Comparison of Drowning People Detection Systems for Water Safety

Yunsu Han

1 Introduction

There are many people who go on vacation every summer. The number of vacationers is usually the maximum from July to August, and the number of people who visit the beach and play in the water is increasing every year. Unfortunately, the number of water accidents during the summer vacation season in Korea is gradually increasing. In particular, safety accidents occur in July and August when vacationers flock to Haeundae Beach. According to the Ministry of Public Administration and Security, 165 casualties have occurred in the water over the past five years, especially in the July-August period (Cho, 2020).

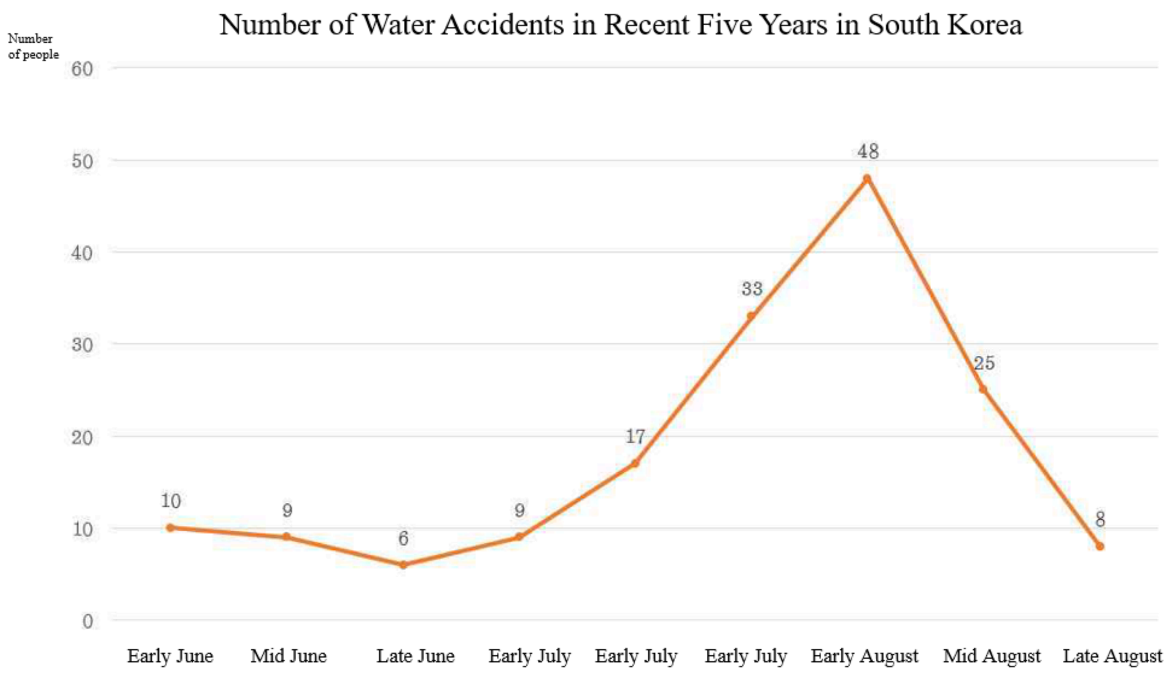

Figure 1. *A graph of the number of water accidents in recent five years in Haeundae Beach, South Korea (Cho, 2020).*

The government has made numerous efforts to reduce such accidents, but it seems difficult to reduce them for various reasons, including a lack of lifeguards. The average number of people visiting Haeundae Beach during the summer vacation season is 77,875, compared with only 48 rescuers who are not 100 percent fully trained because some are part-time workers. In other words, a lifeguard has to endure nearly three months, taking care of 1,622 people a day. As water accidents continue to occur, it is important to detect and prevent further damage before it occurs.

There are several methods that detect drowning. For example, Kanchana et al. (2017) created an automated drowning detection device that uses microcontrollers and pressure sensors, and the RF device detects if the swimmer with the device on his or her wrist stays inside the water for too much time. Also, Prakash (2018) employed a computer science technique called Novel Equations, also known as NEPTUNE, which is an early prediction model with specific equations that calculate the time that a swimmer is inside the water. Moreover, Salehi, N. et al. (2016) implements another computer vision model that adapts active contours, detecting a swimmer real-time and checking whether the swimmer is in danger.

The methods introduced above are valuable and could be employed in real life situations, but they also lack accuracy and have alarm delays. Therefore, this paper discusses several systems that detect drowning with some new methods and compares their advantages and shortcomings in order to find out the probable method to be employed in real-life situations.

2 Background

This paper compares two different methods: Classic Contour Detection and Deep Learning using Convolutional Neural Networks. This section discusses how these methods work in detail.

2.1 Classic Contour Detection

Classic Contour Detection is a well-known skill in computer vision. The program finds and draws the contours, which are curves that join continuous points with the same color or intensity. These contours are used for detecting an object or analyzing the shape of an object. In order to find the contours more accurately, the image must be pre-processed, such as by applying a threshold and canny edge detection.

OpenCV’s findContours() function finds the contours of an object on a black-and-white image in which the object is white and the background is black. The function requires three parameters, which are the source image, contour retrieval mode and contour approximation method. Contour retrieval mode is the way of finding the contour and constructing the hierarchy. A contour approximation method is the way of approximating the contours which stores the points that can draw the contours line. The function returns the contours and the hierarchy, which explains the parent-child relationship between several contour lines.

After finding the contours, the system must draw the contours to visually represent the result. The function cv2.drawContours() takes the image, contour, contour index, color of the contours, and the thickness of the contours and draws the contour on the image.

Figure 2. *Contours drawn on an image of coins. (Rosebrock, 2015)*

2.2 Deep Learning

Deep Learning is one of the machine learning technologies that has greatly improved the original artificial neural network of the 1950s and has made significant contributions to image and video detection since the 2010s. Artificial neural networks go through a process of interpreting images similar to that of humans. A perceptron, which acts as a neuron within the human nervous system, has a single layer and creates a structure in which those layers are connected to each other. This is called a multi-perceptron. A multi-perceptron is a type of neural network consisting of a multi-layered structure with an input layer, an output layer, and hidden layers in between. Figure 4 shows the structure of layers that consist of multi-perceptrons. In addition, this neural network is also called the Deep Neural Network, also known as DNN, in the sense that it is deeper than the original neural network since there exists a hidden layer between the input and output layers.

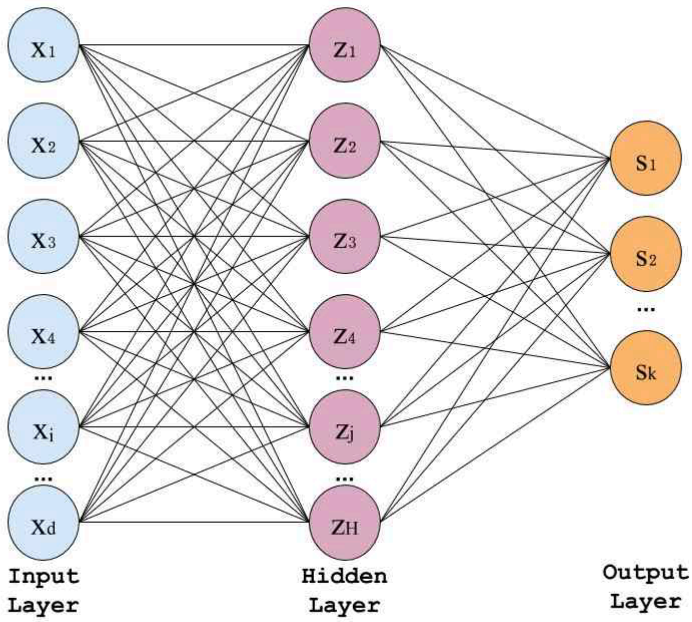

Figure 4. *Deep Neural Network with an Input Layer, Output Layer, and Hidden Layer.*

In these neural networks, nodes represent the input and output values of each layer and are linked to each other with weight values. When input data are applied to the input node, the node becomes active and receives a specific weight. The node applies the specific weight value to the input value and sends the output result to the next node. In the next node, multiple inputs determine whether to activate, which then affects the next node. Continuing through this process to the last node at the end determines whether the final output node is activated. Overall, the weight value between nodes determines the nature of the neural network.

According to Aggarwal's "Data Classification: algorithms and applications," in the 1980s and 1990s, deep learning was not popular due to the limitations in hardware performance. However, entering the modern age, hardware performance has significantly improved, and so people have started to employ deep learning technology in several disciplines. Additionally, since DNN performs better as the number of data increases, and it is easier to access and collect data, it has become better to use those neural networks in modern days.

Thus, if one has enough data, one can build a system with better accuracy and efficiency compared to conventional machine running techniques. Also, if a programmer has a well-filtered data set, he or she can train the system quickly regardless of the type of data.

In this paper, I attempt to collect a large amount of data and train the system to realize the pattern and characteristics of people who are in danger and those who are not using Convolutional Neural Networks (CNN).

2.4 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are often used for image processing. For example, suppose there is a picture with a number 8, and the system wants to print out that picture that has the number 8. To do so, one may create a system with DNN. The system can numerically represent the level of the darkness for each pixel of an 18x18 black-and-white picture. However, after training the system with such data, a small change in the position of the numbers creates a problem: if the number 8 is at the top left, not in the middle, the system does not recognize the number anymore. To feed those image data is also a method to train the system, but the amount of data is so immense that it takes too much time to train the system. CNN can resolve this problem.

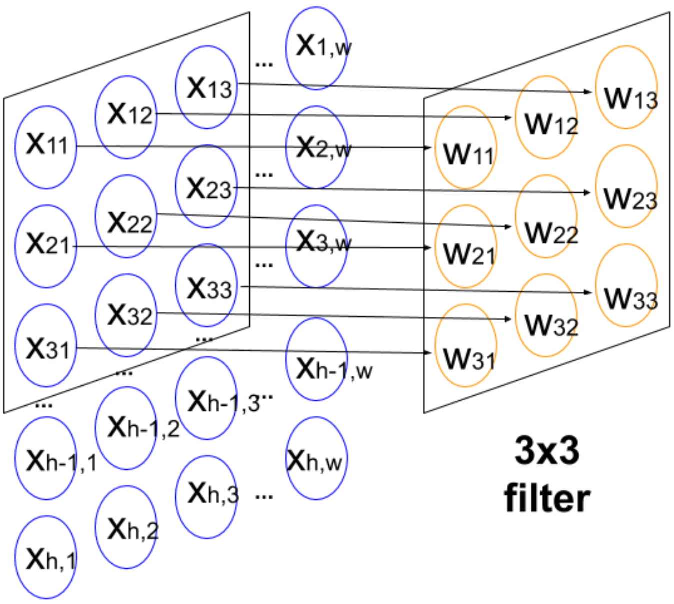

Figure 3. *Convolutional Neural Network applied with a 3x3 filter.*

While DNN only converts each pixel's brightness into numerical values to grasp the features of the image data, a convolutional layer also keeps track of the relationships of the numerical values of the image data. The network has a small filter, multiplies the weight value on the overlapped portion of the data, and applies activation functions. In the case of the above image, a 3x3 filter multiplies the weighted value to each node and adds them up. Later on, the system calculates the weighted value with an applied activation function and hands the result to the next filter. The filter slightly moves until it covers the whole area of the data. This process is called Convolutional Neural Network, also known as CNN. We also call CNN that has more than one hidden layer a Deep Convolutional Neural Network (DCNN).

A CNN trains by feeding some data and the correct output that the programmer wants the network to show as a result after processing the data. After showing the result, the program compares the output and the correct answer and then uses gradient descent to update the weights and filter values to make the result closer to the desired answer. As a result, the program performs better when it is fed more data since the program is able to train more times with new data. In the end, the system needs to be generalized so that it could be employed in images that it has never seen before.

A CNN becomes practical in analyzing data with two-dimensional features, such as image data. The most popular neural network among image classification systems is CNN, and this paper also employs that network to distinguish if a person in the water is in danger.

3 Related Works

3.1 HSV Color Space Threshold

HSV color space analysis from “An Automatic Video-based Drowning Detection System for Swimming Pools Using Active Contours” (Salehi, Keyvanara & Monadjemmi, 2016) explains why HSV color space is the best for detecting drowning among numerous color spaces. The information of several layers in the HSV color space (V channel containing the luminance information of the input image and H and S channel containing the chromaticity information) allows the system to easily track the segmentation of the person in the water. The transformation from RGB color space to HSV color space helps this paper’s classic contour detection to easily find the swimmer and draw contours on him or her.

3.3 Image Classification Based on CNN

Jung, H. et al. (2018) proposes a system that captures the condition of crops to enhance the productivity of smart greenhouses. The research employs CNN to find the reason why powder fungus is formed on tomatoes and uses the image-fusion technique to classify images into different groups. As a result, the algorithm that distinguishes the fungus on tomatoes shows an accuracy of 93.02% for 43 test videos. This paper also employs CNN, which has a similar process with this research by introducing image extraction, conversion, and overlapping. Also, the paper is similar to this system in the way that it also determines if their subjects are in danger or not.

4 Data and Comparison of Methods

This section introduces how the data are collected and a couple of methods -- image classification based on CNN and classic contour detection -- that detect drowning in the water, explains the detailed structure and process of data and compares their strengths and weaknesses.

4.1 Data

In order to feed some images into the CNN system, this paper needs a number of images of both people drowning and swimming in the water, which are used as training and testing datasets for the system. Therefore, I took pictures at beaches to collect images that show people naturally swimming and drowning. However, it was difficult to take pictures carelessly, and so I took pictures and collected a variety of images by searching on the Internet as well. I went to websites that contained downloadable images such as Google Images, Adobe Stock, Getty Images, and Deposit Images, and selected and downloaded images and videos that are realistic, not having watermarks on them. Video files are converted to png files once every 30 frames, using a video filter of a program called VLC. The following table shows the amount of image data:

	Train	Test
Swimming	1154	332
Drowning	908	316

Table 1. *Number of Data Sets of Images*

In this paper, I cropped the images into 32x32. It is true that the bigger the image data are, the easier it is to capture features of people in danger and those not in danger. But, considering that the images are 3-channel, which means they are colored, training the model using images with high resolution and quality requires a lot of time and hardware with extensive specifications. Therefore, I resized the image into 32x32, which is the size equal to CIFAR-10, a data set that CNN beginners study with. Since CIFAR-10 proves that even small images can distinguish objects, a small dimension can also classify people drowning and swimming.

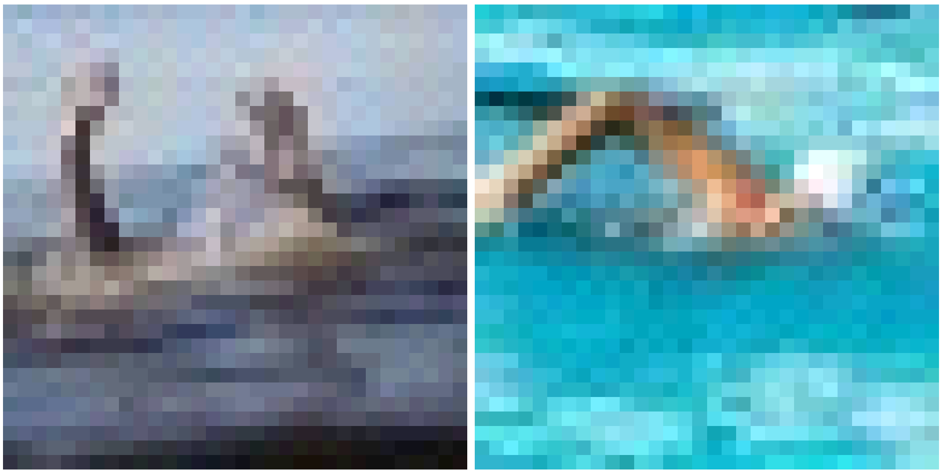

Figure 4. *Two examples of images of people drowning and swimming (Madhavan & Hamburg, 2013).*

To compare the two contour models to the CNN system, the contour models do not need a large dataset of still images, but rather video frames. Therefore, I used two videos of a person swimming and staying inside the water for a long time provided by Maryam Keyvanara, who has written “An Automatic Video-based Drowning Detection System for Swimming Pools Using Active Contours.” The two MPG videos are about a minute and a half long, and the frame rates are both 25 frames per second. The following figure shows two specific frames from one of the videos.

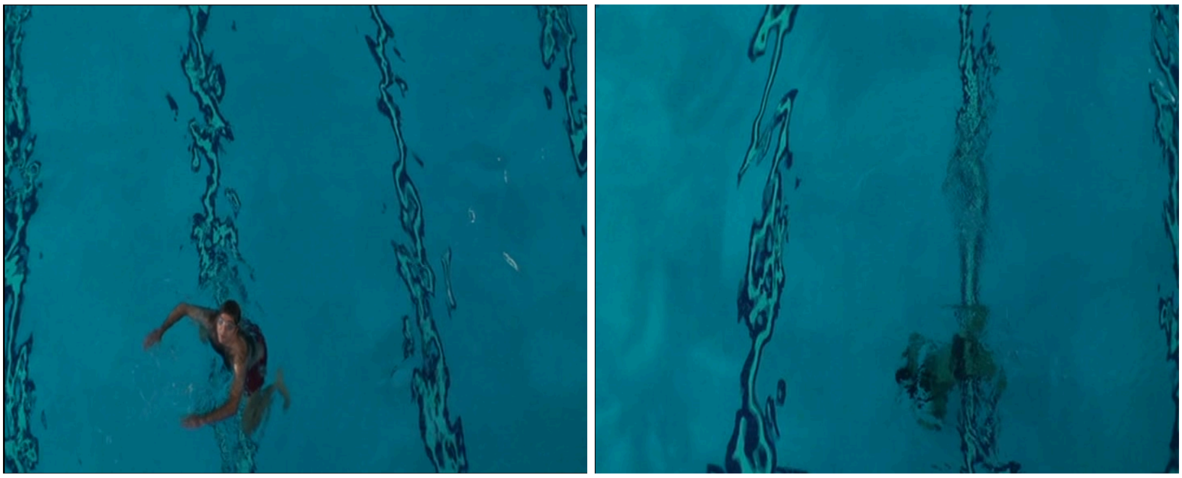

Figure 5. *Two screenshot images of one of the videos used for contour detection (Keyvanara et al., 2016). The left frame is when a swimmer is not in danger, and the right frame is when a swimmer is possibly in danger.*

4.2 CNN Image Classification

CNN has the advantage that the model's accuracy does not drop even though the system increases the number of hidden layers. Additionally, since CNN extracts features by putting an entire image as input data, whereas traditional DNN does so from an image by pixel by pixel and passes the vector of feature values to the network, CNN is the best deep learning algorithm for this experiment.

CNN used in this paper consists of Convolutional Pooling Layers, which extract the characteristics of the image data, and Fully Connected Layers, which classify the data using the information received from the Convolutional Pooling Layers. In this paper, I apply Convolutional Pooling Layers, which consist of two separate layers, a Convolutional Layer and a Pooling Layer, to the initial 32x32x3 data with two 3x3 filters based on Max Pooling. Then, the Fully Connected Layers classify the image data into different classes by utilizing the data passed from the prior layers. Figure 6 is an overall diagram of the CNN model employed in this paper.

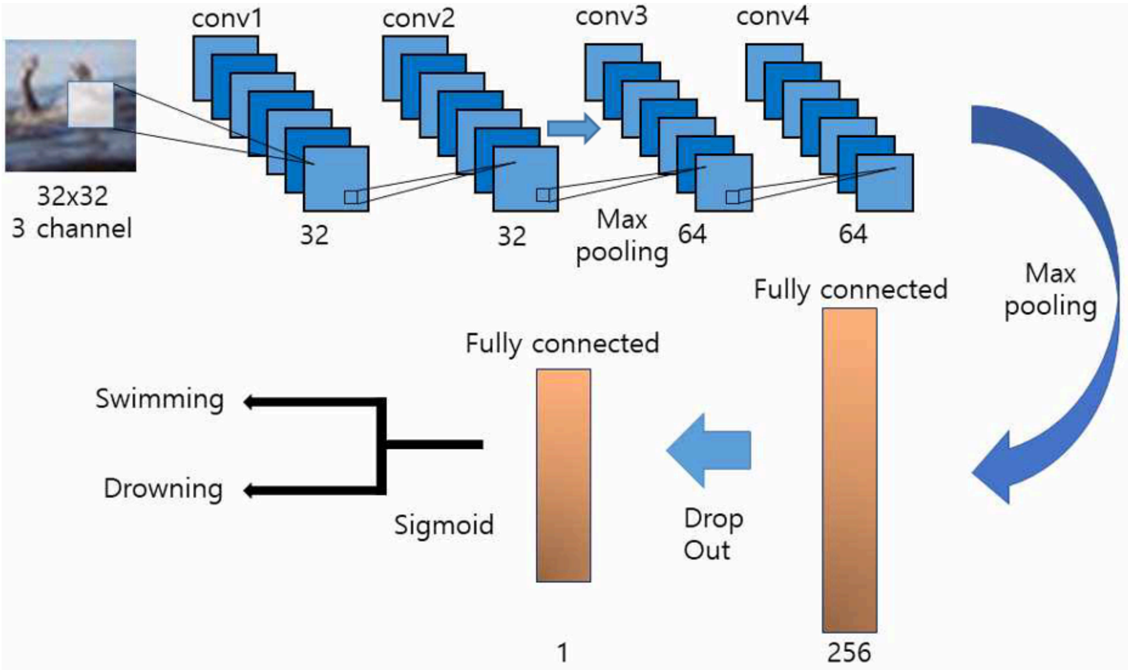

Figure 6. *Overall diagram of the CNN Image Classification system.*

The experiment uses the Python language, CPU of Intel i7-7700, and GPU of Geforce GTX 1060 6GB for the environment. It also uses Tensorflow version 2.0's Keras to construct and train the model and Keras's ImageDataGenerator class to increase data sets needed to train the model.

Keras's ImageDataGenerator class performs data augmentation, which helps the program to increase the generalizability of the model by creating new training samples from the original images. The class translates, rotates, changes in scale, shears, and flips the original images and allows the program to find data points that are not included in the original training set.

In addition, I use the EarlyStopping function and look at the test dataset's loss value to prevent overfitting, which is a modeling error that the model learns too much about the training datasets that it fails to generalize to other data, while training the model. If the loss value does not decrease, the function automatically terminates the training process.

In the compiling process, the system uses adam as its optimizer, and runs over 500 epochs.

4.3 Normal Contour Detection

Normal Contour Detection is also a way to detect a person in the water and check whether the swimmer is drowning. Finding contours is a built-in operation in OpenCV. My program first transforms the color space from RGB to HSV. In order to increase the accuracy of finding the correct object, the system thresholds the image: the system finds the dominant color of the frame by using k-means clustering to create a palette with the two most representative colors in the frame and inRange by adding a default range to the dominant color, and the system morphs the thresholded frame with morphologyEx().

The program then uses a function called findContours() with parameters of a way that finds the contours as cv2.RETR_LIST, which finds all the contour lines but does not form a hierarchy structure, and an approximation method when finding the contours as cv2.CHAIN_APPROX_NONE, which saves all the contours point. The program then draws several contours, which are a convex hull and an enclosing rotated rectangle. A convex hull is a set of points that forms the smallest convex polygon, and an enclosing rotated rectangle is a rectangle that bounds the minimum area around the convex hull.

Figure 7. *A convex hull in cyan and enclosing rotated rectangle in yellow (Keyvanara et al., 2016).*

The program only draws the contours if the area found was greater than 4,000 pixels and the area is smaller than half of the entire frame. If the area is too small or too big, the program might detect something that is not a swimmer as a swimmer. The following diagram, Figure 7, is an overall flow chart of the system.

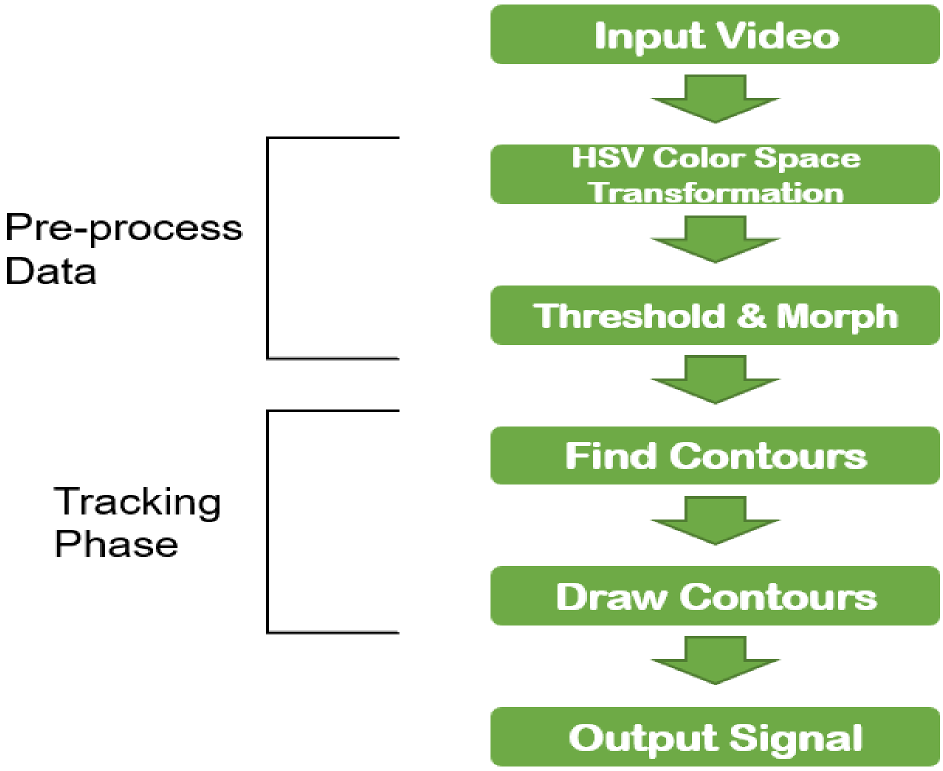

Figure 8. *The flow chart of the normal contour detection system.*

After the program detects the swimmer and draws contours around him or her, it records the number of frames that the swimmer is inside the water. If the swimmer is inside the water for too much time, which is depended by a threshold value, the system sends a signal that the person might be in danger.

5 Results and Evaluation

This section displays the results of the method explained in the previous section and discusses the advantages and disadvantages of the methods.

5.1 CNN Image Classification

This section displays the results of the experiment on the deep learning model based on a confusion matrix and its four indicators of the results, which are accuracy, recall, precision, and F1 score. Table 2 displays the confusion matrix of the test data's results. Also, Table 3 displays the confusion matrix of both the training data set and the testing data set.

	Predict Swimming	Predict Drowning
Swimming	302	30
Drowning	18	298

Table 2. *Confusion Matrix of Image Classification System*

	Accuracy (%)	Recall (%)	Precision (%)	F1 Score (%)
Train Dataset	96.03	92.36	99.21	95.48
Test Dataset	92.56	94.30	90.96	92.70

Table 3. *Accuracy, Recall, Precision, and F1 Score of Table 2.*

Table 3 proves that the system successfully classifies drowning people in their class with an accuracy of 92.6% for the testing data set. Also, it is true that the recall is greater than the precision. This means that there is a high possibility that the model classifies people not in danger as people in danger, but also means that the model successfully detects drowning people.

However, because the testing data set is not big enough, and the difference between FP and TN is not that significant, this might not conclude that the model classifies drowning people well enough.

5.2 Normal Contour Detection

The system sends an alarm signal that the person might be in danger when a person is continuously not detected for 250 frames in a video of 25 frames per second. There are three sequences when a person is inside the water for more than 250 frames, and two of the frames successfully send an alarm signal. However, there are some situations where the system makes an error, which is detecting a swimmer when he or she is underwater. The following figure shows two frames of when a person is underwater; the image on the left shows a successfully undetected swimmer, but the image on the right shows an error of the system.

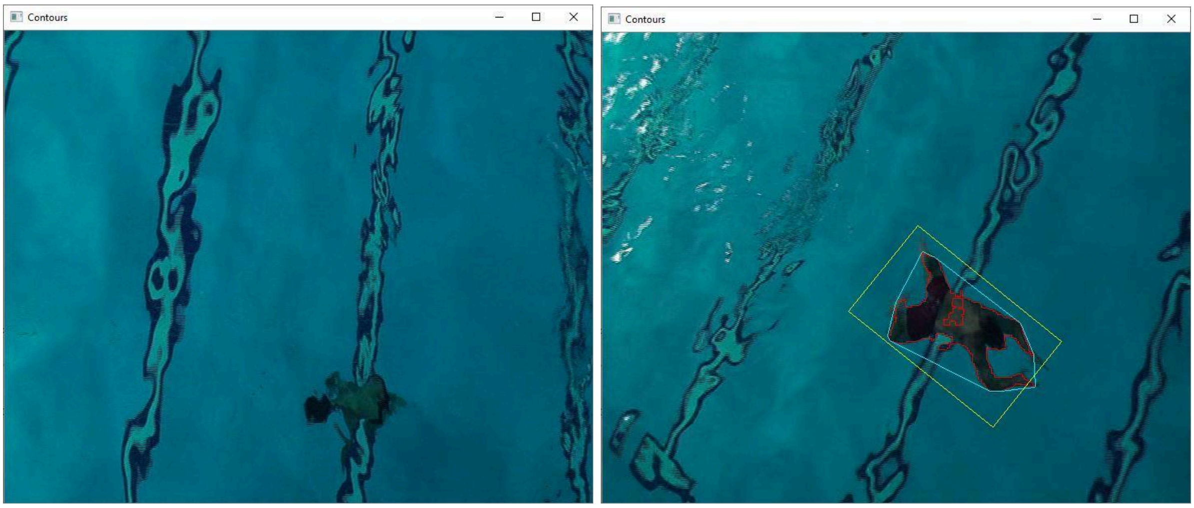

Figure 9. *Two frames of the classic contour detection system (Keyvanara et al., 2016). Left frame successfully does not detect the person underwater, but the right frame detects the swimmer although he is underwater.*

6 Discussion

Based on the deep learning-based CNN and the contour detection model, I present the classification of drowning people to prevent water accidents during the summer. I hope to expand the classification system to a more advanced one that recognizes people who are in danger in real-time by integrating the system with drones or IoT equipment in the future.

In addition, it would be better to have more image and video data for all methods so that the system has greater accuracy and there are more opportunities to check if the system works in other environments, as well.

References

Cho, J. H. (2020). Frequent Drown Accidents in Haeundae at night and early in the morning... Time and blind spot risk without rescuers. https://www.yna.co.kr/view/AKR20160809082900051.

KANCHANA, A., R, K. G., C, K., V, S., & HEGDE, S. (2017). AUTOMATED DROWNING DETECTION AND SECURITY IN SWIMMING POOL. *International Research Journal of Engineering and Technology (IRJET), 4*(6), 2938-2940.

Prakash, B. D. (2018). Near-Drowning Early Prediction Technique Using Novel Equations (Neptune) For Swimming Pools. *Computer Science & Information Technology (CS & IT)*. doi:10.5121/csit.2018.81812

Salehi, N., Keyvanara, M., & Monadjemmi, S. A. (2016). An Automatic Video-based Drowning Detection System for Swimming Pools Using Active Contours. *International Journal of Image, Graphics and Signal Processing, 8*(8), 1-8. doi:10.5815/ijigsp.2016.08.01

Rosebrock, A. (2015). Figure 7. https://www.pyimagesearch.com/2015/11/02/watershed-opencv/.

Aggarwal, C. C. (2015). 8.1 Introduction. In Data classification: algorithms and applications (pp. 206–207). essay, CRC.

Jung, H., Moon, A., An, S.-Y., & Song, Y. E. (2018). CNN-based Tomato Powdery Mildew Recognition Method. Journal of Institute of Control, Robotics and Systems, 24(7), 617–623. https://doi.org/10.5302/j.icros.2018.18.0055

Keyvanara, M. (2016) Input Video [Video Data File] Location: Keyvanara, M.