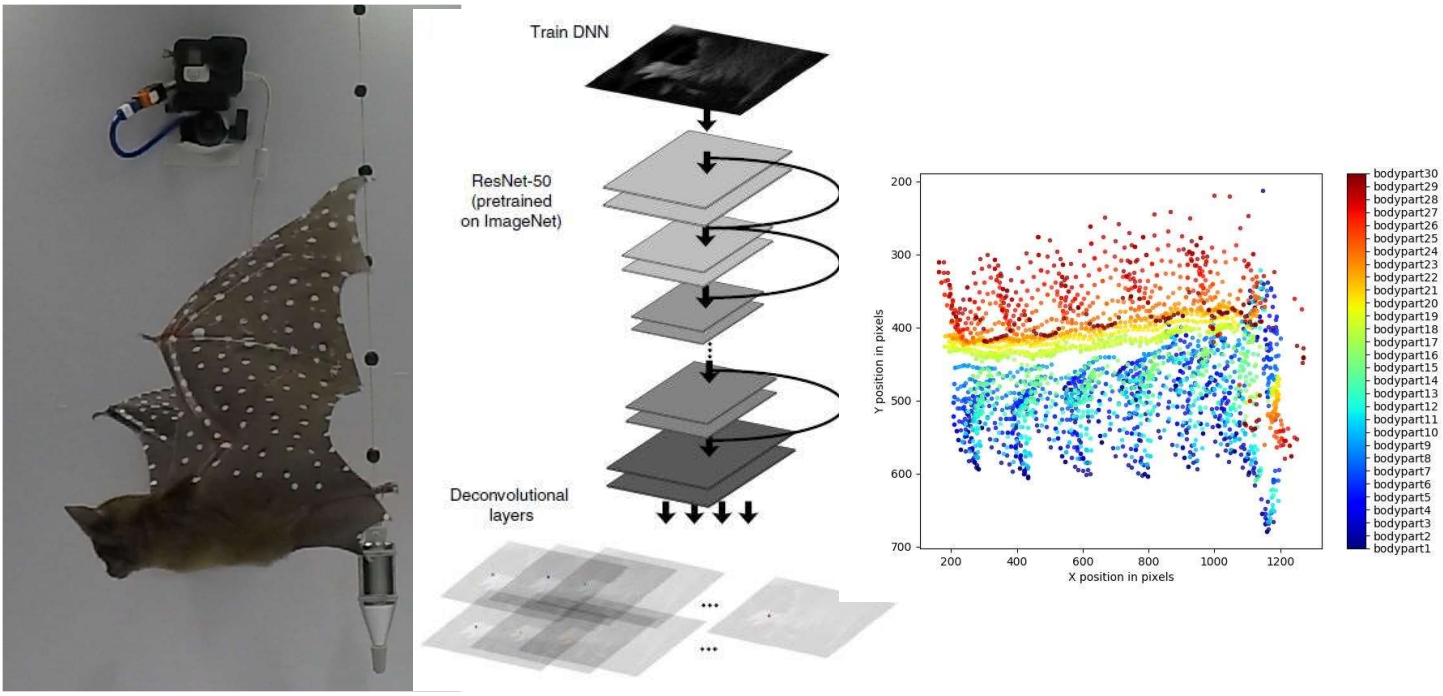
Abstract

Motion tracking for animal behavior research is a hot topic in terms of bio-inspired robot design and applications in the industry area. For flying animals such as bats, this is one of the ideal models we can learn from, the high-dimension muscle control ability could have a lot of potential inspirations. Deep Neural Network combined with big data has provided lots of advantages for solving those kinds of problems. In order to reduce the time consumption, transfer learning has been used to solve general machine learning tasks such as pattern classification and feature detection. Here we took advantage of a pre-trained neural network for bats flight tracking and achieved less than 5 pixels error for tracking flying bats with only 10 % of the data labelled.



Teaser Figure

The goal of this project is to track a flying bat based on a deep learning approach. The bat was marked with 150 white dots on the body, and the pretrained Resnet on ImageNet is used to track bat's trajectories. The figure on the right shows the tracked body parts' trajectories based on the Deeplabcut method. The figure uses different colors to represent different body parts we chose to track, we can see the yellow color dots represent the main body, and red and blue dots represent the two wings of the bat.

Introduction

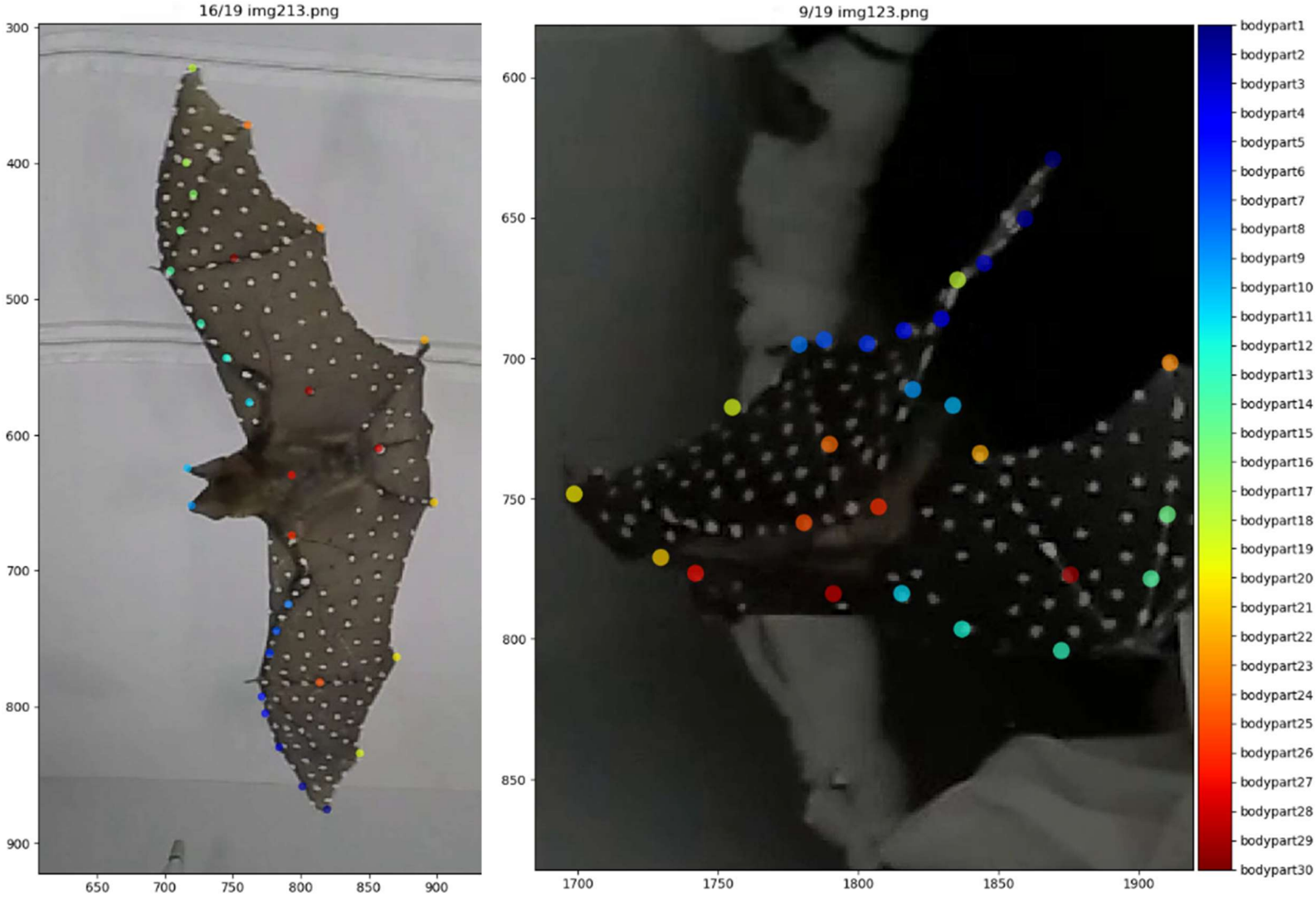
Deep learning approaches have been popular in solving computer vision problems. Big data and new versions of neural network architectures such as Alex-net, Res-net have been used to avoid some inherent disadvantages like overfitting and degradation. The integration of big data and new architecture seems to be the key method to utilize the DNN in a faster, more accurate way. Transfer

learning has given us a shortcut from input to the final step like classification with fewer layers of fine-tuning, which makes the training faster by avoiding the training process of a very deep network. Also, the classification or prediction accuracy has been increased with the benefit of huge pre-trained data sets, like ImageNet. The integration between transfer learning and ResNet-50 has been provided as a tool, Deeplabcut, to do the feature reorganizations and motion tracking work. We have utilized this method to track a flying bat with body markers. The video includes 241 frames and around 30 points marked on bats.

Approach

This project uses deep learning approaches (transfer learning, CNN, Resnet-50) to track the features of a moving animal. Deeplabcut will randomly select a certain percent of the data for manually labeling for training (here we use 10% and 20% of entire data for training), then we fine-tune it based on the pre-trained ResNet 50. The cross-entropy is the loss function for us to do the regression. After training a flight, a video will be saved with auto tracking markers. The DeepLabCut is a method published by Mathis, Alexander [1], and they build a library named deeplabcut, after we import this library, we just need to call the functions. However, here we do need to set some parameters to optimize the tracking accuracy, for example, the manually labeling percentage, the labels number and also the training iterations.

For the training data, after we select the video to track, several randomly selected frames of the video will be extracted, which can cover a wide range of the bat's behavior, for example, bat in different locations, and in different body poses. The following two figures are two typical manually labelled frames. We selected some easily recognized markers along the bats wing and main body parts, and we tried different marker numbers to test neural network's ability to track the points. Some parts are very hard to labelled correctly in different frames, and this results in a higher test error.



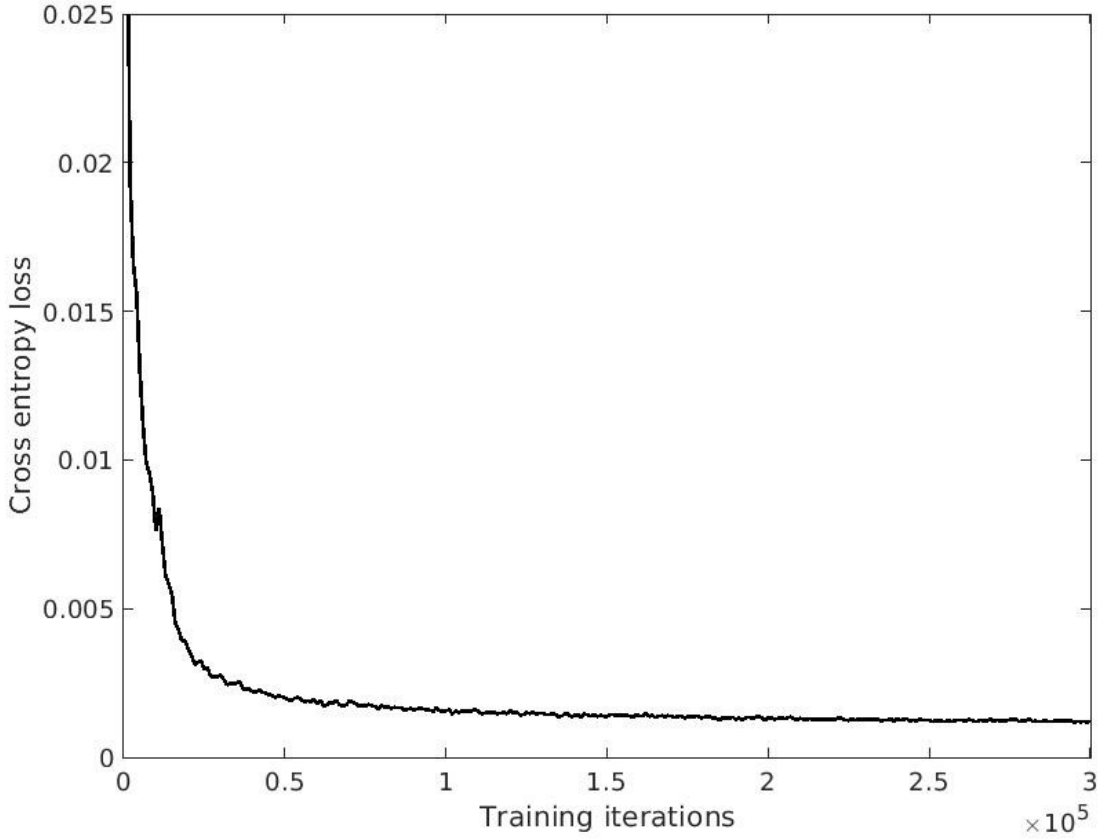
The main challenge we faced here is the data quality, the recording was taken by GoPro 3 with not very high resolution and can easily be influenced by changes of the lighting compared with professional motion-tracking cameras. In addition, the video we used here records a bat flying in a tunnel, and taking a U-turn at the end, this is another challenge for us since the accuracy for the marker matching in different frames will be related to the error rate of the result. That's why in some of the frames a lot of markers are not showing in the image and the results always have some error. It takes about 2 hours to finish labelling 30 body parts in 20 frames , and another 12 hours to finish the training, compared with midterm report, we increase the number of body part to tracked from 4 to 20 and 30, and we also repeated the manually labeling process three times for the training process. The right figure shows clearly the main challenge we had during the tracking, in some of the manually labelled frame, the predecided body part is not invisible or unable to be correctly located owing to the short range coverage of the Gopro.

Experiments and results

In this project, we train the network based on one video which records a bat fly around a lab. This time, two members of our team did three training sessions individually. Two training sessions train the network based on the whole video with different body part numbers and the other one trains the network based on only the second half of the video. The purpose of doing three
The training session is to show how deeply the result will be affected by the accuracy of the labelling process, since the second part of the video is relatively easier to label. The videos are recorded at 20 fps with a total of 241 frames. Deeplabcut randomly extracts 20 frames for us to label manually and then split the labeled frames into training and test datasets. The splitting ratio between training and test data is 0.95. In this case, it means Deeplabcut will randomly select one labelled frame as the test data. After training, we created a fully labeled version of the original video using the output of Deeplabcut to check the performance. This time, we have tested tracking 30 points on the bat's body other than 4 points in the midterm report. We only choose the tips of the bat's body to label because it is easy to keep track of the movement of these points between the frames and they are less likely to get occluded when the bat flaps its wings. To make sure we are marking the same point between different frames, we mark the points by following bat's wings and some special body parts like head and legs. The result of our experiment is shown by the following table:

	Training Iterations (k)	% of training dataset	Shuffle number	Train error (px)	Test error (px)
30 body parts 1	100	95	1	1.76	4.31
30 body parts 2	300	95	1	2.33	6.35
20 body parts	300	95	1	2.73	5.59

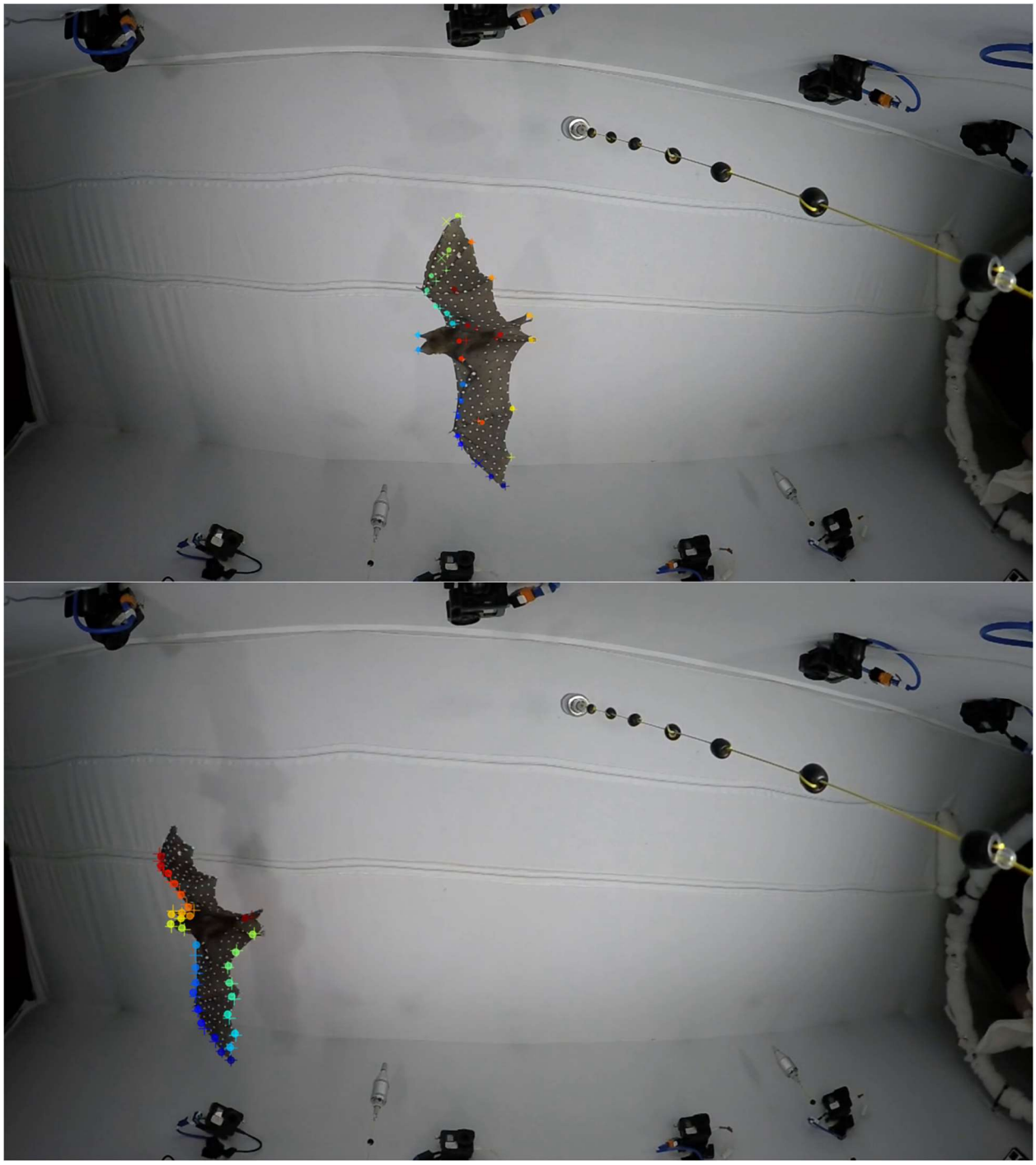
In this project, two members from our team did their individual training session based on the same video. The table show the evaluation result of our experiment of each training session. The error shown in the table is the mean average Euclidean error, which is proportional to the average root mean square error. We can see that the test error with probability cut-off is already below 5 pixels when the size of our training dataset is only 241 frames, which shows the efficiency of Deeplabcut. It can also be seen that the result is heavily affected by the labelling accuracy between the frames. If a body part is not labelled correctly consistently during the whole labelling process, it can lead to much higher error during testing, as shown in the table.



This graph shows the trend of cross-entropy loss as the training iteration increases. As we can see, the cross-entropy loss has already converged after 200000 iterations, which shows that we have done enough training for this network.

Qualitative Result

The following picture shows some results from deeplabcut. The dots indicate the prediction made by the network that passes the probability cutoff, which are the most likely pairs between ground truth label and prediction. The "×" indicates the prediction by the network that does not pass the probability cutoff, which means the network is not confident enough that this is a correct match. The "+" indicates the label marked manually. Each body part is color-coded differently to make it more easily to identify the match between manual labels and Deeplabcut prediction results. The two images shown below are from the two individual training sessions. The picture at the top is corresponded to the one with 300000 training iterations, and the picture at the bottom is corresponded to the one with 100000 training iterations.



It can be seen that the average error of the first picture is higher than that of the second. And there seems to be some outlier predictions that do not match with any label. For example, the yellow “+” at the bottom right of the bat’s wing in the first picture does not seem to match with any label, which indicates a prediction error made by Deeplabcut. In the second picture, all the predictions made by Therefore, we can see that the result of Deeplabcut pose tracking is heavily affected by the

accuracy of the ground truth label provided in the labelling process. Even though the first picture is from the training session with three times the training iterations of the training session the second picture from, the first picture still shows more errors.

Conclusion

In this project, we utilized an open-source software package called DeepLabCut, published by DeepLabCut lab in 2018 [2,3]. The toolbox uses a feature detector from DeeperCut and provides routines to (i) extract distinct frames from videos for labeling, (ii) generate training data based on labels, (iii) train networks to the desired feature sets, and (iv) extract these feature locations from unlabeled data. By using this, the expected results have been achieved, deeplabcut shows impressive ability to track the bat's motion with only 10 % of the data being manually labeled. The test error is less than 5 pixels after 300k training iterations with an input image with a resolution of 1920 * 1080. However, there are some requirements for the input data to achieve high accuracy performance, for example, the camera should cover most or entire data, otherwise the tracking error will be relatively high. In addition, another drawback of this method is it does require a certain number of manually provided labels, so when the data set is extremely big, then the manually labelled process will be time-consuming.

Different conditions of the training process give us a better understanding of this transfer-learning-based method, the key to track the motion well is by gathering decent data. And the philosophy of this method is based on super huge dataset (ImageNet, more than 14 million images with 22 thousand labels) trained on very deep neural network (ResNet-50), which is generally more reliable for the purpose of doing machine learning. We would highly recommend thinking of deeplabcut as a computer vision topic. More details of this project will be shown in the project video, we have a video to show the tracking performance during bat's flight.

Reference:

1 Windes, Peter, et al. "A computational investigation of lift generation and power expenditure of Pratt's roundleaf bat (Hipposideros pratti) in forward flight." PloS one 13.11 (2018): e0207613.

2 Mathis, Alexander, et al. "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning." Nature neuroscience 21.9 (2018): 1281-1289.

3 Nath, Tanmay, et al. "Using DeepLabCut for 3D markerless pose estimation across species and behaviors." Nature protocols 14.7 (2019): 2152-2176.