This is an Accepted Manuscript version of the following work: Áron Fóthi, Kinga B. Faragó, László Kopácsi, Zoltán Á. Milacski, Viktor Varga, András Lőrincz: Multi Object Tracking for Similar Instances: A Hybrid Architecture. Neural Information Processing, Springer International Publishing 2020. This version of the manuscript has been accepted for publication, after final editorial and peer review is complete, but is not the Version of Record. The final authenticated version is available online at: https://doi.org/10.1007/978-3-030-63830-6_37

Multi Object Tracking for Similar Instances: a Hybrid Architecture

Áron Fóthi ✉, Kinga B. Faragó, László Kopácsi, Zoltán Á. Milacski, Viktor Varga, and András Lőrincz

Department of Artificial Intelligence, Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary
{fa2,faragokinga,kopacsi,miztaai,vv,lorincz}@inf.elte.hu

Abstract. Tracking and segmentation of moving objects in videos continues to be the central problem in the separation and prediction of concurrent episodes and situation understanding. Along with critical issues such as collision avoidance, tracking and segmentation have numerous applications in other disciplines, including medicine research. To infer the potential side effects of a given treatment, behaviour analysis of laboratory animals should be performed, which can be achieved via tracking. This presents a difficult task due to the special circumstances, such as the highly similar shape and the unpredictable movement of the subject creatures, but a precise solution would accelerate research by eliminating the need of manual supervision. To this end, we propose Cluster R-CNN, a deep architecture that uses clustering to segment object instances in a given image and track them across subsequent frames. We show that pairwise clustering coupled with a recurrent unit successfully extends Mask R-CNN to a model capable of tracking and segmenting highly similar moving and occluded objects, providing proper results even in certain cases where related networks fail. In addition to theoretical background and reasoning, our work also features experiments on a unique rat tracking data set, with quantitative results to compare the aforementioned model with other architectures. The proposed Cluster R-CNN serves as a baseline for future work towards achieving an automatic monitoring tool for biomedical research.

Keywords: Instance Segmentation · Tracking · Constrained Clustering · Gated Recurrent Unit · Deep Neural Networks

1 Introduction

Multi-Object Tracking (MOT) is a video processing problem with numerous possible applications in practice. Detecting humans and providing their bounding boxes can be used in video surveillance in order to perform various tasks without supervision, such as monitoring crowd density or recognising suspicious behaviour. The Multi-Object Tracking and Segmentation (MOTS) problem expands on this idea by requiring the detected objects to be segmented. Naturally,

it presents a harder problem, while also making it more difficult to label data manually for training purposes. However, precise models allow for finer observations due to the pixel-wise segmentation provided for each object of interest.

The task becomes harder when tracking interest is shifted from humans animals, especially if the subject creatures are very similar to each other. One can find multiple additional purposes for tracking and segmentation. Regarding medicine research for example, where animal trials are commonplace, automated tracking can aid in solving technical and ethical complexities. Behaviour analysis of laboratory mice and rats is traditionally done manually, either in real time or from a video recording (see Fig. 1), which is a highly monotonous task. RFIDs can eliminate this obstacle [3], but the operation has its own complications and minimising the pain and discomfort of these animals is also of utmost importance. Without means of automation, this also requires continuous monitoring, once again imposing the disadvantages mentioned above. The deep network architecture presented in this paper[1] serves as a first step towards a multidisciplinary project, in which domain specific knowledge in neuroscience could be combined with artificial intelligence to perform as many of these tasks as possible without human supervision.



Fig. 1. Examples of rat actions to monitor. From left to right: sniffing, head-to-head, mounting/being mounted (the distinction is important from a given rat's point of view), side by side.

In experiments on social behaviours some of the animals may be medicated and/or genetically manipulated making their respective tracking and motion analysis mandatory. The problem is difficult for a variety of reasons. For one, the animals have to be tracked with meticulous precision to yield accurate results for such a vital research. Losing track of the rats invalidates any following tracking attempts, as they have no means of visual distinction (marking the animals might

[1] https://github.com/aronfothi/mask_cluster_rcnn

influence their behaviour). Frequent mounting and occlusion scenarios, coupled with unpredictable movements make this task even harder.

The structure of our work is as follows. In the next section, we review related architectures and methods (Sect. 2). This is followed by a description of our contributions to the aforementioned rat tracking problem (Sect. 3), implementation details (Sect. 4) and experimental results (Sect. 5.2). Finally, the main observations and expansion possibilities are discussed (Sect. 6), and a summary of our work is provided (Sect. 7).

2 Related work

Multi-Object Tracking. As mentioned before, MOT expects the model to track the bounding boxes of multiple objects simultaneously through frames of a video. Most existing tracker architectures follow the two-step *tracking-by-detection* paradigm:

1. Detect objects on each frame independently,
2. Link the appropriate detections together to form the trajectories.

The Tracktor [1] by Bergmann *et al.* attempts to combine these into a single, unified process. In their case, the bounding boxes of the detected objects serve as the region of interest (RoI) proposals for the next frame. If this proposal is accepted by the regressor with a sufficiently high score, then the corresponding object on the current frame matches the one on the previous image, otherwise the object likely left the camera's field of view. Newly detected objects without a previous proposal might actually be past objects that reappeared on the feed, which is tested by a re-identification algorithm.

Multi-Object Tracking and Segmentation. The MOTS problem [10] was first coined by Voigtlaender *et al.* as an extension of Multi-Object Tracking. Along with the problem specification, the authors present two data sets for the task based on existing ones, a novel accuracy score for evaluation and a baseline architecture, TrackR-CNN. TrackR-CNN uses Mask R-CNN [4] as a backbone, followed by employing 3D convolutions on the feature maps to attain temporal context. The architecture is further complemented by an *association head*, which creates embedding vectors (referred to as 'association vectors') for each detection. The distance between two given vectors from different frames determines whether they belong to the same instance (and therefore trajectory) or not.

Inspired by the new task and data sets, Porzi *et al.* published a paper [9] with several key contributions. One of these refers to a semi-automatic way of generating training data, mending the shortcoming of KITTI MOTS, which contains only a modest amount of annotations. They also propose MOTSNet, which resembles the previously mentioned TrackR-CNN in a sense that a Mask R-CNN backbone is amended with an additional head working with embedding

vectors. However, MOTSNet makes use of the mask in the tracking head, in addition to the features of the feature pyramid network (FPN). The tracking head utilises a batch hard triplet loss, which is added to the losses of the other heads.

Clustering for Instance Segmentation. As opposed to proposal-based methods for instance segmentation, proposal-free methods label pixels without bounding box information. This usually involves learning a pixel-level representation, followed by clustering to obtain pixel groups corresponding to instances. Hsu *et al.* [6] solve the problem in a single forward pass, using a Fully Convolutional Network (FCN) to learn a clustering mechanism based on pixel pair relations. The FCN outputs probability distributions for each pixel as a form of soft clustering, after which the authors use Kullback-Leibler divergence as a distance-like metric. The proposed architecture is fairly fast, performs well, and further supports an unlimited number of instances through a graph colouring inspired extension.

Temporal extension of Mask R-CNN. There have been numerous attempts at injecting temporal information into Mask R-CNN-based architectures. For example, Siamese networks work with input pairs which are later combined into a joint branch. SiamFC [2] takes a crop of the target object and a larger image centered on its last position estimate, and processes them with the same convolutional networks before calculating their cross-correlation. Wang *et al.* propose SiamMask [11], which introduces a binary segmentation head to output masks for each frame.

Incorporating recurrent layers is a more intuitive approach in this regard. Porzi *et al.* [9] employ an RNN in their study, but note that it fails to make a considerable impact on accuracy.

3 Methods

3.1 Annotation and augmentation

Due to the slow process of pixel-level human annotation, MOTS problems are usually lacking labelled data. Therefore, we looked for methods to automatically generate training data. We considered two cases: frames with and without contact between the animals. Due to the cage design it is simple to differentiate rats from the background (the floor), segmentation of the separate animals is fairly straightforward with traditional computer vision algorithms. As for the former group (featuring rats close to one another), generating these samples can be achieved through augmentation, offering various approaches. First, one of the rats is removed and replaced with the background, followed by cloning and translating the other:

- **Rotation:** The cloned creature is placed on an arc near the original.

– **Following:** The remaining rat is duplicated with a slight spatial and temporal offset, essentially following the original instance
– **Crossing:** In addition to moving the duplicate, it is also flipped horizontally giving rise to crossing paths.
– **Mixing:** Rat trajectories from different times are combined to form a rich set of overlaps

3.2 Segmentation by clustering

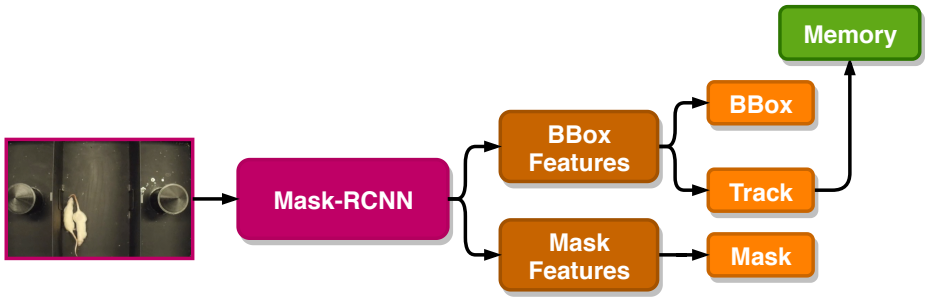

Fig. 2. MaskTrack R-CNN architecture. A tracking head with a memory queue was added to the Mask R-CNN architecture to handle the video instance segmentation problem

MaskTrack R-CNN [13] approaches the video instance segmentation problem by expanding Mask R-CNN with a tracking head and a memory queue to keep track of objects throughout the video (see, Fig. 2). Assigning instance IDs is aided by bounding box information, but ignores segmentation results. As a result, coinciding bounding boxes reveal a weakness in this architecture, which is also mentioned in the paper. This issue is also prominent in our task, as tail curls may produce completely overlapping bounding boxes even without any real occlusion.

We propose a clustering head to tackle the aforementioned problem, inspired by [6]. We integrate their proposal-free clustering method into the proposal-based Mask R-CNN to improve accuracy. Instead of using Kullback-Leibler divergence with instance indexes, we opt for a labelless implementation with constrained clustering likelihood (CCL).

Our contributions to the aforementioned clustering methods are as follows: our network takes the RoI proposals, selects pixels from each and compares them to get their similarity (see Fig. 3). The clustering head also clusters the RoIs, and because a single RoI could contain more than one rat, the cluster mask output is appended to the feature map. This step is similar to the method applied in MOTSNet [9] but, rather, masking the background, we concatenate it to the feature map. In addition to examining positive and negative pairs in a given frame, the same comparison can be done between pixels in different frames to

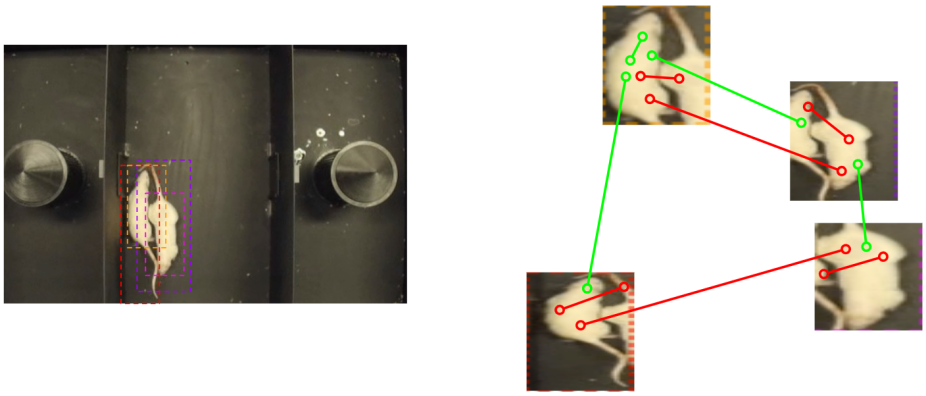

Fig. 3. Mask R-CNN proposes multiple Regions of Interest, which are then resized and aligned. Our custom clustering head selects a given number (20 in our experiments) of pixels from each RoI, which are then compared as seen in this image. Positive and negative pairs are identified inside each region and also across different RoIs.

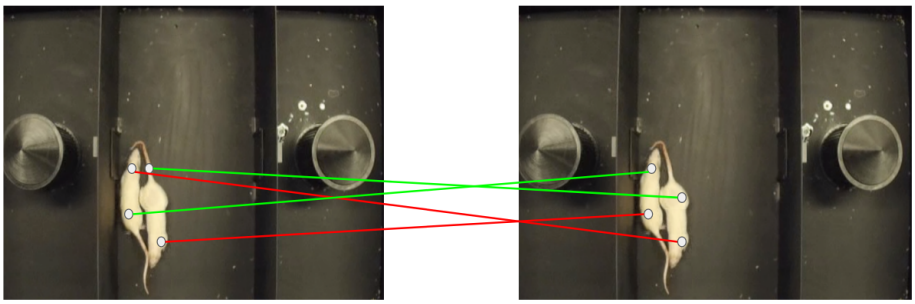

Fig. 4. Constrained Clustering Likelihood extended in time. Positive pairs are shown in green, while negative ones are coloured red.

extend the process through time for tracking purposes (as can be seen in Fig. 4). This approach mends the aforementioned shortcoming of MaskTrack R-CNN, as shown in Fig. 5.

Formulation The clustering method used here has been suggested by [5]. Let $X = \{x_1, \ldots, x_n\}$ be a set of pixels sampled from the RoI proposals throughout the video. Each of these samples belong to a single cluster $c \in \{1, .., k\}$. The clustering head $f_\theta(.)$ calculates the distribution that a given pixel x_i belongs to some cluster c:

$$f_\theta(x_i) = [p_{i1}, p_{i2}, \ldots, p_{ik}] \quad \left(\sum_{c=1}^{k} p_{ic} \geq 1 \right),$$

where $P(y_i = c|x_i) = P(y_i = c) = p_{ic}$.

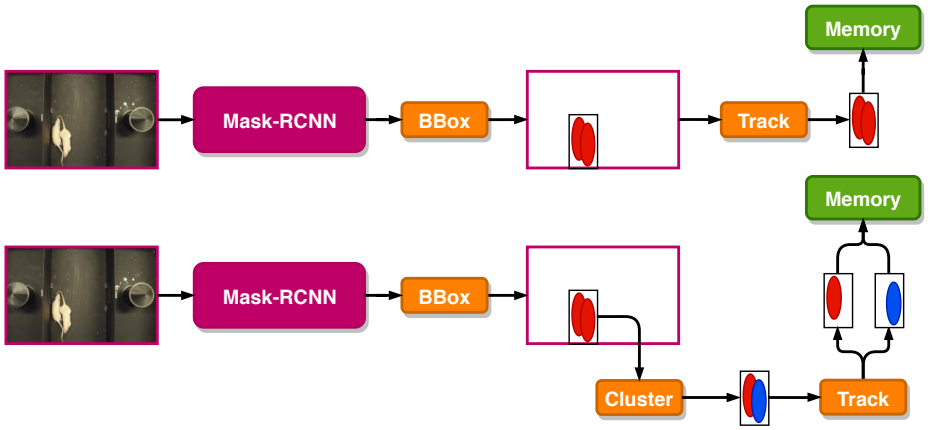

Fig. 5. Fixing MaskTrack R-CNN's weakness by clustering. We solve the coinciding bounding boxes problem by adding a clustering head that makes it possible to separate joint segmentation. The clustering head groups each pixel into $k+1$ categories, where k is the number of objects and $+1$ is for the background. Then we concatenate the resulting clusters to the corresponding feature map as in Mask Scoring R-CNN [7].

We say that two samples are similar, if they belong to the same instance (see Fig. 3):

$$R(x_i, x_j) = \begin{cases} 1, \text{ if } x_i \text{ and } x_j \text{ belong to the same cluster} \\ 0, \text{ otherwise} \end{cases}$$

The probability of two pixels belonging to the same cluster can be determined using the joint probability

$$P\left(y_i = y_j\right) = \sum_{c=1}^{k} P\left(y_i = c, y_j = c\right) = f_\theta\left(x_i\right)^T f_\theta\left(x_j\right).$$

Likewise the probability that they are dissimilar is given by

$$P(y_i \neq y_j) = 1 - P(y_i = y_j).$$

In this way, the loss for all pairs of samples can be calculated by taking the negative logarithm of the CCL [5]:

$$\mathcal{L}_{pair} = -\frac{1}{|T|} \sum_{(i,j)\in T} \log\left[R(x_i, x_j) \cdot P(y_i = y_j) + (1 - R(x_i, x_j)) \cdot P(y_i \neq y_j)\right],$$

where $T = \{(i,j)\}_{\forall i,j \in [1,..,n]}$ is the set of all possible pairs and $|T|$ denotes the cardinality of T.

Following [6], we introduce an additional cluster for the background because of its imbalanced nature. For the background error $\mathcal{L}_{bg}$ we use binary cross-entropy loss as well. Therefore the final loss of the clustering head can be written as

$$\mathcal{L} = \mathcal{L}_{pair} + \mathcal{L}_{bg}.$$

3.3 Error detection

Although we strive for a robust architecture, flawless models are hard, if not impossible to design in such a task. With this remark in mind, we provide two different ways of detecting errors during evaluation. These are beneficial for researchers and other users, due to the ability to mark intervals that need supervision, and those that should be correct. Error may occur:

1. when pixel regions with different cluster IDs neighbor each other closely, or
2. when the number of recognized and tracked objects is less than the actual number of instances, it also indicates an error. This is relevant in the case of MaskTrack R-CNN as discussed in subsection 3.2.

3.4 Architecture overview

We extend the Mask R-CNN [4] architecture (Fig. 6). The input frame is fed to a backbone, with its last feature map serving as input for the recurrent layer, a convolutional gated recurrent unit (ConvGRU) [12, 8]. The hidden representations are then passed to the neck network (FPN) along with the backbone features. The activation of the mask head is changed to softmax and a CCL loss is added on top of its result. We refer to this as *cluster head*. CCL enforces pixels from the same instance to have strongly similar representations, and through backpropagation the recurrent layer can make use of this information. This combination is thus used to extend instance segmentation in time.

4 Implementation

In our experiments, we used a ResNet-50 backbone with an FPN neck, as in the original Mask R-CNN architecture [4]. The RNN's length was set to 10, corresponding to a 10-frame window. This may be reduced during inference to avoid discrepancy between clusters near the beginning and the end of the window being undesirable according to our observations. Training was performed on two RTX TITAN GPUs, with each processing 4 images at a time for 12 epochs. We used Stochastic Gradient Descent and two distinct learning rates: $2 \cdot 10^{-2}$ for auto-annotated samples and $5 \cdot 10^{-3}$ for human-annotated ones.

5 Experiments and results

We evaluate our model on two rat tracking data sets. The first one contains generated samples (refer to section 3.1) similar to the training data. Cluster R-CNN performed exceptionally well on these videos, which is in line with our expectations. The other one measures the robustness of the system on more challenging samples filtered out during the training data preparation.

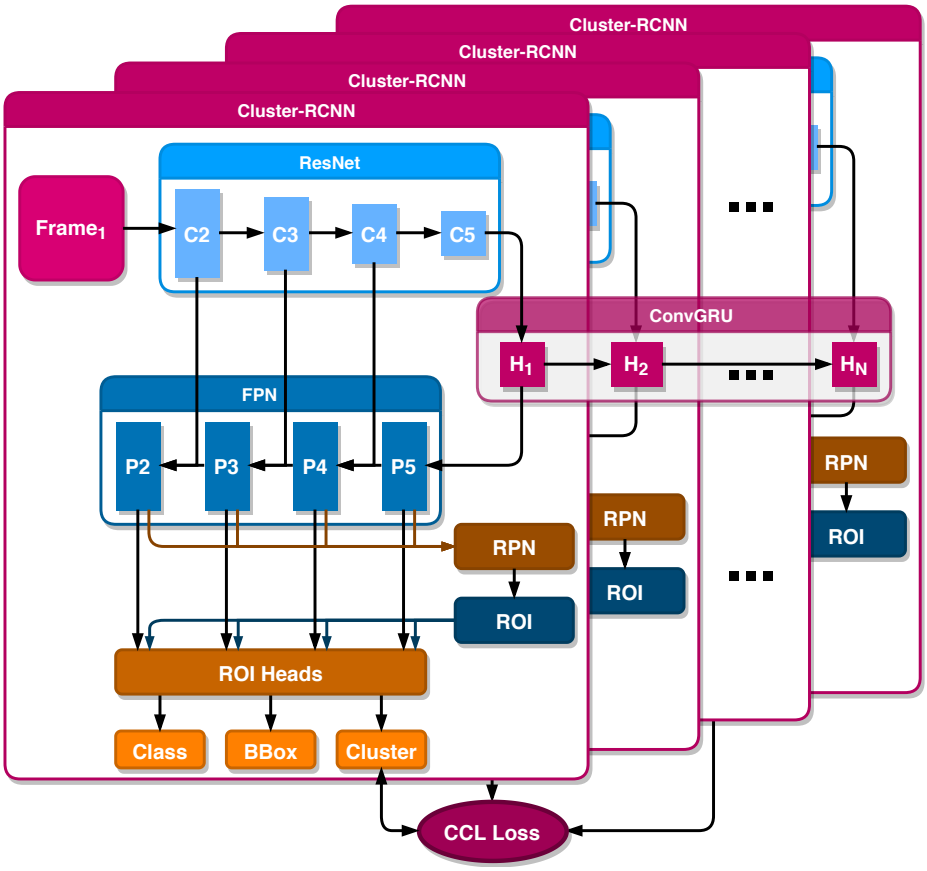

Fig. 6. Cluster R-CNN network architecture. For temporal extension a ConvGRU layer was added to the Mask R-CNN architecture and the mask head was replaced by a cluster head with CCL loss to handle coinciding bounding boxes.

5.1 Segmentation quality

To quantify the segmentation quality of the methods we use the evaluation metrics introduced in the Youtube-VIS [13] benchmark. (See Table 1 for results.) MaskTrack R-CNN has higher average precision (AP) compared to our proposed architecture, which can be contributed to the activation used in their segmentation head network. MaskTrack R-CNN is using sigmoid, while Cluster R-CNN is using softmax, and as it was shown in [4], for segmentation tasks the former one provides better results. On the other hand, our method has higher average recall (AR), which could be justified by the augmentation mechanism (Section 3.1) used to train our model.

Table 1. Qualitative results on the generated validation set using metrics introduced in the Youtube-VIS [13] benchmark. The best results are highlighted in bold. Higher values are better.

Method	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
MaskTrack R-CNN	0.392	0.699	0.414	0.245	0.446
Cluster R-CNN	**0.667**	**0.932**	**0.839**	**0.369**	**0.754**



Fig. 7. From left to right: original video frame, MaskTrack R-CNN and Cluster R-CNN output. This figure presents the most notable case - also described in subsection 3.2 - where our architecture corrects the error of MaskTrack R-CNN. Cluster R-CNN successfully tracks and segments both rats despite their coinciding bounding boxes.

5.2 Robustness

Our goal is to get robust tracking architecture, for this end we aim to detect and correct errors of MaskTrack R-CNN with our proposed method. To quantify robustness of each tracking method we use the total number of track switches. Due to missing ground truth annotations, we conducted the evaluation with human participants in a controlled environment using fourteen 2000 frame-long test videos. Given randomly sampled tracking results from each experiment, the participants had to count the number of track switches (when detected object masks are swapped or lost) on each video.

The results can be seen in Table 2. MaskTrack R-CNN achieved 9 switches, while our proposed method performed with a reasonable 27 switches total without any post-processing steps. These errors are mostly coming from the incorrect re-initialization of the ConvGRU layer, which can be easily filtered out using Hungarian matching algorithm or it could be mitigated by a bigger window size. By combining our method with MaskTrack R-CNN, we can reduce the total number of switches to 5, which is 56% improvement compared to with what we started.

6 Discussion

In the previous section, we focused on tracking accuracy over IoU values. Object masks are also subject to a few noticeable errors in the aforementioned occlusion cases, but the emphasis should always be on flawless tracking. Consider, for example, the mounting action mentioned in the introduction, which can still be

Table 2. Results on the test videos. We measure the robustness of the tracking algorithms using the number of switches. The combined method significantly outperforms the individual methods.

Method	**# of switches**
MaskTrack R-CNN	9
Cluster R-CNN	30
Combination	3

identified with imperfect segmentation, but a tracking mistake will reverse all future interactions (which are generally non-symmetrical).

As stated earlier, precision is vital in this task, and so improving it is a priority over steps to incorporate behaviour recognition. In order to achieve this, one could improve the model or provide more annotated data for harder cases. We consider both options for future work. Tracking multiple body parts of the animals (head, body, tail) would also be valuable in differentiating the rats during mounting scenarios.

7 Conclusions

In this paper, we approached the tracking and segmentation problem from a biological standpoint, operating on a unique data set to process videos of laboratory rats. We examined existing solutions to related tasks, drawing inspiration from several models and designing the final architecture with the problem's nature in mind. The resulting Cluster R-CNN proves the concept with convincing performance in a variety of situations, albeit suffering from the lack of labeled data in more complex interactions. Our model significantly reduces tracking error frequency when coupled with MaskTrack R-CNN, yielding a potent combination that further reduces the amount of manual intervention needed. This work serves as a foundation for future projects, while also providing valuable insight into certain problems and possible solutions in the field.

Acknowledgements.

We thank Bence D. Szalay for his careful help in composing this paper; his work was supported by the Hungarian Government and co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002: EFOP-3.6.3-VEKOP-16-2017-00002, Integrated Program for Training New Generation of Scientists in the Fields of Computer Science). We also thank Árpád Dobolyi and Dávid Keller for providing the database. ÁF and AL were supported by the ELTE Institutional Excellence Program of the National Research, Development and Innovation Office (NKFIH-1157-8/2019-DT) and by the Thematic Excellence Programme (Project no. ED_18-1-2019-0030 titled Application-specific highly reliable IT solutions) of the National Research, Development and Innovation Fund of Hungary, respectively.

References

1. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 941–951 (2019)
2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European conference on computer vision. pp. 850–865. Springer (2016)
3. de Chaumont, F., Ey, E., Torquet, N., Lagache, T., Dallongeville, S., Imbert, A., Legou, T., Le Sourd, A.M., Faure, P., Bourgeron, T., Olivo-Marin, J.C.: Real-time analysis of the behaviour of groups of mice via a depth-sensing camera and machine learning. Nature biomedical engineering **3**(11), 930–942 (2019)
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
5. Hsu, Y.C., Lv, Z., Schlosser, J., Odom, P., Kira, Z.: A probabilistic constrained clustering for transfer learning and image category discovery. arXiv preprint arXiv:1806.11078 (2018)
6. Hsu, Y.C., Xu, Z., Kira, Z., Huang, J.: Learning to cluster for proposal-free instance segmentation. In: 2018 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2018)
7. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring r-cnn. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6409–6418 (2019)
8. Jung, M., Tani, J.: Adaptive detrending for accelerating the training of convolutional recurrent neural networks. In: Proceedings of the 28th Annual Conference of the Japanese Neural Network Society. pp. 48–49 (2018)
9. Porzi, L., Hofinger, M., Ruiz, I., Serrat, J., Bulò, S.R., Kontschieder, P.: Learning multi-object tracking and segmentation from automatic annotations. arXiv preprint arXiv:1912.02096 (2019)
10. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B.: Mots: Multi-object tracking and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7942–7951 (2019)
11. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1328–1338 (2019)
12. 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. In: Advances in neural information processing systems. pp. 802–810 (2015)
13. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5188–5197 (2019)