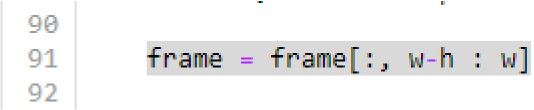
We modified mouse tracker program posted on YouTube and Github (Original link below) for our friend's research video input.
So that the program can work with a different input video.

<Original Author's Posts>
YouTube:
https://www.youtube.com/watch?v=GebcshN4OdE&feature=youtu.be

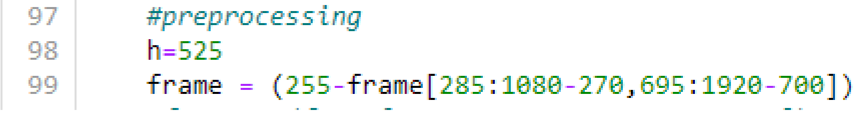
Github:
https://github.com/colinlaney/animal-tracking

Major modification is summarized as follows.
1.Input preprocessing
Dimension of preprocessed each frame has been modified because our input has a different frame dimension.
e.g.
from: frame = frame[:, w-h : w] #Original Preprocessed frame



```
    frame = frame[:, w-h : w]
```

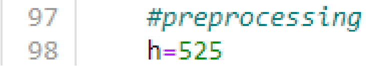
to: frame = (255-frame[285:1080-270,695:1920-700]) #Modified Preprocess frame.



```
    #preprocessing
    h=525
    frame = (255-frame[285:1080-270,695:1920-700])
```

Note: Since mouse in our video input is black, we need to invert the color of the mouse in each frame.
This is why there is '255' term in the modified code.
Furthermore, to make object tracking task easier, we manually found optimized dimension of frame that will be used during tracking.
e.g.frame[285:1080-270,695:1920-700] : chosen frame from a whole big frame which contains unusful information for the tracking task.

2.Manual height of frame modification.
e.g. h=525



```
    #preprocessing
    h=525
```

If you take a look at the original python script closely, we need to adjust variable 'h' whenever our proprocessing on each frame changes row dimension of the frame
because this h is utilized for cv2.warpPerspective(frame, perspectiveMatrix[name], (h,h)) (line 151).
In our practice, we choose height (a.k.a 'h') as 525.

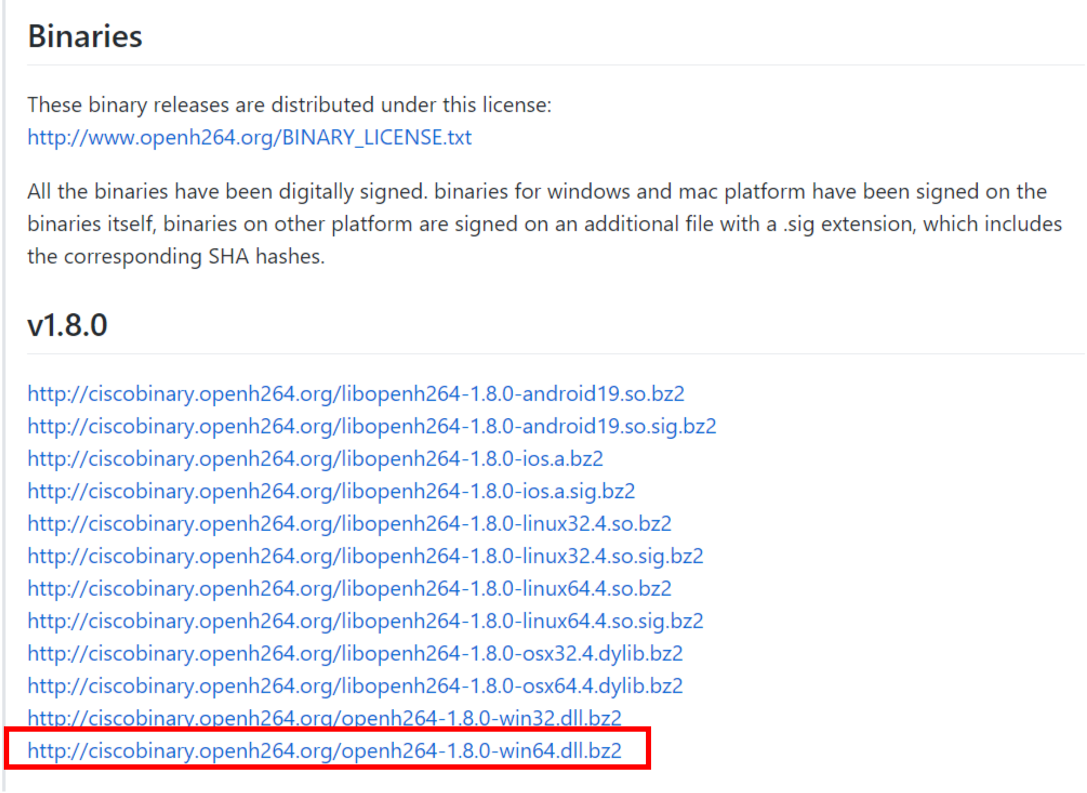
3.Removal of background subtraction step.
Unlike the original script, we removed background subtraction step e.g. cv.subract(frame, background) (line 222).
This is simply because we get a proper processed data for tracking task. This part will be needed to be modified for universal application. i.e., any video input

Note: For the other major steps and software requirments, please refer to the origial post on Github.

<How to run the program>

-Make sure you have installed ‘openh264-1.8.0-win64.dll’. It can be downloaded on the following website. Scroll down to find the proper version.

https://github.com/cisco/openh264/releases



Binaries

These binary releases are distributed under this license:
http://www.openh264.org/BINARY_LICENSE.txt

All the binaries have been digitally signed. binaries for windows and mac platform have been signed on the binaries itself, binaries on other platform are signed on an additional file with a .sig extension, which includes the corresponding SHA hashes.

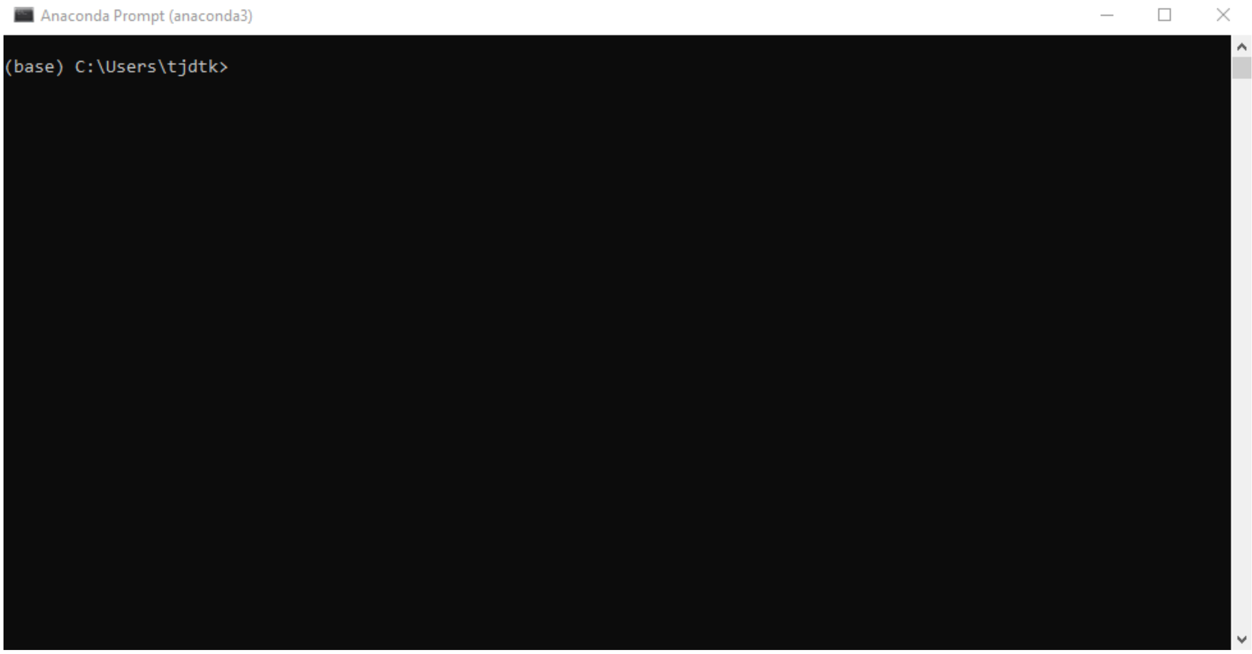
v1.8.0

http://ciscobinary.openh264.org/libopenh264-1.8.0-android19.so.bz2
http://ciscobinary.openh264.org/libopenh264-1.8.0-android19.so.sig.bz2
http://ciscobinary.openh264.org/libopenh264-1.8.0-ios.a.bz2
http://ciscobinary.openh264.org/libopenh264-1.8.0-ios.a.sig.bz2
http://ciscobinary.openh264.org/libopenh264-1.8.0-linux32.4.so.bz2
http://ciscobinary.openh264.org/libopenh264-1.8.0-linux32.4.so.sig.bz2
http://ciscobinary.openh264.org/libopenh264-1.8.0-linux64.4.so.bz2
http://ciscobinary.openh264.org/libopenh264-1.8.0-linux64.4.so.sig.bz2
http://ciscobinary.openh264.org/libopenh264-1.8.0-osx32.4.dylib.bz2
http://ciscobinary.openh264.org/libopenh264-1.8.0-osx64.4.dylib.bz2
http://ciscobinary.openh264.org/openh264-1.8.0-win32.dll.bz2
http://ciscobinary.openh264.org/openh264-1.8.0-win64.dll.bz2

-Make sure you locate an input video file into the same folder where your python script is located.

-Open Anaconda Prompt (Command Prompt may work as well)

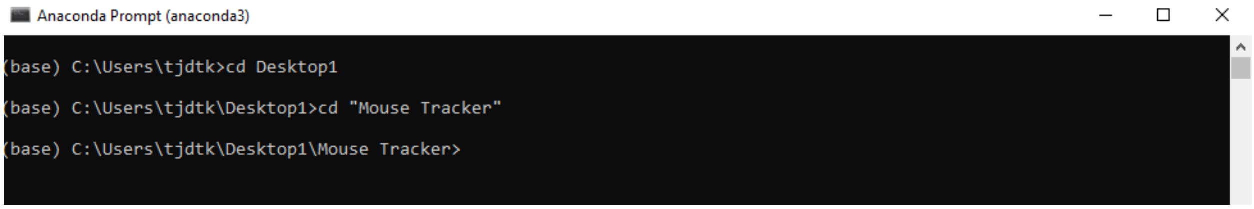
e.g.



```
(base) C:\Users\tjdtk>
```

-Navigate to the folder where you keep the python script you want to run.

e.g.



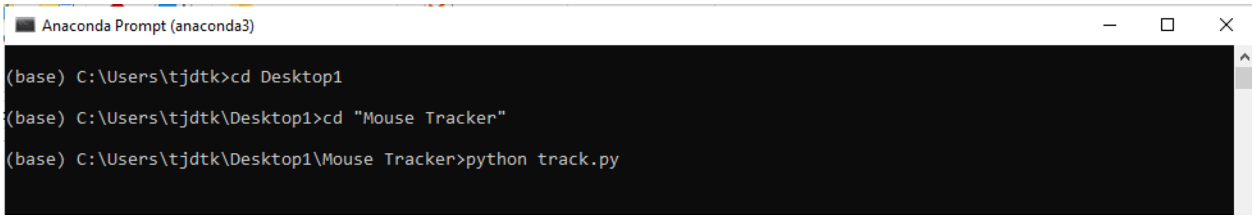
```
(base) C:\Users\tjdtk>cd Desktop1

(base) C:\Users\tjdtk\Desktop1>cd "Mouse Tracker"

(base) C:\Users\tjdtk\Desktop1\Mouse Tracker>
```

-Run the python script

e.g.

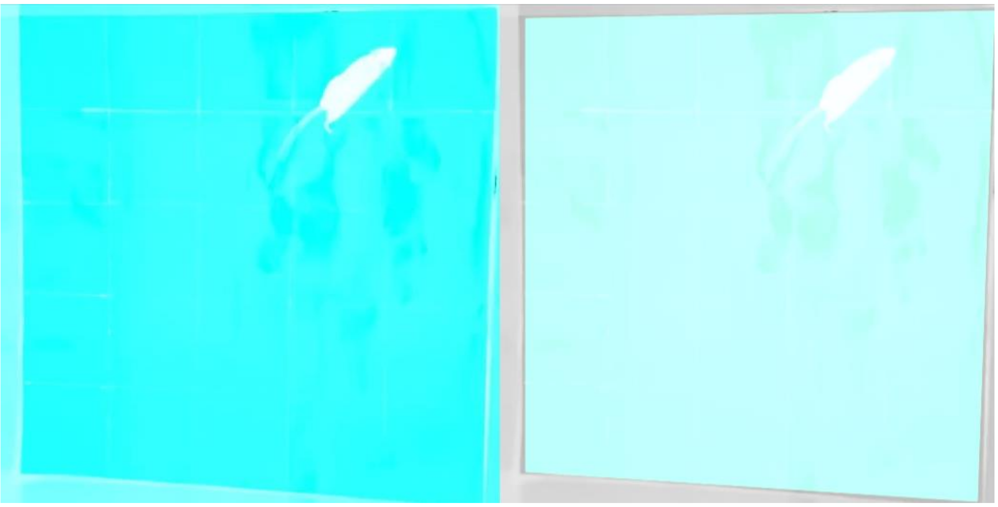


```
(base) C:\Users\tjdtk>cd Desktop1

(base) C:\Users\tjdtk\Desktop1>cd "Mouse Tracker"

(base) C:\Users\tjdtk\Desktop1\Mouse Tracker>python track.py
```

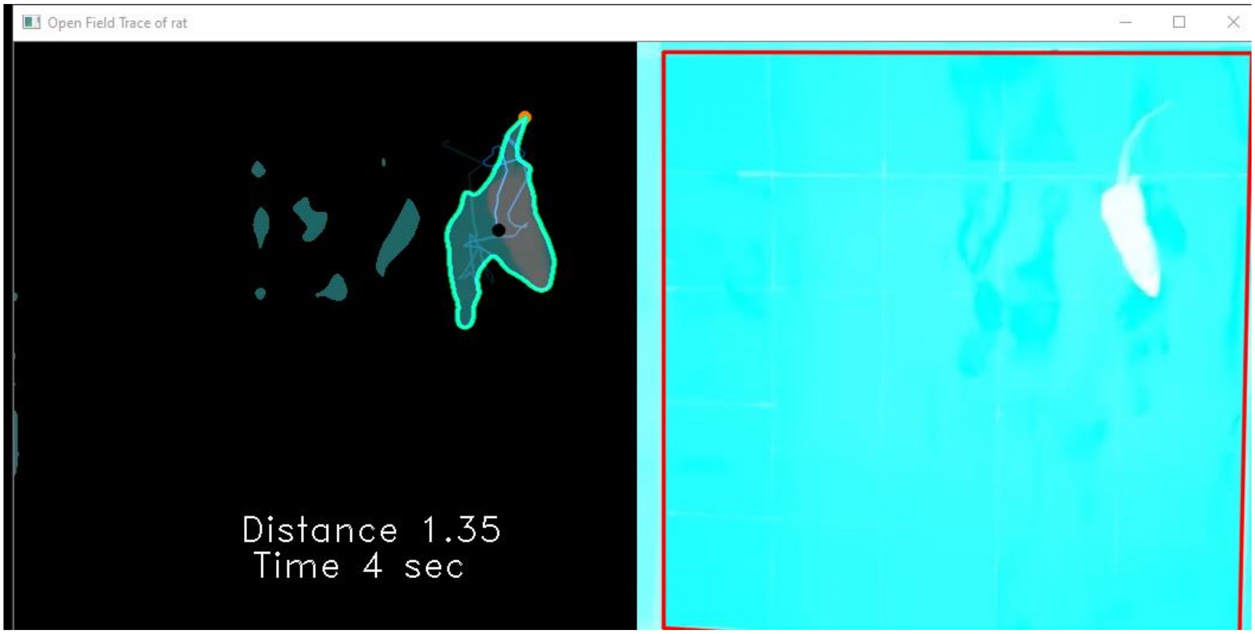
-Once you run the script, a user interface will show up. Select a square region that a mouse will travel around. Please take a look at the video attached for more details.



Note: Red line represents selected region.

-Right click. The user interface will disappear. After 10~20 seconds, new user interfaces will appear. Find a window with title with ‘Open Field Trace of rat’.

e.g.



```
Distance 1.35
 Time  4 sec
```

-Wait until the processing will end.

<Note>

-Since there are two pause moments for the rat in input video, there are two “freeze” moments during tracking process as well. This is not a flaw of the tracking algorithm.

-For the input file used in our project, please contact the email

samwoose@usc.edu

<Video on running program>

https://youtu.be/8TauFjFflss