Modification of software tool for human activity classification

Maxim Farafonov*, Evgeniye* Peresunko, and *Vladislav* Mymlikov*

Siberian Federal University, 79 Svobodny pr., Krasnoyarsk, 660041, Russian Federation

Abstract. The work is devoted to the modification and improvement of a neural network system for classifying human actions by changing the position of points of his skeleton. Within the framework of this paper, the description and results of some experiments on modification of the system developed by the authors several years ago will be presented. The experiments include new methods of augmenting the training data, removing some elements of the network, using a completely different representation of the data as a sequence of pictures rather than a set of points, using memory values and the state of the hidden LSTM cell as the output of the network instead of its direct output, and a variant of the network with the introduction of a new layer of feature encoding at the stage of sending skeletal point data to the recurrent LSTM network. The network with different layer configurations was trained multiple times, and the run data was averaged to remove the influence of randomness. In the end, the best network for the network was the one with pre-coded features.

1 Introduction

Ensuring security in crowded places is one of the most pressing problems of our time. Today, to ensure security at various enterprises, extensive video surveillance systems are used, with the help of which security personnel can observe the actions of people and promptly intervene in dangerous situations. The obvious problem in such cases is the limited human capabilities due to the complexity of simultaneous observation of dozens of cameras located in different parts of the protected area. To improve the efficiency of such surveillance, it is possible to use modern developments from the field of machine learning, and in particular, from the sphere of computer vision. Such systems provide the ability to simultaneously detect and track multiple people visible in the frame, as well as determine their actions, which exceeds the capabilities of an ordinary person. Such systems help to detect unusual and dangerous behavior almost immediately after its occurrence. This allows security personnel to react quickly to a possible threat, preventing potential casualties and damage.

One way of recognizing people's activities from video is to detect changes in the position of human skeletal points over time. This paper is devoted to the improvement of one such system developed earlier by the authors of this paper [1].

* Corresponding author: vladmymlikov@mail.ru

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

2 Related Work

Today, there are already many solutions that allow the identification of human skeleton in the video stream:

- Skeltrack. A free library for real-time tracking of human skeleton movement in front of a depth-sensing camera. Allows to select and track the movement of control points mapped to human limbs and head in a set of changing images [2].

- YOLOv11. A neural architecture for object detection that provides novel output prediction units that can produce keypoints and perform object segmentation in addition to detecting bbox objects. YOLOv11 is trained on the COCO dataset, which has a skeleton keypoint partitioning [3].

- The system developed by V. N. Mymlikov, M. M. Farafonov and P. V. Peresunko [1]. The system is able to classify 4 classes of actions, including squatting, standing, sitting, walking, and everything else is marked as unknown class. Tracking people between frames was done using an original method, spine tracking.

This paper considers work on improving the third system, since this variant has functionality for classifying people's actions and tracking, and is a development of the team of authors of this paper.

3 Experiment description

The system is based on the OpenPose pose detector [4], which receives video stream frames from cameras as input and returns information about detected human skeletons in the form of a set of points. These points are then sent to a classification block where the recurrent layer LSTM [5] plays the leading role. This block sequentially processes sequences of frames, the length of which can be customized. By default, the original network processed triple frames, also previously tested the possibility of processing 5, 7 and 12 frames, in this work the length of the processed sequence was increased to 15 frames.

Currently, the system is still under development, but some hypotheses have already been tested to improve the learnability of the network and enhance its generalization ability. A number of experimental modifications have also been made to improve performance.

The same dataset as before was used for training, namely manually collected material in the form of videos depicting the performance of 5 types of actions: standing, walking, squatting, sitting and unknown for untypical actions.

3.1 New augmentation methods

The first improvement tested was the introduction of a new training data augmentation technique. Previously, training data was augmented by zeroing in a specific subset of skeletal points for the entire dataset, simulating overlap of that body part with the environment. The current augmentation algorithm selects a random set of skeleton points at each frame sequence from every batch and zeroes them. In this way, the data becomes closer to possible real-world situations where any part of the body can be overlapped by an obstacle at any time. In addition to zeroing the skeleton points, a method of data augmentation was tried, which consisted in random small displacement of human skeleton points (noise) along the coordinate axes, thus simulating the fluctuations of points due to the inaccuracy of the pose estimator, which is invariably present during the system operation (Fig 1).

Fig. 1. Example of noisy augmentation (left original, center and right modified skeletons).

3.2 Removal of network elements

The second tested change was to remove some layers from the network, namely the Dropout layer and BatchNormalization; earlier on a sequence of three frames they showed their necessity. The variant of the network without Dropout and Batch Normalization layers was called SimpleNet, the variant without Dropout only was called BatchNormNet, the variant without Batch Normalization only was called DropoutNet, and the original network from previous work was called OriginalNet.

3.3 Adding new coding layers

The third modification was to introduce a feature encoder layer before the LSTM block, to encode a set of point coordinates into a hidden feature space for sending them to the network. Three variants were tested here, the first in the form of manually defined neurons as input (alias CustomWeightNet), the second variant with a linear feature encoding layer before the LSTM layer (FullyEncodedNet), the last variant used a convolutional layer for feature extraction by processing points directly before the LSTM layer (PointConvNet).

3.4 Mapping skeletons into images

The fourth experiment consisted in significant modification of work of the whole system. In this variant after recognition of points of skeletons of people the skeleton points were drawn in the form of a square picture with a drawing of a skeleton. In this variant for more informativeness the skeleton was drawn not as thin lines, but as slightly blurred with Gaussian kernel and smoothed line colors (Fig. 2). This network has a pseudonym ImgConvNet.

Fig. 2. Example of an image sent to convolution layers.

Next, the picture was passed through several convolutional neural layers and further extracted features went to the LSTM.

3.5 Hidden state and cell state

Another experiment was to use as output the hidden state of the LSTM layer from the final step, rather than hidden states from all time steps (the alias of this network is HiddenNet). A variant was also tried in which the value of the cell state of the LSTM layer in the final step (alias CellNet) was used as the output.

3.6 Other experiments

As additional experiments, a network without LSTM layers at all was trained, which used only two linear layers (LinearNet). Also, a variant of the network was tested, which receives as input not information about points directly, but sets of coordinate differences between frames, this variant was called MotionNet.

4 Results

All experiments were performed using the torch module. Each network modification was trained for 1500 epochs, without applying early stopping to ensure a level playing field for candidates. The size of the batch was 256 samples. Adam was used as the optimizer. The learning rate was kept constant throughout all epochs at 0.003, and the other optimizer parameters were left as default. A random seed was fixed before starting the training cycle of each network to minimize fluctuations and ensure reliable reproducibility of the results. Training was performed at 10 different random seeds, after which the results were averaged for each network modification considered. Figures 3 and 4 further show the plots of the variation in validation error and % recognition accuracy, respectively. In this context Original means using an original architecture not learned weights.

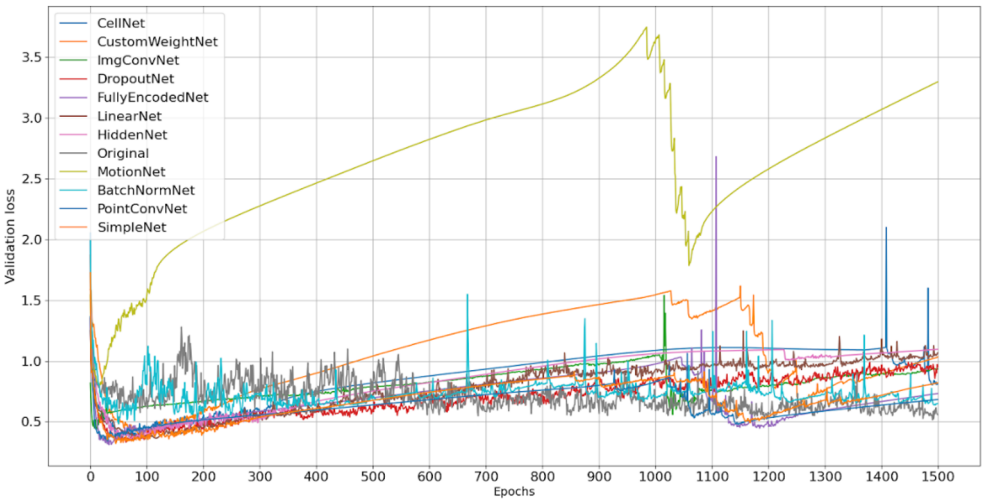

Fig. 3. Plot of Cross-Entropy loss variation on validation for different models.

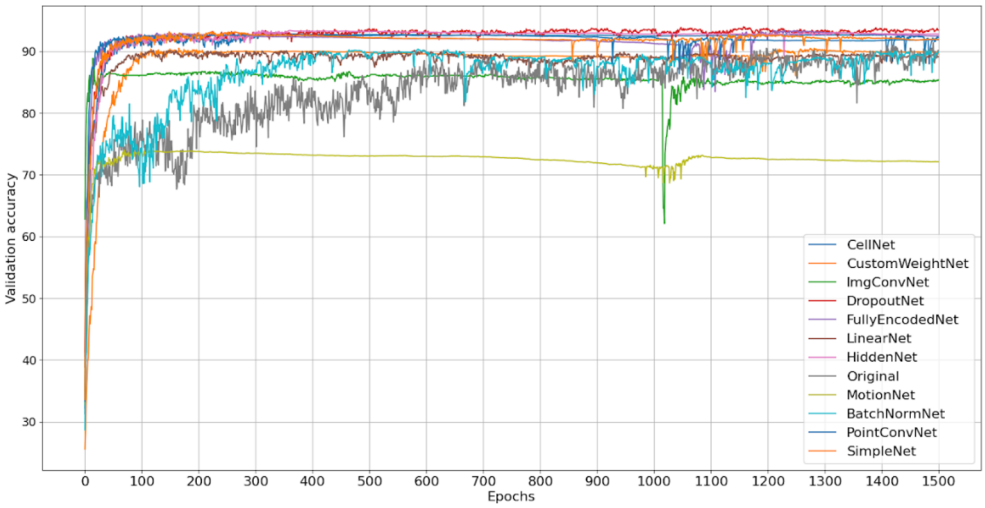

Fig. 4. Plot of % accuracy variation on validation for different models.

The best model was determined using the error on the validation subset of the dataset. As a result of the comparison, the best result was shown by the network with a feature encoding layer in front of the LSTM layer. A slightly more modest result was demonstrated by a network completely similar to the original one, but without the dropout layer and Batch-Normalization layer. The histogram of the error comparison on the validation dataset is shown in Figure 5.

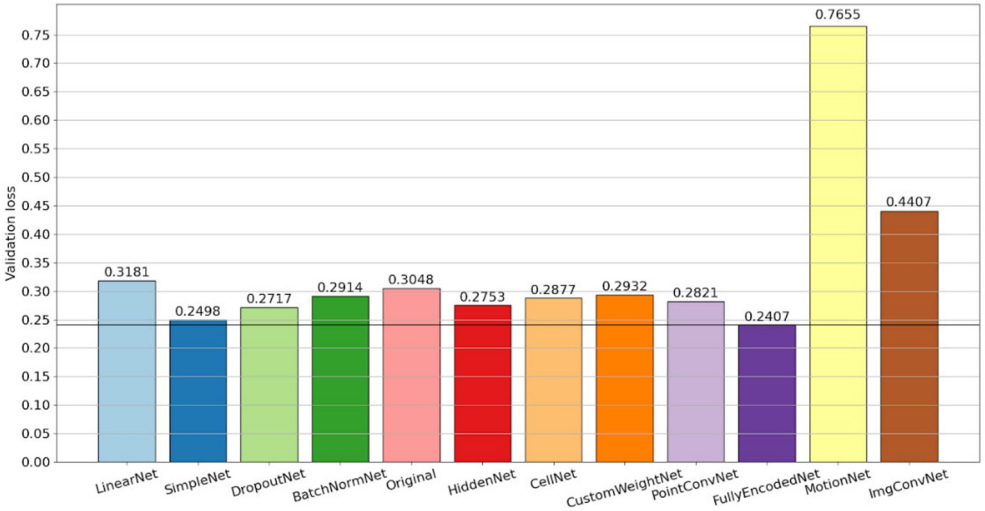

Fig. 5. Histogram of Cross-Entropy loss for validation data from different models.

The new augmentation methods were applied after the best network architecture was determined to discover what impact they would have, whether they would achieve greater robustness to overtraining and better generalizability. The results of FullyEncodedNet training with and without augmentations are shown in Figure 6. The augmentation options are represented as abbreviations: NS[maximum noise value]_[probability of application]_AM[upper limit of removed points]_[probability of application].

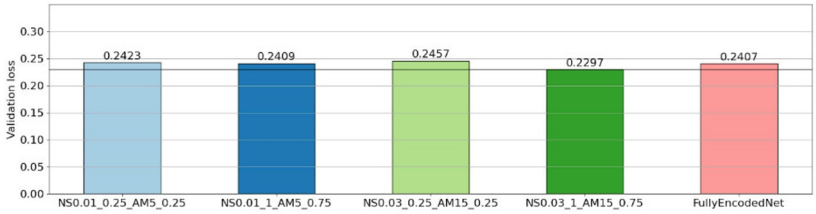

Fig. 6. Cross-Entropy loss on validation data for network trained with and without augmented data.

As a result of comparing the final best model on the test dataset and the model developed years earlier, a significant improvement in the magnitude of the error can be seen. The comparison was performed on the test dataset, and the final comparison is shown in Figure 7.

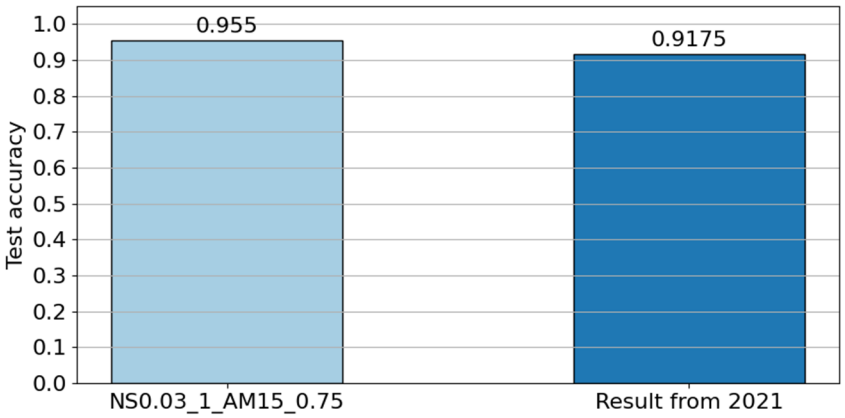

Fig. 7. Comparison of accuracy on the test set of the current and previous (from 2021) models.

5 Conclusion

According to the results of the work it was possible to achieve a significant improvement in the accuracy of recognizing people's actions, namely by 3.75%. This result is largely due to an increase in the number of frames processed, practice has shown that three frames are not enough for effective and reliable determination of the type of activity carried out by a person. In the future it is planned to investigate variants of the network with the number of frames 30, 60 and 90. Also the results showed that the most effective variant is FullyEncodedNet with augmentation. The best augmentation technique is noise warp of coordinates with power 0.03 and probability 1, also the best practice includes zeroing from 1 to 15 points with probability 0.75. All other variants proved to be less efficient to varying degrees, with the dropout-only variant being almost identical to the best. Further work on improving the results is also envisioned in the future, in particular, the collection of a new larger training data set.

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Grant № 075-15-2022-1121).

References

1. V. N. Mymlikov, M.M. Farafonov, P.V. Peresunko *A software tool for identifying a human skeleton in a video stream*, in Proceedings of the Advances in science and technology-DNiT-2021, 10 December 2021, Russian Federation (2021).
2. L. A. Zavala-Mondragon, B. Lamichhane, L. Zhang, G. D. Haan, *CNN-SkelPose: a CNN-based skeleton estimation algorithm for clinical applications.* Journal of Ambient Intelligence and Humanized Computing, **11(6),** 2369-2380 (2020). https://doi.org/10.1007/s12652-019-01259-5
3. R. Khanam, M. Hussain, Yolov11: *An overview of the key architectural enhancements* in arXiv preprint arXiv:2410.17725, UK (2024)
4. G. H. Martınez, *Openpose: Whole-body pose estimation* (Robotics Institute Carnegie Mellon University Pittsburgh, 2019)
5. R. C. Staudemeyer, E. R. Morris, *Understanding LSTM--a tutorial into long short-term memory recurrent neural networks* in arXiv preprint arXiv:1909.09586. (2019)