Readme

1) Eye Tracking Study

- Start the eye tracking glasses by turning the module on
- Turn your laptop on and start the Tobii Pro Controller Software
- Turn on the Wifi of your laptop and connect to the Wifi connection of the Tobii Pro Eye Tracking glasses
- Start the eye tracking and calibrate the glasses using the calibration card
- Start the eye tracking recording / end the eye tracking recording

➔ For further help with these step, please see: https://www.youtube.com/watch?v=Gl_UVvFTuxU

2) Export Eye Tracking Study

- Download the eye tracking study recording in the Tobii Pro Lab software (this is the video-file)
- Close Tobii Pro Lab
- Turn off the eye tracking glasses
- Extract the SD card from the Tobii Pro Glasses module and Insert into your laptop
- Open Tobii Pro Analyzer Software
- Start a new project
- Press "import" and pick the option "Import from Eye Tracking Glasses Recording" and choose the recording you just recorded
- Pick the relevant metrics you want to export (Example: gaze point, event type, etc.)
- Export the data as .csv-file
- Close Tobii Analyzer Software

➔ For further help with these step, please see: https://www.youtube.com/watch?v=Gl_UVvFTuxU

3) Processing of Eye Tracking Data

- Open the exported .csv-file in Excel
- Erase all data that was wrongly exported but is not necessary, such as for example device type. Finally delete the data where the gaze point is missing and all events not classified as fixations by using the Excel filter function
- Rename the data columns with the relevant data to Event Start Trial Time [ms], Event End Trial Time [ms], Event Duration [ms], Visual Intake Position X [px], Visual Intake Position Y [px] and save the file. Be careful to use exactly these titles for the columns. Only the data according to these 5 column categories is relevant for further processing, thus all columns that do not contain start, end, duration and pixel coordinates can be deleted

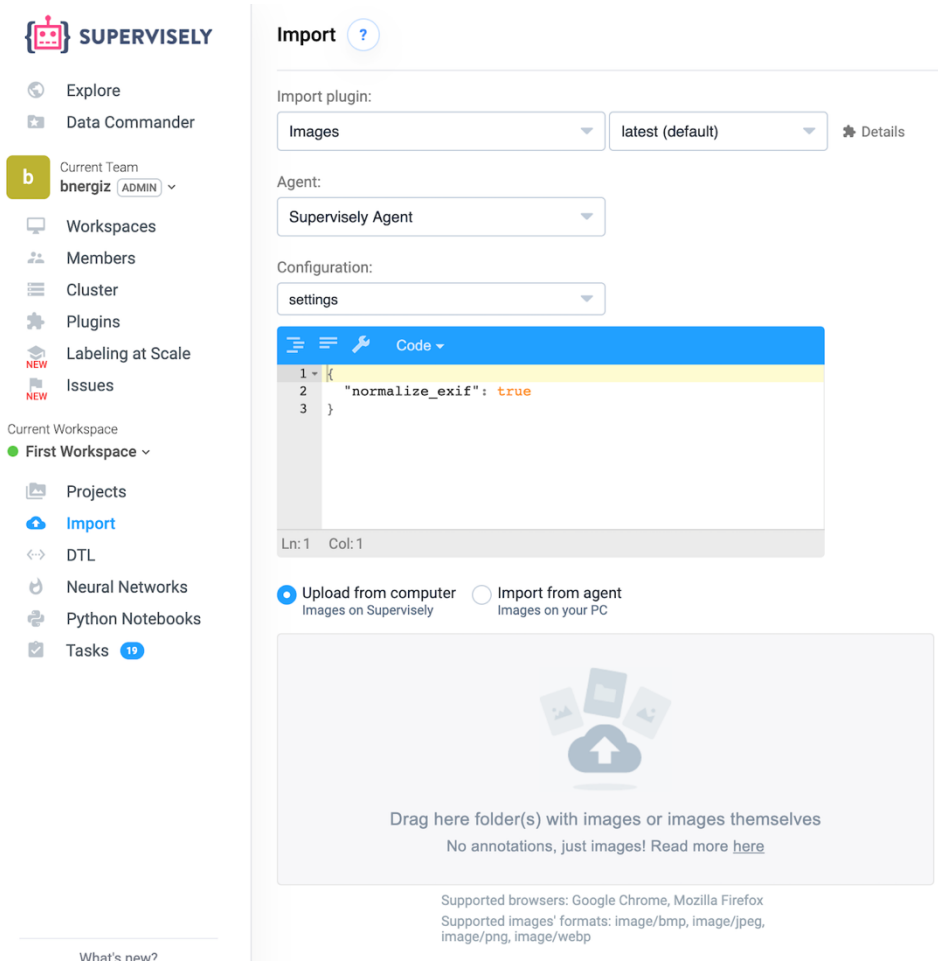
- Copy the .csv-file into the directory where the python-code "process_data_2.py" is (Example: ../cgom_yolo/processing_gaze_data/project_hilti_final) and either rename it to "hilti_project_raw.txt" or change the expected name for the input file in the code
- Open your terminal and go to the same directory (Example: ../cgom_yolo/processing_gaze_data/project_hilti_final)
- Run the command "python process_data_2.py" in your terminal. This will process the data and create new files containing the relevant processed data: the start point of each fixation, the end point of each fixation, the duration of each fixation, the x pixel coordinate and the y pixel coordinate of the gaze point. Each entry now summarises the data contained in each entire fixation. This was previously stored with a 100Hz frequency (100 entries per second and usually multiple entries per fixation) and is now summarised
- Open a new excel sheet and create the columns named exactly as follows from left to right (Event Start Trial Time [ms], Event End Trial Time [ms], Event Duration [ms], Visual Intake Position X [px], Visual Intake Position Y [px]). These should be the same column names as in the previous file.
- Open the newly created file "x_hilti_project_raw.txt" and copy the data into column Visual Intake Position X [px]
- Open the newly created file "y_hilti_project_raw.txt" and copy the data into column Visual Intake Position Y [px]
- Open the newly created file "start_hilti_project_raw.txt" and copy the data into column Event Start Trial Time [ms]
- Open the newly created file "end_hilti_project_raw.txt" and copy the data into column Event End Trial Time [ms]
- Open the newly created file "duration_hilti_project_raw.txt" and copy the data into column Event Duration [ms]
- Save the excel sheet as a .txt-file and close the file

4) Cut Images out of Video

- Move the gaze file (.txt) to the directory ../cgom_yolo/gaze
- Move the video file (.avi) with the eye tracking experiment video to the directory ../cgom_yolo/videos
- Attention ! The file name of the video and the file name of the gaze data have to be the same (example: gaze data: experiment.txt and video data: experiment.avi)
- In your terminal go to the directory ../cgom_yolo/misc
- Run the command "python extract_images_from_video.py" in your terminal and use the number of images you want to extract and the directory of the video as arguments
- Now the newly created images will be in the folder ../cgom_yolo/data_sets and can be used for further processing

5) Import to Supervisely

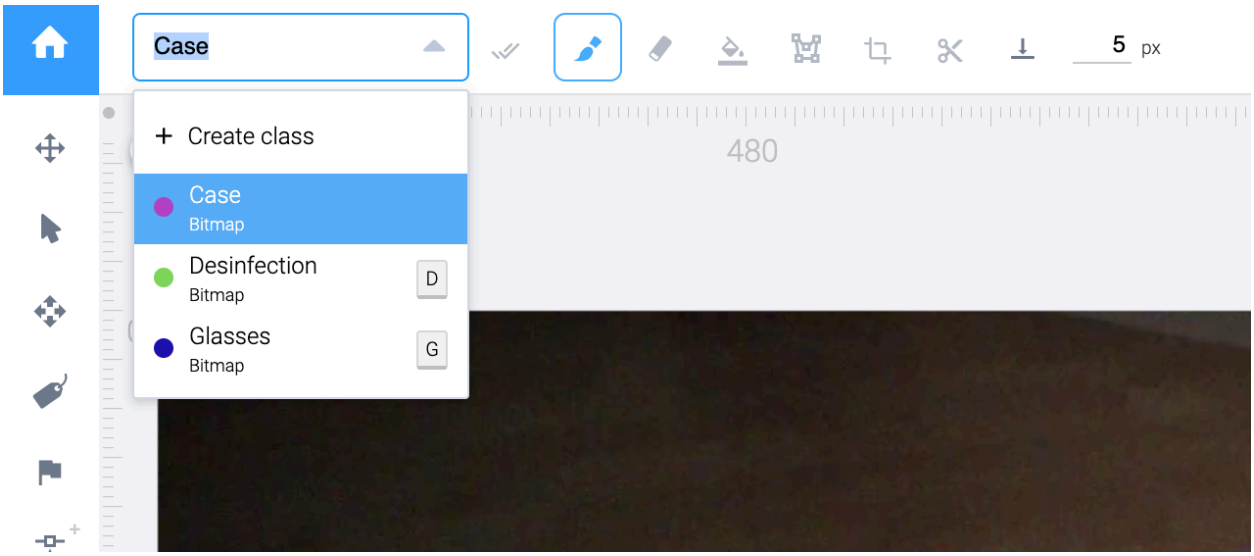
- Open the Supervisely website and Login with your previously created credentials
- Go into import project



- Drag and drop the folder containing the images you want to import
- Save the project with a correct name

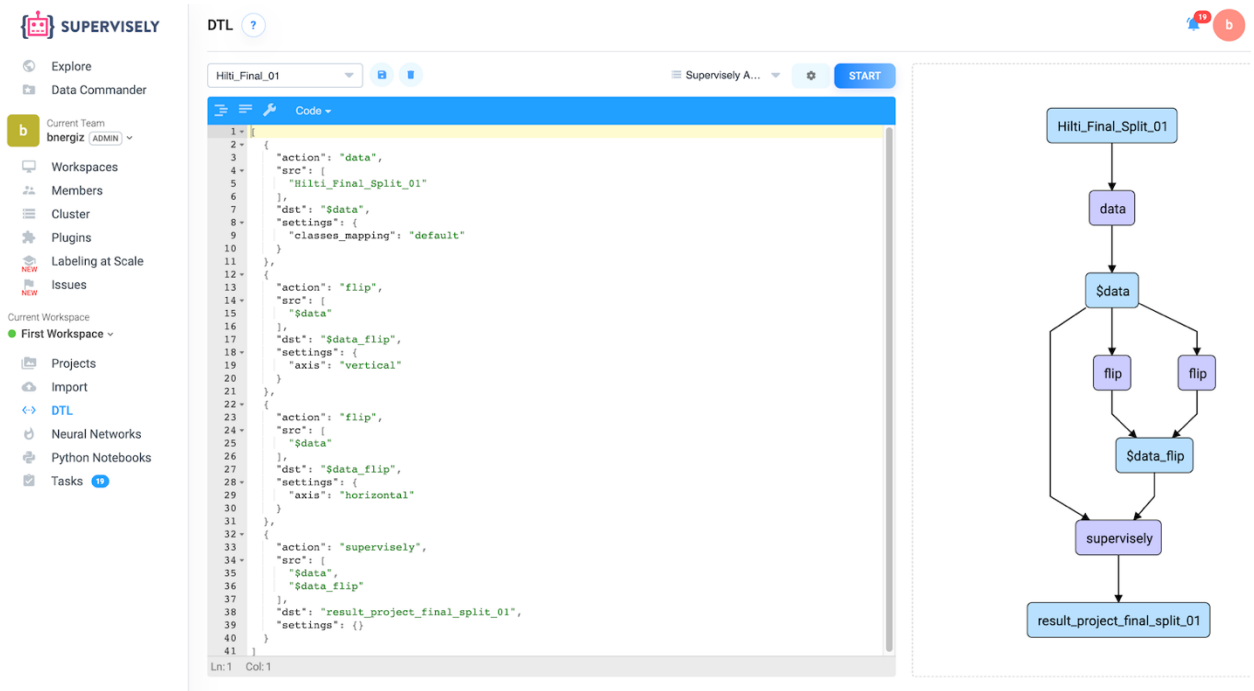
6) Data Labeling

- Open the project and create a new class for each class type you want to label
- Go through all the images and label them manually using the Supervisely tools



7) Data Augmentation

- Go to the section DTL on Supervisely
- Adapt the standard DTL script to augment the data in a way you want (example: vertical and / or horizontal flipping)
- Run the DTL script on your labeled data
- A new project file should be saved on Supervisely with the augmented data



8) Training CNNs

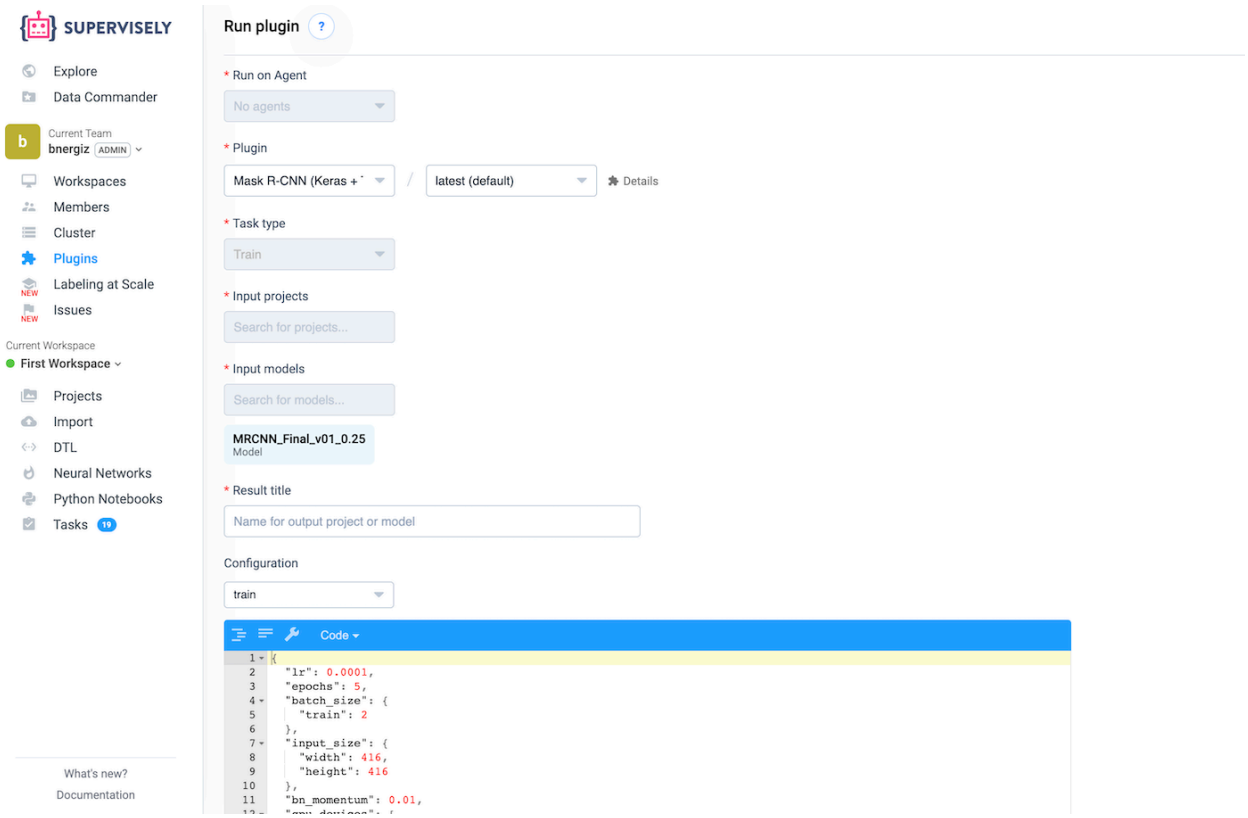
- Open the website of Amazon Web Services and login with your previously created credentials
- Launch a new instance on Oregon
- Link to AWS with Supervisely

➔ Here is a step by step guide on how to do this:
https://legacy.docs.supervise.ly/cluster/ami/

As a summary:

- *In the search bar, insert "Supervisely" and select the instance Supervisely-P2 (NOT Supervisely agent)*
- *Click through the instance creation steps and launch the instance*
- *Once the instance is launched, in your terminal move to the directory containing your AWS key that you previously downloaded and connect to it by running the connection command in your terminal*
- *Once the instance is connected to your computer, move to the section "Cluster" on Supervisely and connect AWS to Supervisely*

- Once AWS is set up as a computational cluster for Supervisely, the Training can be started
- First, go to "Neural Networks", then click on "ADD" and pick a neural network
- A clone of this neural network should be saved and can now be trained
- Go on "Neural Networks " and click on "Train" and select the augmented project with the labeled data
- Run the training with the settings you want
- Once the training is conducted, a newly trained CNN with the name you chose should be saved under "Neural Networks"



9) Running the CNNs using cGOM

- Download the weights of the trained CNN from Supervisely by clicking on the vertical 3 dots next to the neural network and then pressing “Download”
- A folder containing the config file and the weights should be downloaded
- Insert the weights in the directory /cgom_yolo/toolbox_v02/weights and rename the weight file to “weights.sh”
- Insert the config file (model.cfg) in the directory /cgom_yolo/toolbox_v02/cfg
- In your terminal, move to the folder “toolbox_02” and run the command “python3 make_gaze.py” in your terminal (ATTENTION ! Python3 has to be used for this)
- This should start cGOM and create the files of interest in the directory /cgom_yolo/outputs/images/hilti_project (in the case where video and gaze file are called “hilti_project”)