Freie Universität Berlin

Bachelor's Thesis in Computer Science

Berlin, 13. July 2018

Learning about bee behavior by predicting the future.

Abstract: Most of the research on bee behavioral events revolves around stereotypical behavior such as trophallaxis and bee waggle dances. In order to gain new insights on bee behavior, this thesis introduces an unsupervised future frame prediction framework for video data capturing bee behavior, applying state of the art deep learning architectures, such as ConvLSTM networks and 3D convolutional network in U-Net shape.

Author:	Tom Burgert
Email adress:	tom.burgert@gmx.de
Student-ID:	4883076
Supervisor:	Prof. Dr. Tim Landgraf
2nd Reviewer:	Prof. Dr. Raúl Rojas

Statement of originality

I declare that this thesis is the product of my own original work and has not been submitted in similar form to any university institution for assessment purposes. All external sources used, such as books, websites, articles, etc. have been indicated as such and have been cited in the references section.

Berlin, 13. July 2018

Tom Burgert

Contents

1 Introduction

In the light of great advances in the machine learning discipline of object recognition on static images with deep convolutional networks (Krizhevsky et al., 2012; Razavian et al., 2014; He et al., 2015), there has been an increasing interest in the analysis of video data in recent years (S. Ji et al., 2013; Wu et al., 2015). These developments provide both the theoretical concepts and the technology that enable an analysis of animal behavior by video data from a machine learning perspective. Such analysis can be divided into two major parts: (a) the plain detection and tracking of the animal, comprising the estimation of the body pose over time and (b) the detection and classification of its actions by segmenting its motions into meaningful intervals that map onto goals or purposes (Eyjolfsdottir et al., 2016). Eyjolfsdottir et al. used a recurrent neural network to derive information about (b) from (a) that predicts the future motion of fruit flies while mapping inner states and representing high level behavioral phenomena such as fly gender. The model proves that motion prediction is a good auxiliary task for action classification, especially when training labels are scarce.

As the neural system of honey bees comprises about four times more neurons than those of fruit flies (Menzel and Giurfa, 2001; Shenai, 2003), they are capable of more sophisticated behavior (Theobald, 2014) and therefore an appealing object of research that provides an opportunity to build upon Eyjolfsdottir et al.'s scientific discoveries. In contrast to making use of an extracted trajectory, which is already a simple abstraction of behavior, inputting raw video data depicts a real world scenario with virtually no loss of information. This thesis refines the approach of Eyjolfsdottir et al. to a future frame prediction framework for video data capturing bee behavior in order to enable the detection of behavioral events such as waggle dances and trophallaxis and furthermore, try to gain new insights on behavioral patterns of bees that go beyond those already well investigated behavioral events.

2 Related Work

A honeybee colony is a complex system that unfolds remarkable dynamics (Seeley, 1995; Bonabeau et al., 1997). The colony's behavior comprises thousands of individuals interacting mostly on local cues forming a complex society (Wario et al., 2015). When trying to understand behavioral patterns of bees, both as individuals and jointly as a collective with the help of machine learning techniques, the task needs to be looked at from two perspectives, the biological aspect of bee research on one hand and the perspective of the rapidly advancing field of self-learning systems and artificial intelligence on the other.

When it comes to bee research, the second half of the 20th century has been impaired by the issue of data acquisition. Previously, in the case of the probably most investigated behavioral event, the bee waggle dance (Von Frisch, 1967; Seeley, 1995; Grueter and Farina, 2009), scientists were tied to either the real-time collection of data, or else its manual collection from recorded video material (e.g., Visscher and Seeley, 1982; Richter and Waddington, 1993; Beekman et al., 2004). Only recently there has been a shift towards the automation of bee tracking, and thus, the automation of waggle dance detection (De Marco et al., 2008; Landgraf et al., 2011).

The proposed *BeesBook system* (Wario et al., 2015) enables the continuous long-term tracking of honeybees inside the bee hive. It automatically records and stores high-resolution images of tracks of uniquely identifiable bees. Its features include software for recognizing and identifying uniquely marked bees, as well as software for detecting waggle dances, dance-following behavior and trophallaxis (Wario et al., 2015).

The extensive database acquired by the *BeesBook system* serves as a source of unique video data depicting bee behavior for this thesis. Even though there has been a lot of research on bee behavior already, many questions about the bee waggle dance as well as other interaction patterns still remain unanswered. This thesis attempts to gain new insights on bee behavior with a future frame prediction framework, following the approach of Eyjolfsdottir et al. (2016) who used an architecture adapted from recurrent neural networks for modeling the behavior of fruit flies, simultaneously classifying actions and predicting future motions.

The artificial intelligence research's first approaches to video predictive learning have mostly focused on synthetically generated video data, such as bouncing balls or moving numbers with highly predictable future frames (Oh et al., 2015). However, the prediction of complex real-world video sequences still remains an open challenge, although progress has been made in recent years (Mathieu et al., 2015; Finn et al., 2016; Kalchbrenner et al., 2016).

Tran et al. (2017) showed the effectiveness of using three-dimensional convolutional networks to learn spatiotemporal features. By extending the formerly often used 3D convolutional networks for video analysis to a video prediction framework, Liu et al. (2017) tried to tackle the anomaly detection problem, making use of a slight modification of the U-Net architecture (Ronneberger et al., 2015).

Inspired by the success of convolutional networks in feature extraction of images (Krizhevsky et al., 2012; He et al., 2015; Razavian et al., 2014) and the recent breakthroughs in time series prediction (Sundermeyer et al., 2012: Graves, 2013; Wen et al., 2015) based on the concept of LSTM networks (Hochreiter and Schmidhuber, 1997) it is quite intuitive to think about a mixture of both when it comes to video frame prediction, namely the sequential prediction of images.

Oh et al. (2015), as one of the first, suggested a combination of RNNs with CNNs for spatiotemporal future frame prediction in Atari Games. Elaborating this idea, Shi et al. (2015) then proposed the ConvLSTM network extending the conventional LSTM network by plugging in a convolutional operation into the recurrent connections. Instead of only forwarding one-dimensional data, the ConvLSTM architecture enables three-dimensional input data. In the field of precipitation nowcasting, with the goal of the prediction of future rainfall intensity, the ConvLSTM architecture has been successfully applied to one as well as multichannel radar data (Shi et al., 2015; Kim et al., 2017). By introducing Causal LSTM cells and Gradient Highways the PredRNN++ Wang et al. (2018) further refines the basic LSTM cell to enable the prediction of consecutive future frames for the moving MNIST Dataset as well as the KTH Action Dataset containing 6 types of human action. In 2016, Lotter et al. proposed the PredNet inspired by the concept of "predictive coding" from the neuroscience literature that also makes use of a ConvLSTM cell. Its architecture allows the prediction of future frames in video sequences, both of synthetically generated video data and real-world video data stemming from car-mounted camera videos from the KITTI data set.

Following the recent advances of ConvLSTM networks and 3D convolutional networks using the U-Net architecture in predicting the future in video data, this thesis will make use of similar architectures in order to investigate it's abilities for bee behavior prediction and action classification.

However, the limitations of the success of previously applied future frame prediction models lay in the type of data that was used to build the models. Generally, synthetically generated video data, e.g the moving MNIST dataset and video data from video games tend to be easier to predict, since both have underlying, pre-programmed moving patterns, thus an accessible ground truth function for neural networks to learn. Even weather radar data, assuming nature laws follow concrete rules and equations of physics, have underlying structures and patterns that might be less complex than bee interaction in the collective behavior of thousands of individual bees. Solely the KITTI dataset and the KTH action dataset are depicting a real-world environment of movement. However, the challenging task of future frame prediction on bee behavior consists of learning an animal's basic behavioral patterns without any supervision.

3 Implementation

3.1 Data

The database of the *BeesBook system* contains the annual tracking data of an entire bee colony. At the beginning of every season the bees get tagged manually by a decodable marker making it possible to track its position in the beehive at almost any time. To make this possible, the thin, cuboid-shaped beehive gets filmed by four individual cameras, two for each face. The pre-built database table the thesis makes use of contains information about successfully tracked paths of bees. A single track consists of the corresponding frame IDs, the camera ID and the positional data of the tracked bee, as well as the orientation value, which gives information about the bee's current body rotation angle. A single track usually lasts from a few seconds up to some minutes, depending on the ability to track down the bee in the beehive.

To access the grayscale video database of the whole beehive, the pipeline provided in the *BeesBook backend* is used. The class Frameplotter is able to return a pre-cropped image placing the tag of the bee in the center of the image when providing the cropping coordinates as well as the frame ID. With the Frameplotter class and the coordinates from the tracking database table it is possible to extract the corresponding frames per track.

When preparing the dataset, several problems arise. Due to the large surface of the beehive, which is covered by four individual cameras, the exposure of different areas in different frames varies a lot. Therefore all extracted images are normalized in regards to the contrast and the lightness by applying the histogram equalization algorithm CLAHE (Pizer et al., 1987).

To avoid an overfitting on the markers during the training procedure of the neural network (e.g. to prevent the machine from learning the markers "by heart"), a mask is applied to all extracted images replacing the tag of the bee by a gray circle. The value for the gray circle is calculated by the mean pixel value of some hundred sampled images.

Another issue that could impede the success of the machine learning task is that the tracks contained in the tracking database table can have time gaps of one or several frames. To tackle this issue, a linear interpolation is applied for the missing values of the coordinates and rotation angles over all time gaps t smaller equal 2, filtering out tracks with more than a maximum of two consecutively missing frames. By making use of another database table all consecutive time steps with the corresponding frame ID were extracted, in order to interpolate the tracking data for the missing frames. Since an exact interpolation cannot be guaranteed, the interpolation limit is chosen to be t smaller equal 2 avoiding too much additional noise on the dataset.

Since Convolutional Networks are not rotation invariant, it is inevitable to normalize the rotation angle of the bees in every frame in beforehand. By making use of the orientation value, all pre-copped images get rotated such that the bee is pointing towards the north before cropping the frame a second and final time. Unfortunately, the orientation values are not as accurate as the coordinates are. To lower the impact of outliers in the orientation values, a gaussian filter is applied to each track to smooth its orientation values and reduce possible noise.

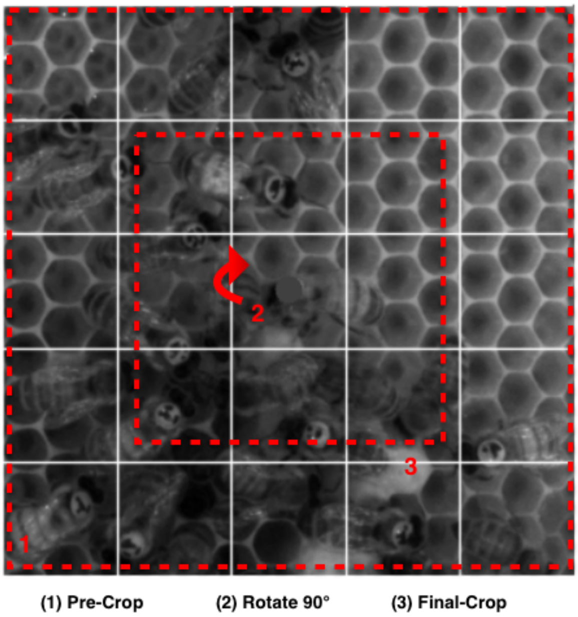

Fig. 1: Example for the cropping process for one normalized frame in a data sample (track) of the data set.

In the final dataset, each data sample represents a three dimensional datapoint. It consists of a time series of n consecutive frames of a certain frame size of width w and height h [n=16, w,h=(200,200)]. The corresponding label has the same shape. In a more simple model, each next time step in a data sample could be interpreted as the label for the current time step and an explicit label for each time step in the data sample is obsolete. Therefore, in order to force the network on learning the actual movement (and behavior) of the bee, an additional label is generated to capture the bee's movement between the current time step and the next time step. Let t be the label of a data sample x that depicts a time series of length n. To generate the i-th position of the label t, the same cropping coordinates and rotation angle that were used to generate xi will be applied to the next frame xi+1. This lets the bee move inside the same image and does not require any further normalization of the images in the label. That way, the current time step in the label ti and the next time step in the data sample xi+1 represent the same bee in the same video frame, but with different cropping coordinates and a different rotation angle.

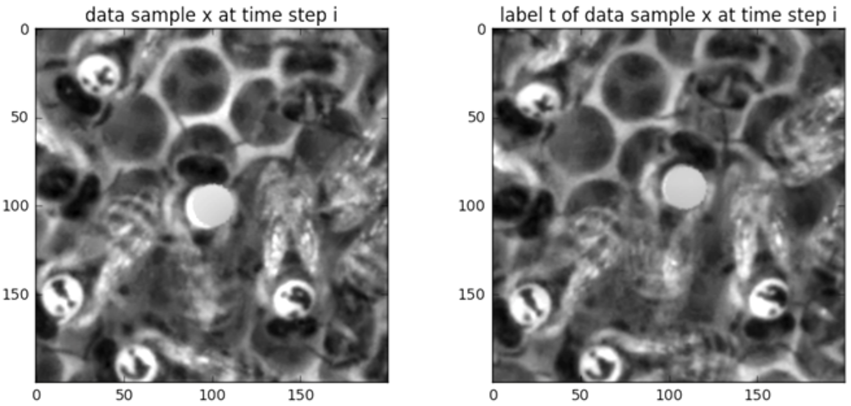
data sample x at time step i
label t of data sample x at time step i
0
50
100
150

Fig. 2: Example of one timestep of a data sample and its explicit label. It can be seen that the bee moves from the middle to the upper right in one time step.

To assure that the dataset is coherent, it is necessary to filter out tracks that got tracked close to the borders of the cameras, otherwise the *BeesBook backend* is not able to crop the frames such that the bee is centered. Additionally, all tracks with a very heavy movement, such that one of the time steps in the label will not be able to capture the whole body of the bee, because the bee moves out of the cropping area, need to be filtered out as well. To prevent filtering out important events such as dance events, tracks of 26 already labeled events were checked with the satisfying result of a filtration rate of 0.0%. None of the important events seem to take place in the outer regions of the beehive.

Eventually the generated data samples are saved in a HDF5 file format divided into three different types of datasets according to their individual variance of the rotation angle over the time dimension. This is done to possibly pre-train the networks with the highest variance dataset to focus on actual movement only, rather than allowing the networks to cut short by just reproducing the very last image of the track, when training on tracks with little movement.

3.2 Loss Functions

The future frame prediction task requires the estimation of many individual pixel values using regression techniques. In the paradigm of regression the most applied loss functions are either the L1 (MAE) loss or L2 (MSE) loss. For the dataset of bee behavior it is necessary to crop the images such that possible interactions between the bees are still captured while also putting a higher focus on the behavior of the bee. To meet those requirements a weighted mask is introduced, favoring the center of the bees body and reducing the impact of the outer regions of the image. In order to weight the loss function it gets multiplied with a gaussian kernel, heavily weighting the bees center in an oval shape giving more impact to events happening in the close surrounding of the bee.

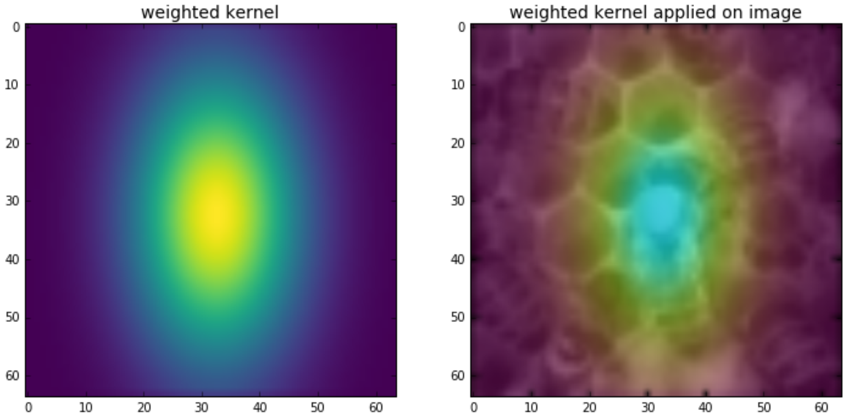

Fig. 3: The left picture shows the weighted mask that is applied to calculate the weighted MSE loss. Yellow means multplication by 1, dark blue means multication by 0. In the right picture the importance of certrain regions of the generated image for the calculation of the loss can be seen.

In the experiments a weighted L2 loss function and a weighted SSIM loss function are used. In contrast to the MSE which estimates the absolute error the structural similarity index measure (Zhou Wang et al., 2004) is a perception-based model putting a higher focus on the inter-dependencies of pixel that are spatially close.

3.3 Models

In the experiments two mayor architectures are used: 3D convolutional networks and the above described ConvLSTM networks.

(1) 3D convolutional networks

3D-ConvNet multi-temporal output: A simple 3D convolutional network architecture composed of 5 layers with ReLUs (Glorot et al., 2011), kernel size 3 and padding 1 for spatial dimension. In the first and last layer the temporal dimension with kernel size 3 and padding 1 is allowed to see information from neighboring time steps, while the other convolutional layers will not gain any additional temporal information due to kernel size 1. Since the network does not reduce the temporal dimension to one, only the last activation of the temporal axis is used as the prediction in order to match the two-dimensional spatial target (the last time step of the unlabeled data sample) for computing the loss. The filter size used is 32.

3D-ConvNet single-temporal output: A more complex multilayered 3D convolutional network composed of 12 layers using ReLUs divided into pairs of two. Each pair consists of one layer doubling the filter size while halving the temporal dimension by stride 2, eventually mapping the temporal dimension to one, and a second layer preserving the shape of the input in the output. Mapping the temporal dimension to one makes it easier to compute the loss between the network's output and the target. The network's initial filter size is 16, the kernel size is 3 and 3D batch norm is used.

3D-ConvNet using U-Net architecture for single-temporal output: A 3D convolutional network that makes use of the U-Net architecture by extending it to the temporal dimension. The downsampling part remains the same as in the original paper, stacking 5 layers of downsampling until the temporal dimension has converged to one. In the upsampling part only the spatial dimensions get upscaled while the temporal dimension remains the same. When concatenating the deconvolutional activation with the activation of the layer with the same filter size in the downsampling part, a maxpooling over the temporal dimension is applied such that the temporal dimension gets mapped to one. The final activation of the network is the two-dimensional future frame prediction. The initial filter size used is 32.

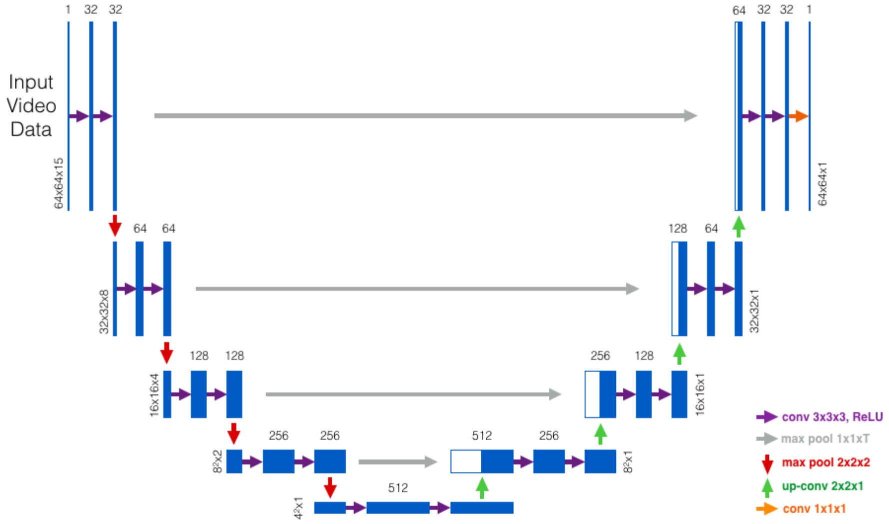

Fig. 4: Architecture of the 3D-ConvNet in U-Net shape with filter size 32. Purple: The convolutional layers use padding 1 to preserve the shape of the input. Gray: The parameter T equals the temporal dimension of the corresponding level, such that the pooling layer maps it to one and enables the concatenation with the upscaled activations, e.g on level 2 the parameter T is 8, on level 3 it is 4. Figure adapted from U-Net paper (Ronneberger et al., 2015).

(2) ConvLSTM networks

All ConvLSTM models make use of the ConvLSTM cell proposed by Shi et al. (2015). The convolutional layers used inside the cells have kernel size 3, while the cells have 64 channels in the hidden state. To map the multiple output channels to one, two 3D convolutional layers with ReLUs and kernel size 3 and padding 1 for the spatial dimension as well as kernel size 1 and padding 0 for the temporal dimension are applied to the ConvLSTM's output. The training loss is computed over the outputs of all time steps, setting as a target either the next time step or the generated labels of the data samples. All ConvLSTM models share most of the above mentioned hyper-parameter, with only small changes in the basic architecture.

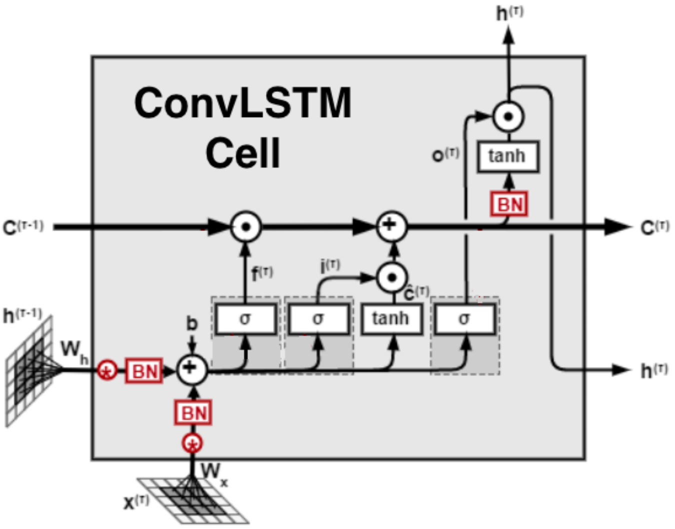

Fig. 5: The ConvLSTM cell. Figure adapted from Sautermeister (2016).

ConvLSTM simple: A simple version of the ConvLSTM network with no additional parameter.

ConvLSTM with Resblock: The convolutional layers to map the output channels of the LSTM network to one are replaced by residual blocks (He et al., 2015) with the same kernel size and padding.

ConvLSTM 2-layer: Two layers of ConvLSTM layers are stacked, in order to gain more complexity.

ConvLSTM PreResblock: A simple convolutional network with Resblocks with hidden layer filter size 32 is concatenated with the ConvLSTM model.

4 Evaluation

In order to evaluate the quality of learning, and to compare different models, a naive loss is introduced as an evaluation metric. For the test set the naive loss is calculated by interpreting the last image of the video sequence as the prediction. Only reproducing the last image might be a shortcut solution, which the model could choose to reduce the training loss. In the experiments, the goal is to improve the naive loss as much as possible. When training different models with the weighted SSIM loss, it appears impossible to improve the naive loss by more than 10%, indicating an optimization with the weighted SSIM to be ineffective. Thus, the following experiments only focus on a model fitting that uses the weighted MSE loss, where improvement rates could hit around 30%.

In the beginning, the experiment was divided into two parts, one of which making use of the explicit label and one only implicitly interpreting the next frame as the label for the current one. All models were trained with a joined dataset containing all variance classes, preserving an equal balance between tracks with little or no movement and tracks with stronger movement. In the real distribution of the variance classes the tracks with little or no movement outweigh by 70%. By adding more stronger moving tracks, it is expected to prevent the model from learning the naive solution. Before the training, all images were downscaled from 200x200 pixel to a more adequate size of 64x64 to meet the computational capacity of the computers. During the first training run the models were trained for 50 epochs long with in-between validations between two epochs, leading to an early break if 10 continuous validations could not improve the best validation score in order to prevent the model from overfitting.

Tab. 1: Results of first training run on the validation set. Column “Improvement” in respect to the naive loss. Mt. o. = multi-temporal output. St.o. := single-temporal output.

	Min MSE	Improvement	SSIM	PSNR
without explicit label				
3D-ConvNet mt. o. simple	0.003296	33,12%	0.4570	**2466.7**
3D-ConvNet mt. o. complex	0.003285	33,35%	0.4571	2174.0
3D-ConvNet st. output	0.003301	33,02%	0.4491	1878.9
3D-ConvNet st. o. U-Net	**0.003218**	**34,70%**	**0.4694**	2379.1
ConvLSTM simple	0.003256	33,93%	**0.4624**	2469.7
ConvLSTM PreResblock	0.003247	34,11%	0.4627	2601.9
ConvLSTM 2-layer	0.003267	33,70%	0.4614	2619.0
ConvLSTM with Resblock	**0.003238**	**34,29%**	0.4594	**2657.1**
with explicit label				
ConvLSTM simple	0.002793	29,06%	0.5725	2822.1
ConvLSTM PreResblock	0.002765	29,76%	0.5780	2896.2
ConvLSTM 2-layer	0.002763	29,81%	0.5746	2882.8
ConvLSTM with Resblock	**0.002743**	**30,33%**	**0.5814**	**2941.7**

The results for the validation set in table 1 show that the weighted MSE loss is a good choice for fitting models to predict the future frames in video data. The models with the highest improvement of the naive loss for the validation set also have the best or one of the best scores for the structural similarity index measure (Zhou Wang et al., 2004) as well as for the peak signal to noise ratio (Huynh-Thu and Ghanbari, 2012). When comparing the SSIM and PSNR of the labeled and the unlabeled learning tasks, it is apparent that by introducing an explicit label the focus on the actual movement of the bee is reinforced. In average, both measures improved about 20%, suggesting that the predicted images and the real targets are more similar, thus, the models have an easier job predicting the surroundings, such as combs or other bees, leaving more capacity for the desired task of predicting the actual behavior of the bee.

Interestingly, the best scores of the 3D Convolutional architectures and the ConvLSTM architectures only differ slightly. It can be assumed that both styles of architecture find similar patterns in the training data and are able to learn the same structures.

While all 3D Convolutional models stopped the training earlier due to the drop-out validation, some of the ConvLSTM models could continue learning. During a second run the very best ConvLSTM models are examined. The two most promising architectures as well as a mixture of both, a 2Layer Resblock ConvLSTM network, are trained for 150 epochs with the same validation drop-out.

Tab. 2: Results of the second training run with the best performing models on the validation set.

	Min MSE	Improvement	SSIM	PSNR
without explicit label				
ConvLSTM 2-layer	0.003248	34,10%	0.4678	2599.7
ConvLSTM with Resblock	0.003235	34,35%	0.4652	2528.8
ConvLSTM 2-layer Resblock	**0.003216**	**34,75%**	**0.4684**	**2776.0**
with explicit label				
ConvLSTM 2-layer	0.002737	30,47%	0.5844	2939.6
ConvLSTM with Resblock	0.002722	30,85%	0.5836	2985.2
ConvLSTM 2-layer Resblock	**0.002714**	**31,07%**	**0.5871**	**3007.4**

The results for the validation set of the second run prove the current state of sciences, which can be adapted and also shown in the future prediction framework, suggesting that the Resblock architecture for convolutional networks achieves the best results.

When comparing the predicted images of the best networks, namely the 3D-Convolutional U-Net for the unlabeled data set and the ConvLSTM 2layer Resblock for the labeled data set, it is apparent that the networks are learning more than only reproducing the last image of the track. Still, the results are not completely satisfying. It can be clearly seen that with more movement the bee itself, but even more so its surrounding, become very blurry, whereas with little movement, the predicted image is still relatively sharp and examinable by the human eye. The blurriness suggests either the inherent uncertainty of the network when predicting the next frame or else arises due to the impossibility of learning the required task. This could have several reasons, such as the applied method does not work properly, the desired task is impossible to learn or data is just not accurate enough. The latter, for example, could correspond to the very small frame rate of the video camera. A frame rate of 3fps might not be sufficient for the models to make sense of the bee's moving patterns. Another aspect possibly adding too much noise on the training data could be inaccurate values in the database, in particular those of the orientation values. Even though they have already been treated in the data sampling process, the performance of the normalization might be still not good enough.

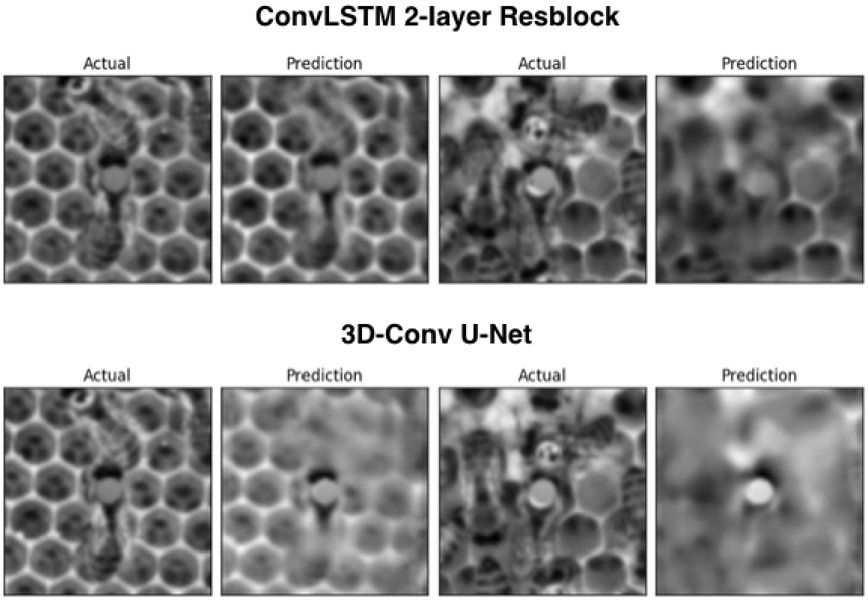

Fig. 6: Example frames for the original target frame and its prediction for the best two models.

Since the prediction of a high resolution image did not work flawlessly, but the models apparently learned something much better than the naive loss, an analysis of the hidden states of the ConvLSTM network and the bottom of the 3D U-Net network can suggest, whether the model has learned something useful in regards to action classification.

To better examine what happens inside the hidden states, correctly labeled tracks that either capture the events of trophallaxis or waggle dance were added to the data set. The extended data set got forwarded through the pre-trained model in order to extract the activations of the hidden states. To reduce the activation vector of shape 64x64x64 the mean values of the neurons of the second and third dimension representing the pixel values of width and height got calculated, only leaving a one-dimensional vector of size 64. Then, the averaged activations were clustered by firstly reducing the space to 50 components using PCA (Wold et al., 1987) and further applying the t-SNE algorithm (van der Maaten and Hinton, 2008) to reduce the dimensionality to two. When plotting the dimensionality reduction, there are two apparent clusters that mostly consist of tracks with heavy movement. Interestingly, all labeled tracks that contain a trophallaxis event are only tightly clustered into two regions. Examining the unlabeled tracks that were clustered into that same region, none of them actually depicts a trophallaxis event, but all appear to be depicting a trophallaxis-like movement with either another bee vertically facing the bee or heavy movement of the sensors and the body of the bee while staying around the same comb.

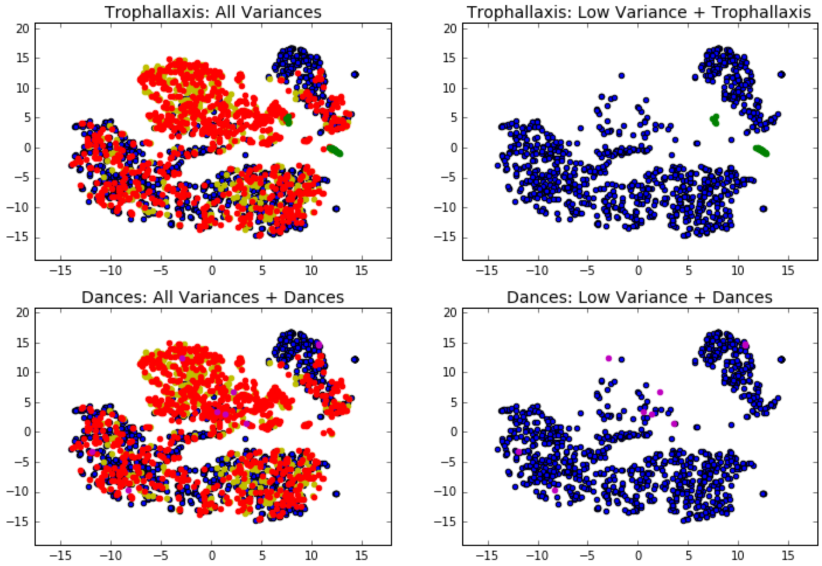

Fig. 7: Visualization of the hidden states of the best ConvLSTM network after applying t-SNE. Blue: tracks with little or no movement at all (low variance). Yellow: tracks with medium movement (medium variance). Red: tracks with strong movement (high variance). Green: tracks labeled as trophallaxis events. Purple: tracks labeled as dance events.

A similar result can be obtained when examining the bottom anchor of the U-Net architecture in a 8-filter U-Net. While the temporal dimension is already mapped to one, an additional pooling layer is added to further reduce the 4x4 activation block of width and height to one, only leaving 256 filter of a 1x1x1 vector. After repeating the same cluster procedure, all trophallaxis events are clustered tightly into two regions as well. The little difference to the ConvLSTM hidden states is that a bigger amount of individual clusters on the whole data set is apparent, while there are still clusters that consist of tracks with strong movement only, like in the hidden states of the ConvLSTM network.

Applying the same procedure to the tracks depicting dance events shows less unambiguous results. The dance tracks are broadly spread over many clusters. Though, in contrast to the trophallaxis event, a 16 frame track might not be enough to capture the whole event, making it possible that different parts of the waggle dance get captured in different tracks, leading to the broad spreading over the whole space.

5 Discussion and Future Work

The experiment shows that the task of predicting future frames in a realistic environment remains an open challenge. Especially generating a high resolution image of a video sequence seems to challenge even state-of-the-art neural network architectures. To further investigate how reasonable the discoveries about the inner states' representations are, it is necessary to acquire more labeled data about trophallaxis events and waggle dance events. By using the additionally acquired labeled data, the hidden states of the models could serve as a simple classifier to either predict or classify a certain behavioral event. Further investigations could also include examining in detail the clusters of the hidden states for similarities in the video sequences of the tracks. It could be possible that different, so far not-yet-well investigated behavioral events could appear to be clustered in different areas of the space. To optimize the generated future frames, the quality of the data set could be improved by implementing cameras with a higher frame rate or improving the *BeesBook backend* in order to make the orientation values more accurate. Finally, utilizing a GAN loss on top of the weighted L2 loss could improve the quality of the outputted images.

References

M. Beekman, D. J. T. Sumpter, N. Seraphides, and F. L. W. Ratnieks. Comparing foraging behaviour of small and large honey-bee colonies by decoding waggle dances made by foragers. *Functional Ecology*, 18(6):829–835, 2004. URL http://onlinelibrary.wiley.com/doi/10.1111/j.0269-8463.2004.00924.x/full.

Eric Bonabeau, Guy Theraulaz, Jean-Louls Deneubourg, Serge Aron, and Scott Camazine. Self-organization in social insects. *Trends in Ecology & Evolution*, 12(5):188–193, May 1997. ISSN 0169-5347. doi: 10.1016/S0169-5347(97)01048-3. URL http://www.sciencedirect.com/science/article/pii/S0169534797010483.

R. J. De Marco, J. M. Gurevitz, and R. Menzel. Variability in the encoding of spatial information by dancing bees. *Journal of Experimental Biology*, 211(10):1635–1644, May 2008. ISSN 0022-0949, 1477-9145. doi: 10.1242/jeb.013425. URL http://jeb.biologists.org/cgi/doi/10.1242/jeb.013425.

Eyrun Eyjolfsdottir, Kristin Branson, Yisong Yue, and Pietro Perona. Learning recurrent representations for hierarchical behavior modeling. *CoRR*, abs/1611.00094, 2016. URL http://arxiv.org/abs/1611.00094.

Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A Connection between Generative Adversarial Networks, Inverse Reinforcement Learning, and Energy-Based Models. *arXiv:1611.03852 [cs]*, November 2016. URL http://arxiv.org/abs/1611.03852. arXiv: 1611.03852.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. In Geoffrey Gordon, David Dunson, and Miroslav DudÃk, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, April 2011. PMLR. URL http://proceedings.mlr.press/v15/glorot11a.html.

Alex Graves. Generating Sequences With Recurrent Neural Networks. *arXiv:1308.0850 [cs]*, August 2013. URL http://arxiv.org/abs/1308.0850. arXiv: 1308.0850.

Christoph Grueter and Walter M. Farina. The honeybee waggle dance: can we follow the steps? *Trends in Ecology & Evolution*, 24(5):242–247, May 2009. ISSN 0169-5347. doi: 10.1016/j.tree.2008.12.007.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. URL http://arxiv.org/abs/1512.03385. arXiv: 1512.03385.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 0899-7667.

Quan Huynh-Thu and Mohammed Ghanbari. The accuracy of PSNR in predicting video quality for different video scenes and frame rates. *Telecommunication Systems*, 49(1):35–48, January 2012. ISSN 1572-9451. doi: 10.1007/s11235-010-9351-x. URL https://doi.org/10.1007/s11235-010-9351-x.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural Machine Translation in Linear Time. *arXiv:1610.10099 [cs]*, October 2016. URL http://arxiv.org/abs/1610.10099. arXiv: 1610.10099.

Seongchan Kim, Seungkyun Hong, Minsu Joh, and Sa-kwang Song. DeepRain: ConvLSTM Network for Precipitation Prediction using Multichannel Radar Data. *arXiv:1711.02316 [cs]*, November 2017. URL http://arxiv.org/abs/1711.02316. arXiv: 1711.02316.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=2999134.2999257.

Tim Landgraf, RaÃºl Rojas, Hai Nguyen, Fabian Kriegel, and Katja Stettin. Analysis of the waggle dance motion of honeybees for the design of a biomimetic honeybee robot. *PloS one*, 6(8):e21354, 2011.

Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future Frame Prediction for Anomaly Detection – A New Baseline. *arXiv:1712.09867 [cs]*, December 2017. URL http://arxiv.org/abs/1712.09867. arXiv: 1712.09867.

William Lotter, Gabriel Kreiman, and David Cox. Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning. *arXiv:1605.08104 [cs, q-bio]*, May 2016. URL http://arxiv.org/abs/1605.08104. arXiv: 1605.08104.

Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv:1511.05440 [cs, stat]*, November 2015. URL http://arxiv.org/abs/1511.05440. arXiv: 1511.05440.

Randolf Menzel and Martin Giurfa. Cognitive architecture of a mini-brain: the honeybee. *Trends in cognitive sciences*, 5 2:62–71, 2001.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard Lewis, and Satinder Singh. Action-Conditional Video Prediction using Deep Networks in Atari Games. *arXiv:1507.08750 [cs]*, July 2015. URL http://arxiv.org/abs/1507.08750. arXiv: 1507.08750.

Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B. Zimmerman, and

Karel Zuiderveld. Adaptive Histogram Equalization and its Variations. *Computer Vision, Graphics, and Image Processing*, 39(3):355–368, 1987. ISSN 0734-189X. doi: 10.1016/S0734-189X(87)80186-X.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN Features off-the-shelf: an Astounding Baseline for Recognition. *arXiv:1403.6382 [cs]*, March 2014. URL `http://arxiv.org/abs/1403.6382`. arXiv: 1403.6382.

Monica Raveret Richter and Keith D. Waddington. Past foraging experience influences honey bee dance behaviour. *Animal Behaviour*, 46(1): 123–128, July 1993. ISSN 0003-3472. doi: 10.1006/anbe.1993.1167. URL `http://www.sciencedirect.com/science/article/pii/S000334728371167X`.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv:1505.04597 [cs]*, May 2015. URL `http://arxiv.org/abs/1505.04597`. arXiv: 1505.04597.

S. Ji, W. Xu, M. Yang, and K. Yu. 3d Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, January 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.59.

Sautermeister. Adapted figure convlstm cell. `http://bsautermeister.de/research/frame-prediction/`, 2016. Accessed on 26.06.2018.

Thomas D. Seeley. *The wisdom of the hive: the social physiology of honey bee colonies.* Harvard University Press, Cambridge, Mass, 1995. ISBN 978-0-674-95376-5.

Jayant P. Shenai. The newborn brain: Neuroscience and clinical applications. *Journal of Perinatology*, 23:260–261, 2003.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *arXiv:1506.04214 [cs]*, June 2015. URL `http://arxiv.org/abs/1506.04214`. arXiv: 1506.04214.

Martin Sundermeyer, Ralf Schlueter, and Hermann Ney. *LSTM Neural Networks for Language Modeling.* 2012.

Jamie Theobald. Insect neurobiology: How small brains perform complex tasks. *Current Biology*, 24(11):R528 – R529, 2014. ISSN 0960-9822. doi: https://doi.org/10.1016/j.cub.2014.04.015. URL `http://www.sciencedirect.com/science/article/pii/S0960982214004539`.

Thanh Thi Viet Tran, Long Ke Phan, and Jean-Dominique Durand. Diversity and distribution of cryptic species within the Mugil cephalus species complex in Vietnam. *Mitochondrial DNA Part A*, 28(4):493–501, 2017. doi: 10.3109/24701394.2016.1143467. URL https://doi.org/10.3109/24701394.2016.1143467.

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL http://www.jmlr.org/papers/v9/vandermaaten08a.html.

P. Kirk Visscher and Thomas D. Seeley. Foraging Strategy of Honeybee Colonies in a Temperate Deciduous Forest. *Ecology*, 63(6):1790–1801, 1982. ISSN 0012-9658. doi: 10.2307/1940121. URL http://www.jstor.org/stable/1940121.

Karl Von Frisch. The dance language and orientation of bees. 1967.

Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and Philip S. Yu. PredRNN++: Towards A Resolution of the Deep-in-Time Dilemma in Spatiotemporal Predictive Learning. *arXiv:1804.06300 [cs, stat]*, April 2018. URL http://arxiv.org/abs/1804.06300. arXiv: 1804.06300.

Fernando Wario, Benjamin Wild, Margaret Jane Couvillon, Raul Rojas, and Tim Landgraf. Automatic methods for long-term tracking and the detection and decoding of communication dances in honeybees. *Frontiers in Ecology and Evolution*, 3:103, 2015.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. *arXiv:1508.01745 [cs]*, August 2015. URL http://arxiv.org/abs/1508.01745. arXiv: 1508.01745.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists*, 2(1):37–52, August 1987. ISSN 0169-7439. doi: 10.1016/0169-7439(87)80084-9. URL http://www.sciencedirect.com/science/article/pii/0169743987800849.

Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pages 461–470, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3459-4. doi: 10.1145/2733373.2806222. URL http://doi.acm.org/10.1145/2733373.2806222.

Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861.

A Appendix

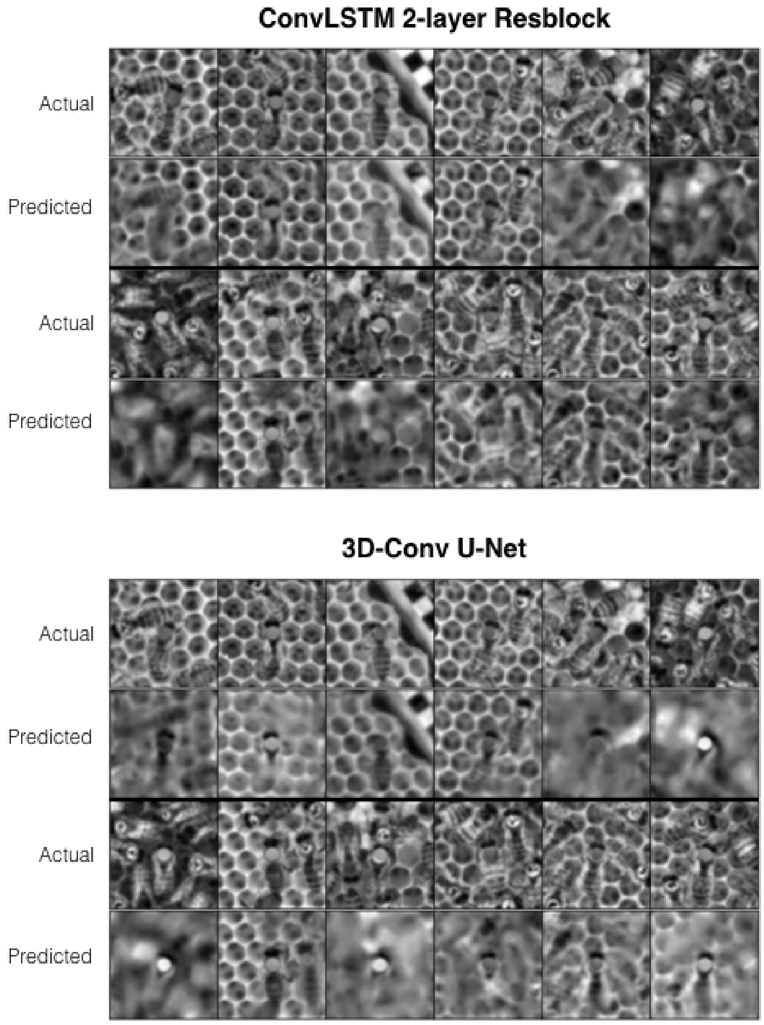

Fig. 8: More example frames for the original target frame and its prediction.

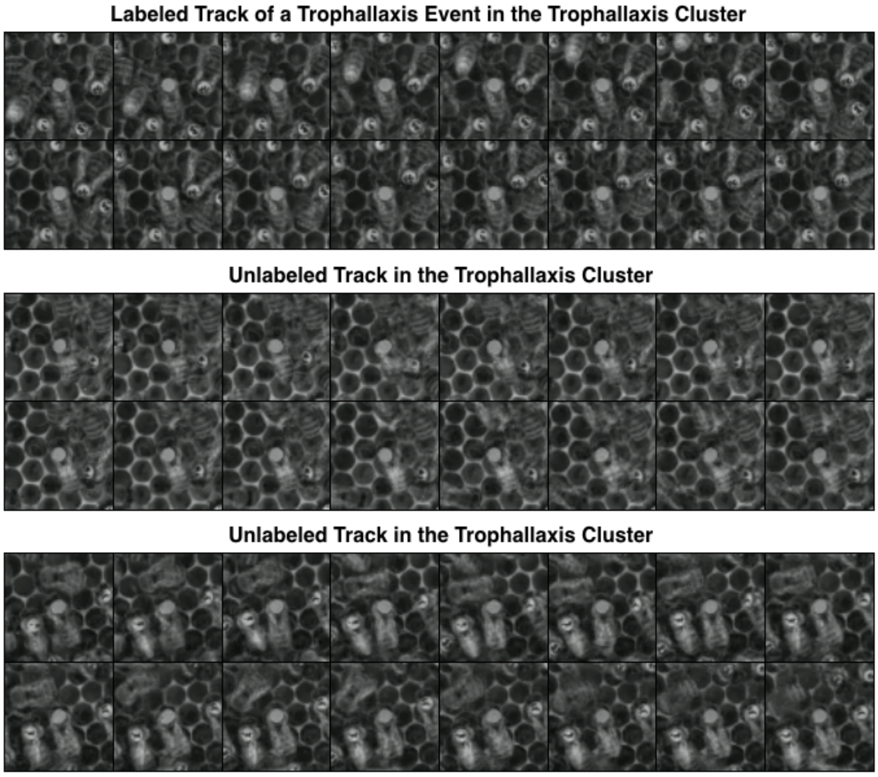
Labeled Track of a Trophallaxis Event in the Trophallaxis Cluster
Unlabeled Track in the Trophallaxis Cluster
Unlabeled Track in the Trophallaxis Cluster

Fig. 9: Comparison of a complete track showing a trophallaxis event and two unlabeled tracks that were clustered into the same “trophallaxis” cluster.