Child Action Recognition Based on Our Dataset

Andrey Yershov
Nazarbayev University
Nur-Sultan, Kazakhstan
andrey.yershov@nu.edu.kz

Aigerim Keutayeva
Nazarbayev University
Nur-Sultan, Kazakhstan
aigerim.keutayeva@nu.edu.kz

Serzhan Safin
Nazarbayev University
Nur-Sultan, Kazakhstan
serzhan.safin@nu.edu.kz

***Abstract*—The paper presents a continued work on [1], that aims for real-time action recognition specifically tailored for child-centered research. The assumption is that the action recognition model which is trained on videos with adults performing the labeled actions is less accurate with child action recognition, and training the model using dataset where children perform the actions would increase the accuracy. A state-of-the-art method was used with the dataset that was collected by our peers previously. The hypothesis was proven correct, with 33% evaluation accuracy when trained on adults' actions and 88% evaluation accuracy when trained on children's actions. Another approach where child actions were recognized is based on 2D Skeleton joints with 24 OpenPose body keypoints. The assumption is that the training on 5 actions with 14 keypoints is more accurate than on 7 actions with 24 keypoints. Finally, the hypothesis for this approach was proven partially, and the best accuracy (test = 85.62%, train = 91.65%) was taken from a training on 5 actions without 'running' and 'going forward' with 24 keypoints included.**

***Index Terms*—action recognition, transfer learning, LSTM, RNN, 2D skeleton, robotics, children**

I. Introduction

Human action recognition is an important research field due to its real-world utility in applications such as human-robot interaction, security, surveillance and entertainment. Action recognition (also known as activity recognition) consists of classifying various actions from a sequence of frames, such as "reading" or "drinking". Hypothesis of our paper is that the state-of-the-art human action recognition models being trained on adults results in significantly decreased accuracy in child action recognition. This issue was addressed in [1] by creation of dataset consisting of seven actions: boxing, waving, clapping, running, jogging, walking towards the camera, and walking from side to side. Around 200 children aged 6-11 years old performed 7 actions. Overall, around 1400 clips of average duration of 15 seconds were recorded. The dataset was provided in the form of RGB videos and skeleton body keypoints.

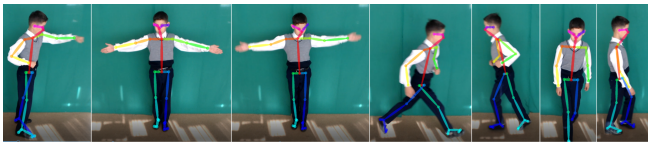


Fig. 1: Examples of actions from our dataset [1]

To prove the hypothesis, a state-of-the-art action recognition method [2] was fine-tuned using almost identical to ours dataset (KTH) consisting of 6 actions performed by adults: walking, jogging, running, boxing, hand waving and hand clapping [3]. This dataset perfectly suited the role of verification dataset for the hypothesis, as our dataset was created based on KTH and thus their content was almost identical. The obtained model was then evaluated on our dataset [1]. Another model was then trained using our dataset and evaluated using the videos from the dataset that the model did not train on.

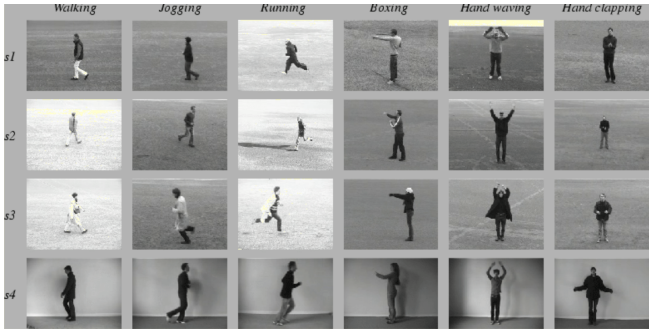


Fig. 2: Examples of actions from KTH dataset [3]

Additionally, 2 videos were recorded with a child in a setting that significantly differed from both datasets for validation purposes. A boy was asked to perform the listed actions for 5-6 seconds. He has not seen the datasets. With this validation method, probable cultural differences and verification of the model not-overfitting to the dataset were addressed.

There is another approach, where human activities can be divided into a sequence of 3D based skeletal joints [8]. Body of the human can be detected through the OpenPose (real-time system) [5] by detecting the keypoints (skeleton data). Some actions are considered important while others for the different time and specific frames have less value. For example, for the waving, clapping and boxing hands' joints are significantly important rather than foot joints. In this regard, they were removed for better estimation of the performance of the action recognition. Class of the LSTM Network was used for skeleton-based action recognition and in this context the joints from OpenPose were chosen selectively (unnecessary ones were removed in order to improve the overall performance). LSTM (Long Short-Term Memory) Network demonstrated its effectiveness in various applications and in processing sequential data [7]. In any case, for action identification, the original LSTM does not have good attention

capability. This limitation is primarily due to the limitation of LSTM in perceiving the video sequence's global background information, which is, however, often very relevant for the global classification issue, skeleton-based activity recognition.

II. Methods

A. R(2+1)D model

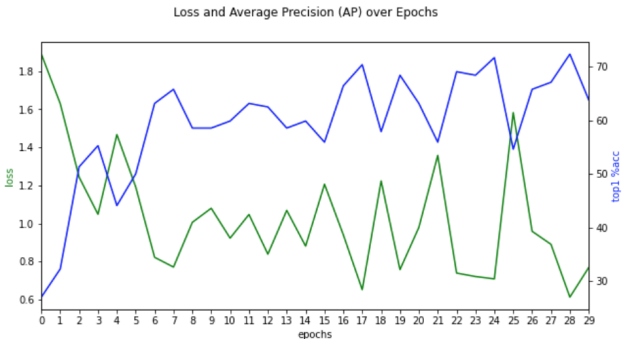
Action recognition is an active field of research, with large number of approaches being published every year. One of the approaches which stands out is the R(2+1)D model which is described in the 2019 paper "Large-scale weakly-supervised pre-training for video action recognition" [2].

R(2+1)D is highly accurate and at the same time significantly faster than other approaches. Its accuracy comes in large parts from an extra pre-training step which uses 65 million automatically annotated video clips. Its speed comes from simply using video frames as input. Many other state-of-the-art methods require optical flow fields to be pre-computed which is computationally expensive. This method was chosen because it shows the greatest accuracy over a range of default action recognition datasets [2] and its implementation was available for usage with our dataset [4].

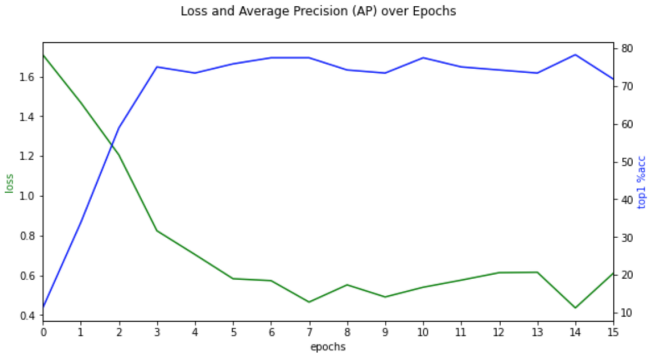
Despite all the advantages, that the method brings, there were several problems with the implementation that remained unsolved. The greatest drawback was that only 1 sample from each input could be taken per epoch. This was not a limitation of the dataset, but of the method implementation itself, as the problem arose with any dataset as an input. While this limitation does not significantly affect datasets where the input videos are short (for instance, in the KTH dataset [3] the videos were 4 seconds long in average - around 100 frames), it could be seen that the learning curve is more stable, than that of our dataset, where average video duration was 15 seconds (around 375 frames).

The noisiness of learning curve with our dataset could be explained in several ways. Even with the model input size of 32 frames (8 frames per input is suggested as a faster method, 32 frames - more accurate), there was a high chance of sampling the video at the point where no action was taking place (blank period at the beginning and at the end of the videos). Another explanation is the noisiness of the dataset itself, as children are hard to control in a setting where the dataset collection is going on. This caused situations where multiple children were present in the frame. Another learning complication was caused by the limitation of the camera hardware - there were frequent defocused frames present during the child moving across the frame. Also, the camera was too close to the child in some settings, what caused actions 'run' and 'go' to be recorded inaccurately - the child was either filmed not in full body (for 'go') or was too short time in the frame (for 'run').

The model training was performed using half of the videos from the dataset (90 out of 170 per action in average), as the other half was unavailable at the time of research due to technical difficulties with one of the .rar archives, where half of the dataset was saved. This, however, allowed to conduct



(a) Our dataset



(b) KTH dataset

Fig. 3: Learning curves of the datasets with R(2+1)D

Fig. 4: Examples of dataset noise [1]

more experiments as the training phase lasted shorter than it would with the full dataset. For the KTH dataset, the amount of videos was decreased as well to match the amount that was available for our dataset for fair comparison.

An important hyper-parameter of the model was the number of consecutive frames used as an input to the DNN. 32 frames input size was only working after the Google Colaboratory was upgraded to 25Gb RAM version, as the standard 12Gb was overflowing. Different number of epochs was used, a range between 25 and 35 seemed to be optimal with our dataset. Batch size was reduced from 8 to 5 due to memory overflow issues. Learning rate was set by "One Cycle Policy", as this improved both the accuracy of the trained model and protected it from overfitting.

B. 2D skeleton

Skeleton-based model implements multi-layer Recurrent Neural Network (RNN, LSTM) for training or sampling from image models. The RNN is the network that is popular in its important potentials in practical applications when working with sequential models [9].Compared to a classical approach, using a Recurrent Neural Network (RNN) with Long Short-Term Memory cells (LSTM) it requires less feature extraction, where the data can be fed directly into the neural network [8]. And the RNN LSTM is found to be much more efficient when uses less number of batches [8]. In this experiment, the batch size was decreased to 128, and 50000 iterations was implemented. In addition, the RNN "many-to-one" architecture was used (Fig. 5).

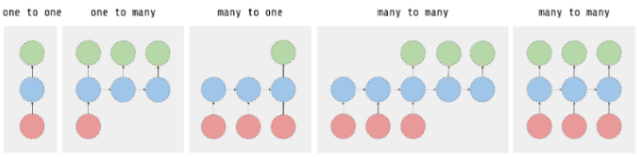


Fig. 5: Neural Network Architecture types [6]

In Fig. 5, the inputs are given red, and the outputs are green, while the blue circles represent the RNN steps. In order to implement this architecture, some steps needed to be taken.

The first steps were based on a dataset, where .csv files (frame, child_id, label, xj, yj) were provided by Dr. Sandygulova [1]. Those datasets were then filtered into input (x) and output (y) data as in Fig. 6.

The length of the data for each of 7 actions	Test dataset			Train dataset		
	original	input	output	original	input	output
box	3082	3072	96	25329	25312	791
clap	2036	2016	63	21420	21408	669
go	1040	1024	32	9633	9632	301
wave	2069	2048	64	12569	12544	392
jog	1037	1024	32	6547	6528	204
walk	1191	1184	37	7071	7040	220
run	651	640	20	4915	4896	153
Total (7 actions)	11106	11008	344	87484	87360	2730
Total (5 actions)	9415	9344	292	72936	72832	2276

Fig. 6: The length of the dataset for each action and total

This dataset was comprised of 7 subjects doing the following actions: box, clap, go forward, wave, jog, walk, run.

In total, there are 175 sample videos per action (2 were missing) made up of 98590 individual frames (87360 = training, 11106 = testing).

For the following experiment, very little preprocessing has been done to the dataset. As in the Fig. 6, the database of associated activity class number and corresponding series of joint 2D positions were filtered and divided into input (x) and output (y) .txt files, where inputs are based on 2D poses, and outputs are on labels for each 32 frame.

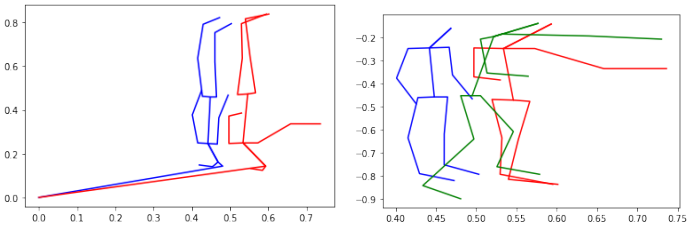


Fig. 7: Noise in the keypoints

Another way of filtering was based on an idea of noise, where the keypoints (with negligibly small values) that caused noise were taken away from the dataset (see Figure 7). In the Figure 6, the visualization shows much less noise. In other cases, the whole data for a specific subtle activity classes, such as walking versus running, agitated movement (run) versus calm movement (jog), were taken away to make the accuracy higher. Totally, this experiment considers these cases:

- 7 actions with 24 keypoints
- 7 actions with 18 keypoints
- 7 actions with 14 keypoints
- 5 actions with 24 keypoints
- 5 actions with 18 keypoints
- 5 actions with 14 keypoints

III. Results

A. R(2+1)D model

There are 10 models that were trained using the R(2+1)D method. The initial accuracies with the default settings of 8 consecutive frames used as an input to the DNN, learning rate of 0.0001 and 15 epochs were 63% and 56% with the models pre-trained on ig65m and kinetics datasets respectively. Authors of [2] also reported the lower accuracy of the model pre-trained on the kinetics dataset, thus it was excluded from further experimentations. Usage of 32 consecutive frames as input and 'One Cycle Policy' have significantly increased the evaluation accuracy up to 72%.

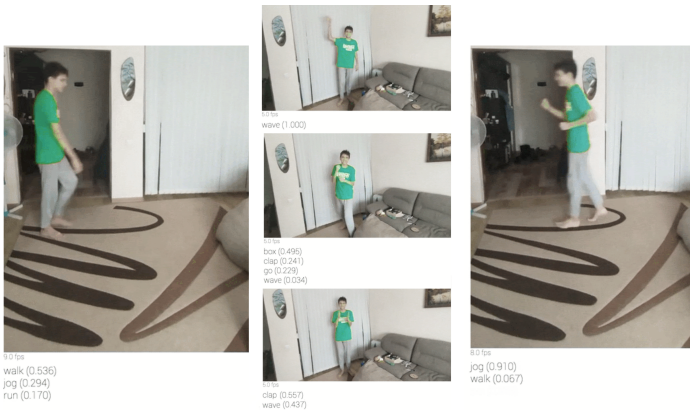


Fig. 8: Validation on the model trained with 7 actions

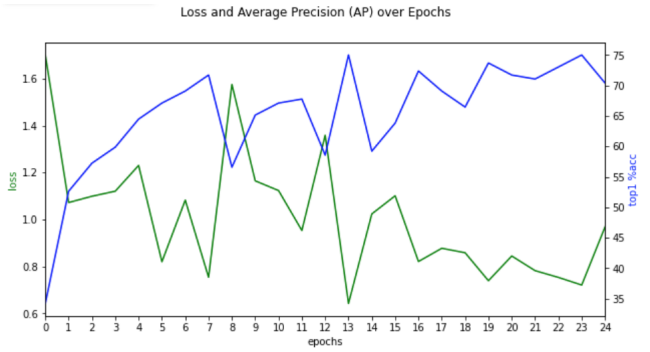
This model has showed great result with the validation video that was recorded with A. Yershov's younger brother,

predicting all of the actions correctly. The results could be seen in Fig. 8. This model was decided to be the best model achievable with R(2+1)D method on our dataset without any exclusions.

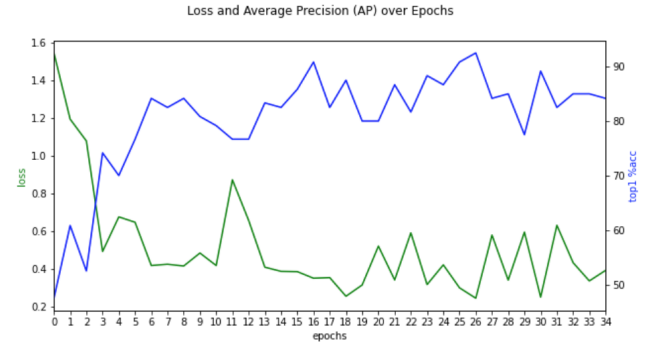
Another model with 8 consecutive frames as input was trained for 45 epochs and reached accuracy of 75% on evaluation on the dataset, however, failed completely on the validation videos with the brother. It was assumed that overfitting to the dataset has happened. These were the best results with all of the actions from the dataset included in the model. The noisiness of the learning curve was still significant, as could be seen in Fig. 9, (a). It was assumed that 'run' and 'go' action cause this noisiness due to camera hardware limitations, as it was described before. Thus, it was decided to train a model excluding 'run' action, 'go' action, and both of them.

The results showed that exclusion of the 'run' action have significantly improved the model evaluation accuracy up to 88% and made the learning curve less "jumpy" as well, as it could be seen on Fig. 9, (b). This model was decided to be the best result with the R(2+1)D method. Validation on the videos with brother were accurate as well, the model recognized the actions more confidently than the 75% model. Exclusion of both 'run' and 'go' actions have increased the accuracy for merely 2%, with higher influence on the noisiness of the learning curve (it became even more smooth, Fig 9, (c)). This result was decided to be insignificant when compared to exclusion of another one of the actions, so the model was discarded. Exclusion of the 'go' action have not increased the accuracy of the model (being around 73%). The learning curve has become slightly less noisy however, but since this does not have any real benefits, it was not included in the figure.

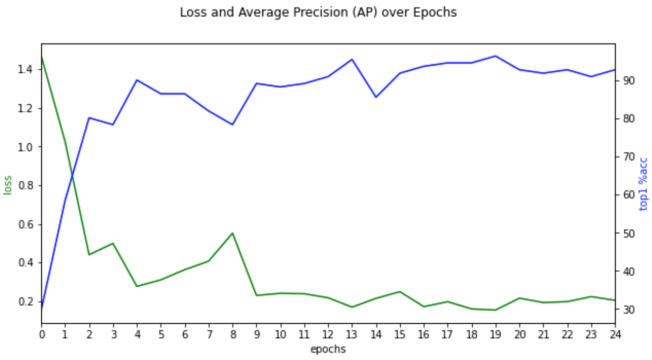
Finally, a model was trained on the KTH dataset [3] with 'run' action excluded for comparable accuracy. The learning curve of this model could be observed in Fig. 3 (b). The resultant model was then evaluated on our dataset with children. It showed an accuracy of 33% at most. This has proven the hypothesis that the model that is trained on adults shows significantly lower accuracy when evaluated on children performing the same actions.



(a) All actions, 72% evaluation accuracy



(b) Excluded 'run' action, 88% evaluation accuracy



(c) Excluded 'run' and 'go' actions, 90% evaluation accuracy

Fig. 9: Learning curves with different dataset [1] configurations with R(2+1)D

B. 2D skeleton

Overall, the results showed better performance when 5 actions rather than 7 were used for both training and testing parts as it can be clearly seen in Figure 10. The possible explanation for that might be the confusion between similar activities such as "jogging" and "running", "running" and "going forward" (see Figure 14). A little bit confusion was partially presented in activities "clap" and "wave", but compared with activities mentioned above it is less confused. However, the assumption was that the accuracy would be better for 5 actions with less keypoints. That is, by reducing the number of the keypoints, the noise would be reduced too while increasing the accuracy. It means that reducing the number of keypoints from 24 to 14 generally had a negative impact on the results. For example. In the Figure 10, it can bee seen that the accuracy for both train and test were decreased when the number of keypoints was reduced.

The result was similar for 5 actions included. However, there was no noticeable difference between 24 keypoints and 14 keypoints in 5 actions case. According to the Figure 10, it is 91.65% and 90.16% for 24 and 14 keypoints in training, respectively, while 85.62% and 85.27% for 24 and 14 keypoints in testing, respectively.

On the other hand, there are some drawbacks of the training process, which is related to the time taken for the process. In case of 5 actions the time was larger in 24 and 18 keypoints (see Figure 11). Generally, it took longer time to proceed with the 24 rather than 14 keypoints. One possible solution could

2D Skeleton OpenPose body joint	7 actions (box, clap, go, wave, jog, walk, run)		5 actions (box, clap, wave, jog, walk)	
	Train	Test	Train	Test
24 keypoints	87.55%	81.10%	91.65%	85.62%
18 keypoints	87.14%	78.20%	89.85%	81.16%
14 keypoints	82.97%	77.03%	90.16%	85.27%

Fig. 10: Train and test accuracy data in percentages.

2D Skeleton OpenPose body joints	7 actions (box, clap, go, wave, jog, walk, run)	5 actions (box, clap, wave, jog, walk)
24 keypoints	3.41 hours	3.55 hours
18 keypoints	2.47 hours	3.16 hours
14 keypoints	3.15 hours	2.65 hours

Fig. 11: Time taken for training in hours.

be changes in the batch size and number of iterations. Also, by decreasing the number of frames to 8, the process could be made faster, but 32 frames were chosen specifically to make training more accurate.

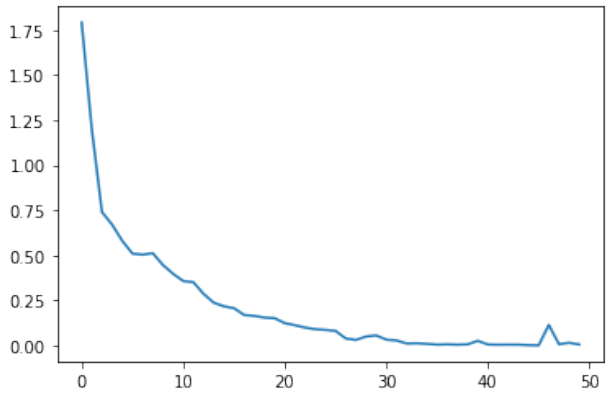


Fig. 12: Losses for 24 keypoints using 7 actions.

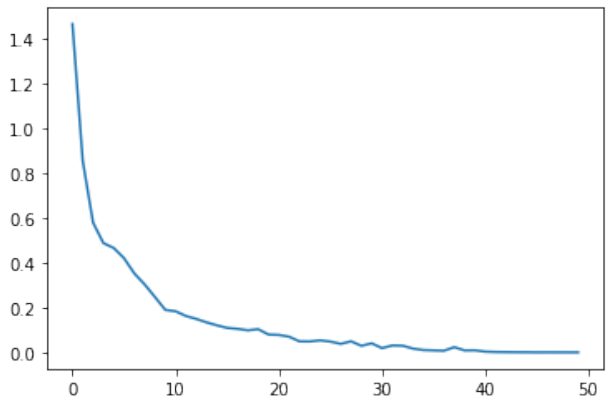
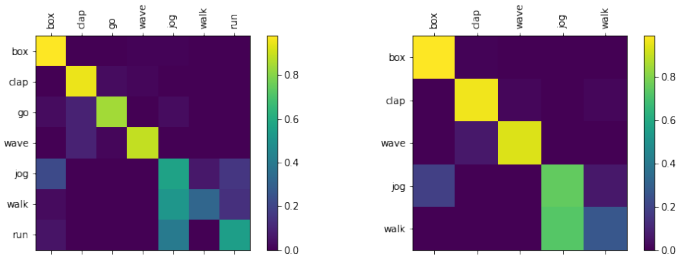
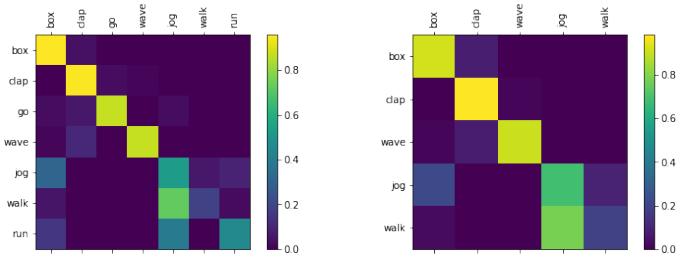


Fig. 13: Losses for 24 keypoints using 5 actions.

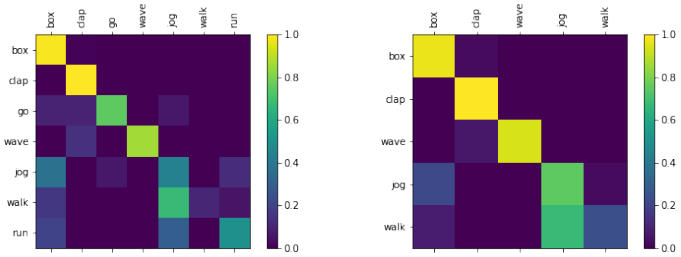
In Figures 12 and 13, it is possible to see all the losses during the training process. By comparing both graphs, the one with 5 actions had less fluctuations, while in the experiment with 7 actions, there were noticeable changes between 40000 and 50000 iterations.



(a) Confusion matrices for 24 keypoints with 7 and 5 actions



(b) Confusion matrices for 18 keypoints with 7 and 5 actions



(c) Confusion matrices for 14 keypoints with 7 and 5 actions

Fig. 14: Confusion matrices for the 2D skeleton based method

Overall, the line graphs for all losses demonstrates the decreasing trend, as the process made for training the models. Especially, in the places of fluctuations, there was a misclassification of actions, however after each of it (or fluctuation in the graph) the loss was reduced, and the model finally was trained to recognize child actions. For analysis, there are included other for process losses graphs (see Figure 14)

Looking at the Figure 14, it is possible to see the confusion matrices are depicted. The primary confusion is appeared mainly between jogging, walking and running, where 7 actions were used. It can be noticed by observing the squares on the picture of confusion matrices and they are depicted by the blurred blue-green color. As for 5 actions, the primary confusion took place between jogging and walking. There were little confusions between clap and wave (right confusion matrix).

IV. Conclusion

Overall, the method with 2D skeletal keypoints showed higher accuracy in children action recognition for the full dataset (all actions included) with highest results of 81.1% versus 75%. However, the R(2+1)D model showed the greatest results with one (88%) and two (90%) actions excluded, whereas exclusion of the actions from dataset affected the accuracy of 2D skeletal keypoints method less.

The R(2+1)D model showed high bias for the training data, as it was able to predict the child actions from validation video well. This promising result suggests that the use of large-scale weakly-supervised pre-training for video action recognition is applicable both for adults and children. Even though the pre-training phase was mostly done on adults, the fine-tuning phase allows to recognise the actions with high accuracy.

Final test accuracy of 85.62% for the skeleton-based action recognition was reported. The best model was for 5 actions with all 24 keypoints, considering the training took about 4 hours. Results show that the noise in the keypoints did not affect the training, while the specific subtle differences between jogging and running, and between going and walking made some noticeable confusion. In terms of the applicability of this method to a wider dataset, it is assumed to be able to work for any activities in which the training included a views from different angles.

Comparing the two models, R(2+1)D was much faster on the training and evaluation phase, with average epoch duration of 600 seconds and 300 milliseconds per video for the evaluation. This would make the R(2+1)D model more suitable for the real-time application. The time gap could be even larger in case if extraction of the skeleton keypoints was needed for the data. The 2D skeletal keypoints has showed less noise susceptibility, as the keypoints could still be extracted if the image is blurry or another person enter the frame.

For further research, age categories can be studied to determine the effect of age on the training and testing accuracy of a given experiment. This will also require age-specific movement data for children. That is, to divide the data of certain participants into age categories and thus identify performance differences in training and testing, and also to identify the effect of a gradual increase in age on the data and thereby study the poor detection of actions trained on children and then tested on adults and vice versa. Cultural differences is another issue to be addressed, as children with different cultural background could perform the actions differently. The dataset could also be expanded with more actions included.

References

[1] Aizada Turarova, Aida Zhanatkyzy, Zhansaule Telisheva, Arman Sabyrov, and Anara Sandygulova. 2020. Child Action Recognition in RGB and RGB-D Data. In Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (HRI '20 Companion), March 23–26, 2020, Cambridge, United Kingdom. ACM, New York, NY, USA, 2 pages. doi: 10.1145/3371382.3378391.

[2] Ghadiyaram Deepti Mahajan Dhruv. 2019. Large-Scale Weakly-Supervised Pre-Training for Video Action Recognition. CVPR 2019. 12038-12047. doi: 10.1109/CVPR.2019.01232.

[3] Christian Schuldt, Ivan Laptev, and Barbara Caputo. 2004. Recognizing human actions: a local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., Vol. 3. IEEE, 32–36. doi: 10.1109/ICPR.2004.1334462.

[4] Microsoft. 2020. Computer Vision Recipes. [Source Code]. https://github.com/microsoft/computervision-recipes

[5] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In arXiv preprint arXiv:1812.08008.

[6] Andrej Karpathy, The Unreasonable Effectiveness of Recurrent Neural Networks, 2015, http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[8] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in Proc. CVPR, 2015, pp. 1110–1118.

[9] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, Tieniu Tan, Skeleton-based action recognition with hierarchical spatial reasoning and temporal stack learning network, Pattern Recognition, Volume 107, 2020, 107511, ISSN 0031-3203. doi.org/10.1016/j.patcog.2020.107511.

V. Contribution of Authors

Andrey worked with the R(2+1)D model and the datasets in video format. Aigerim and Serzhan worked with the 2D skeleton model in skeleton keypoints format. All authors equally contributed in creation of the presentation for the project and this report.