Deep learning improves automated rodent behavior recognition within a specific experimental setup

Elsbeth A. van Dam\textsuperscript{a,b,c,*,} Lucas P.J.J. Noldus\textsuperscript{a,b} Marcel A.J. van Gerven\textsuperscript{c}

\textsuperscript{a} Noldus Information Technology BV, Wageningen, The Netherlands
\textsuperscript{b} Department of Biophysics, Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, The Netherlands
\textsuperscript{c} Department of Artificial Intelligence, Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, The Netherlands

ARTICLE INFO

Keywords:
Rodent behavior recognition
Deep learning
Data augmentation
Continuous video analysis
Cross-setup validation

ABSTRACT

Automated observation and analysis of rodent behavior is important to facilitate research progress in neuroscience and pharmacology. Available automated systems lack adaptivity and can benefit from advances in AI. In this work we compare a state-of-the-art conventional rat behavior recognition (RBR) system to an advanced deep learning method and evaluate its performance within and across experimental setups. We show that using a multi-fiber network (MF-Net) in conjunction with data augmentation strategies within-setup dataset performance improves over the conventional RBR system. Two new methods for video augmentation were used: video cutout and dynamic illumination change. However, we also show that improvements do not transfer to videos in different experimental setups, for which we discuss possible causes and cures.

1. Introduction

Observation and analysis of rodent behavior are widely used in studies in neuroscience and pharmacology. Laboratory rats and mice are valuable animal models for psychiatric and neurological disorders to study the behavioral effects of genetic variation, pharmacological treatment, optogenetic stimulation, and other interventions. However, manual annotation of animal behavior by human observers is labor-intensive, error-prone and subjective. Several automated systems are available that have been reported to perform on par with human annotations. They offer the advantage of quick and consistent annotation and are insensitive to bias, drift and the limited sustained attention of human observers. Yet most of them can only recognize behaviors as performed in the training material, recorded in the exact same setting as the training environment. This works fine with in standardized test cages, but in reality a lot of variation exists between rodent test environments used in different laboratories. The performance of the behaviors might also vary with treatment. There is still no adequate solution that works out of the box in the diverse real-life scenarios faced by behavioral researchers.

During the last 20 years, there have been several publications on automated rodent behavior recognition from video. One of the first to publish on this topic were Rousseau et al. (2000), who trained a neural network on pose features and reached an overall agreement of 63% on nine rat behaviors (49% average recall). Note that overall agreement is calculated over frames only, whereas average recall is the average of the proportion of correct frames per class. Subsequently, Dollar et al. (2005) used the bag-of-words approach for activity recognition with 72% average recall on five mouse behaviors. This was followed by the work of Jhuang et al. (2010), who applied a Support Vector Machine and Hidden Markov Model (SVM-HMM) on a combination of biologically-inspired video filter output and location features. They report 76% average recall for eight mouse behaviors. They also compared human to human scoring, which resulted in 72% overall agreement and 76% average recall. Van Dam et al. (2013) presented the EthoVision XT RBR system for automated rat behavior recognition that applies Fisher dimension reduction followed by a quadratic discriminant on highly normalized, handcrafted features derived from tracking and optical flow. They addressed the importance of cross-setup validation in order to assess out-of-sample generalization, and reached 72% overall agreement and 63% average recall on ten classes for both within-setup and across-setup evaluation.

Meanwhile activity recognition research for human activities progressed tremendously, particularly with the advent of deep learning (Simonyan and Zisserman, 2014; Tran et al., 2015, 2018; Wang et al., 2016; Carreira and Zisserman, 2017; Huang et al., 2017). Deep neural networks allow an abstraction from input data to output categories by learning increasingly higher-level representations of the input. By feeding labeled input examples to the network, the network can compare its own output with the desired output and can amplify features...
that are important for discrimination and ignore irrelevant information. Deep networks vary in their architecture: the number and size of layers and the way information can flow through. Ideally, the network learns the mapping from input data to output class without any preprocessing, in a so-called end-to-end manner.

A few deep learning models have been applied to rodent behavior: Kramida et al. (2016) applied an LSTM model to the VGG-features and report 7% overall failure on a highly imbalanced test set with four mouse behavior classes. More recently Le and Murari (2019) applied a combination of 3D-CNN and LSTM on the dataset from Jhuang et al. (2010) and report comparable results as Jhuang et al. with only end-to-end input. Finally, Jiang et al. (2019) propose a hybrid deep learning architecture with a combination of unsupervised and supervised layers followed by an HMM. They outperform Jhuang et al. on their mouse behavior dataset with overall agreement of 81.5% vs 78.3% and an average recall of 79% vs 76%. They also show that, after retraining, their method is applicable to another task with different classes in a slightly different setup.

As stated above, in order to be useful in behavioral research, an automated system must be able to recognize behaviors independent of treatment and laboratory setup. Good recognition performance on a dataset recorded in one setup is an important step. However, retraining supervised systems on a new setup requires a lot of data and brings back the manual annotation task for a significant number of video segments. Three approaches are possible to get closer to the goal. One direction is to standardize laboratory setups (Arroyo-Araujo et al., 2019). Second is to aim for quick adaptation of a classifier towards a new setup with minimal annotation effort, i.e. learn from limited data. The third is to strive for generic recognition with robust and adaptive methods.

Deep learning might provide the key to achieving these goals. Development time is reduced since laborious handcrafting of features is not needed anymore. Without dedicated features we might also be less restricted in the application, and less preprocessing avoids noise being introduced by it. Furthermore, trained networks can be partially reused so we do not need to train from scratch in a different but comparable scenario. The downside of neural networks is the amount and variety of data needed to train them.

The goal of this study is to compare our earlier handcrafted rodent behavior classification system to end-to-end classification by an advanced deep learning network for action recognition, to evaluate the flexibility of the recognition on unseen setups, and to learn how it can be improved. In Section 2 we explain network, metrics, datasets, sampling and augmentation. In Section 3 we present classification results of within-setup and across-setup recognition, which we discuss in Section 4. We conclude in Section 5.

2. Methods and materials

We address two questions. First, what is the performance of an advanced action recognition deep learning network on a rodent behavior dataset? We experiment on a dataset of short rat behavior clips and apply two different input schemes, namely (a) end-to-end input without preprocessing, vs (b) region-based input from tracking information, i.e. regions of interest around the animal + optical flow to capture the motion. We train with and without data augmentation. The second question we address is aimed to investigate applicability in real-life scenarios: what is the performance of this network on continuous videos and across setups? We evaluate videos of rat behavior using the best performing input scheme and compare within-setup and across-setup classification performance.

2.1. Network

As network architecture we used the multi-fiber network (MF-Net) described in Chen et al. (2018). Fig. 1 shows the diagram of the network. The choice of this network was based on its good performance on the currently most important benchmark datasets for activity recognition, e.g. UCF-101 (Soomro et al., 2012), HMDB-51 (Kuehne, 2012) and Kinetics (Carreira and Zisserman, 2017), and its efficiency compared to other well performing networks, namely it needs 9 × less calculations than I3D (Carreira and Zisserman, 2017) and 30 × less calculations than R(2+1)D (Tran et al., 2018) to get to the same results. The crux of the MF-Net architecture is that it uses an ensemble of lightweight networks (fibers) to replace a complex neural network, reducing the computational cost while improving recognition performance. Multi-plexer modules are used to facilitate information flow between fibers. We used the available code1 with modified sampling, augmentation and performance metric. Furthermore we adjusted the number of layers and kernels to deal with our specific input layout considering resolution and channels.

The network consists of one 3-dimensional convolutional (conv3d) layer followed by four multi-fiber convolution (MFconv) blocks. Each MFconv block consists of multiple MF units, and each MF unit consists of four (five for the first block unit) conv3d layers. All conv3d layers are followed by batch normalization and a rectified linear unit (ReLU). The final layers of the network are an average pooling layer and a fully connected layer. Since the middle layer of every MF unit uses a (3,3,3) kernel there is temporal convolution in 17 conv3d layers and additionally in the last average pooling layer during aggregation of the final eight frames.

We did not initialize the network with a pretrained model since our input channel layout differs from the colored 3-channel human activity datasets that available pretrained networks are trained on.

2.2. Metrics

In large scale activity recognition nowadays the most popular performance metric is top-1 or top-k accuracy, where the top-1 accuracy denotes the overall agreement across frames, i.e. the proportion of the input where the model’s prediction was right and top-k denotes the proportion of the input where the target class was in the model’s top k most likely predictions. However, these measures are misleading for imbalanced datasets with equally important classes as is the case with sampling from continuous videos. Suppose the dominant class covers 80% of the samples and the network classifies all samples as belonging to this class. Then the overall agreement of this obviously bad classifier would be 80%. More informative measures in this situation are precision and recall per class, precision of a class being the proportion of found frames that is correct and recall being the proportion of correct frames found. In this study, we use average recall as an aggregated measure, calculated by taking the average of the behavior class recalls. Although this does not take into account the precision, all ill-labeled samples contribute negatively to the average recall since we take all classes into account. In comparison to the averaged F1-score, false positives of rare classes have more negative impact than those of frequent classes, which is preferable. For comparison with related work, we also report overall agreement per video for the cross-setup evaluation, calculated as the proportion of correct frames per video.

Note that behavior itself is not discrete and behavior changes take time. Therefore, it is good to keep in mind that 100% accuracy is not feasible because of inherent ambiguity at behavior bout boundaries. In all experiments and evaluations, frames not belonging to one of the nine classes are left out of the evaluation. The goal of this study is to compare handcrafted feature classification to end-to-end classification, within and across setups. Although the question how to detect ‘other’ behavior is important for applicability it was left outside the scope of this study.

1 https://github.com/cypw/PyTorch-MFNet.
2.3. Within-setup experiments on clips

2.3.1. Dataset

For the within-setup experiments we used the high quality dataset described by Van Dam et al. (2013). It consists of 25.3 video hours of six Sprague–Dawley rats in a PhenoTyper 4500 cage (http://www.noldus.com/phenotypes) at 720 × 576 pixel resolution, 25 frames per second and with infrared lighting, hence gray-scale. Subsets of these recordings (~2.7 h in 14 subvideos) were annotated by a trained observer using The Observer XT 10.0 annotation software (http://www.noldus.com/observer), and manually checked and aligned afterwards to ensure frame accurate and consistent labeling. Checking and alignment took one hour per five-minute video for 14 classes (including subatomic classes). In this study we focused on the nine most frequent state behavior classes ‘drink’, ‘eat’, ‘groom’, ‘jump’, ‘rest’, ‘rear unsupported’, ‘rear wall’, ‘sniff’ and ‘walk’. The tenth behavior, ‘twitch’, is a point behavior without annotated duration and was left out of the comparison.

The data is presented to the network in two different ways. End-to-end input consists of the gray-scale videos resized to square 224 × 224 resolution. The square crop was made from the center of the arena, after the resize. The end-to-end model trained on the clips subset is referred to as E2e-c. As an alternative to the end-to-end input we removed the tracking task and provided the model with a 88 × 88 moving region-of-interest around the animal. Frame motion information was added with the optical flow (x and y) in the second and third channel. The tracking, flow calculation and cropping was done with EthoVision XT 14.0 (http://www.noldus.com/ethovision), which was modified for this purpose. The second type of input format is referred to as Roi+ flow.

2.3.2. Network details

Because the input resolution differs for the end-to-end (224 × 224 × 1 × 32) and Roi+ flow inputs (88 × 88 × 3 × 32) the network layouts are slightly different. The main difference is that the max pooling layer was omitted in the Roi+ flow layout, because the Roi+ flow resolution needs less spatial reduction. For both networks the total size is ~7.7 M parameters. See Tables 1 and 2 for more details.

2.3.3. Sampling

The within-setup experiments are applied on a set of behavior clips sampled from the within-setup dataset. We perform fourfold cross-validation over different random train/test splits (80/20 per class). In each fold there are 2314 training clips and 398 test clips. Each clip contains 32 consecutive frames. The clip label is the behavior in the middle of the clip, i.e. the annotation of the 17th frame. Clips were randomly picked with the constraint that there is no behavior transition in the middle of the clip, between frame 14 and 19. In the training set, the clips have a maximum overlap of 29 frames and there were never more than four clips selected per behavior bout, with a maximum of 400 clips per behavior. For testing, the maximum overlap is 25 frames and there are never more than two clips selected per behavior bout, with a maximum of 50 clips per behavior. For the exact number of test clips per behavior see Table 4. Clips from the same behavior bout are always together in the same split, so either in the training or in the test set.

2.3.4. Data augmentation

To prevent overfitting the data is augmented by applying a random combination of the following known filters: resized crop, horizontal and vertical flip, inverse, rotation (90/180/270°), luminance variation (brightness, contrast and gamma), additional Gaussian noise, additional salt & pepper noise, image blur. Additionally we applied two new filters: video cutout and dynamic illumination change. Video cutout is the 3D version of 2D-cutout introduced by DeVries and Taylor (2017). It implies adding occlusions to the clip by replacing randomly located cuboids with the mean clip value. Dynamic illumination change is created by adding a random 3D Gaussian to the clip, which has the effect of gradually turning on or dimming a spotlight on a random time and location in the clip. For Roi+ flow the flow was calculated after random rotation and inverse of the video frame, and modified implementations where made for the flipping filters to flip the optical flow vectors. Resized crop was omitted and luminance variation was only applied to the gray-scale channels. Dynamic illumination change was also omitted for Roi+ flow since it would affect the optical flow calculation.

After augmentation, the clips were normalized to have a mean of 0 and standard deviation of 1. Normalization was done per channel to
avoid mixing image and optical flow information.

2.4. Continuous and cross-setup experiments

2.4.1. Dataset

In these experiments we used the cross-setup validation dataset from Van Dam et al. (2013) illustrated in Fig. 2 and Table 3. It contains one video from the within-setup dataset and four videos recorded with different resolution, animal strain, illumination, background and feeder and spout positions. Frame rate and camera viewpoint were not changed, and all recordings were made with constant lighting and good contrast between animal and background. Table 5 presents the performance of the conventional RBR system on these videos.

2.4.2. Sampling

In order to estimate robustness in real-life scenarios we next evaluate the performance across experimental setups and on continuous videos. Unlike the within-setup experiments that were conducted on a balanced subset of clips and ignored clips around behavior bout transitions, the model is now deployed on sliding-window clips (32 frames wide, step size 1 frame). These clips contain more ambiguous data than the subset of clips used in our within-experiments and the set is not balanced anymore.

In the cross-setup experiments we consider only the end-to-end input scheme. We applied the E2e-c model that was trained on the entire balanced clips dataset (2712 clips) to the sliding-window clips of the test videos. Alternatively, we retrained the model on all sliding-window clips from the within-setup dataset (32 frames wide, step size 4). This model is referred to as E2e-s. The sliding-window clips set is much bigger (52,560 clips) and not balanced anymore. To account for the imbalance during training we used weighted random sampling. This way during every epoch the less frequent behaviors are presented to the network more often. Since random augmentation is applied, the network sees different versions of the clips. For evaluation of within Video 1, the models were retrained without the clips of Video 1.

3. Results

All experiments were conducted on a Dell Precision T5810 with

| Layers | #MF units | #channels | #frames | Width | Height | Kernel | Stride | Padding |
|--------|-----------|-----------|---------|-------|--------|--------|--------|---------|
| Input  | 1         | 32        | 224     | 224   |
| conv3d | 16        | 16        | 112     | 112   |
| maxpool| 16        | 16        | 56      | 56    |
| MFCconv1 | 3     | 96        | 56      | 56    |
| MFCconv2 | 4     | 192       | 56      | 56    |
| MFCconv3 | 6     | 384       | 14      | 14    |
| MFCconv4 | 3     | 768       | 8       | 8     |
| Avg pool | 1       | 1         | 1       | 1     |
| FC     | 400       |           |         |       |

Table 2 Details of the MF-Net architecture in Roi+ flow experiments.

| Layers | #MF units | #channels | #frames | Width | Height | Kernel | Stride | Padding |
|--------|-----------|-----------|---------|-------|--------|--------|--------|---------|
| Input  | 3         | 32        | 88      | 88    |
| conv3d | 32        | 32        | 44      | 44    |
| MFCconv1 | 3     | 96        | 11      | 11    |
| MFCconv2 | 4     | 192       | 6       | 6     |
| MFCconv3 | 6     | 384       | 1      | 1     |
| MFCconv4 | 3     | 768       | 1      | 1     |
| Avg pool | 1       | 1         | 1       | 1     |
| FC     | 400       |           |         |       |

Table 3 Description of the five videos with different setups used for cross evaluation.

| Video | Duration | Resolution | Setup                      |
|-------|----------|------------|----------------------------|
| Video 1 | 13.7 min | 720 × 576  | One video from the within-setup dataset |
| Video 2 | 10.5 min | 360 × 320  | Half resolution, different sawdust |
| Video 3 | 5.0 min  | 720 × 576  | Different strain (Wistar) |
| Video 4 | 5.3 min  | 768 × 576  | Visible light |
| Video 5 | 2.5 min  | 720 × 576  | Visible light, no sawdust |

Table 4 Recall per behavior of the end-to-end model, tested on clips test sets and on continuous video, all within-setup. For the model applied to the clips test set both the fourfold and best fold results are presented.

| Video | Clips test set | Frames | Fourfold | Best fold | E2e-c | E2e-s |
|-------|----------------|--------|----------|-----------|-------|-------|
| Video 1 | 398          | 18,872 | 0.75     | 0.79      | 0.55  | 0.65  |
| Overall agreement | 398 | 18,872 | 0.76     | 0.80      | 0.48  | 0.77  |

Bold values is the best performing model, measured on the same test set (Best model on Video 1, per behavior).
Table 5

|         | RBR  | E2e-s |
|---------|------|-------|
| Within  |      |       |
| Video 1 | 0.71 | 0.77  |
| Across  |      |       |
| Video 2 | 0.65 | 0.50  |
| Video 3 | 0.80 | 0.27  |
| Video 4 | 0.67 | 0.51  |
| Video 5 | 0.65 | 0.59  |

Bold values is the best performing model, measured on the same test set (Best model on entire video).

32 GB memory and a NVIDIA Titan X (Pascal) GPU with 12 GB, running Ubuntu 18.04, with Python 3.7 using the PyTorch framework (0.4.1).

3.1. Within-setup evaluation on clips

Fig. 3 presents violin plots showing the classification results on the balanced clips dataset with and without data augmentation for the two different input schemes, for all folds. The end-to-end input scheme with data augmentation yields the best result of 75% average recall. The results per behavior are listed in Table 4, for both average fold and best fold. The effect of increasing data augmentation is shown in Fig. 4. The confusion matrices in Fig. 5 show that accuracy is high for almost all classes, except 'rest' (that only has 16 frames). Therefore, we observe that the network overtrains without augmentation. The network can learn without augmentation.

3.2. Across-setup evaluation on continuous videos

First, we evaluated the E2e performance on continuous videos. We compare two models: E2e-c trained on the cleaner and balanced clip dataset, and E2e-s trained on the much bigger but noisier dataset of sliding-window clips as it contains also clips with behavior bout transitions in the middle. We test both models on the sliding-window clips of Video 1. In Table 4 we see that having good performance on an unseen set of clips is not enough to guarantee performance on continuous videos. Instead, the E2e-s model performs better on all behaviors except ‘rest’ (Video 1). Figure 7 shows the event log for within-setup Video 1.

Next, we evaluated the E2e-s model on our set of videos in varying setups. Table 5 presents the overall agreement per video, Table 6 shows the recall per behavior. Compared to handcrafted-feature classification, E2e-s outperforms RBR on the within-setup evaluation, but not on the cross-validation task. This holds for all four cross-setup videos and for all classes except ‘groom’ and ‘jump’. Performance was decreased especially for Video 3, which is mostly due to the large amount of false negatives for ‘sniff’ and false positives on ‘rest’ (see the event log in Fig. 8). From the results per class the misclassifications of ‘drink’, ‘eat’ and ‘rear wall’ behaviors stand out compared to RBR.

4. Discussion

First, we interpret the within-setup mistakes of E2e-c. Looking at the confusion matrix in Fig. 5 we see that most confusion comes from ‘jump’/’walk’, ‘sniff’/’eat’ and to a lesser extent from ‘eat’/’sniff’ and ‘sniff’/’walk’. These are understandable mistakes since these are gradually overlapping behaviors that can be performed more or less at the same time and hence easily subject to interpretation differences. In these cases, automated annotation is probably even more consistent than human annotation that is more sensitive to context.

Second, we interpret the mistakes made by E2e-s on continuous video, within setup. The event logs for Video 1 (Fig. 7) show very good correlation between human and E2e-s annotation. It stands out that there are more behavior switches in the E2e-s annotation. This suggests that although E2e-s classification contains many temporal filters, it could still benefit from post processing, either by explicitly averaging the soft-label output over time or by adding a recurrent layer after the FC layer. Many of the related work methods use recurrence in their classification. This is helpful to smoothen the output and helps the algorithm to suppress detection of unlikely behavior sequences. However, this makes these systems less applicable to annotate behavior of drug-treated animals. In these cases, the behavioral transition probabilities might be altered, and instead of being part of the model, these changed transitions are a result of the experiment.

Thirdly, let us examine the poor results of E2e-s on the cross-setup tasks. Looking at event logs for Video 3 in Fig. 8 it is notable that many ‘sniff’ frames are mistakenly detected as ‘rest’ or ‘rear unsupported’.
Also, many ‘eat’ behaviors are mistaken for ‘rest’. Looking at the video reveals that the animal in Video 3 is very cautious and pauses a lot during its movements. Although this behavior was labeled by the human as ‘sniff’ it is a type of sniff that was not in the training data where the animals are more at ease. RBR does not suffer from these mistakes, possibly because the decision making is more integrated over time and ‘rest’ will be only detected when the animal does not move for a longer period.

The E2e model especially fails to recognize environment-dependent behaviors in the cross-setup task. Behaviors ‘drink’, ‘eat’ and ‘rear wall’ score below 40% while the recognition of these three behaviors is over 80% in the within-setup evaluation. Handcrafted RBR has an advantage since the location of the drinking spout, feeder zone and walls are provided by the user. However, the network should be able to ‘see’ the feeder and the edges of the floor in all setup videos, and deduce that the drinking spout is always on the side of the arena. Also during augmentation all clips are rotated and resized hence the model should be robust to changes in the exact position of walls, feeder and spout.

Future work will be to experiment with adding a recurrent layer to the network, adding augmentation that varies backgrounds and adding explicit visible environment cues to the input video. Alternatively, we can optimize networks for specific setups. Still the most challenging problem will be to address the unseen behavior variation that caused wrong automated annotations of Video 3. A first step can be to detect...
abnormal behavior sequences and let the user tell the network how to interpret the sequence. This requires learning from fewer data examples.

Finally a word on speed. RBR (including video IO, tracking and feature extraction by EthoVision) can annotate $\sim 124$ frames/s, i.e. almost five times faster than real-time on a CPU (Dell Precision T3620 with 8 GB, Intel Xeon E3-1240 v6 @3.7 GHz). MF-Net with end-to-end input runs the forward call $\sim 230$ frames/s on the Titan X GPU.

5. Conclusion

In this study, we addressed the problem of automated rodent behavior recognition and compared the accuracy performance of an advanced deep learning approach (MF-Net) to conventional handcrafted classification (RBR). For within-setup performance on a clipped dataset we showed that MF-Net with end-to-end input outperforms both handcrafted RBR and MF-Net with Roi+flow input, provided sufficient data augmentation. For cross-setup performance on continuous video, we showed that MF-Net with end-to-end input could not outperform RBR. We argue that the end-to-end model has difficulty recognizing environment cues and is not robust to differences in behavior sequences observed, which is a problem for animals behaving different than

| Event log | Video 1 | Ground Truth | Event log | Video 1 | MFNet E2e-s |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Drink (689) | 0.42 | 0.39 | (1181) | 0.93 | 0.38 |
| Eat (3186) | 0.79 | 0.53 | (6691) | 0.76 | 0.10 |
| Groom (331) | 0.74 | 0.86 | (1900) | 0.58 | 0.73 |
| Jump (0) | – | – | (73) | 0.48 | 0.99 |
| Rear unsupported (559) | 0.70 | 0.76 | (871) | 0.74 | 0.63 |
| Rear wall (2908) | 0.69 | 0.94 | (2240) | 0.63 | 0.05 |
| Rest (16) | 0.00 | 0.00 | (1003) | 0.16 | 0.27 |
| Sniff (2453) | 0.70 | 0.80 | (15,352) | 0.67 | 0.65 |
| Walk (10,290) | 0.86 | 0.89 | (2127) | 0.60 | 0.55 |
| Average recall (18,872) | 0.62 | 0.65 | (31,446) | 0.62 | 0.45 |

Bold values is the best performing model, measured on the same test set (Best model per behavior, on Video 1, and best model per behavior on Video 2-5).
normal, for instance due to treatment. We conclude that deep learning networks give us good performance on fixed setups with known behavior, but that more research is necessary to reach adaptive and flexible human-like performance that is independent of the setup and behavior performance.

Authors’ contribution

Elsbeth van Dam: conceptualization; data curation; formal analysis; investigation; methodology; project administration; software; validation; visualization; roles/writing – original draft. Lucas Noldus: conceptualization; funding acquisition; resources; supervision; writing – review & editing. Marcel van Gerven: conceptualization; methodology; supervision; writing – review & editing.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. EvD and LN are fully employed by Noldus Information Technology BV. EvD and MvG designed the study, EvD conducted the experiments and wrote the paper, MvG and LN reviewed the article.

References

Arroyo-Araujo, M., Graf, R., Maco, M., van Dam, E., Schenker, E., Drinkenburg, W., Koopmans, B., de Boer, S.F., Callum-Doyle, M., Noldus, L.P.J.J., Loos, M., van Dommelen, W., Spooren, W., Biemann, B., Buhl, D.L., Kas, M.J., 2019. Reproducibility via coordinated standardization: a multi-center study in a Shank2 genetic rat model for autism spectrum disorders. Sci. Rep. 9 (1), 11602.
Carreira, J., Zisserman, A., 2017. Quo vadis, action recognition? A new model and the kinetics dataset. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 6299–6308.
Chen, Y., Kalantidis, Y., Li, J., Yan, S., Feng, J., 2018. Multi-fiber networks for video recognition. Proceedings of the European Conference on Computer Vision (ECCV) 352–367.
DeVries, T., Taylor, G.W., 2017. Improved Regularization of Convolutional Neural Networks with Cutout. arXiv:1708.06552.
Dollar, P., Rabaud, V., Cottrell, G., Belongie, S., 2005. Behavior recognition via sparse spatio-temporal features. 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance 65–72.
Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2261–2269.
Jhuang, H., Garrote, E., Yu, X., Khilnani, V., Poggio, T., Steele, A.D., Serre, T., 2010. Automated home-cage behavioural phenotyping of mice. Nat. Commun. 1 (6), 1–9.
Jiang, Z., Crookes, D., Green, B.D., Zhao, Y., Ma, H., Li, L., Zhang, S., Tao, D., Zhou, H., 2019. Context-aware mouse behavior recognition using hidden Markov models. IEEE Trans. Image Process. 28 (3), 1133–1148.
Kuehne, H., 2012. Hmdb: A Large Video Database for Human Motion Recognition. ieeeexplore.ieee.org.
Le, V.A., Murari, K., 2019. Recurrent 3D convolutional network for rodent behavior recognition. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 1174–1178.
Rousseau, J.B.I., Van Lochem, P.B.A., Gispen, W.H., Spruijt, B.M., 2000. Classification of rat behavior with an image-processing method and a neural network. Behav. Res. Methods Instrum. Comput. 32 (1), 63–71.
Simonyan, K., Zisserman, A., 2014. Two-stream convolutional networks for action recognition in videos. Advances in Neural Information Processing Systems 568–576.
Soomro, K., Zamir, A.R., Shah, M., 2012. UCFC101: A Dataset of 101 Human Actions Classes From Videos in The Wild.
Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M., 2015. Learning spatiotemporal features with 3D convolutional networks. Proceedings of the IEEE International Conference on Computer Vision (ICCV) 4489–4497.
Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M., 2018. A Closer Look at Spatiotemporal Convolutions for Action Recognition. CVPR.
Van Dam, E.A., Van Der Harst, J.E., Ter Braak, G.J.F., Tegelenbosch, R.A.J., Spruijt, B.M., Noldus, L.P.J.J., 2013. An automated system for the recognition of various specific rat behaviours. J. Neurosci. Methods 218 (2), 214–224.
Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L., 2016. Temporal segment networks: towards good practices for deep action recognition. Springer European Conference on Computer Vision (ECCV) 20–36.