fused Feature Maps with LSTM Applied to Action Recognition. *Future Internet* **2019**, *11*, 42. [[CrossRef](#)]
31. Mateus, B.C.; Mendes, M.; Farinha, J.T.; Cardoso, A.M. Anticipating Future Behavior of an Industrial Press Using LSTM Networks. *Appl. Sci.* **2021**, *11*, 6101. [[CrossRef](#)]
32. Khan, L.; Amjad, A.; Afaq, K.M.; Chang, H.-T. Deep Sentiment Analysis Using CNN-LSTM Architecture of English and Roman Urdu Text Shared in Social Media. *Appl. Sci.* **2022**, *12*, 2694. [[CrossRef](#)]
33. Mazzia, V.; Angarano, S.; Salvetti, F.; Angelini, F.; Chiaberge, M. Action Transformer: A Self-Attention Model for Short-Time Pose-Based Human Action Recognition. *Pattern Recognit.* **2022**, *124*, 108487. [[CrossRef](#)]
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385. Available online: <http://arxiv.org/abs/1512.03385> (accessed on 19 November 2022).
35. Chen, S.; Xu, K.; Jiang, X.; Sun, T. Pyramid Spatial-Temporal Graph Transformer for Skeleton-Based Action Recognition. *Appl. Sci.* **2022**, *12*, 9229. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.