Journal of Visualized Experiments

Tracking rats in operant conditioning chambers, using a versatile homemade video camera and DeepLabCut

--Manuscript Draft--

Article Type:	Invited Methods Collection - JoVE Produced Video
Manuscript Number:	JoVE61409R2
Full Title:	Tracking rats in operant conditioning chambers, using a versatile homemade video camera and DeepLabCut
Keywords:	Operant conditioning; Cognition; Video analysis; Rodent behavior; Raspberry Pi; DeepLabCut
Corresponding Author:	Erik Karl, Håkan Clemensson Lunds Universitet Lund, SWEDEN
Corresponding Author's Institution:	Lunds Universitet
Corresponding Author E-Mail:	erik.clemensson@med.lu.se
Order of Authors:	Erik Karl, Håkan Clemensson
	Morteza Abbaszadeh
	Silvia Fanni
	Elena Espa
	Angela Cenci
Additional Information:	
Question	**Response**
Please indicate whether this article will be Standard Access or Open Access.	Standard Access (US$2,400)
Please indicate the **city, state/province, and country** where this article will be **filmed**. Please do not use abbreviations.	Sweden

Dear Editor and reviewers,

We greatly appreciate the constructive feedback given by all of you. We have made substantial changes to both text and figures to adhere to them and we believe that the manuscript has improved greatly as a result. We hope you find it to your liking.

We have attached both a clean version of the manuscript, and one with tracked changes marked. All references made in our responses in the rebuttal letter relate to the numbering in the clean revised document.

Best regards,

Erik Clemensson

TITLE:

Tracking Rats in Operant Conditioning Chambers, Using a Versatile Homemade Video Camera and DeepLabCut

AUTHORS & AFFILIATIONS:
Erik K. H. Clemensson[1], Morteza Abbaszadeh[1], Silvia Fanni[1], Elena Espa[1], M. Angela Cenci[1]

[1]Basal Ganglia Pathophysiology Unit, Dept. Experimental Medical Science, Lund University, Lund, Sweden

Corresponding Author:
Erik Clemensson (erik.clemensson@med.lu.se)

Email Addresses of Co-Authors:
Morteza Abbaszadeh (morteza.abbaszadeh@med.lu.se)
Silvia Fanni (silvia.fanni@med.lu.se)
Elena Espa (elena.espa@med.lu.se)
M. Angela Cenci (angela.cenci_nilsson@med.lu.se)

KEYWORDS:
Operant conditioning, cognition, video recording, rodent behavior, Raspberry Pi, DeepLabCut.

SHORT ABSTRACT:
This protocol describes how to build a small and versatile video camera, and how to use videos obtained from it to train a neural network to track the position of an animal inside operant conditioning chambers. This is a valuable complement to standard analyses of data logs obtained from operant conditioning tests.

LONG ABSTRACT:
Operant conditioning chambers are used to perform a wide range of behavioral tests in the field of neuroscience. The recorded data is typically based on the triggering of lever and nose-poke sensors present inside the chambers. While this provides a detailed view of when and how animals perform certain responses, it cannot be used to evaluate behaviors that do not trigger any sensors. As such, assessing how animals position themselves and move inside the chamber is rarely possible. To obtain this information, researchers generally have to record and analyze videos. Manufacturers of operant conditioning chambers can typically supply their customers with high-quality camera setups. However, these can be very costly and do not necessarily fit chambers from other manufacturers or other behavioral test setups. The current protocol describes how to build an inexpensive and versatile video camera using hobby electronics components. It further describes how to use the image analysis software package DeepLabCut to track the status of a strong light signal, as well as the position of a rat, in videos gathered from an operant conditioning chamber. The former is a great aid when selecting short segments of interest in videos that cover entire test sessions, and the latter enables analysis of parameters that cannot be obtained from the data logs produced by the operant chambers.

INTRODUCTION:

In the field of behavioral neuroscience, researchers commonly use operant conditioning chambers to assess a wide range of different cognitive and psychiatric features in rodents. While there are several different manufacturers of such systems, they typically share certain attributes and have an almost standardized design[1-3]. The chambers are generally square- or rectangle-shaped, with one wall that can be opened for placing animals inside, and one or two of the remaining walls containing components such as levers, nose-poke openings, reward trays, response wheels and lights of various kinds[1-3]. The lights and sensors present in the chambers are used to both control the test protocol and track the animals' behaviors[1-5]. The typical operant conditioning systems allow for a very detailed analysis of how the animals interact with the different operanda and openings present in the chambers. In general, any occasions where sensors are triggered can be recorded by the system, and from this data users can obtain detailed log files describing what the animal did during specific steps of the test[4,5]. While this provides an extensive representation of an animal's performance, it can only be used to describe behaviors that directly trigger one or more sensors[4,5]. As such, aspects related to how the animal positions itself and moves inside the chamber during different phases of the test are not well described[6-10]. This is unfortunate, as such information can be valuable for fully understanding the animal's behavior. For example, it can be used to clarify why certain animals perform poorly on a given test[6], to describe the strategies that animals might develop to handle difficult tasks[6-10], or to appreciate the true complexity of supposedly simple behaviors[11,12]. To obtain such articulate information, researchers commonly turn to manual analysis of videos[6-11].

When recording videos from operant conditioning chambers, the choice of camera is critical. The chambers are commonly located in isolation cubicles, with protocols frequently making use of steps where no visible light is shining[3,6-9]. Therefore, the use of infra-red (IR) illumination in combination with an IR-sensitive camera is necessary, as it allows visibility even in complete darkness. Further, the space available for placing a camera inside the isolation cubicle is often very limited, meaning that one benefits strongly from having small cameras that use lenses with a wide field of view (e.g., fish-eye lenses)[9]. While manufacturers of operant conditioning systems can often supply high-quality camera setups to their customers, these systems can be expensive and do not necessarily fit chambers from other manufacturers or setups for other behavioral tests. However, a notable benefit over using stand-alone video cameras is that these setups can often interface directly with the operant conditioning systems[13,14]. Through this, they can be set up to only record specific events rather than full test sessions, which can greatly aid in the analysis that follows.

The current protocol describes how to build an inexpensive and versatile video camera using hobby electronics components. The camera uses a fisheye lens, is sensitive to IR illumination and has a set of IR light emitting diodes (IR LEDs) attached to it. Moreover, it is built to have a flat and slim profile. Together, these aspects make it ideal for recording videos from most commercially available operant conditioning chambers as well as other behavioral test setups. The protocol further describes how to process videos obtained with the camera and how to use the software package DeepLabCut[15,16] to aid in extracting video sequences of interest as well as tracking an animal's movements therein. This partially circumvents the draw-back of using a stand-alone camera over the integrated solutions provided by operant manufacturers of conditioning systems, and offers a complement to manual scoring of behaviors.

Efforts have been made to write the protocol in a general format to highlight that the overall process can be adapted to videos from different operant conditioning tests. To illustrate certain key concepts, videos of rats performing the 5-choice serial reaction time test (5CSRTT)[17] are used as examples.

PROTOCOL:

All procedures that include animal handling have been approved by the Malmö-Lund Ethical committee for animal research.

1. Building the video camera

NOTE: A list of the components needed for building the camera is provided in the **Table of Materials**. Also refer to **Figure 1**, **Figure 2**, **Figure 3**, **Figure 4**, **Figure 5**.

1.1. Attach the magnetic metal ring (that accompanies the fisheye lens package) around the opening of the camera stand (**Figure 2A**). This will allow the fisheye lens to be placed in front of the camera.

1.2. Attach the camera module to the camera stand (**Figure 2B**). This will give some stability to the camera module and offer some protection to the electronic circuits.

1.3. Open the camera ports on the camera module and microcomputer (**Figure 1**) by gently pulling on the edges of their plastic clips (**Figure 2C**).

1.4. Place the ribbon cable in the camera ports, so that the silver connectors face the circuit boards (**Figure2C**). Lock the cable in place by pushing in the plastic clips of the camera ports.

1.5. Place the microcomputer in the plastic case and insert the listed micro SD card (**Figure 2D**).

NOTE: The micro SD card will function as the microcomputer's hard drive and contains a full operating system. The listed micro SD card comes with an installation manager preinstalled on it (New Out Of Box Software (NOOBS). As an alternative, one can write an image of the latest version of the microcomputer's operating system (Raspbian) to a generic micro SD card. For aid with this, please refer to official web resources[18]. It is preferable to use a class 10 micro SD card with 32 Gb of storage space. Larger SD cards might not be fully compatible with the listed microcomputer.

1.6. Connect a monitor, keyboard and a mouse to the microcomputer, and then connect its power supply.

1.7. Follow the steps as prompted by the installation guide to perform a full installation of the microcomputer's operating system (Raspbian). When the microcomputer has booted, ensure that it is connected to internet either through an ethernet cable or Wi-Fi.

1.8. Follow the steps outlined below to update the microcomputer's preinstalled software packages.

1.8.1. Open a terminal window (**Figure 3A**).

1.8.2. Type "sudo apt-get update" (excluding quotation marks) and press the Enter key (**Figure 3B**). Wait for the process to finish.

1.8.3. Type "sudo apt full-upgrade" (excluding quotation marks) and press enter. Make button responses when prompted and wait for the process to finish.

1.9. Under the **Start** menu, select **Preferences** and **Raspberry Pi configurations** (**Figure 3C**). In the opened window, go to the **Interfaces** tab and click to **Enable** the **Camera** and **I2C**. This is required for having the microcomputer work with the camera and IR LED modules.

1.10. Rename **Supplementary File 1** to "Pi_video_camera_Clemensson_2019.py". Copy it onto a USB memory stick, and subsequently into the microcomputer's /home/pi folder (**Figure 3D**). This file is a Python script, which enables video recordings to be made with the button switches that are attached in step 1.13.

1.11. Follow the steps outlined below to edit the microcomputer's rc.local file. This makes the computer start the script copied in step 1.10 and start the IR LEDs attached in step 1.13 when it boots.

CAUTION: This auto-start feature does not reliably work with microcomputer boards other than the listed model.

1.11.1. Open a terminal window, type "sudo nano /etc/rc.local" (excluding quotation marks) and press enter. This opens a text file (**Figure 4A**).

1.11.2. Use the keyboard's arrow keys to move the cursor down to the space between "fi" and "exit 0" (**Figure 4A**).

1.11.3. Add the following text as shown in **Figure 4B**, writing each string of text on a new line:

```
sudo i2cset -y 1 0x70 0x00 0xa5 &
sudo i2cset -y 1 0x70 0x09 0x0f &
sudo i2cset -y 1 0x70 0x01 0x32 &
sudo i2cset -y 1 0x70 0x03 0x32 &
sudo i2cset -y 1 0x70 0x06 0x32 &
sudo i2cset -y 1 0x70 0x08 0x32 &
```

sudo python /home/pi/Pi_video_camera_Clemensson_2019.py &

1.11.4. Save the changes by pressing **Ctrl + x** followed by **y** and **Enter**.

1.12. Solder together the necessary components as indicated in **Figure 5A**, and as described below.

1.12.1. For the two colored LEDs, attach a resistor and a female jumper cable to one leg, and a female jumper cable to the other (**Figure 5A**). Try to keep the cables as short as possible. Take note of which of the LED's electrodes is the negative one (typically the short one), as this needs to be connected to ground on the microcomputer's general-purpose input/output (GPIO) pins.

1.12.2. For the two button switches, attach a female jumper cable to each leg (**Figure 5A**). Make the cables long for one of the switches, and short for the other.

1.12.3. To assemble the IR LED module, follow instructions available on its official web resources[19].

1.12.4. Cover the soldered joints with shrink tubing to limit the risk of short-circuiting the components.

1.13. Switch off the microcomputer and connect the switches and LEDs to its GPIO pins as indicated in **Figure 5B**, and described below.

CAUTION: Wiring the components to the wrong GPIO pins could damage them and/or the microcomputer when the camera is switched on.

1.13.1. Connect one LED so that its negative end connects to pin #14 and its positive end connects to pin #12. This LED will shine when the microcomputer has booted and the camera is ready to be used.

1.13.2. Connect the button switch with long cables so that one cable connects to pin #9 and the other one to pin #11. This button is used to start and stop the video recordings.

NOTE: The script that controls the camera has been written so that this button is unresponsive for a few seconds just after starting or stopping a video recording.

1.13.3. Connect one LED so that its negative end connects to pin #20 and its positive end connects to pin #13. This LED will shine when the camera is recording a video.

1.13.4. Connect the button switch with the short cables so that one cable connects to pin #37 and the other one to pin #39. This switch is used to switch off the camera.

1.13.5. Connect the IR LED module as described in its official web resources[19].

2. Designing the operant conditioning protocol of interest

NOTE: To use DeepLabCut for tracking the protocol progression in videos recorded from operant chambers, the behavioral protocols need to be structured in specific ways, as explained below.

2.1. Set the protocol to use the chamber's house light, or another strong light signal, as an indicator of a specific step in the protocol (such as the start of individual trials, or the test session) (**Figure 6A**). This signal will be referred to as the "protocol step indicator" in the remainder of this protocol. The presence of this signal will allow tracking protocol progression in the recorded videos.

2.2. Set the protocol to record all responses of interest with individual timestamps in relation to when the protocol step indicator becomes active.

3. Recording videos of animals performing the behavioral test of interest

3.1. Place the camera on top of the operant chambers, so that it records a top view of the area inside (**Figure 7**).

NOTE: This is particularly suitable for capturing an animals' general position and posture inside the chamber. Avoid placing the camera's indicator lights and the IR LED module close to the camera lens.

3.2. Start the camera by connecting it to an electrical outlet via the power supply cable.

NOTE: Prior to first use, it is beneficial to set the focus of the camera, using the small tool that accompanies the camera module.

3.3. Use the button connected in step 1.13.2 to start and stop video recordings.

3.4. Switch off the camera by following these steps.

3.4.1. Push and hold the button connected in step 1.13.4 until the LED connected in step 1.13.1 switches off. This initiates the camera's shut down process.

3.4.2. Wait until the green LED visible on top of the microcomputer (**Figure 1**) has stopped blinking.

3.4.3. Remove the camera's power supply.

CAUTION: Unplugging the power supply while the microcomputer is still running can cause corruption of the data on the micro SD card.

3.5. Connect the camera to a monitor, keyboard, mouse and USB storage device and retrieve the video files from its desktop.

NOTE: The files are named according to the date and time when video recording was started. However, the microcomputer does not have an internal clock and only updates its time setting when connected to the internet.

3.6. Convert the recorded videos from .h264 to .MP4, as the latter works well with DeepLabCut and most media players.

NOTE: There are multiple ways to achieve this. One is described in **Supplementary File 2**.

4. Analyzing videos with DeepLabCut

NOTE: DeepLabCut is a software package that allows users to define any object of interest in a set of video frames, and subsequently use these to train a neural network in tracking the objects' positions in full-length videos[15,16]. This section gives a rough outline for how to use DeepLabCut to track the status of the protocol step indicator and the position of a rat's head. Installation and use of DeepLabCut is well-described in other published protocols[15,16]. Each step can be done through specific Python commands or DeepLabCut's graphic user interface, as described elsewhere[15,16].

4.1. Create and configure a new DeepLabCut project by following the steps outlined in[16].

4.2. Use DeepLabCut's frame grabbing function to extract 700–900 video frames from one or more of the videos recorded in section 3.

NOTE: If the animals differ considerably in fur pigmentation or other visual characteristics, it is advisable that the 700–900 extracted video frames are split across videos of different animals. Through this, one trained network can be used to track different individuals.

4.2.1. Make sure to include video frames that display both the active (**Figure 9A**) and inactive (**Figure 9B**) state of the protocol step indicator.

4.2.2. Make sure to include video frames that cover the range of different positions, postures and head movements that the rat may show during the test. This should include video frames where the rat is standing still in different areas of the chamber, with its head pointing in different directions, as well as video frames where the rat is actively moving, entering nose poke openings and entering the pellet trough.

4.3. Use DeepLabCut's Labeling Toolbox to manually mark the position of the rat's head in each video frame extracted in step 4.2. Use the mouse cursor to place a "head" label in a central position between the rat's ears (**Figure 9A,B**). In addition, mark the position of the chamber's house light (or other protocol step indicator) in each video frame where it is actively shining (**Figure 9A**). Leave the house light unlabeled in frames where it is inactive (Figure 9B).

4.4. Use DeepLabCut's "create training data set" and "train network" functions to create a training data set from the video frames labeled in step 4.3 and start the training of a neural network. Make sure to select "resnet_101" for the chosen network type.

4.5. Stop the training of the network when the training loss has plateaued below 0.01. This may take up to 500,000 training iterations.

NOTE: When using a GPU machine with approximately 8 GB of memory and a training set of about 900 video frames (resolution: 1640 x 1232 pixels), the training process has been found to take approximately 72 h.

4.6. Use DeepLabCut's video analysis function to analyze videos gathered in step 3, using the neural network trained in step 4.4. This will provide a .csv file listing the tracked positions of the rat's head and the protocol step indicator in each video frame of the analyzed videos. In addition, it will create marked-up video files where the tracked positions are displayed visually (**Videos 1, 3**, **4**, **5**, **6**, **7**, **8**).

4.7. Evaluate the accuracy of the tracking by following the steps outlined below.

4.7.1. Use DeepLabCut's built-in evaluate function to obtain an automated evaluation of the network's tracking accuracy. This is based on the video frames that were labeled in step 4.3 and describes how far away on average the position tracked by the network is from the manually placed label.

4.7.2. Select one or more brief video sequences (of about 100–200 video frames each) in the marked-up videos obtained in step 4.6. Go through the video sequences, frame by frame, and note in how many frames the labels correctly indicate the positions of the rat's head, tail, etc., and in how many frames the labels are placed in erroneous positions or not shown.

4.7.2.1. If the label of a body part or object is frequently lost or placed in an erroneous position, identify the situations where tracking fails. Extract and add labeled frames of these occasions by repeating steps 4.2. and 4.3. Then retrain the network and reanalyze the videos by repeating steps 4.4-4.7. Ultimately, tracking accuracy of >90% accuracy should be achieved.

5. Obtaining coordinates for points of interest in the operant chambers

5.1. Use DeepLabCut as described in step 4.3 to manually mark points of interest in the operant chambers (such as nose poke openings, levers, etc.) in a single video frame (**Figure 8C**). These are manually chosen depending on study-specific interests, although the position of the protocol step indicator should always be included.

5.2. Retrieve the coordinates of the marked points of interest from the .csv file that DeepLabCut automatically stores under "labelled data" in the project folder.

6. Identifying video segments where the protocol step indicator is active

6.1. Load the .csv files obtained from the DeepLabCut video analysis in step 4.6 into a data management software of choice.

NOTE: Due to the amount and complexity of data obtained from DeepLabCut and operant conditioning systems, the data management is best done through automated analysis scripts. To get started with this, please refer to entry-level guides available elsewhere[20-22].

6.2. Note in which video segments the protocol step indicator is tracked within 50–100 pixels of the position obtained in section 5. These will be periods where the protocol step indicator is active (**Figure 6B**).

NOTE: During video segments where the protocol step indicator is not shining, the marked-up video might seem to indicate that DeepLabCut is not tracking it to any position. However, this is rarely the case, and it is instead typically tracked to multiple scattered locations.

6.3. Extract the exact starting point for each period where the protocol step indicator is active (**Figure 6C: 1**).

7. Identifying video segments of interest

7.1. Consider the points where the protocol step indicator becomes active (**Figure 6C**: 1) and the timestamps of responses recorded by the operant chambers (section 2, **Figure 6C**: 2).

7.2. Use this information to determine which video segments cover specific interesting events, such as inter-trial intervals, responses, reward retrievals etc. (**Figure 6C**: 3, **Figure 6D**).

NOTE: For this, keep in mind that the camera described herein records videos at 30 fps.

7.3. Note the specific video frames that cover these events of interest.

7.4. (Optional) Edit video files of full test sessions to include only the specific segments of interest.

NOTE: There are multiple ways to achieve this. One is described in **Supplementary File 3**. This greatly helps when storing large numbers of videos and can also make reviewing and presenting results more convenient.

8. Analyzing the position and movements of an animal during specific video segments

8.1. Subset the full tracking data of head position obtained from DeepLabCut in step 4.6 to only include video segments noted under section 7.

8.2. Calculate the position of the animal's head in relation to one or more of the reference points selected under section 5 (**Figure 8C**). This enables comparisons of tracking and position across different videos.

8.3. Perform relevant in-depth analysis of the animal's position and movements.

NOTE: The specific analysis performed will be strongly study-specific. Some examples of parameters that can be analyzed are given below.

8.3.1. Visualize path traces by plotting all coordinates detected during a selected period within one graph.

8.3.2. Analyze proximity to a given point of interest by using the following formula:

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

8.3.3. Analyze changes in speed during a movement by calculating the distance between tracked coordinates in consecutive frames and divide by 1/fps of the camera.

REPRESENTATIVE RESULTS:

Video camera performance

The representative results were gathered in operant conditioning chambers for rats with floor areas of 28.5 cm x 25.5 cm, and heights of 28.5 cm. With the fisheye lens attached, the camera captures the full floor area and large parts of the surrounding walls, when placed above the chamber (**Figure 7A**). As such, a good view can be obtained, even if the camera is placed off-center on the chamber's top. This should hold true for comparable operant chambers. The IR LEDs are able to illuminate the entire chamber (**Figure 7B, C**), enabling a good view, even when all other lights inside the chamber are switched off (**Figure 7C**). However, the lighting in such situations is not entirely even, and may result in some difficulties in obtaining accurate tracking. If such analysis is of interest, additional sources of IR illumination might be required. It is also worth noting that some chambers use metal dropping pans to collect urine and feces. If the camera is placed directly above such surfaces, strong reflections of the IR LEDs' light will be visible in the recorded videos (**Figure 7B**). This can, however, be avoided by placing paper towels in the dropping pan, giving a much-improved image (**Figure 7C**). Placing the camera's IR or colored LEDs too close to the camera lens may result in them being visible in the image periphery (**Figure 7B**). As the camera is IR sensitive, any IR light sources that are present inside the chambers may be visible in the videos. For many setups, this will include the continuous shining of IR beam break sensors (**Figure 7C**). The continuous illumination from the camera's IR LEDs does not disturb the image quality of well-lit chambers (**Figure 7D**). The size of the videos recorded with the camera is approximately 77 Mb/min. If a 32 Gb micro SD card is used for the camera, there should be about 20 Gb available following the installation of the operating system. This leaves room for approximately 260 min of recorded footage.

The fisheye lens causes the camera to have a slightly uneven focus, being sharp in the center of the image but reduced sharpness towards the edges. This does not appear to affect the accuracy of tracking. Moreover, the fisheye lens results in the recorded image being distorted. For example, the distances between equally spaced points along straight lines will show artificially reduced spacing towards the periphery of the image (**Figure 9A,B**). If the camera is used for applications where most of the field of view or absolute measurements of distance and speed are of interest, it is worth considering correcting the data for this distortion[23] (**Supplementary File 4**). The distortion is, however, relatively mild in the center of the image (**Figure 9B**). For videos gathered in our operant chamber, the area of interest is limited to the central 25% of the camera's field of view. Within this area, the effect of the fisheye distortion is minimal (**Figure 9C–F**).

Accuracy of tracking with DeepLabCut

The main factors that will determine the tracking accuracy of a trained network are (i) the number of labeled frames in its training data set, (ii) how well those labeled frames capture the behavior of interest and (iii) the number of training iterations used. DeepLabCut includes an evaluate function, which reports an estimate on how well it performs. This output provides an estimate on how far away (in numbers of pixels) its tracking can be expected to be from the actual location of an object. This, however, does not necessarily give a good description of the number of frames where an object is lost and/or mislabeled (**Figure 10**), prompting the need for additional manual assessment of tracking accuracy.

For analyzing behaviors inside an operant chamber, a well-trained network should allow the accurate identification of all events where the protocol step indicator is active. If not, retraining the network or choosing a different indicator might be needed. Despite having a well-trained network, tracking of the protocol step indicator may on occasion be disrupted by animals blocking the camera's view (**Figure 10B**). This will cause breaks in the tracking that are reminiscent of episodes where the indicator is inactive. The frequency of this happening will depend on animal strain, type of behavioral protocol and choice of protocol step indicator. In the example data from the 5CSRTT that is used, here it occurred on four out of 400 trials (data not shown). All occasions were easily identifiable, as their durations did not match that of the break step that had been included in the protocol design (**Figure 6A**). Ultimately, choosing an indicator that is placed high up in the chamber and away from components that animals interact with is likely to be helpful.

A well-trained network should allow >90% accuracy when tracking an animal's head during video segments of interest (**Video 1**). With this, only a small subset of video frames will need to be excluded from the subsequent analysis and usable tracking data will be obtainable from virtually all trials within a test session. Accurate tracking is clearly identifiable by markers following an animal throughout its movements (**Video 2**) and plotted paths appearing smooth (**Figure 10C**). In contrast, inaccurate tracking is characterized by markers not reliably staying on target (**Video 3**) and by plotted paths appearing jagged (**Figure 10D**). The latter is caused by the object being tracked to distant erroneous positions in single video frames within sequences of accurate tracking. As a result of this, inaccurate tracking typically causes sudden shifts in calculated movement speeds (**Figure 10E**). This can be used to identify video frames where tracking is inaccurate, to exclude them from subsequent analysis. If there are substantial problems with tracking accuracy, the occasions where tracking fails should be identified and the network should be retrained using an expanded training set containing labeled video frames of these events (**Figure 10A,E**).

Use of video tracking to complement analysis of operant behaviors

Analyzing how an animal moves and positions itself during operant tests will provide multiple insights into the complex and multifaceted nature of their behaviors. By tracking where an animal is located throughout a test session, one can assess how distinct movement patterns relate to performance (**Figure 11A,B**). By further investigating head movements during specific protocol steps, one can detect and characterize the use of different strategies (**Figure 11C–E**).

To exemplify, consider the representative data presented for rats performing the 5CSRTT test (**Figure 6A, Figure 11**). In this test, animals are presented with multiple trials that each start with a 5 s waiting step (inter-trial interval - ITI) (**Figure 6A**: 1). At the end of this, a light will shine inside one of the nose poke openings (randomly chosen position on each trial, **Figure 6A**: 2). Nose-poking into the cued opening is considered a correct response and is rewarded (**Figure 6A**: 3). Responding into another opening is considered incorrect. Failing to respond within 5 s following the presentation of the light is considered an omission. Tracking head movements during the ITI of this test has revealed that on trials where rats perform a response, they are fast at moving towards the area around the nose poke openings (**Figure 11A,B**, **Video 4**). In contrast, on the majority of omission trials, the rats fail to approach the area around the openings (**Figure 11B**, **Video 5**). This behavior is in line with the common interpretation of omissions being closely related to a low motivation to perform the test[3,16]. However, on a subset of omission trials (approximately 20% of the current data set), the rats showed a clear focus towards the openings (**Figure 11B**, **Video 6**) but failed to note the exact location of the cued opening. The data thus indicate that there are at least two different types of omissions, one related to a possible disinterest in the ongoing trial, and another that is more dependent on insufficient visuospatial attention[3]. Head tracking can also be used to distinguish apparent strategies. As an example, two distinct attentional strategies were revealed when analyzing how the rats move when they are in proximity to the nose poke openings during the 5CSRTT (**Figure 11C–E**). In the first strategy, rats showed an extremely focused approach, maintaining a central position throughout most of the ITI (**Figure 11C**, **Video 7**). In contrast, rats adopting the other strategy constantly moved their heads between the different openings in a search-like manner (**Figure 11D**, **Video 8**). This type of behavioral differences can conveniently be quantified by calculating the amount of time spent in proximity to the different openings (**Figure 11E**). Finally, by analyzing which opening the rat is closest to at the time of cue light presentation (**Figure 11F**), it can be demonstrated that being in a central position (**Figure 11G**) and/or in close proximity to the location of the cued opening (**Figure 11H**) seems to be beneficial for accurate performance on the test.

FIGURE AND TABLE LEGENDS:

Figure 1: Sketch of the listed microcomputer. The schematic shows the position of several components of interest on the microcomputer motherboard. These are marked with circled numbers as follows: 1: Connector for camera ribbon cable; 2: LED light indicating when computer is running; 3: Micro USB for power cable; 4: Micro USB for mouse/keyboard; 5: General-purpose input/output pins (GPIO pins), these pins are used to connect the microcomputer to LEDs, switches, and the IR LED module; 6: Mini HDMI output; 7: Micro SD card slot. In the lower portion of the figure, a cropped and enlarged part of the GPIO pins is shown to indicate how to count along them to correctly identify the position of a specific pin.

Figure 2: Building the main body of the camera. The figures illustrated the main steps for building the body of the camera. (**A**) Attach the magnetic metal ring to the camera stand. (**B**) Attach the camera module to the camera stand. (**C**) Connect the camera module to the microcomputer via the flat ribbon cable. Note the white arrows indicating how to open and close the camera ports present on both the microcomputer and the camera module. (**D**) Place the microcomputer into the plastic casing and insert a micro SD card.

Figure 3: Updating the microcomputer's operating system and enabling the peripherals. The figure shows four different screenshots depicting the user interface of the microcomputer. (**A**) Terminal windows can be opened by clicking the "terminal" icon in the top left corner of the screen. (**B**) Within the terminal, one can type in different kinds of commands, as detailed in the protocol text. The screenshot displays the command for updating the system's software packages. (**C**) The screenshot displays how to navigate to the configurations menu, where one can enable the use of the camera module and the I2C GPIO pins. (**D**) The screenshot displays the /home/pi folder, where the camera script should be copied in step 1.10 of the protocol. The window is opened by clicking the indicated icon in the top left corner of the screen.

Figure 4: Configuring the microcomputer's rc.local file. The figure displays two screenshots of the microcomputer's rc.local file, when accessed through the terminal as described in step 1.11.1. (**A**) A screenshot of the rc.local file in its original format. The arrow indicates the space where text needs to be entered in order to enable the auto-start feature of the camera. (**B**) A screenshot of the rc.local file after it has been edited to shine the IR LEDs and start the python script controlling the camera upon startup of the microcomputer.

Figure 5: Connecting of switches and LEDs to microcomputer's GPIO pins. (**A**) Schematic showing a button switch with female jumper cables (top) and a LED with resistor and female jumper cables (bottom). (1) Button switch, (2) female jumper cables, (3) LED, (4) resistor. (**B**) Schematic image showing how the two button switches, the colored LEDs and the IR LED board are connected to the GPIO pins of the microcomputer. Blue cables and GPIO pins indicate ground. The position of two GPIO pins are indicated in the figure (GPIO pins #2 and #40): (1) Button for starting/stopping video recording. (2) LED indicating when video is being recorded. (3) Button for

switching off camera. (4) LED indicating when the camera has booted and is ready to be used. (5) IR LED module. Note that circuits with LEDs also contain 330 Ω resistors.

Figure 6: Using DeepLabCut tracking of protocol step indicator to identify sequences of interest in full-length videos. (**A**) Schematic of the steps of a single trial in the 5-choice serial reaction time test (5CSRTT): (1) First, there is a brief waiting period (ITI). Arrow indicates an actively shining house light. (2) At the end of the ITI, a light will shine in one of the five nose poke openings (arrow). (3) If a rat accurately responds by performing a nose poke into the cued opening, a reward is delivered (arrow). (4) The rat is allowed to retrieve the reward. (5) To enable the use of the house light as a protocol step indicator, a brief pause step where the house light is switched off (arrow) is implemented before the next trial begins. Note that the house light is shining during subsequent steps of the trial. (**B**) An example graph depicting the x-coordinate of the active house light, as tracked by DeepLabCut, during a video segment of a 5CSRTT test. During segments where the house light is shining (indicator active - 1), the position is tracked to a consistent and stable point (also note the red marker (indicated by arrow) in the example video frame), comparable to that of the house light's position in Figure 8C (x, y: 163, 503). During segments where the house light is not shining (indicator inactive - 2, note the white arrow in the example video frame), the tracked position is not stable, and far away from the house light's actual coordinate. (**C**) Table 1 shows an example of processed output obtained from DeepLabCut tracking of a protocol step indicator. In this output, the starting point for each occasion where the indicator is active has been listed. Table 2 depicts an example of data obtained from the operant conditioning system, giving relevant details for individual trials. In this example, the duration of the ITI, position of the cued opening and latencies to perform a response and retrieve the reward have been recorded. Table 3 depicts an example of data obtained by merging tracking results from DeepLabCut and data recorded from the operant conditioning system. Through this, the video frames for the starting point of the ITI (step 1 in A), the starting point of the cue light presentation (step 2 in A), the response (step 3 in A) and retrieval (step 4 in A) for an example trial have been obtained. (**D**) An example graph depicting the x-coordinate of the house light, as tracked by DeepLabCut, during a filmed 5CSRTT trial. The different steps of the protocol are indicated: (1) ITI; (2) presentation of a cue light (position indicated by white arrow); (3) response; (4) reward retrieval. The identification of video frames depicting the start and stop of these different protocol steps was done through a process comparable to that indicated in D.

Figure 7: Image characteristics of camera. (**A**) Uncropped image obtained from a camera placed on top of an operant conditioning chamber. The image was captured while the chamber was placed in a brightly lit room. Note the (1) house light and (2) reward pellet trough along the chamber's left wall and (3) the row of five nose poke openings along the chamber's right wall. Each nose poke opening contains a small cue light. (**B**) Uncropped image displaying the strong reflection caused by (1) the metal dropping pan, as well as reflections caused by sub-optimal positioning of the camera's (2) indicator LEDs and (3) IR LED module. (**C**) Cropped image of the chamber in complete darkness. Note that the lights from the IR beam break detectors in the five nose poke openings along the chamber's right wall are clearly visible (arrow). (**D**) Cropped image of the chamber when brightly lit.

Figure 8: Positional tracking of protocol step indicator and body parts of interest. (**A**) The picture shows the position of a protocol step indicator (red) as well as the head (yellow) and tail (green) of a rat, as tracked by DeepLabCut. As indicated by the tracking of the lit house light, the video frame is taken from a snapshot of an active trial. (**B**) The picture shows the position of the head (yellow) and tail (green) as tracked by DeepLabCut during a moment when a trial is not active. Note the lack of house light tracking. (**C**) The positions of points of interest used in the analysis of data shown in **Figure 6** and **Figure 11**; (1) House light, in this case used as protocol step indicator, (2–6) Nose poke openings #1–5.

Figure 9: Image distortion from fisheye lens. (**A**) Image of a checker-board pattern with equally sized and spaced black and white squares taken with the camera described in this protocol. Image was taken at a height comparable to that used when recording videos from operant conditioning chambers. Black squares along the central horizontal and vertical lines have been marked with DeepLabCut. (**B**) Graph depicting how the spacing of the marked squares in (A) change with proximity to the image center. (**C**) Image depicting measurements taken to evaluate impact of fisheye distortion effect on videos gathered from operant chambers. The corners and midpoints along the edge of the floor area, the central position of each individual floor rung and the position of the five nose poke openings have been indicated with DeepLabCut (colored dots); (1) spacing of floor rungs, (2) width of chamber floor along the middle of the chamber, (3) spacing of nose poke openings. (**D**) Spacing of floor rungs (averaged for each set of three consecutive rungs), numbered from left to right in (C). There is a small effect of the fisheye distortion, resulting in the central rungs being spaced roughly 3 pixels (8%) further apart than rungs that are positioned at the edges of the chamber floor. (**E**) Width of the chamber floor in (C) measured at its left and right edges, as well as midpoint. There is a small effect of the fisheye distortion, resulting in the width measured at the midpoint being roughly 29 pixels (5%) longer than the other measurements. (**F**) Spacing of nose poke openings in (C), numbered from the top of the image. There is a small effect of the fisheye distortion, resulting in the spacing between the central three openings (H2, H3, H5) being roughly 5 pixels (4%) broader than the spacing between H1-H2 and H4-H5. For D-F, data were gathered from four videos and graphs depict group mean + standard error.

Figure 10: Reviewing accuracy of DeepLabCut tracking. (**A**) A table listing training information for two neural networks trained to track rats within operant chambers. Network #1 used a smaller training data set, but high number of training iterations compared to Network #2. Both networks achieved a low error score from DeepLabCut's evaluation function (DLC test error) and displayed a low training loss towards the end of the training. Despite this, Network #1 showed very poor tracking accuracy upon manual evaluation of marked video frames (measured accuracy, estimated from 150 video frames covering a video segment comparable to those in **Video 2** and **Video 3**). Network #2 represents the improved version of Network #1, after having included additional video frames of actively moving rats into the training data set, as described in (E). (**B**) Image depicting a rat rearing up and covering the chamber's house light (**Figure 7A**) with its head, disrupting the tracking of it. (**C**) Video frame capturing a response made during a 5CSRTT trial (**Figure 6A**: 3). The head's movement path during the response and preceding ITI has been superimposed on the image in yellow. The tracking is considered to be accurate. Note the

smooth tracking during movements (white arrow). A corresponding video is available as **Video 2**. Network #2 (see A) was used for tracking. (**D**) Video frame capturing a response made during a 5CSRTT trial (**Figure 6A**: 3). The head's movement path during the response and preceding ITI has been superimposed on the image in yellow. Data concerns the same trial as shown in (C) but analyzed with Network #1 (see A). The tracking is considered to be inaccurate. Note the path's jagged appearance with multiple straight lines (white arrows), caused by occasional tracking of the head to distant erroneous positions (black arrows). A corresponding video is available as **Video 3**. (**E**) Graph depicting the dynamic changes in movement speed of the head tracking in (C) and (D). Identifiable in the graph are three major movements seen in **Video 2** and **3**, where the rat first turns to face the nose poke openings (initial turn), makes a small adjustment to further approach them (adjustment), and finally performs a response. The speed profile for the good tracking obtained by Network #2 (A) displays smooth curves of changes in movement speed (blue arrows), indicating an accurate tracking. The speed profile for the poor tracking obtained by Network #1 (A) shows multiple sudden spikes in movement speed (red arrows) indicative of occasional tracking errors in single video frames. It is worth noting that these tracking problems specifically occur during movements. To rectify this, the initial training set used to train Network #1 was expanded with a large amount of video frames depicting actively moving rats. This was subsequently used to train Network #2, which efficiently removed this tracking issue.

Figure 11: Use of positional tracking through DeepLabCut to complement the behavioral analysis of operant conditioning tests. (**A**) A top view of the inside of an operant conditioning chamber. Three areas of the chamber are indicated. The area close to the reward pellet trough (Pellet), the central chamber area (Center) and the area around the nose poke openings (Openings). (**B**) A graph depicting the relative amount of time rats spend in the three different areas of the operant chamber outlined in (A) during the ITI step of the 5CSRTT. Note that on trials with a response, rats initially tend to be positioned close to the pellet trough (black) and chamber center (grey), but as the ITI progresses, they shift towards positioning themselves around the nose poke openings (white). In contrast, on typical omission trials, rats remain positioned around the pellet trough and chamber center. On a subset of omission trials (approximately 20%) rats clearly shift their focus towards the nose poke openings, but still fail to perform a response when prompted. Two-way ANOVA analysis of the time spend around the nose poke openings using trial type as between-subject factor and time as within-subject factor reveal significant time ($p < 0.001$, $F(4,8) = 35.13$), trial type ($p < 0.01$, $F(2,4) = 57.33$) and time x trial type interaction ($p < 0.001$, $F(8,16) = 15.3$. Data gathered from four animals performing 100 trials each. Graph displays mean + standard error. (**C**) Heat map displaying all head positions tracked in proximity of the nose poke openings, by one specific rat during 50 ITIs of a 5CSRTT test session. Note that the rat has a strong tendency to keep its head in one spot close to the central nose poke opening. (**D**) Heat map displaying all head positions tracked in proximity of the nose poke openings, by one specific rat during 50 ITIs of a 5-CSRTT test session. Note that the rat shows no clear preference for any specific opening. (**E**) Graph depicting the relative amount of time that the two rats displayed in (C) and (D) spend being closest to the different nose poke openings during 50 ITIs of the 5CSRTT. The rat displaying a focused strategy (C) (black) shows a strong preference for being closest to the central opening while the rat with a search-like strategy (D) (white) shows no preference for any particular opening. The graph depicts average + standard error. (**F**) An image

of a rat at the time of cue presentation on a 5CSRTT trial (**Figure 6A**). Note that the rat has positioned its head closest to the central opening (white arrow), being two openings away from the cued opening (black arrow). (**G**) A graph depicting performance accuracy on the 5CSRTT (i.e., frequency of performing correct responses) in relation to whether the head of the rats was closest to the central opening or one of the other openings at the time of cue presentation (**Figure 6A2**). Data gathered from four animals performing roughly 70 responses each. Graph displays group mean + standard error (matched t-test: $p < 0.05$). (**H**) A graph depicting performance accuracy on the 5CSRTT in relation to the distance between the position of the cued opening and the position of a rat's head, at the point of signal presentation. The distance relates to the number of openings between the rats' head position and the position of the signaled opening. Data gathered from four animals performing roughly 70 responses each. Graph displays mean + standard error (matched one-way ANOVA: $p < 0.01$). For the presented analysis, Network #2 described in **Figure 10A** was used. The complete analyzed data set included approximately 160,000 video frames and 400 trials. Out of these, 2.5% of the video frames were excluded due the animal's noted movement speed being above 3,000 pixels/s, indicating erroneous tracking (**Figure 10E**). No complete trials were excluded.

Video 1: Representative tracking performance of well-trained neural network. The video shows a montage of a rat performing 45 trials with correct responses during a 5CSRTT test session (see **Figure 8A** for protocol details). Tracking of the house light (red marker), tail base (green marker) and head (blue marker) are indicated in the video. The training of the network (Network #2 in **Figure 10A**) emphasized accuracy for movements made along the chamber floor in proximity to the nose poke openings (right wall, **Figure 7A**). Tracking of these segments show on average >90% accuracy. Tracking of episodes of rearing and grooming are inaccurate, as the training set did not include frames of these behaviors. Note that the video has been compressed to reduce file size and is not representable of the video quality obtained with the camera.

Video 2: Example of accurately tracked animal. The video shows a single well-tracked trial of a rat performing a correct response during the 5CSRTT. Tracking of the house light (red marker), tail base (green marker) and head (blue marker) are indicated in the video. Neural network #2 described in **Figure 10A** was used for tracking. Note how the markers follow the movements of the animal accurately. Also refer to **Figure 10C,E** for the plotted path and movement speed for the head tracking in this video clip.

Video 3: Example of poorly tracked animal. The video shows a single poorly tracked trial of a rat performing a correct response during the 5CSRTT. Tracking of the house light (red marker), tail base (green marker) and head (blue marker) are indicated in the video. Neural network #1 described in **Figure 10A** was used for tracking. The video clip is the same as the one used in **Video 2**. Note that the marker for the head is not reliably placed on top of the rat's head. Also refer to **Figure 10D,E** for the plotted path and movement speed for the head tracking in this video clip.

Video 4: Example of movements made during a 5CSRTT trial with a response. The video shows a single well-tracked trial of a rat performing a correct response during the 5-CSRTT. Tracking of the house light (red marker), tail base (green marker) and head (blue marker) are indicated in the video. Note how the rat at first is positioned in clear proximity to the pellet receptacle (left wall, **Figure 7A**) and then moves over to focus its attention on the row of nose poke openings.

Video 5: Example of a typical omission trial during the 5CSRTT. The video shows a single well-tracked trial of a rat performing a typical omission during the 5CSRTT. Tracking of the house light (red marker), tail base (green marker) and head (blue marker) are indicated in the video. Note how the rat maintains its position around the pellet receptacle (left wall, **Figure 7A**) and chamber center, rather than turning around to face the nose poke openings (right wall, **Figure 7A**). The displayed behavior and cause of the omission can be argued to reflect low interest in performing the test.

Video 5: Example of an atypical omission trial during the 5CSRTT. The video shows a single well-tracked trial of a rat performing an atypical omission during the 5CSRTT. Tracking of the house light (red marker), tail base (green marker) and head (blue marker) are indicated in the video. Note how the rat positions itself towards the nose poke openings along the right wall of the chamber (**Figure 7A**). This can be argued to indicate that the animal is interested in performing the test. However, the rat faces away from the cued opening (central position) when the cue is presented (5 s into the clip). In contrast to the omission displayed in **Video 4**, the one seen here is likely related to sub-optimal visuospatial attention processes.

Video 6: Example of an animal maintaining a focused central position during an ITI of the 5CSRTT. The video shows a single well-tracked trial of a rat performing a correct response on a trial of the 5CSRTT. Note how the rat maintains a central position during the ITI, keeping its head steady in proximity to the central nose poke opening along the chambers right wall (**Figure 7A**). Tracking of the house light (red marker), tail base (green marker) and head (blue marker) are indicated in the video.

Video 7: Example of an animal displaying a search-like attentional strategy during an ITI of the 5CSRTT. The video shows a single well-tracked trial of a rat performing a correct response on a trial of the 5CSRTT. Note how the rat frequently repositions its head to face different nose poke openings along the right wall of the chamber (**Figure 7A**). Tracking of the house light (red marker), tail base (green marker) and head (blue marker) are indicated in the video.

Video 8: Rats constantly move their heads between the different openings in a search-like manner.

DISCUSSION:

This protocol describes how to build an inexpensive and flexible video camera that can be used to record videos from operant conditioning chambers and other behavioral test setups. It further demonstrates how to use DeepLabCut to track a strong light signal within these videos, and how that can be used to aid in identifying brief video segments of interest in video files that cover full test sessions. Finally, it describes how to use the tracking of a rat’s head to complement the analysis of behaviors during operant conditioning tests.

The protocol presents an alternative to commercially available video recording solutions for operant conditioning chambers. As noted, the major benefit of these is that they integrate with the operant chambers, enabling video recordings of specific events. The approach to identifying video segments of interest described in this protocol is more laborious and time-consuming compared to using a fully integrated system to record specific events. It is, however, considerably cheaper (a recent cost estimate for video monitoring equipment for 6 operant chambers was set to approximately 13,000 USD. In comparison, constructing six of the cameras listed here would cost about 720 USD). In addition, the cameras can be used for multiple other behavioral test setups. When working with the camera, it is important to be mindful of the areas of exposed electronics (the back of the camera component as well as the IR LED component), so that they do not come into contact with fluids. In addition, the ribbon cable attaching the camera module to the microcomputer and cables connecting the LEDs and switches to the GPIO pins may come loose if the camera is frequently moved around. Thus, adjusting the design of the camera case may be beneficial for some applications.

The use of DeepLabCut to identify video segments of interest and track animal movements offers a complement and/or alternative to manual video analysis. While the former does not invalidate the latter, we have found that it provides a convenient way of analyzing movements and behaviors inside operant chambers. In particular, it provides positional data of the animal, which contains more detailed information than what is typically extracted via manual scoring (i.e., actual coordinates compared to qualitative positional information such as “in front of”, “next to” etc.).

When selecting a protocol step indicator, it is important to choose one that consistently indicates a given step of the behavioral protocol, and that is unlikely to be blocked by the animal. If the latter is problematic, one may consider placing a lamp outside the operant chamber and film it through the chamber walls. Many operant conditioning chambers are modular and allow users to freely move lights, sensors and other components around. It should be noted that there are other software packages that also allow users to train neural networks in recognizing and tracking user-defined objects in videos[24-26]. These can likely be used as alternatives to DeepLabCut in the current protocol.

The protocol describes how to track the central part of a rats’ head in order to measure movements inside the operant chambers. As DeepLabCut offers full freedom in selecting body parts or objects of interest, this can with convenience be modified to fit study-specific interests. A natural extension of the tracking described herein is to also track the position of the rats’ ears and nose, to better judge not only head position but also orientation. The representative data shown here was recoded with Long Evans rats. These rats display considerable inter-individual variation in their pigmentation pattern, particularly towards their tail base. This may result in some difficulties applying a single trained neural network for the tracking of different individuals. To limit these issues, it is best to include video frames from all animals of interest in the training set for the network. The black head of the Long Evans rat provides a reasonably strong contrast against the metal surfaces of the chamber used here. Thus, obtaining accurate tracking of their heads likely requires less effort than with albino strains. The most critical step of obtaining accurate tracking with DeepLabCut or comparable software packages is to select a good number of diverse video frames for the training of the neural network. As such, if tracking of an object of interest is deemed to be sub-optimal, increasing the set of training frames should always be the first step towards improving the results.

ACKNOWLEDGMENTS:

The authors wish to acknowledge the Wenner-Gren foundation, Parkinsonfonden, the Åhlén foundation and the foundation Blanceflor Boncompagni Ludovisi, née Bildt.

DISCLOSURES:

While materials and resources from The Raspberry Pi foundation has been used and cited in this manuscript, the foundation was not actively involved in the preparation or use of equipment and data in this manuscript. The same is true for Pi-Supply. The authors have nothing to disclose.

REFERENCES:

1. Pritchett, K., Mulder, G.B. Operant conditioning. *Contemporary Topics in Laboratory Animal Science.* **43** (4), (2004).

2. Clemensson, E.K.H., Novati, A., Clemensson, L.E., Riess, O., Nguyen, H.P. The BACHD rat model of Huntington disease shows slowed learning in a Go/No-Go-like test of visual discrimination. *Behavioural Brain Research* **359**, 116-126, 10.01016/j.bbr.2018.10.036 (2019).

3. Asinof, S.K., Paine, T.A. The 5-choice serial reaction time task: a task of attention and impulse control for rodents. *Journal of Visualized Experiments.* (90), e51574, 10.3791/51574 (2014).

4. Coulbourn instruments, Graphic State: Graphic State 4 user's manual, 12-17 & 28, Holliston, Massachusetts.

5. Med Associates Inc, Med-PC IV: Med-PC IV programmer's manual, 21-24 & 39-44, St. Albans, Vermont.

6. Clemensson, E.K.H, Clemensson, L.E., Riess, O., Nguyen, H.P. The BACHD rat model of Huntingon disease shows signs of fronto-striatal dysfunction in two operant conditioning tests of short-term memory. *PloS One*. **12** (1), 10.1371/journal.pone.0169051 (2017).

7. Herremans, A.H.J, Hijzen, T.H., Welborn, P.F.E., Olivier, B., Slangen, J.L. Effect of infusion of cholinergic drugs into the prefrontal cortex area on delayed matching to position performance in the rat. *Brain Research*. **711** (1-2), 102-111, 10.1016/0006-8993(95)01404-7 (1996).

8. Chudasama, Y., Muir, J.L. A behavioral analysis of the delayed non-matching to position task: the effects of scopolamine, lesions of the fornix and of the prelimbic region on mediating behaviours by rats. *Psychopharmacology.* **134** (1), 73-82, 10.1007/s002130050427 (1997).

9. Talpos, J.C., McTighe, S.M., Dias, R., Saksida, L.M., Bussey, T.J. Trial-unique, delayed nonmatching-to-location (TUNL): A novel, highly hippocampus-dependent automated touchscreen test of location memory and pattern separation. *Neurobiology of Learning and Memory*. **94** (3), 341-325, 10.1016/j.nlm.2010.07.006 (2010).

10. Rayburn-Reeves, R.M., Moore, M.K., Smith, T.E., Crafton, D.A., Marden, K.L. Spatial midsession reversal learning in rats: Effects of egocentric cue use and memory. *Behavioural Processes*. **152**, 10-17, 10.1016/j.beproc.2018.03.005 (2018).

11. Gallo, A., Duchatelle, E., Elkhessaimi, A., Le Pape, G., Desportes, J. Topographic analysis of the rat's behavior in the Skinner box. *Behavioural Processes*. **33** (3), 318-328, 10.1016/0376-6357(94)00032-C (1995).

12. Iversen, I.H. Response-initiated imaging of operant behavior using a digital camera. *Journal of the Experimental Analysis of Behavior.* **77** (3), 283-300, 10.1901/jeab.2002.77-283 (2002).

13. Med Associates Inc, Video monitor: Video monitor SOF-842 user's manual, 26-30, St. Albans, Vermont (2004).

14. Coulbourn Instruments, 2020, https://www.coulbourn.com/product_p/h39-16.htm.

15. Mathis, A., *et al*. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience.* **21** (9), 1281-1289, 10.1038/s41593-018-0209-y (2018).

16. Nath, T., Mathis, A., Chen, A.C., Patel, A., Bethge, M., Mathis, M.W. Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nature Protocols*. **14** (7), 2152-2176, 10.1038/s41596-019-0176-0 (2019).

17. Bari, A., Dalley, J.W., Robbins, T.W. The application of the 5-chopice serial reaction time task for the assessment of visual attentional processes and impulse control in rats. *Nature Protocols.* **3**(5=, 759-767, 10.1038/nprot.2008.41 (2008).

18. Raspberry Pi foundation, 2020, https://thepi.io/how-to-install-raspbian-on-the-raspberry-pi/

19. Pi-supply, 2018, https://learn.pi-supply.com/make/bright-pi-quickstart-faq/

20. Python, 2020, https://wiki.python.org/moin/BeginnersGuide/ NonProgrammers

21. MathWorks, 2020, https://mathworks. Com/academia/highschool/

22. R, 2020, https://cran.r-project.org/manuals.html

23. Liu, Y., Tian, C., Huang, Y. Critical assessment of correction methods for fisheye lens distortion. The international archives of the photogrammetry, remote sensing and spatial information sciences. 10.5194/isprsarchives-XLI-B1-221-2016 (2016).

24. Pereira, T.D., *et al.* Fast animal pose estimation using deep neural networks. *Nature Methods*. **16** (1), 117-125, 10.1038/s41592-018-0234-5 (2019).

25. Graving, J.M., *et al.* DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *Elife.* **8** (e47994), 10.7754/eLife.47994 (2019).

26. Geuther, B.Q, *et al.* Robust mouse tracking in complex environments using neural networks. *Communications Biology.* **2** (124), 10.1038/s42003-019-0362 (2019).

Figure 1

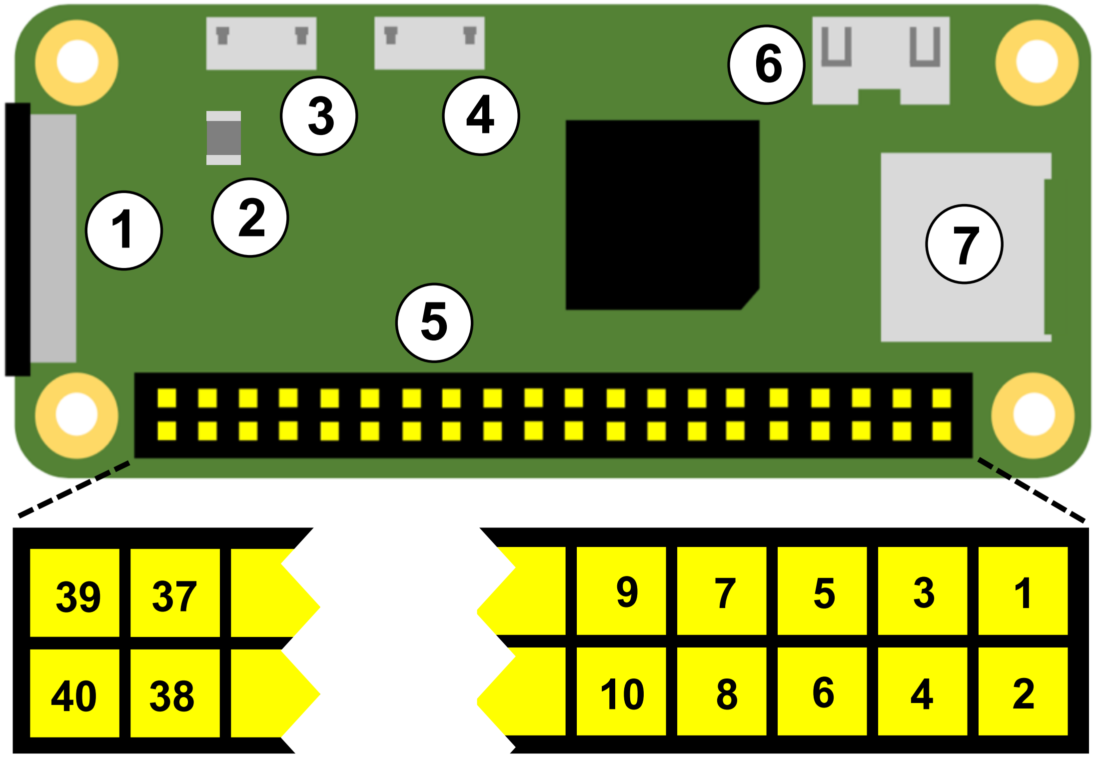


Figure 2 Click here to access/download;Figure;Figure_2.pdf

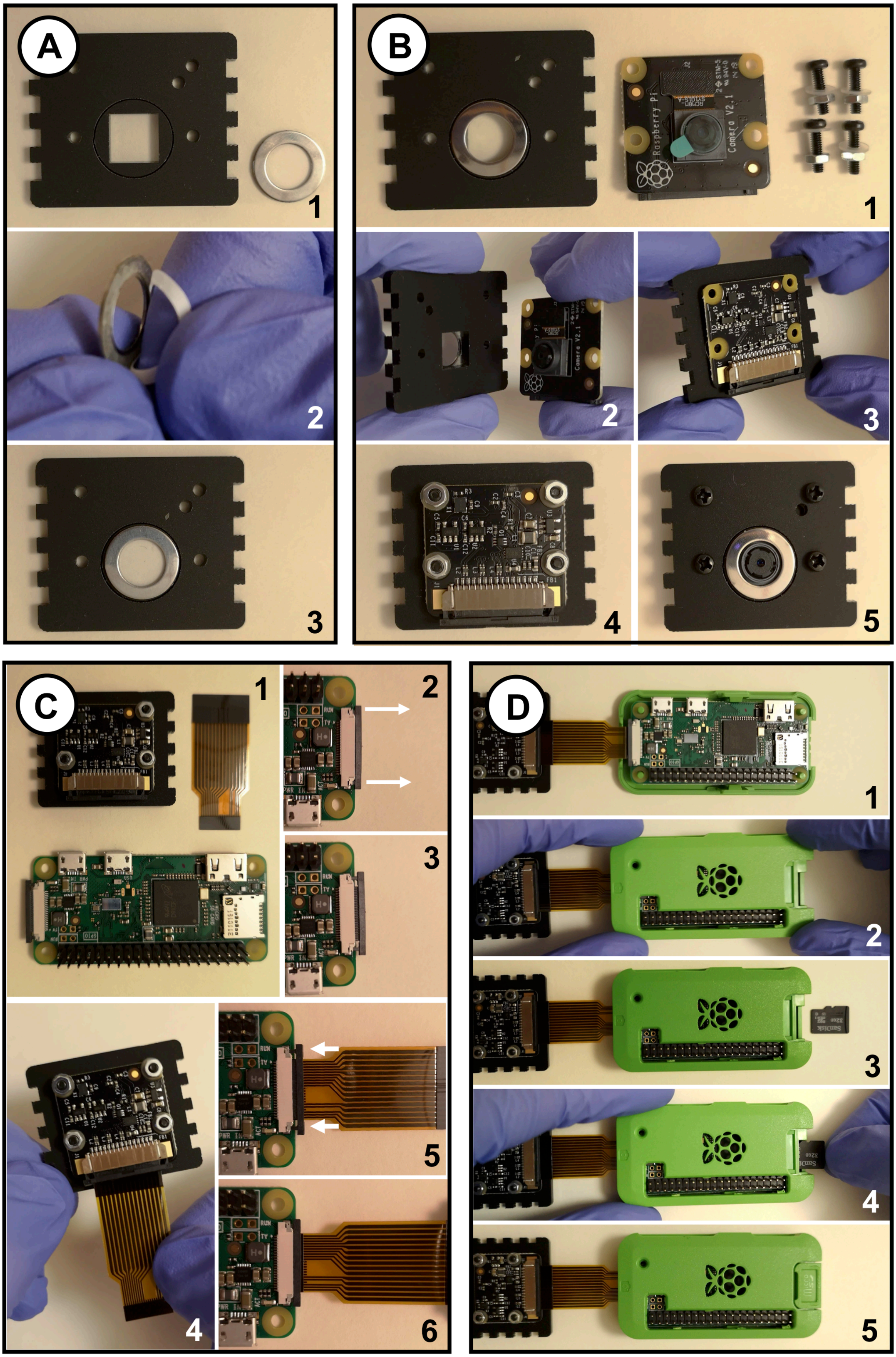


Figure 3

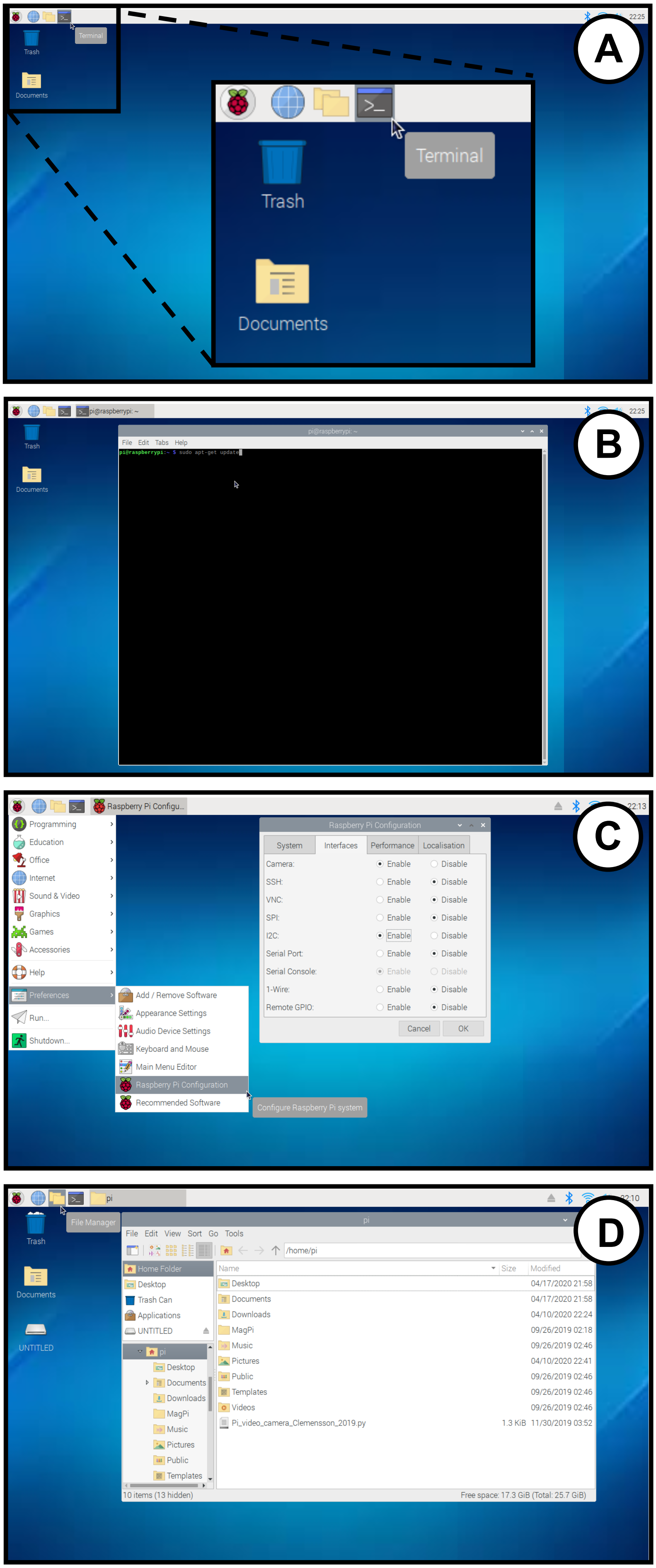

A
Terminal
Trash
Documents
B
pi@raspberrypi: ~
File Edit Tabs Help
C
Raspberry Pi Configuration
System
Interfaces
Performance
Localisation
Camera:
SSH:
VNC:
SPI:
I2C:
Serial Port:
Serial Console:
1-Wire:
Remote GPIO:
Enable
Disable
Cancel
OK
Programming
Education
Office
Internet
Sound & Video
Graphics
Games
Accessories
Help
Preferences
Run...
Shutdown...
Add / Remove Software
Appearance Settings
Audio Device Settings
Keyboard and Mouse
Main Menu Editor
Raspberry Pi Configuration
Recommended Software
Configure Raspberry Pi system
D
File Manager
UNTITLED
File Edit View Sort Go Tools
/home/pi
Home Folder
Desktop
Trash Can
Applications
Name
Size
Modified
Desktop 04/17/2020 21:58
Documents 04/17/2020 21:58
Downloads 04/10/2020 22:24
MagPi 09/26/2019 02:18
Music 09/26/2019 02:46
Pictures 04/10/2020 22:41
Public 09/26/2019 02:46
Templates 09/26/2019 02:46
Videos 09/26/2019 02:46
Pi_video_camera_Clemensson_2019.py 1.3 KiB 11/30/2019 03:52
10 items (13 hidden)
Free space: 17.3 GiB (Total: 25.7 GiB)

Figure 4

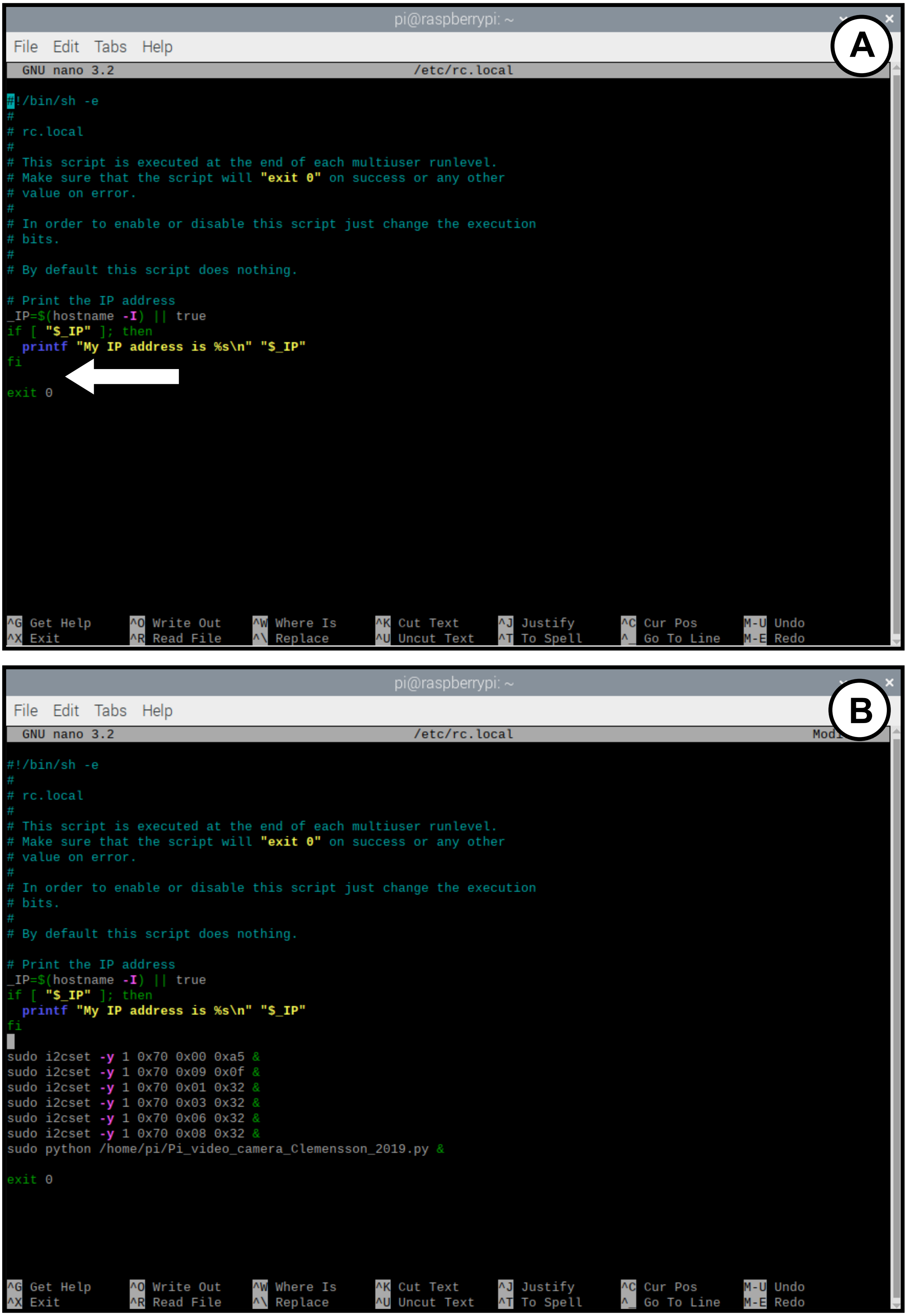

A
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 3.2 /etc/rc.local
#!/bin/sh -e
#
rc.local
#
This script is executed at the end of each multiuser runlevel.
Make sure that the script will "exit 0" on success or any other
value on error.
#
In order to enable or disable this script just change the execution
bits.
#
# By default this script does nothing.
# Print the IP address
_IP=$(hostname -I) || true
if ["$_IP"]; then
printf "My IP address is %s\n" "$_IP"
fi
exit 0
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line M-E Redo
B
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 3.2 /etc/rc.local Mod
#!/bin/sh -e
#
rc.local
#
This script is executed at the end of each multiuser runlevel.
Make sure that the script will "exit 0" on success or any other
value on error.
#
In order to enable or disable this script just change the execution
bits.
#
# By default this script does nothing.
# Print the IP address
_IP=$(hostname -I) || true
if ["$_IP"]; then
printf "My IP address is %s\n" "$_IP"
fi
sudo i2cset -y 1 0x70 0x00 0xa5 &
sudo i2cset -y 1 0x70 0x09 0x0f &
sudo i2cset -y 1 0x70 0x01 0x32 &
sudo i2cset -y 1 0x70 0x03 0x32 &
sudo i2cset -y 1 0x70 0x06 0x32 &
sudo i2cset -y 1 0x70 0x08 0x32 &
sudo python /home/pi/Pi_video_camera_Clemensson_2019.py &
exit 0
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line M-E Redo

Figure 5

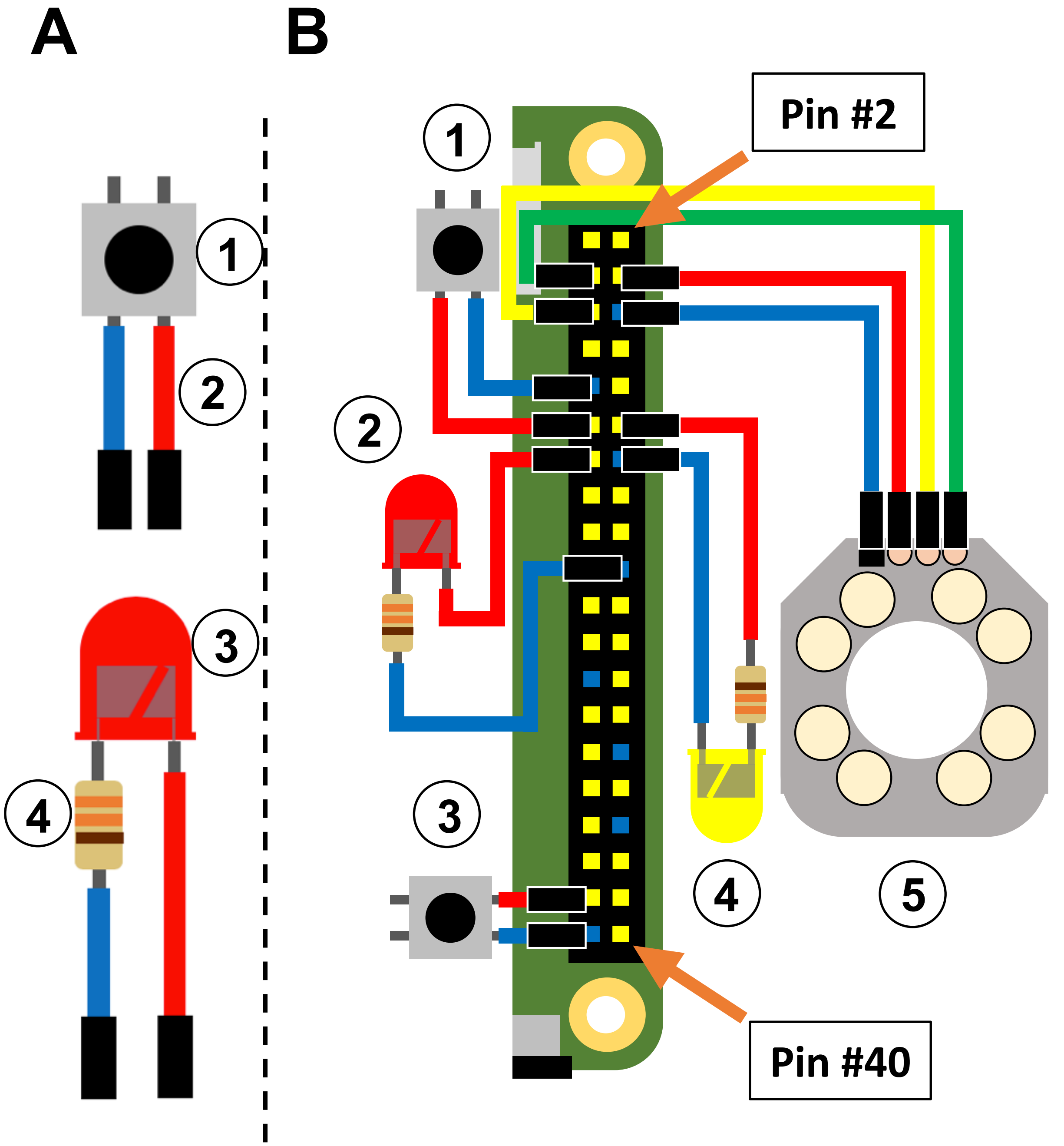


Figure 6

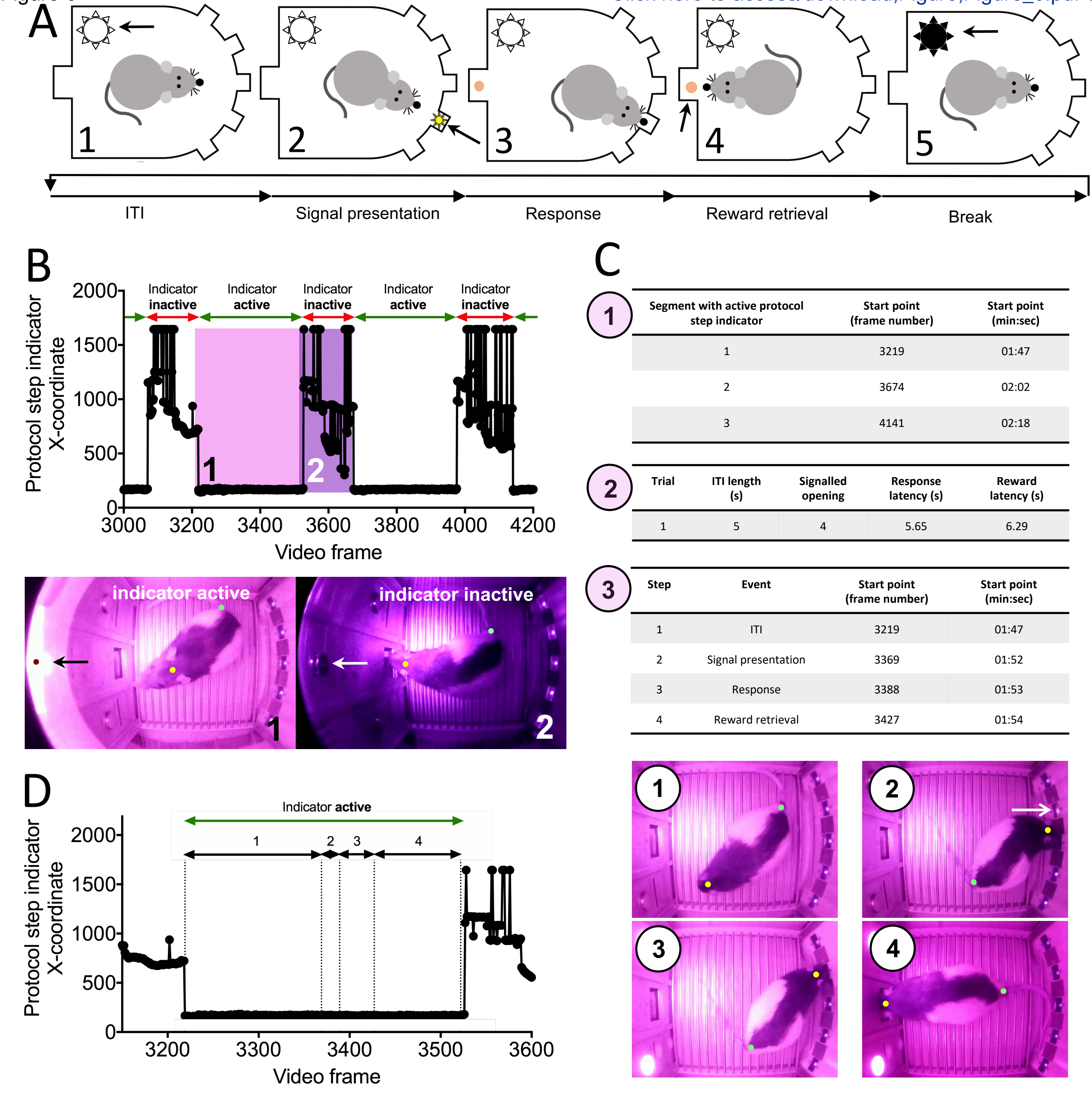

A
1
2
3
4
5
ITI
Signal presentation
Response
Reward retrieval
Break
B
Protocol step indicator X-coordinate
Video frame
Indicator inactive
Indicator active
Indicator inactive
Indicator active
Indicator inactive
0
500
1000
1500
2000
3000
3200
3400
3600
3800
4000
4200
1
2
indicator active
indicator inactive
C
1
Segment with active protocol step indicator | Start point (frame number) | Start point (min:sec)
1 | 3219 | 01:47
2 | 3674 | 02:02
3 | 4141 | 02:18
2
Trial | ITI length (s) | Signalled opening | Response latency (s) | Reward latency (s)
1 | 5 | 4 | 5.65 | 6.29
3
Step | Event | Start point (frame number) | Start point (min:sec)
1 | ITI | 3219 | 01:47
2 | Signal presentation | 3369 | 01:52
3 | Response | 3388 | 01:53
4 | Reward retrieval | 3427 | 01:54
D
Indicator active
1
2
3
4
Protocol step indicator X-coordinate
Video frame
0
500
1000
1500
2000
3200
3300
3400
3500
3600

Figure 7

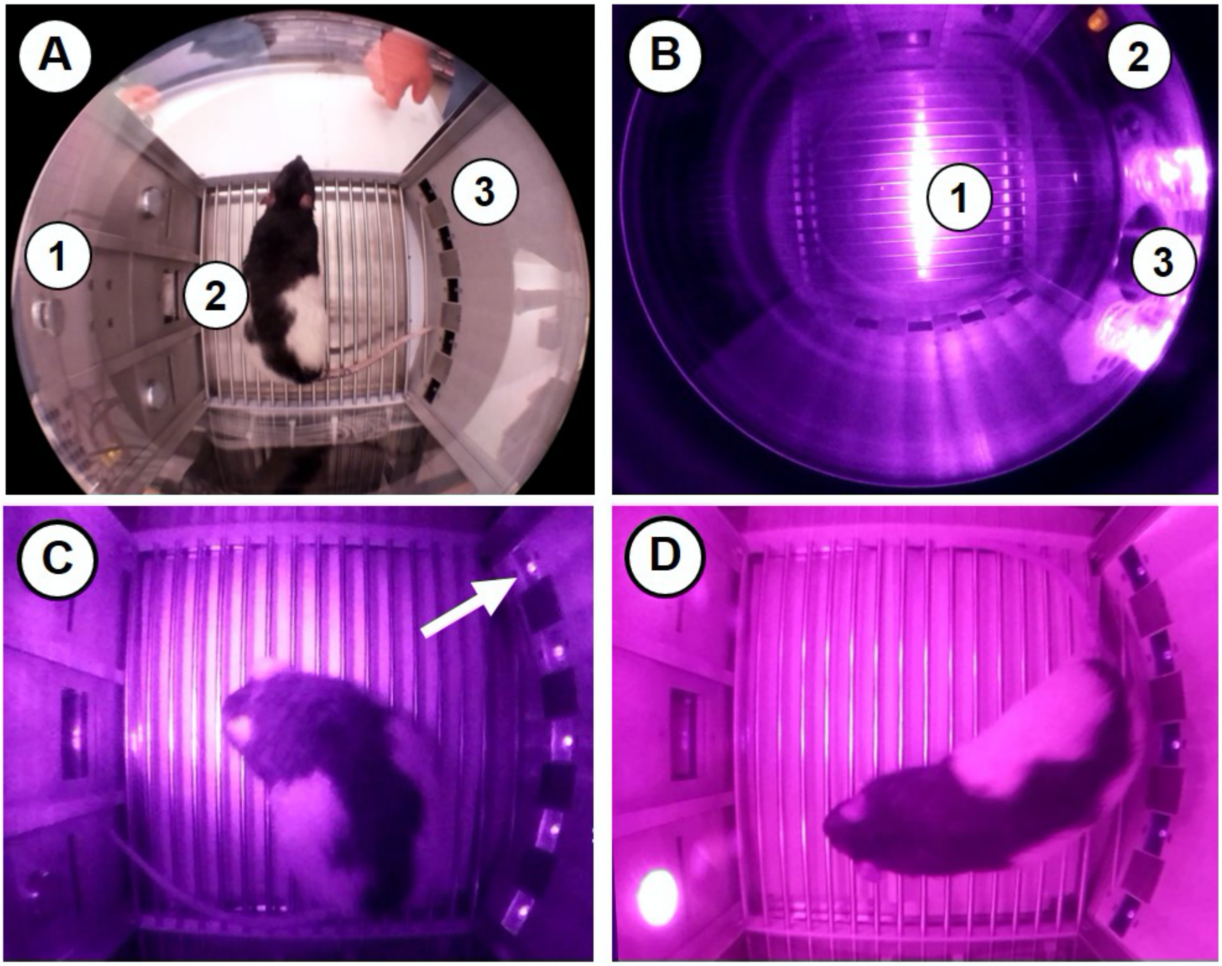


Figure 8

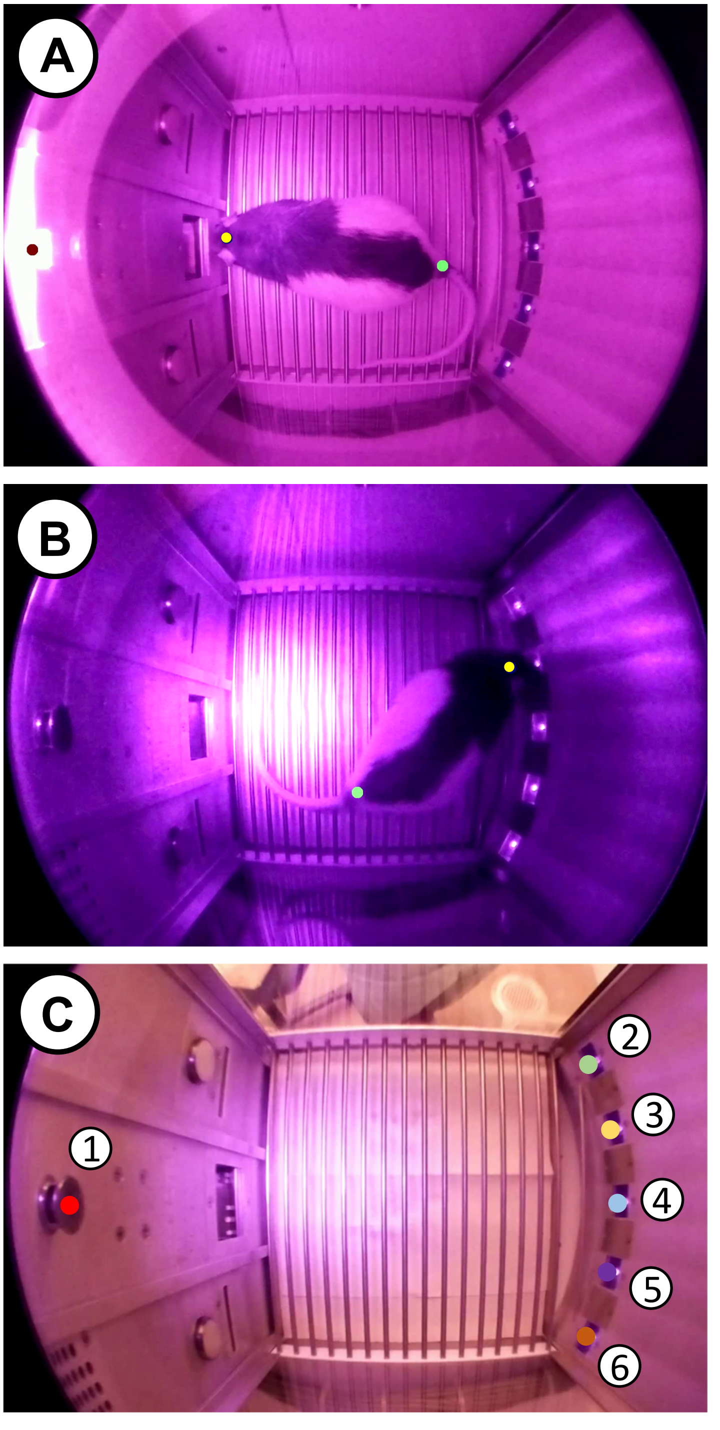


Figure 9

Click here to access/download;Figure;Figure_9.pdf

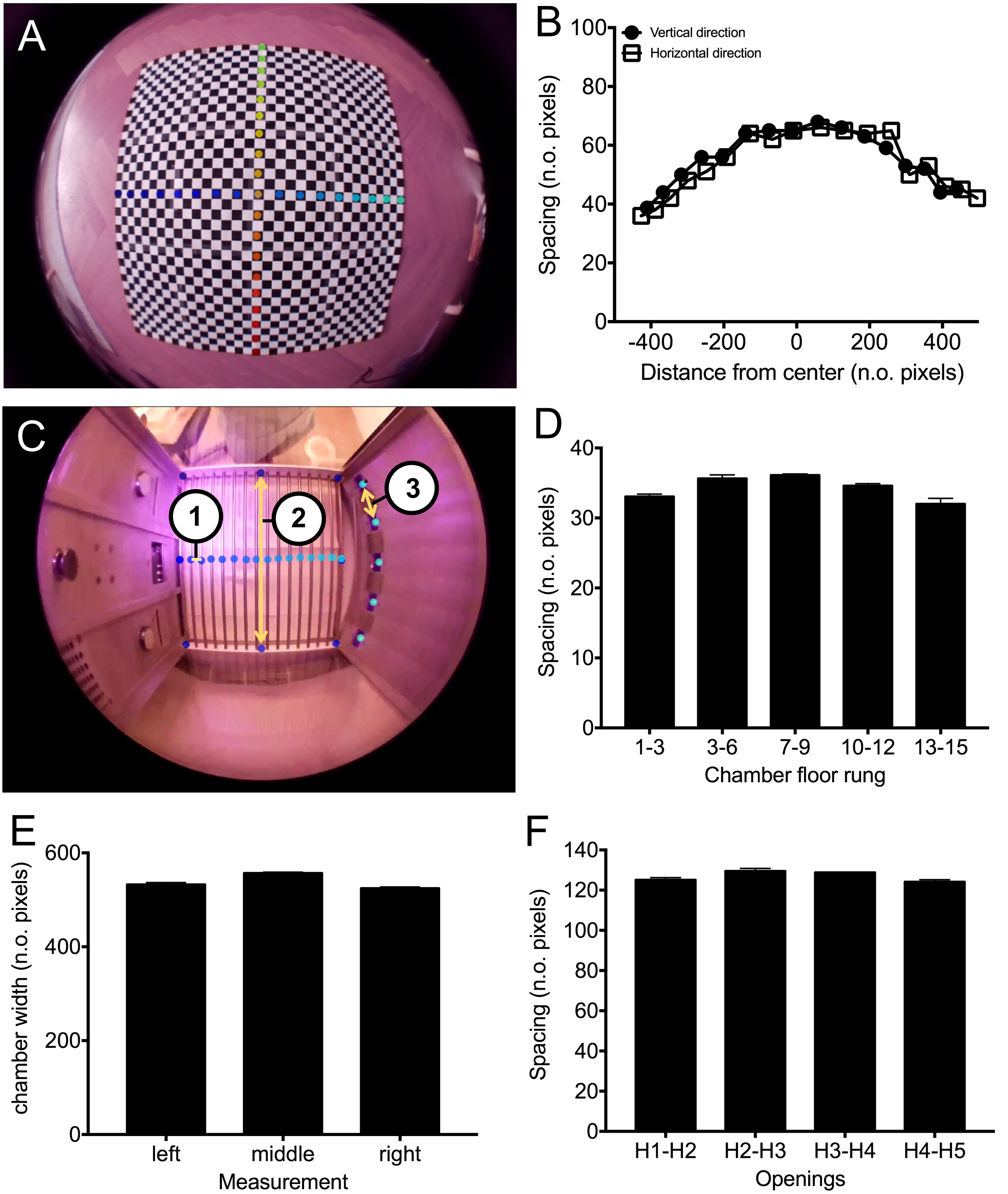
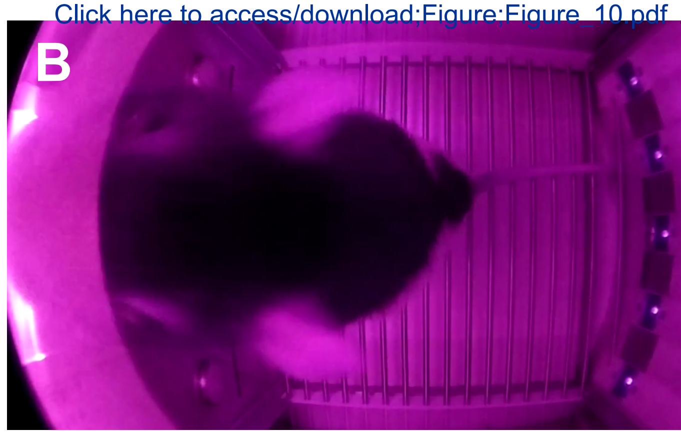


Figure 10

A

Network	#1 (Poor)	#2 (Good)
Frames in training set	240	1 100
Training iterations	900 000	500 000
DLC test error (pixels)	5.7	7.7
DLC training loss (pixels)	< 0.01	< 0.01
Measured accuracy	62.4%	90.7%



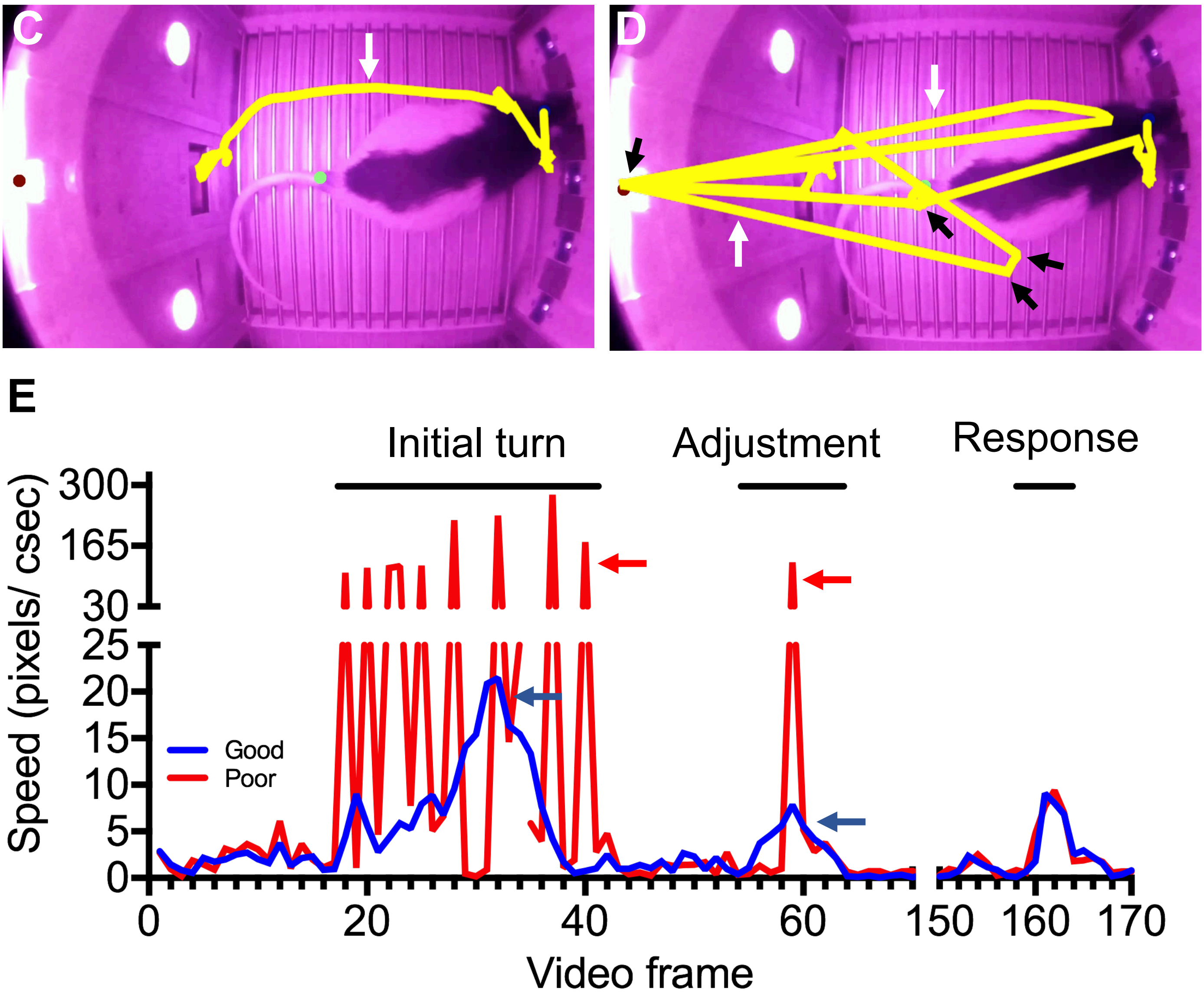
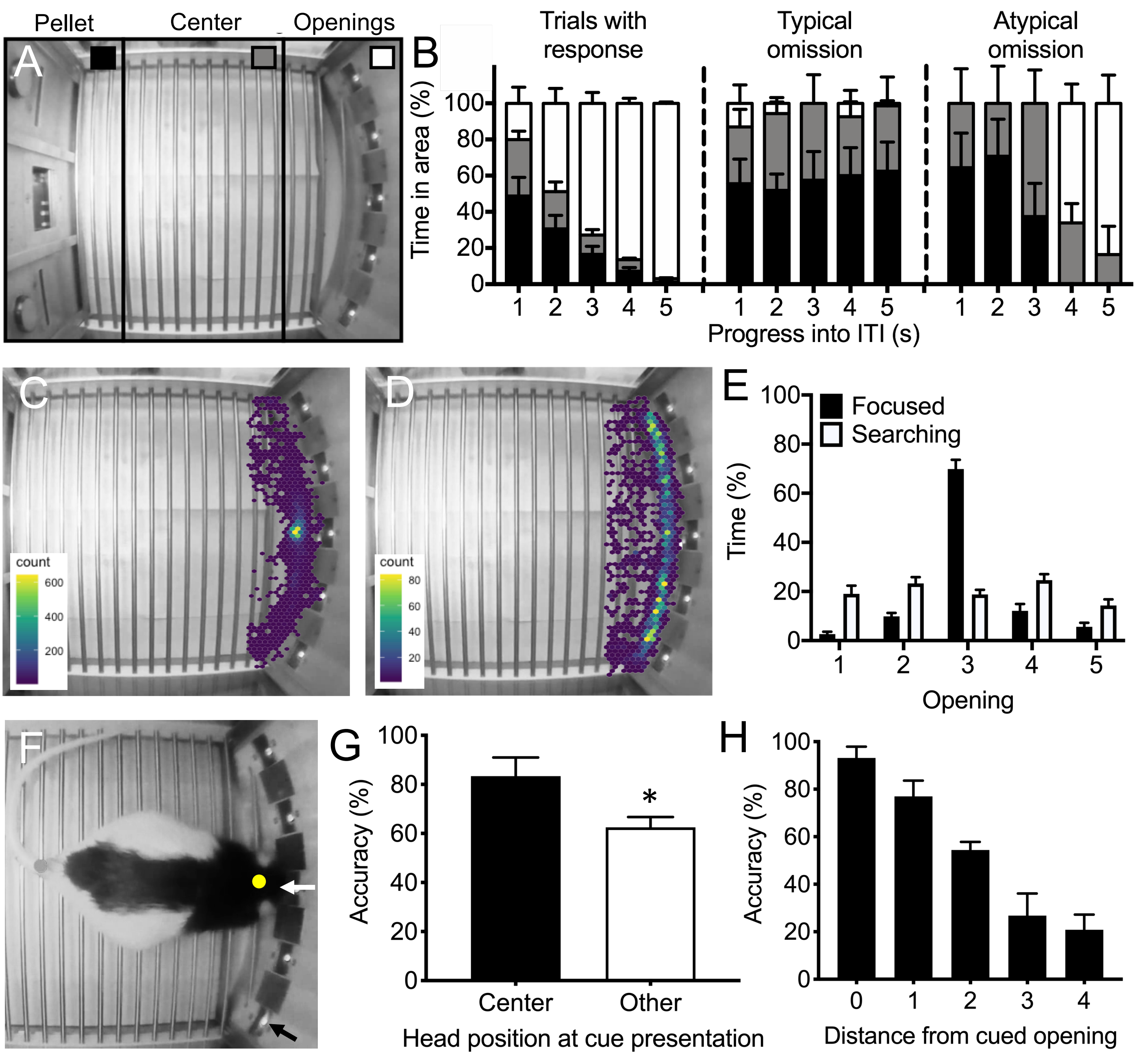


Figure 11


Pellet
Center
Openings
A
B
Trials with response
Typical omission
Atypical omission
Time in area (%)
Progress into ITI (s)
C
count
600
400
200
D
count
80
60
40
20
E
Focused
Searching
Time (%)
Opening
F
G
Accuracy (%)
Center
Other
Head position at cue presentation
H
Accuracy (%)
Distance from cued opening

Video 1

Click here to access/download
Video or Animated Figure
Video_1_Correct_response_montage.avi

Video 2

Click here to access/download
Video or Animated Figure
Video_2_Well_tracked.mp4

Video 3

Click here to access/download
Video or Animated Figure
Video_3_Poorly_tracked.mp4

Video 4

Click here to access/download
Video or Animated Figure
Video_4_Correct_response_example.mp4

Video 5

Click here to access/download
Video or Animated Figure
Video_5_Typical_omission_example.mp4

Video 6

Click here to access/download
Video or Animated Figure
Video_6_Atypical_omission_example.mp4

Video 7

Click here to access/download
Video or Animated Figure
Video_7_Focused_example.mp4

Video 8

Click here to access/download
Video or Animated Figure
Video_8_Search_example.mp4

Name of Material/ Equipment	Company	Catalog Number	Comments/Description
32 Gb micro SD card with New Our Of Box Software (NOOBS)	The Pi hut (https://thpihut.com)	32GB	
330-Ohm resistor	The Pi hut (https://thpihut.com)	100287	This article is for a package with mixed resistors, where 330-ohm resistors are included.
Camera module (Raspberry Pi NoIR camera v.2)	The Pi hut (https://thpihut.com)	100004	
Camera ribbon cable (Raspberry Pi Zero camera cable stub)	The Pi hut (https://thpihut.com)	MMP-1294	This is only needed if a Raspberry Pi zero is used. If another Raspberry Pi board is used, a suitable camera ribbon cable accompanies the camera
Colored LEDs	The Pi hut (https://thpihut.com)	ADA4203	This article is for a package with mixed colors of LEDs. Any color can be used.
Female-Female jumper cables	The Pi hut (https://thpihut.com)	ADA266	
IR LED module (Bright Pi)	Pi Supply (https://uk.pi-supply.com)	PIS-0027	
microcomputer motherboard (Raspberry Pi Zero board with presoldered headers)	The Pi hut (https://thpihut.com)	102373	Other Raspberry Pi boards can also be used, although the method for automatically starting the Python script only works with Raspberry Pi zero. If using other models, the python script needs to be started manually.
Push button switch	The Pi hut (https://thpihut.com)	ADA367	
Raspberry Pi power supply cable	The Pi hut (https://thpihut.com)	102032	
Raspberry Pi Zero case	The Pi hut (https://thpihut.com)	102118	
Raspberry Pi, Mod my pi, camera stand with magnetic fish eye lens	The Pi hut (https://thpihut.com)	MMP-0310-KIT	

Dear Editor and reviewers,

We greatly appreciate the constructive feedback given by all of you. We have made substantial changes to both text and figures to adhere to them and we believe that the manuscript has improved greatly as a result. We hope you find it to your liking.

All references to specific lines made in our responses relate to the numbering in the clean copy of the revised manuscript.

Best regards,

Erik Clemensson

Line-by-line responses:

Editorial Comments:

• Add email addresses for coauthors on lines 14-18

Response: This has been done, please see lines 15 to 18.

• Adjust the highlighted text so that it fits the limit for the video recording

Response: This has been addressed

• List webpage citations in reference list rather than in text

Response: This has been done, please see references 14, 18-22.

• Number your supplementary files, and reference then accurately in the text

Response: This has been done. Please see lines 152, 275, 388, 445.

• Define GPIO

Response: This has been done. Please see line 187 & 529-530.

• Update numbering of protocol points following editorial changes

Response: This has been done.

• Define ITI

Response: This was already defined on line 498.

• Line 581 “8C”, Fig. 8?

Response: Yes, this has been addressed.

• Specify button clicks, menu selections, etc. for numerous points relating to the use of DeepLabCut.

Response: The initial and revised version of the steps relating to the use of DeepLabCut were written in accordance with how other authors have described this process in JoVE protocols (Bova et al 2019). In that protocol, the authors describe how to build an automatic pellet reaching chamber for rats, how to train rats to reach for pellets in it, how to track the movements of individual knuckles on the rats’ forepaw during the reaching movements, and how to analyse these movement patterns. The authors give a description of how to use DeepLabCut for this purpose that is very similar to the one written in the earlier versions of our protocol, in terms of the amount of details given.

While providing a more in-depth protocol of how to work with DeepLabCut is appealing, we don't consider it particularly practical or suitable to do in this protocol. This is because each of the steps involve multiple button clicks, menu selections and descriptions of either several different aspects of the GUI, or the specific Python commands being used. If you are interested, we have pasted a more thorough descriptions detailing the button clicks and menu selections of the DeepLabCut GUI at the end of this rebuttal letter. The text is roughly 2.5 pages long, meaning that if the different detailed steps of this process are to be included in the video protocol, there would not be space for much else. Further, the steps and specific button clicks are well-described in other recent publications (Ref. 16). Thus, it makes more sense that this protocol primarily refers to these while giving some tips that relate more specifically to how DeepLabCut is applied in this particular protocol (such as training a network to specifically recognize the position of a light when it is shining, but not when it is switched off).

We have rewritten parts of the protocol text in an attempt at making the general process more understandable (lines 279-285) and specifying the specific functions of DeepLabCut that are used in the different steps (e.g. "Use DeepLabCut's frame grabbing function" (line 289), Use DeepLabCut's Labeling Toolbox (line 305). As you did not ask us to clarify anything in point 4.7.1 (previously 6.1.), which was written in a similar way, we assume that this level of detail is sufficient for aiding the narrative.

However, if this does not provide a satisfactory solution for you we can definitely add the longer description shown at the end of the rebuttal letter, with accompanying figures, to the protocol. It is, however, our belief that that is unnecessary.

Note: The points noted above were adapted from comments made in the manuscript file, and placed here with corresponding responses for convenience.

• Please take this opportunity to thoroughly proofread the manuscript to ensure that there are no spelling or grammatical errors.

Response: This has been checked.

• **Protocol Language:** Please ensure that all text in the protocol section is written in the imperative voice/tense as if you are telling someone how to do the technique (i.e. "Do this", "Measure that" etc.) Any text that cannot be written in the imperative tense may be added as a "Note", however, notes should be used sparingly and actions should be described in the imperative tense wherever possible. Examples NOT in imperative voice: 1.2, 1.17, 1.19, 2.2.1

Response: The noted points have been addressed, and additional text has been worked over. Please let us know if we overlooked any points.

• **Discussion:** JoVE articles are focused on the methods and the protocol, thus the discussion should be similarly focused. Please ensure that the discussion covers the following in detail and in paragraph form (3-6 paragraphs): 1) modifications and troubleshooting, 2) limitations of the technique, 3) significance with respect to existing methods, 4) future applications and 5) critical steps within the protocol.

Response: We have looked through the discussion and believe that it adheres well with these guidelines. If there are concerns, please specify which aspect of the discussion are not in accordance with the guidelines.

• **Figure/Table Legends:**
1) Label the video as video 1, include a legend, and include a call out to the video in text.

Response: This has been addressed for all videos included in the manuscript.

• **References:**Please spell out journal titles.

Response: This has been addressed.

• **Commercial Language:**JoVE is unable to publish manuscripts containing commercial sounding language, including trademark or registered trademark symbols (TM/R) and the mention of company brand names before an instrument or reagent. Examples of commercial sounding language in your manuscript are Raspberry Pi-, Raspberry Pi NoIR v.2,
1) Please use MS Word's find function (Ctrl+F), to locate and replace all commercial sounding language in your manuscript with generic names that are not company-specific. All commercial products should be sufficiently referenced in the table of materials/reagents. You may use the generic term followed by "(see table of materials)" to draw the readers' attention to specific commercial names.

Response: This has been addressed.

• **Table of Materials:**Please sort items in alphabetical order.

Response: This has been addressed.

• If your figures and tables are original and not published previously or you have already obtained figure permissions, please ignore this comment. If you are re-using figures from a previous publication, you must obtain explicit permission to re-use the figure from the previous publisher (this can be in the form of a letter from an editor or a link to the editorial policies that allows you to re-publish the figure). Please upload the text of the re-print permission (may be copied and pasted from an email/website) as a Word document to the Editorial Manager site in the "Supplemental files (as requested by JoVE)" section. Please also cite the figure appropriately in the figure legend, i.e. "This figure has been modified from [citation]."

Response: All figures are original to this publication.

Comments from Peer-Reviewers:

Reviewer #1:
Manuscript Summary:
This manuscript describes instructions for building a small Raspberry Pi-based camera, with all accompanying software, and a generalized method for using DeepLabCut to track spatial features of rodent location in operant behavioral chambers. Additionally, the authors describe

how to collect data for two new metrics (head location and velocity). The technical design and description of the Raspberry Pi-based camera set-up is robust with wide implications across the behavioral field;with minor edits, the instructions portion of the manuscript are detailed enough for most researchers to successfully build the proposed camera and train DeepLabCut.

Response: Thank you for this positive feedback. We have revised the previous figures (Fig. 1 and 5), added additional figures (Fig. 2, 3, 4) and improved the protocol text to give a more detailed description of how to prepare the camera. We have also had two people with no prior experience in using the equipment or programming carry out the protocol successfully.

However, the DeepLabCut and behavior metric portion appears to echo the original DeepLabCut publication without new insight or validation.

Response: To some degree it does echo what was displayed in the original publication. However, tracking the condition of a light inside the operant conditioning chamber as a way to identify short segments of interest in videos that cover entire test sessions (often 30-minutes or longer) represents something that to our knowledge has not been shown previously. This is an important feature for how we are currently using DeepLabCut to improve our workflow for video analysis. Moreover, the protocol does not simply describe that DeepLabCut can be used to track positions (such as in the original publication), but also that knowledge of an animal's position at certain points during cognitive tasks enables the analysis of behavioural aspects that cannot be obtained through the data logs produced by the systems.

We have rewritten large parts of the introduction, preliminary data and discussion to put clearer emphasis on these aspects.

Additionally, much of the manuscript is missing citational support and the manuscript rhetoric needs significant editing.

Response: Please see detailed comments below. Several claims that were made too strongly before have now been removed or edited. If the rhetoric needs additional changes, please specify which sentences /sections are of concern.

Overall, this work is worth publication, though will need to add methodological validation and considerable revision.

Response: Thank you. We hope that our revisions have sufficiently addressed your concerns.

Major Concerns:
-Overall, the manuscript uses lengthy prose and too many vague terms (ex: line 59-60 "…what the animal did during specific steps of a given protocol", line 107-108 "The animal is then allowed to perform a response…" and line 67-69 "But it can also allow a more thorough understanding of what might be perceived as a simple behavior, such as the act of just responding to operanda"). This leads to some ambiguities in interpretation (ex: the obscurity relating to Figure 4 representing one or two trial/recordings/events - described later)

Response: We have gone through the text as a whole and made multiple changes to sentence structure, and also addressed the specific examples you point out. If there are additional specific examples, please point them out so that we can address them.

-I'm having trouble figuring out what message the authors are trying to communicate: the Raspberry Pi camera set-up or that DeepLabCut can be used for its intended purpose of tracking animal position from video.

Response: The messages that we want to communicate are the following: (1) Equipping your operant conditioning chambers with video cameras are important to enable thorough descriptions of how animals behave in complex tests of cognitive function (2) One can either buy suitable camera equipment to do this, or construct a versatile and inexpensive camera using this protocol (3) DeepLabCut can be used to track the state of a strong light signal inside the chambers, which will help greatly when identifying and extracting short segments of interest from the long videos that are recorded. (4) The camera described here provide videos of good enough quality to allow tracking of animal position with DeepLabCut (5) These tracking results can subsequently be used to obtain an improved understanding of the animals' behaviours inside the chambers.

We have rewritten large parts of the introduction, preliminary data and discussion to better highlight this.

The camera portion of this manuscript would be extremely useful to the behavioral field at large and would open this technological option to many researchers without engineering confidence or background. Furthermore, there are many other applicable situations that would benefit from a small, cheap camera like the one described in this manuscript (recording in an animal cage placed on a cage rack in an animal facility, for example).

Response: Thank you for this positive feedback. We hope the additional revisions that have been made to this segment are to your liking. We have added a short sentence commenting that the applications of the camera do not stop at recording videos from operant conditioning chambers (please see line 85-86 in the clean revised document), but for the messages we would like to convey it still serves as the most prominent example.

However, the DeepLabCut portion is a chicken and egg dilemma: alone, it does not add to the field unless specific types of behaviors have been identified, but researchers need to learn how to use DeepLabCut in order to identify said behaviors. The way the behaviors in this manuscript (head location and velocity) are shown, it is impossible to judge their validity or utility. It would help if the authors added data showing how their method correctly tracks a validated, relevant, and widely used behavior such as tracking rearing, grooming, or whisking. This can be performed using the same videos used to derive the head location and velocity metrics.

Response: The aim of the DeepLabCut portion is to trigger an interest in using this or similar tools for analysing animal behaviours during complex cognition tests performed in operant conditioning chambers. In our experience, many researchers avoid video recording their test sessions simply because they do not consider manual scoring of animal behaviours to be efficient or accurate enough. By showing here that one can use an open source tool to aid in segmenting the long videos recorded from test sessions as well as track the position of the animals, it would further motivate researchers to build the described camera and use it in their research. However, we do see how the representative data described in the initial submission did not clearly convey the utility of the method. We have addressed this by describing in more detail the type of behaviors that can be evaluated, and their utility (see Fig. 11). Regarding this, analysis of an animals' general movements and head placement can be used to evaluate strategies used by rodents on complex tests of cognition (see refs 6-10). For this protocol, it is therefore more interesting to show that one can reliably track position and head movements rather than specific behaviours such as grooming, rearing and whisking. What we are interested in conveying is that DeepLabCut can be used to track the animal's position inside the chamber, and that this tracking in turn can be used to gain insight into an animal's performance on a complex cognitive test. We see that the representative data shown in the initial submission was not sufficient to clearly communicate this and have therefore changed large parts of the introduction and added additional representative data (Fig. 11) to further showcase this.

-The manuscript highlights that automated methods of behavioral video-analysis (such as the DeepLabCut protocol described here) offer options to measure "complex behaviors," but the authors do not show that capability - though they mention this shortcoming in lines 465-467.

Response: What was meant with "complex behaviors" primarily related to strategies and mediating behaviors that animals use to manage complex test of cognition. As explained, an understanding of this can be obtained by analysing movement and position of an animal, and does not necessarily require tracking of specific complex movements (e.g. grooming). To make this clearer, we have refrained from talking about "complex behaviors".

The authors spend a good portion of the instructions describing how to use DeepLabCut without showing its validated efficacy and which already has detailed instructions elsewhere (citation 14 in this manuscript).

Response: The parts relating to DeepLabCut have largely been modelled on another JoVE publication where it is used to track paw movements in a reaching test (Bova et al. 2019). The protocol steps and representative results described in our initial submission are from what we can see comparable to what was presented there. The authors of that publication also, from what we can see, do not specifically address the efficacy or accuracy of DeepLabCut in general, or details regarding their example data.

We have added additional information detailing DeepLabCut's performance in our hands to the representative data (See Fig. 10 and Vid. 1 in particular). The protocol parts have also been expanded somewhat, to try to address comments from both you and the editor.

Minor Concerns:
-Line 45 - "run" is inaccurate

Response: This has been addressed, please see line 48 (clean file).

-Line 46-65 - provide appropriate citations for sentences claiming how typical operant systems are comprised and what data is collected

Response: The statements made here are very general and would typically not require specific citation, as they are not in any way controversial. As such, finding research articles or other scientific literature that clearly relate to these sentences is tricky. We have adjusted the text a bit and added a few references and hope that it is to your liking, please see lines 48-60 (clean file).

-Specifically, line 62-64 "As such….non-existent" needs support of a citation.

Response: The sentence has been rewritten and additional citations have been added. Please see lines 60-62.

-Line 70-71 - the sentence beginning with "Unfortunately…" needs citation support

Response: In the process of the extensive rewrites, this statement has been removed.

-Line 73 - remove the comma after "static"

Response: In the process of the extensive rewrites, this statement has been removed.

-Line 77-78 - "This type of analysis, however, suffers from a low throughput, a strong component of subjectivity and cannot reliably be used to assess complex and discreet aspects of behaviors." These types of statements need a citation, or some other support, and more explanation. Though manual scoring is prone to human bias and error, manual video scoring is well-suited to assess complex and discreet aspects of behavior - it is still the accepted ground-truth or gold standard with common intra class correlation coefficients and Cohen's kappa of >0.9 for many scoring types. Manual scoring can be better-suited than automated algorithms which often have their own significant error and bias (common intra class correlation coefficients and Cohen's kappa tend to be in the 0.7-0.9 range). Perhaps describing "complex and discreet aspects of behavior" in more detail would help clarify your meaning.

Response: We agree that this was strongly overstated in the initial submission and have removed this and similar claims throughout the manuscript.

o Some relevant citations:
o 1) Zheyuan Wang1 & S. Abdollah Mirbozorgi1 & Maysam Ghovanloo. An automated behavior analysis system for freely moving rodents using depth image. Medical & Biological Engineering & Computing (2018) 56:1807-1821 https://doi.org/10.1007/s11517-018-1816-1.
o 2) Wong, J., Shah, P.K. 3D Kinematic Gait Analysis for Preclinical Studies in Rodents. J. Vis. Exp. (150), e59612, doi:10.3791/59612 (2019) doi:10.3791/59612
o 3) Haji Maghsoudi, O., Vahedipour, A. & Spence, A. A novel method for robust markerless tracking of rodent paws in 3D. J Image Video Proc. 2019, 79 (2019). https://doi.org/10.1186/s13640-019-0477-9

Response: We would happily add these references, but in the protocol's current state we do not clearly see where these are appropriate. Comments regarding how DeepLabCut compares to other tracking software have been removed.

-Line 84-85 - change to "These have, in some studies, successfully…"

Response: In the process of the extensive rewrites, this statement has been removed.

-Line 85 - need to define what delineates a "classic" tracking software from a "neural networks" approach

Response: Sentences contrasting DeepLabCut with classical tracking softwares have been removed.

-Line 92 - there is no logical rhetoric flow between confirming the need to develop neural networks to building a camera

Response: The introduction has been extensively rewritten in order to change the background of the protocol. It currently focuses more strongly on using the camera in combination with DeepLabCut as an alternative to commercially available equipment for recording videos in operant chambers.

-Line 93 - need to define IR and LEDs (I assume IR = infrared, but you must write out any acronym the first time you use it in a paper regardless of how common it feels to your field)

Response: This has been addressed, please see lines 70 and 84.

-Line 100-113 - this should be 2 paragraphs: 1st paragraph describes the protocol's intent and limitations; 2nd paragraph describes the details of the example data (which need to be edited to include the relevant details and how it is being used as an example)

Response: Details regarding the example data has been moved to the result section, to improve flow of the text. Please refer to lines 496-502, Fig. 6A, Fig. 10 and Fig. 11.

-Line 131 - need to give the full name of NOOBS too (I assume New Out Of Box Software?)

Response: This has been addressed, please see line 124.

-Line 136 - the "magnetic metal ring" is not listed anywhere on the parts list, nor is the recommended SD card

Response: The recommended SD card was listed on row 5 in the initial submission's material list. It is now listed on row 2. The magnetic ring comes together with the component for the camera stand and lens, which has now been specified.

-Line 154 - "open a terminal window" needs more description

Response: We have added an additional figure (Fig. 3) to better explain these points.

-Line 143-157 - this portion needs an image of the GUIs with callouts if it's not going to be included in the JoVE video

Response: We have added an additional figure (Fig. 3) to better explain these points.

-Line 161 - "edit the rc.local file" needs more description

Response: The description of how to edit the rc.local file is given in the steps that follow this particular one (under its related sub headers). We have rewritten the step to make that clearer (see line 157. We have also added an additional figure (Fig. 4) to better explain these points.

-Line 167 - "... bottom of it (?), and make some space" needs more description

Response: We have added an additional figure (Fig. 4), and updated the text (lines 157-170) to better explain these points.

-Line 176 - hold all 3 commands or do them in sequence?

Response: The exact way of writing in the commands are displayed in Fig.4 and explained lines 157-170.

-Line 177 - Figure 1B has insufficient detail to follow these instructions. The figure needs text describing the components - possibly even call-outs to the instruction number for how to attach each component (more on this below)

Response: The text now refers to Fig. 5A (please see line 181). The intention with this figure is to show a schematic view of the switches and LEDs and not to give a detailed description of how to solder them. Guides on soldering are available on various internet sources, and we do not think it is necessary to include that here. More detailed description of how to attach the components to the GPIO pins are now given on lines 204-220. Information regarding how to count the pins on the microcomputer have been added in Fig 1 and 5.

-Line 177 - step 1.15 should be included in the video.

Response: We have addressed this, please see line 181.

-Line 261 - change "is" to "in"

Response: This particular sentence is no longer in the manuscript, a comparable one is written on lines 299-303. It should not contain this particular typo.

-Line 269 - even though DeepLabCut has been described elsewhere, "start the training…" needs more description

Response: We have rewritten the steps describing how to use DeepLabCut somewhat, but as you do not specify what information you think is lacking it is difficult to know if the rewriting has efficiently addressed your concerns (lines 287-343). The initial description of how to use DeepLabCut was similar to that given in other JoVE protocols using DeepLabCut (Bova et al. 2019). To try to address comments from the editor, we have added some additional detail to these steps. However, as pointed out above, an in-depth description of how to work with DeepLabCut is outside the scope of this particular protocol.

-Line 271-272 - 48 hours for what video lengths?

Response: The training of the neural network is done with a set of video frames, not video sequences. The time needed to train a neural network is primarily dependent on the resolution of the video frames and the type of neural network that one chooses (resnet setting). We have adjusted the sentence a bit to better convey this (line 318-320).

-Line 284 - Step 7 should also include a calibration protocol for converting video information into real distances and accounting for the severe parallax created by using a fish-eye lens

Response: While this would be important for some applications of the camera, we have not seen the need to do either at this time. The parameters we evaluate concern the animal's position in relation to certain points within the chamber, and whether these are calculated with number of pixels or mm is of little importance. Moreover, the area of the operant chamber is limited to the central part of the camera's field of view, where the parallax created by the fish-eye lens is very mild. The general topic is, however, more clearly addressed in the representative results (Fig. 9) (lines 441-449) and in Supplementary file 4. We believe that this is sufficient.

-Line 299 - Load the data files to where?

Response: The protocol points relating to the data management (lines 354-412) have been reduced to improve readability. This point is no longer in the protocol.

-Line 352 - "reflexes" should be changed to "reflections"

Response: We have addressed this, please see lines 427 and 604.

-Figure 4 - ambiguitiy about whether this is a single trial/recording/event. Perhaps include a table showing what % of points were accurately tracked vs missed for all trials. As it stands, Fig 4 gives a hopefully false impression that DeepLabCut does not work well to track rat head position using the current set-up

Response: We have noted more specifically what one should aim/expect to achieve in terms of tracking accuracy (lines 342, 461-462, 474-477, 709-713, Fig. 10, Vid. 1). The example data has been changed to specifically show inaccurate and accurate tracking of one specific example trial (analysed with two different networks), and there should not be any ambiguity regarding this (Fig. 10) (Vid. 2, 3), (line 651-658).

-Figures 4 and 5 - do not use the same colors (blue and red) to mean different things. In Figure 4 red means "tracked" and blue means "missed" but that changes in Figure 5 for red to mean "erroneously tracked" and blue to mean "correctly tracked"

Response: These particular examples are no longer part of the representative data. They have been replaced with Fig. 10C-E, which should not contain the noted discrepancy.

-Line 395-408 - This level of detail should be included in the instruction portion of the main manuscript. As mentioned previously, more instructional detail should be included, specifically in Fig 1B (i.e. "button to the left" described in line 402 is unclear because no buttons are labeled on the figure)

Response: This has been addressed. Numbering of the GPIO pins are displayed in Fig. 1 and 5. Labels have been added to the components in Fig. 5. In addition, the specific pins used for the components are described on lines 204-220.

-Line 410-420 - again, more detail is needed. Elements called out in the text (ex: "reward tray" from line 412) are not labeled on the figure. Readers unfamiliar with operant assays will have difficulty understanding.

Response: Numbered labels of noted components of the chambers have been added in Fig. 7 and 8.

-Line 415 - "reflexes" should be changed to "reflections", and the reflections should be called out with arrows or text on the image

Response: This has been addressed, the noted reflections have been given numbered labels in Fig. 7.

-Line 416 - "reflex" should be changed to "reflection" and called out in the image using arrows or text

Response: This has been addressed, the noted reflections have been given numbered labels in Fig. 7.

-Line 418 - again, call out the IR beam break detectors in the image using arrows or text

Response: One of these have been indicated with an arrow in Fig. 7C. We chose to not label all of them to avoid cluttering the image.

-Line 420 - "house lights" and "cue lights" should be better described in the text. There's some ambiguity in how they are used throughout the text that could be misinterpreted to indicate the "room" lights, "operant chamber" lights, or the LEDs attached to the Raspberry Pi set-up. These lights should also be labelled using text on the respective images in which they appear

Response: This particular sentence has been removed, as it is not important to specify which lights are shining, but only that the chamber is brightly lit (see lines 607-608). Regarding the use of “house light” and “cue light” these should be consistently used throughout all text as far as we can tell. The position of the house light is the same in all example images, and have been indicated with specific labels in Fig. 7A, 8C, and arrows/called out colored dot from DeepLabCut in Fig. 6B, 8A. The cue light of the nose poke openings is very difficult to see in still images from the videos, but where they are of importance, they have been noted with arrows (Fig. 6D, Fig. 11F).

-Line 422-427 - the red, blue, and green dot colors are difficult to discern on the images. Making them brighter and labelling them with text on the image would help

Response: The red and green dot seem fine to us. We have superimposed a yellow dot on top of the blue one for the images, to more clearly note the rat’s head position. Videos currently retain the original label colors as relabelling them would require quite some processing time with DeepLabCut, which we have not had the time for. As the dots mark the head and tail of a rat, we do not believe that readers need additional guidance with written labels to find them. The position of the house light has also been noted in several figures and is consistent between figures.

-Line 431 - unclear what "For both trackings…" means

Response: This particular figure is no longer part of the representative data.

-Line 429-441 - the body text and the figure legend for Figure 4 are ambiguous and do not appear to match. It is unclear in the body text that Fig 4 represents 2 separate trials/recordings/events - the "red line" trial where the animal was accurately tracked and the "blue" trial where the animal was not accurately tracked. The description can be misinterpreted to indicate the correctly tracked and incorrectly tracked portions of a single trial. This needs to be better clarified in the body text and image

Response: This particular figure is no longer part of the representative data. It has been replaced with Fig 10, which we believe is clearer in these regards.

-Line 429-441 - features such as "smooth change" and "jagged and exaggerated changes" need to be called out via text in Fig 4B

Response: This comment now better refers to Fig 10C-E, where this has been addressed through a liberal use of arrows.

-Line 457 - remove the comma after "positions"

Response: This particular sentence is no longer in the manuscript.

-Line458-460 - the sentence beginning with "In comparison…" needs citation support

Response: The sentence has been removed.

-Line 488 - change "selecting" to "select"

Response: The sentence has been removed.

-Line 489 - change "in deemed" to "is deemed"

Response: The sentence has been removed.

-Line 508-514 - alternately, using a visual 3-dimensional calibration grid can get around the parallax ("curving") created by the fisheye lens

Response: We believe that Fig.9 and the text on lines 441-449 addresses these concerns well.

-There is not strong justification for why a Raspberry Pi camera needs to be used instead of any other camera until the discussion (lines 492-506)- the authors need to highlight the unique features of this camera and what need it fulfills much earlier in the manuscript.

Response: The introduction and general aims of the protocol have been rewritten in order to remove points that previously claimed that DeepLabCut tracking is superior to manual video scoring. In this process, the use of the camera, together with DeepLabCut, as an alternative to commercially available video recording setups for operant chambers has been more strongly highlighted. Please see lines 68-91.

-This manuscript is completely lacking any sort of quantitative validation for how well

DeepLabCut performed against manual tracking of the same behaviors, though the authors claim manual scoring and tracking is inferior.

Response: We acknowledge that this constituted a strong overstatement on our part and have removed all such claims, as gathering such data is not feasible at this time.

Reviewer #2:
Manuscript Summary:
Both the title and abstract are appropriate for this article given that they reflect clearly what will be presented and discussed.

Response: Thank you. We hope that this is still the case, following the extensive rewrites that were done on the manuscript.

The authors discuss that this protocol if for operant conditioning analysis, and they do not seem to leave out that it can also be applied to operant tasks other than de 5CSRTT.

All materials discussed are listed.

All the critical steps seem to be highlighted in this Raspberry Pi/ DeepLabCut protocol, and the anticipated results seem reasonable and useful as a reference for readers who will be applying this protocol to this 5CSRTT or other operational tasks.

Response: Thank you. To address the points raised by reviewer #1, we have added several figures relating to both the protocol and representative results. We hope that it is still to your liking.

The references included are useful, but it seems important to include more about basic programming and what deep convolutional neural networks are.

Response: We have included a few sentences to try to help new-comers find suitable references to get going with the programming needed for the data management (please see lines 359-361). Building the camera does not require any programming skills. This has been tested by having two persons with no programming skills or prior experience with the equipment carry out the protocol successfully. We have also downplayed the deep convolutional neural network aspect of DeepLabCut compared to the initial submission. The descriptions given should be largely in line with what is presented in other JoVE articles using DeepLabCut (Bova et al. 2019).

Minor Concerns:
Although there are many steps described, they seem to be all present and lead to the described outcome. The steps seem clearly explained for someone familiar with programming; however, I suggest that they clarify further for people not familiarized with programming. It would be useful to include a basic programming guide (or at least cite a recommend paper) for those not familiarized with the programs described in the protocol (Python, DeepLabCut programming, etc.). Also, it seems important that they define in the introduction of what a deep convolutional neural network is.

Response: We have added additional clarifications of the protocol text, and figures to make the camera construction more accessible (note specifically Fig 3 and 4). We have specifically asked people with no programming experience to carry out the protocol, and they have done so successfully.

We don't believe that adding a general introduction to programming would be suitable or practical, but have added valuable references for anyone who wants to start developing these skills (please see lines 359-361). We have also added some supplementary text (Supplementary file 2), and a related python script (Supplementary file 3), to help readers with little prior knowledge manage the conversion of the video files, as well as editing them a bit (video editing supplement). We agree that this is beneficial, but suggest to keep it separate from the main protocol.

Following the extensive rewrites of the introduction and other parts of the protocol, there is much less emphasis on neural networks. As such, an introduction to this topic does at this point seem superfluous. This should be in line with other JoVE publications that use DeepLabCut (Bova et al. 2019).

Alternative text for steps 4 and 5 of the protocol:

4. Neural network training using DeepLabCut

NOTE: This section gives a rough outline for how to use DeepLabCut to train a neural network in tracking the protocol step indicator and the head of a rat. Installation and use of DeepLabCut is well-described in other published protocols[15,16]. Each step can be done through specific Python commands or the program's graphic user interface. The steps outlined below refer to the latter.

4.1. Install DeepLabCut and start its graphic user interface.

4.2. Create and configure a new project by following the steps outlined below.

4.2.1. Under the "Manage Project" tab, select "Create new project" and enter a project name as well as the name of an experimenter.

4.2.2. Click on the "Load Videos" button and select the videos recorded in step 3.

4.2.3. Click on the "Ok" button. This creates a folder for your project.

4.2.4. Click on the "Edit config file" button.

4.2.5. In the opened text file, navigate to the header "bodyparts" and replace the automatically added "- bodypart1", "- bodypart2", "- bodypart3" and "- objectA" with "-head" and "-protocolstepindicator".

4.3. Extract video frames from one or more videos by following the steps outlined below.

4.3.1. Under the "Extract frames" tab, select "manual" extraction method and click on the "Ok" button.

4.3.2. In the window that opens, click the "Load videos" button and select a video from the project's video folder.

4.3.3. Use the slider bar to navigate through the video and click the "Grab Frames" button to save individual video frames into the project folder.

4.3.4. Extract 700-900 video frames.

4.3.4.1. Make sure to include video frames that display both the active (Fig. 8A) and inactive (Fig. 8B) state of the protocol step indicator.

4.3.4.2. Makes sure to include video frames that cover the range of different positions, postures and head movements that the rat may show during the test. This should include video frames where the rat is standing still in different areas of the chamber, with its head kept both along the chamber floor and angled downwards video frames where the rat is actively moving, entering nose poke openings and the pellet trough.

NOTE: If the animals differ considerably in fur pigmentation or other visual characteristics, it is advisable that the 700-900 extracted video frames are split across several videos. Through this, one trained network can be used to track different individuals.

4.4. Manually label the extracted video frames by following the steps outlined below.

4.4.1. Under the “Label frames” tab, click the “Label frames” button. In the window that opens, click the “Load frames” button and select the folder for the video that was used in step 4.3.

4.4.2. Use the “< <Previous” and “Next> >” buttons to jump between the video frames extracted in step 4.3.

4.4.3. Mark the position of the rat’s head in each video frame by placing the mouse cursor at a point centrally between the rat’s ears and right-clicking (ensure that “head” is selected under “Select a bodypart to label”. If the initial label placement is not satisfactory, move the label by clicking and holding the left mouse button, when hovering over it with the mouse cursor.

4.4.4. Mark the position of the houselight or other protocol step indicator in video frames where it is active (Fig. 8A,B).

4.5. Create a training data set from the labeled video frames by navigating to the “Create training dataset” tab, selecting “resnet_101” under “Select the network” and clicking the “Ok” button.

4.6. Start the training of a neural network by navigating to the “Train network” tab and clicking the “Ok” button.

NOTE: Throughout steps 4.3. to 4.6, ensure that the path to the project’s config.yaml file is correct.

4.7. Stop the training of the network when the training loss has plateaued below 0.01. This may take up to 500,000 training iterations.

NOTE: When using a GPU machine with approximately 8 GB of memory and a training set of about 900 video frames (resolution: 1640 x 1232 pixels), the training process has been found to take approximately 72 h.

5. Reviewing tracking accuracy and analyzing videos using the trained network

5.1. Use the trained network to analyze videos by following the steps outlined below.

5.1.1. In the DeepLabCut graphic user interface, navigate to the “Analyze videos” tab. Select videos in the drop-down menu next to “Choose the videos”, specify the file format in the drop-down menu under “Specify the videotype”, select “Yes” under “Want to save result(s) as csv?” and click the “Ok” button. This creates data files listing the coordinates for the tracked positions of the head and protocol step indicator in each frame of the analyzed videos.

5.1.2. Once the analysis is done, prepare marked-up videos that show the tracked positions of the points of interest. For this, navigate to the “Create Videos” tab, select the videos to be marked under the drop-down menu next to “Choose the videos”, specify the file format in the drop-down menu under “Specify the videotype” and click the “RUN” button.

5.2. Evaluate the accuracy of the tracking by following the steps outlined below.

5.2.1. In DeepLabCut’s graphic user interface, navigate to the “Evaluate network” tab and click on the “Ok” button. This provides an automated evaluation based on the video frames that were labeled in step 4.4, detailing how far away on average the position tracked by the network is from the manual label.

5.2.3. Select one or more brief video sequences (of 200 video frames each) in the marked-up videos obtained in step 5.1.2. Go through the video sequences, frame by frame, and note in how many frames the labels correctly indicate the positions of the rat’s head, tail, etc, and in how many frames the labels are placed in erroneous positions or lost.

5.2.4. If the tracking of an object is frequently lost or mislabeled, identify the situations where tracking fails. Extract and add labeled frames of these occasions by repeating steps 4.3. and 4.4. Then retrain the network and reanalyze the videos by repeating steps 4.5-4.7 and 5.1. Ultimately, tracking accuracy of >90% accuracy should be achieved.

Supplementary video editing protocols

The following steps were not included in the main protocol file to keep it more focused on its main themes and aims. However, for researchers with limited experience in video editing and the use of terminal commands, these steps constitute helpful advice in working with the video files created by the camera.

1. Converting the recorded videos from .h264 to .MP4.

As described in the main protocol, it is worth converting the .h264 files created by the camera into another format. We suggest using .MP4. There are multiple ways that this conversion can be achieved. We have found it most convenient to work with ffmpeg (http://ffmpeg.org), a software package for video editing that operates through the terminal of an operating system. ffmpeg is available for all major operating systems, although the steps detailed below specifically refer to Mac OSX system. Readers may benefit from first getting acquainted with using terminal commands (Apple, 2020, https://support.apple.com/guide /terminal/welcome/mac).

1.1. On a system using mac OSX, Go to Applications > Utilities > Terminal, to open a terminal window.

1.2. Type in "/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"", and press enter. This will install the package manager HomeBrew. You may be prompted to type in your password to allow the installation to proceed. **Note:** The first and last quotation marks should not be included in the command. It is best to type this text in as opposed to copying it from the document, as the latter might interfere with formatting.

1.3. Type in "brew install ffmpeg" and press enter. This will install ffmpeg on your computer, through HomeBrew.

1.4. In the terminal window, navigate to a folder where the .h264 videos are stored. This is done by typing in "cd" followed by the path to the folder, ex: "cd Documents/Videos/"

1.5. Type in "ffmpeg -framerate 30 -i FILE_NAME.h264 FILE_NAME.mp4" and press enter. **Note:** The first "FILE_NAME" in the command specifies the video file that is to be converted while the second "FILE_NAME" specifies the name you wish to give to the converted file. The video recording speed of the camera described in this protocol is set to 30 frames per second (fps). It is important that this is specified when converting the video files into MP4, or another format.

2. Editing full-length video files to include only segments of interest.

Once one has located the starting points and durations of a set of interesting video segments, it is often beneficial to edit the full video files of a test session to only include the segments that are of primary interest. This makes storing of videos as well as reviewing and/or presenting the results much more convenient. To achieve this, ffmpeg is once again used, but this time through a Python script, rather than simply using the terminal.

2.1. Install Python, and a Python integrated development environment (IDE). This allows you to run python scripts. **Note:** Readers may benefit from first reviewing resources available for persons new to this type of work (Python, 2020, https://wiki.python.org/moin/BeginnersGuide/NonProgrammers).

2.3. Download and open Supplementary file 3 in a Python IDE program.

2.4. Edit the script to update the paths to where ffmpeg is installed on your computer, and where videos are kept on your system. **Note:** The code in the script is written in a generic way, with comments added to guide the editing of the script.

2.5. Run the script and review the results.

Fish-eye correction supplement

As described in the representative results shown in the main protocol, the use of a fisheye lens results in the recorded image being distorted. There are different approaches one can take in order to correct such distortions, if it is deemed necessary. We have implemented a proposed industrial algorithm[1] which is based on the framework proposed by Scaramuzza et al[2]. This approach uses a linear regression model to correct for compression and expansion of pixel coordinates.

Given the fact that the center of image corresponds to the center of the fisheye lens, an intrinsic parameter for the lens (f) is defined, using image width and horizontal field of view (FOVhor):

$$f = \frac{image\_width}{4\sin(\frac{FOV}{2})}$$

Thereafter, an intermediate parameter of image distortion (λ) is calculated:

$$\lambda = \sqrt{\frac{x^2 - y^2}{f}}$$

Note: x and y represent the uncorrected coordinates of pixels obtained from the fish eye equipped camera. In relation to the current protocol, they more specifically represent the uncorrected coordinates obtained from DeepLabCut.

Then the final correction for coordinates (xf and yf) can then be obtained through:

$$x_f = \frac{(2xSin\left(\text{atan}\left(\frac{\lambda}{2}\right)\right))}{\lambda}$$

$$y_f = \frac{(2ySin\left(\text{atan}\left(\frac{\lambda}{2}\right)\right))}{\lambda}$$

As noted in the representative data for this protocol, the distortion induced by the fish eye lens is minimal within the field of view that covers the operant chamber. In line with this, correcting the path travelled and speed profile of some example data showed only minor effects (Fig. S1). Still, it is noteworthy that the correction resulted in a consistent reduction in peak speeds (Fig. S1A) and the tracked path being slightly compressed towards the image center (Fig. S1B). However, average movement speeds across 25 5CSRTT trials for four different animals show virtually no overall effect of the correction (Fig. S2). As such, impact of and need for correcting the fish eye distortion will depend on the parameters of interest and other study-specific concerns.

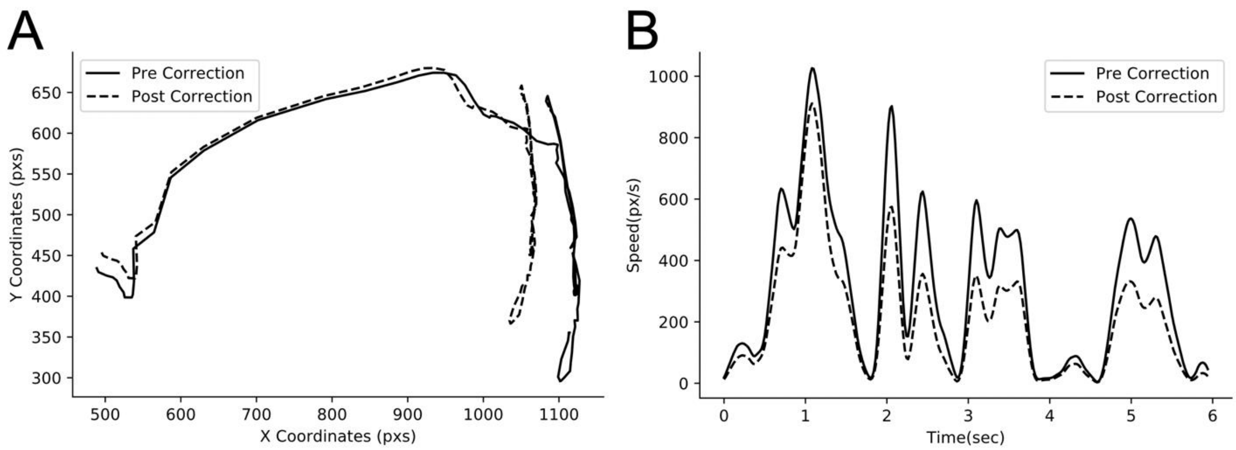


Fig. S1. Example data displaying effect of mathematical correction of fish eye distortion on plotted path during an ITI and subsequent response on a 5CSRTT trial (A), and the changes in movement speed of the animal's head during this behavioral sequence (B). Notably, while the effects of the correction are discreet, the plotted path is more constrained horizontally, and the peak speeds are slightly reduced. Data relates to the performance of one animal on a single 5CSRTT trial either corrected or not corrected for fish eye distortion. Abbreviations: pxs: pixels.

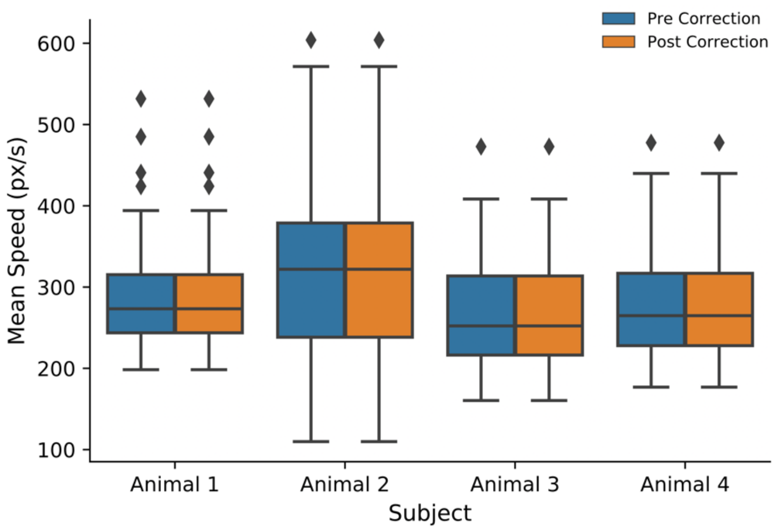


Fig. S2. Example data displaying the effect of mathematical correction of fish eye distortion on the average speed during 25 trials of the 5CSRTT, for four different rats. Box plots indicate mean and range (25 and 75 percentile). Outliers are indicated with black diamonds. Abbreviations: pxs: pixels.

1. Altera corp, A Flexible Architecture for Fisheye Correction in Automotive Rear-View Cameras, *White paper*. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01073-flexible-architecture-fisheye-correction-automotive-rear-view-cameras.pdf (2008)

2. Scaramuzza, D., Martinelli, A., Siegwart, R. A Toolbox for Easy Calibrating Omnidirectional Cameras. *Proceedings to IEEE International Conference on Intelligent Robots and Systems, (IROS)*. Beijing, China, October 7–15, (2006)

Supplementary file 1

Click here to access/download
Supplemental Coding Files
Supplementary_file_1_Pi_video_camera_Clemensson_2019.py

Supplementary file 3

Click here to access/download
Supplemental Coding Files
Supplementary_file_3_Video_edit_ffmpeg.py