Explanations of Algorithms

1. Functions

   The program is consists of 3 main functions:

   parameters(videoInputName):

   If a video is opened it gets its dimensions and frame rate so we can use that information on display() method to determine the proper flow of video in terms of time and also to create a window that has the right dimensions.

   It also gives an error message if the video is not found.

   findRedCircles(masked_image, image_itself):

   The function accepts 2 parameters one of them is a masked image that is used to capture the circles.

```
circles = cv2.HoughCircles(cimg[:, :, 0], cv2.HOUGH_GRADIENT, 1,20 , param1=12, param2=10, minRadius=1,
                    maxRadius=500)
```

   The masked image is converted to grayscale first.
   "cv2.HOUGH_GRADIENT" defines the method to detect circles in images.
   20 here is the minimum distance between center coordinates of detected circles. If it is too big you may not be able to detect some circles. It should be adjusted according to the camera and situation.
   "param1" is for adjusting gradient values to have proper edge detection.
   "Param2" is the accumulator threshold value. If the threshold value is smaller, more circles will be detected.
   "minRadius" is the minimum size of the radius.
   "maxRadius" the maximum size of the radius.

   The important thing about .HoughCircles() is that it requires adjustment according to the environment, camera, and the wanted features of the circle. If we want to have perfect circles without any interior circles inside of a red area we can adjust param1 and param2 to make that happen. However, in the given conditions it was not the case.

```
cv2.putText(img, 'Red Circle', (i[0], i[1]), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 1, cv2.LINE_AA)
cv2.circle(img, (i[0], i[1]), i[2]+1, (0, 255, 0), 1)
```

For putText() and .circle() method we give the image that we want to process, the center of the circle to put the text, and the boundary around. Also, we pass the font, color, and size values.

display1()

The function can be considered as our main function because it uses all other functions and displays the video which is processes and displays the circles in it.

The function mainly uses cv2.CideoCapture() if you pass 0 to it it means it uses the first camera but if you pass the location of a given video it gets the video.

We use the parameters() function to get the dimensions of the window and fps then store them into the variables.

After we get all the necessary elements we use a while loop to go through all the frames and determine the locations of the circles.

The program starts with converting the color space from BGR to HSV. HSV gives us a more accurate distinction of the colors. So what it does next is generating masks for the color values. You need to determine the intervals for the color you want to mask. In the program, np.array is used. Which is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers.

After creating the mask, the median blur is used to have more accurate edges throughout the images. It softens the images so our edges have softer edges. It helps to understand where is the circle.

The next step is for being able to use HoughCircles(). Our image is blurred and now it is in the grayscale. It is now passed to the findRedCircle() method.

main

```
main
  │
  ▼
display1()
  │
  ▼
 end
```

display()

```
display()
  │
  ▼
get parameters
parameters(captured_video)
  │
  ▼
capture the video ◄───────┐
  │                       │
  ▼                       │
adjust the frame          │
dimentions.               │
  │                       │
  ▼                       │
convert image from        │
BGR to HSV                │
  │                       │
  ▼                       │
apply median blur.        │
  │                       │
  ▼                       │
findRedCircle(cimg,       │
img)                      │
  │                       │
  ▼                       │
returns image ──yes───────┘
  │
  no
  ▼
 end
```

parameters(captured_video)

```
parameters(captured_video)
  │
  ▼
if video is ──────────► get height and width
opened                           │
  │                              ▼
  │                     if height or ──yes──► give an error
  ▼                     width == None                │
return None                      │                   │
  │                              ▼                    │
  │                     return the values             │
  │                              │                    │
  ▼                              ▼                    ▼
 end ◄──────────────────────────┴────────────────────┘
```

findRedCircles(masked image, image)

```
findRedCircles(masked
image, image)
  │
  ▼
find circles with
HoughCircles
under given
parameters
  │
  ▼
if circles is not ──yes──► get the values to find
None                        circle and add
  │                         boundary and text to
  │                         center on image
  │                              │
  ▼                              ▼
 end ◄───────────────────────────┘
```