Comparative analysis of 3D convolutional and LSTM neural networks in the action recognition task by video data

R.J. Portsev¹ and A.V. Makarenko¹
¹Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia

poruss@mail.ru, avm.science@mail.ru

Abstract. In the present paper a comparative analysis of two architectural neural network approaches (based on 3D convolutional and LSTM) in the recognition of actions on video is made. The problem was being solved on 10 behavior classes separated from the UCF50 dataset. The original neural network architectures were developed and pre-trained. It was found that the network based on 3D convolutions has better generalization ability and is more stable in the training.

1. Introduction
Extensive literature is devoted to the action recognition of moving objects in the video, so the recognition of actions is used in management tasks related to medicine [1, 2], human-machine interaction [3, 4], observation [5, 6], etc. Deep neural networks are now actively used to solve various problems in this area [7]. At the same time, the issue of recognition of actions (behavior) of moving objects in the video are very complicated and currently do not have an unambiguous effective solution.

The proposed report is devoted to the comparison of two architectural neural network approaches (based on 3D convolutional and LSTM cells) in the recognition of actions on video.

2. The input data and dataset preparation
To solve the problem we chose the UCF50 data set [8] for action recognition, which consists of realistic YouTube videos. The UCF50 dataset includes 50 classes, and in each class the video clips are grouped into about 25 groups, where each group consists of at least 4 action clips ("behavioral lines"). Video clips from the same group may have common features, such as the same person, similar background, etc. To make an experiment from the initial dataset, we have allocated 10 classes: VolleyballSpiking, Skiing, BaseballPitch, Punch, Drumming, Basketball, PlayingViolin, PlayingTabla, PullUps, RockClimbingIndoor. The examples of frames from the selected classes are shown in Fig. 1.
Taking into account the mentioned features, the division of video clips into training and test parts was carried out the following way: all clips from g03, g07 and g12 groups were placed in the test set, all other clips were placed in the training set. This separation allows avoiding explicit "infiltration" of training data into the test set.

Then, in each class, the video clips were randomly selected from 10 selected ones. Then, samples were formed from each video in 2-frame increments. A sample is a series of 8 color video frames at a resolution of 320x240 pixels (RGB palette, 3x8 color depth), taken in 8 frame increments (duration of the selected fragment is approximately 2.5 seconds). 1500 samples of each class were obtained for the training set, and 150 samples of each class for the test set. The final data set includes 15000 samples for the training and 1500 samples for the test.

3. Architectures and training of neural networks

3.1. The description of the neural networks

Two neural networks were synthesized for the experiment: Net3D – based on 3D convolutional [9]; NetLSTM – using 2D convolutional (with identical weights for all frames) and LSTM layer [9]. Fig. 2 schematically shows the differences between Net3D and NetLSTM network architectures without specifying all layers in detail. The number of layers and trainable parameters of the Net3D neural network was 19 and 4,074,618 respectively, while NetLSTM was 21 and 4,860,170.

During training of neural networks the loss function categorical crossentropy was used [9], and for the quality estimation F1-score was used. Both networks were trained in 20 epochs by SGD optimizer with batch size of 240 but with different initial values of learning rate parameter (lr): lr = 0.02 and...
Ir = 0.06 for Net3D and NetLSTM, respectively. Such methods of regularization [9] as batch normalization and dropout were used to combat overfitting.

Training of neural networks was carried out on NVIDIA Titan RTX graphics accelerator with the help of Pytorch deep learning library (build 1.5.0).

3.2. Results
The process of training of a neural network with 3D convolution is proved to be more stable in comparison with training a neural network with LSTM layer. Thus, to obtain $F_1 > 0.8$ quality on the test set for Net3D, less than 10 variants of random initialization of weights were required, while NetLSTM network could not exceed $F_1 > 0.7$ on the test set with 80 variants of random initialization of weights.

Fig. 3 shows the graphs of neural network training, where vertical dashed lines indicate the best epochs (blue color corresponds to Net3D, red – NetLSTM). The maximum value of $F_1 = 0.8377$ on the test set for Net3D is reached on epoch 9 (loss = 0.0666, val loss = 0.6054), and for NetLSTM $F_1 = 0.69$ – on epoch 20 (loss = 0.0553, val loss = 1.4276).

![Graphs showing neural network training metrics depending on the epoch number](a) – loss function, (b) – $F_1$ score.

As you can see from Fig. 3, the Net3D neural network behaves much more steadily during training than the NetLSTM.

To compare the quality of trained neural networks, Fig. 4 shows the corresponding confusion matrices calculated from the test set.

| No | Class             | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|----|-------------------|----|----|----|----|----|----|----|----|----|----|
| 1  | VolleyballSpiking | 148| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 2  |
| 2  | Skiing            | 0  | 140| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 3  | BaseballPitch     | 0  | 0  | 130| 0  | 0  | 12 | 0  | 0  | 0  | 0  |
| 4  | Punch             | 0  | 0  | 0  | 150| 0  | 0  | 0  | 0  | 0  | 0  |
| 5  | Drumming          | 0  | 0  | 0  | 0  | 144| 0  | 0  | 0  | 0  | 0  |
| 6  | Basketball        | 41 | 0  | 0  | 0  | 0  | 100| 0  | 0  | 0  | 9  |
| 7  | PlayingViolin     | 0  | 0  | 0  | 0  | 32 | 0  | 110| 7  | 1  | 0  |
| 8  | PlayingTabla      | 0  | 0  | 1  | 0  | 0  | 0  | 149| 0  | 0  | 0  |
| 9  | PullUps           | 0  | 0  | 0  | 0  | 0  | 0  | 32 | 18 | 68 | 0  |
| 10 | RockClimbingIndoor| 14 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 110|

![Confusion matrices for the trained Net3D (a) and NetLSTM (b) networks.](a) (b)

In the confusion matrix for the Net3D network you should pay attention to three classes (1, 4 and 8) with a high value of Recall > 0.98, but the leaders of $F_1$ are 2, 3 and 8 classes with a value of $F_1 > 0.95$. Classes 1 and 4, compared to class 8, contain a significant number of false negatives (in
other words, a low Precision value). The clear outsiders for $F_1$ are 6, 9 and 10 classes with a value of $F_1 < 0.8$, with most of the errors of class 9 being concentrated on class 10 and errors of class 10 being found in the other 6 classes.

In the confusion matrix for the NetLSTM network, there are 1, 3, 4 classes with a high Recall value, but the leader for $F_1$ is class 4 with a value of $F_1 = 0.9865$. The clear outsider of $F_1$ is class 8 with a value of $F_1 = 0.4211$.

As you can see from the data in Fig. 4, there are significant differences between the error patterns of both networks, although both networks have equally poorly recognized classes: 6-Basketball, 7-PlayingViolin, 9-PullUps and 10-RockClimbingIndoor. We will do more research to find out the reasons for this artefact.

4. Experiments
For the in-depth comparison of the trained networks, 2 experiments with modified datasets were conducted: an experiment with randomly shuffling frames in video and an experiment with removing dynamic in video.

4.1. The experiment with randomly shuffling frames in video
The essence of the experiment was that video samples were fed to the trained network, where the sequence of frames was randomly shuffled. The experiment was conducted in order to study neural network stability to the order of the frames in the video.

Let's analyze histograms (see Fig. 5), where the values of loss function and $F_1$-score of trained networks are compared while shuffling frames in samples. It must be noted that 30 variants with random frame stirring in sampling both in test and training sets were generated for histograms construction.

It can be clearly seen from Fig. 5, that after feeding the sequences of shuffled frames, the loss value increased significantly in both the test (Fig. 5a) and training (Fig. 5b) sets in the Net3D neural network, which indicates a decrease in the stability of decision-making by the neural network, while in the NetLSTM the loss value increase is not so significant (see Fig. 5e,f).

But the $F_1$ situation is somewhat different: in both neural networks the $F_1$ value has decreased, but in quantitative terms it is quite in significant (this is indicated by the median values against values without shuffling). At the same time, in the training set (Fig. 5c,g) the difference is more significant (probably due to the effect of memorization of training data) than in the test set (Fig. 5d,h). Thus, it follows from the results of the analysis that both networks practically do not react to changing the sequence of the video frames.

Then let's analyze which classes were most affected by randomly shuffling frames. To facilitate the analysis, let's calculate for each class, $\Delta F_1^i = F_1^i - \bar{F}_1^i$, where $\bar{F}_1^i$ are median $F_1$ values for the $i$-th class, $F_1^i$ are $F_1$ values for the $i$-th class without shuffling, and let's build diagrams of the "sagging" of the classes (Fig.6), where on the abscissa axis the values for the train set lie, and on the ordinate axis – the values for the test set, and the point is the number of the class (the number from 1 to 10).
Fig. 5. Histograms of the loss and $F_1$-score of the pre-trained Net3D and NetLSTM networks in case of randomly shuffling frames in video (blue color corresponds to Net3D, red – to NetLSTM; bright colors correspond to the values on the training set, pale – to the values on the test set; vertical dotted lines correspond to the median values, vertical black lines – to the values with the correct sequence of frames).

In Fig. 6 you can clearly see that for the Net3D network on the test set 2 classes are the most "sagged" (6-Basketball and 4-Punch), and there are classes for which the shuffling had a positive effect, i.e. the value of $F_1$ improved (8-PlayingTabla, 9-PullUps, 10-RockClimbingIndoor, 5-Drumming).

For the NetLSTM network on the test set class 4-Punch is the most "sagged" and less than 0.1 – classes: 1-VolleyballSpiking, 7-PlayingViolin, 6-Basketball and 10-RockClimbingIndoor, and there is also a class for which the shuffling had a positive effect (9-PullUps).

We can also note that the class 3-BaseballPitch is invariant for the random frame shuffling on both neural networks.

Fig. 6. Scatter plots of "sagging" classes for Net3D (a) and NetLSTM (b) networks when randomly shuffling frames in video.
Next, let us analyze the error structure of neural networks at a random frame shuffling. Fig. 7 shows confusion matrices by median values for trained Net3D and NetLSTM networks when shuffling frames in a video sequence.

In the course of the experiment at 30 variants of random frame shuffling, 30 confusion matrices were obtained. For the analysis we will take the median values, \( \bar{x} \), and compare them with the values from the confusion matrix without shuffling, \( x \). On the diagonal, the cells are painted red if \( \bar{x} < x \), and green if \( \bar{x} > x \), for all the other cells the rule is reversed. In other words, a red hue indicates that the median value is worse than the original confusion matrix and a green hue indicates an improvement. The intensity of the color is the same as \( |\bar{x} - x| \).

As was shown earlier, for the Net3D network, two classes (6-Basketball and 4-Punch) are the most "sagged". The main reason for the "sagging" of class 6-Basketball is a significant decrease of the Recall, compared to the original confusion matrix, 9 correct classifications moved to class 10-RockClimbingIndoor. While class 4-Punch "sagged" due to the fact that there were 9 additional errors in class 10-RockClimbingIndoor. At the same time, the structure of errors in class 10-RockClimbingIndoor itself has changed significantly, which indicates the instability of the network on samples of this class. We can also see a clear improvement in class 9-PullUps due to random frame shuffling, again due to class 10-RockClimbingIndoor.

For the NetLSTM network, frame shuffling significantly affected the 4-Punch class, which "sagged" due to a decrease in Recall, 3 correct classifications moved to class 7-PlayingViolin. In Fig. 7 you can also see that frame shuffling had a greater impact on Net3D than on NetLSTM.

4.2. An experiment with removing dynamic in video

An experiment was carried out to remove dynamic in video, which consisted of the following. In each sample, a series of 8 frames was replaced by a randomly selected frame from the same sample, after which the data was sent to the neural network. As in the first experiment, 30 variants with random selection of frames were generated.

The results of the experiment are shown in Fig. 8, where the values of the loss function and \( F_1 \)-score are compared. On the histograms, vertical black lines correspond to values at a correct sequence of frames, vertical grey dashed lines correspond to the median values at frame shuffling, and vertical dashed color lines correspond to the median values of the new experiment.
From Fig. 8 it can be clearly seen that in the experiment with removing dynamic in video the networks are significantly more degraded in all indicators, relatively the situation with random frame shuffling in video, and for the Net3D neural network quality decline is much stronger than for NetLSTM. It follows that frame shifting in the video is one of the key moments for a decision to be made by trained neural networks.

Fig. 8. Histograms of the loss and $F_1$-score of pre-trained Net3D and NetLSTM networks when removing dynamic in video (color shades are similar to Fig. 5; vertical dotted lines correspond to the median values, vertical black lines correspond to values at correct frame sequence, vertical grey dotted lines correspond to the median values at frame shuffling).

Fig. 9 clearly shows which classes were most affected by the removal of dynamics. For example, the 4-Punch class was the most sensitive to dynamic in video for the Net3D neural network, while in the NetLSTM we can distinguish classes 1-VolleyballSpiking and 7-PlayingViolin.

Fig. 9. Scatter plots of "sagging" classes for Net3D (a) and NetLSTM (b) networks when removing dynamic in video.
It can be seen, however, that removing dynamics in video did not improve the $F_1$ value for any class, as was the case in the first experiment.

Let's analyze the confusion matrices by median values for Net3D and NetLSTM for the experiment with removing dynamic in video shown in Fig. 10.

![Confusion matrices](image)

(a) 1-VolleyballSpiking 2-Skiing 3-BaseballPitch 4-Punch 5-Drumming 6-Basketball 7-PlayingViolin 8-PlayingTable 9-Pullups 10-RockClimbingOutdoor

(b) 1-VolleyballSpiking 2-Skiing 3-BaseballPitch 4-Punch 5-Drumming 6-Basketball 7-PlayingViolin 8-PlayingTable 9-Pullups 10-RockClimbingOutdoor

The main reason for the "sagging" of class 4-Punch in the Net3D neural network is a significant decrease in the Recall value, compared to the original confusion matrix, with only 21 of 150 correct classifications left.

For the NetLSTM network, the removal of dynamics significantly affected classes 1-VolleyballSpiking and 7-PlayingViolin, which "sagged" also due to the decrease in Recall.

In Fig. 10 you can also see that removing dynamic in video had a greater impact on Net3D than on NetLSTM.

5. Conclusion

As a result of an experiment of the action recognition on video using two deep neural networks with different Net3D and NetLSTM architectures, it was found that the Net3D neural network provides better quality than NetLSTM, with $F_1$ values on the test set for Net3D and NetLSTM being 0.8377 and 0.6900 respectively. However, the number of trainable parameters for Net3D is approximately 16% lower than for NetLSTM (4,074,618 vs. 4,860,170). In addition, the learning process of a 3D convolution neural network is more stable (and more robust against changes of the optimizer's hyperparameters) than that of an LSTM neural network.

From an experiment with randomly shuffling frames in samples, it was found that the trained neural networks hardly react to the very sequence of frames (precisely, to change it). It follows from that, as part of the experiment, both neural networks are weakly sensitive to changes in the temporal context (i.e. the order in which the frames in the video sequence follow). There are two reasons for this: (i) - the low dynamic entropy of the video clips themselves (a feature of the UCF50 dataset); (ii) - a deep neural network is capable of restoring the correct temporal order of video frames in its latent space (the order is formed at the training step).

From the experiment with removing dynamic in video it follows that for trained neural networks, changing frames in video is essential for correct classification of scenes. At the same time, the lack of dynamics in video deteriorated the quality of Net3D from $F_1 = 0.8377$ to $F_1 = 0.5683$ on the test set, while the quality of NetLSTM decreased from $F_1 = 0.69$ to $F_1 = 0.6515$.

Thus, the Net3D results support the first hypothesis about the ability of a neural network to restore in its latent space the correct temporal order of frames on video (learned during the network training...
step). And the results for NetLSTM indicate a weak variability in the prepared dataset, because NetLSTM does not lose much in quality if the sequence of frames in the video is replaced by the same frame.

We are currently planning additional experiments to clarify these hypotheses, including the use of other datasets with other configurations and characteristics.

References
[1] Avci, A.; Bosch, S.; Marin-Perianu, M.; Marin-Perianu, R.; Havinga, P. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In Proceedings of the 23th International Conference on Architecture of Computing Systems 2010, Hannover, Germany, 22–23 February 2010; pp. 1–10.
[2] Mulroy, S.; Gronley, J.; Weiss, W.; Newsam, C.; Perry, J. Use of cluster analysis for gait pattern classification of patients in the early and late recovery phases following stroke. Gait Posture 2003, 18, 114–125.
[3] Rautaray, S.S.; Agrawal, A. Vision based hand gesture recognition for human computer interaction: A survey. Artif. Intell. Rev. 2015, 43, 1–54.
[4] Mitra, S.; Acharya, T. Gesture recognition: A survey. IEEE Trans. Syst. Man Cybern. Part Appl. Rev. 2007, 37, 311–324.
[5] Vishwakarma, S.; Agrawal, A. A survey on activity recognition and behavior understanding in video surveillance. Vis. Comput. 2013, 29, 983–1009.
[6] Lin, W.; Sun, M.T.; Poovandran, R.; Zhang, Z. Human activity recognition for video surveillance. In Proceedings of the 2008 IEEE International Symposium on Circuits and Systems, Seattle, WA, USA, 18–21 May 2008; pp. 2737–2740.
[7] I. Rodríguez-Moreno, J. M. Martinez-Otza, B. Sierra, I. Rodriguez and E. Jauregi, Video activity recognition: State-of-the-art, Sensors, vol. 19, p. 3160, Jan. 2019
[8] Reddy, K. K., & Shah, M. (2013). Recognizing 50 human action categories of web videos. Machine vision and applications, 24(5), 971-981.
[9] Goodfellow I., Bengio Y., Courville A., Deep Learning. Cambridge: MIT press, 2016.