Read video from computer using moviepy.editor
Trim and crop video to isolate droplet
Resave file to desktop
Read file from desktop using opencv
Define camera parameters

- Resolution
- Frame rate
- Focal length

Begin looping through each frame
Find distance between droplets
Find velocity of droplet
Append frame # with only 1 droplet per frame
Decide whether to use a color threshold or canny edge detection threshold
Find contours using cv2.findCountours

- Decide whether to use RETR_TREE or RETR_EXTERNAL
- Decide with to use CHAIN_APPROX_NONE or CHAIN_APPROX_SIMPLE

Detect droplet count
Loop through generate contours
Draw contours
Use contourArea (greens theorem)
Find image moments
display centroid of droplet
Display area of droplet
Display perimeter of droplet

Key Functions:

- Image = cv2.imread('image name')
    - Reads image from path
- image.copy()
    - Copies image
- cv2.cvtcolor(image, parameter)
    - Parameter
        - cv2.COLOR_BGR2GRAY = converts image to grayscale
        - cv2.COLOR_BGR2HSV = convert to hue, saturation, value
            - OpenCV uses H: 0-179, S: 0-255, V: 0-255
            - HSV better for object detection
- cv2.inRange(image, lower color bound, upper color bound)
    - Bounds in the form of numpy arrays (hue, sat, val)
- cv2.moments(contours)
    - Image moments= weight averages of pixels intensities
    - M['m00'] = the area
    - Theory is complicated and not in scope

- cv2.findcontours(image, p1, p2)
    - P2 = chain aprox none or chain aprox simple
        - CAN = all boundary points stored
        - CAS = redundant points are neglected
    - P1 = relationship between one contour and the next
        - External gives outer contours
        - Tree gives full hierarchy of contours.
            - Hierarchy= shapes nest in other shapes
- cv2.canny(image, lower bound, upper bound)
    - Lower and upper bounds calculated from my cannymanipulator.py
    - Theory way to complicated and not in scope
- cv2.resize(image, size)
    - Size = tuple = (scaled width, scaled height)
- cv2.circle(image, (cx,cy), radius, (B,G,R), thickness)
- Binary image = 2 color image

"

maxtestcodecv2:

- applys edge detection to an image

cannymanipulator:

- easy way to figure out canny edge detection thresholds. (upper and low bounds)

photoboothcalibration:

- tried to convert pixels to mm using photbooth resolution, macbook display resolution, and a picture that measures about 15 cm long in real life

objectmeasurmentinMM:

- straightforward way of detecting area using canny edge detection

-resize image
-convert to grayscale
-apply canny
-find contours
-use image moments or contourArea to find the area under the curve(greens theoreom)

cannyarea:

- another way of determining area. outputs same as objectmeasurementinMM expcet uses cv2.RETR.EXTERNAL instead o cv2.RETR.TREE

dont know why one is better than the other

alternateareameas:

- doesnt really work. no point in trying to figure this one out

fills the contours drawn with a color

cameracalibrationexample:

- tried but too complicated

realtimevideocap:

- provides real time detection of area even when the object moves.

uses hsv and masks instead of canny edge detection to detect the objectbuy providing a color input

videocapturetest:

- testing video capture functions in opencv. trimming video changing size etc

uses moviepy.editor to trim video

colorthresholdframe:

- tring to understand mask upper and lower bounds hsv for detection a shape via its color to analyze area etc.

HSVmanipulator:

- allows you to change the HSV values to fit the given video objects

MOVING FORWARD
- i understand how to find area, perimeter and centroid
- ned to figure out if canny can be used in replace of HSV
- have an idea of where to go to calibrate and turn pixels into mm. will depend on camera parameters, focal length and a bunch of other stuff
- need to figure out if only one droplet needs to be in frame at a time.
- determine if

1)turning pixels into mm
2)getting 1 droplet per frame

'''
0. CV_CAP_PROP_POS_MSEC Current position of the video file in milliseconds.
1. CV_CAP_PROP_POS_FRAMES 0-based index of the frame to be decoded/captured next.
2. CV_CAP_PROP_POS_AVI_RATIO Relative position of the video file

3. CV_CAP_PROP_FRAME_WIDTH Width of the frames in the video stream.
4. CV_CAP_PROP_FRAME_HEIGHT Height of the frames in the video stream.
5. CV_CAP_PROP_FPS Frame rate.
6. CV_CAP_PROP_FOURCC 4-character code of codec.
7. CV_CAP_PROP_FRAME_COUNT Number of frames in the video file.
8. CV_CAP_PROP_FORMAT Format of the Mat objects returned by retrieve() .
9. CV_CAP_PROP_MODE Backend-specific value indicating the current capture mode.
10. CV_CAP_PROP_BRIGHTNESS Brightness of the image (only for cameras).
11. CV_CAP_PROP_CONTRAST Contrast of the image (only for cameras).
12. CV_CAP_PROP_SATURATION Saturation of the image (only for cameras).
13. CV_CAP_PROP_HUE Hue of the image (only for cameras).
14. CV_CAP_PROP_GAIN Gain of the image (only for cameras).
15. CV_CAP_PROP_EXPOSURE Exposure (only for cameras).
16. CV_CAP_PROP_CONVERT_RGB Boolean flags indicating whether images should be converted to RGB.
17. CV_CAP_PROP_WHITE_BALANCE Currently unsupported
18. CV_CAP_PROP_RECTIFICATION Rectification flag for stereo cameras (note: only supported by DC1394 v 2.x backend currently)

can i write each frame to a directory and then read each from from that directory - solved

can i just append each fame to a list within the program and then loop through that list frame by frame - solved

can I just loop through an interval of frames from reading the video and anylze them that way - solved

can i have it autmaticlly pick out which frames the droplet is in based on fps, #frames, speed of droplet, camera FOV yes i can but how to do it automatically is the question - solved

How to detect a droplet moving past a line to count how many droplets, frequency and even space between droplets? - solved
####

the basic idea is that ur take frames per second, figure out how far the object moves from on frame to the other. - solved
we know the distance between droplets so we can calculate how many frames till the next droplet is in the middle of th frame - solved

the next challenge is to determine a consistent way of applying edge detection to find contours for each frame - solved