```
In[∘]:= (***Input identifying information***)

In[∘]:= date = ToString[Evaluate[Input["Input the date of the experiment"]]]

Out[∘]= 071223

In[∘]:= mouse = ToString[Evaluate[Input["Input the mouse identity (e.g. Mouse123)"]]]

Out[∘]= Mouse23666

In[∘]:= sessionNum = Evaluate[Input["Input the session number"]]

Out[∘]= 2

        (***Get the names of all the tiff files in the folder.  The tiff files should
         be images of the eye, cropped from the full video of the mouse's face***)
        fileNames = FileNames["*.tif", StringJoin["S:/Imaging/Garrett/FMB208_2PRig/",
            date, "/", mouse, "/Session", ToString[sessionNum], "/", date,
            "_", mouse, "_Session", ToString[sessionNum], "_Eyes/"]];

In[∘]:= (***Import all the tiff files, and join them into one list***)
        rawImages = Catenate[Import /@ fileNames];

        (***Visualize the tiff files to both create a mask that blacks out
         all parts of the frame except for the parts that contain the pupil,
        and to delineate a range of pixel intensity values associated with the pupil.  Copy
         one of the image frames and paste into notebook.  Click on image.  Go to the
         mask tool and choose "Freehand shape".  Once finished completing the mask,
        go to "Copy"→"Mask as image".  Finally, paste this image in "mask="***)

In[∘]:= Manipulate[rawImages[[n]], {n, 1, Length[rawImages], 1}]

        mask =;

        (***Specify the range of grayscale values that best encompass the pupil***)

In[∘]:= pupilRange = {0.9, 1};

In[∘]:= (***Binarize the images and mask out the pixels that don't contain the pupil***)
        binarizedImages =
          Table[Binarize[Binarize[ImageMultiply[rawImages[[n]], mask], pupilRange]],
           {n, 1, Length[rawImages]}];

        (***Inspect the binarized pupil video series***)

In[∘]:= Manipulate[binarizedImages[[n]], {n, 1, Length[binarizedImages], 1}]

In[∘]:= (***Perform edge detection on the binarized images.  For the largest shape
         formed by edge detection (i.e. the shape with the maximal authalic radius,
          which should correspond to the pupil),
        find the semi-axes of the best-fit ellipse to the detected edges,
        as well as the coordinates of the center of
         mass of the detected edges (this is pupil position)***)

        pixelRange = 13;

        detectedEdges =
          Table[EdgeDetect[binarizedImages[[n]], pixelRange], {n, 1, Length[binarizedImages]}];
        (***Inspect the quality of the edge detection,
        which is red overlayed on the pupil video series.  Adjust the pixel range as needed***)

In[•]:= Manipulate[Blend[{rawImages[[n]], ColorReplace[detectedEdges[[n]], White → Red]}],
         {n, 1, Length[rawImages], 1}]

In[•]:= pupRadAndListPos =
          Table[list = Values[ComponentMeasurements[detectedEdges[[n]], "AuthalicRadius"]];
           {Max[list], Flatten[Position[list, Max[list]]][[1]]}, {n, 1, Length[detectedEdges]}];

In[•]:= pupPos = Table[Values[ComponentMeasurements[detectedEdges[[n]], "Centroid"]][[
            pupRadAndListPos[[n, 2]]]], {n, 1, Length[detectedEdges]}];

In[•]:= (***Import pupil camera frame times***)
        syncTimes =
          Drop[Drop[Import[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/", mouse,
               "/Session", ToString[sessionNum], "/", date, "_", mouse, "_Session",
               ToString[sessionNum], "_CamSync.txt"], "List"], 16], -1];

        (***Calculate camera frame rate***)
        sampRate = Round[Length[syncTimes] / (Last[syncTimes] - First[syncTimes])];

        (***Verify that length of syncTimes = length of tiff series, +/- 2***)

In[•]:= Length[syncTimes]

In[•]:= Length[rawImages]

In[•]:= pupRadTimeSeries = Partition[Riffle[syncTimes, Part[#, 1] & /@ pupRadAndListPos], 2];

In[•]:= pupPosTimeSeries = Partition[Riffle[syncTimes, pupPos], 2];

In[•]:= (***Export data***)

In[•]:= Export[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/", mouse,
            "/Session", ToString[sessionNum], "/Pupil/", date, "_", mouse, "_Session",
            ToString[sessionNum], "_PupilRadiusTimeSeries.txt"], pupRadTimeSeries];

In[•]:= Export[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/", mouse,
            "/Session", ToString[sessionNum], "/Pupil/", date, "_", mouse, "_Session",
            ToString[sessionNum], "_PupilPositionTimeSeries.txt"], pupPosTimeSeries];

In[•]:= Export[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date, "/", mouse,
            "/Session", ToString[sessionNum], "/Pupil/", date, "_", mouse, "_Session",
            ToString[sessionNum], "_PupilRadListPos.txt"], pupRadAndListPos];

In[•]:= Export[StringJoin["S:/Imaging/Garrett/FMB208_2PRig/", date,
            "/", mouse, "/Session", ToString[sessionNum], "/Pupil/", date, "_",
            mouse, "_Session", ToString[sessionNum], "_PupilPos.txt"], pupPos];
```