ISSN 1870-4069

# Recognition of Stereotypical Motor Movements in Children with Autism Spectrum Disorder Using a ConvLSTM Network

Magdiel García-Juárez[1], José Anibal Arias-Aguilar[1],
Alberto Elías Petrilli-Barceló[2]

[1] Universidad Tecnológica de la Mixteca,
División de Estudios de Postgrado, Oaxaca,
Mexico

[2] Tokyo University of Science,
Faculty of Science and Technology,
Japan

ps2017280001@ndikandi.utm.mx, anibal@mixteco.utm.mx,
petrilli@rs.tus.ac.jp

**Abstract.** Stereotypical motor movements (SMM) are behaviors typically found in children with autism spectrum disorder (ASM). They consist of repetitive movements such as hand flapping, body rocking, and finger flicking. Automatically monitoring and recognizing these movements in a way that is reliable and efficient over time could provide important information that would improve our understanding and contribute to an intervention strategy around a central ASM symptom. This paper proposes a model based on deep-learning techniques for automatically classifying video-recorded SMMs, which are present in children with ASM. For obtaining spatial information, the YOLOv2 object-detection architecture is used, in which each video frame is processed and converted into a new abstract representation of 13 x 13 x 1024 descriptors, which are then input into a Recurrent Convolutional Network to find temporal information and classify the actions. The proposed architecture is trained by the SSDB skill-assessment dataset, composed of low-quality videos in uncontrolled environments in which children with ASM are performing SMMs. To compensate for the disadvantage of having a low number of examples, we propose doing a pretraining with the HMDB51 dataset, which has actions with a movement dynamic similar to the proposed actions.

**Keywords:** Autism, deep learning, HAR.

## 1 Introduction

Autism spectrum disorder (ASM) is a neurodevelopmental disorder characterized by diagnostic criteria that include significant problems in communication and social interaction; limited patterns of behavior, interests, or activities; and repetitive movements called stereotypical motor movements (SMM) [1, 2]. The most frequent

SMMs include repetitive movements such as body rocking and complex hand-and-finger movements [3].

The skill-assessment process of children with ASM generally involves asking them to perform certain actions by giving them a set of instructions and monitoring their responses as they happen. Interviews with specialists, behavioral observations, and parental reports are needed.

This process involves carrying out work-intensive tasks such as recording the observations of their action responses to a set of stimuli over long periods of time [4]. The automatic, reliable, and efficient detection and monitoring of SMMs over time could provide benefits not only for diagnosing autistic children, but also for a better understanding of and accounting for diverse elements in developing an intervention strategy for a central ASM symptom [5, 6].

To confront this challenge, deep learning (DL) techniques for human action recognition (HAR) by processing video frames allow for the automatization of many SSM-monitoring tasks in ASM therapy, making the diagnosis of autism easier. In this paper, we propose a model based on deep learning techniques for the automatic detection of SMMs in children with ASM (see Fig. 1).

We evaluate their performance using the open-access dataset SSDB [7], which is composed of videos compiled from public-access websites posted by parents or caregivers that were filmed in unregulated settings, and thus may have obstructions, poor lighting, and a smaller number of each example type. To compensate for the fact that the lack of data may affect the model is learning, we propose pretraining the network with the HMDB51 dataset [8], from which we had to choose action categories that had similar movements to the SSDB categories.

One contribution of this paper is its evaluation of the performance of the YOLOv2 architecture, which was originally proposed to detect objects, in its adaptation to the task of the recognition of actions performed over time, using the cascade setting with a Convolutional Long Short-Term Memory (ConvLSTM) Network.

Using YOLOv2, each video frame is converted into a new, more complex representation of descriptors with dimensions of 13 x 13 x 1024, which provides spatial information. These are then input into the ConvLSTM, which will allow temporal information to be extracted for the classification of actions with a Fully Connected (FC) layer. We propose using a ConvLSTM to avoid the problem of flattening input data and losing spatial information that could be important for classifying actions.

## 2 Related Papers

Video human action recognition is one of the main challenges for the field of computer vision. HAR has had an important impact on applications for children with ASM, as indicated in the paper by Zunino et al., in which hand movements are analyzed while performing the actions of picking up, placing, pouring out, and passing a bottle [9]. The tests showed that children can be classified into groups based on whether they had ASM or not. Using the VGG-16 network in cascade with an LSTM, an accuracy rate of 82% is achieved.

Using the same dataset, Tian et al. [10] and Sun et al. [11] report an improvement in accuracy, with 87.17% and 95.2%, respectively. To detect typical and stereotypical

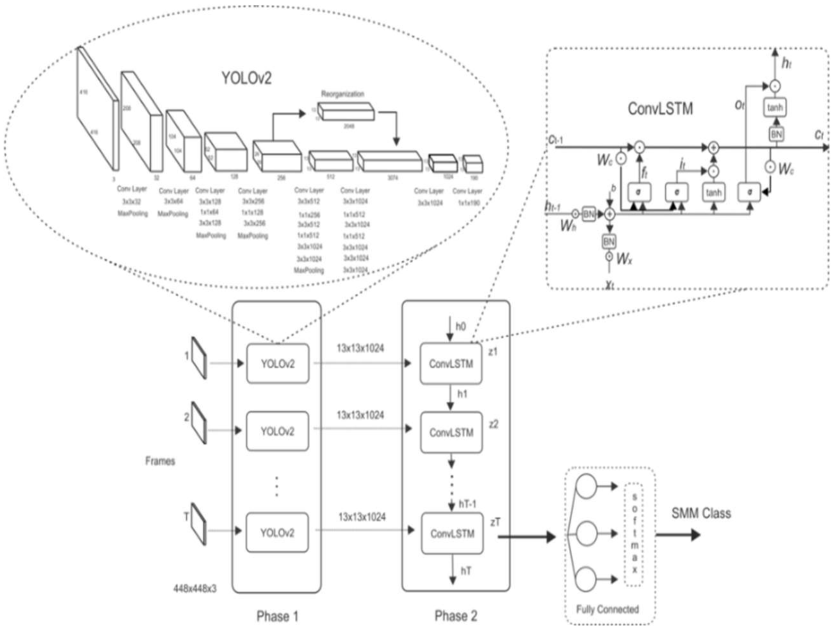

YOLOv2
Reorganization
Conv Layer 3x3x32 MaxPooling
Conv Layer 3x3x64 MaxPooling
Conv Layer 3x3x128 1x1x64 3x3x128 MaxPooling
Conv Layer 3x3x256 1x1x128 3x3x256 MaxPooling
Conv Layer 3x3x512 1x1x256 3x3x512 1x1x512 3x3x1024 3x3x1024 MaxPooling
Conv Layer 3x3x1024 1x1x512 3x3x1024 1x1x512 3x3x1024 3x3x1024 3x3x1024
Conv Layer 3x3x1024
Conv Layer 1x1x190
ConvLSTM
$h_t$
$o_t$
tanh
BN
$c_{t-1}$
$c_t$
$W_c$
$f_t$
$i_t$
$b$
$h_{t-1}$
$W_h$
$W_x$
$x_t$
σ
1
2
T
Frames
448x448x3
YOLOv2
13x13x1024
h0
h1
hT-1
hT
z1
z2
zT
ConvLSTM
Phase 1
Phase 2
softmax
Fully Connected
SMM Class

**Fig. 1.** Model for recognizing SMMs in children with ASM.

**Table 1**. Pairing of actions from the SSDB and HMDB51 datasets.

| **SSDB** | | | **HMDB51** | | |
|---|---|---|---|---|---|
| Action | Train | Eval | Action | Train | Eval |
| Arm Flapping | 45 | 11 | Clap | 76 | 20 |
| Head Banging | 23 | 6 | Brush Hair | 72 | 20 |
| Spinning | 28 | 5 | Turn | 191 | 48 |

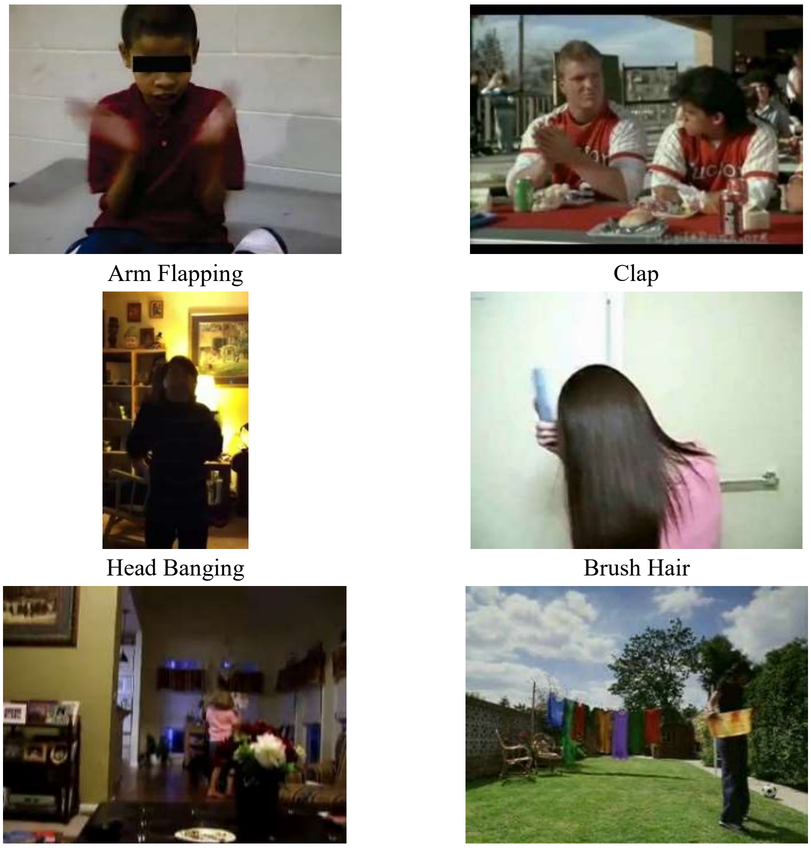
actions in children with ASM, Silva et al. [12] use the Intel RealSense camera and the SDK Nuitrack to detect and extract the coordinates of the body’s joints.

A CNN classifies the different behavioral patterns and obtains an average accuracy of 92.6 percent in the test data. In their paper, Pandey et al. address the problem of action recognition on SSDB dataset classes and a real-world autism dataset.

For the training, the guided weak supervision (GWS) technique was proposed, in which dataset classes are matched through output vectors using the posterior likelihood maximization principle [13].

The YOLOv2 network has shown a high level of performance in the detection of objects, which is why it is a reliable architecture for the task of extracting spatial information. In connecting it in cascade to ConvLSTM, as is shown in Fig. 1, a temporal information-processing model is added.

This collaborative architecture for spatiotemporal processing should provide an efficient model for the task of action recognition for children with ASM. Because few



Arm Flapping

Clap

Head Banging

Brush Hair

Spinning

Turn

**Fig. 2**. Examples of actions in the SSDB (left) and HMDB51 (right) datasets.

**Table 2**. Hyperparameters for the ConvLSTM.

| **Hyperparameter** | **Value** |
|---|---|
| Dropout | 25% |
| Learning rate | 10-4 |
| Loss function | Cross entropy |
| Timestep | 20 units |
| Optimizer | Gradient descent |

data are available for training, we have here adopted the methodology proposed by Pandey to compensate for this lack.

Pandey's methodology consists in pretraining the model using an alternate dataset with actions that are similar to the target-action movements. With this pairing of actions, the goal is for the network to learn from actions as similar as possible to the target actions during the pretraining.

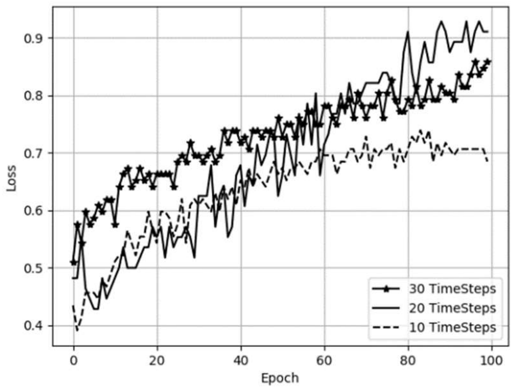


**Fig. 3**. Accuracy tests with a configuration, in which $timeSteps = [30, 20, 10]$.

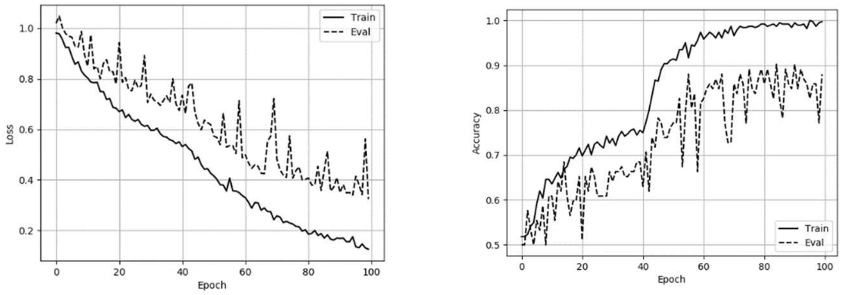


**Fig. 4.** ConvLSTM-FC's error (left) and accuracy (right) with the HMDB51 dataset.

# 3 Background

In this section, we discuss the general features of the architectures used for this paper.

## 3.1 YOLOv2

YOLOv2 addresses object detection as a regression problem in which a single neural network predicts each object's bounding box (BB), and class probabilities are obtained directly from the image to be evaluated [14]. Fig. 1 shows a general diagram of YOLOv2's architecture within the dotted oval.

It is chiefly made up of convolutional layers with batch normalization, which help regularize the model. The input consists of images whose dimensions are 416 x 416, which are divided into $S$ x $S$ cells, where every cell predicts $B$ bounding boxes, which define the frame containing an object, and $C$ class probabilities for every BB. Thus, the output vector size for each image is defined as:

$$S \times S \times (B \times (5 + C)). \tag{1}$$

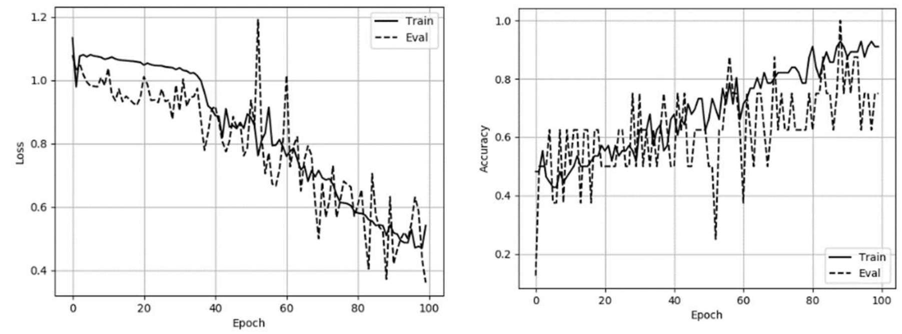


**Fig. 5**. ConvLSTM-FC error (left) and accuracy (right) with the SSDB dataset.

**Table 3.** Loss and accuracy in the ConvLSTM-FC training with the HMDB51 and SSDB datasets.

| | **Loss** | | **Accuracy** | |
|---|---|---|---|---|
| | Train | Eval | Train | Eval |
| HMDB51 | 0.1 | 0.35 | 98 % | 88 % |
| SSDB | 0.5 | 0.45 | 90% | 85 % |

The previous version does not have a location-prediction restriction, which makes the early training iterations unstable. For YOLOv2, an Anchor Box (AB) approach was adopted. Anchor Boxes consist of a predefined set of BBs that are better adjusted to the desired objects and that are calculated by the K-means clustering method.

Instead of predicting the BB coordinates relative to the entire image, coordinates are predicted relative to the location of the cell that contains the bounding box. This limits values to between 0 and 1, for which reason the σ(.) logistic activation function is used to restrict the network predictions to this range. In this way, the final BBs are recovered through:

$$b_x = \sigma(t_x) + c_x, \tag{2}$$

$$b_y = \sigma\left(t_y\right) + c_y, \tag{3}$$

$$b_w = p_w e^{t_w}, \tag{4}$$

$$b_h = p_h e^{t_h}, \tag{5}$$

$$P(\text{Object}) \times IoU(b, \text{Object}) = \sigma(t_0), \tag{6}$$

where the terms $t_{x,y}$ are the normalized BB center predictions with respect to the cell that contains it, $t_{w,h}$ are the normalized width and height predictions with respect to an AB, $c_{x,y}$ represent the coordinates of the cell that contains the upper left corner of the prediction, and $p_{w,h}$ are the AB's dimensions.

Finally, $t_0$ is the confidence of having found an object, which is defined as the probability that an object exists multiplied by the Intersection over the Union of the object's BB against the label.

**Table 4.** Model comparison of HAR performance with the SSDB dataset.

| Model | Accuracy |
|---|---|
| ECO [16] | 80.1 % |
| R(2+1)D [17] | 88.3 % |
| **YOLOv2-ConvLSTM** | **90 %** |
| TSM [18] | 90.5 % |
| I3D+GWS+DR [13] | 95.7 % |

### 3.2 LSTM Convolutional Networks

Recurrent Neural Networks (RNN) are a type of artificial neural network that, because of their architecture, allow for sequential-data processing. An RNN can be abstracted as though it had multiple copies of itself, with each copy processing the input data in an instant of time *t* and sharing the information with its successor through a cycle called time step (*timeStep*), which is repeated $N$ times. The RNN model's output equation in time step *t* is described as:

$$h_t = g(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \tag{7}$$

$$z_t = g(W_{hz}h_t + b_z), \tag{8}$$

where $h_t \in R^N$ is the hidden state with $N$ hidden units, determined by the value input in network $x_t$ in time step *t* and the previous state's vector $h_{t-1}$ in time step $t-1$. The network's response to input $x_t$ in time step $t$ is defined as $z_t$.

The matrices $W$ correspond to the network's weights and $b$ is a bias vector. The nonlinear activation function $g(.)$ is commonly defined as sigmoid or hyperbolic tangent.

The LSTM networks improve upon the RNN and avoid the vanishing gradient problem, thanks to the fact that they are able to remember more information in configurations with long periods of time. Generally, LSTM networks adapt well to applications where input data are one-dimensional vectors.

Nevertheless, in applications where the data are *n-dimensional,* they must be flattened, resulting in the loss of information on spatial correlations during the repetitions. This problem can be addressed if we use LSTM (ConvLSTM) convolutional networks [15]. The representation of the ConvLSTM in a time step *t* given an input $x_t$ and the hidden state $h_{t-1}$ is given by:

$$i_t = \sigma(W_{xi} \otimes x_t + W_{hi} \otimes h_{t-1} + W_{ci} \odot c_{t-1} + b_i), \tag{9}$$

$$f_t = \sigma(W_{xf} \otimes x_t + W_{hf} \otimes h_{t-1} + W_{ci} \odot c_{t-1} + b_f), \tag{10}$$

$$o_t = \sigma(W_{x0} \otimes x_t + W_{hc} \otimes h_{t-1} + W_{co} \odot c_t + b_o), \tag{11}$$

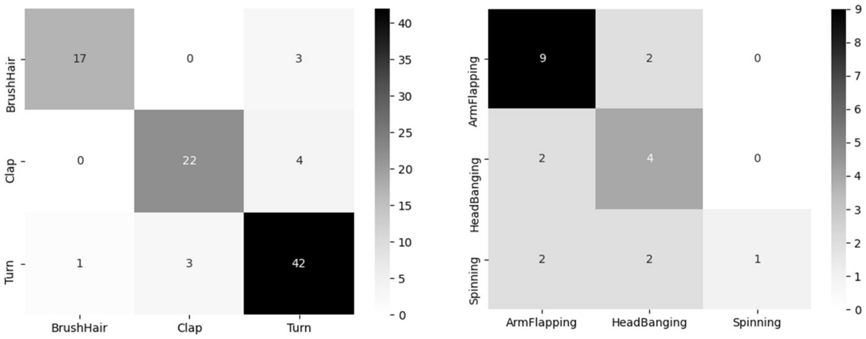


**Fig. 6.** Confusion matrix with the HMDB51 (left) and SSDB (right) validation data.

$$g_t = tanh(W_{xc} \otimes x_t + W_{hc} \otimes h_{t-1} + b_c), \quad (12)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t, \quad (13)$$

$$h_t = o_t \odot \tanh(c_t). \quad (14)$$

In addition to unit $h_t$, the ConvLSTM includes the following: an input gate $i_t$, a gate for forgetting information from earlier states $f_t$, an output gate $o_t$, an input modulation gate $g_t$ and a memory cell $c_t$, which stores information from previous states.

The output gate $o_t$ learns how much is transferred from the memory cell to the hidden state. The symbol $\otimes$ represents the convolutional operation, $\odot$ is the vector product, and $\sigma(.)$ represents the activation function. In the dotted box in Fig. 1, a representation of the basic ConvLSTM unit is shown.

## 4 Proposed Method

The model proposed for the automatic recognition of the actions of children with ASM by video makes use of DL techniques. A diagram showing the elements that make up the model can be seen in Fig. 1.

The YOLOv2 network in cascade is used with the ConvLSTM-FC network to find a function that assigns input video $V_i = \{v_1, v_2, \ldots, v_T\}$ with a length of $T$ frames to the corresponding label $Y \epsilon R^C$, where $C$ is the number of output classes.

Spatial information is sought in phase 1, and thus each frame of video sequence $V_{th}$ is processed using YOLOv 2 to obtain a new representation $X_{th} = \{x_1, x_2, \ldots, x_T\}$ corresponding to the features map of the penultimate layer, and having a dimension of 13 x 13 x 1024.

This new representation is used in phase 2 during the training of the ConvLSTM, during which the images' time order and long-range dependencies in the modelling of human actions are recorded. The ConvLSTM output can be defined as $h_{cLSTM} = ConvLSTM(x_{th})$, and the classification result in the FC layer output is given as $y_{th} = softmax(h_{cLSTM})$.

## 5 Training

The ConvLSTM training is done using the HMDB51 dataset for 100 epochs, and then using the SSDB dataset for the same number of epochs. The mechanism for pairing the HMDB51 and SSDB classes consists in choosing the classes in which the actions are visually similar to the target actions. Table 1 shows the pairing and number of elements per class for training and evaluation. The image resolution varies in each example, with a maximum of 320 x 240 pixels.

In some SSDB videos, it happens that an action is repeated at different moments in the video, or an action might appear that does not correspond to its label. To stop the model from receiving videos in which actions appear that do not belong to the label, the video fragment in which such an action appears was cut out.

In this way, we were able to increase the number of examples from 75 to 113. Some examples of actions from both of the datasets that were used are shown below in Fig. 2, in which very dynamic settings, as well as poor lighting, can be seen.

Due to limitations in computational resources, the new video-frame representations are done offline.Each video of the datasets is processed by the YOLOv2 architecture with the model's original weights, and their new representation is recovered from the penultimate layer of the network, having a dimension of 13 x 13 x 1024.

These new representations are used for training the ConvLSTM-FC network. Table 2 shows the ConvLSTM-FC hyperparameters used during the training.

The computer used has an Intel Xeon 3.5 GHz x 8 processor, 48 GB of RAM memory, and two GeForce RTX 2080 Ti graphics cards with 11 GB of memory each. The open-access TensorFlow v1.6 libraries and the Python 3.0 programming language are run in the Ubuntu 18 operating system.

## 6 Results

To determine the best option for the timeSteps value in the ConvLSTM network, different tests were run to evaluate the best configuration. Fig. 3 shows the model's performance with a configuration of 30, 20, and 10.

One can see that with a configuration in which *timeSteps*=20, the network performs best. For both datasets, the cost and accuracy functions are monitored during the ConvLSTM-FC network training. Fig. 4 shows the action-classification performance with the HMDB51 dataset, taking into account that only the actions mentioned in Table 1 were used.

Once the network has learned to classify actions that are similar to the target actions, training the network with the SSDB dataset begins in order to refine the knowledge. Fig. 5 shows the network's performance during the training, in which we see less learning stability, especially in the validation data.

The results shown for the cost and accuracy functions in the figures above are summarized in Table 3. In the case of training with the SSDB dataset, the model shows a better performance and is slightly more stable between epochs 90–95. For this reason, the number of epochs is used as a point of reference for evaluating the model.

One way of representing the model's classification performance is by using a confusion matrix. Fig. 6 shows the results with the HMDB51 and SSDB evaluation data. Table 4 shows a comparison of different models for HAR in their performance with the SSDB dataset. Results with training data are taken into account for this comparison.

## 7 Conclusions

For this study, a model based on deep learning techniques was implemented for the classification of the actions of children with ASM by video. The experimental tests show that the model made up of the YOLOv2 network in cascade with the ConvLSTM obtains a result within the acceptable range for the state of the art.

Using video frames processed by the YOLOv2 architecture helps the ConvLSTM-FC network in its task of classifying SMM actions. In spite of the fact that the model commits a higher rate of errors with SSDB than with HMDB51, it shows a favorable response in accuracy both with training and evaluation data.

Upon comparing the results obtained with other HAR models using the SSDB dataset, we see that the model proposed here maintains an acceptable range for the state of the art. For future research, we suggest using this model and modifying the cost function for the task of human action localization, which consists of determining the moment in the video in which a human action begins or ends.

For this problem, we suggest considering not just confidence and class probabilities, but also the percentage of the action performed in each frame and considering a new labelling, as well as the incorporation of attention mechanisms in the video frames.

## References

1. Kossyvaki, L.: Adult interactive style intervention and participatory research designs in autism: Bridging the gap between academic research and practice. Routledge (2017) doi: 10.4324/9781315719375
2. Kang, J., Han, X., Song, J., Niu, Z., Li, X.: The identification of children with autism spectrum disorder by SVM approach on EEG and eye-tracking data. Computers in biology and medicine. vol. 120 (2020) doi: 10.1016/j.compbiomed .2020.103722
3. Loftin, R. L., Odom, S. L., Lantz, J. F.: Social interaction and repetitive motor behaviors. Journal of Autism and Developmental Disorders, vol. 38, pp. 1124–1135 (2008) doi: 10.1007/s10803-007-0499-5
4. Costa, A. P., Charpiot, L., Lera, F. R., Ziafati, P., Nazarikhorram, A., van Der Torre, L., Stffgen, G.: More attention and less repetitive and stereotyped behaviors using a robot with children with autism. In: Proceedings of 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), IEEE pp. 534–539 (2018) doi: 10.1109/ROMAN.2018.8525747
5. Sadouk, L., Gadi, T., Essoufi, E. H.: A novel deep learning approach for recognizing stereotypical motor movements within and across subjects on the autism spectrum disorder. Computational intelligence and neuroscience, vol. 2018 (2018) doi: 10.1155/2018/ 7186762
6. Groekathöfer, U., Manyakov, N. V., Mihajlovic, V., Pandina, G., Skalkin, A., Ness, S., Bangerter, A., Goodwin, M. S.: Automated detection of stereotypical motor movements in autism spectrum disorder using recurrence quantification analysis. Frontiers in neuroinformatics, vol. 11, no. 9 (2017)

7. Rajagopalan, S., Dhall, A., Goecke, R.: Self-stimulatory behaviours in the wild for autism diagnosis. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 755–761 (2013)
8. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: 2011 International Conference on Computer Vision, IEEE pp. 2556–2563 (2011) doi: 10.1109/ICCV.2011.6126543
9. Zunino, A., Morerio, P., Cavallo, A., Ansuini, C., Podda, J., Battaglia, F., Veneselli, E., Becchio, C., Murino, V.: Video gesture analysis for autism spectrum disorder detection. In: Proceedings of 24th International Conference on Pattern Recognition (ICPR), IEEE pp. 3421–3426 (2018)
10. Tian, Y., Min, X., Zhai, G., Gao, Z.: Video-based early ASD detection via temporal pyramid networks. In: Proceedings of IEEE International Conference on Multimedia and Expo (ICME), pp. 272–277 (2019) doi: 10.1109/ICME.20 19.00055
11. Sun, K., Li, L., He, N., Zhu, J.: Spatial attentional bilinear 3D convolutional network for video-based autism spectrum disorder detection. In: Proceeding of ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3387–3391 (2020) doi: 10.1109/ICASSP40776.20 20.9054641
12. Silva, V., Soares, F., Esteves, J. S., Vercelli, G.: Human action recognition using an image-based temporal and spatial representation. In: Proceedings of 12th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT), IEEE, pp. 41–46 (2020) doi: 10.1109/ICU MT51630. 2020.9222408
13. Pandey, P., Prathosh, A., Kohli, M., Pritchard, J.: Guided weak supervision for action recognition with scarce data to assess skills of children with autism. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 463–470 (2020)
14. Redmon, J., Farhadi, A.: YOLO9000: Better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
15. Xingjian, S., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., Woo, W. C.: Convolutional LSTM Network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, pp. 802–810 (2015)
16. Zolfaghari, M., Singh, K., Brox, T.: Eco: Efficient convolutional network for online video understanding. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 695–712 (2018)
17. Ghadiyaram, D., Tran, D., Mahajan, D.: Large-scale weakly-supervised pretraining for video action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12046–12055 (2019)
18. Lin, J., Gan, C., Han, S.: TSM: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7083–7093 (2019)