TeleMedC Pupil detector

I initially spent some time researching methods online to solve this problem. Some of the most common seemed to be using a HaarCascade classifier and Hough circles. But I also found people getting success by using thresholding followed by finding contours. To get an initial result I decided to go down this latter path.

I found that by using thresholding and by appropriately adjusting the threshold parameter, we could obtain good results in isolating the pupil in the frame (I first converted the frame to greyscale to allow for thresholding and also added some Gaussian blurring to reduce some noise in the frame). From here, we could find the contours in the frame and it would successfully detect the pupil outline, along with several other objects in the frame.

One of the objects that was often detected was on the boundary/edge of the frame. I noticed this boundary remained constant (I assume it's related to the camera used to capture the video), and could potentially lead to false detections, so I cropped out the outer ~10% of the frame to remove this boundary.

By this point, the pupil contour was detected, along with smaller contours, mostly small blobs inside the pupil itself (eye glints/light reflections). I tried using morphological transformations like erosion and dilation to get rid of them, but the parameters to successfully remove the small blobs also seemed to cause the pupil's contour shape to lose its integrity and get somewhat distorted, so I abandoned this approach. I saw an example online of filtering contours based on its size/area. Since at this stage the largest contour being detected was the pupil itself, I decided to use that approach, and only work with the largest contour in the frame.

Once we had the largest contour, I applied the convex hull function to smoothen the contour, and I then used the moments function to calculate the center of mass of the contour, which would be the estimate of the centroid of the pupil, and also fitted an ellipse to the contour, which would be the outline of the pupil itself. I had initially drawn a circle instead of an ellipse, and the results are very similar for this sample video. But I thought that the ellipse would be a more robust result if the eye happened to be at more of an angle (looking away from the camera), as then the pupil shape on the frame would no longer be circular. I drew the centroid and pupil outline on the frame, and this was my working solution for the problem. The results looked good to me, as the pupil was consistently detected, without falsely detecting other areas of the frame, and the centroid also looked to be in the appropriate position without jittering much. The frames that don't look so good are when the eye is blinking, as the contour shape gets distorted, and the pupil outline is noticeably larger or smaller than the portion of the pupil that remains in frame. But I hope that's not a big issue functionally, as the pupil outline looks appropriate again after the eye blink has completed.

Other things I tried/things to potentially try

- Hough circles: I tried using this after the thresholding step, when the pupil shape was fairly distinct, but it couldn't detect any circles. I tried a few different parameters

and couldn't get it to work. Perhaps with more parameter tuning it could work, but to save time I abandoned this approach.
- SimpleBlobDetector: I tried using this blob detector, but it seemed to be better at detecting the small blobs inside the pupil (glint/reflections) than the pupil itself, so I also abandoned this approach.
- I read about HaarCascades, but many people seemed to suggest it didn't work too well, and some suggested it was slow, so I didn't try this for the sake of time.
- This wasn't needed for the sample video provided, but if some of the other contours were as large or larger than the pupil itself, we could potentially filter the contours based on measuring the circularity of the contour (using a function of the area and perimeter of the contour), so that we only use contours that are approximately circular in the frame, which should hopefully isolate the pupil again.

In terms of measuring accuracy, as I mentioned my email to Seth, it seems to me that any robust notion of accuracy would require knowing the ground truth to measure against. For the sample video, I think the detection rate meets the 80% threshold just by looking at how well the centroid and ellipse drawn on the frame tracks the actual pupil as it moves throughout the frames (except during blinking). If we did have the ground truth, I would probably measure accuracy by something like "accurate if the estimated center is at most a 3-pixel radius from the true center".

This doesn't apply to the sample video, but we could potentially apply some heuristics to measure roughly how well we're doing, such as: the pupil outline is accurate if most of the pixels inside it are darker than the pixels outside it. Moreover, we could also assume that the true center of the pupil should not differ much between frames (allowing some slack for natural eye movement). So, if the position of the estimated centroid changes drastically between frames, it's probably a sign of a false detection as we're now looking at a different contour/object than we were in the previous frame, and we could potentially filter that new contour out. This assumes of course that the previous estimated contour is accurate, so it probably is only useful if we find false detections being introduced in the middle of the video, and not initially.