

Jul 30, 2020

A step by step guide to using Visual Field Analysis

Mathilde Josserand¹, Bastien S Lemaire²¹Ecole Normale Supérieure Lyon; ²University of Trento, Center for Mind/Brain Sciences**1** Works for me dx.doi.org/10.17504/protocols.io.bicvkaw6

Bastien Lemaire

University of Trento, Center for Mind/Brain Sciences

ABSTRACT

In this protocol, we provide a step by step guide to using Visual Field Analysis (VFA) successfully. VFA is a python program based on DeepLabCut toolbox (Nath et al., 2019). Using our program, it is possible to score reliably the eye use, activity, and time spent in different zones of different animal species and experimental paradigms for more reproducible research.

DOI

dx.doi.org/10.17504/protocols.io.bicvkaw6

DOCUMENT CITATION

Mathilde Josserand, Bastien S Lemaire 2020. A step by step guide to using Visual Field Analysis.

protocols.iodx.doi.org/10.17504/protocols.io.bicvkaw6

KEYWORDS

python, computational method, animal behaviors, eye use, activity, tracking

LICENSE

————— This is an open access document distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

CREATED

Jul 08, 2020

LAST MODIFIED

Jul 30, 2020

DOCUMENT INTEGER ID

39029

A step by step guide to using Visual Field Analysis

In this protocol, we provide a step by step guide to using Visual Field Analysis (VFA) successfully. VFA is a python program based on DeepLabCut toolbox (Nath et al., 2019). Using VFA, it is possible to score reliably the eye use, activity, and time spent in different zones of different animal species and experimental paradigms for more reproducible research.

1. Requirements

Video

Our method is entirely based on video recordings. Therefore, a good quality recording is required. However, the highest resolution

and frame rate do not necessarily provide the most accurate results. The best settings are specific to the experimental condition and DeepLabCut process (Nath et al. 2019). The camera choice and its settings can be manipulated depending on experimental conditions and animal models. If working with very fast-moving animals, we recommend using a higher frame rate. Even though the camera choice and recording settings can be modified at will, there are specific parameters that are essentials and must be strictly followed.

List of requirements for the videos recordings:

- the video must be encapsulated as .avi or .mp4;
- the camera should not move during the acquisition;
- the video recording should be taken from above with no distortions (avoid fisheye lenses), to get accurate measurements of the eye-use;
- the apparatus and the stimuli must be already visible at the beginning of each analyzed video recording, to accurately set the arena borders and stimuli location (2. 'data processing', step 1);
- the stimuli observed by the subjects should always be visible on the video recording, to allow measurements of the eye-use.
- the framerate used to record videos must be an integer: if you need to have a number of framerate with decimals (7.5 f/s), then you could not select the option "gather by seconds" later.

DeepLabCut

Video recordings must be tracked using DeepLabCut toolbox. During stage II of the DeepLabCut process (Nath et al. 2019) the head areas (called body parts in DeepLabCut) must be named as follows: 'leftHead', 'topHead', and 'rightHead' (Figure 1A). If the stimuli observed by the animals are moving objects, they should be tracked using DeepLabCut too. If only one stimulus is present, it should be named 'stimulus' (Figure 1B). In this case, the stimuli should be named 'stimulusLeft', 'stimulusRight' in a left/right arena configuration, or 'stimulusTop', 'stimulusBottom' in a top/bottom arena configuration (Figure 1C, information about the arena configuration is provided in 2. 'complete launcher information', step 14).



Figure 1: Screenshots showing the modifications that have to be made to the 'config.yaml' file during the DeepLabCut process. Figure A shows how the file must be modified to track only the head areas (for static stimuli that do not need to be tracked). Figure B shows how the file must be modified to track a moving stimulus and the three head areas. Figure C shows how the file must be modified to track two moving stimuli (in a top/bottom arena configuration) and the three head areas.

During stage IV ('labelling of the frames') of the DeepLabCut process, a certain amount of frames must be labelled by the experimenter indicating the position of the three points previously defined during stage II (200 frames of different situations labelled manually usually provide accurate results, Nath et al. 2019). The accuracy of this manual labelling step is crucial to obtain accurate tracking. If in these frames, the eyes are not visible, we suggest placing the labels as close as the real eye positions, to obtain the most accurate tracking (Figure 2).

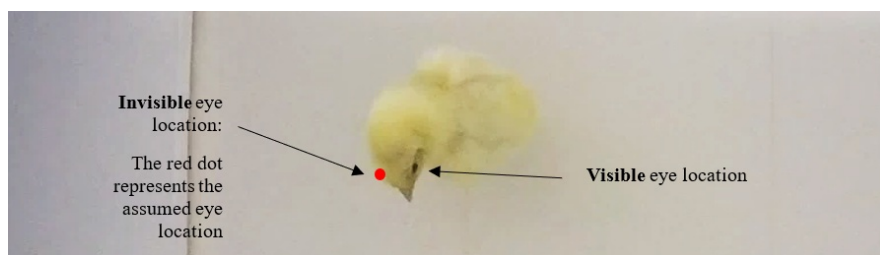


Figure 2: Screenshot showing typical labeling strategy when the eye location is invisible.

The 'leftHead' label must be placed on the left eye or as close as possible to it, while the 'rightHead' label should be located on the right eye or as close as possible to it. The 'topHead' label must be placed between these two points, but in a slightly more rostral position (as in Figure 3). The labels must represent the head orientation of the animal correctly.

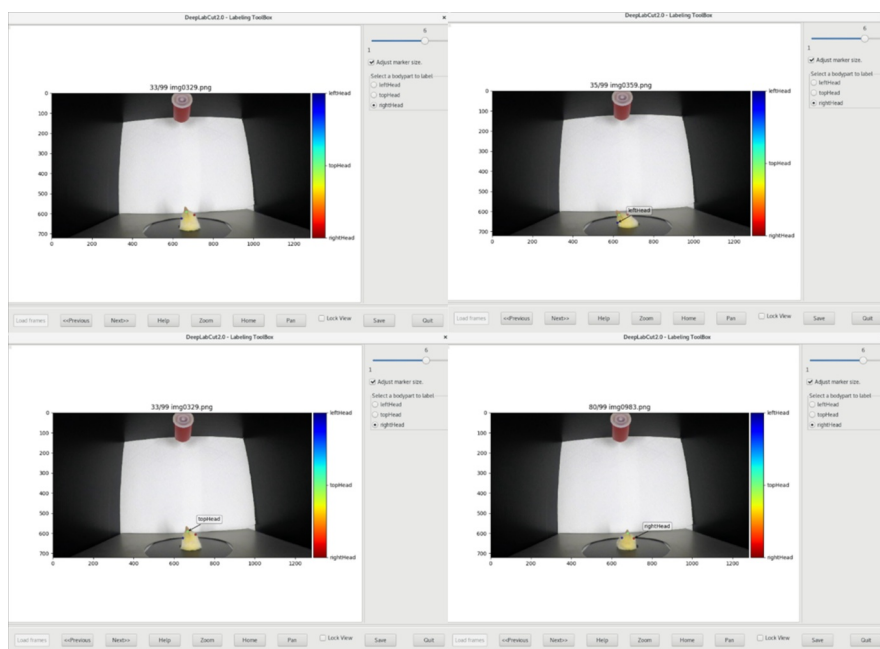


Figure 3: Screenshots showing typical label placement, taken from the example available in our Github (more labelled frames can be seen in our [Github](https://github.com/DeepLabCut/DeepLabCut2.0/blob/master/Example/chick1_labeled) in the folder [Example/chick1_labeled](https://github.com/DeepLabCut/DeepLabCut2.0/blob/master/Example/chick1_labeled)).

During stage IX of the DeepLabCut process, the command 'save_as_csv=True' must be added in the 'deeplabcut.analyse_videos' function, to obtain files with the right extension (.csv, Figure 4).

bodyparts	leftHead	leftHead	leftHead	topHead	topHead	topHead	rightHead	rightHead	rightHead
coords	x	y	likelihood	x	y	likelihood	x	y	likelihood
0	688.9025717	-0.938052654	5.57E-07	683.9526	1.474733591	2.06E-06	0.656967402	4.416795194	3.04E-07
1	688.0762591	-0.937449932	6.64E-07	684.08199	1.275731325	2.70E-06	0.677842855	4.385871112	2.82E-07
2	688.139585	-0.867773533	6.95E-07	684.0908821	1.533869028	3.02E-06	0.693254232	4.49374786	3.35E-07
3	687.9290609	-0.946468353	7.72E-07	684.4860428	1.264309406	3.64E-06	0.608977556	4.468123704	3.61E-07
4	688.1614289	-0.956752777	4.94E-07	684.6658507	0.927484512	2.35E-06	0.635291815	4.436759323	3.25E-07
5	688.4816031	-0.890663624	6.36E-07	684.4815995	0.939720631	2.70E-06	0.673726082	4.491577566	2.91E-07
6	688.1262636	-1.261356354	4.80E-07	684.4243289	1.053180695	2.31E-06	0.662264824	4.488142282	3.03E-07
7	688.0491824	-0.97201252	7.45E-07	684.3053758	1.149925709	3.46E-06	0.676820278	4.329409838	2.81E-07
8	688.0996294	-1.011076927	6.07E-07	684.6206324	1.297753572	3.06E-06	0.548386812	4.334684581	2.77E-07
9	688.1692386	-1.008854866	5.00E-07	684.8844037	1.174372673	2.77E-06	0.629817963	4.335759252	2.96E-07
10	687.9327788	-0.921701431	6.64E-07	684.6713573	0.864924669	3.11E-06	0.606004238	4.366250455	2.97E-07
11	688.2950258	-0.937612534	4.81E-07	684.6300523	1.090647459	2.70E-06	0.522583246	4.300421476	2.67E-07
12	688.0367789	-0.78297472	5.45E-07	684.7807072	1.073469877	3.51E-06	0.56496644	4.357623756	2.77E-07
13	688.0737991	-0.897288799	6.18E-07	684.6477183	1.149011135	3.49E-06	0.578607798	4.308730066	2.60E-07
14	687.9604092	-0.813302994	6.18E-07	684.7036611	0.990899801	3.48E-06	0.597542048	4.287555695	2.61E-07

Figure 4: Screenshot of the output file produced by DeepLabCut (input 2) in our example. Each row contains data for an individual frame. For each labelled point (leftHead, topHead, and rightHead), different columns report the x coordinate (expressed in pixels) and the likelihood ratio assigned by the software.

Excel

Every sheet should contain at least the animal identification number (all animals should have a different number) and the starting/ending time (filed in the following format: hh:mm:ss) of the portion of the video to be analyzed. Visual Field Analysis will analyze only the frames contained between the starting and the ending time located in the excel sheet. New columns can also be added to incorporate additional variables that the experimenter wishes to record about each subject or trial (Figure 5). This information will then be automatically copied within the output files produced by Visual Field Analysis.

If the same animal is tested several times and the different trials are recorded on separate videos, we suggest using for each video a decimal animal identification number corresponding to the id of the animal and the trial number (such as 1.1, 1.2, etc.). If the trials are recorded as one continuous video, we suggest indicating the starting time of the first trial and the ending time of the last trial, so that Visual Field Analysis will analyze all the trials at once. The output produced by our application can then be easily divided manually or computationally into different sessions.

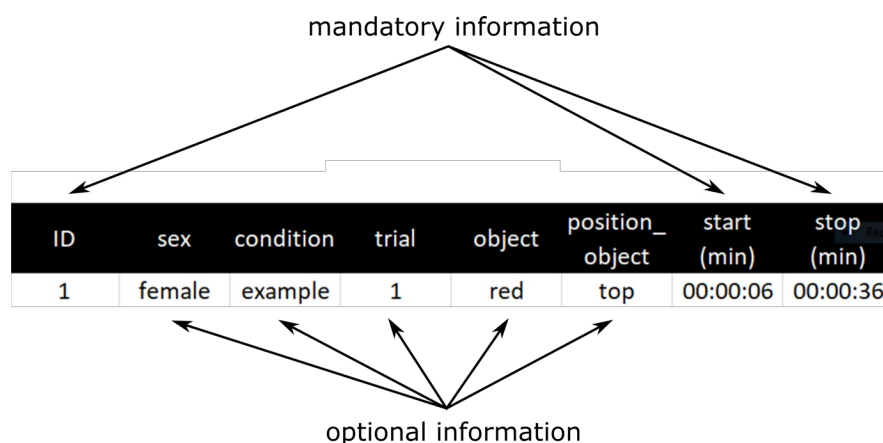


Figure 5: Screenshot of the excel sheet.

2. Visual Fields Analysis Installation

To run our application, the 'Visual Field Analysis' must be downloaded on our [GitHub \(mathjoss/VisualFieldsAnalysis\)](https://github.com/mathjoss/VisualFieldsAnalysis). Visual Field Analysis is based on Python 3. We suggest installing Anaconda (version 1.9.2 or latest) with the latest Python version (3.7 at the moment) and Spyder (version 3.3.6). Furthermore, the following libraries must be installed: pandas, matplotlib, cv2, NumPy and xlrd. We used a lot of different versions which all worked. OpenCV works with version 3.4.2, and so does matplotlib with version 2.0.2. To run our example, the 'Example' folder within our [GitHub](https://github.com/mathjoss/VisualFieldsAnalysis) can be downloaded, and the next steps followed.

Files and folders naming

For each experiment, [we advise creating different directories \(see 'Example' architecture in our GitHub\)](#) where to locate the input files (input 1, 2, labelled and 3 should be located in different directories).

Videos

. We recommend gathering all videos in a folder. The videos can be named with any name as soon as this name is coherent within

videos. Only a number corresponding to the animal identification number can vary between the different videos. For example, videos can be named “mychick1.mp4”, “mychick4.mp4”, “mychick1.3.mp4”, but should not be named “firstchick4.mp4”, “secondchick1.3.mp4”.

DeepLabCut files

Each video is linked to its DeepLabCut files. DeepLabCut files must be gathered in a folder. However, the files’ names must be created according to the following conditions: first, the animal name, second, the animal identification number, and third “_dlc.csv”. In our example, our files name are “chick1_dlc.csv”. Only the identification number must vary between the videos. Files must always follow this format: “animalNUMBER_dlc.csv”.

Excel file.

The name of the excel file does not follow any specific constraints. However, the name of the sheets inside the excel file must correspond to the animal name and its identification number. In our example, our sheets must be called “chick1” and “chick2”. It is impossible to have alternative names for the animal (“firstchick1”, “firstchick2” will not work).

Complete launcher information

To start the application, open ‘main_coordinator.py’ using Spyder (located within the [GitHub](#) directory downloaded previously). The program will open an interface where information for the experiment can be selected (Figure 6). In the next paragraph, we will illustrate every step of this procedure, referring again to the experiment we performed as an example.

The figure displays two screenshots of the 'Visual Fields Analysis Launcher' application interface. The left screenshot shows the initial setup form with various input fields and buttons. The right screenshot shows the final configuration screen with numerical inputs for area lengths and angles, and a 'Let's start!' button.

Left Screenshot (Initial Setup):

- Write animal type : chick, fish...:
- Select path of DLC files folder: (No folder path selected yet)
- Select path for videos files folder: (No folder path selected yet)
- Have you experienced the problem with too heavy videos?: ☒ Yes ☐ No
- If yes, select path with reduced videos. If no, leave it empty.: (No folder path selected yet)
- Write your video names, and replace the number by %s:
- Example: if your videos are called fish1.avi, fish2.avi... write: fish%s.avi
- Select excel file: (No file path selected yet)
- On which side of the apparatus are/is the stimuli?: ☒ Bottom and/or top ☐ Left and/or right
- Stimuli on top and/or bottom: ☐ Stimuli on left and/or right: ☐
- Is/are the stimuli fixed or moving?: ☒ fixed ☐ moving and tracked by deeplabcut
- Is there 1 stimulus on one side or 2 stimuli on both side?: ☒ 1 stimulus ☐ 2 stimuli
- Do you want to group your frames by seconds?: ☒ Yes ☐ No
- Write all the ID of the animals you want to analyze, separated by a space:
- Write the threshold for detecting outliers (Recommended : 3):

Right Screenshot (Final Configuration):

- write first area length:
- write second area length:
- write third area length:
- write fourth area length:
- write fifth area length:
- Write frontal angle for the animal (for ex. 16 for the fish):
- write lateral angle for the animal (for ex. 150 for the fish):
- Let's start!

Figure 6: Visual Fields Analysis Launcher.

Step 1:

Write the animal type. This name should be coherent with the name of your animal written in your excel sheets and DeepLabCut files. In our case, the input names all started with ‘chick’, so this is what we entered into the program.

Step 2:

Browse the folder where you stored the DeepLabCut output files.

Step 3:

Browse the folder where you stored the videos.

Step 4 and 5:

If the program has worked, but you could not visualize frames, reduce the size of the videos by two and store the new videos in a folder. Then, answer “yes” to this question and browse the folder with the reduced videos. Please note that in step 3, you should still enter the folder with the non-reduced videos.

Step 6:

Enter the format of the video name and replace the number by %s. For example, if your videos are named "myfish1.1.mp4" and "myfish1.2.mp4", you should write "myfish%s.mp4".

Step 7:

Browse the location of the excel file. Please note that the file itself must be selected, and not the folder like in the previous steps.

Step 8:

Select if the apparatus has a top/bottom or left/right orientation. In our chick 1 example, the stimulus is located on the top of the video recording, so we chose the top/bottom configuration. For our chick 2 example, we selected left/right configuration.

Step 9 and 10:

Indicate the number of stimuli and if they are moving or static. Visual Field Analysis allows to track simultaneously two stimuli only if they are located on the two opposite sides of the arena (top and bottom or left and right). It is impossible to simultaneously track more than one moving stimulus if they are located on the same side of the arena.

Step 11:

Choose whether to group the results by second or to perform a frame by frame analysis.

Step 12:

Specify the animals' identification number to analyze. In our example, we only tracked one animal, but the program can analyze multiple animals at a time.

Step 13:

Define an error 'threshold'. This threshold specifies the acceptable level of between-frames variability in the distances between the three points tracked on the head of the animal. In some cases, DeepLabCut may track some frames inaccurately. Consequently, we created an error threshold to help to exclude frames in which tracking accuracy was low. Later in the program, the distance between each tracked body part ('leftHead' to 'rightHead', 'leftHead' to 'topHead', and 'rightHead' to 'topHead') will be computed. This allows us to check for potential errors in the DeepLabCut file. In our example, the average distance between the 'leftHead' and 'rightHead' points was of 58 pixels. If in other frames the distance between these same two points is 20 or 80 pixels, it is highly probable that the labels have been wrongly located. These frames should be considered as outliers and excluded from the results. Through the threshold choice, the program will identify potential outliers. A higher threshold will allow a greater variability and will include a higher number of frames into the analysis. This means that we will keep in the analysis frames for which the inter-points distance shows a substantial variation from the average value obtained for that video. In contrast, a lower threshold will be more restrictive and lead to a high number of frames excluded as outliers. The value for this threshold can be as low as 0, which will exclude any variability in the tracking. To choose the best threshold, outlier frames can be visualized later in the program. This allows verifying if the excluded frames did represent cases of inaccurate tracking (Figure 10). It is thus advisable to initially set a lower threshold and proceed to increase it, until only inaccurately tracked frames are excluded. After visualization of the excluded frames, the threshold can be changed. A threshold of 3 has been found to be effective in most of the tests conducted in our laboratory.

Step 14:

Select the number of areas in which the arena should be divided and their lengths (the arena cannot be divided into more than five areas). This information will be later used to determine the time spent in different areas. The different areas must form virtual rectangles of identical width, juxtaposed side by side along with the arena (Figure 7). The overall length of all areas equals the arena length (Figure 7). For example, to divide the arena into three areas (left/top, center, right/bottom), the length of area1 and area5 should be given a value of 0, while areas2, 3, and 4 must be given a different length. We advise setting the length of each area so that it corresponds to its actual size in cm. In our example with chick 1, we were not interested in computing the time spent by the chick in different areas, since the animal was immobile. Consequently, we attributed a value of 20 (corresponding to the length in cm of the arena drawn later in the program, see stage 3, step 1) to the length of the center zone (area3), while the lengths of all the other areas were attributed a value of 0.

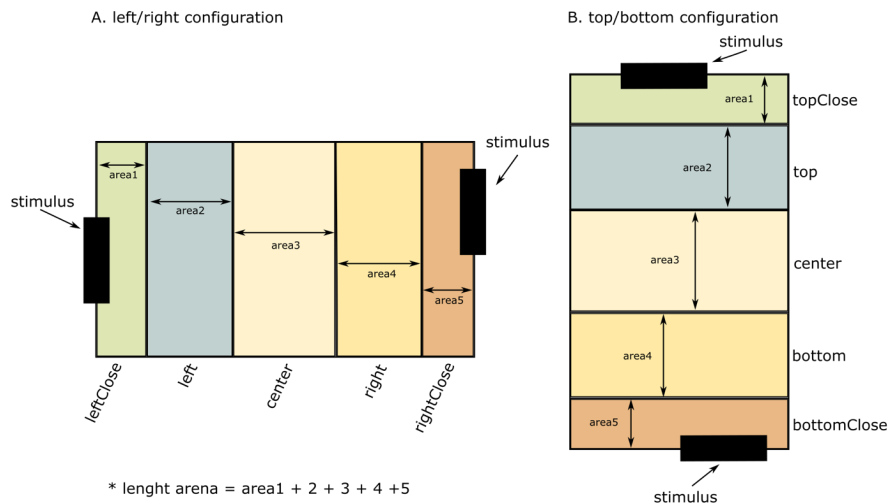


Figure 7: Schematic representation of the settings that can be used to subdivide the arena into different areas. Each arena type (horizontal, A; or vertical, B) can be divided into a maximum of five different areas. The labels 'leftClose, left, center, right, rightClose' and 'topClose, top, center, bottom, bottomClose' correspond to the names of the columns in the output produced by Visual Field Analysis. Hypothetical stimulus placement is also shown.

Step 15:

Define the size of each portion of the visual field. Visual Field Analysis will automatically score which hemifield is predominantly used to look at a stimulus. This will be done using projection lines from the head's 'midline'. The midline is perpendicular to the imaginary line connecting the label points corresponding to the left eye (leftHead) and the right eye (rightHead, Figure 8). At this stage, two visual fields within each hemifield should be defined: the frontal and lateral visual fields. The angle from the midline must be defined for the left visual fields (the sum of the frontal and lateral visual hemifields must be 180° maximum). The program will automatically apply the same values to the right hemifield. The value entered for each visual field corresponds to the angle starting from the midline, which is considered as 0°. In our example, we defined the left frontal visual field as 15° wide from the midline (30° in total for the sum of the left and right frontal visual fields), while the lateral visual field was defined as ending at 150° wide from the midline (135° for each lateral visual field, excluding the width of the frontal visual field, Figure 8). Values from 150° to 180° in each hemifield will automatically be defined as the blind spot of the animal.

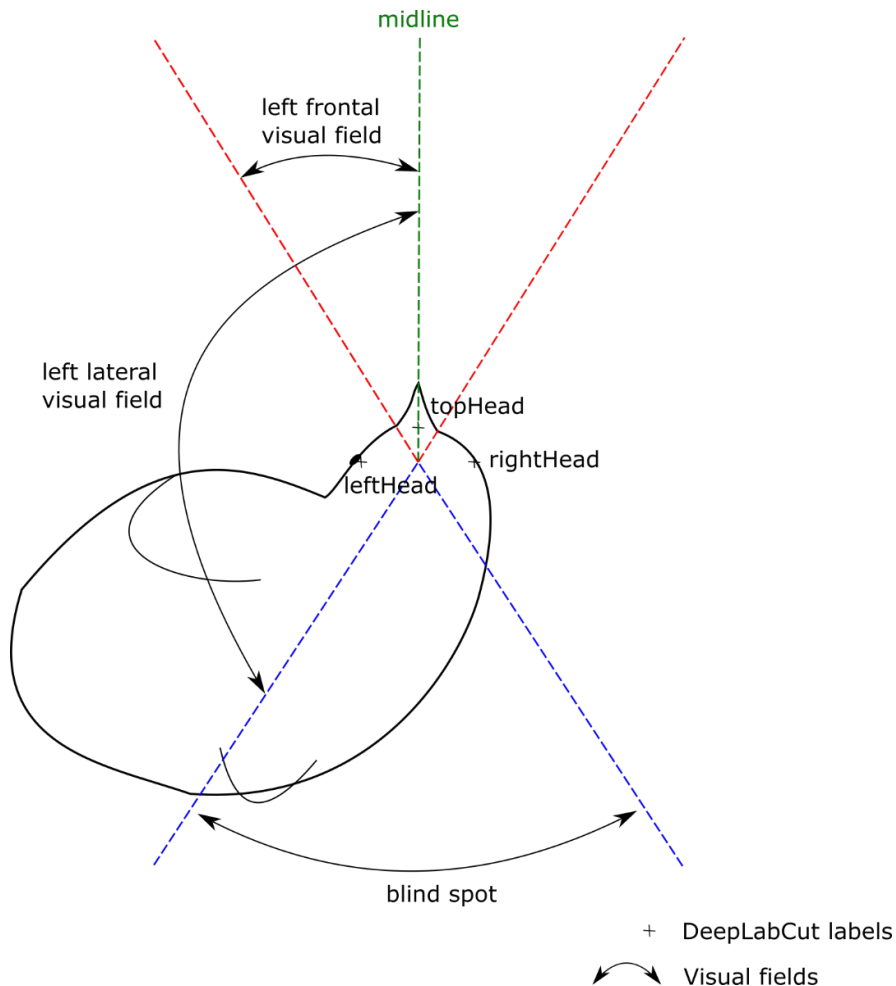


Figure 8: Schematic representation of the visual fields of a domestic chick, as defined in Visual Field Analysis for our example. Each visual hemifield can be divided into two further areas, frontal and lateral.

Data processing

The program performs the computations that are specific to every video, to measure the behaviors accurately.

Step 1:

Specify the borders of the arena and the position of the stimuli. These data must be defined separately for every video. Indeed, the camera may accidentally move from one video to another, slightly altering the position of the arena and stimuli, from one video to another.

At this step, the program opens the first frame of each video recording, for each animal analyzed. To specify the arena borders, the user needs to mark the four corners of the arena. Similarly, for tests with static stimuli, also the stimuli location needs to be defined at this stage. This can be done by marking two points on its borders. (The position of moving stimuli will be tracked by DeepLabCut, see above). The instructions should be followed and the borders placed using the left click of the mouse. Important: the border of the arena should include all possible positions where the animal can be tracked on the video, which can depend on the camera angle (Figure 9). The total length of the arena (Figure 7) corresponds to the distance top/bottom or left/right, depending on the arena orientation.

- ✗ Borders of the arena defined by the experimenter
- ✕ Borders of the stimulus defined by the experimenter

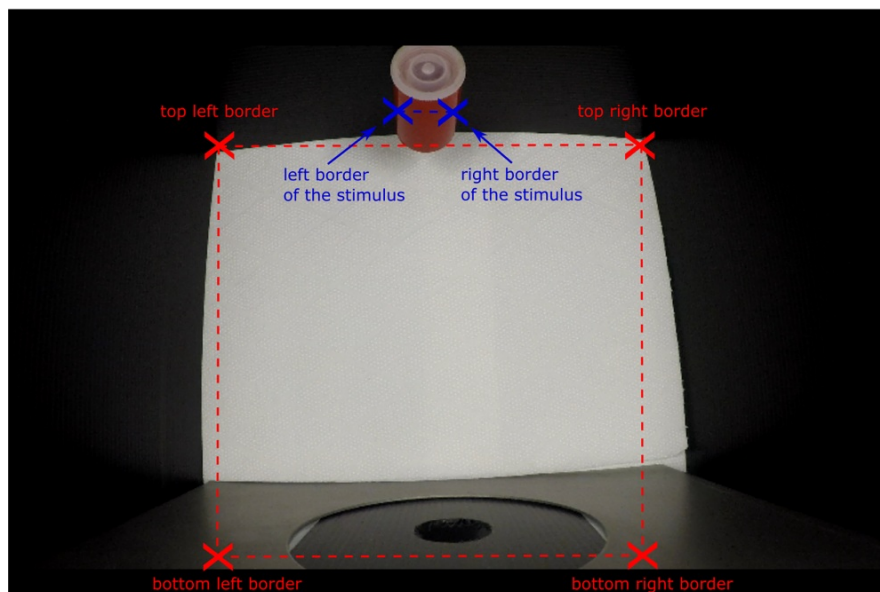


Figure 9: Definition of the borders of the arena and the stimulus, in our example. Please note that the borders of the arena include all the portions of the video in which the animal can appear, which in this case also include part of the bottom wall of the arena itself, due to the visual camera angle.

The precise location of the borders is particularly important to compute how much time the animal spends in different areas of the arena (Figure 7) but does not matter if you are only interested in the visual fields used by the animal to look toward a stimulus. This step creates the first output file, which will automatically be generated within a new directory called 'files', located inside the folder where DeepLabCut files are stored. This output contains the information provided in input 3 and the exact positions (in pixels) of the borders defined at this step. The location of the borders will be used in step 3 to assess the time spent in different areas. The location of the stimuli will then be used in step 4 to assess eye-use.

Step 2:

Check for outliers. At this step, the program checks for errors and computes the distance between pairs of labels. Doing so, it selects outlier frames according to the threshold previously indicated. In our example, we chose a threshold of 3. With this threshold, only 1.33% of the frames were counted as outliers and excluded from the analysis. These frames can be visualized to address the accuracy of this process. In Figure 10, we report two frames that were removed in this step. In both cases, we can see that the labels were not correctly placed on the head of the animal. If the selected threshold is not satisfying, it should be changed running the program again. With this procedure, it is possible to find the most appropriate threshold to each experimental condition.

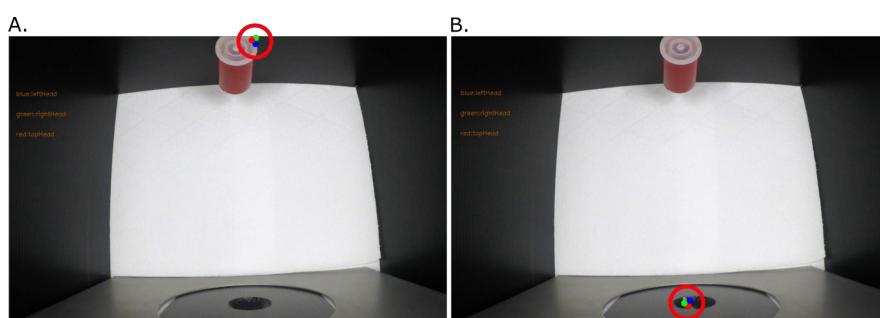


Figure 10: Two frames considered as outliers in our example, with a threshold of 3. The red circles on the images indicate the position of the labels. On image A, the chick has not placed its head inside the round opening yet, but DeepLabCut incorrectly placed the 'leftHead' (blue dot), 'topHead' (green dot) and 'rightHead' (red dot) on an empty portion of the screen, close to the stimulus. On image B, the chick started to insert its head in the round opening, but most of it is still invisible. DeepLabCut incorrectly located the 'leftHead', 'topHead', and 'rightHead' labels on the animal's beak.

Step 3:

The program also automatically excludes the frames from the analysis where the likelihood ratio reported in the DeepLabCut tracking file (Figure 4) is lower than 0.9. If the option selected is “moving stimulus tracked by DeepLabCut”, all frames where the stimulus is absent are included inside this percentage. At the end of the analysis, the percentage of frames excluded due to this criterion will be reported in the analysis output, with and without the stimuli outliers. If too many frames are excluded, it is probably better to improve the DeepLabCut tracking by refining the labels (see DeepLabCut process, Nath et al. 2019) and/or optimizing the video recordings (modifying brightness or contrast for example).

Step 4:

At this stage, the program offers you the possibility to visualize the projection lines of the visual fields on random frames, to control that the program works appropriately and accurately (Figure 11). Using the visual fields defined previously (Figure 7) and the location of stimuli, the program will assess in which hemifield the stimuli fall in each frame. Besides, the software will also assess whether the stimuli fall in the frontal or lateral portion of each hemifield. If the stimulus(i) is located within a visual field, a value of 1 will be attributed to it (see the light-green dash line on Figure 11.A). If the stimulus is straddling on two visual fields, the proportion of the object located within each visual field is attributed to each one of them (see the light-green dash line on Figure 11.B). The output for eye-use data varies depending on the location and number of stimuli. If there is one stimulus on each side of the arena, Visual Field Analysis computes eye-use for both stimuli. However, if there is only one stimulus on one side of the arena, Visual Field Analysis computes eye-use for this stimulus, but also for the corresponding empty spot on the opposite side of the arena (see the dark-green dash lines in Figure 11). This data can be used as a control representing eye-use to monitor a neutral region of space or simply disregarded.

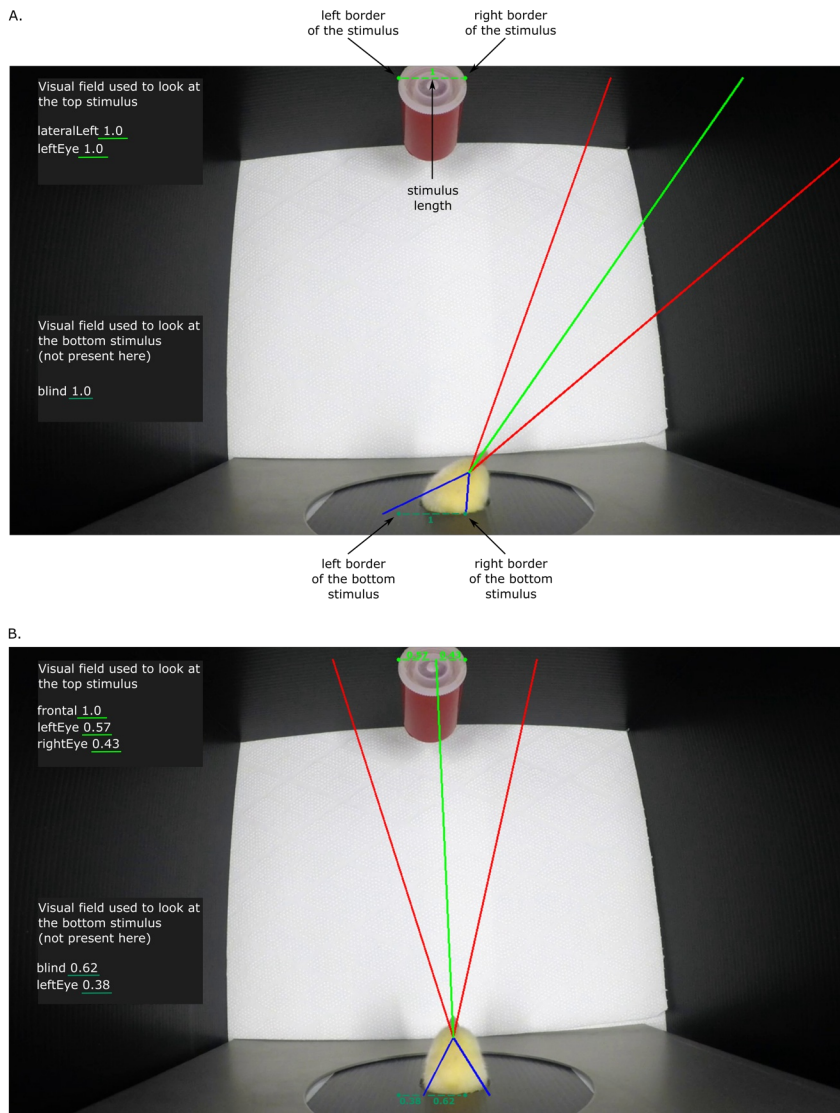


Figure 11: Visualisation of the projection lines defining each region of the visual field. The green line indicates the midline and delimitates each hemifield, providing the nasal margin of the frontal visual fields. The red lines delimitate the frontal visual field from the lateral visual fields. The blue lines delimitate the lateral visual fields from the blind spot. In these pictures, we can see which visual field is used to look at the stimulus, and we can compare this to the information reported on the left side of the pictures (within the dark rectangles on top and bottom left corners of the images). On the top of each image, the visual fields used by the animal to look at the top stimulus are reported. On the bottom of each image, the visual fields used by the animal to look at the bottom stimulus are reported (can be ignored in this example). A value is assigned to every visual field, indicating whether the stimulus was located inside it. A value of 1 for a given visual field, indicates that the stimulus is entirely located within that visual field, such as in figure A. However, the stimulus can be straddling into two visual fields such as in figure B. Consequently, the program attributes different values depending on the portion of the stimuli extent (light-green dash lines) located in a visual field (the stimulus extent is here defined by its borders, as set in stage 3, step 1). For example, if half of the stimulus is located within the frontal visual field and the other half within the lateral visual field, a value of 0.5 will be attributed to both. In figure B, the application tells us that 0.57 of the stimulus is located in the left frontal visual field and 0.43 in the right frontal visual field.

Step 5:

For each video, the program produces a second output, located within a new directory called 'results' which is inside the folder where DeepLabCut output files are stored. This output includes all the information specified within the input 3 and the behavioral measurements obtained by Visual Field Analysis (see Figure 12).

- The first columns of this output correspond to the columns contained in input 3.
- The following column, named 'distanceMoved' provides information concerning **the activity level of the animal**, operationalized as the total distance covered of the 'topHead' label. In our example, this measurement gives direct information about head movements done by the chick, since it was standing in a fixed position and only its head was moving. Instead, if the animal observed is moving freely within an arena, this value corresponds to the distance moved by the animal in the environment,

plus its head movements.

- The next five columns provide **the time spent**

in each area frame by frame (Figure 12A) or by seconds (Figure 12B) depending on your previous choice. In our main example, this information is not meaningful since the animal could not move across the arena, which was thus not subdivided into different areas. In Figure 12.B, we provide an additional example of this kind of data. To do so we report the output produced by Visual Field Analysis for one chick tested for its preference for two moving stimuli placed at the opposite ends of an arena, subdivided into five zones defined as in Figure 7.

- The following columns provide **eye-use measurements**

Depending on the orientation of the apparatus (left/right or top/bottom) the column names will be different. The last word of the column name always indicates the position of the stimulus. In our main example, we had only one stimulus located on the top so we should focus only on the columns having the extension 'top' written at the end of the column name (columns with the extension 'bottom' should be ignored in this case, since they refer to the 'ghost stimulus' located at the bottom of the apparatus and are not meaningful in this context). In Figure 12, values above zero in 'frontaltop' indicate that the stimulus (or part of it) was located in the frontal field of the animal. Values above zero in 'blindtop' indicate that the stimulus was not seen by the animal. Values above zero in 'lateralLefttop' and 'lateralRighttop' indicate that the stimulus was located within the left or right lateral visual field of the animal, respectively. Values in 'leftALLtop', 'rightALLtop' indicate which hemifield was used to look at the stimulus (frontal and lateral visual fields pulled together, see Figure 12.A).

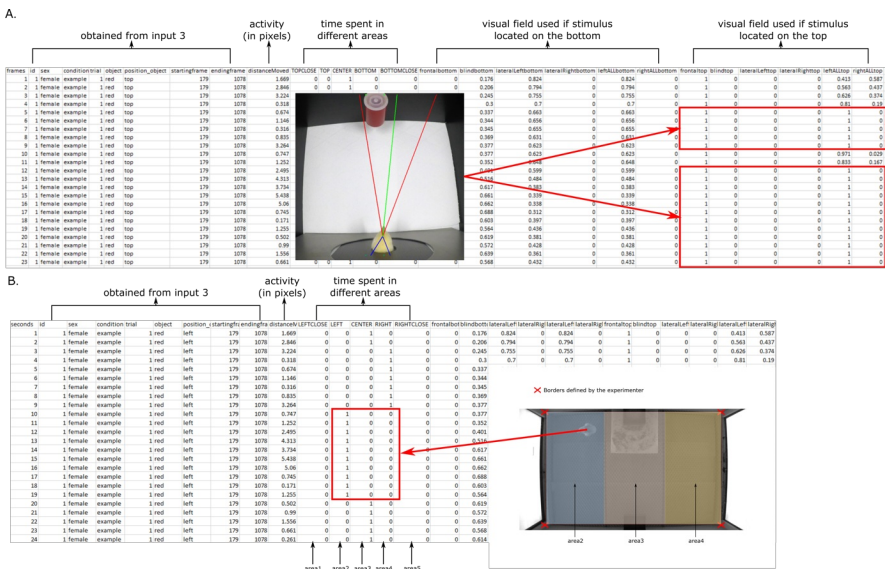


Figure 12: For every video analyzed, one .csv output file will be produced. This output will report all the information previously entered in input 3 and the behavioral measurements scored by Visual Field Analysis. Figure A shows an annotated screenshot of the output produced by VFA and illustrates where the data concerning eye-use is located (frame by frame analysis). Figure B illustrates where the data concerning the location of the individual is located in an arena divided into 3 equal sub-areas. Each row corresponds to one second of the video, while five different columns represent the presence of the animal in the corresponding area. A value of 1 indicates the presence of the animal in the corresponding area in the given second, whereas a value of 0 indicates its absence. In this additional example, only three areas had been defined; thus 'area1' and 'area5' will always contain null values. We have highlighted in red cells that correspond to the situation visible in the two video frames overlaid on the picture (A stimulus located on the frontal left visual field; B chick located inside Area 2).

Our application is entirely open-source and freely available on [GitHub](#).

Visual Field Analysis may evolve and improve with time. All further versions will be made available to the scientific community on our [GitHub](#).

Reference

Nath, T., Mathis, A., Chen, A.C. *et al.* Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nat*

