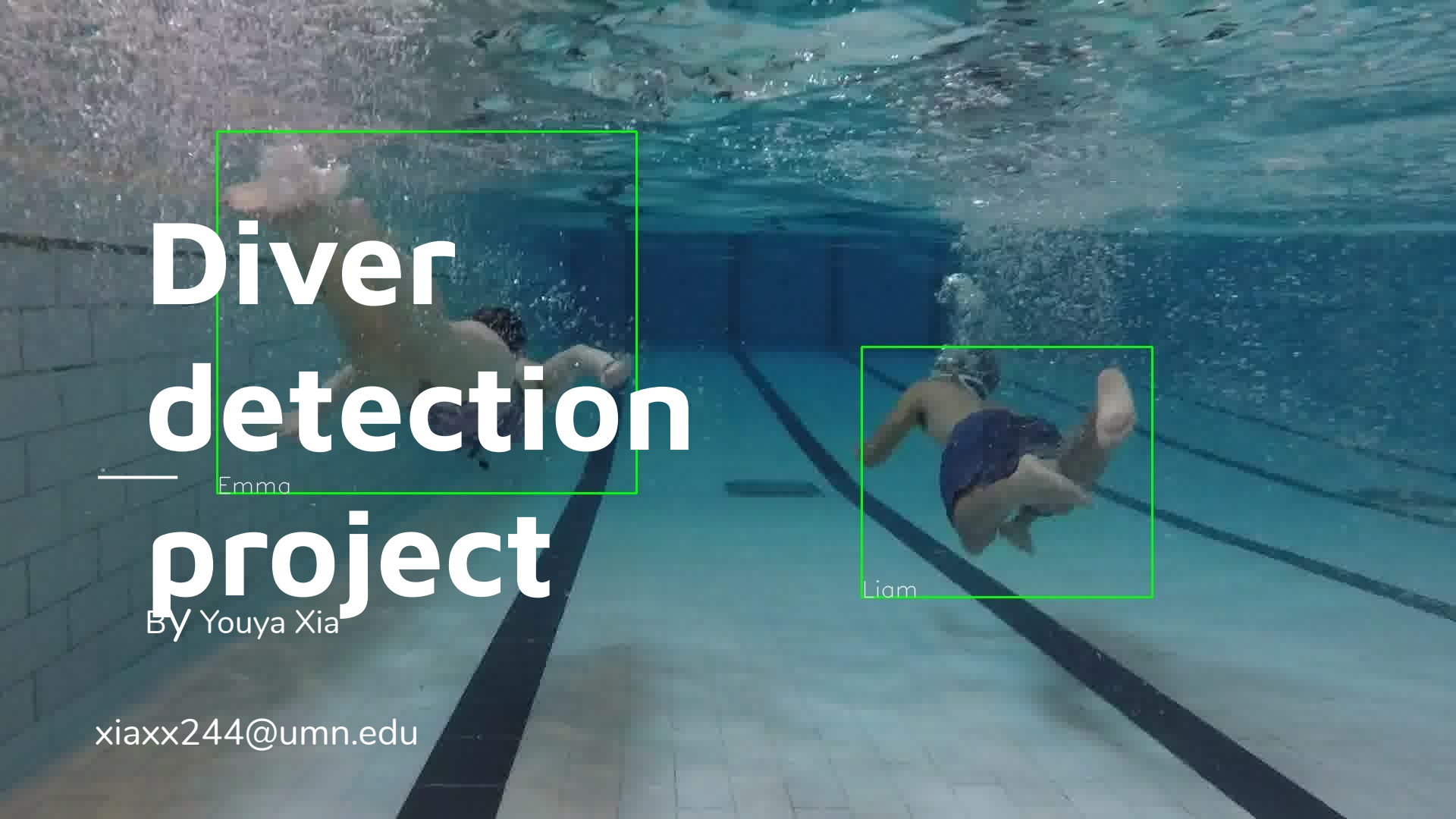


Diver detection project

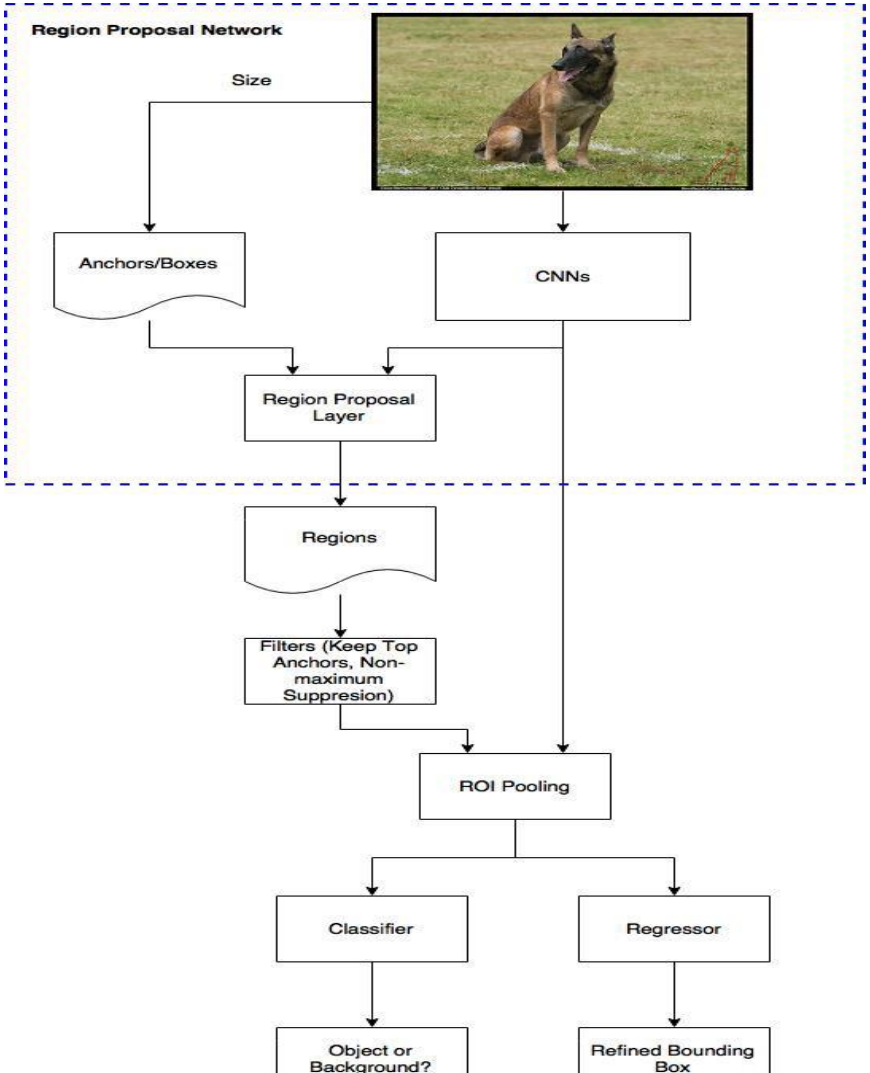
By Youya Xia

xiaxx244@umn.edu

1.background introduction

(1):importance of knowing who's who underwater:Since in the ocean or deep water,it may take a long time in order to find the diver who need to be extricated,which lead to the tenuous surviving possibility of divers.Thus,if a robot can follow a specific diver,it can sends the diver's location back to the rescue team,so that they can save the diver's life quickly.

(2):the biggest challenge in this project:lack of sufficient features to detect!



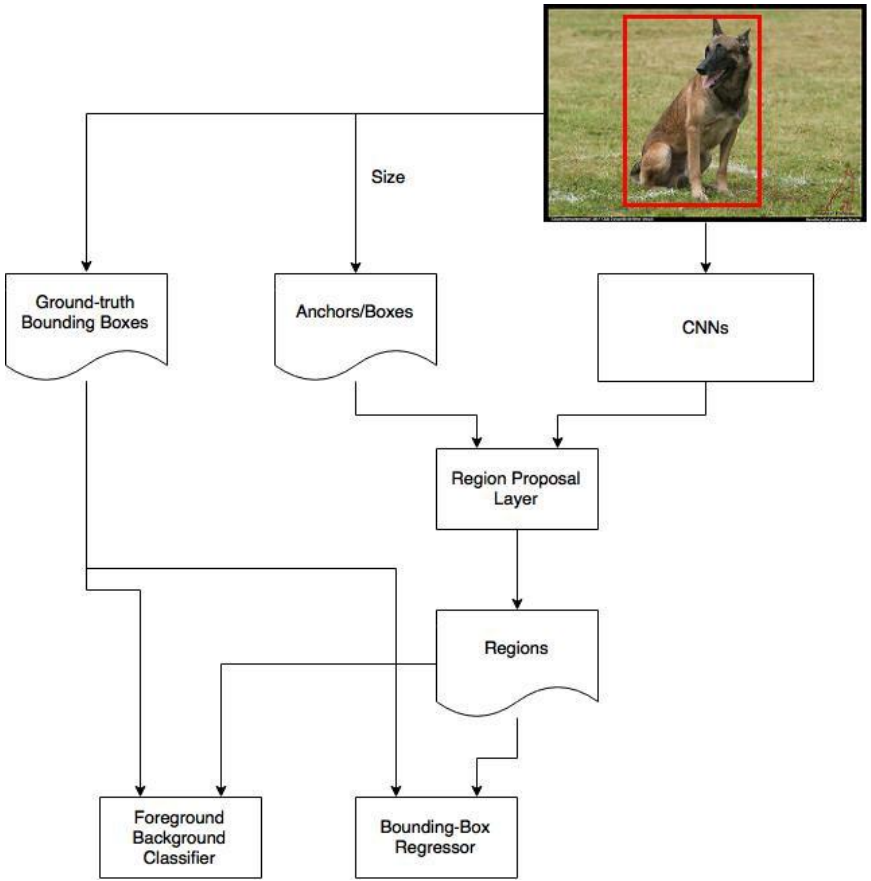
2.My approach to tackle this problem-(1):general diver detection

Faster RCNN;:region proposal network (RPN) for generating region proposals and a network(fast RCNN) using these proposals to detect objects

Pass the bounding box generated by RPN to fast-RCNN to generate a classification and tightened bounding boxes.

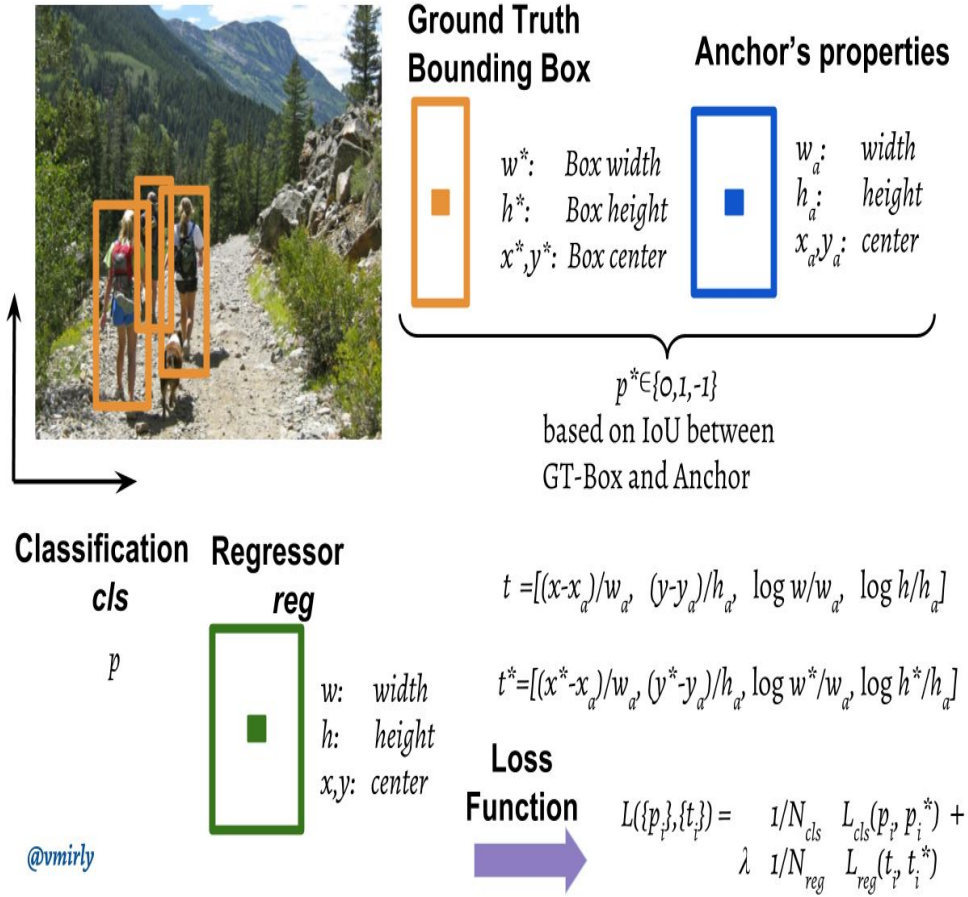
A cnn model

First layer:
A filter (which acts as a feature identifier) is sliding(convolving) around the input image to multiply the value in the filter with the pixel value in the original image,then we sum the all nonzero values in the matrix which we get from multiplication to just a single number and repeat the whole process until we get an array to represent the image.We call the array a feature map or an activation map
other layers :are used to control overfitting and improve the robustness of the network
Last layer:fully connected layer(takes an input volume and outputs N dimensional vector and N is the number of classes)(look at the previous layer and determine which features are most correlated to a class)



RPN:Faster RCNN adds a fully convolutional neural network on top of the features of cnn.

How it works:1.go through a convolutional neural network(which offers a set of convolutional feature maps on the last layer).2.pass a sliding window over cnn's feature map-output k potential bounding boxes(anchor box) and give each box a score(anchor box is designed to find the best box that can fit the object tightly)(a value p* is computed to estimate how much an ancho box is overlapping with a ground-truth box)

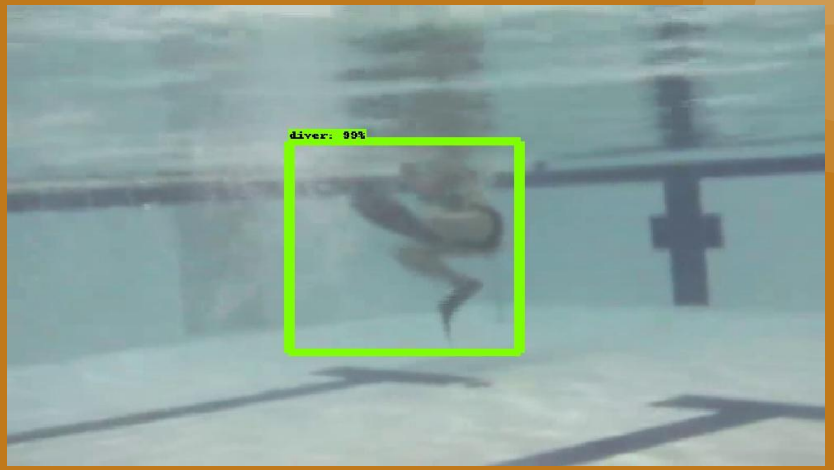


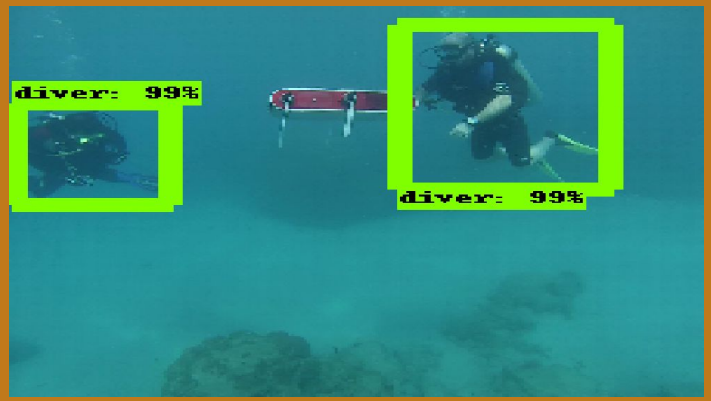
3.finally,the n*n feature maps(e.g. 3*3) are fed into a smaller network which has two purposes classification and regression.

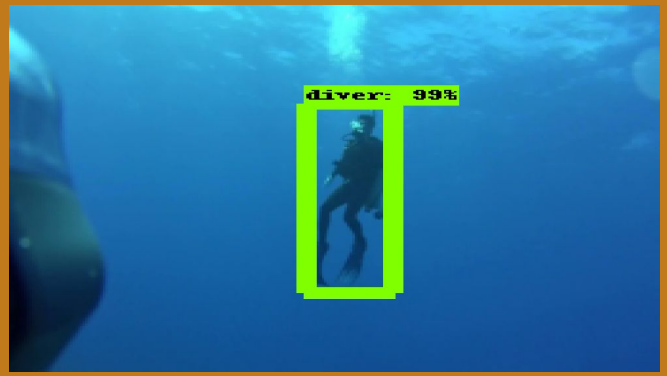
Output regressor :determine the bounding box.

Classification:calculate the probability that the bounding box really contains a object(confidence)

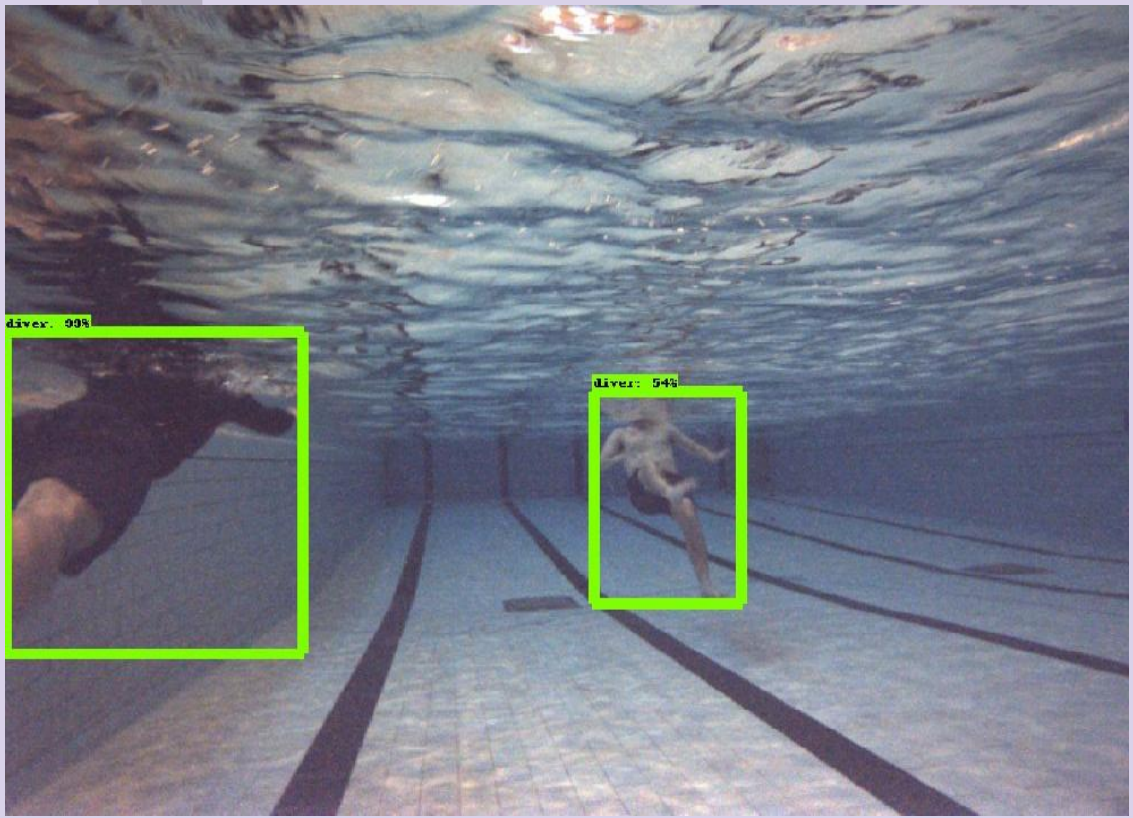
Sample results I get in this step:



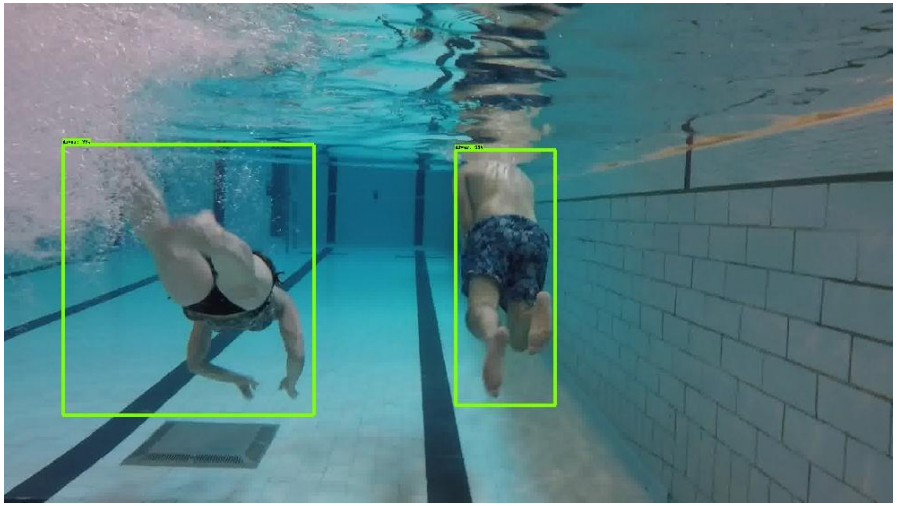


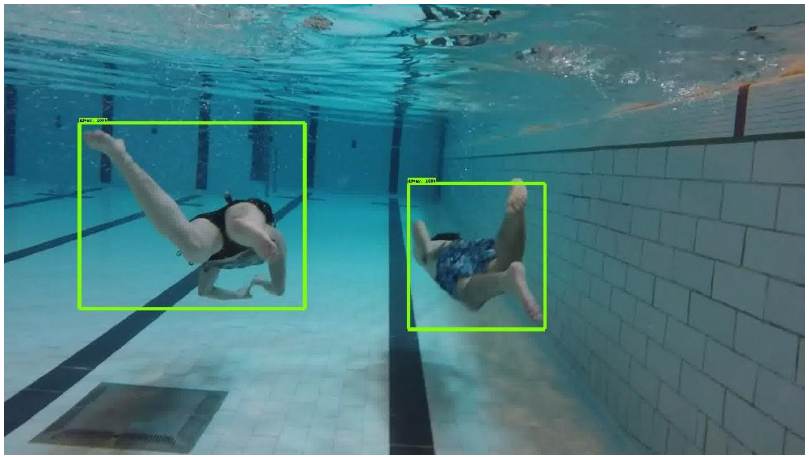


(2):feature extraction-color(rgb value) intensity



Many divers have different complexion,outfit which may lead to their different rgb value appears in the image,therefore,I use color as a feature(I split the diver image into four regions and use the color intensity value in each of the four regions)





Remove water from the image

Notice that since the in each box that we get,we inevitable incorporate water in the box,which can act as a distract from the feature extraction of the image,especially when we using the color.Thus,I split the image into b,g,r value respectively,and only considering g and r value when using color as a feature.

the moment's invariants

$$I_1 = \eta_{20} + \eta_{02}$$

$$I_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].$$

where η_{ji} stands for $\texttt{Moments::nu}_{ji}$

These are well-known as *Hu moment invariants*.

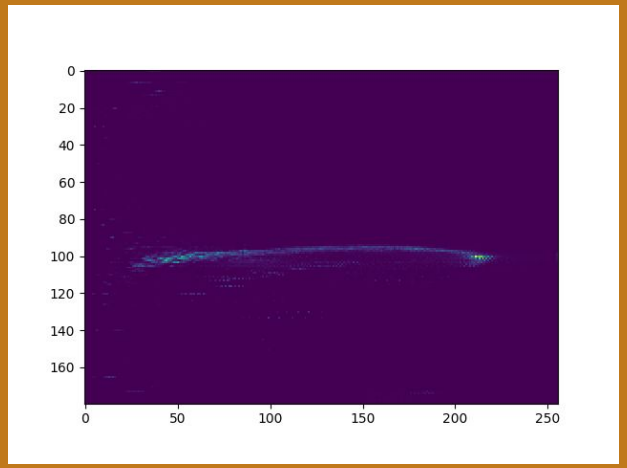
The first one, I_1, is analogous to the moment of inertia around the image's centroid, where the pixels' intensities are analogous to physical density. The last one, I_7, is skew invariant, which enables it to distinguish mirror images of otherwise identical images.

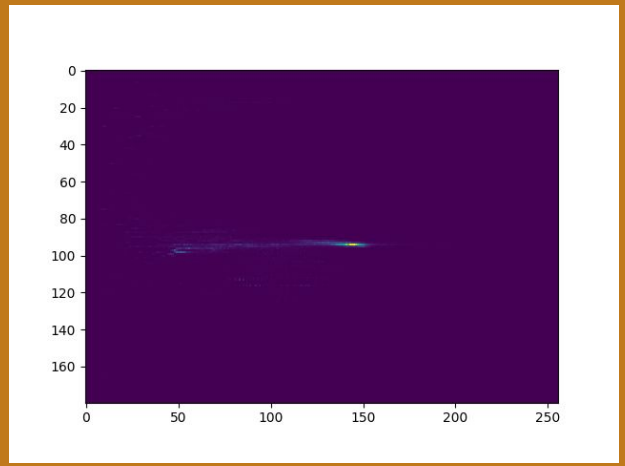
--wikipedia

Shape approximation

canny edge detection:filter noise(Gaussian filter)-find the intensity gradient of the image-non maximum suppression-threshold used-find contors-approximate each contour as a polygon

Since the shape of the diver maybe different,thus,I find the contour of each diver and use the contour to approximate the shape of the diver.In my implementation,for every contour of the diver,I approximate each to be a polygon and compute the number of vertices each contour has.finally compute the average of the vertices each contour has and use it as a feature.





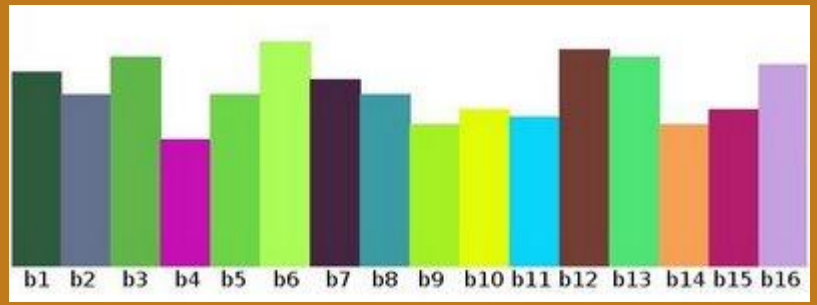
Histogram from different divers from the previous two pool trials

Find histogram of the diver

1.definition:the histogram is the intensity distribution of an image.It can offer us the information about the contrast,brightness,intensity distribution from the histogram(dims,bins,range)

2.how it works:separate the image into subins for example

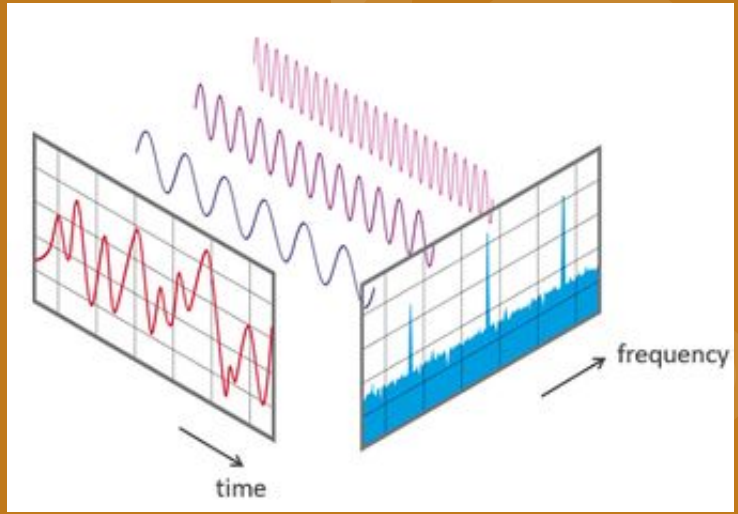
$$[0, 255] = [0, 15] \cup [16, 31] \cup .... \cup [240, 255]$$
$$\text{range} = \text{bin}_1 \cup \text{bin}_2 \cup .... \cup \text{bin}_{n=15}$$



Fast-Fourier transform

$$x(t) = A\sin(2\pi f t)$$

What is fast fourier transformation:
Simply put,it is a recipe that tells you the exact ingredient you need in order to reconstruct the original signal.
It transforms a function of time into a function of frequency.



Ideas based on the paper:
Understanding Human motion and Gestures for Underwater Human-Robot collaboration(written by Jahidul Islam,Marco Ho,Junaed Sattar)
Visual Identification of biological motion for underwater human-Robotics Interaction(written by Junaed Sattar and Gregory Dudek)

Discrete fourier transform

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{2\pi i}{N} kn}$$

$$= \sum_{n=0}^{N-1} x_n \cdot [\cos(2\pi kn/N) - i \cdot \sin(2\pi kn/N)],$$

Disadvantages of DFT:computationally expensive and slow!

An example of DFT:

Let $N = 4$ and $\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2-i \\ -i \\ -1+2i \end{pmatrix}$. Here we demonstrate how to calculate the DFT of $\mathbf{x}$ using **Eq.1**:

$$X_0 = e^{-i2\pi 0 \cdot 0/4} \cdot 1 + e^{-i2\pi 0 \cdot 1/4} \cdot (2-i) + e^{-i2\pi 0 \cdot 2/4} \cdot (-i) + e^{-i2\pi 0 \cdot 3/4} \cdot (-1+2i) = 2$$

$$X_1 = e^{-i2\pi 1 \cdot 0/4} \cdot 1 + e^{-i2\pi 1 \cdot 1/4} \cdot (2-i) + e^{-i2\pi 1 \cdot 2/4} \cdot (-i) + e^{-i2\pi 1 \cdot 3/4} \cdot (-1+2i) = -2-2i$$

$$X_2 = e^{-i2\pi 2 \cdot 0/4} \cdot 1 + e^{-i2\pi 2 \cdot 1/4} \cdot (2-i) + e^{-i2\pi 2 \cdot 2/4} \cdot (-i) + e^{-i2\pi 2 \cdot 3/4} \cdot (-1+2i) = -2i$$

$$X_3 = e^{-i2\pi 3 \cdot 0/4} \cdot 1 + e^{-i2\pi 3 \cdot 1/4} \cdot (2-i) + e^{-i2\pi 3 \cdot 2/4} \cdot (-i) + e^{-i2\pi 3 \cdot 3/4} \cdot (-1+2i) = 4+4i$$

$$\mathbf{X} = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} 2 \\ -2-2i \\ -2i \\ 4+4i \end{pmatrix}$$

$$X_k = \sum_{i=0}^{N-1} x_i W_N^{ik}, \quad k = 0, \ldots, N-1, \quad W_N = e^{-j2\pi/N}$$

$$X(z) = \sum_{i=0}^{N-1} x_i z^{-i}. \quad z = W_N^{-k}:$$

$$X(z) = \sum_{i=0}^{N-1} x_i z^{-i} = \sum_{l=0}^{r-1} \sum_{i \in I_l} x_i z^{-i},$$

$$X(z) = \sum_{l=0}^{r-1} z^{-i_{0l}} \sum_{i \in I_l} x_i z^{-i+i_{0l}}.$$

Start from general divide and conquer

Keep periodicity compatible with periodicity of the input sequence

Use decimation

$$I_{n_l} = \{n_2 N_1 + n_1\},$$

$$n_1 = 0, \ldots, N_1 - 1, \quad n_2 = 0, \ldots, N_2 - 1,$$

$$N = N_1 \cdot N_2. \quad \{x_i | i = 0, \ldots, N-1\} \quad \{x_i | i \in I_l\}$$

$$X(z) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_2 N_1 + n_1} z^{-(n_2 N_1 + n_1)}$$

$$X(z) = \sum_{n_1=0}^{N_1-1} z^{-n_1} \sum_{n_2=0}^{N_2-1} x_{n_2 N_1 + n_1} z^{-n_2 N_1},$$

$$X_k = X(z)\big|_{z = W_N^{-k}}$$

$$= \sum_{n_1=0}^{N_1-1} W_N^{n_1 k} \sum_{n_2=0}^{N_2-1} x_{n_2 N_1 + n_1} W_N^{n_2 N_1 k}.$$

$$W_N^{iN_1} = e^{-j2\pi N_1 i/N} = e^{-j2\pi i/N_2} = W_{N_2}^i$$

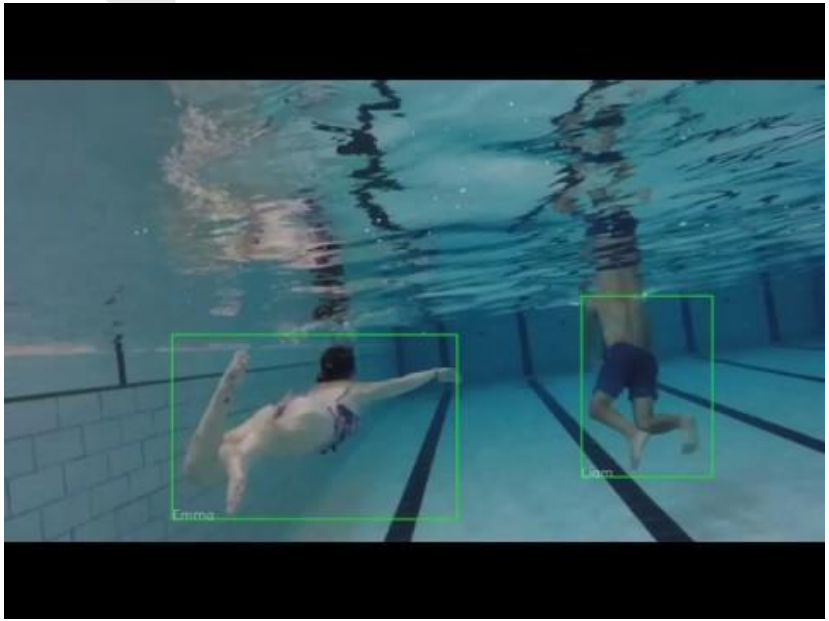
$$X_k = \sum_{n_1=0}^{N_1-1} W_N^{n_1 k} \sum_{n_2=0}^{N_2-1} x_{n_2 N_1 + n_1} W_{N_2}^{n_2 k}.$$

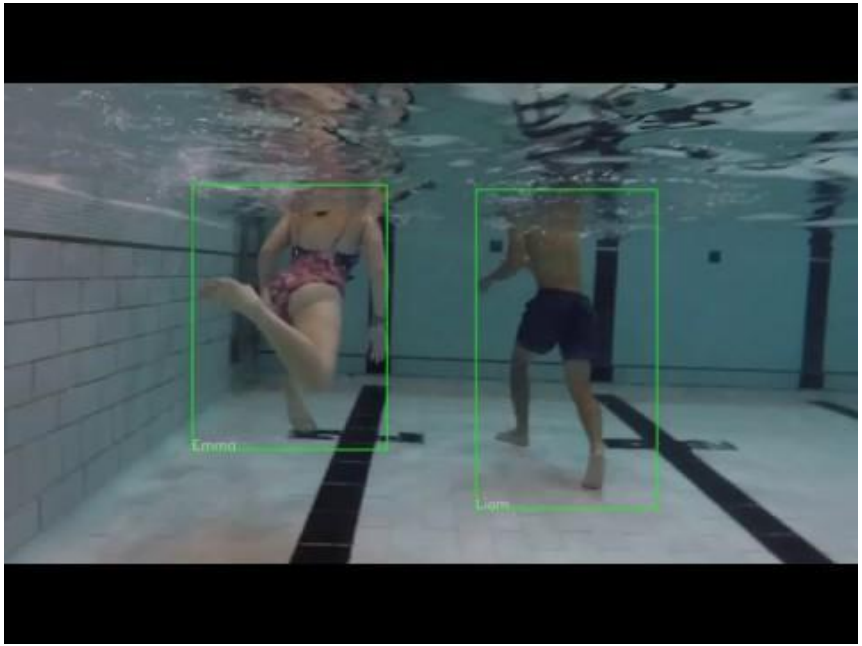
almost N_1 DFTs of size N_2

An example algorithm of FFT(Cooley-Turkey)

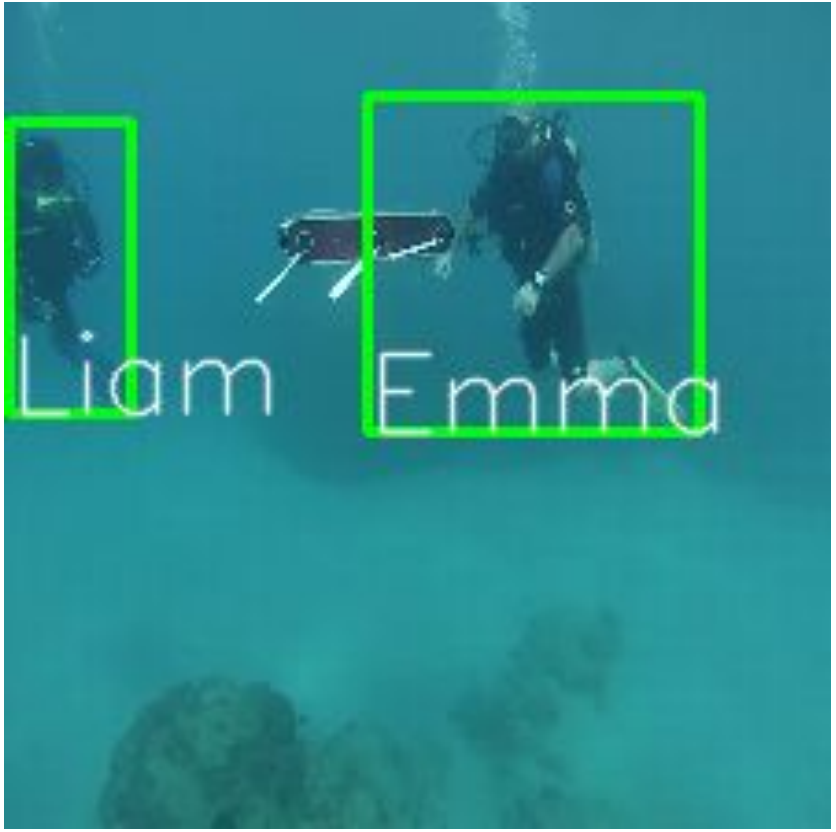
Last step using k-means clustering for training(the k is defined to be the maximum number of diver appears in all of the frames!)

Results I get:






Liam
Emma


Liam
Emma

Feature work:(in all of these three situations sometimes the precision rate tend to fall from 85 percent to around 60 percent)

1.sometimes,bubbles in the water can undermine the credibility of the features I extract.

2.when two divers' appearances are almost the same(i.e.,their outfit,the color in their body,their center of mass and the shape of their body),my method tend to fail and get wrong results.

3.in the image with low quality or in the image where divers appear too far from the robot so that there are very limited feature to extract,in this case,the failure rate tend to be higher than other situation.

Source:

https://www.quora.com/How-does-the-region-proposal-network-RPN-in-Faster-R-CNN-work

http://cs231n.stanford.edu/reports/2017/pdfs/112.pdf

https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_calculation/histogram_calculation.html

https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html

https://en.wikipedia.org/wiki/Image_moment

https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-973-communication-system-design-spring-2006/lecture-notes/lecture_8.pdf